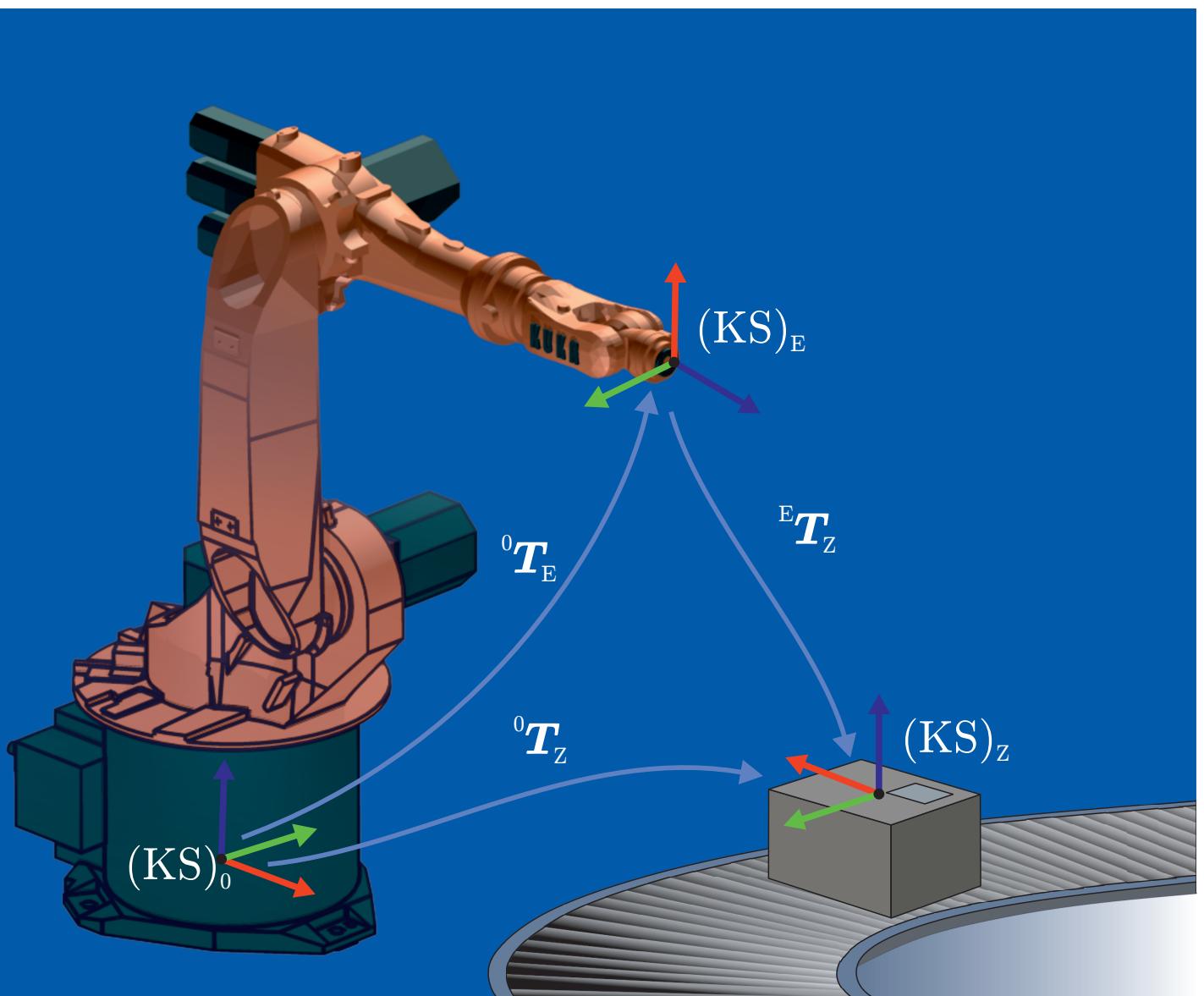


M. Sc. T.-L. Habich, Dr.-Ing. M. Schappler, Prof. Dr.-Ing. T. Seel, Prof. Dr.-Ing. T. Ortmaier

Skript zur Vorlesung Robotik I



Impressum

Herausgeber

Fakultät für Maschinenbau der Gottfried Wilhelm Leibniz Universität Hannover
Institut für Mechatronische Systeme (imes)

M. Sc. T.-L. Habich
Dr.-Ing. M. Schappler
Prof. Dr.-Ing. T. Seel
Prof. Dr.-Ing. T. Ortmaier

Adresse: Institut für Mechatronische Systeme
An der Universität 1, 30823 Garbsen

Telefon: +49 (0)511 / 762 – 4179
Fax: +49 (0)511 / 762 – 19976
Internet: www.imes.uni-hannover.de

Vorwort

Liebe Studierende,

vor dem eigentlichen Start der Vorlesung „Robotik I“ möchte ich einige persönliche Worte an Sie richten sowie ein paar organisatorische Details erläutern:

Inhalt der Veranstaltung sind moderne Verfahren der Robotik, wobei insbesondere Fragestellungen der (kinematischen und dynamischen) Modellierung als auch (fortgeschrittene) regelungstechnische Methoden im Zentrum stehen. Nach erfolgreichem Besuch sollen Sie in der Lage sein, serielle Roboter mathematisch zu beschreiben, hochgenau zu regeln und für Applikationen geeignet anzupassen. Das hierfür erforderliche Methodenwissen wird in der Vorlesung behandelt und anhand von Übungen vertieft, so dass ein eigenständiges wissenschaftliches Arbeiten möglich ist. Darüber hinaus bieten wir über Stud.IP weiterführende Literatur zu ausgewählten Themen für das Selbststudium an.

Das vorliegende Skript kann den Besuch und die aktive Teilnahme an Vorlesung und Übung nicht ersetzen. Vielmehr soll der Schreibaufwand reduziert werden, da der Inhalt des Skriptes sich mit den Folien deckt. An einigen Stellen finden Sie Lücken im Skript, die mit Herleitungen und Beispielrechnungen zu füllen sind. Erst mit diesen Ergänzungen, die ein besseres Verständnis gewisser mathematischer Zusammenhänge zum Ziel haben, ist das Skript vollständig und bildet so (hoffentlich) eine gute Grundlage für die Prüfungsvorbereitung und Ihr späteres Berufsleben.

Auch wenn immer wieder der Verdacht geäußert wird, dass wir durch die Lücken in den Unterlagen Sie nur zum Besuch der Veranstaltung zwingen möchten, so haben wir uns doch etwas anderes gedacht: Der Wechsel zwischen Tafelanschrieben und Folien gestaltet die Veranstaltung weniger monoton als eine reine PowerPoint-Präsentation, das Tempo wird verlangsamt und (zumindest behaupten dies die Lernforscher) das Geschriebene prägt sich besser ein als das rein gehörte Wort. Sollten die Argumente für Sie nicht überzeugend sein, so stehen Ihnen auch die zahlreichen Kopierer an der Universität zur Verfügung oder Sie warten das Ende der Veranstaltung ab. Dann wird das vollständige Skript mit allen Tafelanschrieben in StudIP hochgeladen. Wir würden uns auf jeden Fall freuen, Sie in Vorlesung und Übung regelmäßig zu sehen.

Für das eigenständige Überprüfen des Lernerfolgs und Vertiefen der Inhalte haben wir uns eine Reihe verschiedener Maßnahmen überlegt:

- Am Ende eines jeden Kapitels sind Fragen zur Selbstkontrolle im Skript eingepflegt. Diese werden in der Hörsaalübung nicht gerechnet, sondern dienen als zusätzliches Material zur Vorbereitung auf die Prüfung.
- Neben den Fragen zur Selbstkontrolle gibt es eine Computerübung (Umfang 1 ECTS-Punkt), die Sie begleitend zur Veranstaltung durchführen und welche den fünften Credit des Moduls bildet. An einem realen Roboter werden hierbei Themen der Hörsaalübung eigenständig am Rechner vertieft. Die Bearbeitung der Computerübung erfolgt im Internetbrowser mittels Matlab Grader. Infos zur Computerübung werden zeitnah in der Hörsaalübung bekannt gegeben. Sie erhalten den Leistungspunkt, wenn Sie drei von vier Laboren erfolgreich bestanden haben. Besonders motivierte Studierende können im ReLab (Remote Laboratory) Zusatzaufgaben der Computerübung bearbeiten und den realen Roboter von zu Hause aus fernsteuern. Die Besten bekommen am Ende des Semesters dafür eine Urkunde.

Fragen zu den vorgenannten Angeboten beantworten wir selbstverständlich jederzeit gerne. Wir möchten Sie ermutigen, diese intensiv, auch im Hinblick auf die Klausur, zu nutzen.

Sollten Sie an der Robotik Gefallen gefunden haben, so können Sie Ihre Kenntnisse im nächsten Semester in der „Robotik II“ Vorlesung erweitern. Hier stehen parallelkinematische Roboter als Erweiterung, der in dieser Vorlesung behandelten seriellen Ketten auf der Agenda. Darüber hinaus beschäftigen wir uns mit Fragen der Parameteridentifikation und Optimierung als zentrales Element der Modellierung und geben eine Einführung in grundlegende Verfahren des maschinellen Lernens. Des Weiteren bieten wir die Veranstaltung Robot Challenge an: In einem Roboterwettkampf treten studentische Gruppen gegeneinander an und lösen mit mobilen Robotern spannende Aufgaben. Zu guter Letzt: Im Rahmen der Ausbildungsinitsiativ „Roboterfabrik“ gibt es zusätzlich noch die Möglichkeit an unserem Hackathon „Mobile Robotik“ teilzunehmen. Nähere Informationen stehen auch auf unserer Internetseite. Darüber hinaus finden am Institut zahlreiche studentische Arbeiten auf dem Gebiet der Robotik statt – ein regelmäßiger Blick auf die Institutshomepage lohnt sich. Oder Sie sprechen uns bei Interesse nach der Vorlesung/Hörsaalübung an.



Homepage: imes

Zu guter Letzt möchte ich mich bei einigen Personen bedanken, die durch ihr Engagement die Erstellung des umfangreichen Skripts, der Folien, etc. überhaupt erst ermöglicht haben. Hier ist an erster Stelle Herr Dr.-Ing. Jens Kotlarski zu nennen, der die Vorlesung jahrelang betreut und die Unterlagen zu Vorlesung und Hörsaalübung erstellt hat. Seine Nachfolge übernahm Frau Dipl.-Ing. Kathrin Nülle, die zusammen mit Tobias Klenke und Thomas Pankratz eine erste Version der Computerübung erstellt hat. Herr M. Sc. Marvin Stüde überführte die Computerübung zu Matlab Grader, Herr M. Sc. Tim-Lukas Habich überarbeitete die Computerübung und baute zusammen mit M. Sc. Tim Sterneck das ReLab auf, welches erstmals im WS 2022/2023 startet. Frau Jessica Specht M.A. hat die erste Version des Skripts mühevoll in Adobe Indesign erstellt. Die nun vorliegende LaTex Version des Skriptes (die eine bessere Lesbarkeit der Formeln zum Ziel hat) wurde von Frau B. Sc. Nazli Bozoglu maßgeblich gestaltet. Herrn Dr.-Ing. Jan-Philipp Kobler danke ich für das akkurate Korrekturlesen. Herrn Professor Heimann danke ich abschließend für die Überlassung seiner Unterlagen.

Sollten Sie noch Anregungen zur weiteren Verbesserung haben, so sprechen Sie uns an. Wir bemühen uns, Ihre Vorschläge aufzunehmen. Der Dank der nachfolgenden Kommilitonen ist Ihnen sicher.

Nun wünsche ich Ihnen viel Spaß und noch mehr Erfolg mit der „Robotik I“ Vorlesung!

Herzlichst, Ihr

Professor Dr.-Ing. Tobias Ortmaier



Homepage: IRT

Die Vorlesung wird mit gleichem Inhalt im Wechsel mit dem Institut für Regelungstechnik (IRT) der Fakultät für Elektrotechnik und Informatik gehalten. Vorlesung und Prüfung werden dabei im Sommersemester vom IRT und im Wintersemester vom imes durchgeführt.



Homepage: luhbots

PS: Interesse an nationalen und internationalen Roboterwettkämpfen? Das Team luhbots vertritt äußerst erfolgreich die Leibniz Universität Hannover und setzt sich aus Studenten verschiedenster Fakultäten zusammen. Kontaktieren Sie einfach den Teamleiter ...

Inhaltsverzeichnis

1 Einführung	1
1.1 Begriffsdefinition	1
1.2 Klassifizierung von Roboterarten	2
1.2.1 Industrieroboter	2
1.2.2 Serviceroboter	6
1.2.3 Entertainment- und Edutainmentroboter	7
1.3 Verzahnung der Vorlesungsinhalte	7
2 Koordinatentransformationen	11
2.1 Joint- und Taskspace	11
2.2 Koordinatensysteme und Vektoren	13
2.3 Rotationsmatrizen	16
2.3.1 Elementardrehungen	18
2.3.2 Zusammengesetzte Drehungen	18
2.3.3 Direkter und inverser Zusammenhang	21
2.3.4 Zusammenfassung	22
2.4 Homogene Transformationen	22
2.5 Alternative Rotationsdarstellungen	25
2.5.1 Achse-Winkel-Darstellung	25
2.5.2 Quaternionen	27
3 Direkte und inverse Kinematik	31
3.1 Kinematik serieller Roboter	31
3.2 Direkte Kinematik	32
3.3 Denavit-Hartenberg Notation	34
3.3.1 Aufstellen der Denavit-Hartenberg Matrizen	34
3.3.2 Bestimmung der Lage des Endeffektors	38
3.4 Inverse Kinematik	39
3.4.1 Analytische Berechnung der inversen Kinematik	40
3.4.2 Geometrische Ermittlung der inversen Kinematik	43
3.4.3 Numerische Berechnung der inversen Kinematik	43
4 Jacobi-Matrix – Geschwindigkeiten und Kräfte	47
4.1 Differentielle Kinematik	47
4.2 Analytische Jacobi-Matrix	49
4.3 Geometrische Jacobi-Matrix	50
4.4 Manipulierbarkeit	56
4.5 Statisches Kraft- und Momentmodell	58
4.5.1 Wechselwirkung mit der Umwelt	58
4.5.2 Steifigkeitsanalyse	59
4.6 Singularitäten	61
5 Kinematisch redundante Roboter	69
5.1 Redundanz und differentielle Kinematik	70

5.2	Nullraum als homogene Lösung der differentiellen Kinematik	74
5.3	Redundanz und inverse Kinematik	78
6	Bahnplanung	83
6.1	Einführung	84
6.2	Bahnplanung einer Einzelachsbewegung	87
6.3	Bahnplanung im Jointspace	91
6.4	Bahnplanung von translatorischen Bewegungen im Taskspace	92
6.5	Bahnplanung von rotatorischen Bewegungen im Taskspace	96
6.5.1	Interpolation der Rotationswinkel zusammengesetzter Drehungen	97
6.5.2	Drehung um äquivalente Achse mit einem bestimmten Winkel	97
6.5.3	Interpolation von Quaternionen	99
6.6	Ganzheitliche Bahnplanung im Taskspace	101
6.7	Anwendungsbeispiel: Energieoptimale Roboterbahnen	102
6.8	Roboterprogrammierung	104
6.9	Automatisierte Bahnplanung	107
6.9.1	Wiederholung: Arbeitsraum und Konfigurationsraum	108
6.9.2	Diskretisierung des Konfigurationsraum	110
6.9.3	Diskrete Planungsprobleme	111
6.9.4	Algorithmen für die automatisierte Bahnplanung	112
7	Dynamik: Newton-Euler und Lagrange	121
7.1	Begriffsdefinitionen und Grundlagen	121
7.2	Bewegungsgleichung nach Newton-Euler	123
7.3	Bewegungsgleichung nach Lagrange	132
7.4	Berücksichtigung von Motor und Getriebe	137
7.4.1	Getriebeübersetzung	138
7.4.2	Reibeffekte	138
7.4.3	Motor- und Getriebeträgheiten	139
8	Einzelachsregelung	143
8.1	Einleitung	143
8.1.1	Robotersteuerung	144
8.1.2	Regelung	144
8.2	Rückkopplungsstrukturen	145
8.2.1	Dezentrale Regelung	145
8.2.2	Zentrale Regelung	146
8.2.3	Klassische Rückführungen	146
8.2.4	Frequenzbereich	149
8.3	Einzelachsregelung im Jointspace	150
8.3.1	Modell einer Roboterachse	151
8.3.2	Vereinfachtes Modell einer Roboterachse	156
8.3.3	Einzelachsregelung	157
8.4	Übertragungsfunktionen und deren Eigenschaften	160
8.4.1	Stabilitätsuntersuchung (Wiederholung)	162
8.4.2	Bleibende Regelabweichung	165
8.5	Reglerauslegung	167
8.5.1	Polvorgabe	167
8.5.2	Heuristische Einstellregeln	169
8.6	Vorsteuerung	169

8.7 Einzelachsregelung im Taskspace	172
9 Mehrachsregelung	177
9.1 Einführung	177
9.2 Feedforward-Steuerung (Momentenvorsteuerung)	177
9.3 Feedback-Regelung	179
9.4 Verzahnung der Vorlesungsinhalte	182
10 Fortgeschrittene Regelverfahren	185
10.1 Einleitung	185
10.2 Passive Steifigkeitsregelung	185
10.3 Aktive Steifigkeitsregelung	187
10.3.1 Grundlagen	187
10.3.2 Gravitationskompensation	189
10.3.3 Aktive Steifigkeitsregelung im Kontaktfall	191
10.4 Aktive Steifigkeitsregelung inklusive Rückführung	194
10.5 Impedanzregelung	196
10.6 Direkte Kraftregelung	201
10.6.1 Lagebasierte direkte Kraftregelung (Admittanzregelung)	201
10.6.2 Kraftbasierte direkte Kraftregelung	203
10.7 Hybride Kraft-/Lageregelung	204
11 Sensoren	209
11.1 Kenngrößen	209
11.2 Winkelmesssysteme	211
11.2.1 Resolver	211
11.2.2 Optische Encoder	213
11.3 Inertiale Messeinheit	213
11.3.1 Beschleunigungssensoren	214
11.3.2 Drehratensensoren	217
11.4 Kraft-Moment-Sensor	218
11.4.1 Messung einer Kraft	218
11.4.2 Messung eines Moments	220
Literatur	225
12 Übungen zur Vorlesung Robotik I	227
13 Musterklausur Robotik I	247

Abbildungsverzeichnis

1.1	Mechatronik – ein interdisziplinäres Fachgebiet	1
1.2	Wesentliche Komponenten eines Roboters	2
1.3	Sensomotorischer Regelkreis eines Roboters	2
1.4	Veranschaulichung der Absolutgenauigkeit	3
1.5	Veranschaulichung der Wiederholgenauigkeit	4
1.6	Beispiele serieller Kinematiken	5
1.7	Beispiele paralleler Roboter	5
1.8	Beispiele von Servicerobotern	6
1.9	Beispiele von Entertainment- und Edutainmentrobotern	7
1.10	Verzahnung der Vorlesungsinhalte in Form eines Regelkreises	8
2.1	Roboter mit sechs Minimalkoordinaten (Freiheitsgraden)	12
2.2	KR 10 scara, Quelle KUKA	12
2.3	PaliDA, Quelle IFW	12
2.4	Ebene Roboter mit unterschiedlichen Freiheitsgraden	13
2.5	Kartesisches, rechtwinkliges Koordinatensystem (KS_i)	13
2.6	Inertial- und körperfestes Koordinatensystem	14
2.7	Ortsvektoren	14
2.8	Vektor a als Repräsentant eines Vektorfeldes im Raum frei verschiebbar.	16
2.9	Elementardrehungen	18
2.10	Nichtkommutativität der endlichen Drehungen	19
2.11	Gimbal-Lock-Problematik mit Kardan-Winkel	20
2.12	Direkter Zusammenhang	22
2.13	Inverser Zusammenhang	22
2.14	Translation und Rotation zur Beschreibung von Ortsvektoren	23
2.15	Homogene Transformation 0T_B	23
2.16	Verkettung von Transformationen	25
2.17	Rotation um eine Achse u mit dem Winkel θ	26
3.1	Grafische Abgrenzung von direkter und inverser Kinematik	31
3.2	Serieller Roboter inklusive wesentlicher Koordinatensysteme	32
3.3	Kinematische Kette eines RRR-Roboters	33
3.4	Planarer RR-Roboter	33
3.5	Denavit-Hartenberg Notation	34
3.6	Sonderfälle der Denavit-Hartenberg Notation	36
3.7	Vorgehen zur Ermittlung der Denavit-Hartenberg Parameter	37
3.8	Symmetrische Konfigurationen eines planaren RR-Roboters	39
3.9	Planarer RR-Roboter	43
3.10	Schematische Darstellung des Newton-Raphson-Verfahrens	44
4.1	Planarer RR-Roboter mit zwei Freiheitsgraden	49
4.2	Vektoren und Transformationen zur Bestimmung der geom. Jacobi-Matrix	52
4.3	Kraftwechselwirkung mit der Umwelt	59
4.4	Planarer RRR-Roboter	61

4.5	Singuläre Konfigurationen des planaren RRR-Roboters	62
4.6	Roboterbewegung von Punkt P_0 zu P_1 bzw. P_2	64
4.7	Gelenkwinkel und Gelenkwinkelgeschwindigkeiten entlang der Bahn P_0P_1	65
4.8	Gelenkwinkel und Gelenkwinkelgeschwindigkeiten entlang der Bahn P_0P_2	65
4.9	Kraftangriff in y -Richtung	66
4.10	Kraftregelung	67
5.1	KUKA LBR	69
5.2	Planarer RRRR-Roboter in unterschiedlichen Konfigurationen	70
5.3	Planarer RRR-Roboter	74
5.4	Beispiel redundanter, planarer RRR-Roboter	74
6.1	Zusammenhang der Planungsräume	85
6.2	Varianten von Roboterbahnen	85
6.3	Bahnplanung durch Interpolation (zeitkontinuierlich)	86
6.4	Typisches Geschwindigkeitstrapezprofil	87
6.5	Bang-Bang-Steuerung	90
6.6	Geschwindigkeits- und Beschleunigungsverlauf (C^2 -stetige Trajektorie)	91
6.7	Gegenüberstellung von C^1 - und C^2 -Stetigkeit	91
6.8	Klassischer Ablauf der Bahnplanung im Jointspace	92
6.9	Synchronisation auf die „langsamste Gelenkbewegung“	93
6.10	Klassischer Ablauf der Bahnplanung im Taskspace	93
6.11	Kreisbahn	94
6.12	Unverschliffene Bahn	96
6.13	Verschliffene Bahn	96
6.14	Orientierungsinterpolation	97
6.15	Interpolation von Orientierungen mittels Drehung um äquivalente Achse	98
6.16	Projizierte Interpolationskurve der vierdimensionalen Einheitssphäre	101
6.17	Grafische Darstellung zur Interpolation von Orientierungen	101
6.18	Energieaustausch über Zwischenkreisverbund	102
6.19	Beispiel einer Bahnoptimierung	103
6.20	Grafische Bedienoberfläche des Yuanda Roboters	105
6.21	Digitaler Zwilling	106
6.22	Das Bahnplanungsproblem zwischen dem Startzustand und dem Zielzustand	107
6.23	Anwendungsbeispiele für eine manuelle und automatisierte Bewegungsplanung	108
6.25	Der Arbeits- und Konfigurationsraum von verschiedenen Kinematiken	110
6.26	Beispiel eines Sichtgraphs	110
6.27	Beispiel einer Zellzerlegung	111
6.28	Beispiel einer Brute-Force-Koordinaten-Diskretisierung	111
6.29	Beispiel der Pfadknoten der Breadth-First-Suche	114
6.30	Beispielverlauf einer Breath-First-Suche	114
6.31	Beispiel der Pfadknoten der Depth-First-Suche	115
6.32	Beispielverlauf einer Depth-First-Suche	115
6.33	Beispiel der Pfadknoten und -kosten des Djikstra-Algorithmus.	116
6.34	Beispielverlauf des Djikstra-Algorithmus.	116
6.35	Beispielverlauf der Greedy-Suche	117
6.36	Vergleich der Planung der Greedy-Suche und des Djikstra-Algorithmus	118
6.37	Beispiel der Pfadknoten und -kosten des A*-Algorithmus.	119
6.39	Vergleich der Algorithmen	120

7.1	Direkte Dynamik	122
7.2	Inverse Dynamik	123
7.3	Roboter in Teilkörper zerlegen und Schwerpunkte bestimmen	124
7.4	Bewegung eines Punktes P_x auf einem Starrkörper K_i in $(KS)_0$	125
7.5	Bestimmung der Winkelgeschwindigkeiten und -beschleunigungen	126
7.6	Rekursive Bestimmung der Geschwindigkeiten und Beschleunigungen	127
7.7	Transformation der Geschwindigkeiten und Beschleunigungen	129
7.8	Freigeschalteter Teilkörper i	129
7.9	Transformation des Massenträgheitstensors in das $(KS)_0$	130
7.10	Externer Kraftangriff am Endeffektor	134
7.11	Planarer RR-Roboter	135
7.12	Antriebsstrang einer Roboterachse (Rotationsantrieb)	138
7.13	Reibung	139
7.14	Antriebsstrang einer Roboterachse (Rotationsantrieb) mit Antriebsträgheit	140
8.1	Robotersteuerung (schematische Darstellung)	144
8.2	Allgemeine Struktur einer Roboterregelung	145
8.3	Schematischer Aufbau einer dezentralen Regelung	145
8.4	Schematischer Aufbau einer zentralen Regelung	146
8.5	Modell einer Roboterachse	146
8.6	P-Glied	147
8.7	I-Glied	147
8.8	D-Glied	147
8.9	Struktur eines PD- und PID-Glieds	148
8.10	PID-Regler im Zeit- und Frequenzbereich	149
8.11	Komponenten eines Roboters (einer Roboterachse)	150
8.12	Blockschaltbild einer Einzelachsregelung im Jointspace	151
8.13	Vereinfachtes Modell einer Roboterachse	152
8.14	PI-Stromregelung eines permanent erregten Synchronmotors	153
8.15	Blockschaltbild einer PI-Stromregelung (vereinfachte Darstellung)	153
8.16	Vereinfachung des elektrischen Teilsystems zu einem PT_1 -Glied	154
8.17	Vereinfachtes Modell (Vernachlässigung Dynamik des el. Teilsystems)	156
8.18	Blockschaltbild einer Roboterachse	156
8.19	Blockschaltbild einer Roboterachse im Frequenzbereich	157
8.20	Blockschaltbild einer Einzelachsregelung im Frequenzbereich	157
8.21	Blockschaltbild einer Einzelachsregelung mit $k_M \approx \hat{k}_M$	158
8.22	Blockschaltbild einer PID-Einzelachsregelung im Frequenzbereich	158
8.23	Typische Kaskadenregelung eines Synchronmotors	159
8.24	P/P-Kaskadenregelung einer Roboterachse	159
8.25	P/P-Kaskadenregelung einer Roboterachse	159
8.26	Alternativ P/PI-Kaskadenregelung einer Roboterachse	159
8.27	Führungsgrößenübertragungsfunktion einer Einzelachsregelung	160
8.28	Störgrößenübertragungsfunktion einer Einzelachsregelung	161
8.29	Komplexe Ebene nach Euler	162
8.30	Lage der Pole in komplexer Zahlenebene	165
8.31	P/P-Kaskadenregelung einer Roboterachse	170
8.32	P/P-Kaskadenregelung mit Geschwindigkeitsvorsteuerung	170
8.33	P/P-Kaskadenregelung mit Geschwindigkeitsvorsteuerung	171
8.34	P/P-Kaskadenregelung mit Momentenvorsteuerung	172
8.35	P/P-Kaskadenregelung mit Momentenvorsteuerung	172

8.36	PID-Regelung mit Momentenvorsteuerung	173
8.37	Einzelachsregelung im Taskspace	173
8.38	Einzelachsregelung im Taskspace (Sollgeschwindigkeit aus Bahnplanung)	174
9.1	Feedforward-Steuerung (Momentenvorsteuerung)	178
9.2	Feedback-Regelung	179
9.3	Feedback-Regelung (Rückführung nicht explizit dargestellt)	180
9.4	Feedback-Regelung unter Verwendung eines PD-Reglers	182
9.5	Verzahnung der Vorlesungsinhalte in Form eines Regelkreises	182
10.1	Remote Centre of Compliance (ATI Industrial Automation)	186
10.2	Einführen eines Bolzens in eine Bohrung	186
10.3	Modellierung des Roboterverhaltens als Hooke'sche Feder	187
10.4	Whole Arm Manipulator, Quelle Barrett Technology	188
10.5	Blockschaltbild einer aktiven Steifigkeitsregelung im Jointspace	188
10.6	Endeffektorabweichung aufgrund der Gravitation	189
10.7	Am i -ten Roboterglied angreifende Kräfte und Momente	189
10.8	Steifigkeitsregelung im Jointspace inklusive Gravitationskompensation	191
10.9	Bleibende Regelabweichung im Kontaktfall	191
10.10	Blockschaltbild einer aktiven Steifigkeitsregelung im Kontaktfall	191
10.11	Aktive Steifigkeitsregelung mit Drehmomentregelkreis auf Gelenkebene	192
10.12	Aktive Steifigkeitsregelung mit Drehmomentregelkreis auf Gelenkebene	193
10.13	Aktive Steifigkeitsregelung im Taskspace mit Gravitationskompensation	193
10.14	Messung der Kontaktkräfte/-momente durch Kraft-Moment-Sensor	194
10.15	Aktive Steifigkeitsregelung inklusive Kraftrückführung im Taskspace	194
10.16	Darstellung der am Sensor angreifenden Kräfte und Momente	195
10.17	Kompakte Darstellung der Erfassung der Kontaktkräfte und -momente	196
10.18	Aktive Steifigkeitsregelung inklusive Kraftrückführung im Taskspace	196
10.19	Roboter als einfaches Masse/Feder/Dämpfer-System	197
10.20	Blockschaltbild einer Impedanzregelung (der unterlagerte Drehmomentregler ist nicht dargestellt)	199
10.21	Blockschaltbild einer Admittanzregelung (der unterlagerte Positionsregler ist nicht dargestellt)	202
10.22	Blockschaltbild einer Admittanzregelung mit Positionssollwert (der unterlagerte Positionsregler ist nicht dargestellt)	202
10.23	Blockschaltbild einer kraftbasierten direkten Kraftregelung im Taskspace	203
10.24	Blockschaltbild einer hybriden Kraft-/Lageregelung im Taskspace	204
10.25	Einführen eines Bolzens in eine Bohrung	205
11.1	Signalflussdiagramm eines Sensors	209
11.2	Anschauliche Verdeutlichung von Präzision, Richtigkeit und Genauigkeit	210
11.3	Funktionsprinzip eines Resolvers	212
11.4	Funktionsprinzip eines optischen Encoders	214
11.5	Inertiale Messeinheit (ohne Magnetfeldsensoren)	214
11.6	Funktionsprinzip eines Beschleunigungssensors	215
11.7	Prinzipskizze eines Beschleunigungssensors	215
11.8	Verschaltung eines Beschleunigungssensors in einer Halbbrücke	216
11.9	Funktionsprinzip eines Drehratensensors	217
11.10	6-Achsen Kraft-Moment-Sensor	218
11.11	Folien DMS	219

11.12 Wheatstone'sche Vollbrücke	219
11.13 Kraftmessung am Biegebalken	220
11.14 Verformungskörper zur Momentenmessung	221

Tabellenverzeichnis

1.1	Vergleich von seriellen Robotern und Parallelkinematiken	6
3.1	DH Parameter eines Roboters mit sechs Freiheitsgraden	38
10.1	Natürliche Randbedingungen beim Einführen eines Bolzens	205
10.2	Zusammenfassung der natürlichen und künstlichen Randbedingungen	205

1 Einführung

Das Einführungskapitel befasst sich in Abschnitt 1.1 mit grundlegenden, für das Verständnis der nachfolgenden Kapitel erforderlichen, Definitionen und Begriffen. Anhand typischer Vertreter werden Roboter klassifiziert (z. B. Industrie- und Serviceroboter, aber auch neue Entwicklungen, wie die Entertainmentroboter) und gängige Leistungsmerkmale eingeführt (Abschnitt 1.2).

Als weiterführende Literatur wird empfohlen:

- W. Khalil und E. Dombre. *Modeling, Identification & Control of Robots*. New York, USA: Routledge, 2002, S. 1–12
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 1–18

1.1 Begriffsdefinition

Die Robotik ist ein klassischer Vertreter der Mechatronik: Die Disziplinen Mechanik, Elektrotechnik und Informationstechnik sind auf engstem Raum vereint (vgl. Bild 1.1), um dem Anwender die gewünschten Eigenschaften der Roboter zur Verfügung zu stellen. Dabei liegt die besondere Herausforderung bei der Entwicklung von Robotern und Applikationen in der Interdisziplinarität der beteiligten Fachgebiete, die eine enge Zusammenarbeit sowie ein entsprechend breites Wissen der beteiligten Personen voraussetzt. Die wesentlichen Bestandteile eines Roboters sind Mechanik,

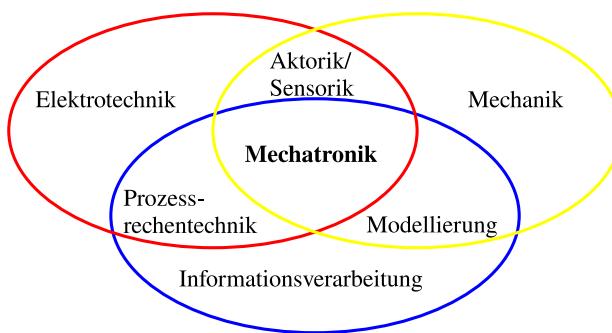


Bild 1.1: Mechatronik – ein interdisziplinäres Fachgebiet

Aktoren, Sensoren und der auf einem (Prozess-)Rechner laufende Regler, der das gewünschte Verhalten sicher stellt:

- Mechanik: Mechanische Struktur, deren Gelenke über Getriebe durch elektrische, pneumatische oder hydraulische Aktoren bewegt/angetrieben werden.
- Aktoren: Erzeugen die Bewegung der Mechanik. In Robotern kommen hauptsächlich elektrische, zuweilen jedoch auch pneumatische oder hydraulische, Antriebe zum Einsatz. Die Antriebe erhalten ihre Vorgaben (z. B. Geschwindigkeit, Position) aus dem Regler.
- Sensoren: Interne und externe Sensoren erfassen sowohl den Zustand des Roboters (z. B. Gelenkwinkel, Kräfte, etc.) als auch die Umgebung (z. B. mittels Kameras oder Ultraschallsensoren). Der Roboter kann Dank einer intelligenten Verarbeitung der Sensorsignale mit seiner Umwelt interagieren.

- Regler/Informationsverarbeitung: Generiert aus Sensorwerten Eingangssignale für die Aktoren, um die gewünschte Aufgabe durchzuführen.

Ein derartiges System ist also in der Lage, mittels seiner Sensoren die Umgebung zu erfassen, die zur Erfüllung der Aufgabe notwendigen Berechnungen durchzuführen und umzusetzen.

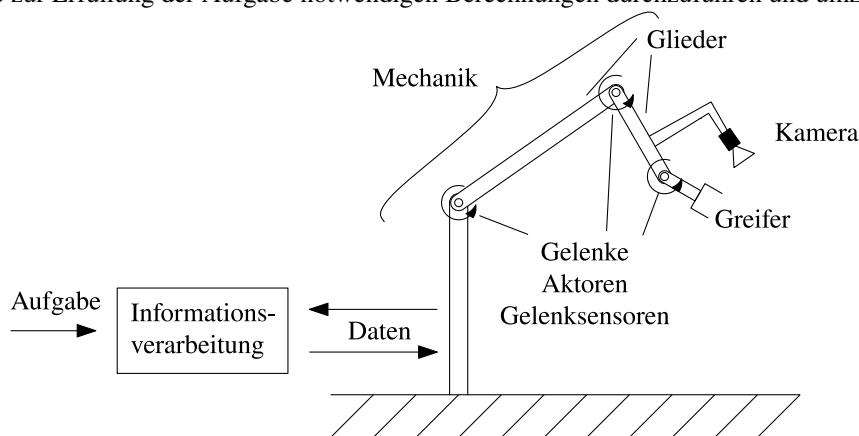


Bild 1.2: Wesentliche Komponenten eines Roboters

Ein wesentliches Kennzeichen von Robotern (oder allgemein intelligenten Systemen) ist der Grad ihrer Autonomie. Dieser setzt sich zusammen aus Flexibilität, Fehlertoleranz sowie Lern- und Entscheidungsfähigkeit. Voraussetzung hierfür ist die Erfassung der Umwelt. Ziel aktueller Forschungen ist es, die Autonomieeigenschaften von Robotern zu maximieren, um mit den Menschen oder für den Menschen nützliche Aufgaben zu erledigen. Dies setzt sensomotorische Regelschleifen voraus (Roboter nimmt seine Umwelt wahr und reagiert entsprechend, siehe Bild 1.3).

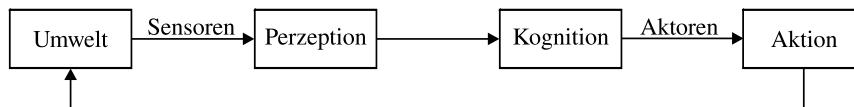


Bild 1.3: Sensomotorischer Regelkreis eines Roboters

1.2 Klassifizierung von Roboterarten

Nachfolgende Abschnitte führen eine Klassifizierung aktueller Robotersysteme anhand unterschiedlicher Anwendungsbereiche durch. Dies beinhaltet Industrieroboter (Abschnitt 1.2.1), Serviceroboter (Abschnitt 1.2.2) und Entertainment-/Edutainmentroboter (Abschnitt 1.2.3). Hierbei handelt es sich lediglich um eine von vielen möglichen Strukturierungen. Ebenso denkbar ist eine Gliederung nach: Autonomiegrad, Größe des Arbeitsbereiches, Traglast, usw.

1.2.1 Industrieroboter

In der DIN EN ISO 8373 werden Industrieroboter wie folgt definiert:

Definition
Industrieroboter

Industrieroboter sind in mehreren Achsen frei programmierbare, mit zweckdienlichen Werkzeugen ausgestattete und universell einsetzbare Maschinen zur Handhabung, Montage oder Bearbeitung von Werkstücken im industriellen Umfeld. Sie sind entweder ortsfest oder können sich frei bewegen.

Industrieroboter arbeiten zurzeit vor allem in abgetrennten Arbeitszellen, so dass keine Kooperation von Mensch und Maschine möglich ist. Diese Trennung ist vor allem Sicherheitsaspekten geschuldet und soll eine versehentliche Verletzung des Bedieners ausschließen. Betritt der Bediener die Arbeitsstelle, so wird dies erkannt (z. B. durch Öffnen einer Tür, Unterbrechen einer Lichtschranke) und die Maschine stoppt. Aktuelle Forschungsarbeiten befassen sich mit der Aufhebung dieser strikten räumlichen Trennung mit dem Ziel eine Zusammenarbeit zwischen Mensch und Roboter umzusetzen.

Zur weiteren Spezifikation der Industrieroboter werden typischerweise folgende Leistungsmerkmale herangezogen:

- Größe des Arbeitsraums,
- Absolute Positioniergenauigkeit,
- Wiederholgenauigkeit,
- Traglast,
- Anzahl der Freiheitsgrade (bei seriellen Kinematiken i. A. Anzahl der Gelenke),
- Steifigkeit der Mechanik,
- Dynamik,
- Energieverbrauch (zunehmend wichtiger).

Absolutgenauigkeit (absolute Positioniergenauigkeit)

Die Absolutgenauigkeit beschreibt (statistisch) den Abstand Δx_A zwischen einem im Inertialkoordinatensystem *gegebenem* Punkt $x_{E,soll}$ und der (wiederholt angefahrenen) Istposition $x_{E,ist}^i$ (vgl. Abbildung 1.4). Die Erfassung der aktuellen Position $x_{E,ist}^i$ erfolgt über ein externes Sensorsystem (z. B. Lasertracker). Die jeweilige Positionsabweichung berechnet sich zu:

$$\Delta x_A^i = \|x_{E,soll} - x_{E,ist}^i\|_2 \quad \text{mit } i = 1, \dots, n. \quad (1.1)$$

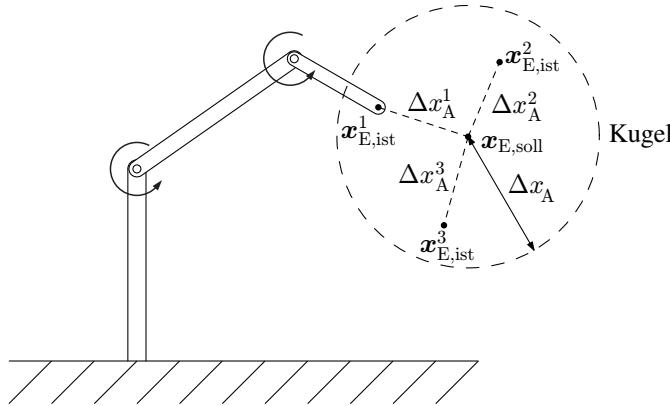


Bild 1.4: Veranschaulichung der Absolutgenauigkeit

Im Allgemeinen definiert Δx_A einen Bereich, in dem z. B. 95% aller Abweichungen Δx_A^i liegen. Die tatsächlich auftretende Abweichung kann also größer oder kleiner sein, da es sich um ein statistisches Maß handelt.

Fehlerursachen sind unter anderem:

- Fertigungstoleranzen
- Achsschiefstände
- Winkelfehler

- Temperaturänderung
- Kalibrationsfehler
- ...

Wiederholgenauigkeit

Für die Berechnung der Wiederholgenauigkeit Δx_W wird ein *geteachter* Punkt x_{E_G} wiederholt angefahren und die Abweichung

$$\Delta x_W^i = \|x_{E_G} - x_{E,\text{ist}}^i\|_2 \quad \text{mit } i = 1, \dots, n. \quad (1.2)$$

bestimmt (vgl. Abbildung 1.5).

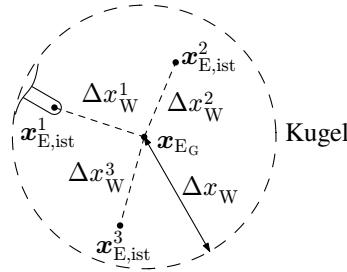


Bild 1.5: Veranschaulichung der Wiederholgenauigkeit

Ähnlich der Absolutgenauigkeit handelt es sich auch hier um ein statistisches Maß. Es gibt an, dass z. B. 95% aller Positionsfehler Δx_W^i kleiner als Δx_W sind.

Fehlerursachen sind Winkelfehler (z. B. Stick-Slip, Reibung, Richtung aus der x_{E_G} angefahren wird, Getriebetemperatur), veränderte Last, Im Allgemeinen gilt:

$$\Delta x_A > \Delta x_W, \quad (1.3)$$

d. h. die Absolutgenauigkeit eines Roboters ist deutlich schlechter als die Wiederholgenauigkeit. Ergänzende Informationen zur Definition von Genauigkeiten finden sich in [DIN99].

Serielle Kinematiken

Serielle Kinematiken zeichnen sich durch eine Baumstruktur aus, das heißt die Gelenke sind sequenziell angeordnet. Bei der Dimensionierung der Antriebe und Glieder ist deshalb darauf zu achten, dass das erste Gelenk das Gewicht der nachfolgenden trägt. Dies hat zur Folge, dass die ersten (basisnahen) Gelenke stärker dimensioniert werden müssen, als die endeffektornahen. Positionsfehler der Antriebe addieren sich im Allgemeinen entlang der kinematischen Kette auf, so dass am Endeffektor des Roboters ein größerer Lagefehler als beispielsweise bei parallelen Kinematiken (siehe nachfolgende Absätze) auftritt. Für die mechanische Steifigkeit der seriellen Roboter ist die begrenzte Steifigkeit der Gelenke maßgeblich. Aufgrund externer Last und dem Eigengewicht der Segmente kommt es zu einer Durchbiegung der Gelenke und so ebenfalls zu einem akkumulierten Lagefehler am Roboterendeffektor. Serielle Roboter verfügen über einen großen Arbeitsbereich (bezogen auf die Standfläche). Bild 1.6 zeigt typische Vertreter serieller Industrieroboter aus der Produktpalette der Firma KUKA Roboter.

Die wesentlichen Eigenschaften serieller Kinematiken sind also:

- Akkumulation der Einzelfehler,
- geringe mechanische Steifigkeit,



Bild 1.6: Beispiele serieller Kinematiken der Firma KUKA Roboter, Quelle KUKA

- großer Arbeitsbereich,
- geringer Modellierungsaufwand.

Diese begründen sich durch die sequenzielle Anordnung der Gelenke.

Parallele Kinematiken

Parallele Kinematiken sind dadurch gekennzeichnet, dass der Endeffektor von mehreren seriell-kinematischen Ketten gehalten wird (vgl. Bild 1.7). Es liegt also keine offene kinematische



Bild 1.7: Beispiele paralleler Roboter, Quellen (von links nach rechts): Adept, ABB, imes, Okuma

Kette, wie bei seriellen Robotern, vor, sondern es handelt sich um mehrere geschlossene. Diese bestehen bei parallelen Kinematiken im Allgemeinen aus einem aktuierten und mehreren passiven (nicht aktuierten) Gelenken. Der Antrieb ist dabei zumeist gestellfest oder zumindest gestellnah angeordnet. Diese mechanische Struktur führt zu folgenden Eigenschaften: Im Gegensatz zu seriellen Kinematiken akkumulieren sich die Gelenkfehler der einzelnen Achsen nicht. Dies resultiert in einer erhöhten Positioniergenauigkeit paralleler Mechanismen. Da die Antriebe nur sich selbst, die zugehörige kinematische Kette sowie anteilig den Endeffektor (inklusive Last) tragen, können diese und die mechanische Struktur entsprechend schwächer ausgelegt werden als bei seriellen Robotern. Aufgrund der geringen bewegten Massen hat dies darüber hinaus eine erhöhte Dynamik zur Folge. Des Weiteren führen die geschlossenen kinematischen Ketten zu einer hohen Steifigkeit des Roboters. Nachteilig ist der durch Singularitäten und Eigenkollisionen beschränkte Arbeitsraum. Parallele Roboter erfordern insbesondere bei Aufstellung der direkten Kinematik und der dynamischen Gleichungen einen erhöhten Modellierungsaufwand. Sie sind Gegenstand der Vorlesung „Robotik II“.

Die wesentlichen Eigenschaften paralleler Kinematiken sind in nachstehender Aufzählung nochmals knapp zusammengefasst:

- Höhere Positioniergenauigkeit,
- höhere Steifigkeit und Dynamik,
- kleinerer Arbeitsraum (bezogen auf den Bauraum),
- komplexer Modellieraufwand.

Tabelle 1.1 stellt sie wichtigsten Charakteristika serieller und paralleler Roboter nochmals gegenüber.

Tabelle 1.1: Vergleich von seriellen Robotern und Parallelkinematiken

	Serielle Kinematiken	Parallele Kinematiken
Absolutgenauigkeit	--	+
Wiederholgenauigkeit	0	++
Steifigkeit	-	+
Traglast/Eigengewicht	--	+
Beschleunigung am Endeffektor	0	++
Arbeitsraum (Größe, Hindernisvermeidung)	++	-
Flexibilität (Anpassbarkeit an Aufgabenstellung)	++	0
Modellierungsaufwand (Kinematik, Dynamik)	0	--

1.2.2 Serviceroboter

Serviceroboter sind mittlerweile ein wichtiger Zweig innerhalb der Robotik geworden. Sie lassen sich wie folgt charakterisieren:

Definition
Serviceroboter

Frei programmierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet. Dienstleistungen sind dabei Tätigkeiten, die nicht der direkten industriellen Erzeugung von Sachgütern, sondern der Verrichtung von Leistungen für Menschen und Einrichtungen dienen (Rolf-Dieter Schraft und Hansjörg Volz. *Serviceroboter: Innovative Technik in Dienstleistung und Versorgung*. Berlin, Deutschland: Springer, 1996).

Die Akzeptanz von Servicerobotik, insbesondere im Pflegebereich, ist stark kulturell geprägt. Aus diesem Grund sind Serviceroboter zurzeit vor allem in Japan und weniger in Europa/USA anzutreffen. Bild 1.8 zeigt einige typische Vertreter. Neben ansprechendem Design und hoher

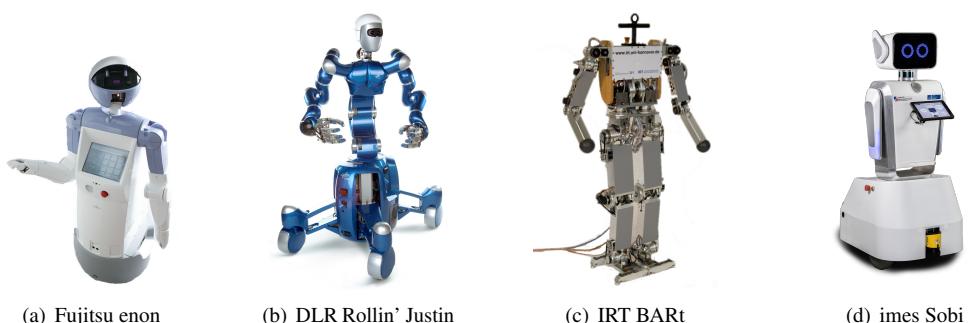


Bild 1.8: Beispiele von Servicerobotern, Quellen (von links nach rechts): Fujitsu, DLR, IRT, imes

Autonomie (der Roboter agiert in einer unstrukturierten und schlecht modellierbaren Umgebung)

kommt der Sicherheitstechnik eine überragende Rolle zu, da der Roboter direkt mit dem Menschen interagiert. Eine Verletzung des Bedieners aufgrund einer unkontrollierten Bewegung, z. B. bedingt durch einen Sensorausfall oder Programmierfehler, ist unbedingt zu vermeiden.

1.2.3 Entertainment- und Edutainmentroboter

Eine relativ neue Richtung innerhalb der Robotik stellen die Entertainment- und Edutainmentroboter dar. Sie werden in zunehmendem Maße an Schulen und Universitäten zu Lehr- und Ausbildungszwecken eingesetzt. Ausgewählte Vertreter sind in Bild 1.9 dargestellt. Der in Bild 1.9(c) gezeigte youBot der Firma KUKA Roboter ist Bestandteil der vom Institut für Mechatronische Systeme angebotenen Veranstaltung „RobotChallenge“.



Bild 1.9: Beispiele von Entertainment- und Edutainmentrobotern, Quellen (von links nach rechts): Sony, Sony, KUKA, VTech

Der Unterhaltungsroboter ist ein selbständiger mechanischer Hausgenosse, der sowohl auf externe Reize reagiert als auch nach eigenem Urteilsvermögen handelt. Die meist modularen Lernroboter hingegen eignen sich besonders für Lern-, Lehr- sowie Forschungsaufgaben (ZDNet).

Definition
Entertainment- und Edutainmentroboter

1.3 Verzahnung der Vorlesungsinhalte

Die von Ihnen gewählte Vorlesung stellt Grundlagen und Methoden der modernen Robotik vor. Ziel ist die Vermittlung von Kenntnissen, die für die Modellierung und Regelung serieller Kinematiken erforderlich sind. Um dies zu erreichen, werden folgende Themengebiete behandelt: Koordinatensysteme und Koordinatentransformationen bilden die Grundlage für die geometrische Modellierung eines Roboters. Dabei steht der Zusammenhang zwischen Gelenkwinkel/-geschwindigkeiten und Endeffektorlage/-geschwindigkeiten im Zentrum der Überlegungen. Es wird also die Frage beantwortet, wie man bei gegebenen Gelenkwinkeln auf die Lage des Endeffektors schließen kann (direkte Kinematik) und umgekehrt (inverse Kinematik). Die in der Robotik zentrale Jacobi-Matrix beschreibt selbigen Zusammenhang, jedoch für die jeweiligen Geschwindigkeiten. Darauf aufbauend werden Aspekte der Bahnplanung (also wie der Roboter von A nach B gelangt) vorgestellt. Soll die Bahn anschließend vom Roboter hochdynamisch und hochgenau abgefahren werden, so ist die exakte Kenntnis der Roboterdynamik für die Bahntreue von zentraler Bedeutung. Die Vorlesung behandelt hierfür das Aufstellen der dynamischen Bewegungsgleichungen nach Newton-Euler und Lagrange. Die so gewonnenen dynamischen und kinematischen Robotermodelle bilden die Grundlage für die sich anschließenden Regelungsverfahren. Dabei

stellt die Einzelachsregelung die Basis dar, auf die die Mehrachsregelung und die fortgeschrittenen Regelungsverfahren (Impedanz-, Admittanz- und hybride Kraft-/Lageregelung) aufbauen. Im letzten Kapitel werden für die Regelung und Robotik wichtige Sensoren behandelt: Winkelgeber, inertiale Messeinheiten sowie Kraft-/Momentsensoren. Der Zusammenhang wesentlicher Inhalte dieser Veranstaltung ist in Bild 1.10 nochmals in Gestalt eines Regelkreises grafisch dargestellt.

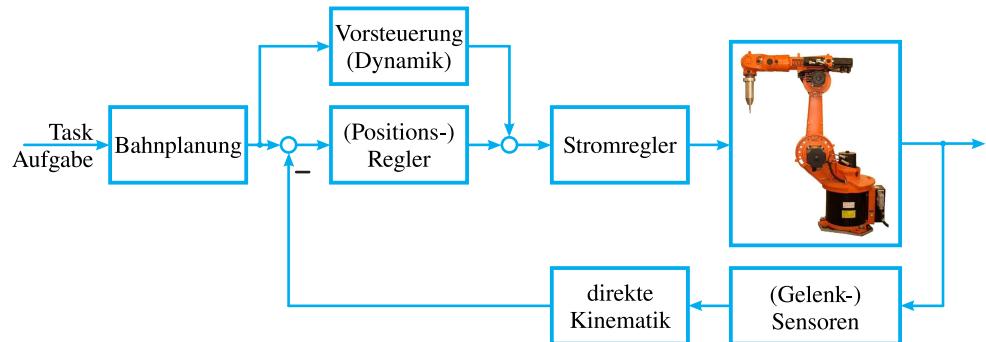
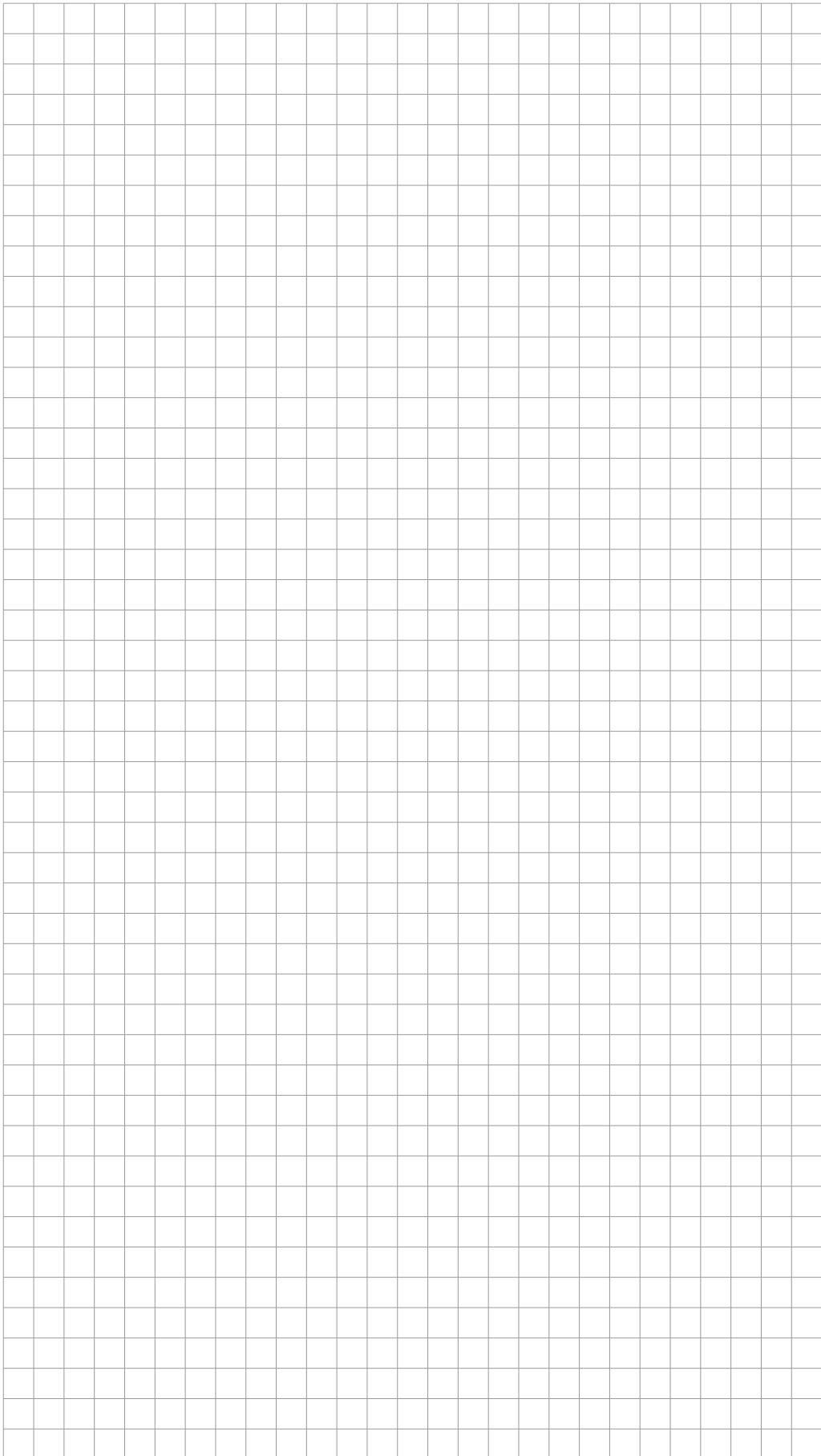
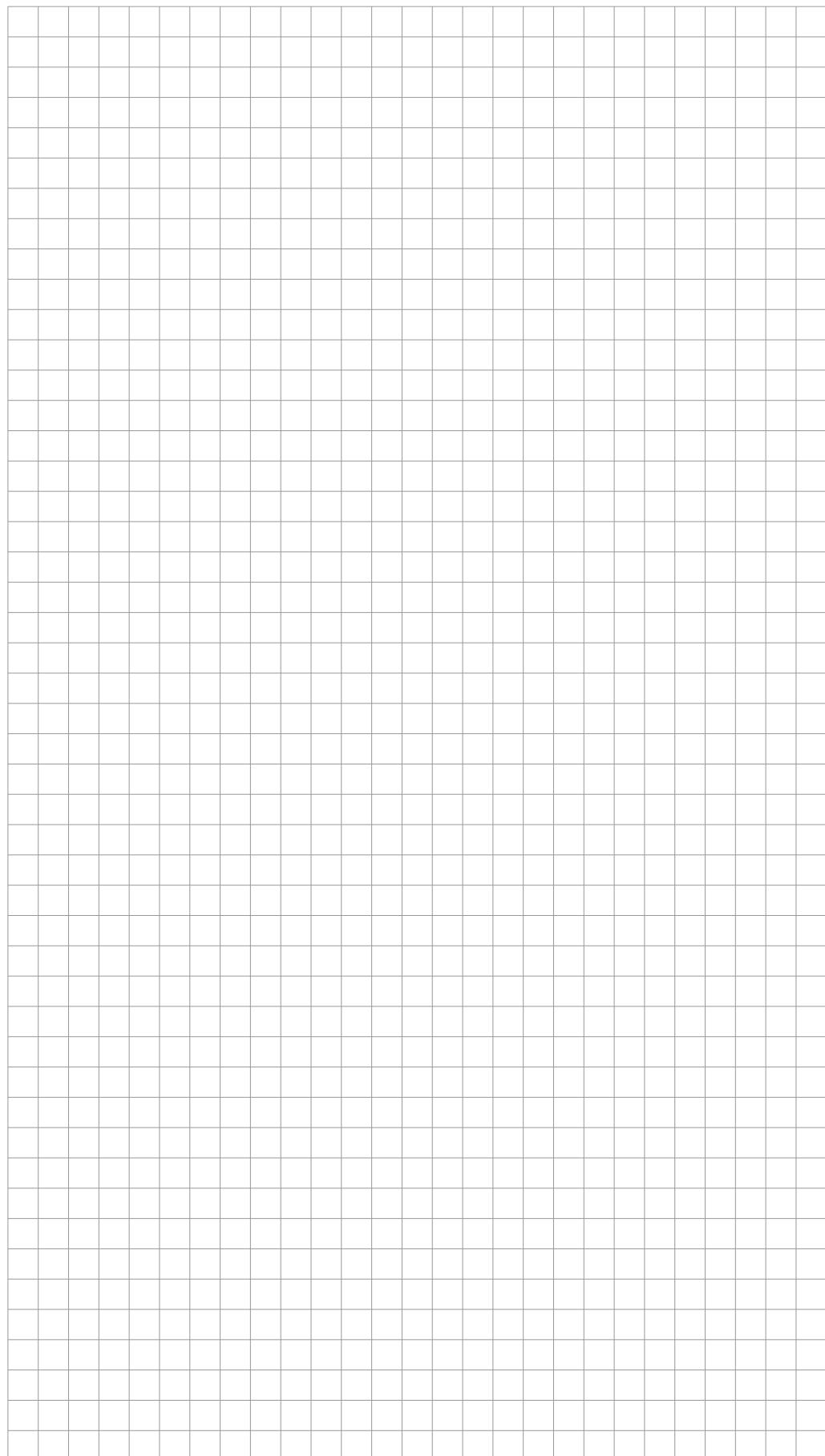


Bild 1.10: Verzahnung der Vorlesungsinhalte in Form eines Regelkreises

Fragen zur Selbstkontrolle

1. Nennen Sie die wesentlichen Bestandteile eines Roboters bzw. eines mechatronischen Systems im Allgemeinen!
 2. Welche Arten von Robotern sind Ihnen bekannt?
 3. Erläutern Sie die Eigenschaften *absolute Positioniergenauigkeit* und *Wiederholgenauigkeit*!





2 Koordinatentransformationen

Koordinatensysteme und Koordinatentransformationen sind wesentliche Bestandteile der Beschreibung von Roboterkinetiken. Grundlage für die nachfolgenden Abschnitte sind hierfür die in der Robotik üblichen Räume Jointspace (Gelenkwinkel- oder Konfigurationsraum) und Taskspace (Arbeitsraum), in denen der Roboter betrachtet wird (Abschnitt 2.1). Das vorliegende Kapitel führt darauf aufbauend die klassischen Koordinatensysteme der Robotik (ortsfest und gelenkfest) ein (Abschnitt 2.2).

Um Orts- oder Richtungsvektoren von einem Koordinatensystem in ein anderes zu transformieren, muss die relative Lage von Koordinatensystemen zueinander beschrieben werden. Diese setzt sich zusammen aus der Orientierung, die beispielsweise durch Rotationsmatrizen (Abschnitt 2.3) repräsentiert wird sowie der Translation. Mittels homogener Transformationen (Abschnitt 2.4) lassen sich beide Komponenten kompakt gemeinsam beschreiben.

Neben den klassischen Rotationsdarstellungen wie Kardan- und Eulerwinkel und den daraus resultierenden Rotationsmatrizen werden abschließend die Achse-Winkel-Darstellung sowie Quaternionen eingeführt (Abschnitt 2.5), da sie gegenüber der Winkelbeschreibung einige Vorteile aufweisen.

Als weiterführende Literatur wird empfohlen:

- W. Khalil und E. Dombre. *Modeling, Identification & Control of Robots*. New York, USA: Routledge, 2002, S. 13–25, 53ff
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 19–59
- V.E. Kremer. *Quaternions and SLERP*. Techn. Ber. Saarbrücken, Deutschland: Department for Computer Science, University of Saarbrücken, 2008
- J. Diebel. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*. Techn. Ber. Stanford, USA: Stanford University, 2006

2.1 Joint- und Taskspace

Unter dem Jointspace, auch Konfigurationsraum genannt, wird die Menge aller möglichen Gelenkwinkelkombinationen \mathcal{Q} verstanden:

$$\mathcal{Q} = \{\mathbf{q} \mid \mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max}\}, \quad (2.1)$$

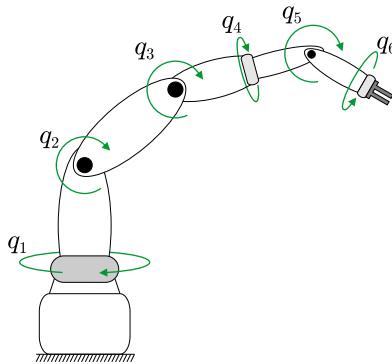
wobei \mathbf{q} die **Gelenkkoordinaten** (oder auch Gelenkwinkel) des Roboters (vgl. Bild 2.1) und \mathbf{q}_{\min} bzw. \mathbf{q}_{\max} deren untere bzw. obere Grenzen darstellen.

Definition
Jointspace/
Konfigurationsraum

Die Gelenkkoordinaten entsprechen den Minimalkoordinaten (siehe Definition unten) eines seriellen Roboters. Ebenfalls gebräuchlich ist die Bezeichnung *verallgemeinerte Gelenkwinkel*, da häufig rotatorische und translatorische Gelenke zusammengefasst werden.

Für die Gelenkkoordinaten

$$\mathbf{q} = (q_1, q_2, \dots, q_n)^T \quad (2.2)$$

**Bild 2.1:** Roboter mit sechs Minimalkoordinaten (Freiheitsgraden)

gilt: n = Anzahl der Freiheitsgrade, q_i = Winkel (Drehaktoren) oder Längen (Schubaktoren).

Definition
Minimalkoordinaten

Minimalkoordinaten sind die voneinander unabhängigen Parameter, durch die die Konfiguration eines Systems (hier: Lage des Roboters) eindeutig beschrieben werden kann, vgl. beispielsweise Bild 2.1 für einen seriellen Roboter mit sechs Gelenken.

Gelenkkarten

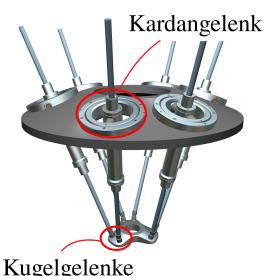
Definition
Gelenkkarten

Die bei seriellen Robotern typischerweise vorkommenden Gelenkkarten (Schub- und Drehgelenk) sind in Bild 2.2 dargestellt. Beide verfügen über jeweils einen Freiheitsgrad (FHG):

- Drehgelenk (rotational, R): 1 FHG
- Schubgelenk (prismatic, P): 1 FHG

Wichtige **Sonderfälle** von Gelenkkarten können der in Bild 2.3 gezeigten Parallelkinematik PaLiDA entnommen werden. Im Gegensatz zu den zuvor dargestellten Gelenken mit jeweils einem Freiheitsgrad verfügen das Kardan- und das Kugelgelenk über mehrere Freiheitsgrade:

- Kardangelenk (universal, U): 2 FHG
- Kugelgelenk (spherical, S): 3 FHG

**Bild 2.2:** KR 10 scara, Quelle KUKA**Bild 2.3:** PaLiDA, Quelle IFW

Umweltkoordinaten

Definition
Taskspace/
Arbeitsraum

Der Taskspace (Arbeitsraum) ist die Menge \mathcal{X} aller möglichen Lagen x_E (Position und Orientierung) des Endeffektors:

$$\mathcal{X} = \{x_E \mid x_E = f(\mathbf{q}) \wedge \mathbf{q} \in \mathcal{Q}\}, \quad (2.3)$$

wobei $f(\mathbf{q})$ die direkte Kinematik (siehe Kapitel 3) des Roboters darstellt, die aus den Gelenkwinkeln \mathbf{q} die Lage des Endeffektors (Umweltkoordinaten) \mathbf{x}_E berechnet (mathematisch formuliert: die den Gelenkwinkelraum auf den Arbeitsraum abbildet).

Die Lage des Endeffektors \mathbf{x}_E lautet

$$\mathbf{x}_E = (x_{E_1}, x_{E_2}, \dots, x_{E_m})^T, \quad (2.4)$$

mit m gleich Anzahl unabhängiger Parameter zur eindeutigen Beschreibung der Endeffektorlage. Hierbei handelt es sich im Allgemeinen um kartesische Koordinaten.

Bild 2.4(a) zeigt einen RR-Roboter, für den $\dim(\mathbf{x}_E) = \dim(\mathbf{q}) = 2$ gilt. Es ist zu erkennen, dass dieselbe Endeffektorposition durch zwei unterschiedliche Konfigurationen (hoch und tief) des Roboters erreicht wird. Weitere Details zu dieser Problematik sind in Kapitel 3 zu finden. Bild 2.4(b) beschreibt einen planaren RRR-Roboter, der über mehr Freiheitsgrade im Gelenkwinkelraum verfügt als zur Positionierung des Endeffektors notwendig ist (Orientierung sei nicht betrachtet). Es handelt sich hierbei um einen redundanten Roboter mit $\dim(\mathbf{x}_E) = 2$ und $\dim(\mathbf{q}) = 3$. Redundante Roboter verfügen im Allgemeinen über unendlich viele Konfigurationen für eine gegebene Endeffektorlage. Weitere Informationen zu redundanten Robotern sind ebenfalls in Kapitel 3 zu finden.

Beispiel
Roboterkonfigurationen

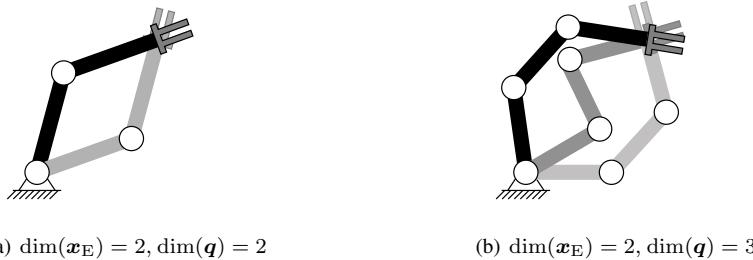


Bild 2.4: Ebene Roboter mit unterschiedlichen Freiheitsgraden; links: zwei Freiheitsgrade, rechts: drei Freiheitsgrade

2.2 Koordinatensysteme und Vektoren

Ein Koordinatensystem in der Robotik spannt in der Regel einen dreidimensionalen geometrischen Raum auf. Klassisch werden rechtwinklige (orthogonale) kartesische Koordinatensysteme in Form eines Rechtssystems verwendet, vgl. Bild 2.5.

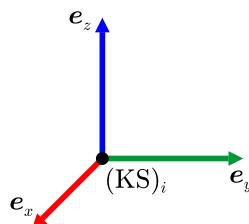


Bild 2.5: Kartesisches, rechtwinkliges Koordinatensystem $(KS)_i$

Die mathematischen Eigenschaften eines rechtwinkligen kartesischen Koordinatensystems lassen sich wie folgt beschreiben:

- Die Basisvektoren lauten:

$$\mathbf{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{e}_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (2.5)$$

- Der Betrag der Basisvektoren ist eins:

$$\|\mathbf{e}_x\|_2 = \|\mathbf{e}_y\|_2 = \|\mathbf{e}_z\|_2 = 1. \quad (2.6)$$

- Die Basisvektoren stehen paarweise senkrecht zueinander:

$$\mathbf{e}_x^T \mathbf{e}_y = \mathbf{e}_x^T \mathbf{e}_z = \mathbf{e}_y^T \mathbf{e}_z = 0. \quad (2.7)$$

- Es handelt sich um ein Rechtssystem:

$$\mathbf{e}_z = \mathbf{e}_x \times \mathbf{e}_y \quad (\det(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z) = +1). \quad (2.8)$$

Klassische Koordinatensysteme in der Robotik

In der Robotik spielen zwei Arten von Koordinatensystemen eine wichtige Rolle: Das ortsfeste Basis- oder Inertialkoordinatensystem $(KS)_0$ und die körperfesten Koordinatensysteme $(KS)_B$.

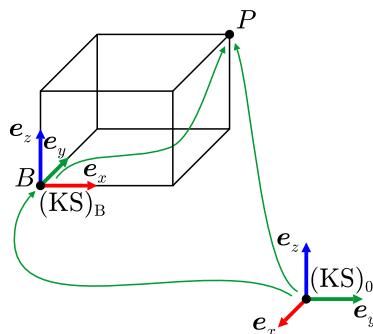


Bild 2.6: Inertial- und körperfestes Koordinaten- system

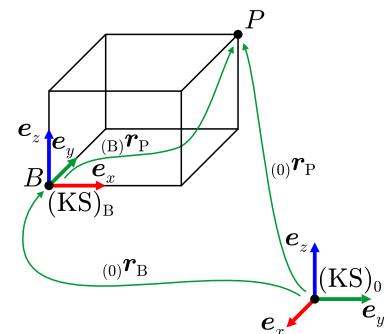


Bild 2.7: Ortsvektoren

Inertialkoordinatensystem $(KS)_0$:

Ortsfestes Koordinatensystem, welches häufig auch als Welt- oder Basiskoordinatensystem bezeichnet wird (befindet sich typischerweise in der Roboterbasis).

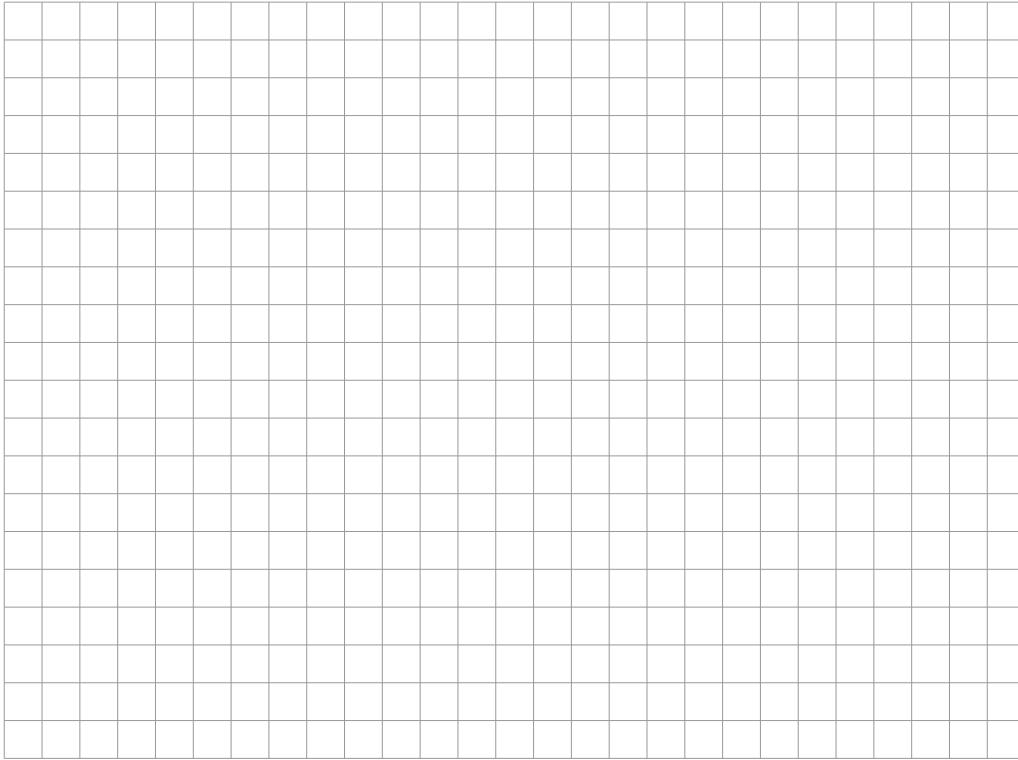
Körperfestes Koordinatensystem $(KS)_B$:

Koordinatensystem, das fest an einen Körper gebunden ist und sich mit diesem bewegt (beispielsweise am Endeffektor oder gelenkfest).

Definition
Lage

Die Lage eines Objekts im Raum wird in der Regel durch ein Koordinatensystem beschrieben, siehe Bild 2.6. Die **Lage** setzt sich aus **Orientierung** und **Position** zusammen.

In den folgenden Abschnitten werden deshalb zunächst die Darstellung der Orientierung mittels Rotationsmatrizen (Abschnitt 2.3) und danach die homogene Transformation als einheitliche Beschreibung von Orientierung und Position (Abschnitt 2.4) behandelt.



Anschließend noch einige Anmerkungen zu **Ortsvektoren** und deren Notation:

Ein Ortsvektor zeigt vom Ursprung eines Koordinatensystems $(KS)_i$ zu einem Punkt P und wird im selben Koordinatensystem $(KS)_i$ angegeben, siehe auch Bild 2.7.

Definition
Ortsvektor

Der in Klammern gesetzte, links von der Variable stehende, Index gibt das Koordinatensystem an, in dem der Vektor ausgedrückt wird. So ist die Position ${}_{(0)}\mathbf{r}_B$ des körperfesten Ursprungs des Koordinatensystems $(KS)_B$, dargestellt im Inertialsystem $(KS)_0$, durch den Ortsvektor

$${}_{(0)}\mathbf{r}_B = ({}_{(0)}x_B, {}_{(0)}y_B, {}_{(0)}z_B)^T = {}_{(0)}(x_B, y_B, z_B)^T \quad (2.9)$$

gegeben. Der Ortsvektor ${}_{(B)}\mathbf{r}_P$ vom Ursprung von $(KS)_B$ zum Punkt P in Körperkoordinaten lautet:

$${}_{(B)}\mathbf{r}_P = ({}_{(B)}x_P, {}_{(B)}y_P, {}_{(B)}z_P)^T = {}_{(B)}(x_P, y_P, z_P)^T. \quad (2.10)$$

Ein Ortsvektor beschreibt, wie weit (entsprechend den einzelnen Komponenten des Vektors) vom Ursprung des Koordinatensystems ausgehend in Richtung der jeweiligen Basisvektoren (die das Koordinatensystem aufspannen) der Punkt entfernt ist. Die ausführliche Schreibweise verdeutlicht diesen (trivialen) Zusammenhang:

$$\begin{aligned} {}_{(0)}\mathbf{r}_P &= {}_{(0)}\mathbf{e}_x^{(0)} {}_{(0)}x_P + {}_{(0)}\mathbf{e}_y^{(0)} {}_{(0)}y_P + {}_{(0)}\mathbf{e}_z^{(0)} {}_{(0)}z_P \\ &= \underbrace{\left({}_{(0)}\mathbf{e}_x^{(0)}, {}_{(0)}\mathbf{e}_y^{(0)}, {}_{(0)}\mathbf{e}_z^{(0)} \right)}_{\mathbf{E}} \begin{pmatrix} {}_{(0)}x_P \\ {}_{(0)}y_P \\ {}_{(0)}z_P \end{pmatrix}. \end{aligned} \quad (2.11)$$

Von den Ortsvektoren sind die **(Richtungs-)Vektoren** zu unterscheiden.

Definition
(Richtungs-)Vektoren

Zu zwei Punkten A_1 und A_2 im Raum gibt es genau eine Parallelverschiebung, die A_1 nach A_2 abbildet [MV03].

Der Vektor \mathbf{a} beschreibt die Verschiebung, die den Punkt A_1 nach A_2 bringt: $\mathbf{a} = \overrightarrow{A_1 A_2}$ (vgl. Bild 2.8). Wird unter \mathbf{a} der Punkt B_1 nach B_2 abgebildet, so handelt es sich um dieselbe Verschie-

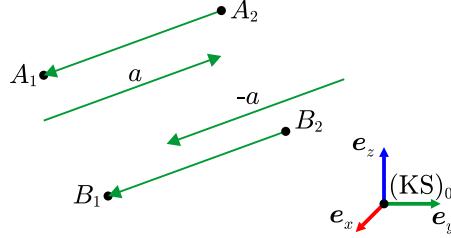


Bild 2.8: Vektor \mathbf{a} als Repräsentant eines Vektorfeldes im Raum frei verschiebbar.

bung und es gilt: $\overrightarrow{A_1 A_2} = \overrightarrow{B_1 B_2}$. Dies hat zur Folge, dass \mathbf{a} frei im Raum parallel verschiebbar ist (\mathbf{a} ist ein Repräsentant eines Vektorfeldes). Gedanklich hängt ${}^{(0)}\mathbf{a}$ nicht am Ursprung des Koordinatensystems $(KS)_0$ sondern wird lediglich in diesem (durch die Basisvektoren von $(KS)_0$) ausgedrückt. Ein Ortsvektor hingegen ist nicht im Raum parallel verschiebbar. Für die in Bild 2.7 gezeigten Vektoren bedeutet dies konkret:

Der Vektor vom Ursprung von $(KS)_B$ zum Punkt P im Inertialkoordinatensystem $(KS)_0$ ausgedrückt (kein Ortsvektor, vgl. Definition oben, sondern Richtungsvektor) lässt sich wie folgt berechnen:

$$\begin{aligned} {}^{(0)}\mathbf{r}_{BP} &= {}^{(0)}\mathbf{e}_x^{(B)} {}_{(B)}x_P + {}^{(0)}\mathbf{e}_y^{(B)} {}_{(B)}y_P + {}^{(0)}\mathbf{e}_z^{(B)} {}_{(B)}z_P \\ &= \underbrace{\left({}^{(0)}\mathbf{e}_x^{(B)}, {}^{(0)}\mathbf{e}_y^{(B)}, {}^{(0)}\mathbf{e}_z^{(B)} \right)}_{{}^0\mathbf{R}_B} \begin{pmatrix} {}^{(B)}x_P \\ {}^{(B)}y_P \\ {}^{(B)}z_P \end{pmatrix} = {}^0\mathbf{R}_B {}_{(B)}\mathbf{r}_P. \end{aligned} \quad (2.12)$$

Die Rotationsmatrix ${}^0\mathbf{R}_B$ beschreibt die Abbildung der Basiskoordinaten von $(KS)_B$ im $(KS)_0$ und ist Gegenstand von Abschnitt 2.3.

2.3 Rotationsmatrizen

Wie zuvor beschrieben setzt sich die Lage eines Objekts im Raum aus Position und Orientierung zusammen. Zunächst sei hier die Darstellung der Orientierung mittels Rotationsmatrizen betrachtet. Die Rotationsmatrix zur Transformation vom Koordinatensystem $(KS)_B$ in das Koordinatensystem $(KS)_0$ lautet:

$${}^0\mathbf{R}_B = \left({}^{(0)}\mathbf{e}_x^{(B)}, {}^{(0)}\mathbf{e}_y^{(B)}, {}^{(0)}\mathbf{e}_z^{(B)} \right) \in \text{SO}(3), \quad (2.13)$$

wobei $\text{SO}(3)$ für „special orthogonal“ (also spezielle orthogonale Gruppe im dreidimensionalen Raum) mit folgenden Eigenschaften steht:

- Die Rotationsmatrizen sind orthogonal:

$$({}^0\mathbf{R}_B)^{-1} = ({}^0\mathbf{R}_B)^T = {}^B\mathbf{R}_0. \quad (2.14)$$

- Die Basisvektoren einer Rotationsmatrix sind orthonormal:

$$\begin{aligned} \left\| {}_{(0)}e_x^{(B)} \right\|_2 &= \left\| {}_{(0)}e_y^{(B)} \right\|_2 = \left\| {}_{(0)}e_z^{(B)} \right\|_2 = 1 \\ \left({}_{(0)}e_x^{(B)} \right)^T {}_{(0)}e_y^{(B)} &= \left({}_{(0)}e_x^{(B)} \right)^T {}_{(0)}e_z^{(B)} = \left({}_{(0)}e_y^{(B)} \right)^T {}_{(0)}e_z^{(B)} = 0. \end{aligned} \quad (2.15)$$

- Die Basisvektoren spannen ein Rechtssystem auf:

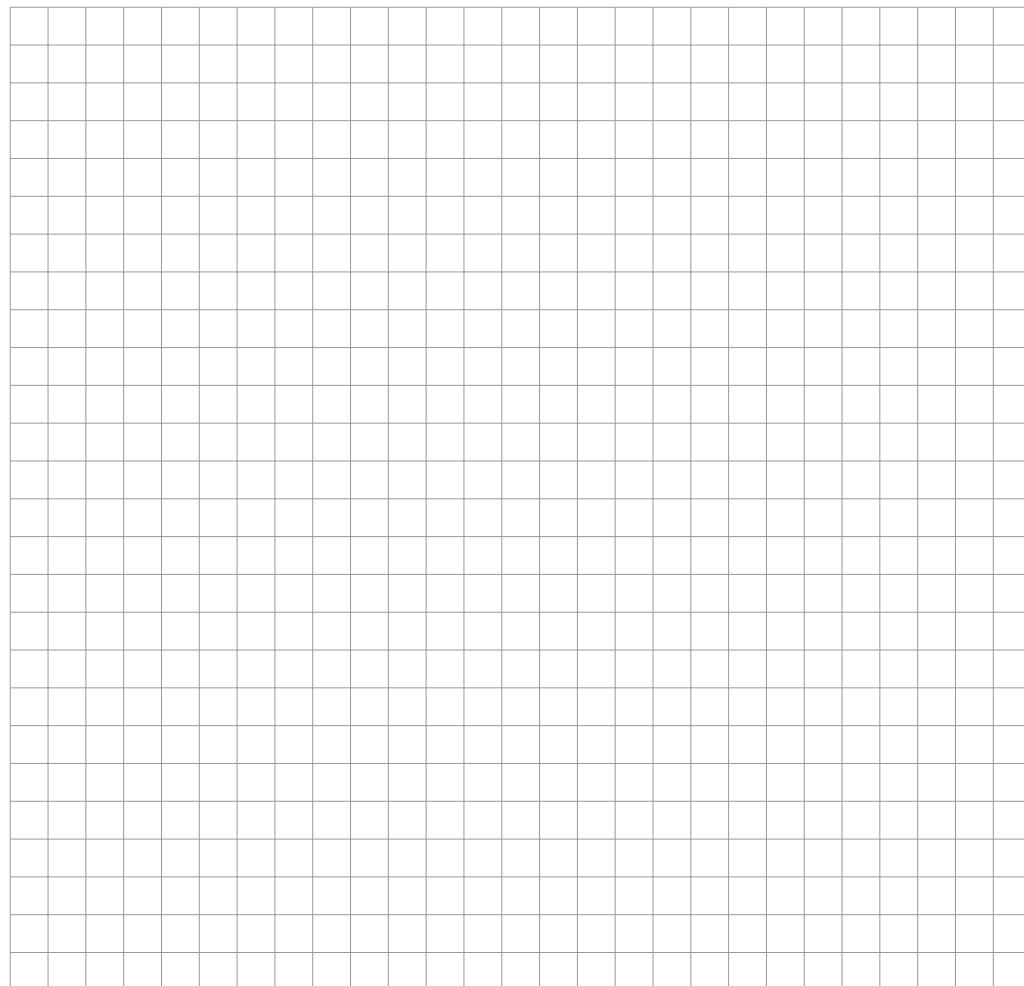
$$\det {}^0R_B = +1. \quad (2.16)$$

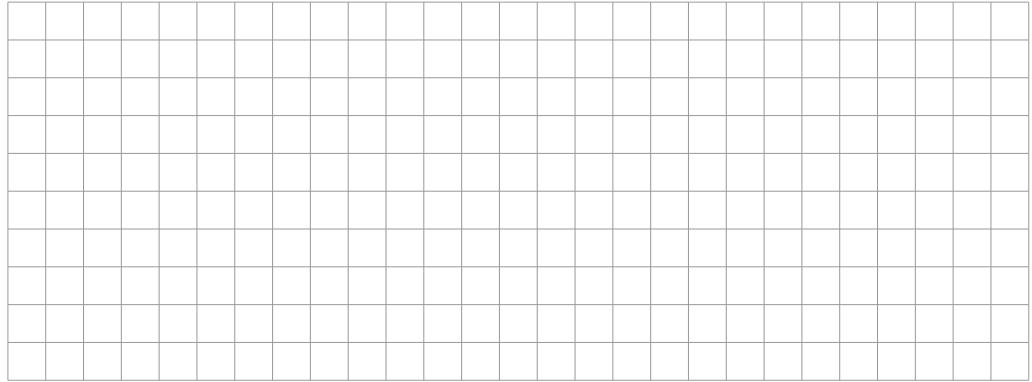
Aufgrund dieser sechs Nebenbedingungen sind drei Parameter zur eindeutigen Beschreibung einer allgemeinen Drehung ausreichend.

Die Spaltenvektoren von 0R_B drücken die Richtung der Basisvektoren von $(KS)_B$ im Koordinatensystem $(KS)_0$ aus. Gedanklich sind die beiden Koordinatensysteme $(KS)_B$ und $(KS)_0$ gegeneinander verdreht, jedoch nicht verschoben, d. h. die Ursprünge sind identisch (reine Be- trachtung der Orientierung).

Mittels Rotationsmatrizen können Richtungsvektoren einfach in einem anderen Koordinatensystem ausgedrückt werden, da die Translation hierbei keine Beachtung findet (Erinnerung: Ein Richtungsvektor kann im Raum verschoben werden, da er ein Repräsentant eines Vektorfeldes ist). Bei Ortsvektoren ist noch die Verschiebung zu berücksichtigen, da sie immer vom Ursprung eines Koordinatensystems ausgehen und nicht verschoben werden können.

Beispiel





2.3.1 Elementardrehungen

Bevor beliebige Drehungen betrachtet werden, erfolgt zunächst eine Einführung von Drehungen um einzelne Achsen (sogenannte Elementardrehungen). Diese sind zusammen mit den zugehörigen Rotationsmatrizen im Folgenden zusammengefasst. Die Spaltenvektoren beschreiben die Abbildung der Basisvektoren des gedrehten Koordinatensystems auf das ursprüngliche.

- Drehung um die x -Achse (Bild 2.9(a)): $\mathbf{R}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$,
- Drehung um die y -Achse (Bild 2.9(b)): $\mathbf{R}_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$,
- Drehung um die z -Achse (Bild 2.9(c)): $\mathbf{R}_z(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

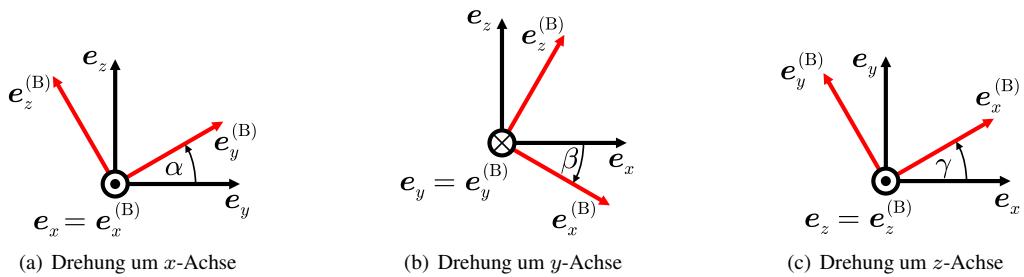


Bild 2.9: Elementardrehungen

2.3.2 Zusammengesetzte Drehungen

Zu beachten ist, dass bei aufeinanderfolgenden Drehungen die Reihenfolge der einzelnen Rotationen wichtig ist. Drehungen sind (ebenso wie Matrizenmultiplikationen) nicht kommutativ $\mathbf{R}_1 \mathbf{R}_2 \neq \mathbf{R}_2 \mathbf{R}_1$, wie in Bild 2.10 dargestellt.

Zur Beschreibung zusammengesetzter Drehungen existieren verschiedene Konventionen, z. B. Kardan-, Euler- und RPY-Winkel, die nachfolgend vorgestellt werden. Wie den Abschnitten zu entnehmen ist, spielt die Reihenfolge der Multiplikationen eine entscheidende Rolle.

Die sogenannte **Rechtsmultiplikation** (also Multiplikation der Elementarrotationen von links nach rechts) beschreibt die Rotation um das jeweils **aktuelle** (mitgedrehte) Koordinatensystem.

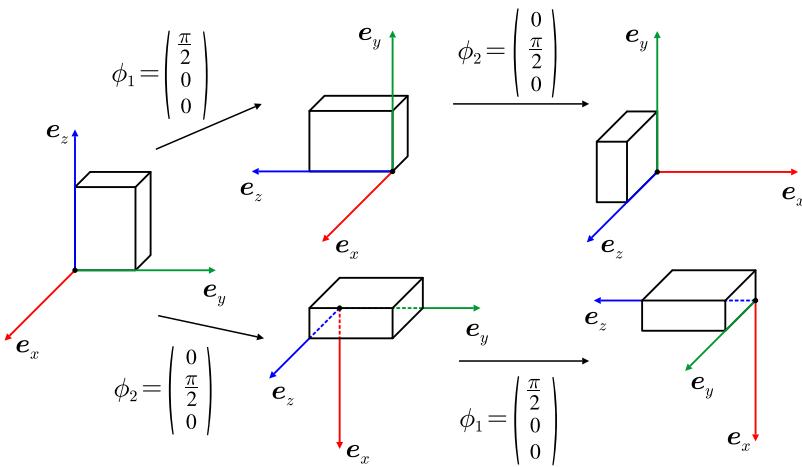


Bild 2.10: Nichtkommutativität der endlichen Drehungen

Die **Linksmultiplikation** (von rechts nach links) hingegen beschreibt die Rotation immer um das **feste**, „alte“ Koordinatensystem.

Zunächst liegen das Ursprungskoordinatensystem und das zu drehende Koordinatensystem über-einander. Die beiden Koordinatensysteme sind also koinzident. Das gedrehte Koordinatensystem wird nun schrittweise, durch Elementarrotationen aus dem Ursprungskoordinatensystem heraus-gedreht.

Kardan-Winkel

Drehungen um die Achsen des mitgedrehten Systems: Basisvektoren des gedrehten Systems werden durch drei aufeinanderfolgende Drehungen um die Achsen des mitgedrehten Systems „erzeugt“. Man erhält die Resultierende durch **Rechtsmultiplikation** der Einzeldrehungen:

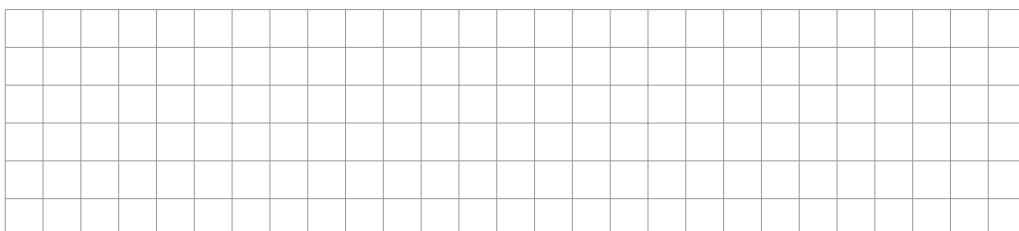
1. Rotation mit dem Winkel α um x -Achse
2. Rotation mit dem Winkel β um neue y -Achse
3. Rotation mit dem Winkel γ um neueste z -Achse

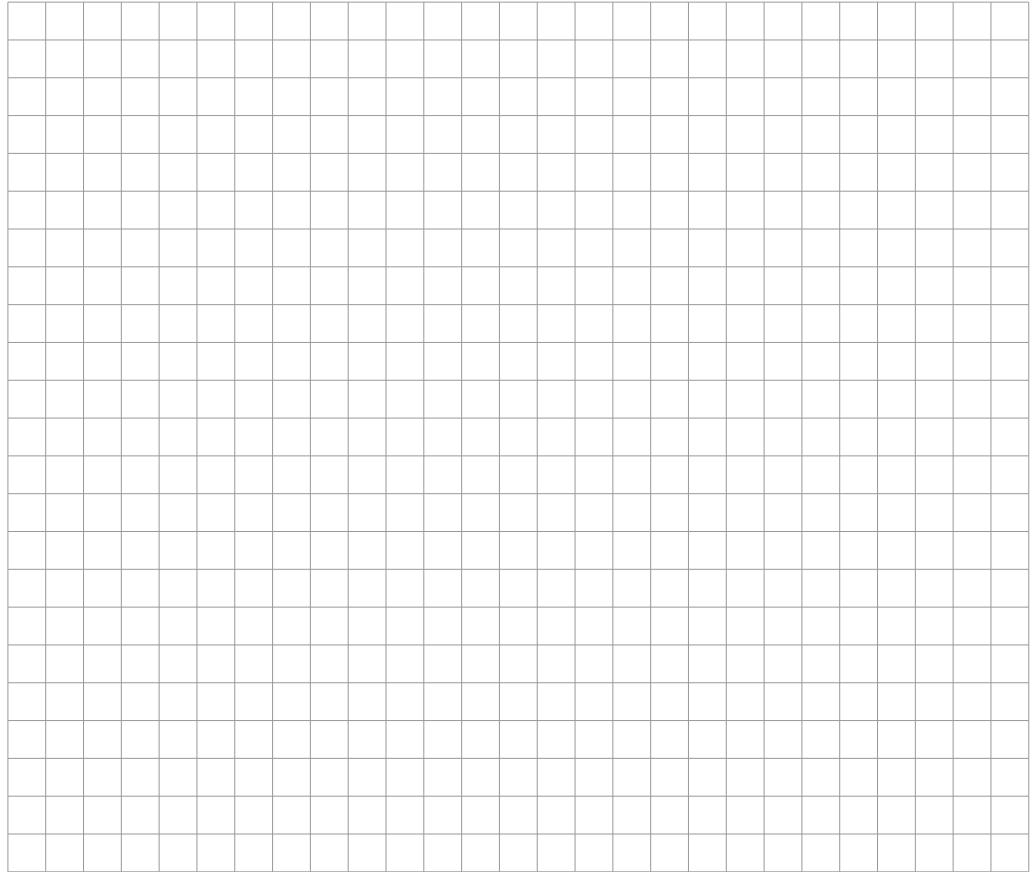
Definition
Kardan-Winkel

Die zusammengesetzte Rotationsmatrix nach Kardan lautet somit:

$$\begin{aligned}
 R_{\text{KARD}}(\alpha, \beta, \gamma) &= \underbrace{\overbrace{R_x(\alpha)}^{1.} \overbrace{R_y(\beta)}^{2.} \overbrace{R_z(\gamma)}^{3.}}_{\text{Rechtsmultiplikation}} \\
 &= \begin{pmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -s_\alpha c_\beta \\ s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma & s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta \end{pmatrix}, \tag{2.17}
 \end{aligned}$$

mit den Abkürzungen $c_\alpha = \cos(\alpha)$, $s_\alpha = \sin(\alpha)$, usw.





Gimbal-Lock Problematik

Zusammengesetzten Drehungen sind eine sehr intuitive Orientierungsdarstellung. Allerdings kann die Verwendung von Rechtsmultiplikation der Einzeldrehungen (mitgedrehte Koordinatensysteme) zur sogenannten **Gimbal-Lock** Problematik führen.

Definition Gimbal-Lock

Diese Phänomen tritt auf, wenn zwei Rotationsachsen zusammenfallen - der resultierende Winkel lässt sich nur als Summe von zwei Freiheitsgraden der Orientierungsdarstellung ausdrücken. Für das inverse Problem entstehen unendlich viele Lösungen (eine Gleichung mit zwei Unbekannten). Durch Zusammenfallen zweier Rotationsachsen kommt es gleichzeitig zum Verlust eines Freiheitsgrades, da insgesamt statt der erforderlichen drei unabhängigen Rotationen nur noch zwei zur Verfügung stehen (singuläre Stellung). Ist eine (wenn auch kleine) Änderung um die Achse des „gesperrten“ Freiheitsgrad gewünscht, so sind zunächst große Änderungen in Richtung der anderen Achsen erforderlich.

Bei Kardan-Winkeln entsteht die Gimbal-Lock Problematik, wenn der zweite Rotationswinkel $\pm \frac{\pi}{2}$ entspricht (vgl. Bild 2.11).

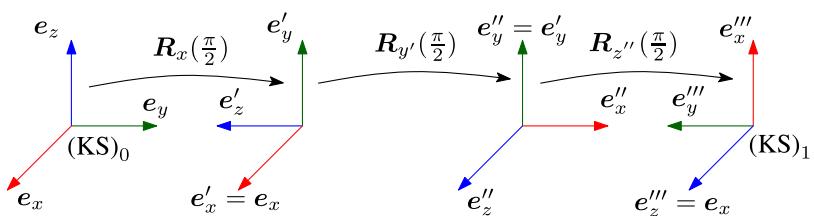


Bild 2.11: Gimbal-Lock-Problematik mit Kardan-Winkel. Unabhängig von dem ersten und dritten Drehwinkel ist es unmöglich, um die ursprüngliche $(0)e_z$ Achse zu drehen, weil ein Rotationswinkel von $\pm \frac{\pi}{2}$ um die e'_y zum Aufeinanderfallen zweier Drehachsen (e_x und e'''_z) führt.

Euler-Winkel

Drehungen um die Achsen des mitgedrehten Systems: Man erhält die Resultierende durch Rechtsmultiplikation der Einzeldrehungen.

Definition
Euler-Winkel

Unglücklicherweise existieren verschiedene Definitionen für Euler-Winkel. Sie unterscheiden sich in der Wahl der Drehachsen. Von insgesamt zwölf möglichen Beschreibungen haben sich drei etabliert:

- x-Konvention:

$$\mathbf{R}_{\text{EUL,x}}(\gamma, \alpha, \gamma') = \overbrace{\mathbf{R}_z(\gamma)}^{1.} \overbrace{\mathbf{R}_x(\alpha)}^{2.} \overbrace{\mathbf{R}_z(\gamma')}^{3.}. \quad (2.18)$$

- y-Konvention:

$$\mathbf{R}_{\text{EUL,y}}(\gamma, \beta, \gamma') = \overbrace{\mathbf{R}_z(\gamma)}^{1.} \overbrace{\mathbf{R}_y(\beta)}^{2.} \overbrace{\mathbf{R}_z(\gamma')}^{3.}. \quad (2.19)$$

- Luftfahrtform:

$$\mathbf{R}_{\text{EUL,LF}}(\gamma, \beta, \alpha) = \overbrace{\mathbf{R}_z(\gamma)}^{1.} \overbrace{\mathbf{R}_y(\beta)}^{2.} \overbrace{\mathbf{R}_x(\alpha)}^{3.}. \quad (2.20)$$

RPY-Winkel

Drehungen um die festen Achsen des Basissystems: Basisvektoren des gedrehten Systems werden durch drei aufeinanderfolgende Drehungen um die Achsen des festen Originalsystems „erzeugt“. Man erhält die Resultierende durch **Linksmultiplikation** der einzelnen Elementardrehungen.

Definition
RPY-Winkel

1. Rotation um x -Achse (gieren, engl. yaw)
2. Rotation um y -Achse (nicken, engl. pitch)
3. Rotation um z -Achse (rollen, engl. roll)

$$\begin{aligned} \mathbf{R}_{\text{RPY}}(\alpha, \beta, \gamma) &= \underbrace{\mathbf{R}_z(\gamma)}_{\text{Linksmultiplikation}} \underbrace{\mathbf{R}_y(\beta)}^{2.} \underbrace{\mathbf{R}_x(\alpha)}^{1.} \\ &= \begin{pmatrix} c_\gamma c_\beta & c_\gamma s_\beta s_\alpha - s_\gamma c_\alpha & c_\gamma s_\beta c_\alpha + s_\gamma s_\alpha \\ s_\gamma c_\beta & s_\gamma s_\beta s_\alpha + c_\gamma c_\alpha & s_\gamma s_\beta c_\alpha - c_\gamma s_\alpha \\ -s_\beta & c_\beta s_\alpha & c_\beta c_\alpha \end{pmatrix}. \end{aligned} \quad (2.21)$$

2.3.3 Direkter und inverser Zusammenhang

Unter dem **direkten Zusammenhang** versteht man die Berechnung der resultierenden Rotationsmatrix ausgehend von gegebenen Drehwinkeln einer Winkeldarstellung. In Abhängigkeit der gewählten Darstellung ist hier die Links- oder Rechtsmultiplikation anzuwenden. Bild 2.12 zeigt dies beispielhaft für die RPY-Winkel.

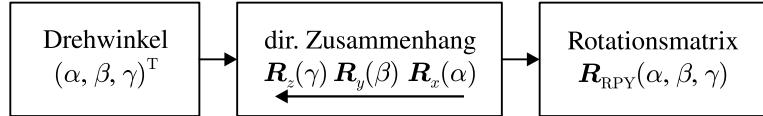


Bild 2.12: Direkter Zusammenhang

Der **inverse Zusammenhang** beschreibt die Bestimmung der Drehwinkel bei gegebener Rotationsmatrix (vgl. Bild 2.13). Hierfür ist ein nichtlineares Gleichungssystem zu lösen. Der inverse Zusammenhang ist wichtiger Bestandteil der in Kapitel 3 behandelten inversen Kinematik (weiterführende Details und Beispiele in der Übung).

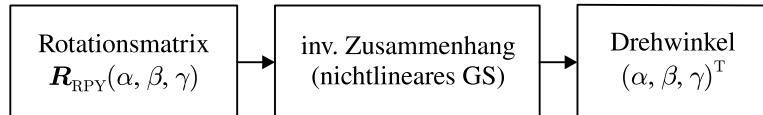


Bild 2.13: Inverser Zusammenhang

2.3.4 Zusammenfassung

Abschließend erfolgt eine knappe Zusammenfassung der wichtigsten Eigenschaften der Orientierungsrepräsentation durch Drehwinkel bzw. unter Verwendung von Rotationsmatrizen.

Vorteile:

- einfache Notation,
- intuitiv nachvollziehbar,
- wenig Parameter.

Nachteile:

- Unter gleichem Namen existieren verschiedene Konventionen (bis zu zwölf Möglichkeiten bei Euler-Winkel),
- Reihenfolge der Multiplikation ist wichtig,
- Gimbal-Lock Problematik bei mitgedrehten Koordinatensystemen,
- Die Interpolation von Winkeln führt zu unvorhergesehenen Pfaden.

2.4 Homogene Transformationen

Wie zuvor dargestellt, setzt sich die Lage eines Objekts im Raum aus Orientierung und Position bezüglich eines Basiskoordinatensystems zusammen. Bisher wurde nur die Orientierung betrachtet. Die noch ausstehende Translation ${}^{(0)}\mathbf{r}_B$ beschreibe nun die Position des Ursprungs des Koordinatensystems $(KS)_B$ im Koordinatensystem $(KS)_0$. Somit ergibt sich für die Abbildung des Ortsvektors ${}^{(B)}\mathbf{r}_P$ von $(KS)_B$ nach $(KS)_0$ folgender Zusammenhang:

$${}^{(0)}\mathbf{r}_P = \underbrace{{}^{(0)}\mathbf{r}_B}_{\text{Translation}} + \underbrace{{}^0\mathbf{R}_B {}^{(B)}\mathbf{r}_P}_{\text{Rotation}}. \quad (2.22)$$

Bild 2.14 veranschaulicht den Zusammenhang nach Gleichung (2.22).

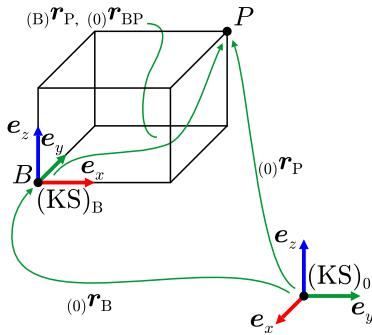


Bild 2.14: Translation und Rotation zur Beschreibung von Ortsvektoren

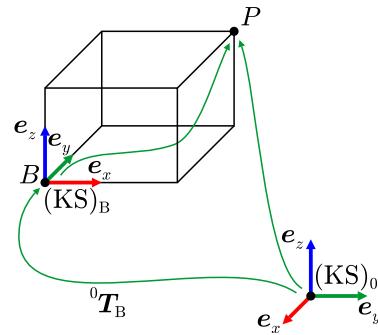


Bild 2.15: Homogene Transformation ${}^0\mathbf{T}_B$

Für die inverse Abbildung von $(KS)_0$ nach $(KS)_B$ gilt:

$${}^{(B)}\mathbf{r}_P = -({}^0\mathbf{R}_B)^T {}^{(0)}\mathbf{r}_B + ({}^0\mathbf{R}_B)^T {}^{(0)}\mathbf{r}_P = -{}^B\mathbf{R}_0 {}^{(0)}\mathbf{r}_B + {}^B\mathbf{R}_0 {}^{(0)}\mathbf{r}_P . \quad (2.23)$$

Durch die Einführung einer homogenen Transformationsmatrix (Dimension: 4×4) ${}^0\mathbf{T}_B$ wird die getrennte Betrachtung von Rotation und Translation zu einer kompakten Einheit zusammengefasst und erlaubt eine einheitliche Behandlung. Die Matrix ${}^0\mathbf{T}_B$ beschreibt die Lage des Koordinatensystems $(KS)_B$ im Koordinatensystem $(KS)_0$ (vgl. auch Bild 2.15):

$${}^0\mathbf{T}_B = \left(\begin{array}{c|c} {}^0\mathbf{R}_B & {}^{(0)}\mathbf{r}_B \\ \hline 0 & 1 \end{array} \right), \quad \text{mit } {}^0\mathbf{T}_B \in \text{SE}(3) = \{\text{SO}(3) \times \mathbb{R}^3\}, \quad (2.24)$$

wobei SE(3) für „special Euclidian“ im dreidimensionalen Raum steht.

Für die Determinante von ${}^0\mathbf{T}_B$ gilt:

$$\det({}^0\mathbf{T}_B) = 1. \quad (2.25)$$

Die Inverse von ${}^0\mathbf{T}_B$ beschreibt die Lage von $(KS)_0$ bezüglich $(KS)_B$ und berechnet sich zu:

$$({}^0\mathbf{T}_B)^{-1} = {}^B\mathbf{T}_0 = \left(\begin{array}{c|c} ({}^0\mathbf{R}_B)^T & -({}^0\mathbf{R}_B)^T {}^{(0)}\mathbf{r}_B \\ \hline 0 & 1 \end{array} \right). \quad (2.26)$$

Durch homogene Transformationsmatrizen wird nicht nur die relative Lage von zwei Koordinatensystemen zueinander beschrieben. Es lassen sich auch Ortsvektoren von einem Koordinatensystem in ein anderes umrechnen.

Definition

homogener Ortsvektor

Hierzu wird der Ortsvektor $(0)\mathbf{r}_P$ zum homogenen Ortsvektor $(0)\mathbf{x}_P$ erweitert:

$$(0)\mathbf{x}_P = ((0)\mathbf{r}_P^T, 1)^T = ((0)x_P, (0)y_P, (0)z_P, 1)^T. \quad (2.27)$$

Der homogene Ortsvektor $(0)\mathbf{x}_P$ beschreibt einen Vektor vom Ursprung von $(KS)_0$ zum Punkt P in $(KS)_0$.

Der homogene Ortsvektor vom Ursprung von $(KS)_B$ zu P in $(KS)_B$ lautet:

$$(B)\mathbf{x}_P = ((B)\mathbf{r}_P^T, 1)^T = ((B)x_P, (B)y_P, (B)z_P, 1)^T. \quad (2.28)$$

Somit ergibt sich folgende Transformationsvorschrift, um Ortsvektoren vom Koordinatensystem $(KS)_B$ in das Koordinatensystem $(KS)_0$ zu transformieren, vgl. hierzu Gleichung (2.22):

$$(0)\mathbf{x}_P = {}^0\mathbf{T}_B (B)\mathbf{x}_P = \left(\begin{array}{ccc|c} {}^0\mathbf{R}_B & & & (0)\mathbf{r}_B \\ 0 & 0 & 0 & 1 \end{array} \right) \left(\begin{array}{c} (B)\mathbf{r}_P \\ 1 \end{array} \right) = \left(\begin{array}{c} (0)\mathbf{r}_B + {}^0\mathbf{R}_B (B)\mathbf{r}_P \\ 1 \end{array} \right). \quad (2.29)$$

Die inverse Transformation lautet:

$$(B)\mathbf{x}_P = {}^B\mathbf{T}_0 (0)\mathbf{x}_P = ({}^0\mathbf{T}_B)^{-1} (0)\mathbf{x}_P = \left(\begin{array}{c} ({}^0\mathbf{R}_B)^T (0)\mathbf{r}_P - ({}^0\mathbf{R}_B)^T (0)\mathbf{r}_B \\ 1 \end{array} \right), \quad (2.30)$$

sie bildet homogene Ortsvektoren von $(KS)_0$ in $(KS)_B$ ab, vgl. hierzu Gleichung (2.23) mit $({}^0\mathbf{R}_B)^T = {}^B\mathbf{R}_0$:

$$(B)\mathbf{r}_P = -({}^0\mathbf{R}_B)^T (0)\mathbf{r}_B + ({}^0\mathbf{R}_B)^T (0)\mathbf{r}_P. \quad (2.31)$$

Formal lassen sich homogene Transformationsmatrizen auch auf (Richtungs-)Vektoren erweitern.

Definition
homogener
Richtungsvektor

In Analogie zu Gleichung (2.27) wird der Richtungsvektor ebenfalls um eine Dimension zu einem homogenen Richtungsvektor erweitert, man erhält für $(0)\mathbf{a}$ im $(KS)_0$:

$$(0)\mathbf{a}' = ((0)\mathbf{a}^T, 0)^T. \quad (2.32)$$

Wendet man nun die homogene Transformation ${}^B\mathbf{T}_0$ auf $(0)\mathbf{a}'$ an, so ergibt sich:

$$(B)\mathbf{a}' = {}^B\mathbf{T}_0 (0)\mathbf{a}' = \left(\begin{array}{c} {}^B\mathbf{R}_0 (0)\mathbf{a} \\ 0 \end{array} \right), \quad (2.33)$$

der Vektor wird nun im $(KS)_B$ ausgedrückt. Die relative Verschiebung $(0)\mathbf{r}_B$ der beiden Koordinatensysteme $(KS)_0$ und $(KS)_B$ spielt dabei keine Rolle (zur Erinnerung: Ein (Richtungs-)Vektor ist Repräsentant eines Vektorfeldes, kann parallel frei verschoben werden und hängt nicht an einem Koordinatenursprung).

Abschließend sei noch angemerkt, dass ein homogener Richtungsvektor aus der Differenz zweier homogener Ortsvektoren entsteht:

$$(0)\mathbf{a} = (0)\mathbf{x}_2 - (0)\mathbf{x}_1 = \left(\begin{array}{c} (0)x_2 - (0)x_1 \\ (0)y_2 - (0)y_1 \\ (0)z_2 - (0)z_1 \\ 0 \end{array} \right), \quad (2.34)$$

und in Analogie zum (Richtungs-)Vektor die Verschiebung zwischen $(0)\mathbf{x}_1$ und $(0)\mathbf{x}_2$ angibt.

Verkettung homogener Transformationen

Ebenso wie Rotationsmatrizen erlauben homogene Transformationen eine Verkettung. So entsteht die Transformation von $(KS)_n$ nach $(KS)_0$

$$(0)\mathbf{x} = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \cdots {}^{n-2}\mathbf{T}_{n-1} {}^{n-1}\mathbf{T}_n (n)\mathbf{x} = {}^0\mathbf{T}_n (n)\mathbf{x} \quad (2.35)$$

aus der Multiplikation der gliedweisen Transformation ${}^i\mathbf{T}_j$ nach Bild 2.16. Ist die Transformation

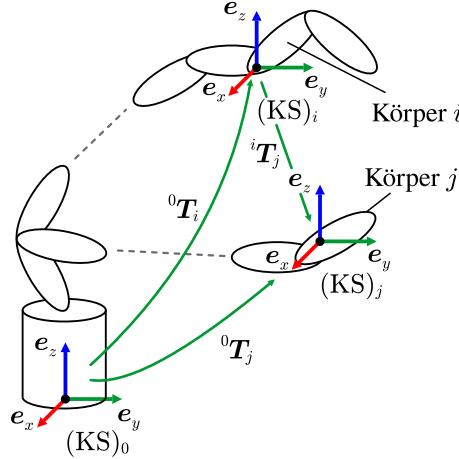


Bild 2.16: Verkettung von Transformationen

von $(KS)_i$ nach $(KS)_0$ ${}^0\mathbf{T}_i$ sowie von $(KS)_j$ nach $(KS)_0$ ${}^0\mathbf{T}_j$ bekannt, so kann die relative Transformation von $(KS)_j$ nach $(KS)_i$ gemäß folgender Gleichung bestimmt werden:

$${}^i\mathbf{T}_j = ({}^0\mathbf{T}_i)^{-1} {}^0\mathbf{T}_j = {}^i\mathbf{T}_0 {}^0\mathbf{T}_j. \quad (2.36)$$

2.5 Alternative Rotationsdarstellungen

Die Verwendung von Rotationsmatrizen und Winkelkonventionen ist nicht die einzige Möglichkeit Orientierungen im Raum zu repräsentieren. Ebenso sind in der Robotik bzw. bei Mehrkörpersystemen im Allgemeinen die Achse-Winkel-Darstellung (Abschnitt 2.5.1) und die Darstellung durch Quaternionen (Abschnitt 2.5.2) gebräuchlich.

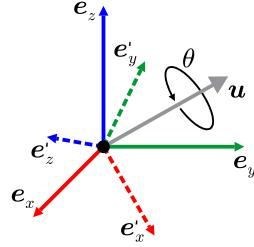
2.5.1 Achse-Winkel-Darstellung

Eine allgemeine Rotation mit 3 Winkeln (z. B. Euler-Winkel) kann äquivalent durch die Rotation um eine Achse \mathbf{u} mit Rotationswinkel θ dargestellt werden (siehe Bild 2.17). Da für die Rotationsachse $\|\mathbf{u}\|_2 = 1$ gilt, stehen für diese Darstellungsform drei Parameter zur Verfügung.

Bei gegebener Achse-Winkel-Darstellung (\mathbf{u}, θ) erhält man für die äquivalente Rotationsmatrix (ohne Beweis)

$$\mathbf{R}_u = \begin{pmatrix} u_x^2(1 - c_\theta) + c_\theta & u_x u_y(1 - c_\theta) - u_z s_\theta & u_x u_z(1 - c_\theta) + u_y s_\theta \\ u_x u_y(1 - c_\theta) + u_z s_\theta & u_y^2(1 - c_\theta) + c_\theta & u_y u_z(1 - c_\theta) - u_x s_\theta \\ u_x u_z(1 - c_\theta) - u_y s_\theta & u_y u_z(1 - c_\theta) + u_x s_\theta & u_z^2(1 - c_\theta) + c_\theta \end{pmatrix}, \quad (2.37)$$

mit $\mathbf{u} = (u_x, u_y, u_z)^T$ und $c_\theta = \cos(\theta)$, $s_\theta = \sin(\theta)$.

**Bild 2.17:** Rotation um eine Achse \mathbf{u} mit dem Winkel θ

Um aus einer gegebenen Rotationsmatrix \mathbf{R}^* die zugehörige Achse-Winkel-Präsentation zu bestimmen, wird wie folgt vorgegangen. Die gegebene Rotationsmatrix \mathbf{R}^* laute:

$$\mathbf{R}^* = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, \quad (2.38)$$

mit

$$\begin{aligned} \text{Spur}(\mathbf{R}^*) &= r_{11} + r_{22} + r_{33} \stackrel{!}{=} \text{Spur}(\mathbf{R}_u) = \text{Spur}(\mathbf{R}(u, \theta)) \\ &= \underbrace{(u_x^2 + u_y^2 + u_z^2)}_1 (1 - c_\theta) + 3 c_\theta \\ &= 1 + 2 c_\theta. \end{aligned} \quad (2.39)$$

Somit erhält man:

$$\cos(\theta) = \frac{1}{2} (r_{11} + r_{22} + r_{33} - 1). \quad (2.40)$$

Da die Umkehrfunktion \arccos nicht eindeutig ist (Definitionsbereich: $[0; \pi]$), ist θ noch nicht eindeutig bestimmt. Ein weiterer Koeffizientenvergleich zwischen Gleichung (2.37) und Gleichung (2.38) ergibt:

$$\begin{aligned} r_{32} - r_{23} &= 2 u_x s_\theta, \\ r_{13} - r_{31} &= 2 u_y s_\theta, \\ r_{21} - r_{12} &= 2 u_z s_\theta. \end{aligned} \quad (2.41)$$

Durch Quadrieren und Addieren erhält man:

$$\sin(\theta) = \pm \frac{1}{2} \sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}. \quad (2.42)$$

Gleichung (2.42) liefert zwei Lösungen, wobei das negative Vorzeichen nicht zu berücksichtigen ist, da es zur gleichen Drehung führt wie das positive Vorzeichen (siehe Gleichung (2.44) unten). Aus Gleichung (2.40) und Gleichung (2.42) kann der Winkel θ nun eindeutig bestimmt werden:

$$\theta = \arctan2(\sin(\theta), \cos(\theta)). \quad (2.43)$$

Die Drehachse ergibt sich nach Gleichung (2.41) zu:

$$\mathbf{u} = (u_x, u_y, u_z)^T = \frac{1}{2 \sin(\theta)} (r_{32} - r_{23}, r_{13} - r_{31}, r_{21} - r_{12})^T, \quad (2.44)$$

wobei für $|\sin(\theta)| \approx 0$ (also $\theta \approx 0$ und $\theta \approx \pm\pi$) der Nenner gegen null strebt.

Die **Vorteile** der Achse-Winkel-Darstellung sind:

- Zur Beschreibung werden wenige Parameter benötigt,
- gut für die Interpolation von Rotationen geeignet,
- vergleichsweise intuitive Darstellung,
- singularitätsfreie Darstellung.

Nachteilig ist allerdings:

- Verkettung von Rotationen (z. B. durch Multiplikation) nicht möglich.

2.5.2 Quaternionen

Aufbauend auf der Achse-Winkel-Darstellung wird nun die Repräsentation von Orientierungen durch Quaternionen dargestellt. Mittels der von Hamilton¹ 1843 eingeführten Quaternionen lassen sich Rotationen kompakt formulieren.

Bei Quaternionen handelt es sich um eine Erweiterung der komplexen Zahlen und es gelte für ein Quaternion q :

$$q = q_0 + q_1 i + q_2 j + q_3 k, \quad (2.45)$$

mit

$$q_i \in \mathbb{R} \text{ und } i^2 = j^2 = k^2 = ijk = -1. \quad (2.46)$$

Definition
Quaternionen

Das Quaternion q sei zusammengesetzt aus Skalar s und Vektorteil v :

$$q = (s, v^T)^T, \quad \text{mit } s = q_0, v = (q_1, q_2, q_3)^T. \quad (2.47)$$

Es gelten nachstehende Definitionen

- Adjungierte:

$$\bar{q} = (s, -v^T)^T, \quad (2.48)$$

- zweier Norm:

$$\|q\|_2 = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}, \quad (2.49)$$

- Inverse:

$$q^{-1} = \frac{\bar{q}}{\|q\|_2}, \quad (2.50)$$

- neutrales Quaternion:

$$q_E = (1, (0, 0, 0))^T, \quad (2.51)$$

¹W. R. Hamilton, Mathematiker und Physiker, 1805 - 1865

- Quaternionenmultiplikation:

$$\begin{aligned}\mathbf{q}_3 = \mathbf{q}_1 \mathbf{q}_2 &= (s_1 s_2 - \mathbf{v}_1^T \mathbf{v}_2, (s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2)^T)^T \\ &= (s_3, \mathbf{v}_3^T)^T.\end{aligned}\quad (2.52)$$

Wie in Abschnitt 2.5.1 wird die Darstellung der Orientierung durch eine Rotation um eine Achse, beschrieben durch den Einheitsvektor \mathbf{u} mit dem Winkel θ , ausgedrückt (vgl. Bild 2.17).

Das Einheitsquaternion ($\|\mathbf{q}\|_2 = 1$) wird nun verwendet, um aufbauend auf der Achse-Winkel-Darstellung die Rotation zu repräsentieren:

$$\mathbf{q} = (s, \mathbf{v}^T)^T, \quad (2.53)$$

mit

$$s = q_0 = \cos\left(\frac{\theta}{2}\right) \text{ und } \mathbf{v} = \mathbf{u} \sin\left(\frac{\theta}{2}\right). \quad (2.54)$$

Die Umrechnung von Quaternionen in die Rotationsmatrixdarstellung geschieht nach²:

$$\mathbf{R} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (2.55)$$

Die Multiplikation zweier Rotationsmatrizen entspricht dem Produkt der jeweiligen Quaternionen, so dass gilt:

$${}^i\mathbf{R}_j \hat{=} \mathbf{q}_{ij}, \quad (2.56)$$

bzw.

$${}^0\mathbf{R}_1 {}^1\mathbf{R}_2 \cdots {}^{n-1}\mathbf{R}_n \hat{=} \mathbf{q}_{01} \mathbf{q}_{12} \cdots \mathbf{q}_{n-1,n}. \quad (2.57)$$

Bei gegebener Rotationsmatrix können nun, wie in Abschnitt 2.5.1 dargestellt, die dazugehörige Achse-Winkel-Darstellung (und somit das dazugehörige Quaternion) oder wie in Kapitel 3 beschrieben (inverse Kinematik) die dazugehörigen Rotationswinkel berechnet werden.

Die Vor- und Nachteile der Quaternionen sind im Folgenden knapp zusammengefasst:

Vorteile:

- Zur Beschreibung werden wenige Parameter benötigt,
- durch Quaternionenmultiplikation ist eine Verkettung von Rotationen möglich,
- auch komplexe Interpolationen sind möglich (Slerp, vgl. Kapitel 6),
- die Rechentechnik gestaltet sich effektiver als bei Rotationsmatrizen,
- die Darstellung von Rotationen mittels Quaternionen ist singularitätsfrei.

Nachteile:

- Es handelt sich um keine intuitive Darstellung.

²Details finden sich in J. Diebel. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*. Techn. Ber. Stanford, USA: Stanford University, 2006

Fragen zur Selbstkontrolle

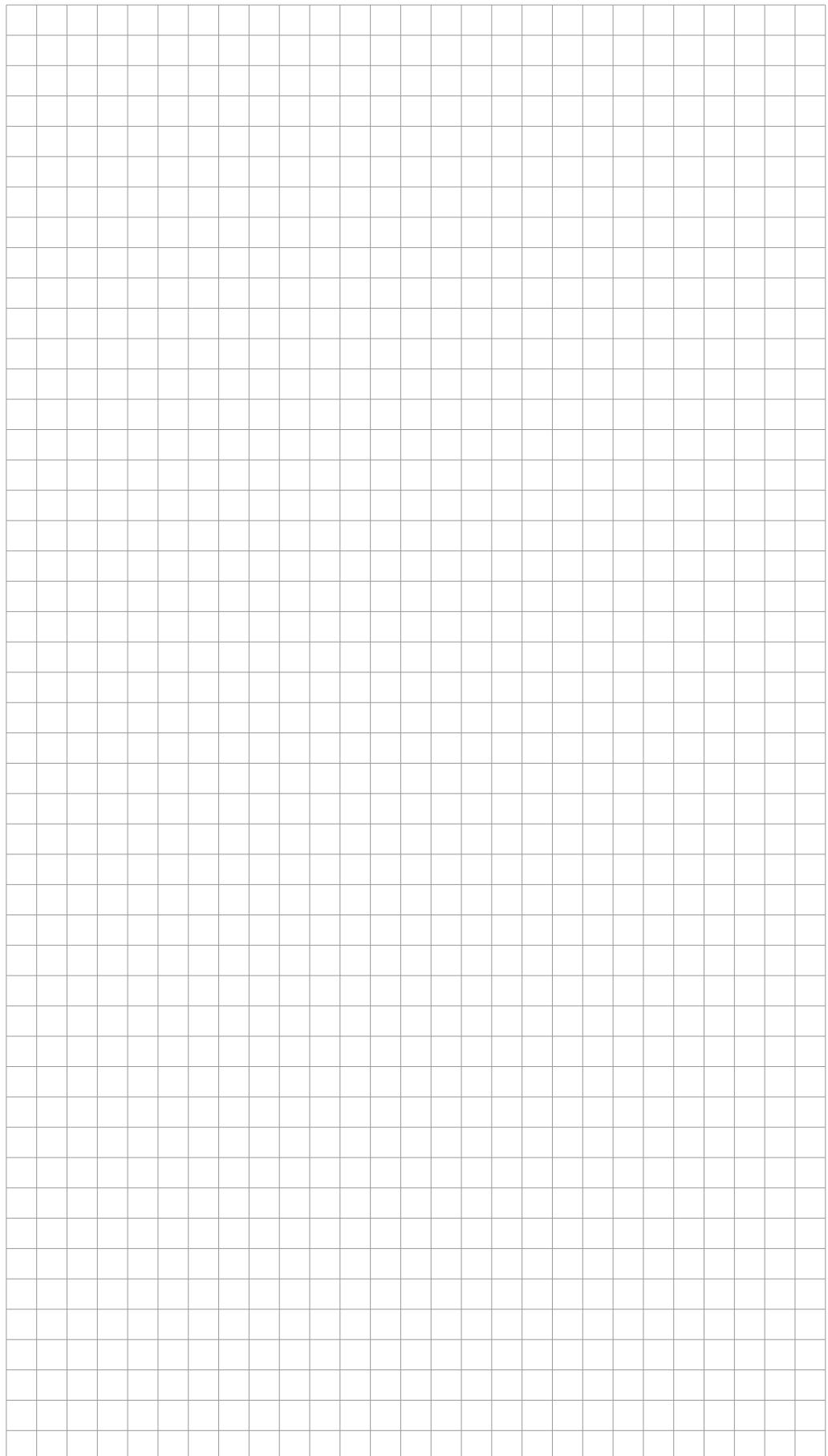
1. Welche Möglichkeiten zur Beschreibung von Rotationen sind Ihnen bekannt? Erläutern Sie die Unterschiede und nennen Sie Vor- und Nachteile der einzelnen Ansätze!
 2. Nennen Sie drei Arten zusammengesetzter Drehungen und geben Sie jeweils die Rotationsvorschriften in Abhängigkeit von Elementardrehungen an!
 3. Bestimmen Sie aus der gegebenen Gesamt-Rotationsmatrix R die Drehwinkel um die einzelnen Rotationsachsen bei einer Rotation nach Kardan!

Gegeben:

$$\boldsymbol{R}(\alpha, \beta, \gamma) = \begin{pmatrix} r_{(1,1)} & r_{(1,2)} & r_{(1,3)} \\ r_{(2,1)} & r_{(2,2)} & r_{(2,3)} \\ r_{(3,1)} & r_{(3,2)} & r_{(3,3)} \end{pmatrix}$$

$$\boldsymbol{R}_{\text{KARD}}(\alpha, \beta, \gamma) = \begin{pmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -s_\alpha c_\beta \\ s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma & s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta \end{pmatrix}$$





3 Direkte und inverse Kinematik

Dieses Kapitel beschreibt die direkte und inverse Kinematik eines Roboters. Unter direkter Kinematik versteht man die Berechnung der Endeffektorlage aus den bekannten Gelenkwinkeln (Abschnitt 3.2). Ein wichtiges Hilfsmittel zur Berechnung der direkten Kinematik ist die in Abschnitt 3.3 eingeführte Denavit-Hartenberg Notation. Die inverse Kinematik (Abschnitt 3.4) dient umgekehrt dazu, Gelenkwinkel ausgehend von einer vorgegebenen Endeffektorlage zu bestimmen. Der Zusammenhang zwischen direkter Kinematik und inverser Kinematik ist in Bild 3.1 grafisch dargestellt.

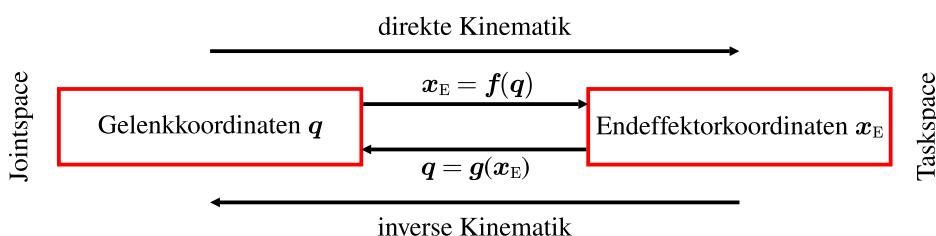


Bild 3.1: Grafische Abgrenzung von direkter und inverser Kinematik

Als weiterführende Literatur wird empfohlen:

- W. Khalil und E. Dombre. *Modeling, Identification & Control of Robots*. New York, USA: Routledge, 2002, S. 35–83
- F. Pfeiffer und E. Reithmeier. *Roboterdynamik*. Stuttgart, Deutschland: Teubner Verlag, 1987, S. 27–59
- W. Khalil und J. Kleinfinger. „A new Geometric Notation for Open and Closed-Loop Robots“. In: *Proc. of the 1986 IEEE International Conference on Robotics and Automation*. San Francisco, USA, 1986, S. 1174–1179
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 68–141

3.1 Kinematik serieller Roboter

Es folgt zunächst die Definition des Begriffs Roboterkinematik. Sie bildet die Grundlage für die weiteren Ausführungen. In diesem Kapitel werden ausschließlich serielle Kinematiken (offene kinematische Kette) behandelt.

Definition
 Roboterkinematik

Unter der Kinematik eines Roboters versteht man die Transformationsvorschrift, die den Zusammenhang zwischen den Gelenkwinkeln q eines Roboters und der Lage des Endeffektors x_E beschreibt.

Sie wird durch die Geometrie des Roboters (z. B. Segmentlängen) bestimmt und beinhaltet zusätzlich die zeitlichen Abläufe einer Bewegung (Geschwindigkeiten und Beschleunigungen), vgl. Gleichung (3.1).

$$\text{Jointspace} \left\{ \begin{array}{l} q \Leftrightarrow x_E \\ \dot{q} \Leftrightarrow \dot{x}_E \\ \ddot{q} \Leftrightarrow \ddot{x}_E \end{array} \right\} \text{Taskspace} \quad (3.1)$$

Die Bestimmung des Zusammenhangs zwischen den Gelenkwinkeln und der Lage des Endeffektors ist Gegenstand der folgenden Abschnitte. Die Beziehungen zwischen den jeweiligen Geschwindigkeiten sowie den Beschleunigungen werden in Kapitel 4 behandelt.

3.2 Direkte Kinematik

Aus jeder Gelenkstellung resultiert genau eine bestimmte Lage des Endeffektors:

$$x_E = f(q). \quad (3.2)$$

Ziel dieses Kapitels sind Verfahren zum Aufstellen von $f(q)$. Ist $f(q)$ bekannt, kann die Lage des Endeffektors in einem Bezugskoordinatensystem, z. B. dem Basiskoordinatensystem (KS_0) ausgedrückt werden. Zu beachten ist, dass die Position des Endeffektors eindeutig festgelegt ist, wohingegen die Darstellung der Orientierung des Endeffektorkoordinatensystems von der gewählten Winkelrepräsentation (z. B. Kardan-, Euler- und RPY-Winkel) abhängt. Für den in Bild 3.2 dargestellten Roboter mit sechs Gelenken gilt:

$$q = (q_1, q_2, \dots, q_6)^T, \quad (3.3)$$

$$x_E = ((0)x_E, (0)y_E, (0)z_E, \phi_E, \psi_E, \theta_E)^T. \quad (3.4)$$

Ebenso sind in Bild 3.2 beispielhaft das Basiskoordinatensystem (KS_0) und das Endeffektorkoordinatensystem (KS_E) eingezeichnet.

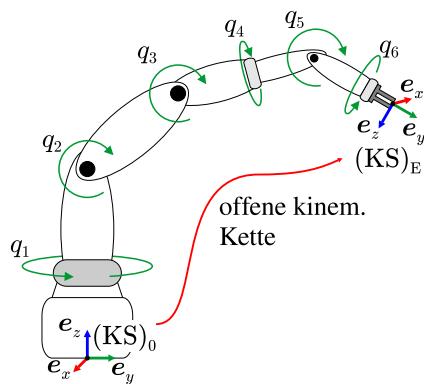


Bild 3.2: Serieller Roboter inklusive wesentlicher Koordinatensysteme

Kinematische Kette

Zur Beschreibung der kinematischen Kette eines seriellen Roboters werden gliedfeste Koordinatensysteme $(KS)_i$ eingeführt. Das Glied i wird durch $(KS)_i$, der Endeffektor durch $(KS)_E$ und die Roboterbasis durch $(KS)_0$ repräsentiert (vgl. Bild 3.3). Diese Koordinatensysteme können ortsfest, wie $(KS)_0$, oder gliedfest, wie $(KS)_i$ und $(KS)_E$ sein. Der Ansatz zur Bestimmung der Endeffektorlage aus den Gelenkwinkeln und Segmentparametern sieht eine Zerlegung der Gesamttransformation 0T_E in Teiltransformationen iT_j von der Roboterbasis hin zum Endeffektor vor.

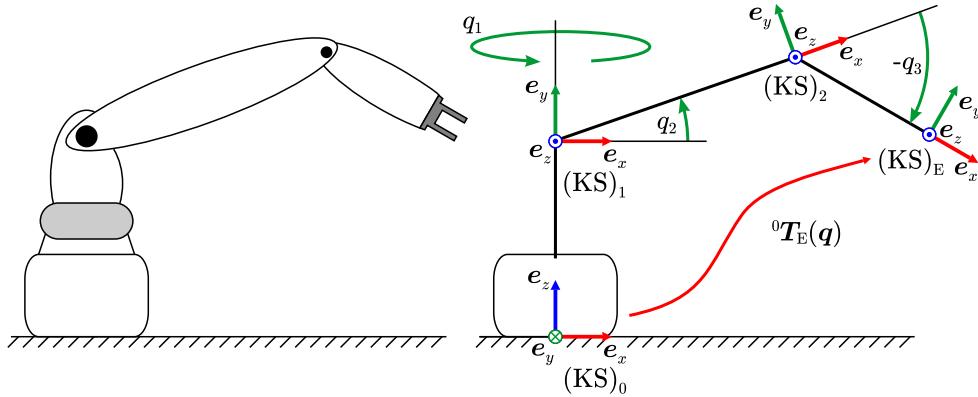


Bild 3.3: Kinematische Kette eines RRR-Roboters

Gesucht ist folgende Modellstruktur:

$${}^0T_E(q) = {}^0T_1(q_1) {}^1T_2(q_2) \cdots {}^{n-1}T_E(q_n). \quad (3.5)$$

Für einfache Roboter, wie beispielsweise den in Bild 3.4 dargestellten planaren RR-Roboter, kann

Beispiel

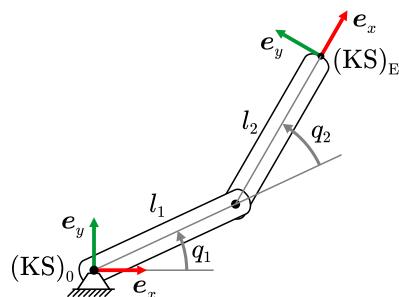
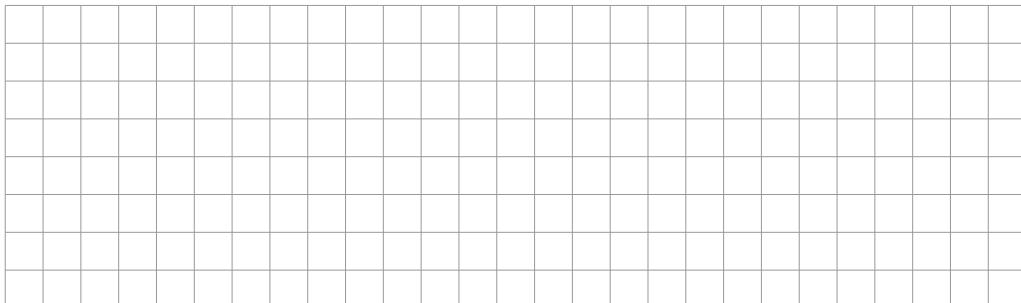


Bild 3.4: Planarer RR-Roboter

die direkte Kinematik durch einfache geometrische Betrachtungen abgelesen werden.



Dieser Ansatz gestaltet sich jedoch für Roboterstrukturen höherer Ordnung sehr komplex und fehleranfällig. Aus diesem Grund wird in Abschnitt 3.3 die Denavit-Hartenberg Notation eingeführt, welche die Bestimmung der direkten Kinematik komplexer Systeme systematisch ermöglicht.

3.3 Denavit-Hartenberg Notation

Bei der Denavit-Hartenberg Notation handelt es sich um eine spezielle Form der homogenen Transformation $i^{-1}T_i(q_i)$ zur Beschreibung der Kinematik von Robotern. Zu beachten ist, dass zwei Konventionen dieser Notation existieren. Im Folgenden wird von der Notation nach *Paul* ausgegangen. Der Grundgedanke der Denavit-Hartenberg Notation sieht vor, die Freiheitsgrade eines komplexen Systems einzeln und unabhängig von der globalen Struktur zu betrachten.

Anmerkung

Voraussetzung für das Aufstellen der Denavit-Hartenberg Matrizen ist die Kenntnis der Roboter-Nulllage (sämtliche Gelenkkoordinaten gleich null).

3.3.1 Aufstellen der Denavit-Hartenberg Matrizen

Grundlage für das Aufstellen der Denavit-Hartenberg Matrizen ist die korrekte Wahl der körperfesten Koordinatensysteme $(KS)_i$. Diese sind wie folgt vorgegeben und in Bild 3.5 dargestellt:

- Der Koordinatenursprung S_{N_i} von $(KS)_i$ liegt im Schnittpunkt der gemeinsamen Normalen von Gelenkkachse i und $i + 1$ mit der Gelenkkachse $i + 1$ und demnach auf der Gelenkkachse $i + 1$
- Die Orientierung des $(KS)_i$ stellt sich wie folgt dar:
 - z_i in Richtung der Gelenkkachse $i + 1$
 - x_i auf gemeinsamer Normalen von z_i und z_{i-1} , in Richtung von $(KS)_i$ zeigend
 - y_i durch Rechtssystem festgelegt

Anmerkung

Die Denavit-Hartenberg Notation ist nicht eindeutig, wie anhand der nachstehenden Sonderfälle noch gezeigt wird.

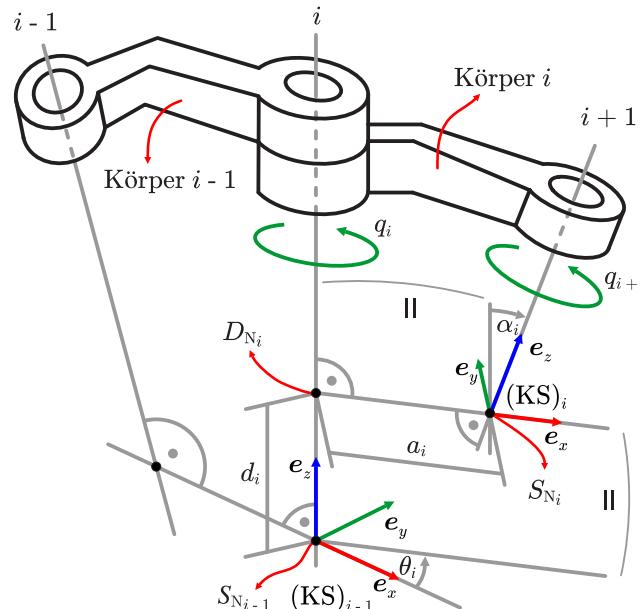


Bild 3.5: Denavit-Hartenberg Notation

Denavit-Hartenberg Parameter

Die Lage von zwei aufeinanderfolgenden Koordinatensystemen wird durch vier Denavit-Hartenberg Parameter bestimmt (siehe auch Bild 3.5):

θ_i : Drehwinkel des Gliedes i um z_{i-1} mit dem Winkel zwischen x_{i-1} und x_i (bei Drehgelenken $\theta_i = q_i$).

d_i : Strecke entlang der Normalenschnittpunkte $S_{N_{i-1}}$ und D_{N_i} (Schnittpunkt der gemeinsamen Normalen von Gelenkachse i und $i+1$ mit der Gelenkachse i), positiv in z_{i-1} -Richtung (bei Schubgelenken $d_i = q_i$).

a_i : Strecke entlang gemeinsamer Normalen der benachbarten, windschiefen Achsen z_{i-1} und z_i , positiv in x_i -Richtung.

α_i : Drehwinkel um x_i mit dem Winkel zwischen z_i und z_{i-1} .

Die Denavit-Hartenberg Parameter charakterisieren also folgende vier Schritte zur Bestimmung des mathematischen Zusammenhangs zweier Koordinatensysteme:

- Rotation von Körper i um z_{i-1} mit dem Winkel θ_i
- Translation entlang z_{i-1} mit d_i
- Translation entlang der gemeinsamen Normalen in Richtung von x_i mit a_i
- Rotation um x_i mit dem Winkel α_i

Zu beachten ist, dass die Rotation um die jeweils neuen Koordinatenachsen erfolgt. Es handelt sich somit um die Rechtsmultiplikation (vgl. Kapitel 2).

Denavit-Hartenberg Matrizen

Aus den Denavit-Hartenberg Parametern von Glied i resultiert die folgende homogene Denavit-Hartenberg Matrix \mathbf{A}_i , zusammengesetzt aus elementaren Rotationen und Translationen (mit $s_{\alpha_i} = \sin(\alpha_i)$, $c_{\alpha_i} = \cos(\alpha_i)$, $s_{\theta_i} = \sin(\theta_i)$, $c_{\theta_i} = \cos(\theta_i)$):

$$\begin{aligned}
 \mathbf{A}_i(q_i) &= {}^{i-1}\mathbf{T}_i(q_i) \\
 &= \underbrace{\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{T}_{rz}(\theta_i)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{T}_t(0, 0, d_i)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{T}_t(a_i, 0, 0)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{T}_{rx}(\alpha_i)} \\
 &= \underbrace{\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{T}_{rz}(\theta_i)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{T}_t(a_i, 0, d_i)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{T}_{rx}(\alpha_i)} \quad (3.6) \\
 &= \begin{pmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}.
 \end{aligned}$$

Damit ist die Transformationsvorschrift zwischen zwei Koordinatensystemen vollständig beschrieben. Sie resultiert aus der sequenziellen Verknüpfung der vier Teilschritte, die durch die Denavit-Hartenberg Parameter festgelegt sind.

Sonderfälle

Folgende Sonderfälle führen zu Vereinfachungen, jedoch auch zu Mehrdeutigkeiten:

1. Parallele Gelenkachsen z_{i-1} und z_i (unendlich viele Normalen):
Ursprung von $(KS)_i$ frei wählbar, sinnvoll: $d_i = 0$. In Bild 3.6 trifft dies auf q_2 und q_3 zu.
2. Gelenkachsen z_{i-1} und z_i schneiden sich (Achsen z_{i-1} und z_i nicht windschief):
Ursprung von $(KS)_i$ im Schnittpunkt ($a_i = 0$), die x_i -Achse steht senkrecht auf z_{i-1} und z_i . In Bild 3.6 gilt dies beispielsweise für q_1 und q_2 .
3. Lage von $(KS)_0$:
Achse z_0 durch erste Gelenkachse bestimmt, Ursprung von $(KS)_0$ nicht eindeutig, da kein Gelenk $i - 1$ existiert, sinnvoll: Koordinatenursprung in der Basis.
4. Lage von $(KS)_n = (KS)_E$:
Achse z_E nicht eindeutig, da kein Gelenk $n + 1$ existiert, übliche Wahl: z_E zeigt aus dem Endeffektor heraus, jedoch nicht immer möglich.

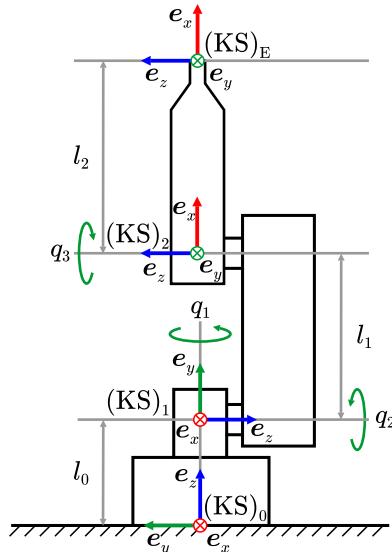


Bild 3.6: Sonderfälle der Denavit-Hartenberg Notation am Beispiel eines RRR-Roboters

In der Praxis treten konstruktionsbedingt insbesondere die Fälle 1 und 2 häufig auf. Bedingt durch die Sonderfälle können Koordinatensysteme so gewählt werden, dass sich bestimmte Denavit-Hartenberg Parameter zu null ergeben, was den Rechenaufwand zur Bestimmung der direkten Kinematik reduziert.

Beispiel

Im Folgenden werden die zur Bestimmung der Denavit-Hartenberg Parameter erforderlichen Schritte anhand des in Bild 3.7(a) gezeigten Beispiels veranschaulicht:

Bei n Gelenkachsen resultieren n generalisierte Koordinaten (q_1, \dots, q_n) , $n + 1$ Koordinatensysteme und n „Sätze“ von Denavit-Hartenberg Parametern.

Es wird davon ausgegangen, dass sich der Roboter in seiner definierten Nulllage befindet.

1. Rotations- bzw. Schubachsen bestimmen und diese jeweils von 1 bis n nummerieren (Bild 3.7(b)).
2. $(KS)_i$ auf Dreh- bzw. Schubachse $i + 1$ positionieren (es sind insbesondere die oben beschriebenen Sonderfälle zu beachten):
 - z_i zeigt in Achsrichtung $i + 1$ (Bild 3.7(c)),

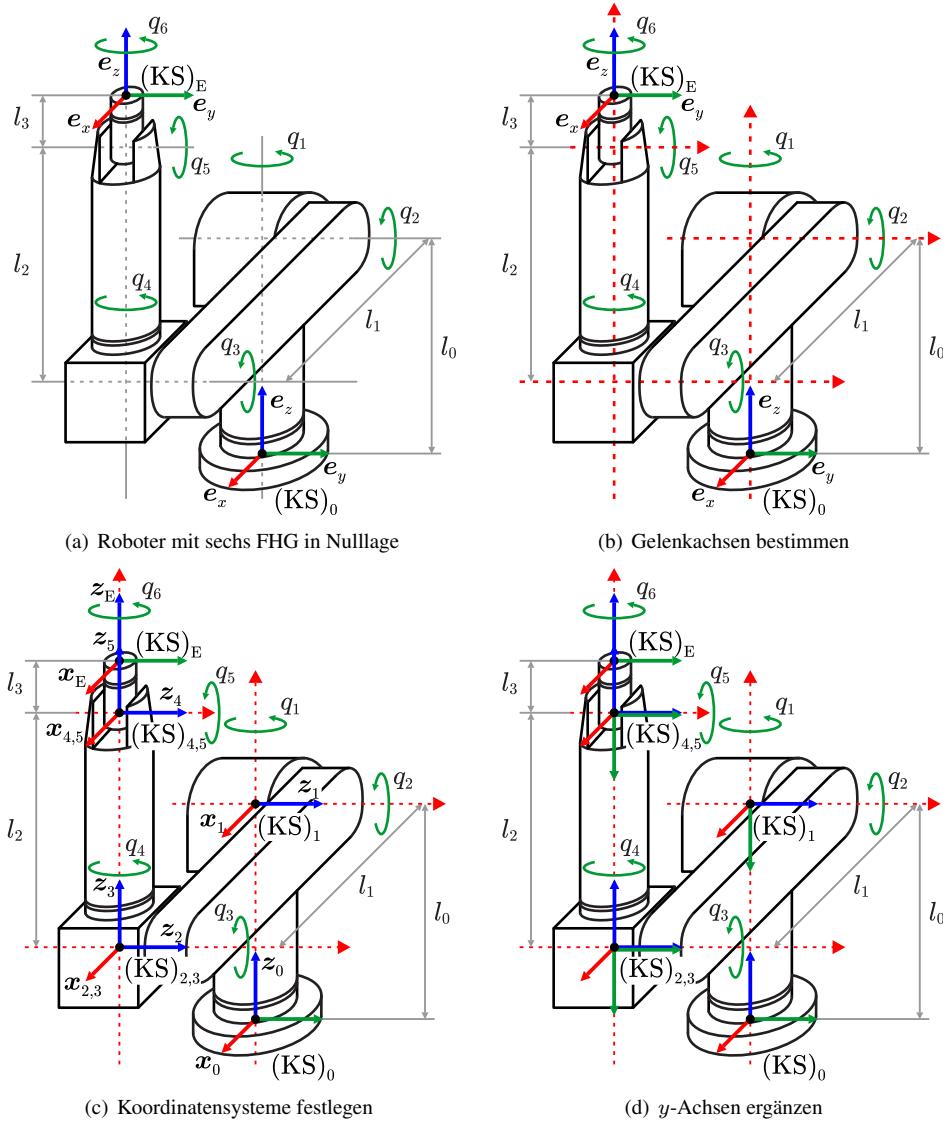


Bild 3.7: Vorgehen zur Ermittlung der Denavit-Hartenberg Parameter am Beispiel eines Roboters mit sechs Freiheitsgraden

- Normale zwischen den Achsen bestimmen und x_i in Richtung der Normalen festlegen (Bild 3.7(c)),
- Koordinatenachse y_i gemäß Rechtssystem bestimmen (Bild 3.7(d)).

Schritt 2 ist nacheinander für alle Koordinatensysteme von der Basis bis hin zum Endeffektor zu wiederholen.

- Denavit-Hartenberg Parameter ablesen. Für das Beispiel erhält man die in Tabelle 3.1 aufgeführten Werte.
- Denavit-Hartenberg Matrizen ermitteln und die Gesamttransformation berechnen. Die Berechnungsvorschrift für die Denavit-Hartenberg Matrix für das i -te Glied lautet (vgl. Gleichung (3.6)):

$$\mathbf{A}_i(q_i) = \mathbf{T}_{\text{r}_z}(\theta_i) \mathbf{T}_{\text{t}}(a_i, 0, d_i) \mathbf{T}_{\text{r}_x}(\alpha_i). \quad (3.7)$$

Durch Verknüpfung aller Matrizen A_i erhält man die vollständige Beziehung zwischen dem Basiskoordinatensystem $(KS)_0$ und dem Endeffektorkoordinatensystem $(KS)_E$:

$${}^0T_E(q) = {}^0T_1(q_1) \cdots {}^5T_E(q_6) = A_1(q_1) \cdots A_6(q_6). \quad (3.8)$$

Tabelle 3.1: Denavit-Hartenberg Parameter des gegebenen Roboters mit sechs Freiheitsgraden

i	θ_i	d_i	a_i	α_i
1	q_1	l_0	0	$-\pi/2$
2	q_2	0	l_1	0
3	q_3	0	0	$\pi/2$
4	q_4	l_2	0	$-\pi/2$
5	q_5	0	0	$\pi/2$
6	q_6	l_3	0	0

3.3.2 Bestimmung der Lage des Endeffektors

Im letzten Schritt des in Abschnitt 3.3.1 gezeigten Formalismus wurde die homogene Transformationsmatrix ${}^0T_E(q)$ durch Multiplikation der Denavit-Hartenberg Matrizen bestimmt:

$${}^0T_E(q) = {}^0T_1(q_1) \cdots {}^{n-1}T_E(q_n) = A_1(q_1) \cdots A_n(q_n) = \left(\begin{array}{c|c} {}^0R_E & {}^{(0)}r_E \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array} \right). \quad (3.9)$$

Die **Position** des Endeffektors im Basiskoordinatensystem $(KS)_0$ kann aus der homogenen Transformationsmatrix ${}^0T_E(q)$ direkt abgelesen werden:

$$\begin{aligned} {}^{(0)}x_E &= {}^0t_{E,(1,4)}, \\ {}^{(0)}y_E &= {}^0t_{E,(2,4)}, \\ {}^{(0)}z_E &= {}^0t_{E,(3,4)}, \end{aligned} \quad (3.10)$$

wobei ${}^0t_{E,(i,j)}$ das i -te Element der j -ten Spalte von 0T_E ist.

Die Bestimmung der **Orientierung** des Endeffektors ist abhängig von der gewählten Winkeleinstellung (z. B. Euler-, Kardan- oder RPY-Winkel). Die erforderlichen Gleichungen sind exemplarisch für die Kardan-Winkel-Notation gegeben. Nachstehende homogene Transformationsmatrix wurde mittels Denavit-Hartenberg Formalismus bestimmt (vgl. Gleichung (3.9)).

$${}^0T_E = \left(\begin{array}{c|c} {}^0R_E & {}^{(0)}r_E \\ \hline 0 & 0 \\ 0 & 0 \\ 1 & \end{array} \right). \quad (3.11)$$

Gleichung (3.12) ergibt sich aus der Definition der Kardan-Winkel (vgl. Kapitel 2):

$$R_E = R_x(\alpha) R_y(\beta) R_z(\gamma) = \begin{pmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -s_\alpha c_\beta \\ s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma & s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta \end{pmatrix}, \quad (3.12)$$

mit folgenden Abkürzungen: $c_\# = \cos(\#)$, $s_\# = \sin(\#)$.

Die Rotationswinkel folgen durch einen Koeffizientenvergleich der Rotationsmatrizen nach Gleichung (3.11) und Gleichung (3.12) und Auflösen nach den gesuchten Winkeln:

$$\phi_E = \alpha = \arctan2(-{}^0t_{E,(2,3)}, {}^0t_{E,(3,3)}), \quad (3.13)$$

$$\psi_E = \beta = \arctan2({}^0t_{E,(1,3)}, {}^0t_{E,(1,1)} c_\gamma - {}^0t_{E,(1,2)} s_\gamma), \quad (3.14)$$

$$\theta_E = \gamma = \arctan2(-{}^0t_{E,(1,2)}, {}^0t_{E,(1,1)}). \quad (3.15)$$

Für die anderen Winkeldarstellungen ist der Lösungsweg analog.

3.4 Inverse Kinematik

Mittels inverser Kinematik werden bei gegebener Lage des Endeffektors \mathbf{x}_E die zugehörigen Gelenkstellungen \mathbf{q} berechnet. Im Folgenden sei zunächst die inverse Kinematik für „normale“, d. h. nicht redundante Roboter ($\dim(\mathbf{q}) = \dim(\mathbf{x}_E)$) betrachtet, wobei $\dim(\mathbf{x}_E)$ die unabhängigen Endeffektorfreiheitsgrade sind. Die inverse Kinematik ist im Allgemeinen bis auf Symmetrien eindeutig lösbar (vgl. Bild 3.8). Eine wichtige Ausnahme bilden singuläre Stellungen, die in Kapitel 4 behandelt werden.

Problematisch bezüglich des Aufstellens der inversen Kinematik erweist sich, dass kein Formalismus zur Gewinnung einer Lösung vorliegt. Der Lösungsansatz muss in Abhängigkeit der jeweiligen Robotergeometrie spezifisch gewonnen werden.

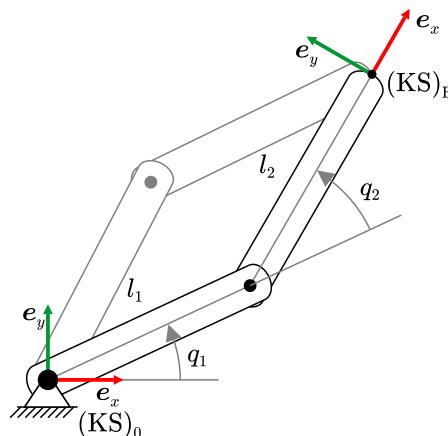


Bild 3.8: Symmetrische Konfigurationen eines planaren RR-Roboters

Bei der analytischen Berechnung der inversen Kinematik werden meist folgende Schritte durchgeführt:

- Aufstellen der direkten Kinematik (z. B. nach Formalismus aus Abschnitt 3.3),
- Gleichsetzen mit der allgemeinen homogenen Transformationsmatrix, welche die geforderte Solllage des Endeffektors beschreibt,
- Lösen des Gleichungssystems führt auf mehrere mögliche Gelenkwinkel (Roboterkonfigurationen), welche die gewünschte Solllage des Endeffektors einstellen.

Im Folgenden werden verschiedene Verfahren zur analytischen (Abschnitt 3.4.1), geometrischen (Abschnitt 3.4.2) und numerischen (Abschnitt 3.4.3) Berechnung der inversen Kinematik vorgestellt. Der planare RR-Roboter (siehe Bild 3.8) dient hierzu als Beispiel.

3.4.1 Analytische Berechnung der inversen Kinematik

Die linke Seite von Gleichung (3.16) sei durch die direkte Kinematik gegeben, während die rechte Seite die gewünschte Solllage beschreibt:

$${}^0\mathbf{T}_E = \left(\begin{array}{ccc|cc} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} & 0 \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 \end{array} \right) \stackrel{!}{=} \left(\begin{array}{ccc|cc} c_{\theta_E} & -s_{\theta_E} & 0 & {}^{(0)}x_E & 0 \\ s_{\theta_E} & c_{\theta_E} & 0 & {}^{(0)}y_E & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 \end{array} \right). \quad (3.16)$$

Die Lösung (Bestimmung der Gelenkwinkel $\mathbf{q} = (q_1, q_2)^T$) erfolgt mittels Koeffizientenvergleich (Lösungsweg siehe [Cra04, S. 122–125]). Die Betrachtung der Position des Endeffektors ergibt:

$${}^{(0)}x_E = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2), \quad (3.17)$$

$${}^{(0)}y_E = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2). \quad (3.18)$$

Werden die Koeffizienten der Rotationsmatrix verglichen, so resultieren zwei weitere Gleichungen:

$$c_{\theta_E} = \cos(q_1 + q_2), \quad (3.19)$$

$$s_{\theta_E} = \sin(q_1 + q_2). \quad (3.20)$$

Insgesamt können also vier nichtlineare Gleichungen für zwei unbekannte Gelenkwinkel aufgestellt werden. Das Gleichungssystem ist überbestimmt. Werden Gleichung (3.17) und Gleichung (3.18) zur Berechnung von \mathbf{q} herangezogen, so ist die Orientierung des Endeffektors θ_E durch ${}^{(0)}x_E$ und ${}^{(0)}y_E$ vorgegeben.

Für q_2 ergeben sich folgende Lösungen:

$$q_2 = \arctan2 \left(\pm \sqrt{1 - \left(\frac{{}^{(0)}x_E^2 + {}^{(0)}y_E^2 - l_1^2 - l_2^2}{2l_1 l_2} \right)^2}, \frac{{}^{(0)}x_E^2 + {}^{(0)}y_E^2 - l_1^2 - l_2^2}{2l_1 l_2} \right). \quad (3.21)$$

Für q_1 erhält man:

$$q_1 = \arctan2({}^{(0)}y_E, {}^{(0)}x_E) - \arctan2(l_2 s_2, l_1 + l_2 c_2). \quad (3.22)$$

Es existieren also zwei Roboterkonfigurationen (Ellenbogen hoch (+) und tief (-) in Gleichung (3.21)), mit denen die gewünschte Sollposition des Endeffektors erreicht wird.

Anmerkung

Die Funktion $\arctan2(y, x)$ wertet die Vorzeichen der Argumente aus und löst das Quadrantenproblem, das aus den Mehrdeutigkeiten bei der Definition der Funktion $\arctan(y/x)$ resultiert.

Zur Bestimmung der analytischen Lösung existieren unterschiedliche Wege. Nachstehend sei beispielhaft ein alternativer Lösungsweg zur Bestimmung der inversen Kinematik des in Bild 3.8 gezeigten Roboters gegeben. Der Ansatz ist analog zu oben. Das Aufstellen der nichtlinearen Gleichungen ergibt wie zuvor:

$${}^{(0)}x_E = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2), \quad (3.23)$$

$${}^{(0)}y_E = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2), \quad (3.24)$$

$$c_{\theta_E} = \cos(q_1 + q_2), \quad (3.25)$$

$$s_{\theta_E} = \sin(q_1 + q_2). \quad (3.26)$$

Zur Berechnung von q_2 werden die oben stehenden Gleichungen quadriert und anschließend addiert (Das Ziel ist dabei die Eliminierung von q_1):

$$(0)x_E^2 + (0)y_E^2 = l_1^2 \cos^2(q_1) + 2l_1l_2 \cos(q_1) \cos(q_1 + q_2) + l_2^2 \cos^2(q_1 + q_2) \quad (3.27)$$

$$+ l_1^2 \sin^2(q_1) + 2l_1l_2 \sin(q_1) \sin(q_1 + q_2) + l_2^2 \sin^2(q_1 + q_2) \quad (3.28)$$

$$= l_1^2 + l_2^2 + 2l_1l_2 (\cos(q_1) \cos(q_1 + q_2) + \sin(q_1) \sin(q_1 + q_2)) \quad (3.29)$$

$$= l_1^2 + l_2^2 + 2l_1l_2 \cos(q_2), \quad (3.30)$$

$$\Rightarrow q_2 = \pm \arccos \frac{(0)x_E^2 + (0)y_E^2 - l_1^2 - l_2^2}{2l_1l_2}. \quad (3.31)$$

Anmerkungen:

Anmerkung

1. Vorzeichen von Gleichung (3.31) entspricht Ellenbogenkonfiguration
2. Argument für \arccos muss betragsmäßig kleiner gleich eins sein; ist dies nicht der Fall, so liegt $(0)x_E, (0)y_E$ außerhalb des Arbeitsbereichs des Roboters.

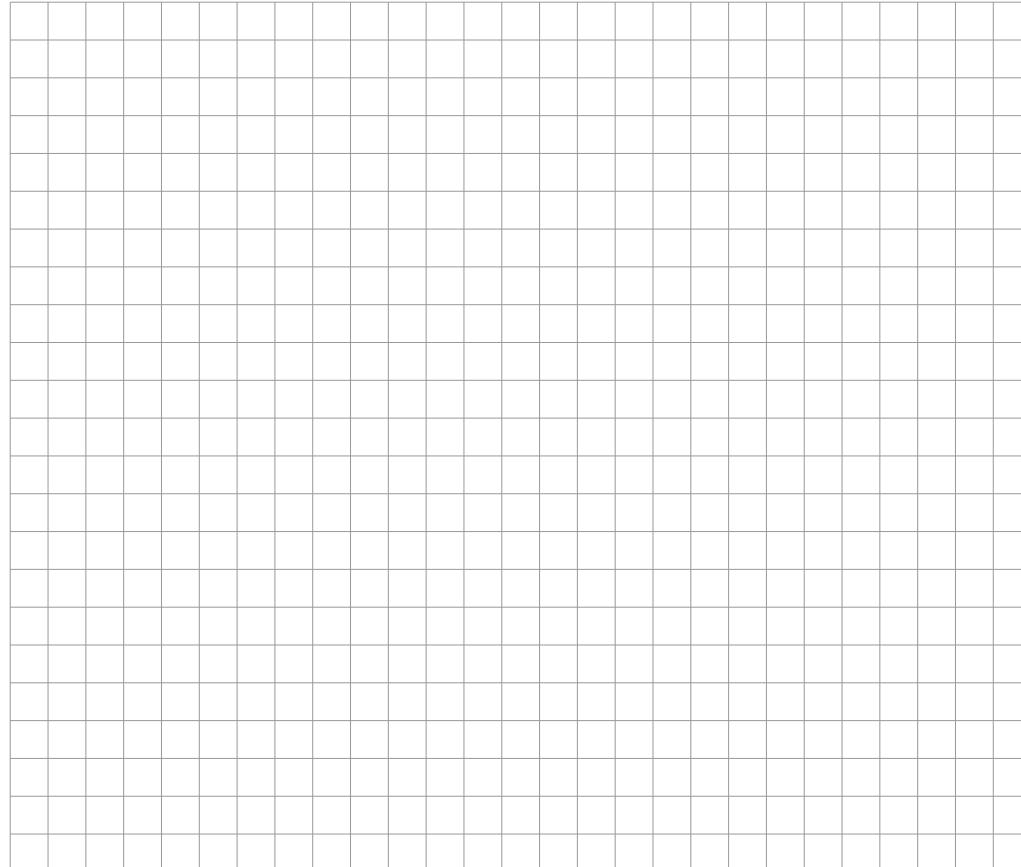
Die Anwendung von Additionstheoremen zur Berechnung von q_1 führt aus Gleichung (3.23) und aus Gleichung (3.24) zu:

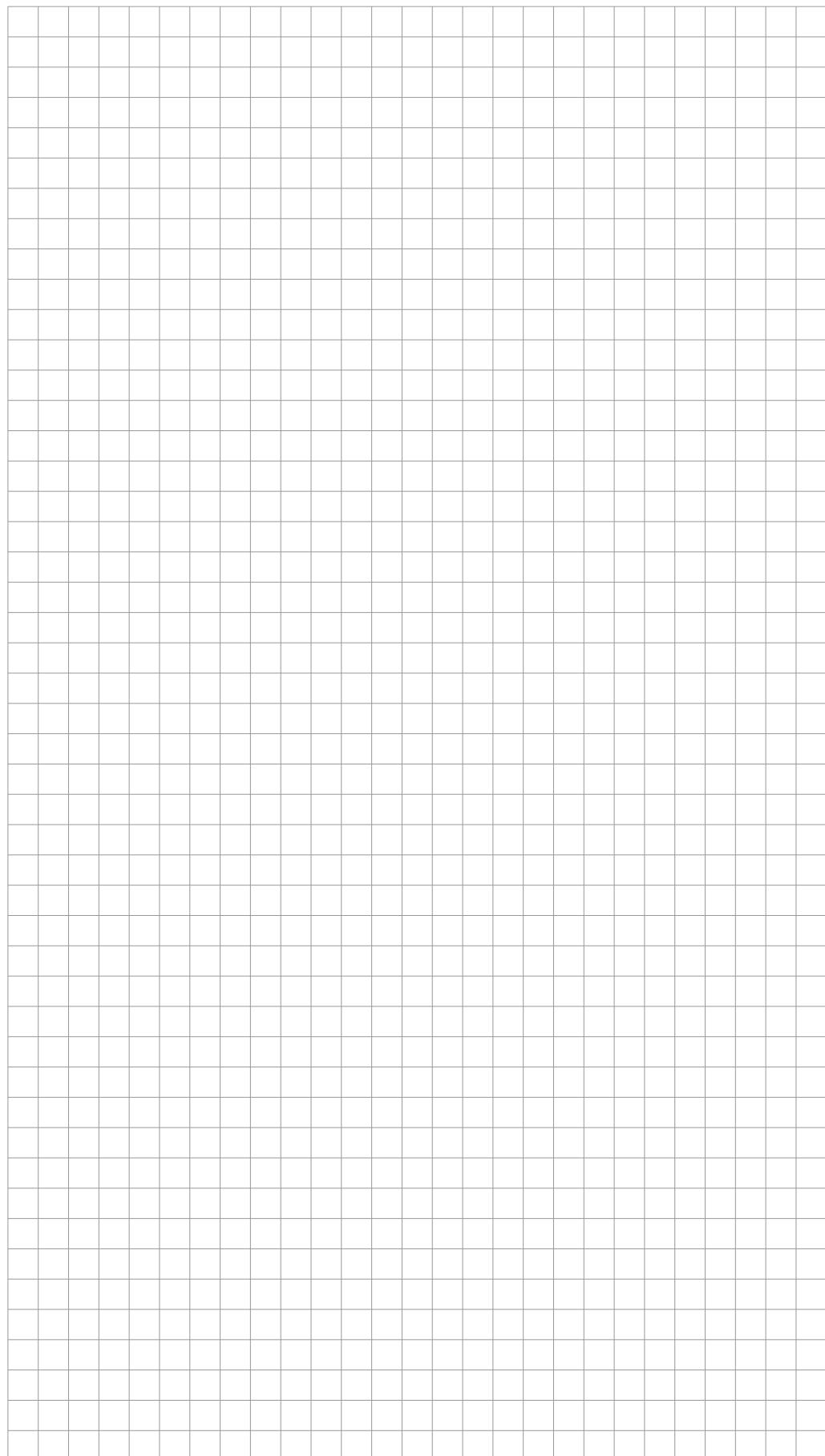
$$(0)x_E = l_1 \cos(q_1) + l_2 \cos(q_1) \cos(q_2) - l_2 \sin(q_1) \sin(q_2), \quad (3.32)$$

$$(0)y_E = l_1 \sin(q_1) + l_2 \sin(q_1) \cos(q_2) + l_2 \cos(q_1) \sin(q_2). \quad (3.33)$$

$$\Rightarrow (0)x_E = (l_1 + l_2 \cos(q_2)) \cos(q_1) - l_2 \sin(q_1) \sin(q_2), \quad (3.34)$$

$$\Rightarrow (0)y_E = (l_1 + l_2 \cos(q_2)) \sin(q_1) + l_2 \cos(q_1) \sin(q_2). \quad (3.35)$$





3.4.2 Geometrische Ermittlung der inversen Kinematik

Neben der zuvor beispielhaft geschilderten analytischen Methode können auch (elementar) geometrische Betrachtungen zur Bestimmung der inversen Kinematik herangezogen werden. Mit zunehmender Anzahl an Freiheitsgraden steigt jedoch die Komplexität dieses Ansatzes.

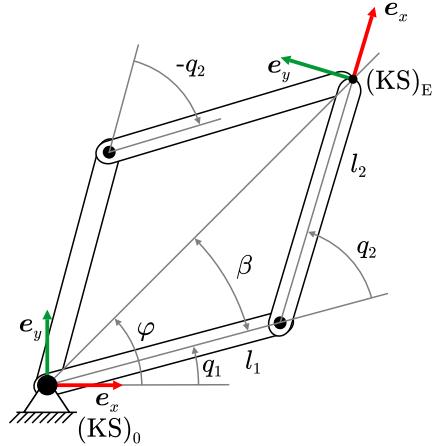


Bild 3.9: Planarer RR-Roboter

Im Folgenden sei die geometrische Ermittlung der inversen Kinematik für den in Bild 3.9 gezeigten Roboter vorgestellt (Lösungsweg siehe [Cra04, S. 126, 127]). Der zur Bestimmung von q_2 herangezogene Cosinussatz liefert:

$${}^{(0)}x_E^2 + {}^{(0)}y_E^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(\pi \pm q_2). \quad (3.36)$$

Die Vorzeichen bedeuten dabei Ellenbogen hoch (+) und tief (-). Aus Gleichung (3.36) folgt:

$$q_2 = \pm \arccos \left(\frac{{}^{(0)}x_E^2 + {}^{(0)}y_E^2 - l_1^2 - l_2^2}{2 l_1 l_2} \right). \quad (3.37)$$

Für den Winkel φ erhält man:

$$\varphi = \arctan2({}^{(0)}y_E, {}^{(0)}x_E). \quad (3.38)$$

Analog zu q_2 erhält man für β :

$$\beta = \arccos \left(\frac{{}^{(0)}x_E^2 + {}^{(0)}y_E^2 + l_1^2 - l_2^2}{2 l_1 \sqrt{{}^{(0)}x_E^2 + {}^{(0)}y_E^2}} \right). \quad (3.39)$$

Daraus ergibt sich für den Gelenkwinkel:

$$q_1 = \varphi \pm \beta. \quad (3.40)$$

3.4.3 Numerische Berechnung der inversen Kinematik

Falls eine analytische Lösung der inversen Kinematik nicht möglich ist, können die gesuchten Gelenkwinkel auch numerisch bestimmt werden. Hierzu wird die direkte Kinematik $\mathbf{x}_E = \mathbf{f}(\mathbf{q})$ umgestellt zu $\mathbf{f}(\mathbf{q}) - \mathbf{x}_E = \mathbf{0}$, so dass die Lösung (Berechnung von \mathbf{q}) einer Nullstellensuche entspricht. Diese kann nun iterativ mittels Newton-Raphson-Verfahren berechnet werden. Grundidee ist die Linearisierung der Funktion in einem Startpunkt und die Verwendung der Nullstelle

dieser Tangente als verbesserte Näherung. Die erhaltene Näherung ist anschließend Startpunkt für einen weiteren Iterationsschritt. Bei hinreichend kleiner Verbesserung wird die Berechnung abgebrochen. Das Verfahren ist in Bild 3.10 skizziert.

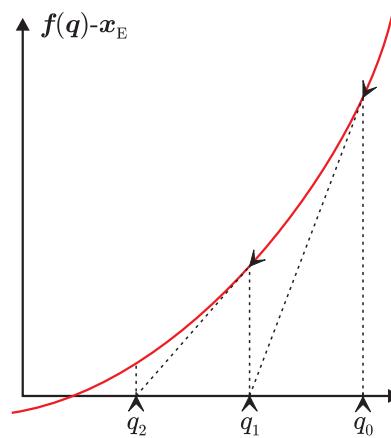


Bild 3.10: Schematische Darstellung des Newton-Raphson-Verfahrens

Dieser Ansatz beinhaltet folgende Nachteile:

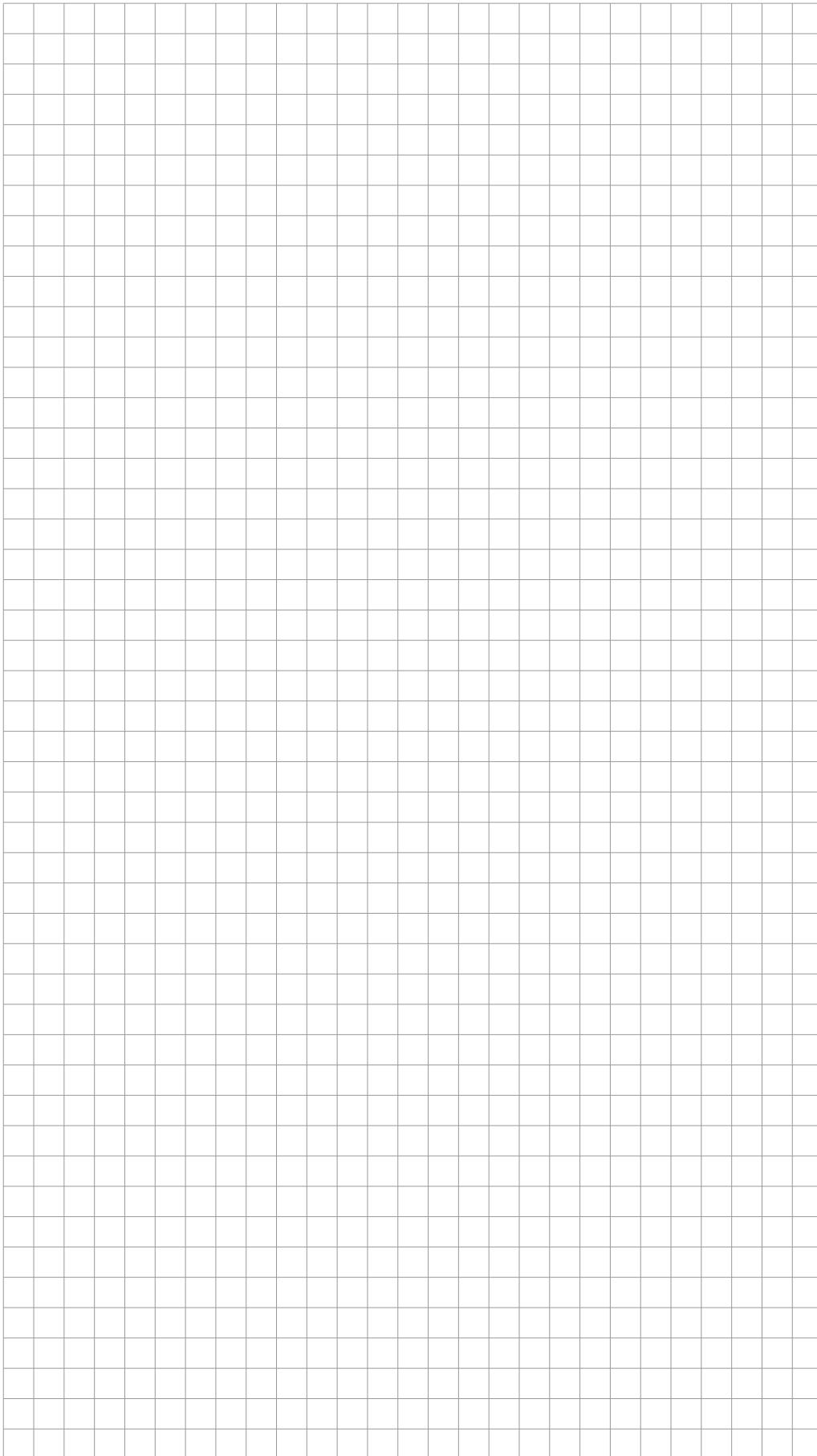
- aufwendiger und langsamer Algorithmus (insbes. für schnelle Regelungen ungeeignet),
 - Konvergenz nicht garantiert.

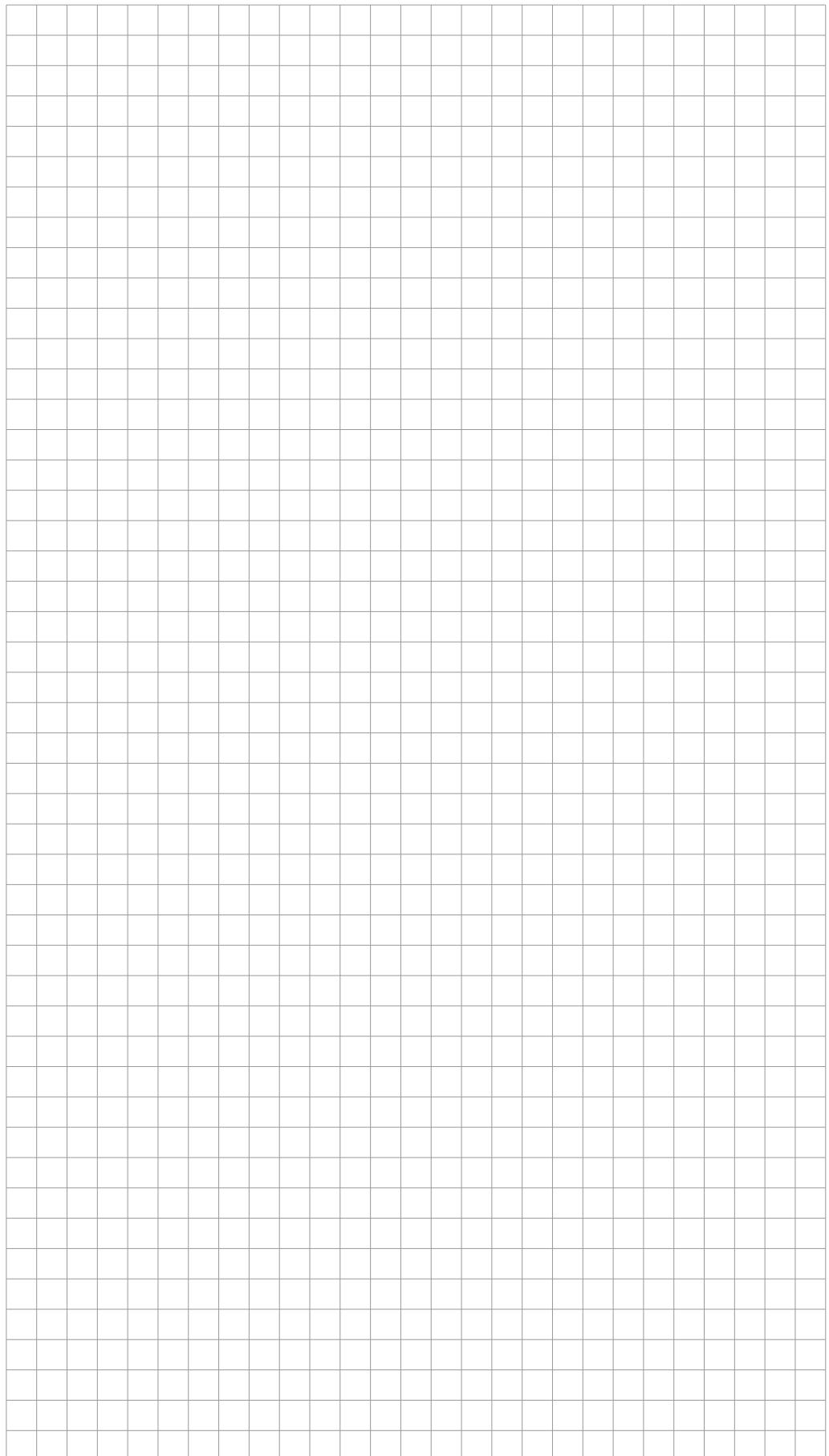
Problematisch bei iterativen Verfahren sind auch lokale Minima, in denen der Algorithmus „hängenbleiben“ kann, sowie die in Abhangigkeit des Startwerts stark schwankende Anzahl an Iterationsschritten fur eine Losung. Ein guter Startwert ist mageblich fur eine schnelle Konvergenz des Verfahrens.

Fragen zur Selbstkontrolle

1. Bestimmen Sie die direkte Kinematik eines planaren RPR-Roboters (zwei Rotationsfreiheitsgrade und ein Linearantrieb) in Abhangigkeit der Gelenkkoordinaten q_1 , q_2 und q_3 ! Verwenden Sie hierfur den „intuitiven“ Ansatz und die Denavit-Hartenberg Notation!
 2. Welche Sonderfalle konnen bei der Verwendung der Denavit-Hartenberg Notation auftreten und wie wahlen Sie in diesen Fallen die Lage der Koordinatensysteme?
 3. Bestimmen Sie die inverse Kinematik eines planaren RP-Roboters (ein Rotations- und ein Linearantrieb) in Abhangigkeit der Endeffektorposition $(0)x_E$ und $(0)y_E$!







4 Jacobi-Matrix – Geschwindigkeiten und Kräfte

Die Jacobi-Matrix ist ein wichtiges Element für die Analyse und Auslegung von Mechanismen. Neben der Vorgehensweise zur Berechnung ist ihre Anwendung Gegenstand der nachfolgenden Ausführungen. Dieses Kapitel beschreibt, mit welchen Mitteln die Endeffektor- und Gelenkwinkelgeschwindigkeiten eines Roboters in Beziehung gesetzt werden können. Die dazu erforderliche Jacobi-Matrix wird definiert und es wird erläutert, wie diese analytisch (Abschnitt 4.2) oder geometrisch (Abschnitt 4.3) zu bestimmen ist. Die Manipulierbarkeit als ein zentrales, auf der Jacobi-Matrix basierendes, Leistungsmerkmal von (Industrie-)Robotern ist Gegenstand von Abschnitt 4.4. Weiterhin wird charakterisiert, wie sich Kräfte und Momente, die auf den Endeffektor eines Roboters wirken, auf die Gelenkmomente und -kräfte abbilden und wie dieser Zusammenhang mathematisch durch die Jacobi-Matrix beschrieben ist (Abschnitt 4.5). Im Rahmen dessen wird auch auf das Vorgehen bei der Steifigkeitsanalyse eines Roboters eingegangen.

Darauf aufbauend lassen sich die singulären Konfigurationen eines Roboters berechnen (Abschnitt 4.6). Es wird beschrieben, welche Auswirkungen Singularitäten auf Gelenkwinkelgeschwindigkeiten und -momente haben und warum sie bei der Bahnplanung zu vermeiden sind.

Als weiterführende Literatur wird empfohlen:

- B. Heimann, W. Gerth und K. Popp. *Mechatronik: Komponenten – Methoden – Beispiele*. München, Deutschland: Carl Hanser Verlag, 2006, S. 190–199
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 152–180
- L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Berlin, Deutschland: Springer, 2000, S. 69–117

4.1 Differentielle Kinematik

Eine häufig im Rahmen der Projektierung von Robotern gestellte Aufgabe ist die Analyse des Zusammenhangs zwischen Endeffektor- und Gelenkwinkelgeschwindigkeiten. Dieser wird mittels der differentiellen Kinematik ausgedrückt:

Durch die differentielle Kinematik wird die Beziehung zwischen den Gelenkwinkelgeschwindigkeiten und -beschleunigungen \dot{q}, \ddot{q} eines Roboters mit der korrespondierenden Geschwindigkeit und Beschleunigung \dot{x}_E, \ddot{x}_E des Endeffektors beschrieben.

Definition
differentielle
Kinematik

Obige Definition schließt einen linearen Zusammenhang zwischen kleinen Veränderungen von Endeffektor- und Gelenkkoordinaten $\Delta x_E, \Delta q$ ein, d. h. für kleine Δt gilt:

$$\Delta x_E \approx \dot{x}_E \Delta t \quad \text{und} \quad \Delta q \approx \dot{q} \Delta t. \quad (4.1)$$

Aus der direkten Kinematik

$$x_E = f(q) \quad (4.2)$$

kann die arbeitspunktabhängige vektorwertige Funktion zwischen inkrementellen Veränderungen von Endeffektor- und Gelenkkoordinaten ermittelt werden:

$$\Delta \boldsymbol{x}_E = \boldsymbol{J}(\boldsymbol{q}) \Delta \boldsymbol{q}, \quad (4.3)$$

bzw.

$$\Delta \boldsymbol{q} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \Delta \boldsymbol{x}_E. \quad (4.4)$$

Bei der Berechnung von Gleichung (4.4) ist auf die Invertierbarkeit von $\boldsymbol{J}(\boldsymbol{q})$ zu achten. Die arbeitspunktabhängige Jacobi-Matrix $\boldsymbol{J}(\boldsymbol{q})$ setzt die Endeffektor- und Gelenkwinkelgeschwindigkeiten linear zueinander in Beziehung (vgl. unten).

Anwendungsmöglichkeiten der differentiellen Kinematik

Beispiel
inkrementelle
Bahnplanung

Die Jacobi-Matrix kann zur inkrementellen Bahnberechnung herangezogen werden. Aus der bereits bekannten Beziehung

$$\Delta \boldsymbol{q} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \Delta \boldsymbol{x}_E \quad (4.5)$$

erhält man

$$\begin{aligned} \boldsymbol{q}_{i+1} &= \boldsymbol{q}_i + \Delta \boldsymbol{q}_i \\ &= \boldsymbol{q}_i + \boldsymbol{J}^{-1}(\boldsymbol{q}_i) \Delta \boldsymbol{x}_{E,i} \\ &= \boldsymbol{q}_i + \boldsymbol{J}^{-1}(\boldsymbol{q}_i) (\boldsymbol{x}_{E,i+1} - \boldsymbol{x}_{E,i}), \end{aligned} \quad (4.6)$$

wobei der Index i den diskrete Zeitpunkt von \boldsymbol{q} kennzeichnet. Dies vermeidet eine wiederholte, aufwendige Bestimmung der inversen Kinematik. Mehr dazu findet sich zum Thema Bahnplanung in Kapitel 6.

Beispiel
Geschwindigkeiten &
Beschleunigungen

Aus den differentiellen Beziehungen

$$\Delta \boldsymbol{x}_E = \boldsymbol{J}(\boldsymbol{q}) \Delta \boldsymbol{q} \Leftrightarrow \Delta \boldsymbol{q} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \Delta \boldsymbol{x}_E \quad (4.7)$$

lässt sich die Relation zwischen der Endeffektorgeschwindigkeit $\dot{\boldsymbol{x}}_E \approx \Delta \boldsymbol{x}_E / \Delta t$ und den korrespondierenden Geschwindigkeiten $\dot{\boldsymbol{q}} \approx \Delta \boldsymbol{q} / \Delta t$ angeben:

$$\dot{\boldsymbol{x}}_E = \boldsymbol{J}(\boldsymbol{q}) \dot{\boldsymbol{q}} \Leftrightarrow \dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \dot{\boldsymbol{x}}_E. \quad (4.8)$$

Gleiches gilt für die Beschleunigungen:

$$\ddot{\boldsymbol{x}}_E = \boldsymbol{J}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} \Leftrightarrow \ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \ddot{\boldsymbol{x}}_E - \boldsymbol{J}^{-1}(\boldsymbol{q}) \dot{\boldsymbol{J}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \boldsymbol{J}^{-1}(\boldsymbol{q}) \dot{\boldsymbol{x}}_E. \quad (4.9)$$

Zur Berechnung von $\dot{\boldsymbol{J}}$ wird die Jacobi-Matrix elementweise nach der Zeit abgeleitet. Auf diese Weise wird aus dem Zusammenhang von Endeffektor- und Gelenkwinkelgeschwindigkeiten die Beziehung von Endeffektor- und Gelenkwinkelbeschleunigungen gewonnen.

Es existieren unterschiedliche Ansätze zur Berechnung der Jacobi-Matrix: die in Abschnitt 4.2 beschriebene analytische Jacobi-Matrix und die in Abschnitt 4.3 eingeführte geometrische Jacobi-Matrix.

4.2 Analytische Jacobi-Matrix

Ist die direkte Kinematik $\mathbf{x}_E = \mathbf{f}(\mathbf{q})$ gegeben, können die entsprechenden differentiellen Zusammenhänge durch die analytische Jacobi-Matrix $\mathbf{J}(\mathbf{q})$ beschrieben werden:

$$\frac{d\mathbf{x}_E}{dt} = \frac{d\mathbf{f}(\mathbf{q})}{dt} = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}. \quad (4.10)$$

Man erhält also folgende Zusammenhänge:

$$\dot{\mathbf{x}}_E = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (4.11)$$

bzw.

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{x}}_E, \quad (4.12)$$

wobei auf die Invertierbarkeit von \mathbf{J} zu achten ist. Die analytische Jacobi-Matrix ist dabei die Ableitung der vektoriellen Funktion $\mathbf{f}(\mathbf{q})$ bezüglich sämtlicher Gelenkvariablen:

$$\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \dots & \frac{\partial f_m}{\partial q_n} \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad (4.13)$$

mit $m = \dim(\mathbf{x}_E)$ und $n = \dim(\mathbf{q})$.

Planarer RR-Roboter mit zwei Freiheitsgraden

Die Anzahl der Gelenkwinkel des in Bild 4.1 dargestellten Roboters ist gleich der Anzahl der

Beispiel

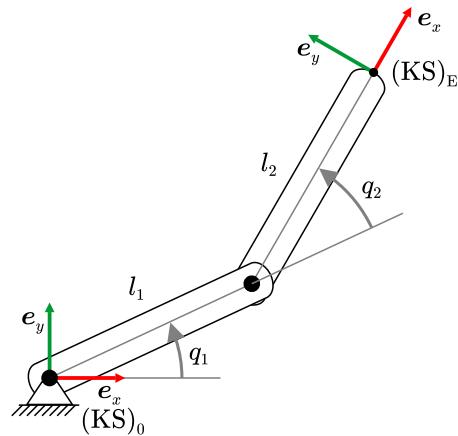


Bild 4.1: Planarer RR-Roboter mit zwei Freiheitsgraden

unabhängigen Endeffektorfreiheitsgrade:

$$\dim(\mathbf{x}_E) = \dim(\mathbf{q}) = 2. \quad (4.14)$$

Die direkte Kinematik lautet (vgl. Abschnitt 3.2):

$$(0)x_E = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2), \quad (4.15)$$

$$(0)y_E = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2). \quad (4.16)$$

**Anmerkung**

Winkel und damit Winkelgeschwindigkeiten des Endeffektors sind je nach gewählter Repräsentation nicht im $(KS)_0$ gegeben (abhängig von zusammengesetzten Drehungen).

4.3 Geometrische Jacobi-Matrix

Die Komponenten der Jacobi-Matrix können auch geometrisch bestimmt werden. Es handelt sich um die Beschreibung des vektoriellen Zusammenhangs zwischen den Gelenkwinkelgeschwindigkeiten \dot{q} und den translatorischen Endeffektorgeschwindigkeiten $\dot{x}_{E,t} = (\dot{x}_E, \dot{y}_E, \dot{z}_E)^T$ sowie den

Winkelgeschwindigkeiten $\dot{\boldsymbol{x}}_{E,r} = (\omega_x, \omega_y, \omega_z)^T$ des Endeffektors im Basiskoordinatensystem $(KS)_0$:

$${}^{(0)}\dot{\boldsymbol{x}}_{E,g} = \begin{pmatrix} {}^{(0)}\dot{\boldsymbol{x}}_{E,t} \\ {}^{(0)}\dot{\boldsymbol{x}}_{E,r} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{j}_{t_1} & \mathbf{j}_{t_2} & \cdots & \mathbf{j}_{t_n} \\ \mathbf{j}_{r_1} & \mathbf{j}_{r_2} & \cdots & \mathbf{j}_{r_n} \end{pmatrix}}_{\mathbf{J}_g(\boldsymbol{q})} \dot{\boldsymbol{q}}, \quad (4.17)$$

mit $\mathbf{j}_{t_i} \in \mathbb{R}^3$ und $\mathbf{j}_{r_i} \in \mathbb{R}^3$. Der Index „g“ steht für „geometrisch“.

Bei der Wahl der Koordinatensysteme nach Denavit-Hartenberg ergibt sich für

- Schubgelenke:

$$\begin{pmatrix} \mathbf{j}_{t_i} \\ \mathbf{j}_{r_i} \end{pmatrix} = \begin{pmatrix} {}^{(0)}\mathbf{e}_z^{(i-1)} \\ \mathbf{0} \end{pmatrix}, \quad (4.18)$$

- Drehgelenke:

$$\begin{pmatrix} \mathbf{j}_{t_i} \\ \mathbf{j}_{r_i} \end{pmatrix} = \begin{pmatrix} {}^{(0)}\mathbf{e}_z^{(i-1)} \times {}^{(0)}\mathbf{r}_{(i-1)E} \\ {}^{(0)}\mathbf{e}_z^{(i-1)} \end{pmatrix}. \quad (4.19)$$

Zur Notation sei auf Bild 4.2 verwiesen. Es werden also für jedes Gelenk (in Abhängigkeit des Gelenktyps) die Auswirkungen auf den Endeffektor analysiert.

Elemente der geometrischen Jacobi-Matrix

Die Elemente sind beispielsweise über bekannte Transformationen bestimmbar. Für die Rotationsachse oder Translationsrichtung in $(KS)_0$ dargestellt gilt:

$${}^{(0)}\mathbf{e}_z^{(i-1)} = \underbrace{{}^0\mathbf{R}_1^{-1} {}^1\mathbf{R}_2 \dots {}^{i-2}\mathbf{R}_{i-1}}_{{}^0\mathbf{R}_{i-1}} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.20)$$

und für den Richtungsvektor vom i -ten Gelenk zum Endeffektor, ebenfalls in $(KS)_0$ ausgedrückt:

$$\begin{pmatrix} {}^{(0)}\mathbf{r}_{(i-1)E} \\ 0 \end{pmatrix} = \underbrace{{}^0\mathbf{T}_1^{-1} {}^1\mathbf{T}_2 \dots {}^{n-1}\mathbf{T}_E}_{{}^0\mathbf{T}_E} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} - \underbrace{{}^0\mathbf{T}_1^{-1} {}^1\mathbf{T}_2 \dots {}^{i-2}\mathbf{T}_{i-1}}_{{}^0\mathbf{T}_{i-1}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.21)$$

Voraussetzung ist, dass die direkte Kinematik nach Denavit-Hartenberg aufgestellt wurde.

Zu beachten ist außerdem, dass die geometrische Jacobi-Matrix bezüglich der Rotation im Allgemeinen nicht der analytischen Jacobi-Matrix entspricht

$${}^{(0)}\dot{\boldsymbol{x}}_{E,g} = ({}^{(0)}\dot{\boldsymbol{x}}_{E,t}^T, {}^{(0)}\dot{\boldsymbol{x}}_{E,r}^T)^T \neq \dot{\boldsymbol{x}}_E. \quad (4.22)$$

Letztere ist von der gewählten Winkelrepräsentation abhängig, während die geometrische Jacobi-Matrix bezüglich des Basiskoordinatensystems $(KS)_0$ definiert ist. Dennoch existiert der im Folgenden beispielhaft für die Kardan-Konvention beschriebene Zusammenhang.

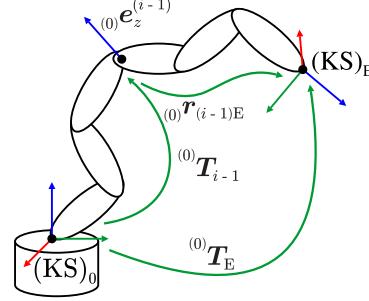


Bild 4.2: Vektoren und Transformationen zur Bestimmung der geometrischen Jacobi-Matrix

Zusammenhang zwischen analytischer und geometrischer Jacobi-Matrix

Die Berechnung von Winkelgeschwindigkeiten $\dot{\boldsymbol{x}}_{E,r} = (\omega_x, \omega_y, \omega_z)^T$ ist abhängig von den gewählten zusammengesetzten Drehungen (Beispiel Kardan-Winkel):

$$\begin{aligned} {}^{(0)}\dot{\boldsymbol{x}}_{E,r} &= \underbrace{\begin{pmatrix} \dot{\phi}_E \\ 0 \\ 0 \end{pmatrix}}_{\text{KARD}} + \mathbf{R}_x(\phi_E) \underbrace{\begin{pmatrix} 0 \\ \dot{\psi}_E \\ 0 \end{pmatrix}}_{\text{KARD}} + \mathbf{R}_x(\phi_E) \mathbf{R}_y(\psi_E) \underbrace{\begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_E \end{pmatrix}}_{\text{KARD}} \\ &= \underbrace{\begin{pmatrix} 1 & 0 & \sin(\psi_E) \\ 0 & \cos(\phi_E) & -\sin(\phi_E) \cos(\psi_E) \\ 0 & \sin(\phi_E) & \cos(\phi_E) \cos(\psi_E) \end{pmatrix}}_{\Omega_{\text{KARD}}} \underbrace{\begin{pmatrix} \dot{\phi}_E \\ \dot{\psi}_E \\ \dot{\theta}_E \end{pmatrix}}_{\text{KARD}}. \end{aligned} \quad (4.23)$$

Für den Fall, dass die zusammengesetzten Drehungen nach Kardan beschrieben werden, lässt sich folgender Zusammenhang zwischen analytischer und geometrischer Jacobi-Matrix aufstellen:

$${}^{(0)}\dot{\boldsymbol{x}}_{E,g} = \begin{pmatrix} E & \mathbf{0} \\ \mathbf{0} & \Omega_{\text{KARD}} \end{pmatrix} \dot{\boldsymbol{x}}_E \Rightarrow \mathbf{J}_g(\boldsymbol{q}) \dot{\boldsymbol{q}} = \begin{pmatrix} E & \mathbf{0} \\ \mathbf{0} & \Omega_{\text{KARD}} \end{pmatrix} \mathbf{J}(\boldsymbol{q}) \dot{\boldsymbol{q}} \quad (4.24)$$

$$\Rightarrow \mathbf{J}_g(\boldsymbol{q}) = \begin{pmatrix} E & \mathbf{0} \\ \mathbf{0} & \Omega_{\text{KARD}} \end{pmatrix} \mathbf{J}(\boldsymbol{q}). \quad (4.25)$$

Beispiel

Anhand des in Bild 4.1 gezeigten Roboters wird das Vorgehen beim Aufstellen der geometrischen Jacobi-Matrix beispielhaft gezeigt.

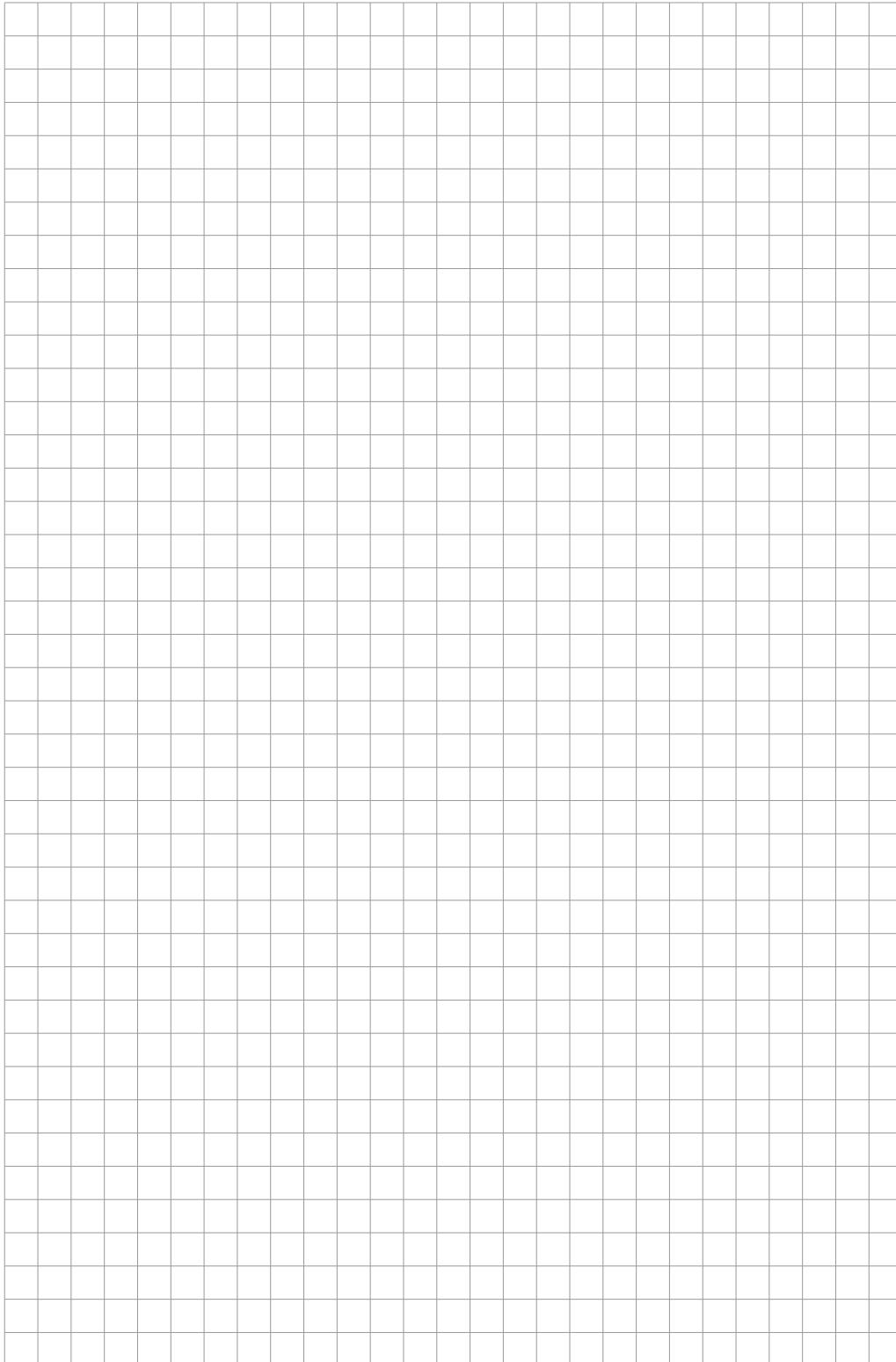
Die homogenen Transformationsmatrizen sind in Kapitel 3 unter Anwendung des Denavit-Hartenberg Formalismus bestimmt worden und lauten:

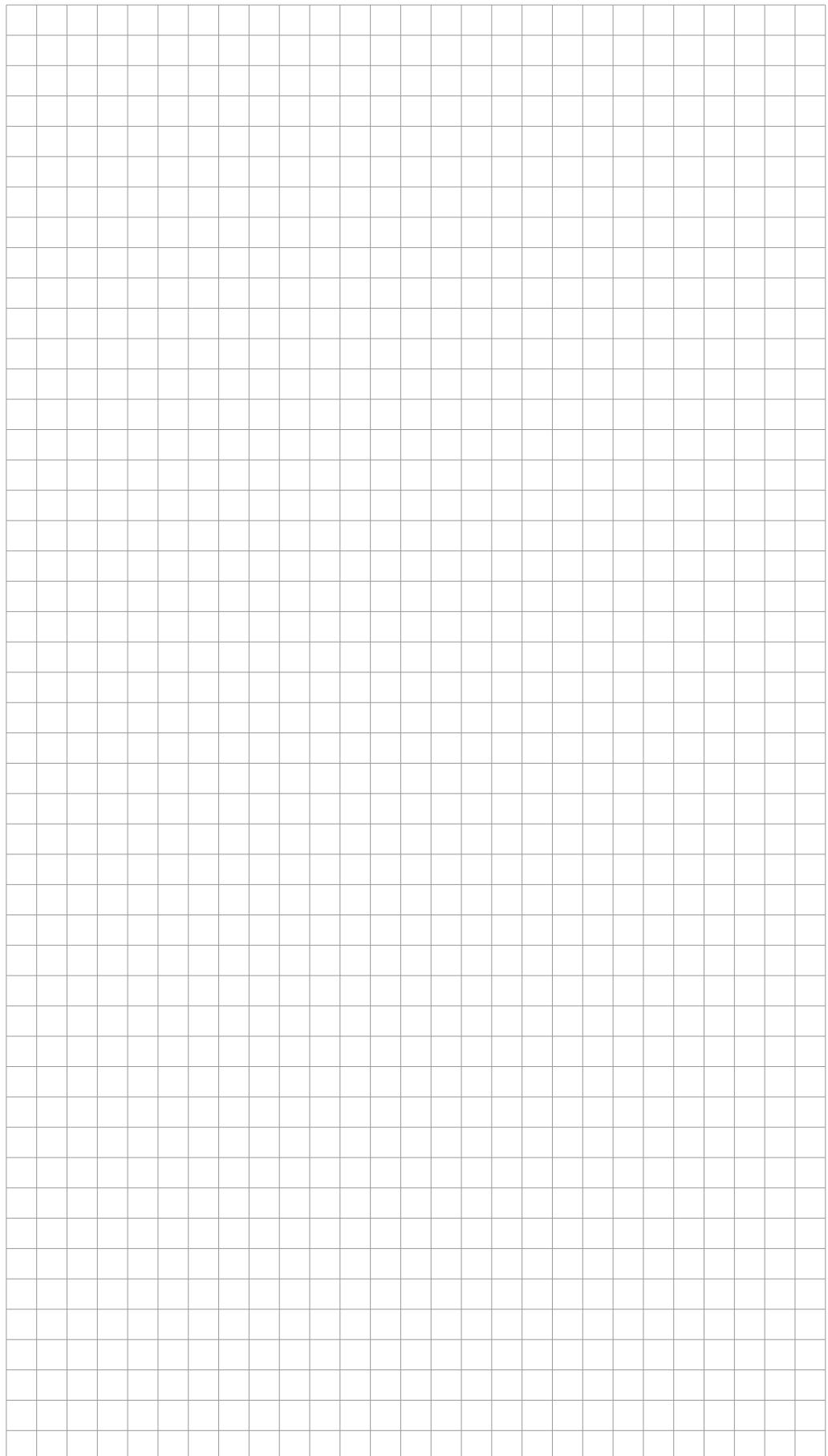
$${}^0\mathbf{T}_1(q_1) = \begin{pmatrix} \cos(q_1) & -\sin(q_1) & 0 & l_1 \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & l_1 \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (4.26)$$

$${}^1\mathbf{T}_E(q_2) = \begin{pmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_2 \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & l_2 \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4.27)$$

Die Rotationsmatrizen lassen sich aus Gleichung (4.26) und Gleichung (4.27) ablesen:

$${}^0\mathbf{R}_1 = \begin{pmatrix} \cos(q_1) & -\sin(q_1) & 0 \\ \sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad {}^1\mathbf{R}_2 = \begin{pmatrix} \cos(q_2) & -\sin(q_2) & 0 \\ \sin(q_2) & \cos(q_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.28)$$





Gelenk $i = 2$:

Die Rotationsachse des zweiten Gelenks ausgedrückt in $(KS)_0$ lautet:

$${}_{(0)}\boldsymbol{e}_z^{(1)} = {}^0\boldsymbol{R}_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(q_1) & -\sin(q_1) & 0 \\ \sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.29)$$

und der Richtungsvektor zum Endeffektor:

$$\begin{pmatrix} {}_{(0)}\boldsymbol{r}_{1E} \\ 0 \end{pmatrix} = {}^0\boldsymbol{T}_E \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} - {}^0\boldsymbol{T}_1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.30)$$

$$= \begin{pmatrix} l_2 \cos(q_1 + q_2) + l_1 \cos(q_1) \\ l_2 \sin(q_1 + q_2) + l_1 \sin(q_1) \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} l_1 \cos(q_1) \\ l_1 \sin(q_1) \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} l_2 \cos(q_1 + q_2) \\ l_2 \sin(q_1 + q_2) \\ 0 \\ 0 \end{pmatrix}. \quad (4.31)$$

Die zweite Spalte der geometrischen Jacobi-Matrix resultiert zu:

$$\begin{pmatrix} \boldsymbol{j}_{t_2} \\ \boldsymbol{j}_{r_2} \end{pmatrix} = \begin{pmatrix} {}_{(0)}\boldsymbol{e}_z^{(1)} \times {}_{(0)}\boldsymbol{r}_{1E} \\ {}_{(0)}\boldsymbol{e}_z^{(1)} \end{pmatrix} \quad (4.32)$$

$$= \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} l_2 \cos(q_1 + q_2) \\ l_2 \sin(q_1 + q_2) \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} -l_2 \sin(q_1 + q_2) \\ l_2 \cos(q_1 + q_2) \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad (4.33)$$

Die vollständige Jacobi-Matrix für den Roboter nach Bild 4.1 lautet:

$$\boldsymbol{J}_g = \begin{pmatrix} \boldsymbol{j}_{t_1} & \boldsymbol{j}_{t_2} \\ \boldsymbol{j}_{r_1} & \boldsymbol{j}_{r_2} \end{pmatrix} = \begin{pmatrix} -l_2 \sin(q_1 + q_2) - l_1 \sin(q_1) & -l_2 \sin(q_1 + q_2) \\ l_2 \cos(q_1 + q_2) + l_1 \cos(q_1) & l_2 \cos(q_1 + q_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}. \quad (4.34)$$

Zusammenfassend ist festzuhalten, dass zwei Möglichkeiten der Berechnung der Jacobi-Matrix gegeben sind:

- Die analytische Jacobi-Matrix lässt sich durch Differenzieren unmittelbar aus der direkten Kinematik gewinnen. Zu beachten ist bei diesem Vorgehen, dass die resultierende Matrix von der gewählten Winkelrepräsentation abhängt.
- Das Verfahren zur Bestimmung der geometrischen Jacobi-Matrix stützt sich auf die Denavit-Hartenberg-Notation. Es wird ausgenutzt, dass die bei der Berechnung verwendeten Rotations- bzw. Translationsachsen unmittelbar mit den Gelenkwinkeln des Roboters korrespondieren. In Folge dessen können die Elemente der geometrischen Jacobi-Matrix mit vergleichsweise geringem Aufwand unter Verwendung bekannter homogener Transformationen bestimmt werden. Des Weiteren beziehen sich die resultierenden translatorischen

Geschwindigkeiten und Winkelgeschwindigkeiten des Endeffektors explizit auf die Roboterbasis.

4.4 Manipulierbarkeit

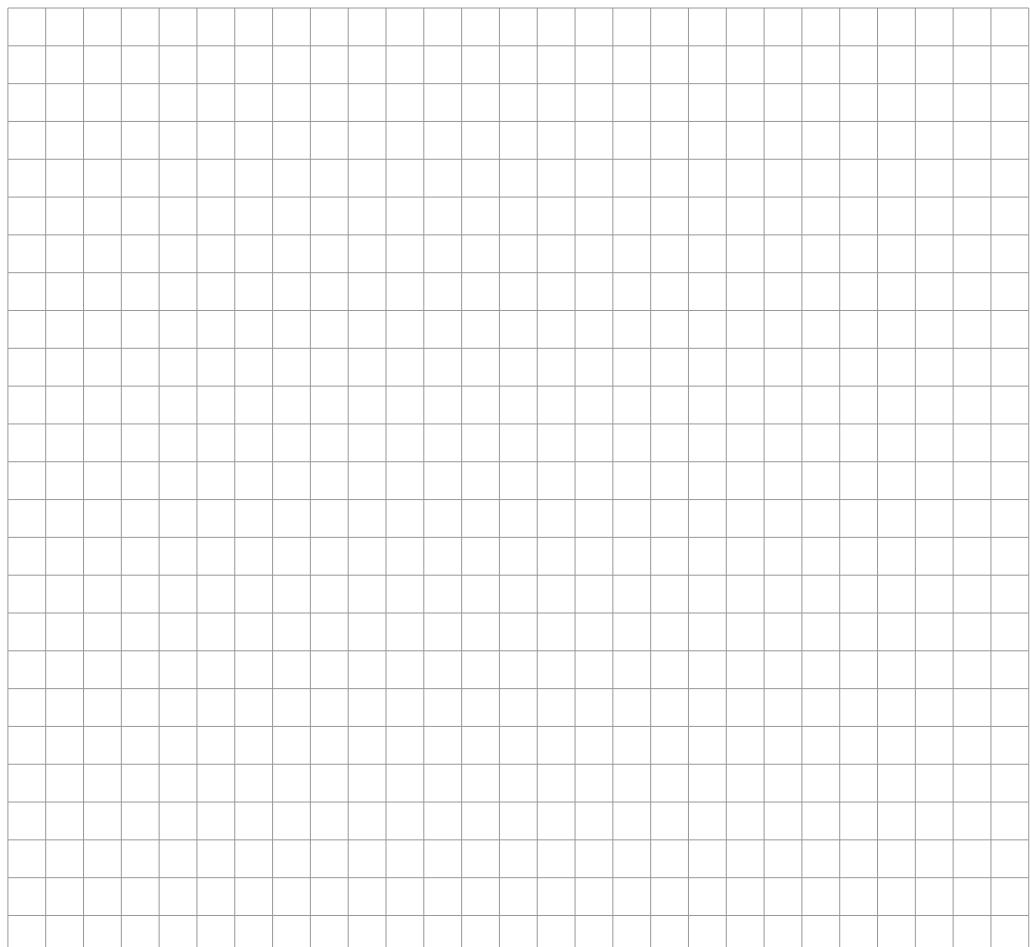
Die Manipulierbarkeit ist ein zentrales Bewertungskriterium für Roboter. Sie gibt an, wie stark sich Gelenkbewegungen auf den Endeffektor auswirken. Eine gute Manipulierbarkeit liegt dann vor, wenn eine kleine Bewegung der Gelenkwinkel Δq zu einer großen Bewegung des Endeffektors Δx_E führt, oder anders ausgedrückt, eine kleine Ursache eine große Wirkung erzielt.

Für kleine Auslenkungen gilt $\Delta q \approx \Delta t \dot{q}$ und $\Delta x_E \approx \Delta t \dot{x}_E$, somit sind für den gesuchten Zusammenhang die Einträge der Jacobi-Matrix $J(q)$ entscheidend:

$$\dot{x}_E = J(q) \dot{q}. \quad (4.35)$$

Aus Gleichung (4.35) resultiert über die Berechnung der Eigenwerte und Eigenvektoren von $J(q)$ und einer Basistransformation folgendes Gleichungssystem (voller Rang und Diagonalisierbarkeit der quadratischen Jacobi-Matrix vorausgesetzt):

$$\dot{x}_E^* = \begin{pmatrix} \lambda_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_n \end{pmatrix} \dot{q}^* = \text{diag}(\boldsymbol{\lambda}) \dot{q}^*. \quad (4.36)$$





Die Manipulierbarkeit einzelner „Freiheitsgrade“ des Endeffektors \dot{x}_E^* ergibt sich nach:

- $|\lambda_i|$ groß: gute Manipulierbarkeit („gute Übersetzung“)
- $|\lambda_i|$ klein: schlechte Manipulierbarkeit („schlechte Übersetzung“)
- $|\lambda_i| = 0$: keine Manipulierbarkeit (Singularität, Rangverlust)

Zu beachten ist, dass die \dot{q}_i^* nicht mehr direkt mit den Gelenkwinkelgeschwindigkeiten \dot{q}_i korrespondieren, da ein Basiswechsel durchgeführt wurde. Selbiges gilt für die Endeffektorfreiheitsgrade \dot{x}_E^* .

Definition
Manipulierbarkeit

Die Manipulierbarkeit des Gesamtsystems ergibt sich zu

$$\mu(\mathbf{q}) = \left| \prod_i^n \lambda_i \right| = |\det(\mathbf{J}(\mathbf{q}))|. \quad (4.37)$$

Ziel ist häufig die Maximierung der Manipulierbarkeit bzw. die Festlegung einer Konfiguration, die optimale Manipulierbarkeit ermöglicht. Dies setzt vor allem voraus, dass kein Eigenwert λ_i gleich null ist (vgl. Abschnitt 4.6).

Anmerkung

Besitzt der Endeffektor translatorische und rotatorische Freiheitsgrade, ist die Jacobi-Matrix inhomogen bezüglich ihrer physikalischen Einheiten. In diesem Fall wird empfohlen, vor der Berechnung der Manipulierbarkeit oder vergleichbarer Leistungsmerkmale eine Homogenisierung der Jacobi-Matrix durchzuführen, um aussagekräftige Resultate zu bekommen. Für die Homogenisierung der Jacobi-Matrix existieren verschiedene Ansätze, wie beispielsweise das Einführen einer charakteristischen Länge als Gewichtungsfaktoren für die Position und Orientierung oder die Beschreibung der Endeffektorlage durch zwei (planarer Mechanismus) bzw. durch drei Punkte (sphärischer Mechanismus). Weiterführendes kann in [SSS98; Ang06] nachgelesen werden und wird in der Vorlesung Robotik II näher behandelt.

4.5 Statisches Kraft- und Momentmodell

Aufbauend auf den in Abschnitt 4.2 und Abschnitt 4.3 hergeleiteten Jacobi-Matrizen kann nun eine Betrachtung der Wechselwirkung zwischen am Endeffektor angreifenden externen Kräften/Momenten sowie den daraus resultierenden Kräften/Momenten der Roboterachsen erfolgen, vgl. Abschnitt 4.5.1. Dies ist die Grundlage für die in Abschnitt 4.5.2 behandelte Steifigkeitsanalyse eines Roboters. Den Überlegungen liegen folgende Annahmen zugrunde:

- starre Roboterglieder in Ruhe,
- Gelenke und Führungen spiel- und reibungsfrei.

4.5.1 Wechselwirkung mit der Umwelt

Anhand der in Bild 4.3 gezeigten Roboterskizze wird der Zusammenhang zwischen einer externen, am Endeffektor des Roboters angreifenden Kraft und den resultierenden Gelenkkräften und -momenten untersucht. Der (verallgemeinerte) Kraftangriffsvektor am Endeffektor lautet:

$$\mathcal{F} = \begin{pmatrix} \mathbf{F} \\ \mathbf{M} \end{pmatrix} \in \mathbb{R}^6. \quad (4.38)$$

Zur Herleitung des Zusammenhangs wird das Prinzip der virtuellen Arbeit angewendet. Dabei wird davon ausgegangen, dass die externe Kraft \mathcal{F} zu einer kleinen Änderung der Gelenkwinkel Δq und somit zu einer kleinen Auslenkung am Endeffektor Δx_E führt.

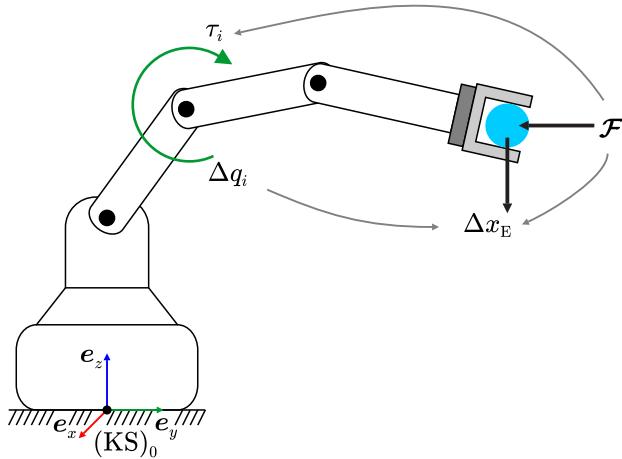


Bild 4.3: Kraftwechselwirkung mit der Umwelt

Die Arbeit ausgedrückt in Endeffektorkoordinaten ist gleich der Arbeit ausgedrückt in Gelenkkordinaten:

$$\Delta x_E^T \mathcal{F} = \Delta q^T \tau. \quad (4.39)$$

Daraus folgt unter Verwendung der differentiellen Kinematik:

$$(\mathbf{J}(q) \Delta q)^T \mathcal{F} = \Delta q^T \tau \Leftrightarrow \Delta q^T \mathbf{J}^T(q) \mathcal{F} = \Delta q^T \tau \quad \forall \Delta q. \quad (4.40)$$

Somit ergibt sich:

$$\tau = \mathbf{J}^T(q) \mathcal{F} \Leftrightarrow \mathcal{F} = (\mathbf{J}^T(q))^{-1} \tau. \quad (4.41)$$

Am Endeffektor angreifende Kräfte werden demnach durch die transponierte Jacobi-Matrix in den Gelenkdrehmomentraum abgebildet.

4.5.2 Steifigkeitsanalyse

Der nach Gleichung (4.41) gewonnene Zusammenhang kann nun zur Analyse der Endeffektorsteifigkeit herangezogen werden. Dabei sei angenommen, dass die Gelenke linear elastisch sind und durch Federkonstanten k_i beschrieben werden können. Es gelte also folgende Beziehung zwischen Drehmomentvektor τ und Auslenkung der Gelenke Δq :

$$\tau = \mathbf{K} \Delta q, \quad \text{mit} \quad \mathbf{K} = \text{diag}(k_1, \dots, k_n) = \text{diag}(\mathbf{k}). \quad (4.42)$$

Dies erlaubt die Bestimmung der aus dem Kraftangriff resultierenden Endeffektorverschiebung:

$$\tau = \mathbf{K} \Delta q, \quad (4.43)$$

$$\mathbf{J}^T(q) \mathcal{F} = \mathbf{K} \Delta q, \quad (4.44)$$

$$\mathbf{J}^T(q) \mathcal{F} = \mathbf{K} \mathbf{J}^{-1}(q) \Delta x_E, \quad (4.45)$$

$$\Delta x_E = \underbrace{\mathbf{J}(q) \mathbf{K}^{-1} \mathbf{J}^T(q)}_{\mathbf{C}_x} \mathcal{F}, \quad (4.46)$$

hierbei ist \mathbf{C}_x die symmetrische Nachgiebigkeitsmatrix ($\mathbf{C}_x = \mathbf{C}_x^T$). Dieser Zusammenhang ermöglicht Aussagen hinsichtlich der Steifigkeit eines Roboters in seinem Arbeitsraum unter der

Annahme, dass lediglich die Gelenke eine von null verschiedene Nachgiebigkeit besitzen. Zu beachten ist, dass die Steifigkeit des Roboters arbeitspunktabhängig ist, da $C_x = C_x(q)$ gilt. Des Weiteren beinhaltet die Nachgiebigkeitsmatrix C_x von null verschiedene Einträge, die nicht auf der Diagonalen liegen. Dies hat zur Folge, dass z. B. eine nur in x -Richtung angreifende Kraft eine Auslenkung in andere Richtungen verursacht, wie nachstehende Gleichung verdeutlicht:

$$\Delta x_E = C_x F_x = \begin{pmatrix} c_{11} & \dots & c_{16} \\ \vdots & \ddots & \vdots \\ c_{16} & \dots & c_{66} \end{pmatrix} \begin{pmatrix} F_x \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} c_{11} \\ \vdots \\ c_{16} \end{pmatrix} F_x. \quad (4.47)$$

Hauptrichtung der Nachgiebigkeit

Oft sind die Hauptrichtungen der Nachgiebigkeit eines Roboters von Interesse. Diese ergeben sich aus den Eigenvektoren (und Eigenwerten) der symmetrischen Nachgiebigkeitsmatrix C_x . Die Bestimmung der Eigenvektoren und Eigenwerte erfolgt nach bekanntem Ansatz:

Ausgehend von

$$\Delta x_E = C_x \mathcal{F} \quad (4.48)$$

erhält man:

$$C_x f_{EV} = \lambda f_{EV} \Rightarrow (C_x - \lambda E) f_{EV} = 0, \quad (4.49)$$

mit der Einheitsmatrix E . Damit lassen sich die Eigenwerte λ_i bestimmen, die mit der Nachgiebigkeit des Roboters korrespondieren. Je größer der Eigenwert, desto nachgiebiger ist dieser:

- betragsmäßig größter Eigenwert: zugehöriger Eigenvektor zeigt in die „weichste“ Richtung,
- betragsmäßig kleinster Eigenwert: zugehöriger Eigenvektor zeigt in die „steifste“ Richtung.

Die Bestimmung der Eigenvektoren f_{EV_i} aus Gleichung (4.49) liefert die Richtungen der durch die Eigenwerte charakterisierten Nachgiebigkeiten (Endeffektorauslenkungen). Es gilt (Voraussetzungen vgl. Abschnitt 4.4):

$$B^{-1} C_x B = \begin{pmatrix} \lambda_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_n \end{pmatrix} = D \Leftrightarrow C_x = B D B^{-1}, \quad (4.50)$$

mit

$$B = (f_{EV_1}, \dots, f_{EV_n}). \quad (4.51)$$

Nach Einsetzen in Gleichung (4.48) folgt:

$$\Delta x_E = B D B^{-1} \mathcal{F} \quad | \rightarrow B^{-1} \quad (4.52)$$

$$\underbrace{B^{-1} \Delta x_E}_{\Delta x_E^*} = D \underbrace{B^{-1} \mathcal{F}}_{\mathcal{F}^*}, \quad (4.53)$$

$$\Delta x_E^* = D \mathcal{F}^*. \quad (4.54)$$

4.6 Singularitäten

Die Jacobi-Matrix eines Roboters kann für bestimmte Gelenkwinkel q singulär werden. In diesen Stellungen, den sogenannten Singularitäten, verliert der Roboter mindestens einen seiner Endeffektorfreiheitsgrade (Degeneration). Mathematisch kommt es zu einem Rangverlust und die Determinante der Jacobi-Matrix resultiert zu null. Dies hat zur Folge, dass die Jacobi-Matrix nicht invertierbar ist. Als Konsequenz tritt, wie im Verlauf des Abschnitts noch gezeigt, ein Verlust von Freiheitsgraden am Endeffektor auf. Singularitäten können sowohl auf der Arbeitsraumgrenze als auch inmitten des Arbeitsraums auftreten. In der Praxis erweisen sich die Singularitäten innerhalb des Arbeitsraums als besonders problematisch, da die Arbeitsraumgrenzen grundsätzlich durch verschiedene Mechanismen steuerungstechnisch überwacht werden. Aus diesem Grund konzentrieren sich die folgenden Ausführungen auf singuläre Stellungen innerhalb des Arbeitsraums. Am Beispiel des planaren RRR-Roboters (vgl. Bild 4.4) wird das Vorgehen zur Bestimmung der Singularitäten erläutert.

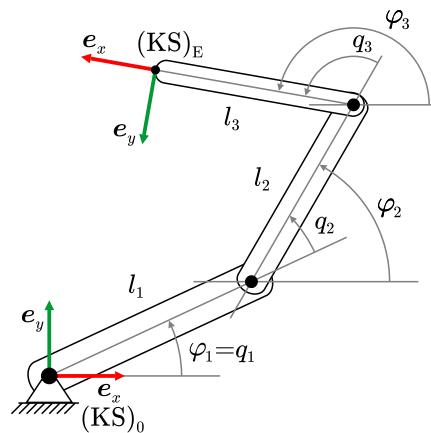


Bild 4.4: Planarer RRR-Roboter

In einer singulären Stellung ist die Manipulierbarkeit $\mu(\mathbf{q}) = |\prod_i^n \lambda_i| = |\det(\mathbf{J}(\mathbf{q}))|$ minimal (null).

Anmerkung

Planarer RRR-Roboter

Die Lage des Endeffektors sei:

$$\mathbf{x}_E = ((0)x_E, (0)y_E, \theta_E)^T. \quad (4.55)$$

Die Gelenkwinkel sind wie folgt definiert:

$$\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \varphi_3)^T = (q_1, q_1 + q_2, q_1 + q_2 + q_3)^T. \quad (4.56)$$

Die Segmentlängen werden vereinfacht zu eins gesetzt:

$$l_1 = l_2 = l_3 = 1. \quad (4.57)$$

Die Jacobi-Matrix des RRR-Roboters lautet:

$$\mathbf{J}(\boldsymbol{\varphi}) = \frac{\partial \mathbf{x}_E(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi}} = \begin{pmatrix} -\sin(\varphi_1) & -\sin(\varphi_2) & -\sin(\varphi_3) \\ \cos(\varphi_1) & \cos(\varphi_2) & \cos(\varphi_3) \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.58)$$

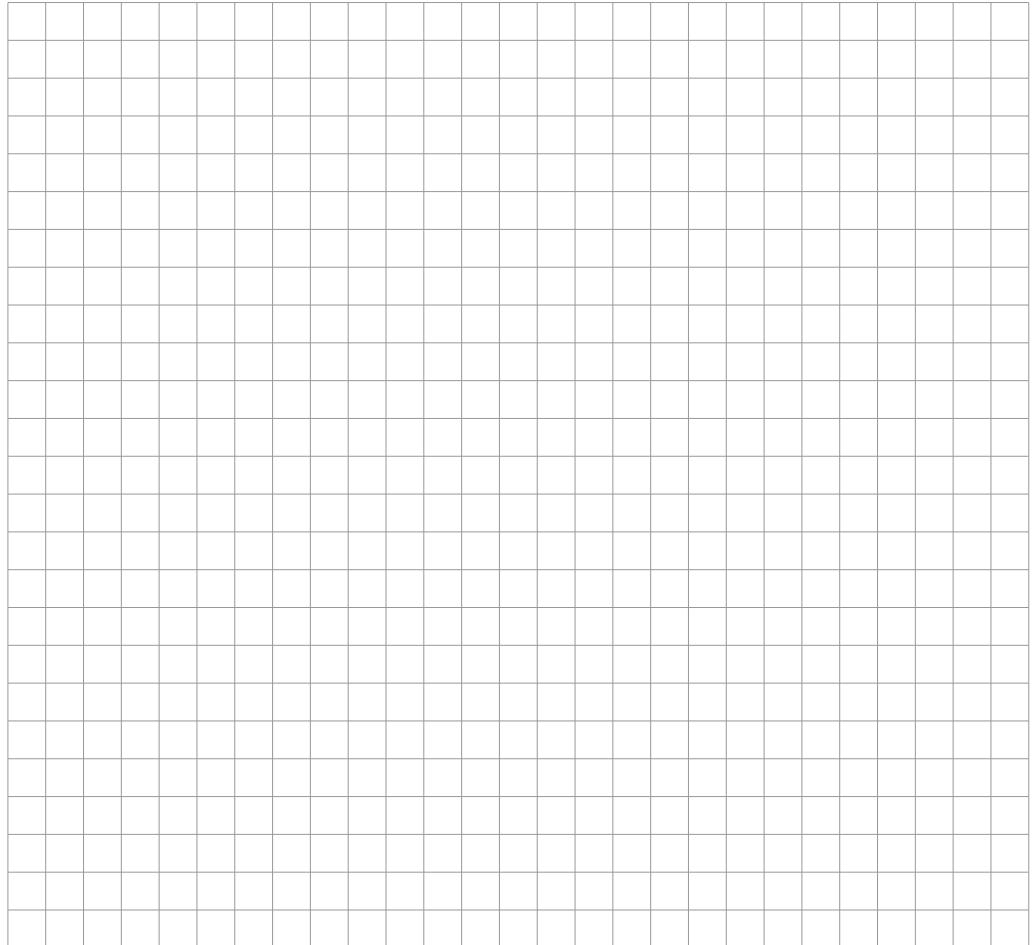
Beispiel

Die Determinante der Jacobi-Matrix berechnet sich zu:

$$\det \mathbf{J}(\varphi) = -\sin(\varphi_1) \cos(\varphi_2) + \cos(\varphi_1) \sin(\varphi_2) \quad (4.59)$$

$$= \sin(\varphi_2) \cos(\varphi_1) - \cos(\varphi_2) \sin(\varphi_1) \quad (4.60)$$

$$= \sin(\varphi_2 - \varphi_1) = \sin(q_1 + q_2 - q_1) = \sin(q_2). \quad (4.61)$$



Die singulären Konfigurationen des RRR-Roboters sind in Bild 4.5 gezeigt.

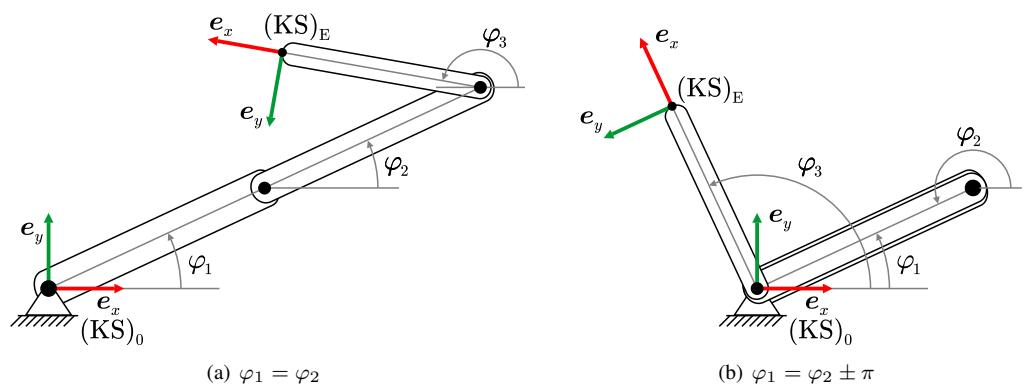


Bild 4.5: Singuläre Konfigurationen des planaren RRR-Roboters

Praktische Bedeutungen

Mathematisch sind Singularitäten bei einer Gelenkstellung $\mathbf{q} = \mathbf{q}_{\text{sing}}$ folgendermaßen charakterisiert:

$$\lim_{\mathbf{q} \rightarrow \mathbf{q}_{\text{sing}}} \det(\mathbf{J}(\mathbf{q})) = \lim_{\mathbf{q} \rightarrow \mathbf{q}_{\text{sing}}} \prod_{i=1}^n \lambda_i = 0. \quad (4.62)$$

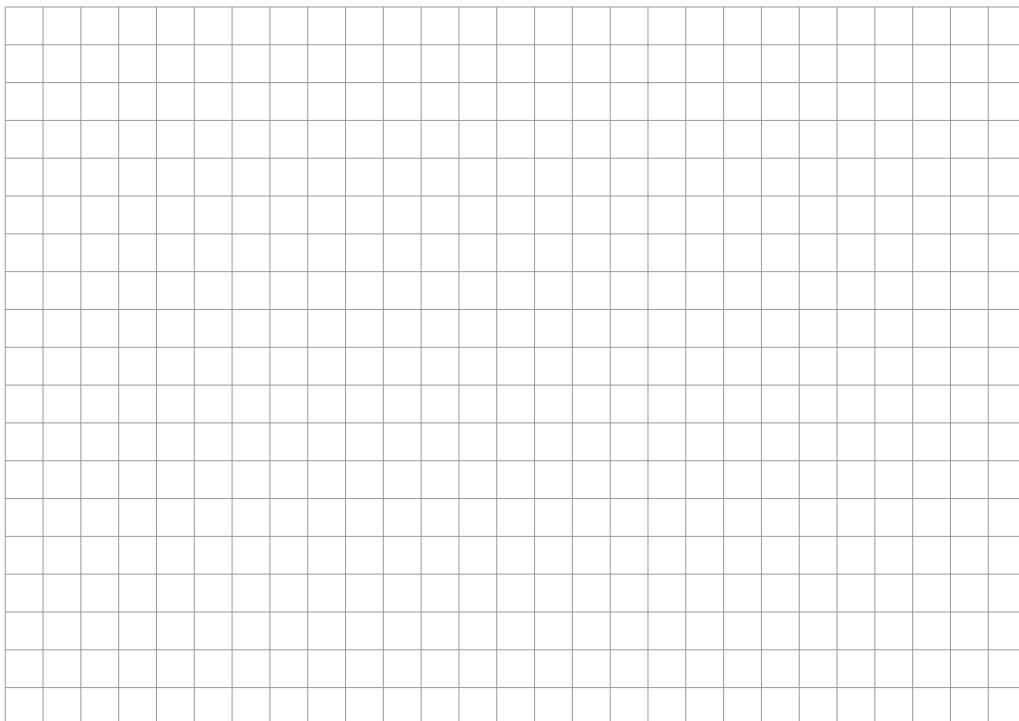
Es kommt zu einem Rangabfall der Jacobi-Matrix, da für mindestens einen Eigenwert λ_i von $\mathbf{J}(\mathbf{q})$ gilt: $\lambda_i = 0$. Daraus folgt, dass mindestens eine Zeile der Jacobi-Matrix als Linearkombination anderer Zeilen ausgedrückt werden kann und nach Elementarumformungen zu $\mathbf{0}^T$ resultiert. Die Variable $\dot{\mathbf{x}}_E^*$ bezeichnet in diesem Zusammenhang den Vektor der Endeffektorgeschwindigkeiten, der sich nach Elementarumformungen aus $\dot{\mathbf{x}}_E$ berechnen lässt:

$$\dot{\mathbf{x}}_E = \mathbf{J}(\mathbf{q}_{\text{sing}}) \dot{\mathbf{q}} \quad \Rightarrow \quad \dot{\mathbf{x}}_E^* = \begin{pmatrix} \cdots \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T \end{pmatrix} \dot{\mathbf{q}}, \quad (4.63)$$

falls für mindestens einen Eigenwert von \mathbf{J} ein Eigenwert $\lambda_i = 0$ existiert.

Demnach kann mindestens ein Freiheitsgrad bzw. eine Freiheitsgradkombination des Endeffektors nicht mehr variiert werden, da der Vektor der Gelenkwinkelgeschwindigkeit $\dot{\mathbf{q}}$ mit $\mathbf{0}^T$ multipliziert wird. In Gleichung (4.63) ist dies die letzte Zeile. Dieser verlorene Freiheitsgrad muss jedoch nicht notwendigerweise mit einem der kartesischen Freiheitsgrade des Roboters zusammenfallen, sondern er ergibt sich aus den zuvor durchgeführten Elementarumformungen. Für die restlichen Freiheitsgrade des Endeffektors stehen allerdings sämtliche Freiheitsgrade des Jointspace zur Verfügung. Es handelt sich also um ein redundantes System (vgl. Kapitel 5).

Wie nachstehend gezeigt, treten in der Nähe von Singularitäten (d. h. $\mathbf{q} \rightarrow \mathbf{q}_{\text{sing}}$) sehr hohe Gelenkgeschwindigkeiten auf, die sowohl aus Applikations- als auch aus Robotersicht zu vermeiden sind.



**Beispiel****Singuläre Stellung eines KUKA KR15 Industrieroboters**

Das folgende Beispiel veranschaulicht die hohen Gelenkgeschwindigkeiten, die in der Nähe singulärer Konfigurationen auftreten. Eine singuläre Stellung liegt in diesem Beispiel (siehe Bild 4.6) vor, wenn sich der Endeffektor entlang der Hochachse des Roboters ausrichtet. Die

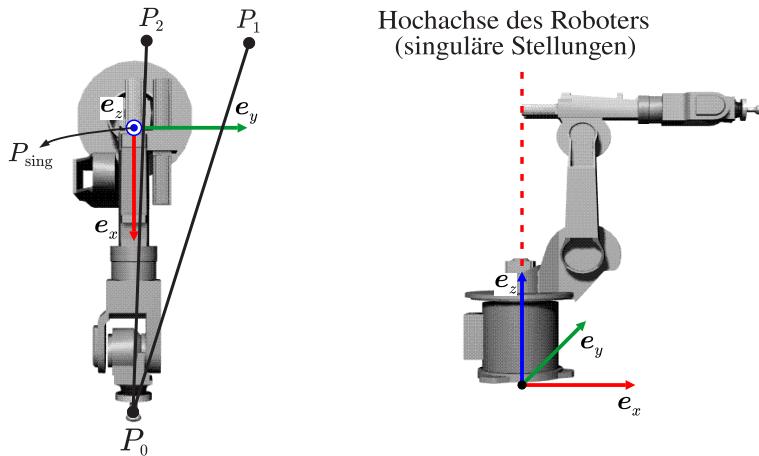


Bild 4.6: Roboterbewegung von Punkt P_0 zu P_1 bzw. P_2 ; die Bahn P_0P_2 verläuft nahe der Hochachse des Roboters (singularitätsnah)

Bewegung von P_0 zu P_2 führt in die Nähe einer singulären Konfiguration (Roboter entlang Hochachse gestreckt), während die Bewegung von P_0 zu P_1 an keiner Singularität vorbeiführt. Bild 4.6 zeigt die Position und Punkte P_0 , P_1 und P_2 im Arbeitsraum des Roboters.

Winkelgeschwindigkeiten

Bewegt sich der Roboter von P_0 nach P_1 , so treten die in Bild 4.7 gezeigten Gelenkwinkelverläufe und Gelenkwinkelgeschwindigkeiten der ersten drei Achsen auf. Es ist zu erkennen, dass keine hohen Geschwindigkeiten resultieren. In der Nähe der Roboterbahn liegen keine Singularitäten.

Anmerkung

Die Berechnung der Gelenkwinkelgeschwindigkeiten erfolgt mittels inverser differentieller Kinematik. Details dazu später in Kapitel 5 und Kapitel 6.

Bei einer Bewegung des Roboterendeffektors entlang der Bahn P_0P_2 treten die in Bild 4.8 gezeigten Gelenkwinkelverläufe und Gelenkwinkelgeschwindigkeiten auf. Es wird deutlich, dass in der Nähe singulärer Stellungen sehr hohe Gelenkwinkelgeschwindigkeiten auftreten. Im vorliegenden Beispiel ist dies bezüglich Gelenk 1 der Fall. Die mathematische Begründung für die hohen Geschwindigkeiten lautet wie folgt.

Endeffektorsteifigkeiten

Anhand desselben Roboters soll nun noch die statische Kraftanalyse in zwei ausgewählten Konfigurationen (P_0 : nicht singulär, sowie die zuvor gezeigte singuläre Konfiguration P_{sing})

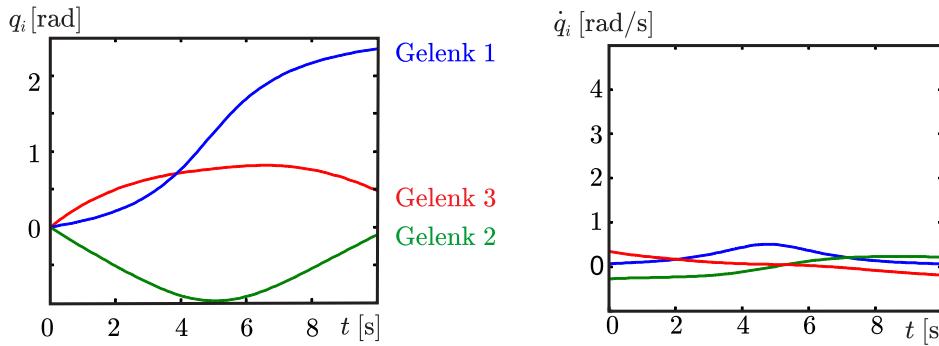


Bild 4.7: Gelenkwinkel und Gelenkinkelgeschwindigkeiten der Achsen 1, 2 und 3 entlang der Bahn P_0P_1

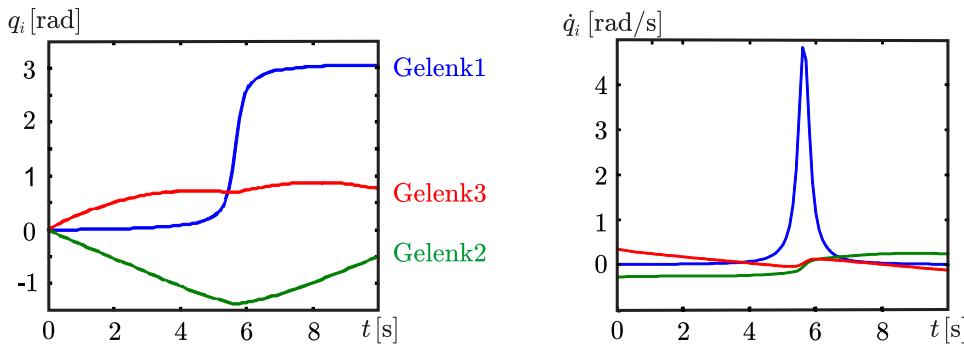


Bild 4.8: Gelenkwinkel und Gelenkinkelgeschwindigkeiten der Achsen 1, 2 und 3 entlang der Bahn P_0P_2

erfolgen. Hierzu wird exemplarisch folgender Kraftvektor \mathcal{F} am Roboterendeffektor eingeleitet (Kraft nur in y -Richtung):

$$\mathcal{F} = (0, F_y, 0, 0, 0, 0)^T. \quad (4.64)$$

Für die Gelenksteifigkeitsmatrix K_q gelte:

$$K_q = \text{diag}(k_1, k_2, \dots, k_6). \quad (4.65)$$

Die kartesische Nachgiebigkeitsmatrix C_x ist:

$$\Delta x_E = \underbrace{\mathbf{J}(q) K^{-1} J^T(q)}_{C_x} \mathcal{F}. \quad (4.66)$$

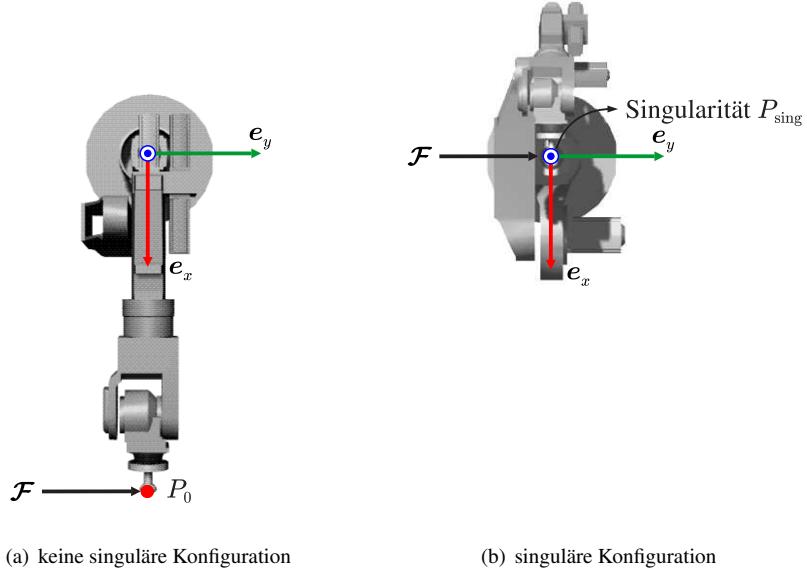
Für die Kraft F_y sowie für die Elemente der Steifigkeitsmatrix k_i sind willkürliche dimensionslose Werte gewählt worden. Hier $k_i = 1000$ und $F_y = 0,1$. Die Roboterglieder sind als ideal steif angenommen, die Nachgiebigkeit resultiert allein aus den Gelenken.

Anmerkung

Aus dem Kraftangriff am Endeffektor für die in Bild 4.9(a) gezeigte Stellung resultiert die folgende Auslenkung:

$$\Delta x_E(q_{P_0}) = \left(0, 128,82, 0, 0, 0, 0, 0, 11 \right)^T. \quad (4.67)$$

Zu sehen ist, dass es aufgrund der Gelenknachgiebigkeiten zu einer Auslenkung in y -Richtung und zu einer Verdrehung um die erste Gelenkachse kommt (fällt hier mit der z -Achse von $(KS)_0$ zusammen).

**Bild 4.9:** Kraftangriff in y -Richtung

zusammen). Wird hingegen die Kraft \mathcal{F} in der singulären Konfiguration (vgl. Bild 4.9(b)) des Roboters eingeleitet, so ergibt sich für die Auslenkung $\Delta \boldsymbol{x}_E$ nach Gleichung (4.46):

$$\Delta \boldsymbol{x}_E(\boldsymbol{q}_{P_{\text{sing}}}) = \begin{pmatrix} 0, 0, 0, 0, 0, 0 \end{pmatrix}^T. \quad (4.68)$$

Anmerkung

Hier handelt es sich um einen Sonderfall. Allgemein gilt für eine singuläre Stellung:

$$\Delta \boldsymbol{x}_E^* = \begin{pmatrix} \cdots \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T \end{pmatrix} \mathcal{F}. \quad (4.69)$$

Dies begründet sich auf der Tatsache, dass

$$\Delta \boldsymbol{x}_E = \mathbf{C} \mathcal{F} = (\mathbf{J} \mathbf{K}^{-1} \mathbf{J}^T) \mathcal{F} \quad (4.70)$$

mittels Elementarumformungen in Gleichung (4.69) überführt werden kann, weil mindestens ein Eigenwert von \mathbf{C} null ist:

$$\det(\mathbf{C}) = \det(\mathbf{J} \mathbf{K}^{-1} \mathbf{J}^T) = \det(\mathbf{J}) \det(\mathbf{K}^{-1}) \det(\mathbf{J}^T) = 0. \quad (4.71)$$

Der Sonderfall zeigt, dass sich der Roboter in diesem Beispiel bei einer extern in y -Richtung angreifenden Kraft nicht bewegt, obwohl die Gelenkelastizitäten dies zuließen. Ursache dafür ist, dass die angreifende Kraft in keinem der Drehgelenke ein Drehmoment erzeugt. Im Allgemeinen ist die Auslenkung in nur einer Achsrichtung bzw. einer Linearkombination von Achsrichtungen (vgl. Gleichung (4.69)) gleich null.

Auswirkungen – Zusammenfassung

Die Auswirkungen singulärer Stellungen sind nachstehend zusammengefasst:

- Verlust von Freiheitsgraden am Endeffektor.
- Für die verbleibenden Freiheitsgrade stehen alle Antriebe zur Verfügung, es handelt sich um ein redundantes System (vgl. Kapitel 5).

- Trotz kleiner Endeffektorgeschwindigkeit kommt es in der Nähe singulärer Konfigurationen zu sehr hohen Gelenkwinkelgeschwindigkeiten.
 - Es kommt zu sehr hohen Steifigkeiten in einzelnen Raumrichtungen. Eine Änderung der Kraft- oder Momenteneinwirkung am Endeffektor in diesen Raumrichtungen hat praktisch keine Auswirkung auf die Aktorkräfte und -momente.

Fazit: Singuläre Konfigurationen sollten bei der Planung einer Roboterbewegung bereits berücksichtigt und vermieden werden.

Maßnahmen zur Vermeidung von Singularitäten

Im Folgenden werden stichpunktartig einige Maßnahmen zur Vermeidung von Singularitäten im Arbeitsraum genannt, auf die im weiteren Verlauf näher eingegangen wird.

- Trajektorienoptimierung

Die Roboterbahnen sind so zu planen, dass singuläre Stellungen vermieden werden.

- Kraftregelung/Impedanzregelung

Die Kraftregelung wird von singulären Stellungen nicht beeinflusst, da bedingt durch die Struktur des Regelkreises keine Invertierung der Jacobi-Matrix notwendig ist (vgl. Bild 4.10).

- kinematische Redundanz

Kinematisch redundante Roboter ermöglichen aufgrund der vorhandenen Anzahl von Gelenkwinkeln (Antrieben) die Vermeidung singulärer Stellungen und werden ausführlich im nachstehenden Kapitel 5 behandelt.

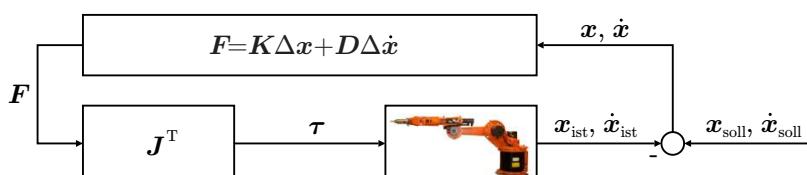
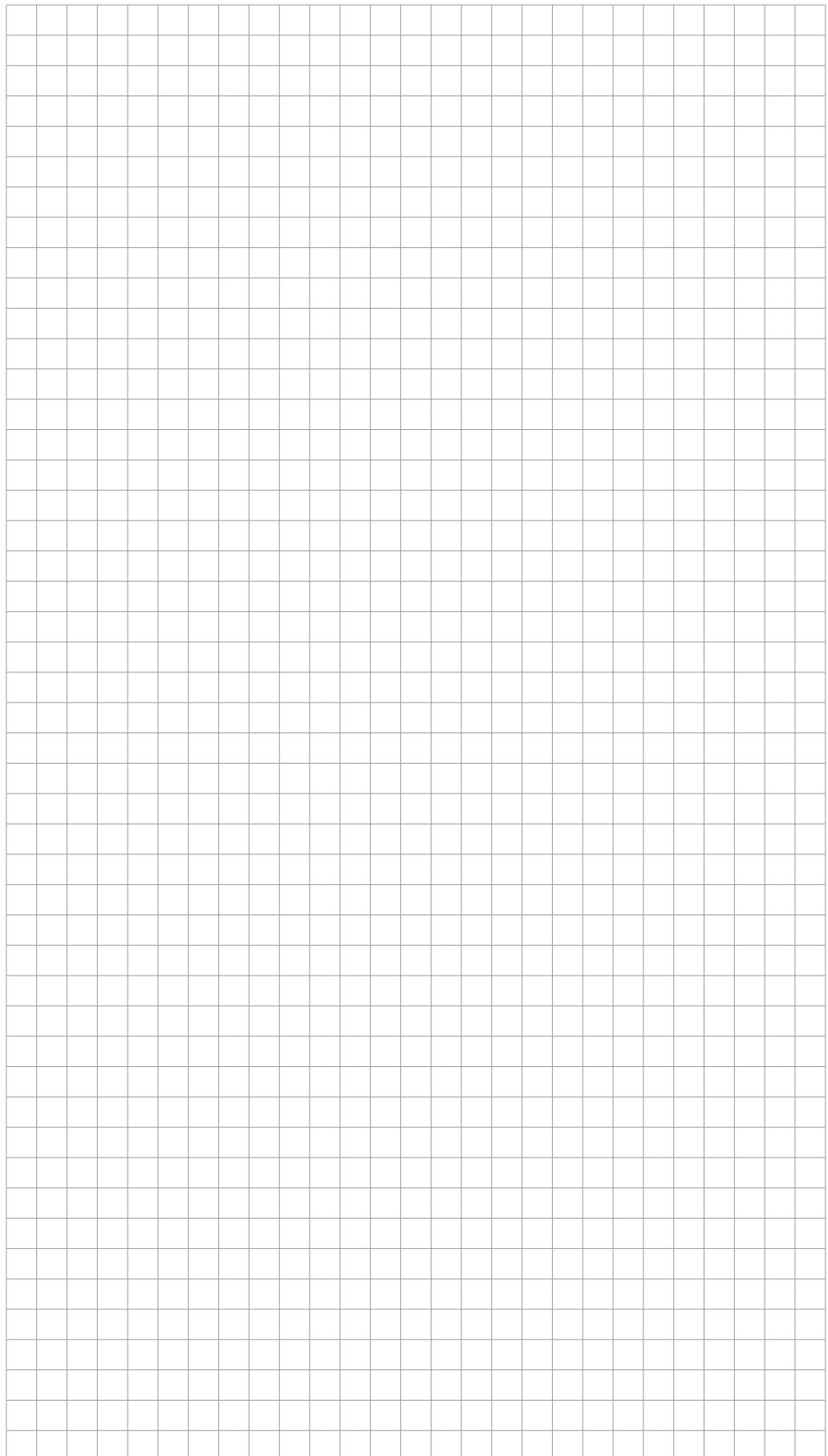


Bild 4.10: Kraftregelung

Fragen zur Selbstkontrolle

1. Wie berechnet sich die analytische und die geometrische Jacobi-Matrix eines allgemeinen seriellkinematischen Roboters? Was ist der wesentliche Unterschied der beiden Jacobi-Matrizen?
 2. Nennen Sie zwei Leistungsmerkmale, die sich basierend auf der Jacobi-Matrix berechnen lassen!
 3. Erläutern Sie den Begriff Singularität und zeichnen Sie qualitativ zwei singuläre Konfigurationen eines planaren RR-Roboters mit zwei Drehantrieben!



5 Kinematisch redundante Roboter

Dieses Kapitel behandelt wesentliche Eigenschaften kinematisch redundanter Roboter. Diese sind dadurch gekennzeichnet, dass sie über mehr Gelenke verfügen als für den Taskspace erforderlich sind (vgl. Definition unten). Die differentielle Kinematik solcher Roboter ist Gegenstand von Abschnitt 5.1 (als Optimierungsproblem) und Abschnitt 5.2 (als Nullraum). Abschnitt 5.3 geht auf die Besonderheiten der inversen Kinematik redundanter Roboter ein.

Als weiterführende Literatur wird empfohlen:

- G. Hirzinger u. a. „DLR’s Torque-Controlled Light Weight Robot III – are we Reaching the Technological Limits now?“ In: *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*. Washington, USA, 2002, S. 1710–1716
- A. Albu-Schäffer u. a. „The DLR Lightweight Robot – Design and Control Concepts for Robots in Human Environments“. In: *Industrial Robot: An International Journal* 34.5 (2007), S. 376–385

Roboter werden als redundant bezeichnet, wenn die Anzahl ihrer Minimalkoordinaten q größer ist als die Anzahl der unabhängigen Endeffektorfreiheitsgrade x_E . Es gilt also:

$$\dim(\mathbf{q}) > \dim(\mathbf{x}_E). \quad (5.1)$$

Definition
kinematische
Redundanz

Ein typischer Vertreter redundanten Kinematiken ist in Bild 5.1 dargestellt. Für diesen Roboter gilt: $\dim(\mathbf{q}) = 7$ und $\dim(\mathbf{x}_E) = 6$. Dem erhöhten technologischen und regelungstechnischen



Bild 5.1: KUKA LBR

Aufwand stehen unter anderem nachstehende Vorteile gegenüber:

- Hindernisvermeidung,
- Singularitätsvermeidung,
- lokale Leistungssteigerung durch Rekonfiguration.

Die Vorteile sind dadurch begründet, dass eine Umkonfiguration der Kinematik vorgenommen werden kann, ohne dass sich die Lage des Endeffektors ändert (vgl. nachstehende Ausführungen). Im Folgenden werden die Auswirkungen kinematischer Redundanz auf die Eigenschaften und Beschreibungsmethoden eines Roboters erläutert.

Kinematische Redundanz hat keine Auswirkung auf die Lösbarkeit der direkten Kinematik $\boldsymbol{x}_E = \mathbf{f}(\boldsymbol{q})$, wohingegen die inverse Kinematik $\boldsymbol{q} = \mathbf{g}(\boldsymbol{x}_E)$ nicht eindeutig lösbar ist, da unendlich viele Lösungen existieren. Das Beispiel des planaren RRRR-Roboters in Bild 5.2 verdeutlicht, dass zu einer gegebenen Endeffektorlage unendlich viele Kombinationen möglicher Gelenkwinkel existieren (abgebildet sind drei). Es sind sogenannte Nullraumbewegungen möglich, bei denen sich die Konfiguration des Roboters ändert, die Lage des Endeffektors im Raum jedoch konstant bleibt.

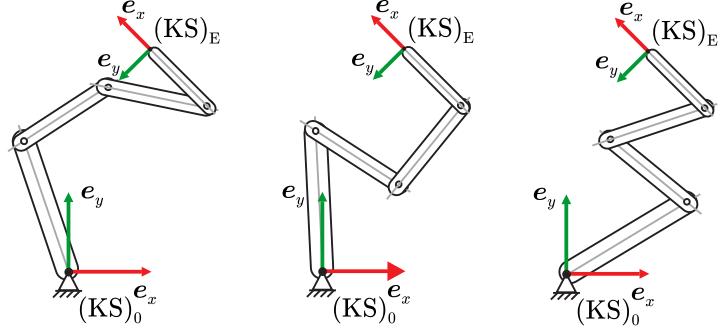


Bild 5.2: Planarer RRRR-Roboter mit $\dim(\boldsymbol{q}) = 4$, $\dim(\boldsymbol{x}_E) = 3$ in unterschiedlichen Konfigurationen

Anmerkung

Im Gegensatz zu kinematisch redundanten Robotern besitzen sogenannte unteraktuierte Manipulatoren mehr unabhängige Endeffektorfreiheitsgrade \boldsymbol{x}_E als Minimalkoordinaten \boldsymbol{q} :

$$\dim(\boldsymbol{q}) < \dim(\boldsymbol{x}_E). \quad (5.2)$$

In diesem Fall ist die direkte Kinematik $\boldsymbol{x}_E = \mathbf{f}(\boldsymbol{q})$ im Allgemeinen nicht lösbar, da notwendige Gelenkinformationen fehlen.

5.1 Redundanz und differentielle Kinematik

Die Jacobi-Matrix eines redundanten Roboters besitzt mehr Spalten als Zeilen ($m < n$):

$$\mathbf{J}(\boldsymbol{q}) = \frac{\partial \mathbf{f}(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{m \times n}, \quad m = \dim(\boldsymbol{x}_E), n = \dim(\boldsymbol{q}). \quad (5.3)$$

Es existieren unendlich viele Lösungen für die inverse differentielle Kinematik, da mehr unabhängige Variablen (n Gelenkwinkelgeschwindigkeiten) als lineare Gleichungen (m Freiheitsgrade am Endeffektor) vorhanden sind. Es handelt sich um ein unbestimmtes lineares Gleichungssystem.

Da die Jacobi-Matrix nicht quadratisch ist, existiert auch \mathbf{J}^{-1} nicht.

Anmerkung

Ein möglicher Ansatz ist die Lösung eines Minimierungsproblems und Ankopplung der differentiellen Kinematik

$$\dot{\boldsymbol{x}}_E = \mathbf{J}(\boldsymbol{q}) \dot{\boldsymbol{q}} \quad (5.4)$$

als Nebenbedingung über Lagrange-Multiplikatoren. Wählt man als zu minimierende Gütfunktion

$$g(\dot{\boldsymbol{q}}) = \frac{1}{2} \dot{\boldsymbol{q}}^T \dot{\boldsymbol{q}} \quad \rightarrow \quad \min, \quad (5.5)$$

so erhält man, wie nachfolgend noch gezeigt:

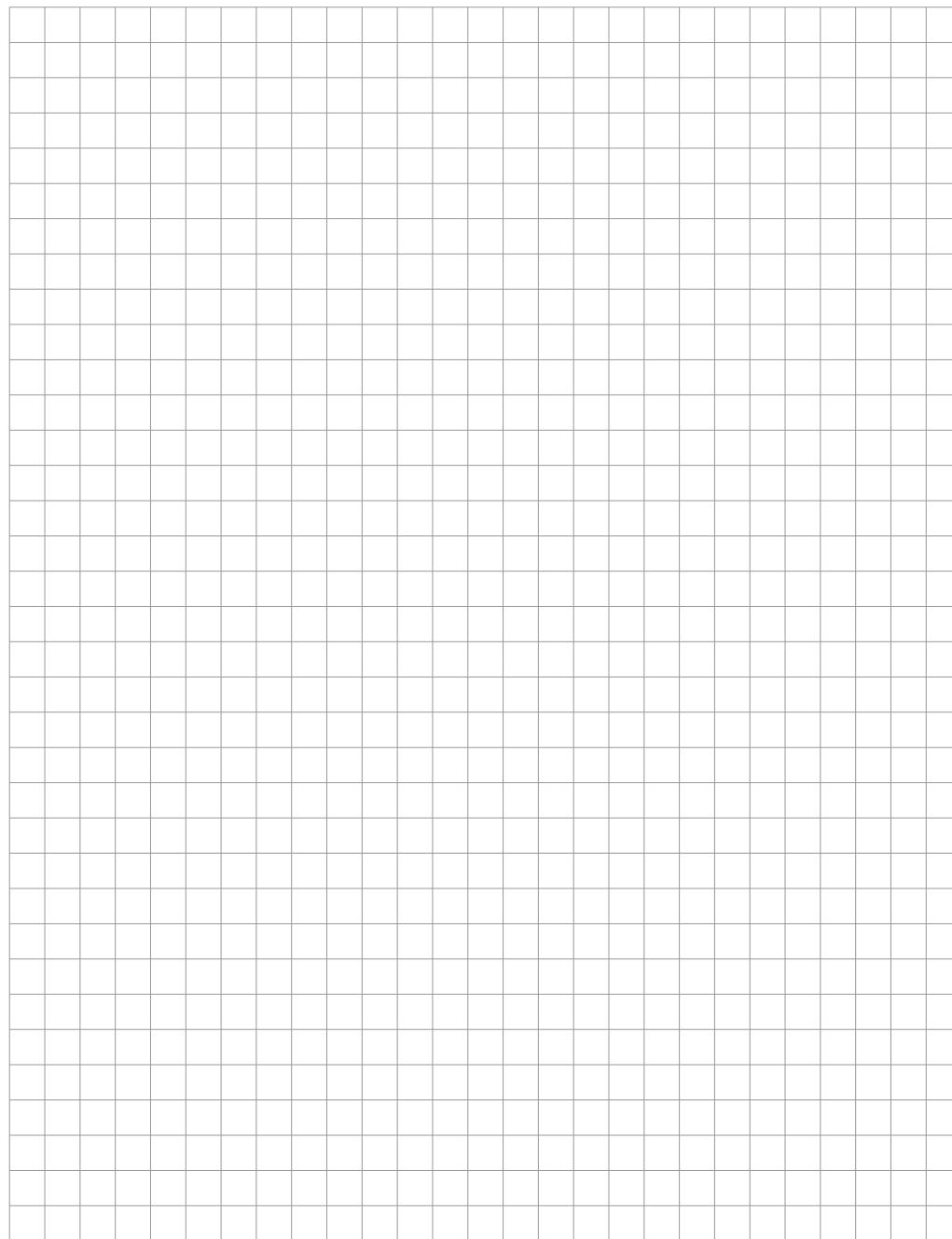
$$\dot{q} = J^+(q) \dot{x}_E, \quad (5.6)$$

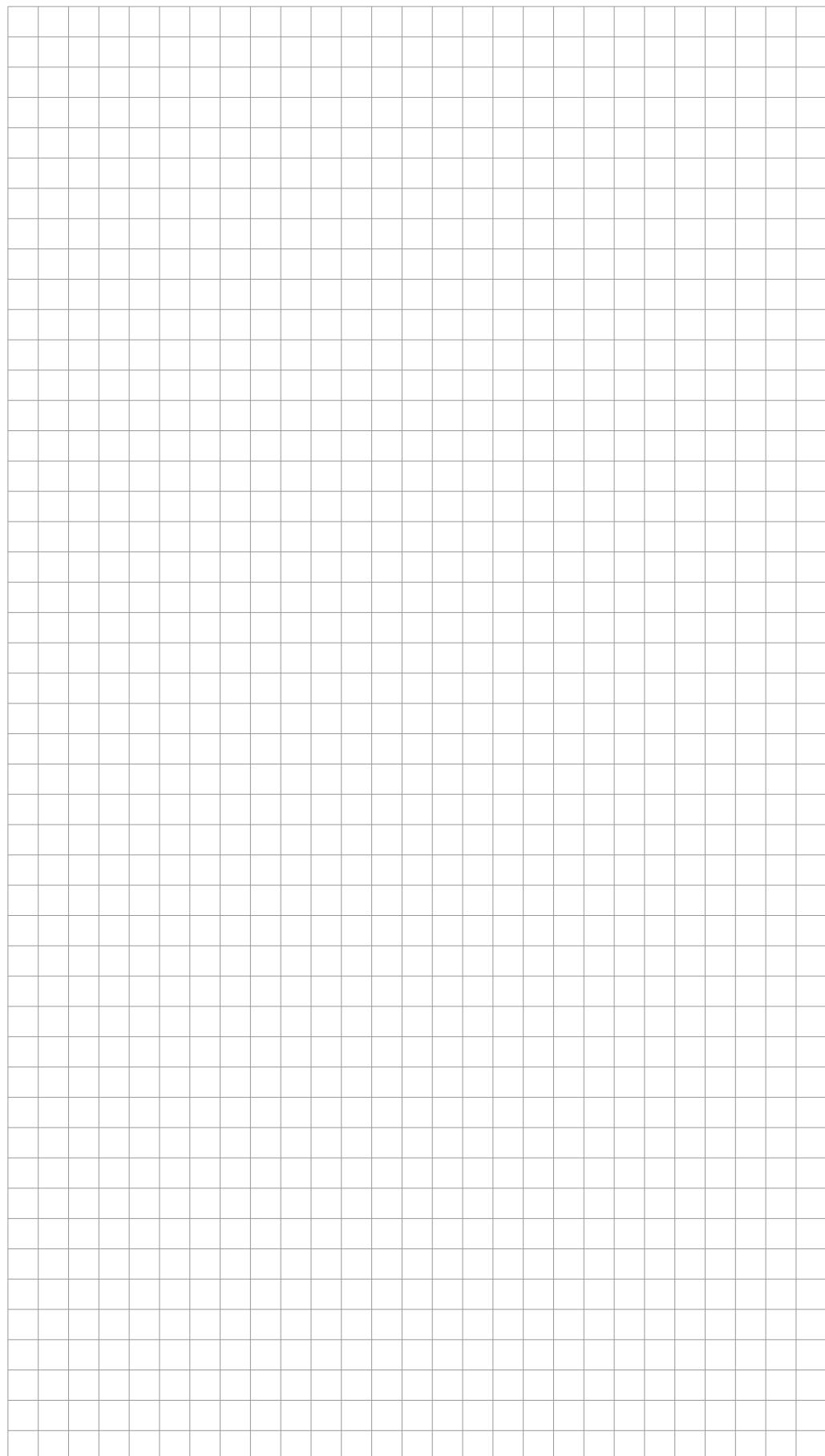
wobei

$$J^+(q) = J^T(q) \left(J(q) J^T(q) \right)^{-1} \quad (5.7)$$

die rechtseitige Moore-Penrose-Inverse (Pseudo-Inverse) ist.

Die Moore-Penrose-Inverse ist die Lösung des oben eingeführten Optimierungsproblems. Es resultiert eine Bewegung des Roboters mit „minimalen Betrag des Gelenkwinkelgeschwindigkeitsvektors“ \dot{q} , wobei gleichzeitig die differentielle Kinematik $\dot{x}_E = J(q) \dot{q}$ berücksichtigt wird, also eine vorgegebene Sollgeschwindigkeit des Endeffektors \dot{x}_E erzeugt wird.





Zur Ableitung von $\lambda^T \mathbf{J} \dot{\mathbf{q}}$ nach $\dot{\mathbf{q}}$ am Beispiel $\mathbf{J} \in \mathbb{R}^{3 \times 4}$:

Anmerkung

$$\begin{aligned} r(\dot{\mathbf{q}}) &= \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} \begin{pmatrix} j_{11} & j_{12} & j_{13} & j_{14} \\ j_{21} & j_{22} & j_{23} & j_{24} \\ j_{31} & j_{32} & j_{33} & j_{34} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} \\ &= \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} \begin{pmatrix} j_{11}\dot{q}_1 + j_{12}\dot{q}_2 + j_{13}\dot{q}_3 + j_{14}\dot{q}_4 \\ j_{21}\dot{q}_1 + j_{22}\dot{q}_2 + j_{23}\dot{q}_3 + j_{24}\dot{q}_4 \\ j_{31}\dot{q}_1 + j_{32}\dot{q}_2 + j_{33}\dot{q}_3 + j_{34}\dot{q}_4 \end{pmatrix} \\ &= \lambda_1(j_{11}\dot{q}_1 + j_{12}\dot{q}_2 + j_{13}\dot{q}_3 + j_{14}\dot{q}_4) + \lambda_2(j_{21}\dot{q}_1 + j_{22}\dot{q}_2 + j_{23}\dot{q}_3 + j_{24}\dot{q}_4) \\ &\quad + \lambda_3(j_{31}\dot{q}_1 + j_{32}\dot{q}_2 + j_{33}\dot{q}_3 + j_{34}\dot{q}_4). \end{aligned}$$

Die Ableitung nach $\dot{\mathbf{q}}$ liefert:

$$\begin{aligned} \frac{d}{d\dot{\mathbf{q}}} r(\dot{\mathbf{q}}) &= \begin{pmatrix} \lambda_1 j_{11} + \lambda_2 j_{21} + \lambda_3 j_{31} \\ \lambda_1 j_{12} + \lambda_2 j_{22} + \lambda_3 j_{32} \\ \lambda_1 j_{13} + \lambda_2 j_{23} + \lambda_3 j_{33} \\ \lambda_1 j_{14} + \lambda_2 j_{24} + \lambda_3 j_{34} \end{pmatrix} = \begin{pmatrix} j_{11} & j_{21} & j_{31} \\ j_{12} & j_{22} & j_{32} \\ j_{13} & j_{23} & j_{33} \\ j_{14} & j_{24} & j_{34} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} \\ &= \mathbf{J}^T \boldsymbol{\lambda}. \end{aligned}$$

Wählt man den allgemeineren Fall für eine Sollbewegung des Endeffektors, bei der sich die Gelenke gewichtet möglichst wenig bewegen, so wird folgende zu minimierende Gütfunktion definiert:

$$g(\dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} \rightarrow \min, \quad (5.8)$$

mit der $n \times n$ Wichtungsmatrix $\mathbf{W} = \text{diag}(w_1, \dots, w_n)$ mit $w_i > 0$. Die Wichtungsmatrix \mathbf{W} ermöglicht es, einzelne Gelenke bzw. deren Gelenkwinkelgeschwindigkeiten unterschiedlich stark zu berücksichtigen. Die Nebenbedingung

$$\dot{\mathbf{x}}_E = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (5.9)$$

wird wie zuvor gezeigt über die Lagrange-Multiplikatortechnik angekoppelt. Als Lösung des Optimierungsproblems erhält man:

$$\dot{\mathbf{q}} = \underbrace{\mathbf{W}^{-1} \mathbf{J}^T(\mathbf{q}) \left(\mathbf{J}(\mathbf{q}) \mathbf{W}^{-1} \mathbf{J}^T(\mathbf{q}) \right)^{-1}}_{\text{Moore-Penrose-Inverse } \mathbf{J}^+(\mathbf{q}), \text{ für } \mathbf{W}=\mathbf{E}} \dot{\mathbf{x}}_E. \quad (5.10)$$

Redundanter, planarer RRR-Roboter

Beispiel

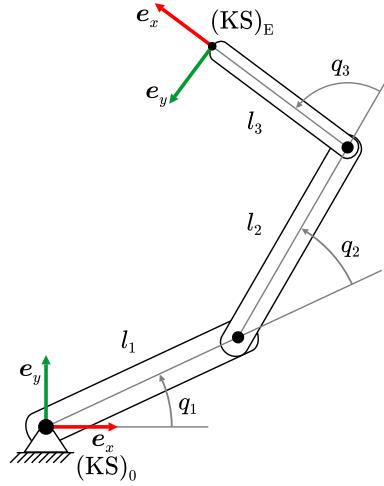
Im Folgenden sei als Beispiel ein planarer RRR-Roboter (siehe Bild 5.3) betrachtet. Die Koordinaten des Endeffektors lauten:

$$\mathbf{x}_E = ((0)x_E, (0)y_E)^T, \quad (5.11)$$

und die Gelenkkoordinaten seien

$$\mathbf{q} = (q_1, q_2, q_3)^T. \quad (5.12)$$

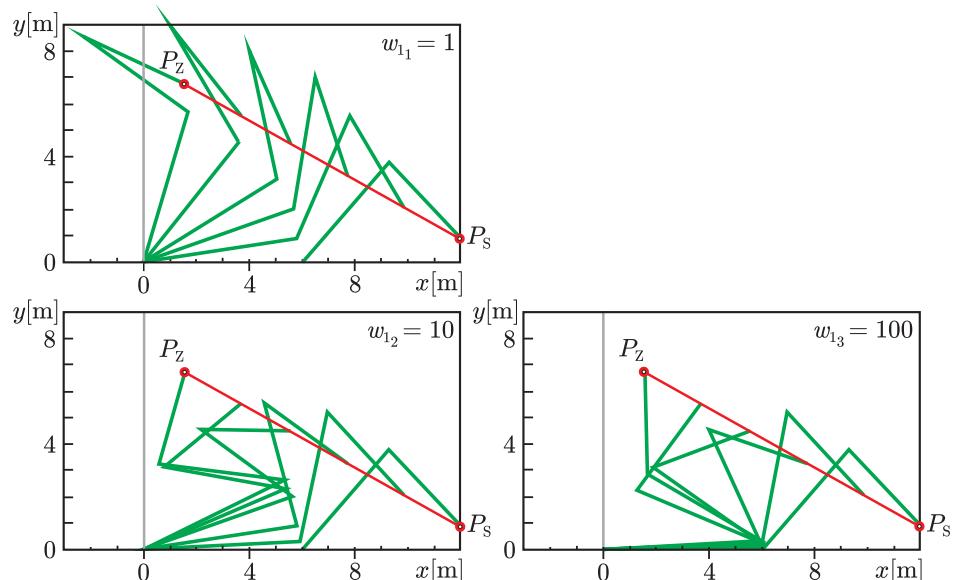
Beispielhaft sind die Segmentlängen gewählt zu: $l_1 = 6 \text{ m}$, $l_2 = 5 \text{ m}$, $l_3 = 4 \text{ m}$.

**Bild 5.3:** Planarer RRR-Roboter

Die Vorgabe lautet: Bewegung von einem Startpunkt P_S zu einem Zielpunkt P_Z mit konstanter Endeffektor-Geschwindigkeit. Die Gelenkwinkelgeschwindigkeiten seien, wie in Gleichung (5.8) dargestellt, mittels Matrix \mathbf{W}_i individuell gewichtet. In diesem Beispiel gelte:

$$\mathbf{W}_i = \begin{pmatrix} w_{1_i} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Anhand der Bild 5.4 wird deutlich, dass mit zunehmender Gewichtung w_{1_i} die Bewegung von q_1 abnimmt.

**Bild 5.4:** Beispiel redundanter, planarer RRR-Roboter

5.2 Nullraum als homogene Lösung der differentiellen Kinematik

Kinematisch redundante Roboter können eine Nullraumbewegung ausführen. Darunter versteht man die Bewegung einzelner oder mehrerer Gelenke, bei der die Endeffektorlage konstant bleibt.

Die Nullraumbewegung eines Roboters kann wie folgt berechnet werden, zunächst jedoch eine kurze Wiederholung der Lösbarkeit linearer Gleichungssysteme:

Die Gleichung

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (5.13)$$

ist lösbar, wenn

$$\text{Rang}(\mathbf{A}|\mathbf{b}) = \text{Rang}(\mathbf{A}). \quad (5.14)$$

Die Lösung hat die allgemeine Form

$$\mathbf{x} = \mathbf{x}_p + \mathbf{x}_h, \quad (5.15)$$

wobei \mathbf{x}_p die spezielle (auch partikuläre) Lösung und \mathbf{x}_h die homogene Lösung ($\mathbf{0} = \mathbf{A}\mathbf{x}_h$) oder auch der Nullraum ist. Die Anzahl der freien Variablen (Dimension des Nullraums) ist $n - \text{Rang}(\mathbf{A})$, mit $n = \dim(\mathbf{x})$.

Die Übertragung obiger Anmerkung auf die differentielle Kinematik ergibt den Nullraum kinematisch redundanter Roboter als homogene Lösung von:

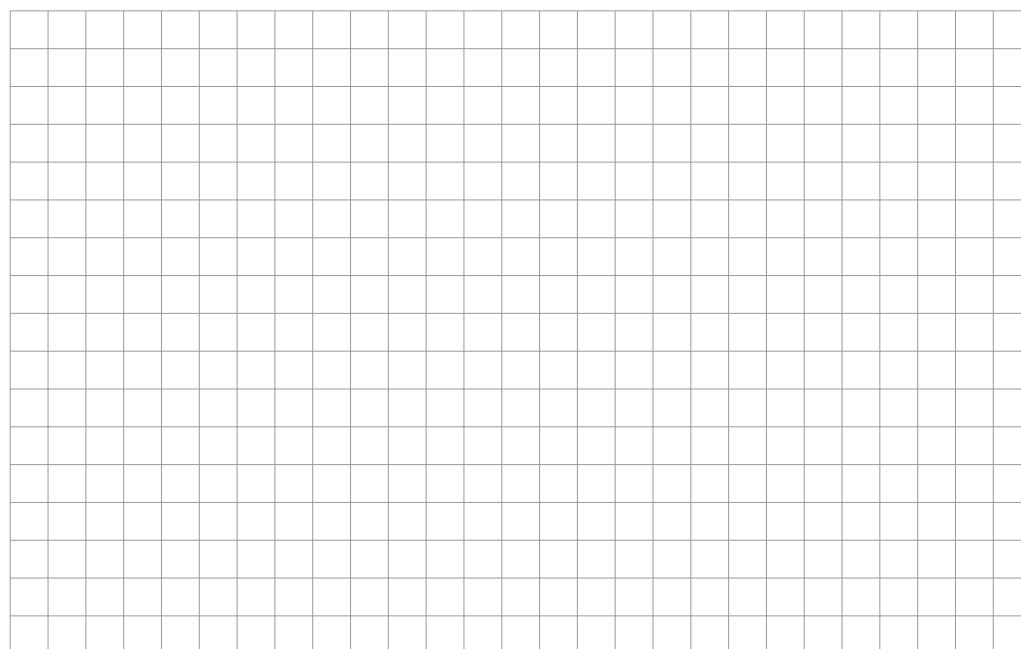
$$\dot{\mathbf{x}}_E = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}, \quad \text{mit} \quad \mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n} \quad (m < n). \quad (5.16)$$

Dabei wird angenommen, dass \mathbf{J} vollen Rang besitzt, der Roboter sich also in keiner singulären Konfiguration befindet:

$$\text{Rang}(\mathbf{J}(\mathbf{q})) = m. \quad (5.17)$$

Das bedeutet im Einzelnen, dass

- das System lösbar ist, da $\text{Rang}(\mathbf{J}(\mathbf{q})|\dot{\mathbf{x}}_E) = \text{Rang}(\mathbf{J}(\mathbf{q})) = m$,
- eine Lösung der Form $\dot{\mathbf{q}} = \dot{\mathbf{q}}_p + \dot{\mathbf{q}}_h$ existiert,
- die Anzahl freier Variablen sich wie folgt berechnen lässt: $\dim(\dot{\mathbf{q}}) - m = n - m$.







Nachtrag zu Singularitäten

Bei Singularitäten tritt ein Verlust mindestens eines Freiheitsgrades im Taskspace auf. Für die differentielle Kinematik gilt die bekannte Beziehung:

$$\dot{x}_E = J(q)\dot{q}, \quad \text{mit } J \in \mathbb{R}^{n \times n},$$

es kommt jedoch zu einem Rangabfall von J ($\text{Rang}(J(q)) < n$).

Beispielhaft sei hier $\text{Rang}(J(q)) = n - 1$ betrachtet. Die Gestalt von J nach Elementarumformungen laute:

$$J^* = \begin{pmatrix} * & ? & ? & ? \\ 0 & * & ? & ? \\ 0 & 0 & * & ? \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Ein „Freiheitsgrad“ (hier: letzte Zeile) kann nicht mehr kommandiert werden; dieser muss nicht zwingend mit einem kartesischen Freiheitsgrad zusammenfallen ($\dot{x}_E = J\dot{q} \rightarrow x_E^* = J^*\dot{q}$ nach Elementarumformungen).

Streicht man diesen Freiheitgrad (hier: die letzte Zeile), so erhält man ein redundantes System, die Jacobi-Matrix lautet somit:

$$J^* = \begin{pmatrix} * & ? & ? & ? \\ 0 & * & ? & ? \\ 0 & 0 & * & ? \end{pmatrix}, \quad \text{mit } J^* \in \mathbb{R}^{3 \times 4}.$$

Es stehen in diesem Beispiel vier Gelenkwinkelfreiheitsgrade für drei (verbliebene) Endeffektorfreiheitsgrade zur Verfügung, die vorherigen Überlegungen gelten analog.

5.3 Redundanz und inverse Kinematik

Neben der differentiellen Kinematik redundanter Roboter soll abschließend die inverse Kinematik redundanter Roboter betrachtet werden. Wie zuvor, existiert auch hier das Problem, dass es sich um ein unterbestimmtes (nichtlineares) Gleichungssystem handelt. Nachstehend werden zwei Möglichkeiten zur Redundanzauflösung bei der Berechnung der inversen Kinematik vorgestellt: Definition eines Optimierungsproblems und Aufstellen zusätzlicher kinematischer Nebenbedingungen.

Der erste Lösungsansatz ist analog zu Abschnitt 5.1. Es wird eine Gütfunktion unter Berücksichtigung der direkten Kinematik als (Gleichungs-)Nebenbedingung optimiert und so die (im Sinne der Gütfunktion) optimalen Gelenkwinkel bei gleichzeitiger Beachtung der Roboterkinematik bestimmt.

Eine gängige Gütfunktion ist die Manipulierbarkeit eines Roboters. Diese wurde für nicht-redundante Roboter in Abschnitt 4.4 eingeführt.

Definition
Manipulierbarkeit
redundanter Roboter

Die Manipulierbarkeit redundanter Roboter berechnet sich zu:

$$\mu(q) = \det(J(q) J^T(q)). \quad (5.18)$$

Die Maximierung der Manipulierbarkeit kann nun zur Lösung der inversen Kinematik redundanter Roboter eingesetzt werden. Die Problemstellung lässt sich wie folgt formulieren:

- Gütfunktion:

$$g(\mathbf{q}) = \det \left(\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}) \right) \rightarrow \max, \quad (5.19)$$

- Nebenbedingung:

$$\mathbf{x}_E - \mathbf{f}(\mathbf{q}) = \mathbf{0}. \quad (5.20)$$

Hier handelt es sich wieder um ein Optimierungsproblem mit Gleichungsnebenbedingung. Die Lösung ist analog zur Herleitung der Moore-Penrose-Inversen in Abschnitt 5.2:

- Ankopplung der Nebenbedingung mittels Lagrange-Multiplikatoren $\boldsymbol{\lambda}$ (man beachte das Minuszeichen vor der Determinanten: berechnet wird ein Minimum):

$$g(\mathbf{q}, \boldsymbol{\lambda}) = -\det \left(\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}) \right) + \boldsymbol{\lambda}^T (\mathbf{x}_E - \mathbf{f}(\mathbf{q})), \quad (5.21)$$

- und Aufstellung der notwendigen Bedingungen:

$$\frac{\partial g(\mathbf{q}, \boldsymbol{\lambda})}{\partial \mathbf{q}} \stackrel{!}{=} \mathbf{0}, \quad \frac{\partial g(\mathbf{q}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{x}_E - \mathbf{f}(\mathbf{q}) \stackrel{!}{=} \mathbf{0}. \quad (5.22)$$

Die Lösung des Gleichungssystems (5.22) ergibt die gewünschte Solllage des Roboterendeffektors bei gleichzeitiger Maximierung der Manipulierbarkeit. Sie ist abhängig vom jeweils vorliegenden Robotertyp, daher kann keine allgemeingültige Lösung angegeben werden.

Der zweite Lösungsansatz zur kinematischen Redundanzauflösung lautet wie folgt: Anstelle einer Gütfunktion können auch kinematische Zwangsbedingungen hinzugefügt werden. Hierzu werden

$$k = \dim(\mathbf{q}) - \dim(\mathbf{x}_E) \quad (5.23)$$

zusätzliche Bedingungen $\mathbf{h}(\mathbf{q}) = \mathbf{0}$ definiert und die direkte Kinematik um k Zeilen erweitert.

Beispielsweise soll die Summe der Gelenkwinkel q_1 und q_2 eines kinematisch redundanten Roboters konstant bleiben. Dies ergibt

$$q_1 + q_2 = c \Rightarrow \mathbf{h}(\mathbf{q}) = q_1 + q_2 - c = 0 \quad (5.24)$$

und führt zu einer Erweiterung der direkten Kinematik:

$$\begin{pmatrix} \mathbf{x}_E \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{f}(\mathbf{q}) \\ \mathbf{h}(\mathbf{q}) \end{pmatrix}. \quad (5.25)$$

Daraus resultiert folgender differentieller Zusammenhang:

$$\begin{pmatrix} \dot{\mathbf{x}}_E \\ \mathbf{0} \end{pmatrix} = \tilde{\mathbf{J}}(\mathbf{q}) \dot{\mathbf{q}}, \quad \text{mit } \tilde{\mathbf{J}}(\mathbf{q}) \in \mathbb{R}^{n \times n} \quad \text{und} \quad \dim(\dot{\mathbf{q}}) = n. \quad (5.26)$$

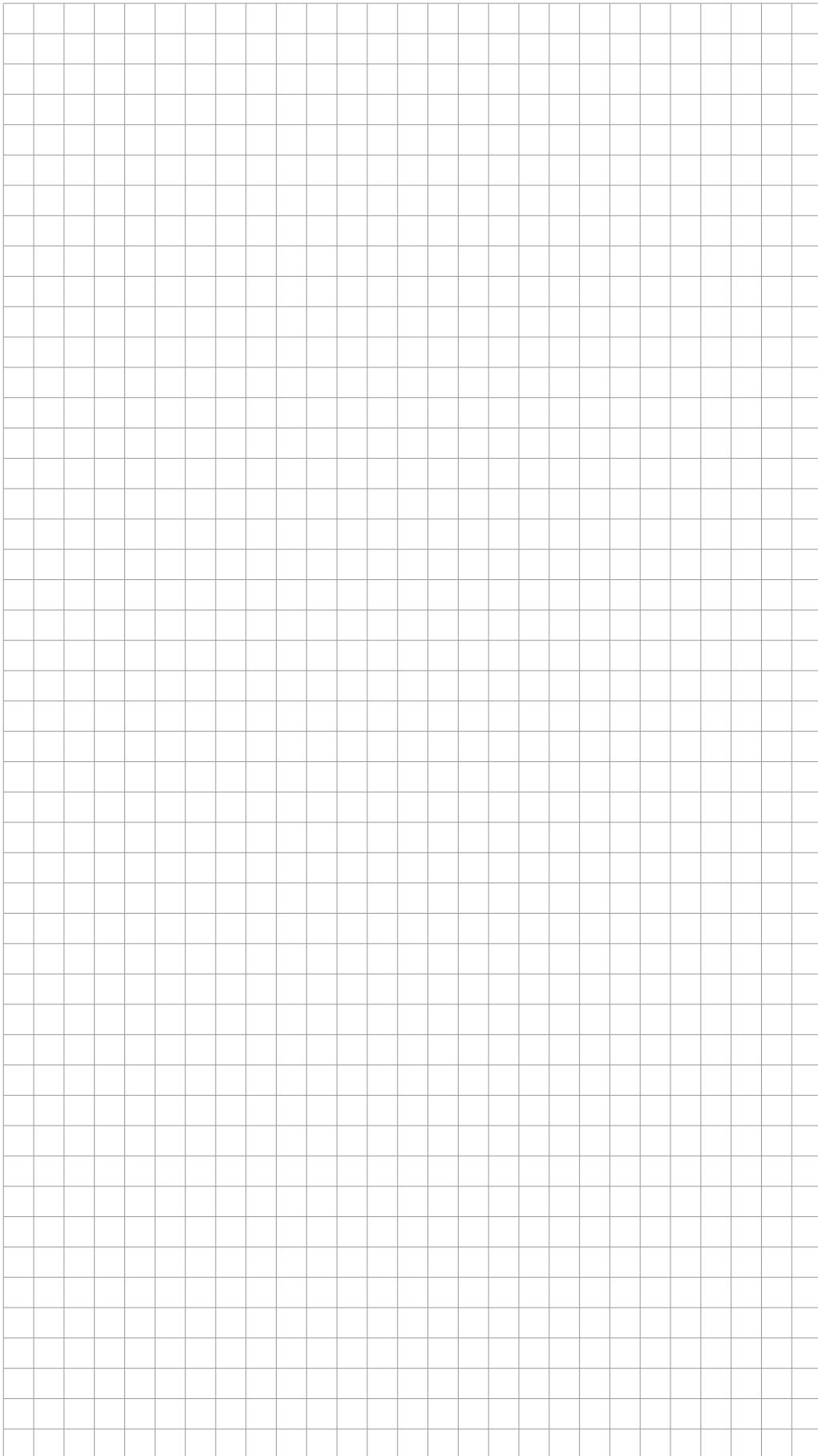
Die Jacobi-Matrix $\tilde{\mathbf{J}}$ ist quadratisch und somit (falls nicht singulär) invertierbar.

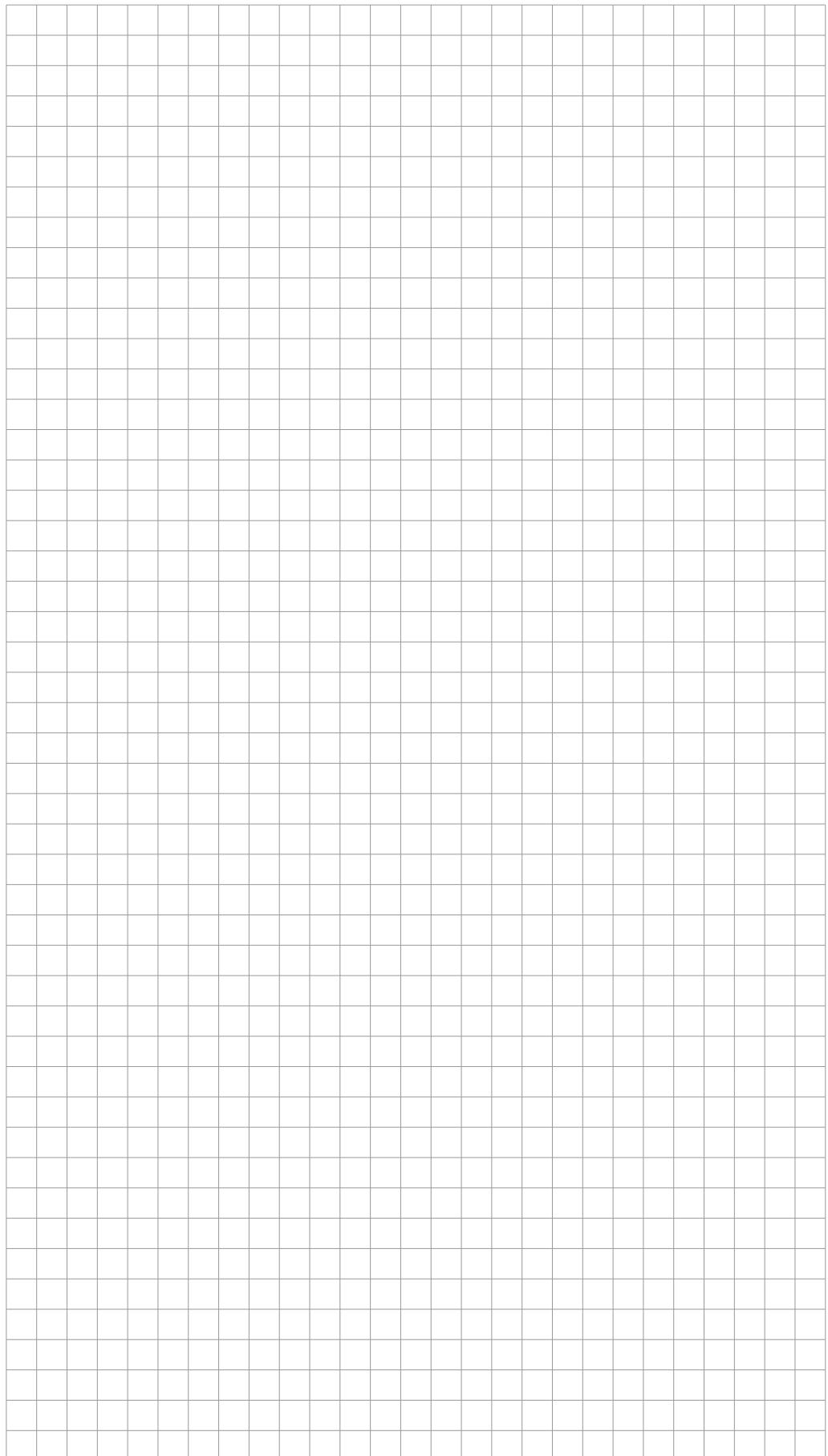
Beispiel

Fragen zur Selbstkontrolle

1. Wodurch zeichnet sich ein kinematisch redundanter Roboter mit seriellkinematischer Struktur aus?

2. Welche Auswirkung haben die zusätzlichen Gelenkfreiheitsgrade auf die direkte und inverse Kinematik?
 3. Nennen Sie Ihnen bekannte Ansätze bzw. Leistungsmerkmale zur (optimalen) Ausnutzung der zusätzlichen Freiheitsgrade!





6 Bahnplanung

Das folgende Kapitel behandelt die Thematik der Bahnplanung. Es wird erläutert, mit welchen Methoden Sollbahnen für Roboter generiert werden können und welche Faktoren in diesem Zusammenhang zu berücksichtigen sind. Zunächst erfolgt eine Definition der Begriffe, die im Zusammenhang mit der Thematik stehen (Abschnitt 6.1). Daraufhin wird die klassische Bahnplanung durch Polynom-Interpolation für eine Einzelachsbewegung beschrieben (Abschnitt 6.2). Auch die Kriterien bezüglich der Auswahl der zur Interpolation eingesetzten Polynome werden diskutiert. Eine gleichzeitige Betrachtung aller Roboterachsen ermöglicht die Bahnplanung im Jointspace (Abschnitt 6.3), welche meist durch eine Synchronisation der Einzelachsbewegungen erfolgt. Anschließend werden die Bahnen von translatorischen Bewegungen im Taskspace geplant (Abschnitt 6.4), gefolgt von der Planung von Rotationen im Taskspace (Abschnitt 6.5). Hierbei werden die Schwierigkeiten bei der Interpolation von Rotationen aufgezeigt und Algorithmen zu deren Bewältigung vorgestellt. Daraufhin wird ein ganzheitlicher Ansatz zur Trajektorienplanung im Taskspace (Abschnitt 6.6) beschrieben. Ein Anwendungsbeispiel für eine energieoptimale Roboterbahn wird anschließend präsentiert (Abschnitt 6.7). Darauf folgen Beispiele zur Robotersteuerung anhand verschiedener Bedieneroberflächen (Abschnitt 6.8). Abschließend werden Algorithmen für die automatisierte Bahnplanung bei komplexen und dynamischen Aufgaben präsentiert und diskutiert (Abschnitt 6.9).

Als weiterführende Literatur wird empfohlen:

- B. Heimann, W. Gerth und K. Popp. *Mechatronik: Komponenten – Methoden – Beispiele*. München, Deutschland: Carl Hanser Verlag, 2006, S. 211–222
- L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Berlin, Deutschland: Springer, 2000, S. 169–196
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 227–256
- W. Khalil und E. Dombre. *Modeling, Identification & Control of Robots*. New York, USA: Routledge, 2002, S. 313–346
- E.B. Dam, M. Koch und M. Lillholm. *Quaternions, Interpolation and Animation*. Techn. Ber. Kopenhagen, Dänemark: Department of Computer Science, University of Copenhagen, 1998, S. 40–48
- D. Eberly. *Quaternion Algebra and Calculus*. Techn. Ber. Washington, USA: Geometric Tools, LLC, 2010
- D. Constantinescu und E. A. Croft. „Smooth and time-optimal trajectory planning for industrial manipulators along specified paths“. In: *Journal of Robotic Systems* 17.5 (2000), S. 233–249
- L. Biagiotti und C. Melchiorri. *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008
- S.M. LaValle. *Planning algorithms*. Cambridge university press, 2006
- T. Asfour und F. Paus. *Robotik I - Einführung in die Robotik, Vorlesung 14 (Grundlagen Bewegungsplanung)*. Karlsruher Institut für Technologie (KIT), 2019
- R. Dillmann und M. Huck. *Planungssysteme*. Springer, 1991, S. 258–266, 286–291

6.1 Einführung

Wie in den vorherigen Kapiteln erfolgt vor der eigentlichen Algorithmik eine Definition und Erläuterung zentraler Begriffe. Hierbei handelt es sich vor allem um die Unterscheidung von Weg und Bahn sowie unterschiedlichen Planungsräumen.

Abgrenzung von Weg und Bahn (Trajektorie)

Definition
Weg und Bahn
(Trajektorie)

Als Weg wird der geometrische Ort aller Punkte/Lagen bezeichnet, die mit der Kinematik zu durchfahren sind. Im Gegensatz dazu enthält eine Bahn/Trajektorie auch Informationen über den zeitlichen Ablauf des Weges.

Wegplanung

Definition
Wegplanung

Dieser Begriff beschreibt die Auswahl eines Weges aus der Menge aller möglichen Wege von einem Startpunkt P_S zu einem Zielpunkt P_Z , beispielsweise den geometrisch kürzesten Weg. Die Zeit spielt hierbei keine Rolle.

Bahnplanung/Trajektorienplanung

Definition
Bahn-/
Trajektorienplanung

Hierunter wird die kinematische Bewegungsplanung verstanden, d. h. die Planung eines stetigen Übergangs von einem Bahnpunkt P_i zu einem folgenden Bahnpunkt P_{i+1} . Die Verknüpfung von Ort und Zeit ergibt sich unter Berücksichtigung einer kinematischen Modellvorstellung (Bewegungsvorgabe).

Räume der Bahnplanung

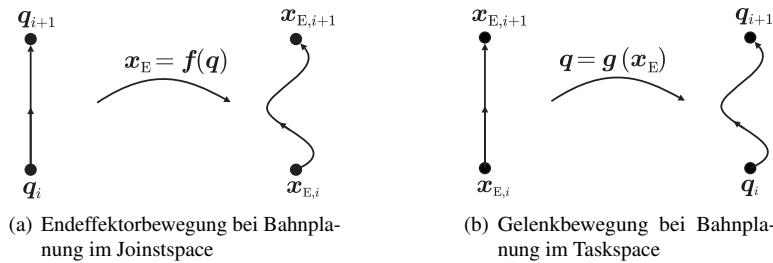
Grundsätzlich kann eine Bahn im Joint- oder Taskspace geplant werden. Je nach Wahl des Planungsraums ergeben sich spezifische Vor- und Nachteile:

- Jointspace (Bahnplanung im Gelenakraum):

Eine lineare Bewegung im Jointspace von einer Gelenkconfiguration q_i nach q_{i+1} führt aufgrund der direkten Kinematik $x_E = f(q)$ in der Regel zu komplexen Bewegungen im Taskspace (siehe Bild 6.1(a)).
- Taskspace (Bahnplanung im Umweltraum):

Analog zum Jointspace führt eine lineare Bewegung im Taskspace von einer Endeffektorlage $x_{E,i}$ nach $x_{E,i+1}$ aufgrund der (häufig rechenintensiven) inversen Kinematik $q = g(x_E)$ zu komplexen Bewegungen im Jointspace (siehe Bild 6.1(b)).

Aus den genannten Zusammenhängen resultiert, dass die schnellste Bewegung eines Roboters von einer Start- in eine Zielkonfiguration in der Regel nicht durch eine geradlinige Bewegung im Taskspace, sondern im Jointspace repräsentiert wird. Die Bahnplanung im Jointspace zeichnet sich gegenüber einer Bahnplanung im Taskspace weiterhin durch einen geringeren Rechenaufwand aus. Außerdem können singuläre Konfigurationen des Roboters im Jointspace problemlos durchfahren werden. Nachteilig ist, dass die aus einer Jointspacebahn resultierende Bewegung im Taskspace schwer vorhersehbar ist. Entsprechend ist für einen sicheren Betrieb eine zusätzliche Überwachung möglicher Kollisionen im Taskspace notwendig.

**Bild 6.1:** Zusammenhang der Planungsräume

Varianten von Roboterbahnen

Im Wesentlichen existieren zwei Ansätze für die Vorgabe von Roboterbahnen, die nachstehend aufgeführt sind:

Point-to-Point-Bewegung (PTP)

Bei einer Point-to-Point- Bewegung wird vom Roboter erwartet, dass dieser in einer vorgegebenen (meist kürzest möglichen) Zeit von einem Startpunkt P_S zu einem Zielpunkt P_Z verfährt. Dies ist die übliche Bewegungsvorschrift für Trajektorien im Jointspace. Für den geometrischen Weg gibt es bei dieser Form der Bewegungsplanung keine weiteren Vorgaben. Die Bahn resultiert aus den begrenzten Geschwindigkeiten und Beschleunigungen der einzelnen Antriebe. Man vergleiche hierzu Bild 6.2(a).

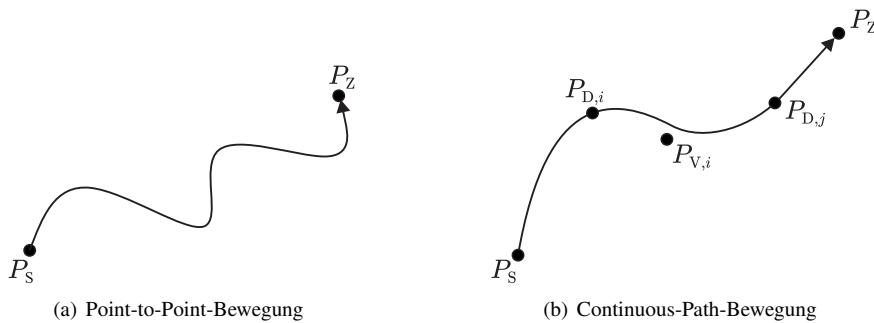
Definition
Point-to-Point-Bewegung

Continuous-Path-Bewegung (CP)

Bei der Continuous-Path-Bewegung soll der Roboter beim Übergang von einer Start- in eine Zielkonfiguration einer vorgegebenen geometrischen Bahn folgen. Der Verlauf kann aus beliebigen mathematischen Vorschriften für den kontinuierlichen geometrischen Übergang vom Start- zum Zielpunkt (Index: S bzw. Z) resultieren. Dies ist die übliche Vorgabe für Endeffektorbewegungen im Taskspace.

Definition
Continuous-Path-Bewegung

In Bild 6.2(b) ist der geometrische Weg einer beispielhaften Continuous-Path-Bewegung dargestellt, der aus einer Splineinterpolation mittels Durchgangspunkten (Index: D) und Viapunkten (Index: V) resultiert. Andere gängige CP-Befehle sind beispielsweise Linear- oder Kreisbewegungen im Taskspace.

**Bild 6.2:** Varianten von Roboterbahnen

Interpolator

Definition
Interpolator

Der Interpolator berechnet im Regeltakt der Steuerung Sollwerte für die Antriebe des Roboters (vgl. Bild 6.3). Hierfür muss die geplante Bahn als zeitdiskrete Funktion definiert werden. Je nach Planungsraum der Sollbahn findet hierbei mithilfe der inversen Kinematik eine Umrechnung der kartesischen Sollwerte in den Gelenkraum statt.

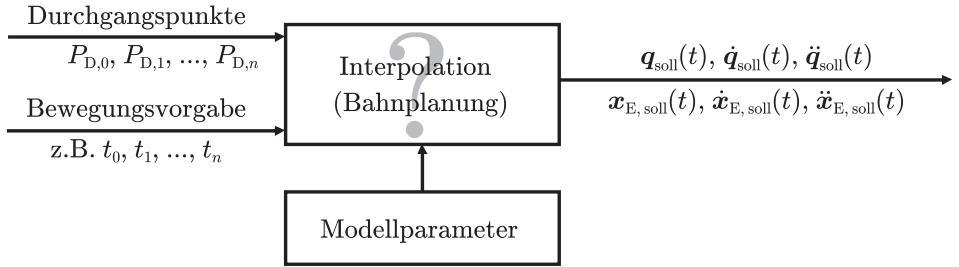


Bild 6.3: Bahnplanung durch Interpolation (zeitkontinuierlich)

Bahnkoordinate

Definition
Bahnkoordinate

Zur Darstellung eines gewünschten Verfahrweges als Raumkurve $\mathbf{x}(s)$ in parametrischer Form ist eine Bahnkoordinate (Laufkoordinate) s , normiert auf $0 \leq s \leq 1$, erforderlich. Diese beschreibt den Verlauf des Ortsvektors \mathbf{x} über den gewünschten Weg in einem Referenzkoordinatensystem. Die erste Ableitung der Raumkurve nach der Bahnkoordinate

$$\mathbf{x}'(s) = \frac{\partial \mathbf{x}(s)}{\partial s} \quad (6.1)$$

ergibt die Steigung, die zweite Ableitung

$$\mathbf{x}''(s) = \frac{\partial^2 \mathbf{x}(s)}{\partial s^2} \quad (6.2)$$

die Krümmung. Die Bahnkoordinate ist üblicherweise als zeitliche Funktion $s(t)$ gegeben, sodass aus der Vorschrift eine Trajektorie $\mathbf{x}(s(t))$ resultiert.

Der Vorteil dieser Darstellung findet sich in der Trennung der Beschreibung des geometrischen Weges vom zeitlichen Verlauf. Insbesondere bei komplexeren Funktionen $\mathbf{x}(s)$ kann so einfach auf den zeitlichen Ablauf Einfluss genommen werden, ohne den geometrischen Weg zu beeinflussen. Es ist allerdings zu beachten, dass die resultierende Beschleunigung des Endeffektors durch Zentripetaleffekte auf gekrümmten Wegsegmenten höher sein kann, als die vorgegebene Bahnbeschleunigung \ddot{s} . Für einen geometrischen Weg $\mathbf{x}(s)$ im kartesischen Taskspace ergibt sich die Endeffektorgeschwindigkeit $\dot{\mathbf{x}}(s(t))$ und -beschleunigung $\ddot{\mathbf{x}}(s(t))$ aus den zeitlichen Ableitungen der Trajektorienfunktion $\mathbf{x}(s(t))$ durch Anwendung der Kettenregel zu

$$\dot{\mathbf{x}}(s(t)) = \frac{\partial \mathbf{x}}{\partial s} \frac{ds}{dt} = \mathbf{x}' \dot{s}, \quad (6.3)$$

$$\ddot{\mathbf{x}}(s(t)) = \underbrace{\mathbf{x}'' \dot{s}^2}_{\text{Zentripetalbeschl.}} + \underbrace{\mathbf{x}' \ddot{s}}_{\text{Bahnbeschl.}}. \quad (6.4)$$

Handelt es sich bei der geplanten kartesischen Bahn um eine reine Rotation von einer Startorientierung definiert durch $(KS)_S$ zu einer Zielorientierung definiert durch $(KS)_Z$, gegeben durch die Rotationswinkel der zusammengesetzten Drehungen $\Phi_S = (\alpha_S, \beta_S, \gamma_S)^T$ und $\Phi_Z = (\alpha_Z, \beta_Z, \gamma_Z)^T$,

so geht die intuitive Interpretation der Bahnkoordinate verloren. Für reine Rotationen gelten lediglich die nachstehend formulierten Randbedingungen:

$$s(0) = 0 \quad \text{und} \quad s(t_e) = \|\Phi_Z - \Phi_S\|_2. \quad (6.5)$$

Auch eine Jointspacetrajektorie kann in Abhängigkeit der Bahnkoordinate $s(t)$ ausgedrückt werden. Um den kürzesten Verfahrweg zu realisieren, wird hierfür die Funktion für die Gelenkwinkel

$$\mathbf{q}(t) = \mathbf{q}(s(t)) = \mathbf{q}_S + s(t)(\mathbf{q}_Z - \mathbf{q}_S) \quad (6.6)$$

definiert. Es ist zu beachten, dass diese Darstellung eine normierte Bahnkoordinate erfordert, sodass $0 \leq s \leq 1$ gilt. Wie leicht zu erkennen ist, gilt für

$$s(0) = 0 : \quad \mathbf{q}(t) = \mathbf{q}_S, \quad (6.7)$$

$$s(t_e) = 1 : \quad \mathbf{q}(t) = \mathbf{q}_S + \mathbf{q}_Z - \mathbf{q}_S = \mathbf{q}_Z \quad (6.8)$$

Qualitativ sind die Verläufe von $q(t)$ von $s(t)$ identisch. Sie unterscheiden sich nach Gleichung (6.6) durch einen Offset und eine Skalierung, sodass die Verläufe unterschiedliche Eckwerte aufweisen.

6.2 Bahnplanung einer Einzelachsbewegung

Im Folgenden wird der zeitliche Verlauf $q(t)$ einer Einzelachsbewegung von q_S nach q_Z , vgl. Bild 6.4, betrachtet. Als Lösungsansatz dienen Profile mit stetiger Geschwindigkeit (Bild 6.4 mitte).

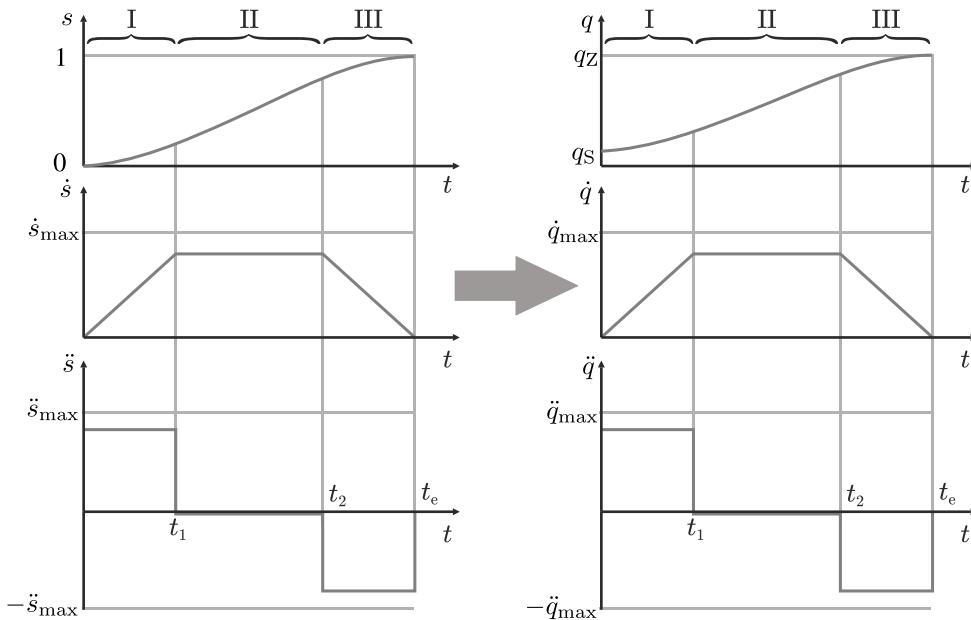


Bild 6.4: Typisches Geschwindigkeitstrapezprofil: Der zeitliche Verlauf der Gelenkwinkeltrajektorie $q(t)$ wird durch das Profil der Bahnkoordinate $s(t)$ festgelegt und unterscheidet sich lediglich durch eine unterschiedliche Skalierung und einen Offset. Qualitativ sind beide Verläufe identisch.

Es handelt sich hierbei um sogenannte C^1 -stetige Trajektorien des Gelenkwinkels q (stoßfrei), approximiert durch lineare Interpolation zwischen t_1 und t_2 mit quadratischen Übergängen

(Geschwindigkeitstrapez). Der Verlauf der Bahnkoordinate ist durch Gleichung (6.9) mit zunächst noch unbekannten Koeffizienten gegeben:

$$s(t) = \begin{cases} a_{01} + a_{11} t + a_{21} t^2, & 0 \leq t \leq t_1 \quad (\text{I}) \\ a_{02} + a_{12} t, & t_1 \leq t \leq t_2 \quad (\text{II}) \\ a_{03} + a_{13} t + a_{23} t^2, & t_2 \leq t \leq t_e \quad (\text{III}) \end{cases} \quad (6.9)$$

Als Ergebnis erhält man einen stetigen Positions- und Geschwindigkeitsverlauf, die Trajektorie ist stoßfrei. Der Beschleunigungsverlauf ist jedoch unstetig, die Trajektorie ist nicht ruckbegrenzt. Prinzipiell gilt auch hier, dass ein unstetiger Beschleunigungsverlauf zu einer hohen mechanischen Belastung führt und daher zu vermeiden ist.

Nachstehende Randbedingungen beschreiben einerseits die Leistungsgrenzen des Roboters und andererseits die geforderte Stetigkeit von Positions- und Geschwindigkeitsverlauf:

- Im Allgemeinen sind Start q_S und Ziel q_Z sowie die insgesamt zur Verfügung stehende Zeit t_e gegeben. Zu Beginn und am Ende der Bewegung soll sich der Roboter im Stillstand befinden:

$$\dot{q}_S = \dot{q}_Z = 0. \quad (6.10)$$

- Die Maximalgeschwindigkeit \dot{q}_{\max} wird durch die zulässigen Geschwindigkeiten des Getriebes und des Motors begrenzt.
- Die Maximalbeschleunigung \ddot{q}_{\max} ergibt sich aus den Maximalmomenten des Motors und des Getriebes. Bei Robotern besteht häufig kein linearer Zusammenhang zwischen Beschleunigung und Moment, bedingt durch diestellungsabhängigen Massenträgheiten der Gelenke. Diese Thematik wird in Kapitel 7 näher betrachtet. Daher gelten die Ausführungen in diesem Abschnitt streng genommen nur für Einzelachsen.
- Zur Bestimmung des Profils der Bahnkoordinate $s(t)$ werden die Randbedingungen durch die umgeformte, zeitliche Ableitung der Gleichung (6.6) in Bahnkoordinaten überführt:

$$\dot{s}_{\max} = \frac{\dot{q}_{\max}}{(q_Z - q_S)}, \quad (6.11)$$

$$\ddot{s}_{\max} = \frac{\ddot{q}_{\max}}{(q_Z - q_S)}. \quad (6.12)$$

- Darüber hinaus existiert noch der geforderte C^1 -stetige Verlauf der Trajektorie für t_1 und t_2 ($i = 1, 2$):

$$\begin{aligned} \lim_{t \rightarrow t_i^+} s(t) &= \lim_{t \rightarrow t_i^-} s(t), \\ \lim_{t \rightarrow t_i^+} \dot{s}(t) &= \lim_{t \rightarrow t_i^-} \dot{s}(t), \end{aligned} \quad (6.13)$$

wobei t_i^+ für den rechtsseitigen und t_i^- für den linksseitigen Grenzwert von t_i stehen. Die Zeiten t_1 und t_2 seien entweder gegeben oder resultieren aus der maximalen Geschwindigkeit bzw. Beschleunigung des Roboters.

Ziel ist demnach die Bestimmung der Koeffizienten der Polynome, aus denen sich die Bahnkoordinate $s(t)$ zusammensetzt, damit der zeitliche Verlauf von $q(t)$ bestimmt wird.



Bang-Bang-Steuerung

Die Bang-Bang-Steuerung ist eine spezielle Form der linearen Interpolation mit quadratischen Übergängen (Geschwindigkeitstrapezprofil) mit dem Ziel, die Trajektorienzeit t_e zu minimieren.

Definition
Bang-Bang-
Steuerung

Bei der Bang-Bang-Steuerung erfolgt die Bewegung stets mit maximal möglicher Beschleunigung, solange bis die maximale Geschwindigkeit erreicht ist (Bild 6.5(a)). Dabei kann es vorkommen, dass die Maximalgeschwindigkeit aufgrund einer zu kurzen Beschleunigungsphase nicht erreicht wird (Bild 6.5(b)).

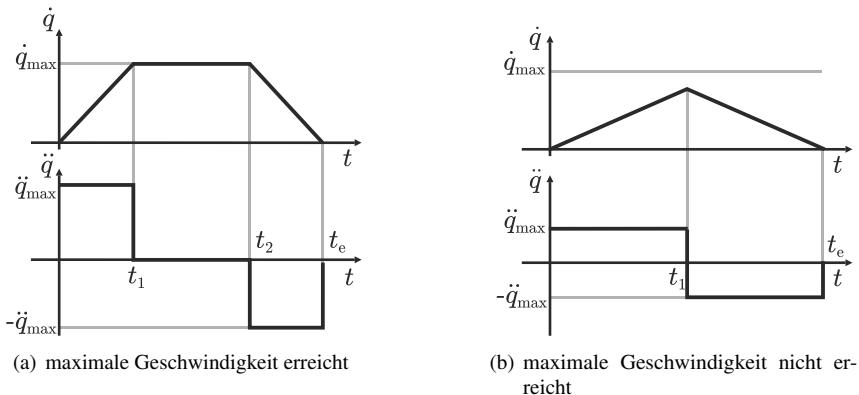


Bild 6.5: Bang-Bang-Steuerung

Point-to-Point-Interpolation – C^2 -stetige Trajektorie

Zur Vermeidung von Sprüngen im Beschleunigungsverlauf (stoß- und ruckfrei¹) kommen C^2 -stetige Trajektorien zum Einsatz. Diese erhält man durch ein Beschleunigungstrapezprofil. Daraus resultiert eine Erhöhung der Ordnung der beschreibenden Polynome:

$$s(t) = \begin{cases} a_{01} + a_{11} t + a_{21} t^2 + a_{31} t^3, & 0 \leq t \leq t_1 & (\text{I}) \\ a_{02} + a_{12} t + a_{22} t^2, & t_1 \leq t \leq t_2 & (\text{II}) \\ a_{03} + a_{13} t + a_{23} t^2 + a_{33} t^3, & t_2 \leq t \leq t_3 & (\text{III}) \\ a_{04} + a_{14} t, & t_3 \leq t \leq t_4 & (\text{IV}) \\ \dots \end{cases} \quad (6.14)$$

Der Geschwindigkeits- und Beschleunigungsverlauf ist Bild 6.6 zu entnehmen. Es ergeben sich stetige Verläufe für Position, Geschwindigkeit und Beschleunigung. Insgesamt sind sieben Polynome notwendig, um die Bewegung von einem Startpunkt q_S zu einem Zielpunkt q_Z zu berechnen. Die Berechnung der a priori unbekannten Polynom-Koeffizienten erfolgt analog zum Geschwindigkeitstrapezprofil (vgl. Gleichung (6.9))

Gegenüberstellung von C^1 - (stoßfrei) und C^2 -Stetigkeit (stoß- und ruckfrei)

In Bild 6.7 sind die Unstetigkeiten im Beschleunigungsverlauf der C^1 -stetigen Sollbahn deutlich erkennbar. Daraus resultiert das hohe Motormoment, das einerseits hohe Dynamik zur Folge hat (kurze Trajektorienzeit), andererseits jedoch die Komponenten des Roboters stark beansprucht und ggf. hochfrequente Schwingungen anregt.

¹Physikalisch betrachtet handelt es sich hierbei um ruckbegrenzte Trajektorien, die in der Literatur allerdings häufig als ruckfrei bezeichnet werden.

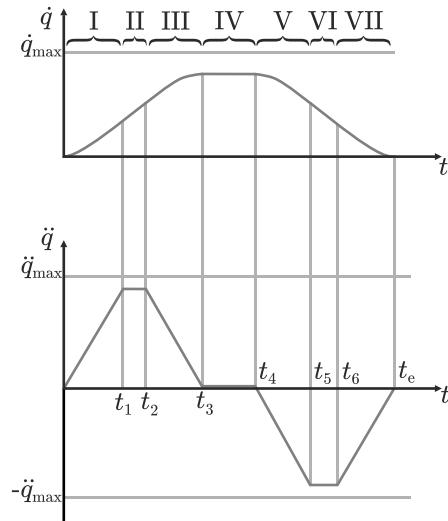


Bild 6.6: Geschwindigkeits- und Beschleunigungsverlauf einer C^2 -stetigen Trajektorie aus zusammengesetzten Polynomen

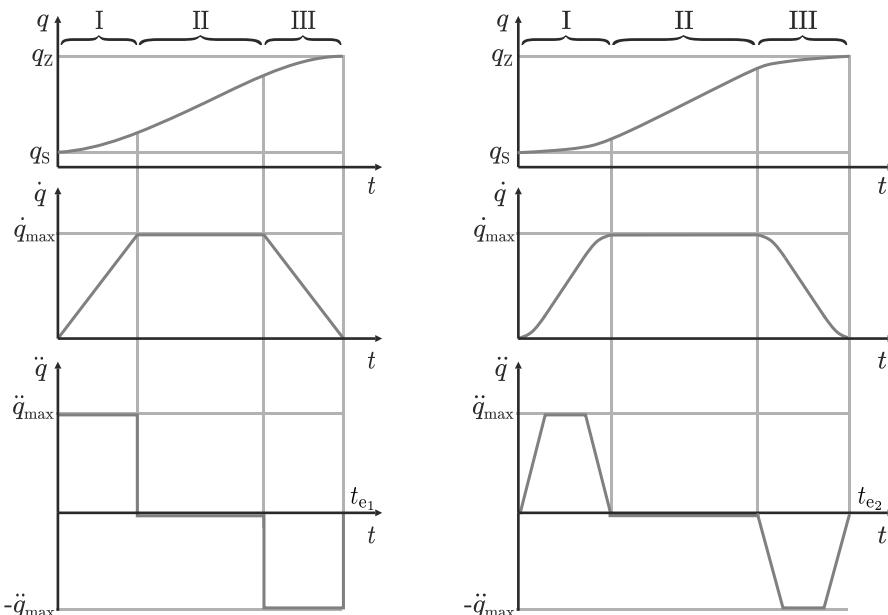
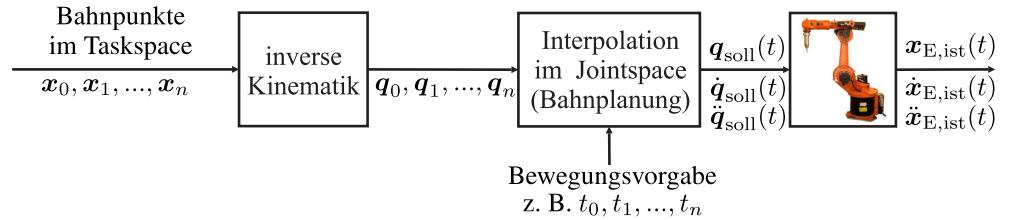


Bild 6.7: Gegenüberstellung von C^1 - und C^2 -Stetigkeit: Positions-, Geschwindigkeits- und Beschleunigungsprofil (im Allgemeinen gilt: $t_{e2} > t_{e1}$)

6.3 Bahnplanung im Jointspace

Bei der Bahnplanung im Jointspace werden zunächst, falls erforderlich, die einzelnen Bahnpunkte x_i unter Verwendung der inversen Kinematik in den Jointspace transformiert. Die resultierenden Gelenkwinkel q_i bilden die Grundlage für die Berechnung der Sollbahn, Sollgeschwindigkeit und Sollbeschleunigung der Gelenke und die anschließende Interpolation. Vorteil hierbei ist, dass die aufwendige Berechnung der inversen Kinematik nicht im Regeltakt der Steuerung erfolgen muss. Der prinzipielle Ablauf ist in Bild 6.8 nochmals grafisch dargestellt.

Problematisch ist die Berücksichtigung von Hindernissen, falls die Bahnplanung im Jointspace durchgeführt wird, da im Allgemeinen Hindernisse im Taskspace gegeben sind. Diese müssen ebenfalls in den Gelenkwinkelraum des Roboters transformiert werden, um dort die Kollisionsver-

**Bild 6.8:** Klassischer Ablauf der Bahnplanung im Jointspace

meidung bereits während der Bahnplanung durchführen zu können. Die berechnete Roboterbahn kann aber auch in einem zweiten Schritt im Taskspace des Roboters auf Kollisionen geprüft und ggf. angepasst werden.

Synchronisation auf die „langsamste Gelenkbewegung“

Für einen Roboter mit n Freiheitsgraden kann die Jointspace Trajektorie entsprechend Gleichung (6.6) mithilfe einer Bahnkoordinate $s(t)$ ausgedrückt werden. Hierbei wird jedoch lediglich ein Kompromiss hinsichtlich der unterschiedlichen Antriebsleistungen gefunden, der nicht immer die zeitoptimale Lösung darstellt. Um diese zu finden, muss zunächst für jedes Gelenk k ein individuelles Bewegungsprofil $q_k(t)$ berechnet werden. Um eine gleichzeitige Bewegung sämtlicher Antriebe zu erzielen, erfolgt anschließend eine Synchronisation auf das „langsamste Gelenk“ des Roboters. Hierzu dient folgende Vorgehensweise:

1. Bestimmung von t_{e_k} für jedes Gelenk k ,
2. Bestimmung der maximalen Trajektorienzeit aller Gelenke: $t_{\max} = \max_k t_{e_k}$,
3. Festsetzung der Trajektorienzeit aller Gelenke basierend auf der Maximalzeit,
4. (Neu-)Berechnung sämtlicher Trajektorien.

Diese Synchronisation, dargestellt in Bild 6.9, ist insbesonders beim Einsatz von Robotern an Produktionsstraßen sinnvoll. Der Durchsatz einer Produktion hängt von der langsamsten Achsbewegung ab. Eine unnötig starke Beschleunigung der übrigen Achsen hat demnach keinen Vorteil in Bezug auf die produzierte Menge. Vielmehr kann die beschriebene Synchronisation zu Energieeinsparungen führen, da die Motoren betroffener Achsen bei vergleichsweise geringer Energieaufnahme betrieben werden können, allerdings wird durch die gleichzeitige Beschleunigungs- und Abbremsphase die Möglichkeit vom Energieaustausch über einen DC-Zwischenkreis „verschenkt“.

Wie in Abschnitt 6.2 erwähnt, hängen die maximal zulässigen Beschleunigungen in der Regel von der Roboterkonfiguration ab. Zur vollständigen Ausnutzung der dynamischen Grenzen könnten demnach nicht die maximale Beschleunigungen der Achsen, sondern die maximalen zur Verfügung stehenden Antriebsmomente zur Trajektorienplanung verwendet werden. Eine solcher Planungsansatz ist jedoch deutlich komplexer und wird zum Beispiel in [CC00] vollständig erläutert.

6.4 Bahnplanung von translatorischen Bewegungen im Taskspace

Die Planung einer Bewegung im Taskspace erfordert neben der Einhaltung von Randbedingungen wie limitierten Geschwindigkeiten und Beschleunigungen meist auch die Vorgabe des geometrischen Weges, den der Endeffektor zurücklegen soll (vgl. Abschnitt 6.1). Eine besondere Herausforderung ist dabei, dass im Fall einer Online-Bahnplanung nun die inverse Kinematik

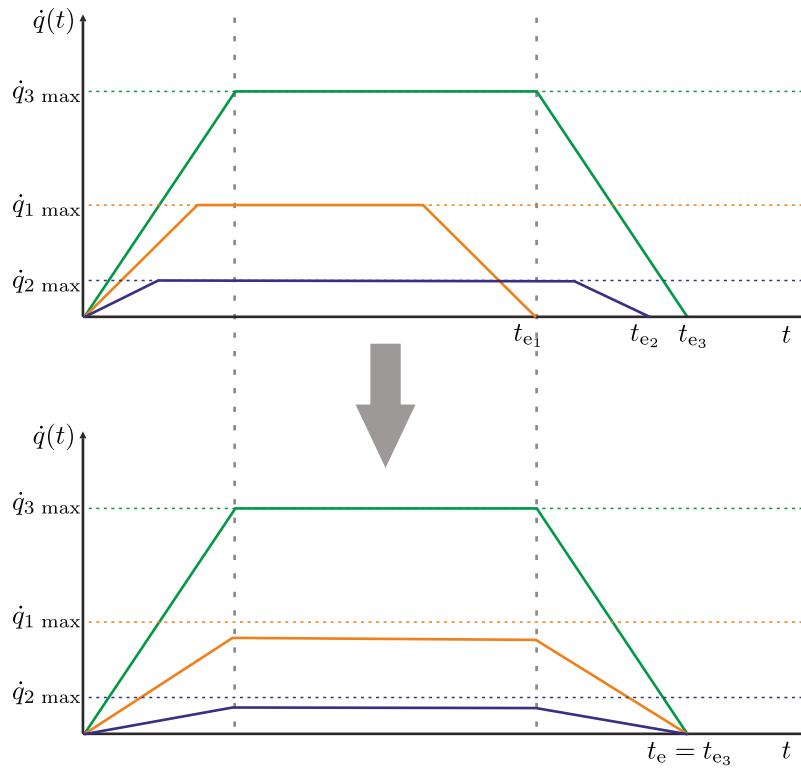


Bild 6.9: Geschwindigkeitsverlauf eines 3-Achs-Roboters vor und nach der Synchronisation auf die „langsamste Gelenkbewegung“, im dargestellten Fall ist dies die Achse 3

im Regeltakt des Roboters (in der Regel 1 ms) benötigt wird (siehe Ablauf in Bild 6.10). Dies erfordert eine erheblich höhere Rechenleistung.

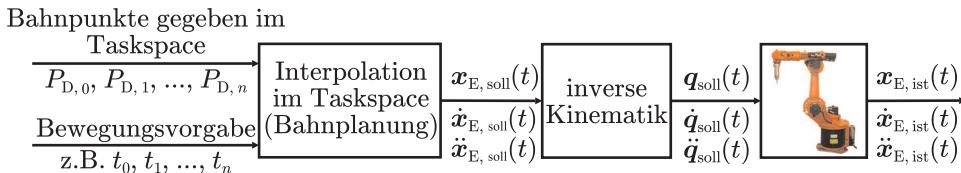


Bild 6.10: Klassischer Ablauf der Bahnplanung im Taskspace

Zusammengesetzte Taskspacetrajektorien aus Geometrieprimiven

Bei Prozessen wie Kleben oder Bahnschweißen soll ein Roboter sein Werkzeug meist entlang einer Geraden und/oder eines Kreissegments führen, daher ist die Beschreibung des Weges anhand geometrischer Primitiven sinnvoll.

Linearsegment

Die einfachste Form der kartesischen Weggeometrie ist die Bewegung des Endeffektors auf einer Geraden. Hierbei wird analog zu Gleichung (6.6) der Übergang von einer Startposition $x_{E,S}$ in eine Zielposition $x_{E,Z}$ durch die Funktion

$$x_E(t) = x_{E,S} + s(t) (x_{E,Z} - x_{E,S}) \quad (6.15)$$

definiert. Dabei ist zu beachten, dass die Bahnkoordinate wie zuvor auf $0 \leq s(t) \leq 1$ normiert ist, um eine intuitive Definition der Geradenfunktion mithilfe der Start- und Zielkoordinaten zu ermöglichen.

Kreisbahn

Eine weitere übliche Basisgeometrie für die kartesische Bahnplanung sind Kreissegmente. Die Funktion soll auch in diesem Fall mithilfe der Bahnkoordinate $s(t)$ interpoliert werden. Die Parametrierung einer solchen Bahn kann auf verschiedene Weisen erfolgen. Üblich ist die Kreisdefinition mithilfe von drei Punkten, vgl. Bild 6.11. Meist ergibt sich der Startpunkt P_1 aus der aktuellen Endeffektorposition oder dem Zielpunkt des Vorgängersegments. Wird zusätzlich ein Zielpunkt P_3 sowie ein Punkt auf der gewünschten Kreisbahn P_2 festgelegt, definieren die drei Punkte ein Dreieck, dessen Umkreis dem vorgegebenen Kreisbogen entspricht. Der Kreismittelpunkt M kann aus den Schnittpunkten der Mittelsenkrechten des Dreiecks ermittelt werden. Eine einfache Darstellung der Kreisbahn ergibt sich, wenn ein lokales Koordinatensystem $(KS)_M$ in diesen Kreismittelpunkt gelegt wird, dessen x -Achse auf den Startpunkt zeigt und dessen z -Achse orthogonal auf der Kreisebene steht, sodass sich die Bewegungsrichtung der Kreisbahn im mathematisch positiven Sinn um diese z -Achse darstellen lässt.

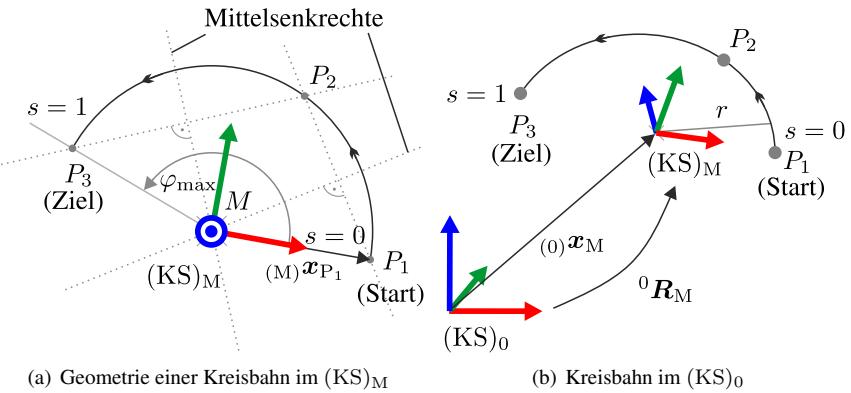


Bild 6.11: Kreisbahn

Wird die Orientierung des $(KS)_M$ durch die Rotationsmatrix ${}^0\mathbf{R}_M$ ausgedrückt, ergibt sich dann für die Kreisbahn folglich der funktionale Zusammenhang

$${}^{(0)}\mathbf{x}_E(t) = {}^{(0)}\mathbf{x}_M + {}^0\mathbf{R}_M \begin{pmatrix} \cos(s(t) \varphi_{\max}) \\ \sin(s(t) \varphi_{\max}) \\ 0 \end{pmatrix} \underbrace{\| {}^{(M)}\mathbf{x}_{P_1} - {}^{(M)}\mathbf{x}_M \|_2}_{=r} \quad (6.16)$$

über den Kreiswinkel φ_{\max} .

Anmerkung

Da bei kartesischen Bahnen die maximale Bahngeschwindigkeit und -beschleunigung des Endeffektors vorgegeben werden, muss zur Berechnung des Zeitprofils der Bahnkoordinate $s(t)$, $\dot{s}(t)$, $\ddot{s}(t)$ das Bewegungsprofil auf den zurückgelegten Weg (in diesem Fall den Umfang des Kreisbogens $\ell = r \varphi_{\max}$) normiert werden.

Verschleifen von Bewegungssegmenten

Besteht eine kartesische Bahn aus mehreren Abschnitten (Segmenten), muss in der Regel an den Übergängen ein Genauhalt, d. h. $\dot{s} = 0$, ausgeführt werden, da die Übergänge nicht stetig

differenzierbar sind. Besonders deutlich wird dies am Beispiel zweier Linearsegmente mit unterschiedlichen Richtungsvektoren (vgl. Bild 6.12). Ohne Stillstand im Segmentübergang würde die plötzliche Änderung der Bewegungsrichtung am Übergang trotz eines stetigen Zeitprofils $s(t)$ eine unendlich hohe Beschleunigung \ddot{x}_E des Endeffektors erfordern. Da in der Praxis ein exaktes Durchfahren der Bahnpunkte oft nicht zwingend erforderlich ist, kann diesem Problem mithilfe von Übergangssegmenten, sogenannten Verschliffsegmenten, entgegengewirkt werden. Hierbei wird ein bestimmter Toleranzradius um die definierten Bahnpunkte der einzelnen Bewegungssegmente gelegt, innerhalb dessen von der Bahngeometrie abgewichen werden darf. Wird nun ein entsprechendes Übergangssegment berechnet, das jeweils mindestens C^1 -stetig an die bestehenden Bahnsegmente anschließt, kann der Übergang ohne Halt durchfahren werden. Das Verschleifen wird häufig zur Verkürzung der Gesamtverfahrzeit genutzt, ist jedoch auch bei Prozessen von Vorteil, die eine kontinuierliche Bahngeschwindigkeit erfordern (z.B. Schweißen oder Kleben). Im Folgenden wird die Berechnung eines solchen Übergangssegments am Beispiel zweier Linearsegmente dargestellt.

Verschliff zweier Linearsegmente Im hier betrachteten Beispiel besteht die programmierte Roboterbahn aus zwei Geraden in der x - y -Ebene, deren Bahnpunkte $\mathbf{x}_{E,i} = (x_{E,i}, y_{E,i})^T$ durch lineare Funktionen verbunden sind. Die zugehörigen Funktionen lauten

Beispiel

$$\mathbf{x}_{E,i}(s) = \mathbf{a}_{0,i} + \mathbf{a}_{1,i} s \quad (6.17)$$

mit den beispielhaften Koeffizienten

$$\mathbf{a}_{0,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{a}_{1,1} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}, \quad \mathbf{a}_{0,2} = \begin{pmatrix} 1 \\ 4 \end{pmatrix} \quad \text{und} \quad \mathbf{a}_{1,2} = \begin{pmatrix} 4 \\ 0 \end{pmatrix}. \quad (6.18)$$

Beim Übergang von einer Geraden zur folgenden soll nun ein Verschliffsegment eingefügt werden (vgl. Bild 6.13). Für den Verschliff werden 25% der jeweiligen Segmentlängen freigegeben, sodass sich die beiden Geradensegmente entsprechend Bild 6.13 verkürzen. Hierdurch ändern sich die Koeffizienten zu

$$\mathbf{a}_{0,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{a}_{1,1} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \quad \mathbf{a}_{0,2} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \quad \mathbf{a}_{1,2} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}. \quad (6.19)$$

Zur Beschreibung des Verschliffsegments V soll ein kubisches Polynom zum Einsatz kommen:

$$\mathbf{x}_{E,V}(s) = \mathbf{a}_{0,V} + \mathbf{a}_{1,V} s + \mathbf{a}_{2,V} s^2 + \mathbf{a}_{3,V} s^3. \quad (6.20)$$

Für einen stetigen Übergang in das vorhergehende sowie nachfolgende Segment müssen entsprechende Randbedingungen erfüllt werden:

$$\mathbf{x}_{E,V}(0) = \mathbf{x}_{E,1}(1) = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \mathbf{x}_{E,V}(1) = \mathbf{x}_{E,2}(0) = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \quad (6.21)$$

$$\mathbf{x}'_{E,V}(0) = \mathbf{x}'_{E,1}(1) = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \quad \mathbf{x}'_{E,V}(1) = \mathbf{x}'_{E,2}(0) = \begin{pmatrix} 3 \\ 0 \end{pmatrix}. \quad (6.22)$$

In diesem Fall ergeben sich die Koeffizienten zu

$$\mathbf{a}_{0,V} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \mathbf{a}_{1,V} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \quad \mathbf{a}_{2,V} = \begin{pmatrix} 0 \\ -3 \end{pmatrix}, \quad \mathbf{a}_{3,V} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (6.23)$$

und es resultiert der in Bild 6.13 dargestellte Verlauf. Es sei darauf hingewiesen, dass in Bild 6.12 und Bild 6.13 die Bahnkoordinaten in jedem Segment jeweils von 0 bis 1 laufen.

An dieser Stelle gibt es noch eine Besonderheit zu beachten. Analog zu den oben vorgestellten Segmenttypen soll auch auf dem Polynomsegment die parametrisierte Endeffektorgeschwindigkeit und -beschleunigung eingehalten werden. Hierfür muss analog die Bogenlänge des Polynomsegments bestimmt werden. Für ein beliebiges dreidimensionales Polynom ergibt sich die Bogenlänge in Abhängigkeit der Bahnkoordinate s zu

$$\ell(s) = \int_0^s \sqrt{(x'(s))^2 + (y'(s))^2 + (z'(s))^2} ds. \quad (6.24)$$

Leider ist dieses elliptische Integral für Polynome der Ordnung ≤ 3 nicht mehr geschlossen lösbar. Daher muss die Bogenlänge numerisch ermittelt werden, beispielsweise durch eine Näherung mittels Aufsummieren.

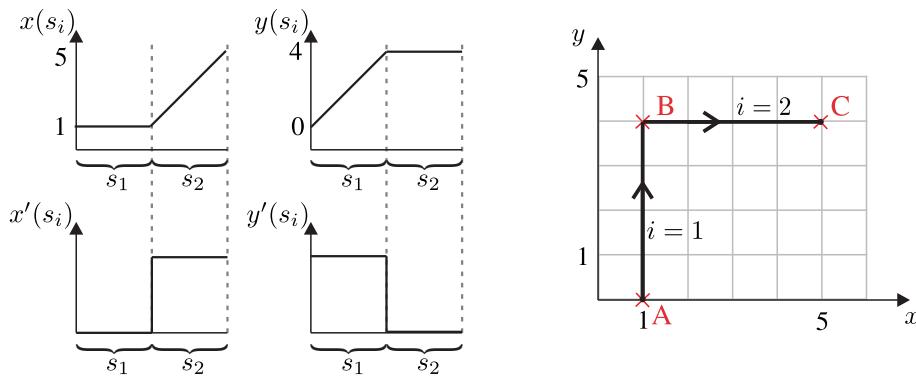


Bild 6.12: Unverschliffene Bahn

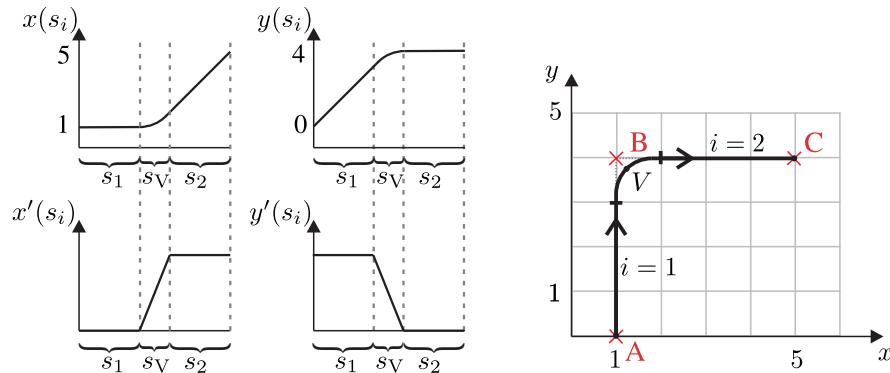


Bild 6.13: Verschliffene Bahn

6.5 Bahnplanung von rotatorischen Bewegungen im Taskspace

Die Positionen im Taskspace können problemlos unter Verwendung der vorgestellten Methoden interpoliert werden. Bei den Orientierungen hingegen ist Folgendes zu beachten: ${}^0\mathbf{R}_S$ und ${}^0\mathbf{R}_Z$ sind die Rotationsmatrizen, welche die Einheitsvektoren vom $(KS)_S$ bzw. $(KS)_Z$ im $(KS)_0$ darstellen (in Bild 6.14 sind die zugehörigen homogenen Transformationsmatrizen angegeben). Die nächstliegende Lösung wäre an dieser Stelle eine Interpolation der Einheitsvektoren. Dies stellt jedoch keine Orthogonalität der aktuellen Rotationsmatrix zu jedem Abtastzeitpunkt t sicher.

Im Folgenden werden drei Ansätze vorgestellt, mit denen die Interpolation von Orientierungen unter Gewährleistung einer Rotationsmatrix, die Element der speziellen orthogonalen Gruppe ist, möglich ist:

Sinnvolle Ansätze:

1. Interpolation der Rotationswinkel der zusammengesetzten Drehungen,
2. Drehung um eine äquivalente Achse mit einem bestimmten Winkel,
3. Interpolation unter Verwendung von Quaternionen (LERP, SLERP).

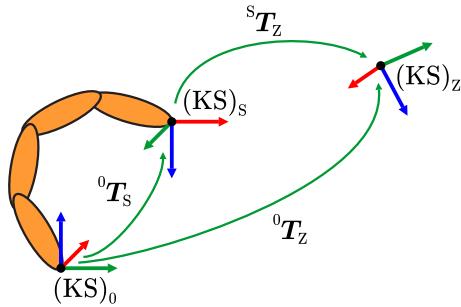


Bild 6.14: Orientierungsinterpolation

Im Folgenden werden die drei erwähnten Ansätze zur Interpolation von Orientierungen detailliert beschrieben.

6.5.1 Interpolation der Rotationswinkel zusammengesetzter Drehungen

Die Interpolation von Rotationswinkel zusammengesetzter Drehungen ist abhängig von der gewählten Darstellung (z. B. Euler, Kardan, etc., vgl. Kapitel 2). Die nachstehenden Erläuterungen erfolgen beispielhaft anhand der Kardan-Winkel, lassen sich jedoch prinzipiell auch auf andere Repräsentationen übertragen. Für die Kardan-Winkel $\Phi_i = (\alpha_i, \beta_i, \gamma_i)^T$ gelte (vgl. Kapitel 2):

$$\begin{aligned}\alpha_i &= \arctan2(-{}^0t_{i,(2,3)}, {}^0t_{i,(3,3)}), \\ \beta_i &= \arctan2({}^0t_{i,(1,3)}, {}^0t_{i,(1,1)} c_\gamma - {}^0t_{i,(1,2)} s_\gamma), \\ \gamma_i &= \arctan2(-{}^0t_{i,(1,2)}, {}^0t_{i,(1,1)}).\end{aligned}\quad (6.25)$$

Nachdem die Rotationswinkel mittels homogener Transformationsmatrizen berechnet sind, kann basierend auf deren Werten eine Interpolation durchgeführt werden. Unter Verwendung einer Bahnkoordinate $s(t)$ folgt für die Rotationswinkel, -geschwindigkeit und -beschleunigung zum Zeitpunkt t (siehe [SS00, S. 192ff] für Details):

$$\Phi(t) = \Phi_S + s(t)(\Phi_Z - \Phi_S), \quad (6.26)$$

$$\dot{\Phi}(t) = \dot{s}(t)(\Phi_Z - \Phi_S), \quad (6.27)$$

$$\ddot{\Phi}(t) = \ddot{s}(t)(\Phi_Z - \Phi_S). \quad (6.28)$$

Aufgrund der Tatsache, dass die Interpolation entlang eines mitgedrehten Koordinatensystems erfolgt, ist die entstehende Bahn intuitiv nur schwer vorstellbar.

6.5.2 Drehung um äquivalente Achse mit einem bestimmten Winkel

Wie bereits in Abschnitt 2.5.1 gezeigt, können Rotationen auch durch eine Drehung um eine äquivalente Drehachse mit einem Drehwinkel beschrieben werden. Diese Darstellung wird nun

verwendet, um zwischen zwei Orientierungen zu interpolieren. Hierzu ist zunächst die Transformationsmatrix zwischen dem Start- und dem Zielkoordinatensystem zu bestimmen (vgl. Bild 6.15). Aus der relativen homogenen Transformation

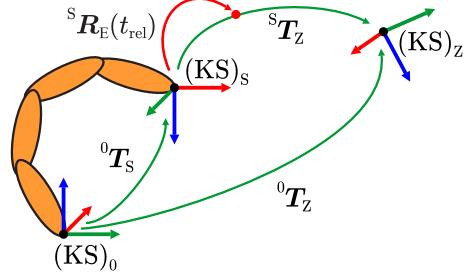


Bild 6.15: Interpolation von Orientierungen mittels Drehung um eine äquivalente Achse

$${}^sT_Z = ({}^0T_S)^{-1} {}^0T_Z = {}^sT_0 {}^0T_Z \quad (6.29)$$

kann die Rotation sR_Z einfach abgelesen werden.

Die Rotation sR_Z kann in eine Drehung um eine normierte Achse \mathbf{u} mit dem Rotationswinkel θ umgerechnet werden. Die Rotationsachse berechnet sich nach Abschnitt 2.5.1 zu:

$$\mathbf{u} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \frac{1}{2 \sin(\theta)} \begin{pmatrix} {}^s r_{Z,(3,2)} - {}^s r_{Z,(2,3)} \\ {}^s r_{Z,(1,3)} - {}^s r_{Z,(3,1)} \\ {}^s r_{Z,(2,1)} - {}^s r_{Z,(1,2)} \end{pmatrix}. \quad (6.30)$$

Der zugehörige Rotationswinkel lautet:

$$\theta = \text{arctan2}(\sin(\theta), \cos(\theta)), \quad (6.31)$$

mit

$$\begin{aligned} \sin(\theta) &= \\ &\pm \frac{1}{2} \sqrt{({}^s r_{Z,(3,2)} - {}^s r_{Z,(2,3)})^2 + ({}^s r_{Z,(1,3)} - {}^s r_{Z,(3,1)})^2 + ({}^s r_{Z,(2,1)} - {}^s r_{Z,(1,2)})^2} \end{aligned} \quad (6.32)$$

und

$$\cos(\theta) = \frac{1}{2} ({}^s r_{Z,(1,1)} + {}^s r_{Z,(2,2)} + {}^s r_{Z,(3,3)} - 1). \quad (6.33)$$

Die Achse, repräsentiert durch \mathbf{u} , ist demnach während der Drehung konstant, wobei der Rotationswinkel zwischen Start und Ziel, bzw. zwischen 0 und θ , interpoliert wird.

Die aktuelle Endeffektororientierung 0R_E relativ zum $(KS)_0$ kann nun als Funktion der relativen Trajektorienzeit $t_{\text{rel}} = t/t_e \in [0, 1]$ berechnet werden:

$${}^0R_E(t_{\text{rel}}) = {}^0R_S {}^sR_E(\mathbf{u}, t_{\text{rel}} \theta). \quad (6.34)$$

Zu Beginn der Interpolation gilt:

$${}^sR_E(t_{\text{rel}} = 0) = \mathbf{E} \quad \Rightarrow \quad {}^0R_E = {}^0R_S. \quad (6.35)$$

Am Ende der Interpolation gilt:

$${}^S\mathbf{R}_E(t_{\text{rel}} = 1) = {}^S\mathbf{R}_Z \Rightarrow {}^0\mathbf{R}_E = {}^0\mathbf{R}_Z. \quad (6.36)$$

Die Umrechnung der Achse-Winkel-Darstellung der interpolierten Rotation geschieht nach der aus Abschnitt 2.5.1 bekannten Formel, wobei θ durch den interpolierten Winkel $\theta_{\text{rel}} = t_{\text{rel}} \theta$ ersetzt wird:

$$\begin{aligned} {}^S\mathbf{R}_E(\mathbf{u}, \theta_{\text{rel}}) = & \\ \begin{pmatrix} u_x^2(1 - c_{\theta_{\text{rel}}}) + c_{\theta_{\text{rel}}} & u_x u_y(1 - c_{\theta_{\text{rel}}}) - u_z s_{\theta_{\text{rel}}} & u_x u_z(1 - c_{\theta_{\text{rel}}}) + u_y s_{\theta_{\text{rel}}} \\ u_x u_y(1 - c_{\theta_{\text{rel}}}) + u_z s_{\theta_{\text{rel}}} & u_y^2(1 - c_{\theta_{\text{rel}}}) + c_{\theta_{\text{rel}}} & u_y u_z(1 - c_{\theta_{\text{rel}}}) - u_x s_{\theta_{\text{rel}}} \\ u_x u_z(1 - c_{\theta_{\text{rel}}}) - u_y s_{\theta_{\text{rel}}} & u_y u_z(1 - c_{\theta_{\text{rel}}}) + u_x s_{\theta_{\text{rel}}} & u_z^2(1 - c_{\theta_{\text{rel}}}) + c_{\theta_{\text{rel}}} \end{pmatrix}. \end{aligned} \quad (6.37)$$

Zu beachten ist, dass die relative Trajektorienzeit t_{rel} ausschließlich Werte im Intervall $[0, 1]$ annimmt und somit gleichwertig zur vorher definierten Bahnkoordinate $s(t)$ ist.

6.5.3 Interpolation von Quaternionen

Für die Interpolation von Quaternionen werden zwei gängige Verfahren vorgestellt: die lineare Quaternionen-Interpolation (LERP) und die sphärische lineare Interpolation (SLERP).

Lineare Quaternionen Interpolation (LERP)

Die LERP-Funktion berechnet für $0 \leq t_{\text{rel}} \leq 1$ die kürzeste Verbindung (Gerade) durch die vierdimensionale Einheitssphäre zwischen zwei Orientierungen, gegeben durch die Quaternionen \mathbf{q}_S und \mathbf{q}_Z (siehe Bild 6.16(a)). Resultat ist eine Rotation mit variabler Geschwindigkeit um eine feste Rotationsachse \mathbf{u} .

Die Interpolation nach LERP berechnet sich zu:

$$\text{Lerp}(\mathbf{q}_S, \mathbf{q}_Z, t_{\text{rel}}) = \mathbf{q}_S + t_{\text{rel}} (\mathbf{q}_Z - \mathbf{q}_S). \quad (6.38)$$

Der grün eingezeichnete Pfad in Bild 6.16(a) ist das Ergebnis der linearen Interpolation nach Gleichung (6.38) und führt durch die vierdimensionale Einheitssphäre hindurch. Nach Normierung der interpolierten Quaternionen auf eins (rote Pfeile in Bild 6.16(a)) liegen diese wieder auf der Einheitssphäre.

Die anschließende Normierung des Betrags des interpolierten Quaternions auf eins (Einheitsquaternionen) führt jedoch zu Konsequenzen, die in der Praxis problematisch sein können. Aufgrund der Interpolation durch und nicht entlang der vierdimensionalen Einheitssphäre resultiert eine variable Rotationsgeschwindigkeit (größere Geschwindigkeit in der Mitte, vgl. Bild 6.16(a)). Dies ist in der Regel unerwünscht, was die Notwendigkeit alternativer Ansätze begründet.

Sphärische lineare Interpolation (SLERP)

Die SLERP-Funktion berechnet für $0 \leq t_{\text{rel}} \leq 1$ die kürzeste Verbindung auf der vierdimensionalen Einheitssphäre (vgl. Bild 6.16(b)). Resultat ist eine Rotation mit konstanter Geschwindigkeit um eine feste Rotationsachse \mathbf{u} .

Es existieren vier äquivalente Ansätze der Interpolation (ohne Herleitung):

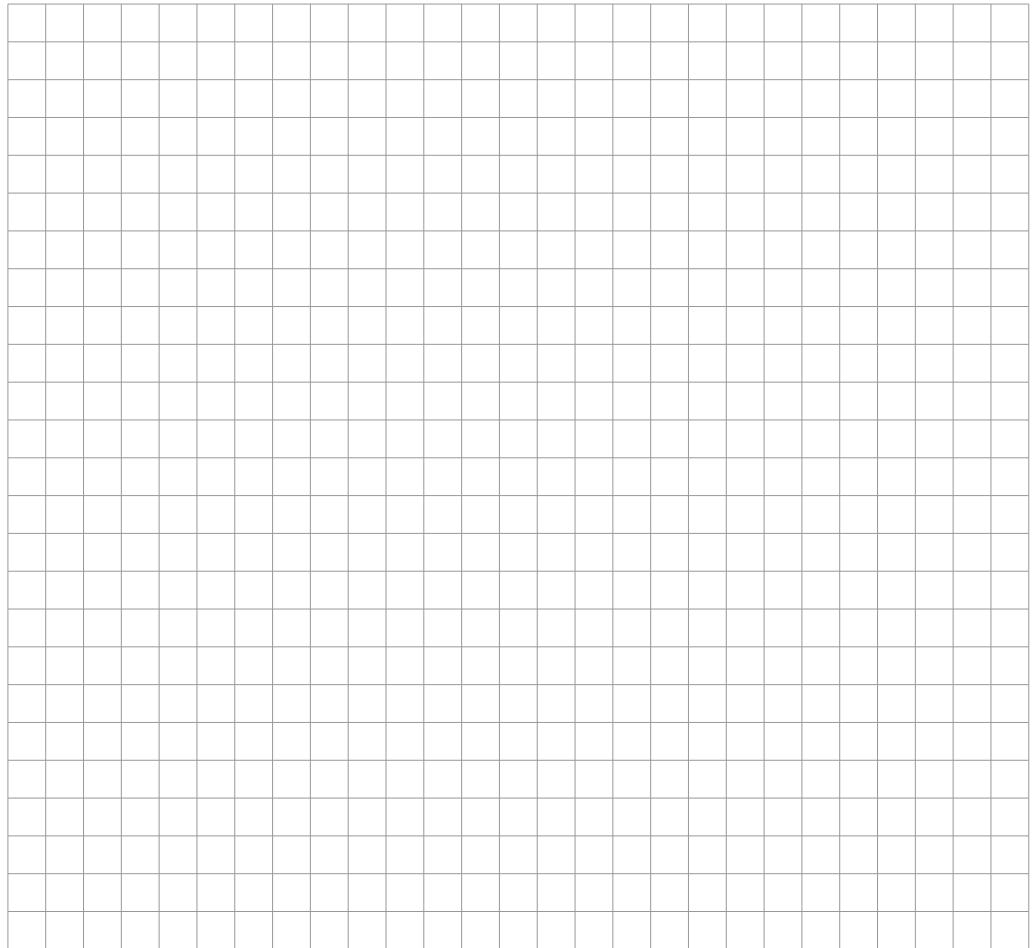
$$\begin{aligned} \text{Slerp}(\mathbf{q}_S, \mathbf{q}_Z, t_{\text{rel}}) &= \mathbf{q}_S (\mathbf{q}_S^{-1} \mathbf{q}_Z)^{t_{\text{rel}}} = (\mathbf{q}_S \mathbf{q}_Z^{-1})^{1-t_{\text{rel}}} \mathbf{q}_Z \\ &= (\mathbf{q}_Z \mathbf{q}_S^{-1})^{t_{\text{rel}}} \mathbf{q}_S = \mathbf{q}_Z (\mathbf{q}_Z^{-1} \mathbf{q}_S)^{1-t_{\text{rel}}}, \end{aligned} \quad (6.39)$$

mit:

$$\mathbf{q}^k = e^{k \log(\mathbf{q})} = \left(\cos \left(k \frac{\theta}{2} \right), \mathbf{v}^T \sin \left(k \frac{\theta}{2} \right) \right)^T, \quad (6.40)$$

wobei

$$\log(\mathbf{q}) = \left(0, \frac{\theta}{2} \mathbf{u}^T \right)^T. \quad (6.41)$$



Zusammengefasst hat die Interpolation von Orientierungen mittels Quaternionen folgende Eigenschaften:

- geringer Berechnungsaufwand,
- einfache Umsetzung ohne Zusatzberechnungen,
- getrennte Interpolation von Position und Orientierung nötig,
- keine Singularitäten.

Zum Vergleich sind die beiden Ansätze LERP und SLERP in Bild 6.16(a) und Bild 6.16(b) nochmals grafisch gegenübergestellt (siehe auch [DKL98, S. 40–48] und [Ebe10]).

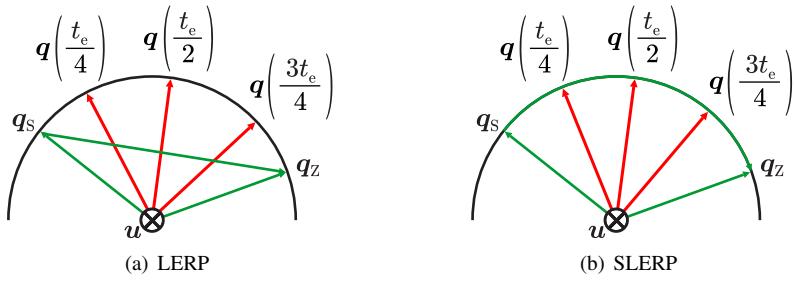


Bild 6.16: Projizierte Interpolationskurve der vierdimensionalen Einheitssphäre

6.6 Ganzheitliche Bahnplanung im Taskspace

Für eine ganzheitliche Bahnplanung ist eine Kombination der bisher vorgestellten Verfahren für Translation und Rotation erforderlich. So ergibt sich exemplarisch für eine geradlinige Translation (vgl. Abschnitt 6.4) mit konstanter Orientierungsänderung (vgl. Abschnitt 6.5) die interpolierte homogene Transformation:

$${}^S\mathbf{T}_E(s(t), t_{\text{rel}}) = \left(\begin{array}{c|c} {}^S\mathbf{R}_E(t_{\text{rel}}) & {}^{(S)}\mathbf{r}_{SE}(s(t)) \\ \hline 0 & 0 \\ 0 & 0 \end{array} \right). \quad (6.42)$$

Diese besteht aus der oben hergeleiteten Interpolation der Orientierung ${}^S\mathbf{R}_E(t_{\text{rel}})$ und einer einfachen Interpolation ${}^{(S)}\mathbf{r}_{SE}(s(t))$ zwischen den beiden Ursprüngen des Start- und Zielkoordinatensystems ${}^{(0)}\mathbf{r}_S$ und ${}^{(0)}\mathbf{r}_Z$ (siehe [SS00, S. 28ff und S. 193ff] für weitere Details). Wie in Bild 6.17 dargestellt, beschreibt der Vektor

$${}^{(0)}\mathbf{r}_{SE} = s(t) ({}^{(0)}\mathbf{r}_Z - {}^{(0)}\mathbf{r}_S) \quad (6.43)$$

die aktuelle Verschiebung des Ursprungs von $(KS)_E$ ausgehend vom Ursprung von $(KS)_S$ im $(KS)_0$ (es handelt sich hier nicht um einen Ortsvektor, sondern um einen Richtungsvektor). Ausgedrückt in $(KS)_S$ gilt:

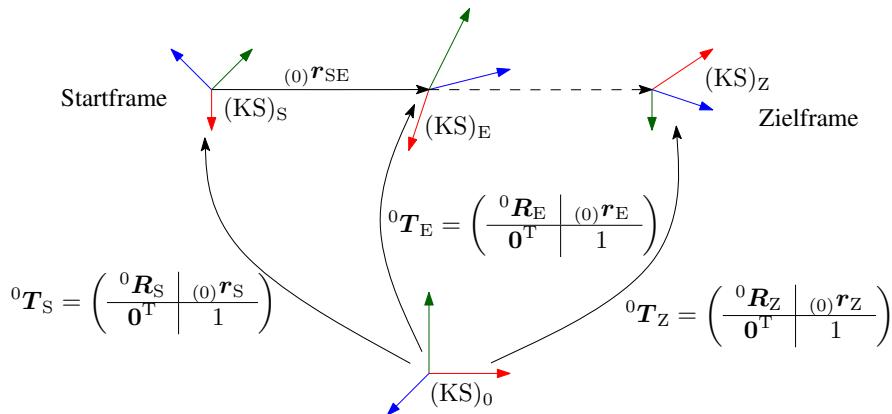


Bild 6.17: Grafische Darstellung zur Interpolation von Orientierungen

$${}^{(S)}\mathbf{r}_{SE} = {}^S\mathbf{R}_0 {}^{(0)}\mathbf{r}_{SE} = {}^S\mathbf{R}_0 s(t) ({}^{(0)}\mathbf{r}_Z - {}^{(0)}\mathbf{r}_S) \quad (6.44)$$

Mit ${}^S R_E = {}^S R_E(t_{\text{rel}}\theta)$ ergibt sich die in Gleichung (6.42) bereits angegebene homogene Transformationsmatrix. Zusammenfassend ist festzuhalten, dass das vorgestellte Verfahren eine Interpolation von Position und Orientierung in einem einheitlichen Konzept erlaubt.

6.7 Anwendungsbeispiel: Energieoptimale Roboterbahnen

Neben der Taktzeitreduktion ist eine immer wichtigere Zielvorgabe bei der Roboterprogrammierung ein möglichst geringer Energieverbrauch. Bisher wurde unter anderem versucht, die während einer Bewegung benötigten quadratischen Antriebsmomente

$$J_\tau = \int_0^{t_e} \sum_{i=1}^6 \tau_i^2(t, p) dt, \quad (6.45)$$

mit τ_i : Motordrehmoment der i -ten Achse, $n = 6$: Anzahl der Gelenke und p : Parameter der Bahn (z. B. Splineparameter) zu minimieren, so dass für die optimale Trajektorie gilt:

$$p_{\text{opt}} = \arg \min_p J_\tau, \quad (6.46)$$

wobei das benötigte Motordrehmoment mit den in Kapitel 7 vorgestellten Methoden berechnet wird. Nachteile dieses Ansatzes sind, dass

- wesentliche Verluste im Antriebsstrang (Motor, Umrichter, Getriebe, etc.) nicht beachtet werden (vgl. Kapitel 8) und
- die Möglichkeit des generatorischen Betriebs der permanent erregten Synchronmaschinen nicht berücksichtigt wird.

Der letzte Aspekt sei nochmals anhand der nachstehenden Bild 6.18 erläutert. Der Motor im oberen

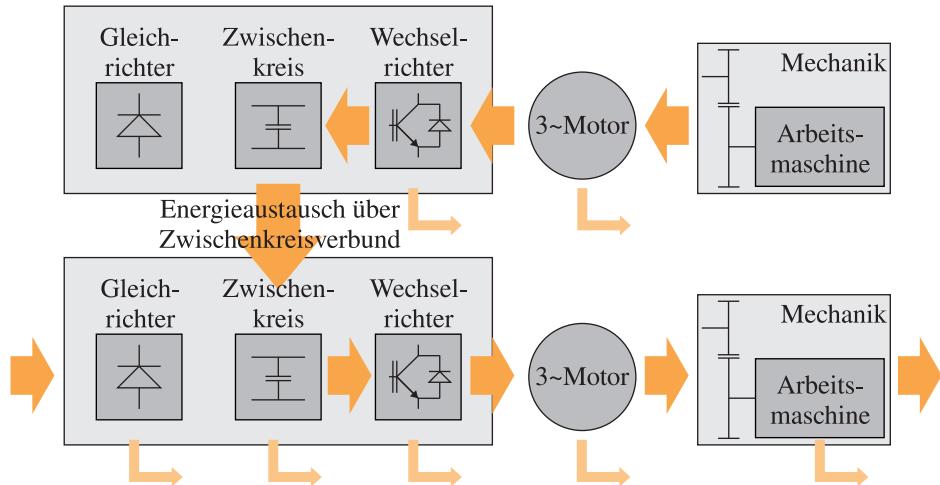


Bild 6.18: Energieaustausch über Zwischenkreisverbund am Beispiel von zwei Gleichstrommotoren

Zweig wird generatorisch betrieben und speist elektrische Energie in den Zwischenkreisverbund, während der untere Motor Energie aus diesem entnimmt. Lediglich die Differenz (inklusive Verluste) muss vom Netz gespeist werden. Ziel einer energieoptimalen Robotertrajektorie ist also, die während der Bewegung aufgenommene Netzenergie

$$J_{dc} = \int_0^{t_e} P_{dc}(t, p) dt \quad (6.47)$$

zu minimieren, wobei im Gegensatz zu Gleichung (6.45) der komplette Antriebsstrang sowie die Zwischenkreiskopplung der Roboterantriebe Eingang in die Kostenfunktion finden. Ein solches hochdimensionales Optimierungsproblem (die Anzahl der Parameter für eine PTP-Bewegung eines Roboters mit sechs Gelenkachsen beträgt bei Splines der Ordnung 5 mindestens 36) wird unter anderem mittels evolutionärer Algorithmen (z. B. Partikelschwarm-Algorithmen, Genetische Algorithmen) gelöst.

Bild 6.19(a) (oben) zeigt die aufgenommene Leistung bei einer exemplarischen Bewegung des in Bild 6.19(b) dargestellten Roboters: Die Achsen zwei und drei bewegen sich von 0 nach $-\frac{\pi}{2}$,

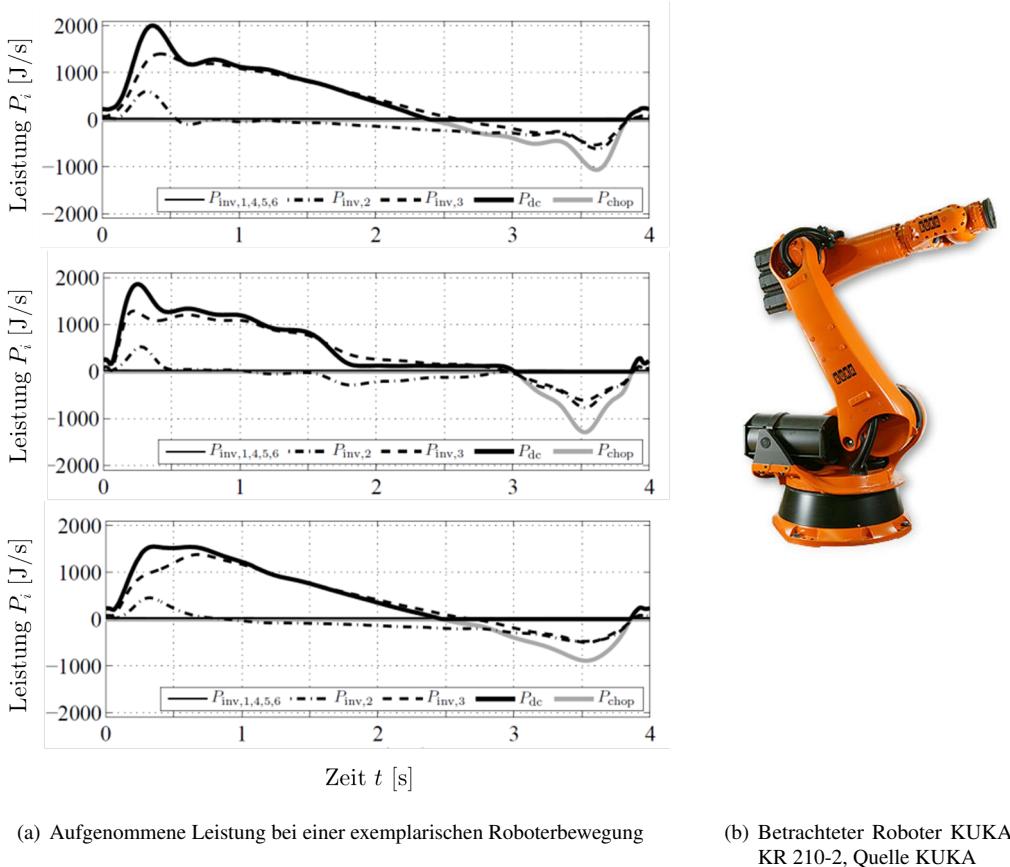


Bild 6.19: Beispiel einer Bahnoptimierung: aktuelle Ergebnisse mit Energieaustausch

die übrigen Antriebe verbleiben in Ruhe. Beide Achsen beschleunigen und bremsen nahezu gleichzeitig, so dass ein Energieaustausch, wie in Bild 6.18 gezeigt, kaum möglich ist. Ab dem Zeitpunkt $t \approx 2,5$ s wird die zurückgespeiste Energie über den Bremswiderstand in Wärme umgewandelt (Verlustleistung P_{chop}). Der Verlauf P_{dc} gibt die dem Netz entnommene Leistung an, während $P_{\text{inv},i}$ den elektrischen Leistungsfluss des i -ten Antriebsstrangs darstellt.

Der in Bild 6.19(a) (mitte) gezeigte Verlauf entsteht nach Optimierung der Trajektorie mit dem Gütekriterium J_{dc} (Gleichung (6.47)) – deutlich zu erkennen ist der ab $t \approx 1,7$ s spiegelbildliche Verlauf der von Achse zwei und Achse drei zurückgespeisten bzw. verbrauchten Leistung – es findet ein Energieaustausch über den Zwischenkreis statt. Im Gegensatz hierzu stellt Bild 6.19(a) (unten) die entsprechenden Verläufe bei einer Optimierung nach quadratischen Antriebsmomenten unter Verwendung des Gütekriteriums J_{τ} (Gleichung (6.45)) dar – der zuvor beschriebene spiegelbildliche Verlauf ist deutlich geringer ausgeprägt. Für eine Optimierung nach J_{dc} ergibt sich eine Reduktion der dem Netz entnommenen Energie um ca. 3,5% während sich für den anderen Fall (J_{τ}) lediglich eine Reduktion von 0,9% ergibt. Auch wenn die Energieeinsparung

gering erscheint, so ist doch zu berücksichtigen, dass diese keinerlei konstruktive Maßnahmen oder dergleichen benötigt, sondern lediglich auf einer (softwareseitig leicht umzusetzenden) geschickten Koordination der Achsbewegungen beruht (ein entsprechender Arbeitsraum des Roboters wird vorausgesetzt).

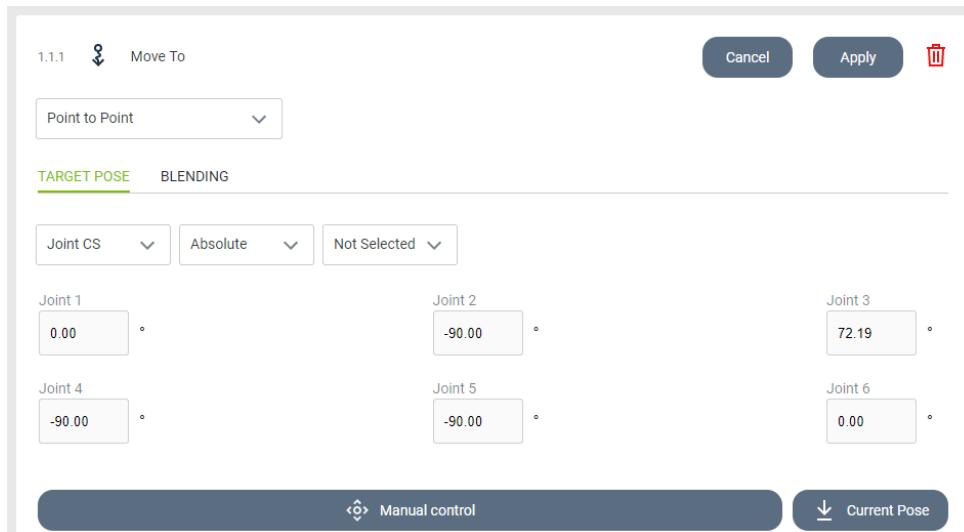
6.8 Roboterprogrammierung

Innerhalb einer Robotersteuerung sind verschiedene Verfahren zur Bahnplanung implementiert, von denen einige in diesem Kapitel vorgestellt wurde. Die Auswahl der Interpolationsmethoden ist aber typischerweise herstellerspezifisch und die Roboterprogrammierung somit ein wichtiges Unterscheidungsmerkmal. So verwendet KUKA beispielsweise die Robotersprache KRL (KUKA Robot Language), die unter anderem Bewegungsprimitive (vgl. Abschnitt 6.4) zur Verfügung stellt. Um die Programmierung des Roboters zu vereinfachen, werden sogenannte Inline-Formulare angeboten – dabei handelt es sich um befehlsspezifische Parameterfelder, die vom Benutzer auszufüllen sind. Darüber hinaus kommen zum Teil auch klassische Programmiersprachen wie Java (KUKA iiwa Roboter), Python (Yuanda Roboter) oder Karel (Pascal-Derivat, FANUC Roboter) zur Anwendung. Diese bieten einen deutlich größeren Befehls- und Parameterumfang, erfordern jedoch auch mehr Fachkenntnisse und Training. Daher sind sie zumeist Experten vorbehalten.

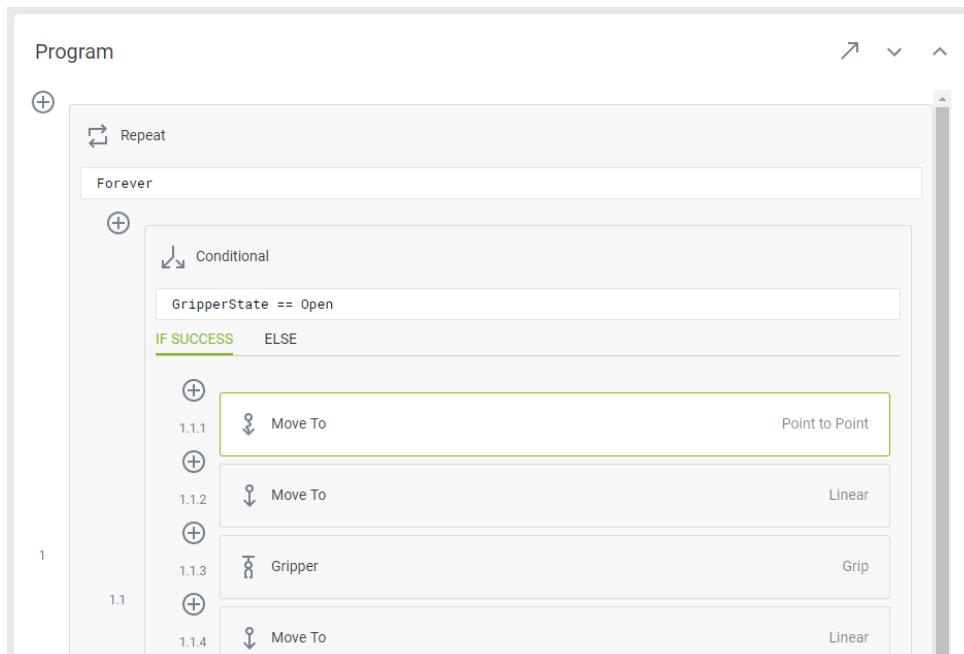
Um den Einsatz von Robotern in KMUs (kleinen und mittelständischen Unternehmen) voranzutreiben, erfreuen sich zunehmend grafische Bedienoberflächen großer Beliebtheit, da sie die Einarbeitungszeit sowie den Schulungsaufwand im Vergleich zu den zuvor genannten Programmiermethoden deutlich reduzieren und eine schnelle Anpassung des Roboterprogramms an neue Aufgaben erlauben. Dadurch wird diese Art der Automatisierung auch für die Produktion kleiner Losgrößen (mit hoher Variantenvielfalt) wirtschaftlich. Bild 6.20 zeigt exemplarisch die grafische Bedienoberfläche eines Yuanda Roboters, wobei in (a) die Parametrierung eines „MoveTo“ Befehls und in (b) ein Ausschnitt eines Programms für eine Pick-and-Place Anwendung zu sehen ist. Der in Bild 6.21 links dargestellte digitale Zwilling gestattet die Überprüfung des Programms zuerst in der Simulation, bevor der eigentliche Roboter (dasselbe) Programm in der Realität ausführt. Somit ist es möglich, Programmierfehler oder falsch gesetzte Parameter frühzeitig zu erkennen, bevor sie Schaden anrichten können.

Fragen zur Selbstkontrolle

1. Welche Varianten von Roboterbahnen sind Ihnen bekannt? Beschreiben Sie die Varianten und nennen Sie die wesentlichen Unterschiede!
2. Zeichnen Sie qualitativ ein typisches Geschwindigkeitstrapezprofil sowie ein zeitoptimales Geschwindigkeitstrapezprofil, bei dem die Maximalgeschwindigkeit nicht erreicht wird! Wie nennt man diesen zeitoptimalen Fall? Was lässt sich über die Stetigkeit eines solchen Profils aussagen?
3. Erläutern Sie das konkrete Vorgehen bei einer Interpolation der Rotationswinkel der zusammengesetzten Drehungen und nennen Sie alternative Ansätze für die Interpolation von Orientierungen!



(a) Parametrierung



(b) Befehlsfolge

Bild 6.20: Auszüge aus der grafischen Bedienoberfläche des Yuanda Roboters, (a) zeigt die Parametrierung einer Punkt-zu-Punkt Bewegung, (b) den beispielhaften Programmablauf einer Pick-and-Place Anwendung, Quelle: Yuanda Robotics GmbH



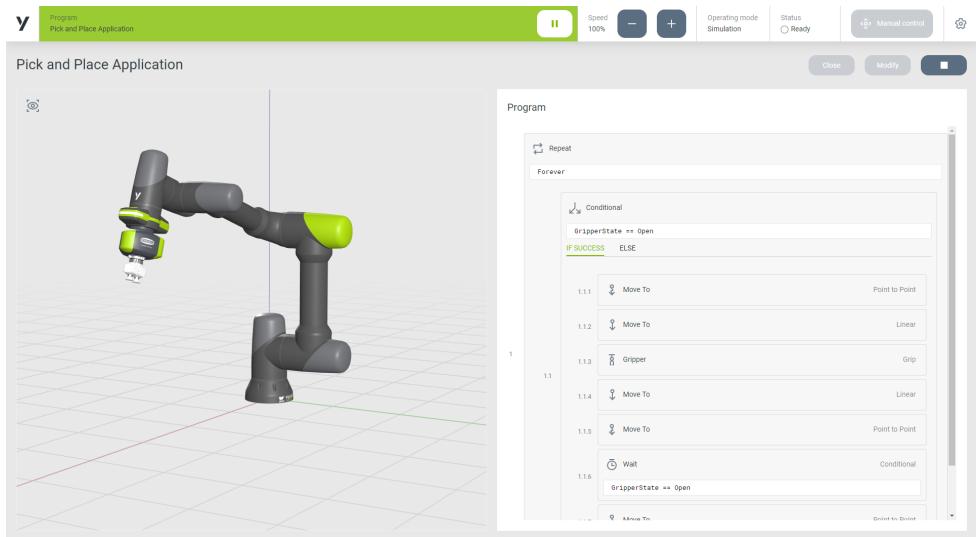


Bild 6.21: Digitaler Zwilling zur Simulation von Bewegungsbefehlen, Quelle: Yuanda Robotics GmbH



6.9 Automatisierte Bahnplanung

Die zentrale Aufgabe der Bahnplanung besteht darin, eine gültige Trajektorie für ein Robotersystem zu berechnen, die es ermöglicht, von einer gegebenen Startkonfiguration zu einer gewünschten Zielkonfiguration zu gelangen. Dabei müssen neben geometrischen Anforderungen, wie der Vermeidung von Kollisionen mit Hindernissen, auch kinematische und dynamische Nebenbedingungen berücksichtigt werden (Bild 6.22). Eine Lösung des Bahnplanungsproblems ist dann gefunden, wenn eine zeitlich konsistente Abfolge von Roboterkonfigurationen vorliegt, die den Übergang vom Start- zum Zielzustand sicher und effizient ermöglicht. Algorithmisch soll die Bahn zwischen Startzustand und Zielzustand zeit-, weg-, kostenoptimal und kollisionsfrei geplant werden.

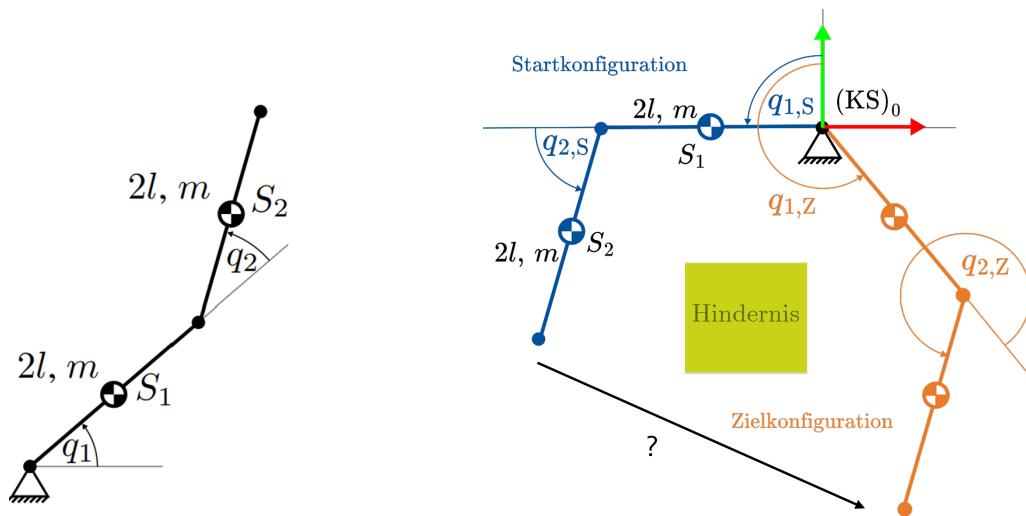


Bild 6.22: Das Bahnplanungsproblem zwischen dem Startzustand und dem Zielzustand am Beispiel eines RR-Roboters. Links ist der RR-Roboter und rechts ist die Start- q_S und Zielkonfiguration q_Z abgebildet.

In der Bewegungsplanung für Roboter lassen sich grundsätzlich zwei Ansätze (vgl. Bild ??) unterscheiden:

- **Manuelle Bahnplanung:** Hierbei werden Bewegungsbahnen direkt durch den Menschen vorgegeben — entweder durch Programmierung im Jointspace (Gelenkwinkel) oder Taskspace (z. B. kartesische Koordinaten). Diese Methode eignet sich besonders für einfache, standardisierte Aufgaben wie das Schweißen und Fräsen in der industriellen Produktion.
- **Automatisierte Bahnplanung:** Bei komplexeren oder dynamischen Aufgaben wird die Bewegung des Roboters algorithmisch berechnet. Dies ist notwendig, wenn sich Hindernisse verändern, Ziele variieren oder der Roboter autonom Entscheidungen treffen soll.

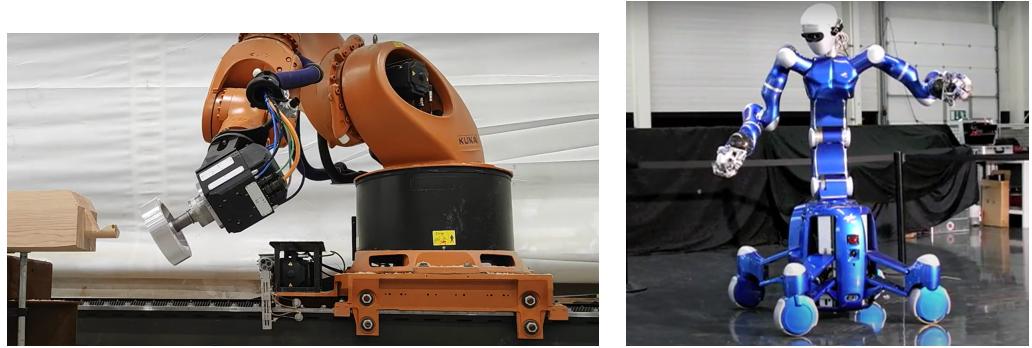


Bild 6.23: Anwendungsbeispiele für die Bewegungsplanung. Links findet die manuelle Bewegungsplanung beim robotergestützten Entgraten von Holz mit einem KUKA-Roboter Anwendung (Quelle: Robotic Hitech Solutions). Rechts findet die automatisierte Bewegungsplanung am DLR-Roboter Rollin' Justin beim Ballfangen (Quelle: Deutsches Zentrum für Luft- und Raumfahrt, DLR) Anwendung.

Beide Verfahren haben ihre Daseinsberechtigung, unterscheiden sich aber erheblich im Aufwand, in der Flexibilität und im Anwendungsspektrum. In der industriellen Praxis werden einfache Bewegungen, wie sie beim Fräsen oder Schweißen vorkommen, oft manuell programmiert. Diese manuelle Bahnplanung im Jointspace (Gelenkkraum) oder Taskspace (Arbeitsraum) funktioniert gut bei vorhersehbaren, sich wiederholenden Aufgaben. In komplexeren Szenarien mit dynamischen Umgebungen, wechselnden Zielpositionen oder einer Vielzahl möglicher Bewegungsoptionen ist jedoch eine automatisierte Bahnplanung unerlässlich, wofür automatisierte Bahnplanungsmethoden eingesetzt werden.

6.9.1 Wiederholung: Arbeitsraum und Konfigurationsraum

Ein wichtiger konzeptueller Unterschied ist jener zwischen dem Arbeitsraum \mathcal{X} und dem Konfigurationsraum \mathcal{Q} :

- Arbeitsraum \mathcal{X} (Taskspace): Der Bereich, in dem sich das Endeffektorwerkzeug des Roboters physisch bewegt (z. B. XYZ-Koordinaten und Orientierungen).
- Konfigurationsraum \mathcal{Q} (Configuration- oder C-Space): Der mathematische Raum aller möglichen Zustände eines Roboters, z. B. Gelenkwinkel bei einem Gelenkarm. Jeder Punkt im C-Space stellt eine eindeutige Konfiguration des Roboters dar.

Die Umrechnung zwischen Arbeits- und Konfigurationsraum erfolgt mittels Kinematik (direkt und invers). Die Bahnplanung erfolgt typischerweise im Konfigurationsraum \mathcal{Q} , da dieser alle Bewegungsfreiheitsgrade und Einschränkungen des Roboters berücksichtigt. Für die Planung werden sowohl Arbeitsraum als auch Konfigurationsraum in kollisionsfreien und kollisionsbehafteten Unterräume unterteilt (Bild. 6.24).

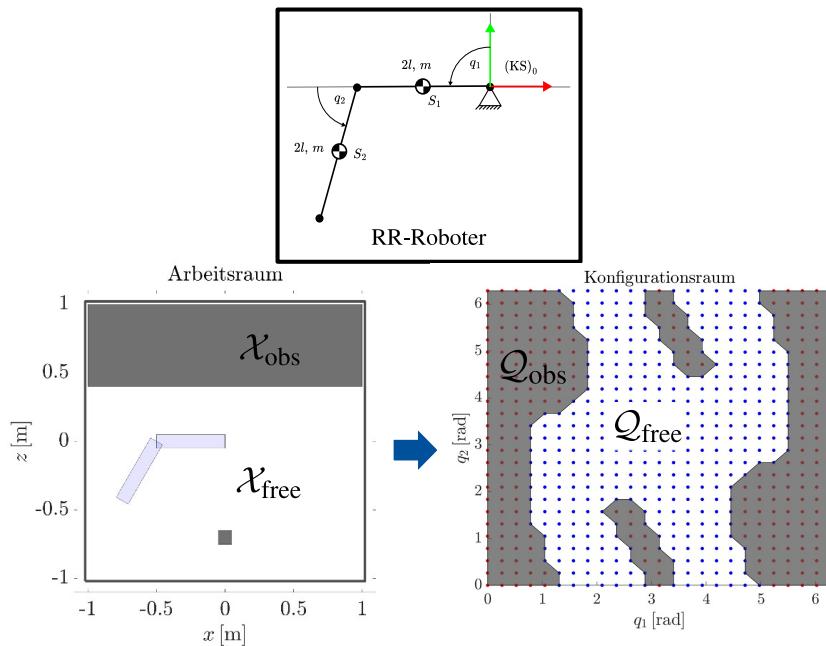


Bild 6.24: Der Arbeitsraum $\mathcal{X} = \mathcal{X}_{\text{free}} \cup \mathcal{X}_{\text{obs}}$ eines Roboters beschreibt den Bereich in dem sich das Endeffektorwerkzeug bewegt, was typischerweise euklidische Koordinaten oder Orientierungen sind. Dabei ist $\mathcal{X}_{\text{free}}$ der kollisionsfreie und \mathcal{X}_{obs} der kollisionsbehaftete kartesische Raum. Der Konfigurationsraum $\mathcal{Q} = \mathcal{Q}_{\text{free}} \cup \mathcal{Q}_{\text{obs}}$ beschreibt alle möglichen Zustände eines Roboters, wie z.B. die Gelenkwinkel des Roboters. Dabei beschreibt $\mathcal{Q}_{\text{free}}$ den kollisionsfreien und \mathcal{Q}_{obs} den kollisionsbehafteten Konfigurationsraum.

Bild 6.25 zeigt drei Systeme: Beim Portalkran entspricht jede Antriebsachse direkt einer kartesischen Raumrichtung, sodass Arbeits- und Konfigurationsraum identisch sind. Der RRR-Roboter besitzt ebenfalls drei Freiheitsgrade, jedoch nichtlineare Kinematik mit Rotationsgelenken. Der Schlangenroboter bewegt sich ebenfalls in $\mathcal{X} \subseteq \mathbb{R}^3$, besitzt aber zehn Gelenke und somit einen hochdimensionalen Konfigurationsraum $\mathcal{Q} \subseteq \mathbb{R}^{10}$.

Portalkran	RRR-Roboter	Schlangenroboter	
Arbeitsraum $(x,y,z) \in \mathcal{X} \subseteq \mathbb{R}^3$	(3D): $(x,y,z) \in \mathcal{X} \subseteq \mathbb{R}^3$	Arbeitsraum $(x,y,z) \in \mathcal{X} \subseteq \mathbb{R}^3$	(3D): $(x,y,z) \in \mathcal{X} \subseteq \mathbb{R}^3$
Konfigurationsraum $(q_1, q_2, q_3) \in \mathcal{Q} \subseteq \mathbb{R}^3$	(3D): $(q_1, q_2, q_3) \in \mathcal{Q} \subseteq \mathbb{R}^3$	Konfigurationsraum $(q_1, q_2, q_3) \in \mathcal{Q} \subseteq \mathbb{R}^3$	(3D): $(q_1, \dots, q_{10}) \in \mathcal{Q} \subseteq \mathbb{R}^{10}$

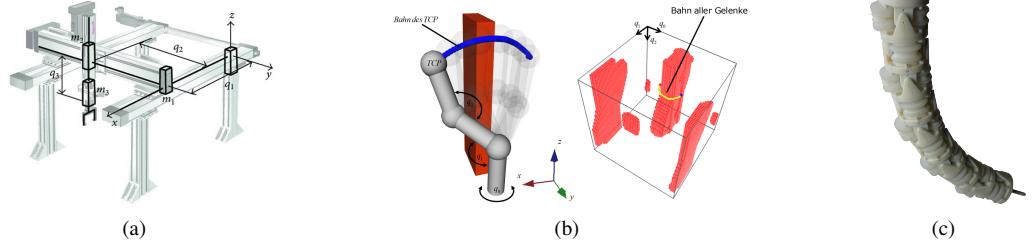


Bild 6.25: Der Arbeits- und Konfigurationsraum von verschiedenen Kinematiken: (a) Portalkran, (b) RRR-Roboter (Quelle: Karlsruher Institut für Technologie, KIT) und (c) imes-Schlangenroboter SPONGE

6.9.2 Diskretisierung des Konfigurationsraum

Zur Anwendung diskreter Planungsverfahren, muss der kontinuierliche Konfigurationsraum durch eine sogenannte Diskretisierung in einen diskreten Raum überführt werden. Das heißt, der kontinuierliche Konfigurationsraum ist in eine endliche Menge diskreter Zustände aufzuteilen. Ziel ist es, die Suche nach kollisionsfreien Pfaden auf dieser diskreten Repräsentation anzuwenden. Es existieren verschiedene Diskretisierungsstrategien. Die Wahl der Methode hängt von der Geometrie des Roboters, der Umgebungsstruktur und dem Komplexitätsgrad der Bewegungsaufgabe ab. Zu den Diskretisierungsstrategien zählen:

- **Sichtgraph:** Es werden nur freie Sichtverbindungen zwischen Konfigurationspunkten erzeugt, typischerweise zwischen Hinderniskanten, wie in Bild 6.26 dargestellt. Besonders geeignet für 2D-Planung mit polygonalen Hindernissen. Allerdings können bei 3D-Problemen die kürzesten Wege nicht gefunden werden.

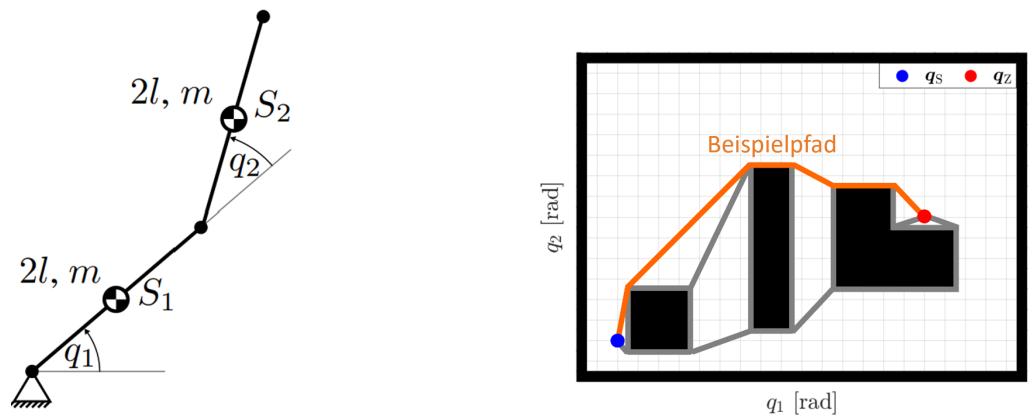


Bild 6.26: Rechts ist ein Beispiel eines Sichtgraphs mit q_S als Startkonfiguration und q_Z als Zielkonfiguration. Dabei sind q_i die Gelenkwinkel des RR-Roboters dargestellt (siehe links).

- **Approximative Zellzerlegung:** Der Raum wird in gleichförmige (z. B. rechteckige) oder adaptive Zellen zerlegt, die in freie und belegte Zellen klassifiziert werden. Dies erlaubt eine flexible Bahnplanung durch das freie Zellnetz (Bild 6.27).

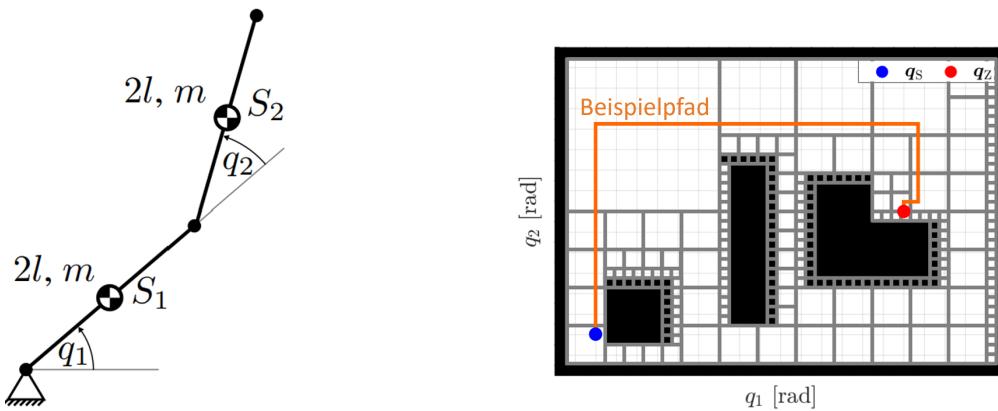


Bild 6.27: Beispiel einer Zellzerlegung mit q_S als Startkonfiguration und q_Z als Zielkonfiguration.

- **Brute-Force-Koordinaten-Diskretisierung:** Der Raum wird regelmäßig in Gitterpunkte unterteilt, wie es in Bild 6.28 exemplarisch dargestellt ist. Jeder Rasterpunkt wird als möglicher Zustand betrachtet. Diese Methode ist einfach umzusetzen, führt jedoch bei hoher Auflösung zu einem sehr großen Konfigurationsraum.

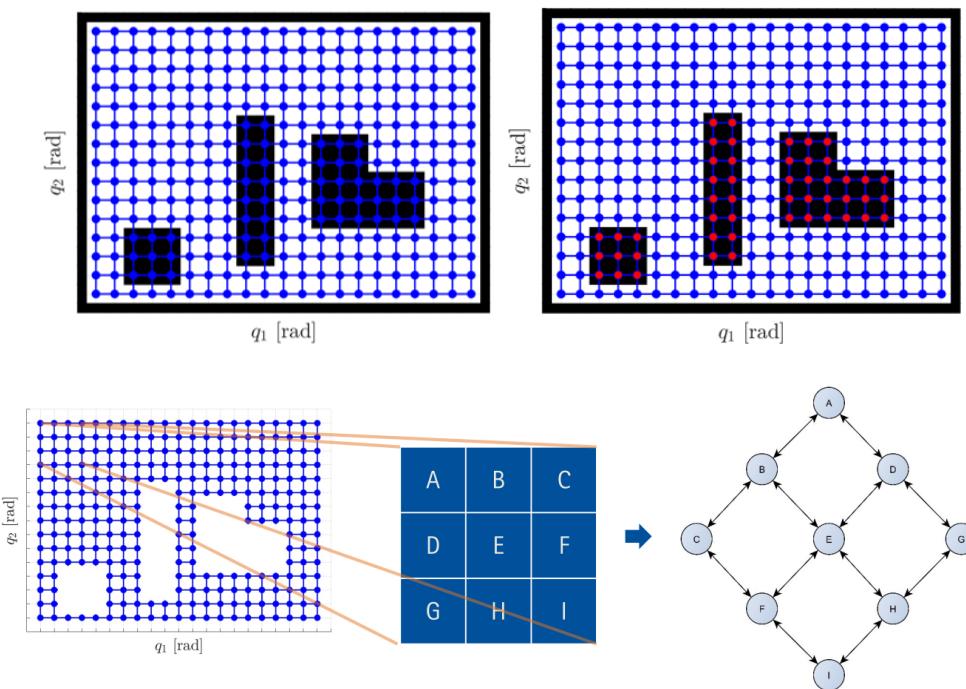


Bild 6.28: Bei der Brute-Force-Diskretisierung wird der Konfigurationsraum in äquidistante, diskrete Zellen unterteilt, wobei jede dieser Zellen einen diskreten Zustand in dem Planungsproblem darstellt. Vorteil der Brute-Force-Diskretisierung ist die Einfachheit des Vorgehens. Allerdings benötigt der Ansatz oftmals sehr große Speichermengen.

6.9.3 Diskrete Planungsprobleme

Ein diskretes Planungsproblem lässt sich durch eine endliche Menge an Zuständen s_i und möglichen Aktionen a_i beschreiben. Der Zustandsraum \mathcal{S} umfasst alle möglichen diskreten Zustände s_i ,

in denen sich das System befinden kann, wie bspw. Rasterpunkte in einem zweidimensionalen Gitter. Der Aktionsraum \mathcal{A} definiert die erlaubten Bewegungen zwischen Zuständen s_i (z. B. Schritt nach oben, unten, diagonal). Eine Bewegung ergibt sich aus der Anwendung einer Aktion a_i auf den aktuellen Zustand s_i und wird durch eine Übergangsfunktion von einem Startzustand s_S in einen Zielzustand s_Z eindeutig bestimmt. Das zentrale Ziel der Planung besteht darin, eine Abfolge von Aktionen a_i , d.h. einen Plan zu finden, der den Startzustand s_S in den Zielzustand s_Z überführt. Zusätzlich können in manchen Problemen Pfadkosten auftreten, die einzelner Verbindungen zwischen Start und Ziel zugeordnet werden. Dadurch entscheidet die Summe der entlang des Pfads anfallenden Kosten über die Güte einer Lösung. Typische Algorithmen für diskrete Planungsprobleme untersuchen den Zustandsraum \mathcal{S} und unterscheiden sich hinsichtlich Effizienz, optimaler Lösungsgarantie und Rechenaufwand.

Definition
Zustandsraum

In den Ingenieursdisziplinen ist der Begriff "Zustandsraum" generell nicht eindeutig definiert, weil der Begriff in sehr vielen Disziplinen verwendet wird und dabei für Unterschiedliches steht. In diesem Skriptabschnitt beschäftigen wir uns mit der diskreten Bewegungsplanung und in diesem Kontext beschreibt der Begriff "Zustandsraum" einen diskreten Raum von möglichen Zuständen, die in dem gegebenen Planungsproblem eingenommen werden können. Somit hängt der Zustandsraum von dem konkreten Planungsproblem ab. Wenn wir ein robotisches Planungsproblem betrachten, ist eine typische Wahl für den Zustandsraum des Planungsproblems der diskretisierte Konfigurationsraum des Roboters oder auch der diskretisierte Arbeitsraum des Roboters. Wenn wir allerdings Planungsprobleme, wie die Bewegung einer Schachfigur oder das Lösen eines Rubiks Cube, betrachten, so wäre der Zustandsraum des Planungsproblems als die Menge aller möglichen Schachpositionen oder Ausrichtungen des Rubiks Cube definiert. Um zu verdeutlichen, dass wir von dem Zustandsraum eines allgemeinen Planungsproblems sprechen, verwenden wir \mathcal{S} als Variable für den Zustandsraum und s für einen Zustand. In dem Fall von einem robotischen Planungsproblem, bei dem der Zustandsraum des Planungsproblems der Konfigurationsraum des Roboters ist, wird \mathcal{Q} für den Konfigurationsraum bzw. den Zustandsraum und q für die Konfiguration bzw. für den Zustand verwendet.

6.9.4 Algorithmen für die automatisierte Bahnplanung

Für die Bahnplanung existieren verschiedene Suchverfahren, die sich in ihrer Vorgehensweise unterscheiden. Die Breadth-First-Suche durchläuft den Zustandsraum \mathcal{S} schrittweise in Schichten. Zunächst werden alle Zustände s_i in unmittelbarer Nähe zum Start untersucht, bevor in einem zweiten Schritt die jeweils weiter entfernten Zustände betrachtet werden. Im Gegensatz dazu folgt die Depth-First-Suche einer Richtung so lange, bis kein weiterer Übergang möglich ist, bevor andere Optionen geprüft werden. Darüber hinaus betrachtet der Dijkstra-Algorithmus nicht nur die Abfolge von Zuständen s_i , sondern auch die mit jeder Aktion a_i verbundenen Kosten. Einen anderen Ansatz verfolgt die Greedy-Suche, die sich an einer Heuristik orientiert. Bei jedem Schritt wird derjenige Nachfolger gewählt, der der Schätzung zufolge dem Ziel am nächsten liegt. Schließlich verbindet der A*-Algorithmus die Ideen von Dijkstra und Greedy, indem sowohl die tatsächlich angefallenen Kosten als auch eine heuristische Schätzung der Restkosten berücksichtigt werden.

Alle genannten Strategien lassen sich unter dem Rahmen der Vorwärtssuche einordnen. Dabei wird ausgehend vom Startzustand der Zustandsraum systematisch erweitert, bis ein Pfad zum Ziel gefunden ist, wobei die Suche stets nach demselben Muster in Alg. 1 abläuft.

Algorithm 1 Vorwärtssuche

- 1: Initialisiere einen Speicher \mathcal{C}
 - 2: Lege den initialen Zustand s_S in den Speicher \mathcal{C}
 - 3: **While**-Schleife: Solange der Speicher \mathcal{C} nicht leer ist:
 - 4: Entnehme den obersten Zustand s aus dem Speicher \mathcal{C}
 - 5: **If**-Bedingung: Zustand s gleich Zielzustand s_Z :
 - 6: Suche erfolgreich beendet
 - 7: **For**-Schleife: Für jede mögliche Aktion a
 - 8: Applizierte Aktion a auf Zustand s für neuen Zustand s'
 - 9: **If**-Bedingung: s' noch nicht besucht
 - 10: Markiere s' als besucht
 - 11: Lege s' in dem Speicher \mathcal{C} ab
 - 12: Ende der **For**-Schleife
 - 13: Ende der **While**-Schleife
 - 14: **If**-Bedingung: While-Schleife wurde durchlaufen, ohne den Zustand s_Z erreicht zu haben
 - 15: Suche erfolglos beendet
-

Welche konkrete Strategie zur Bahnplanung (bspw. Breadth-First-Suche, Depth-First-Suche, Dijkstra-Algorithmus) eingesetzt wird, hängt davon ab, welches Konzept für den Speicher \mathcal{C} (bspw. Last-In-First-Out (LIFO), First-in-First-Out (FIFO), kosten-sortiert) implementiert wird.

Breadth-First

Die Breadth-First-Suche ist eine einfache, aber systematische Methode zur Lösung diskreter Planungsprobleme. Der Zustandsraum wird dabei schrittweise erkundet. Das heißt, sie analysiert zuerst alle Zustände, die vom Startzustand durch eine einzige Aktion erreichbar sind, bevor sie zu weiter entfernten Zuständen übergeht.

Die Breadth-First-Suche kann als Vorwärtssuche nach dem Alg. 1 implementiert werden, indem als Speicher \mathcal{C} , eine Warteschlange nach dem FIFO-Prinzip (First-In-First-Out) verwendet wird. Dadurch wird gewährleistet, dass die Zustände in der Reihenfolge ihrer Generierung abgearbeitet werden. Die Methode ist **vollständig**, das heißt, sie findet garantiert eine Lösung, wenn eine existiert. Allerdings ist sie **nicht optimal** in Bezug auf Pfadkosten, außer wenn alle Aktionen identische Kosten besitzen. Darüber hinaus, erfordert die Methode einen **hohen Speicherbedarf**, da viele Zustände gleichzeitig in der Warteschlange liegen können. Bild 6.29 zeigt ein Beispiel eines Suchbaums mit Pfadknoten. Ein Beispielverlauf einer Breadth-First-Suche bei dem die Suche schrittweise ausgeführt wird, ist in Bild 6.30 dargestellt, sodass der Ablauf nachvollzogen werden kann.

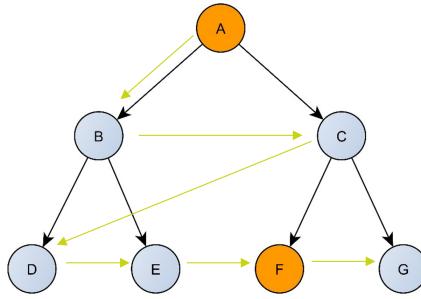


Bild 6.29: Beispiel der Pfadknoten der Breadth-First-Suche

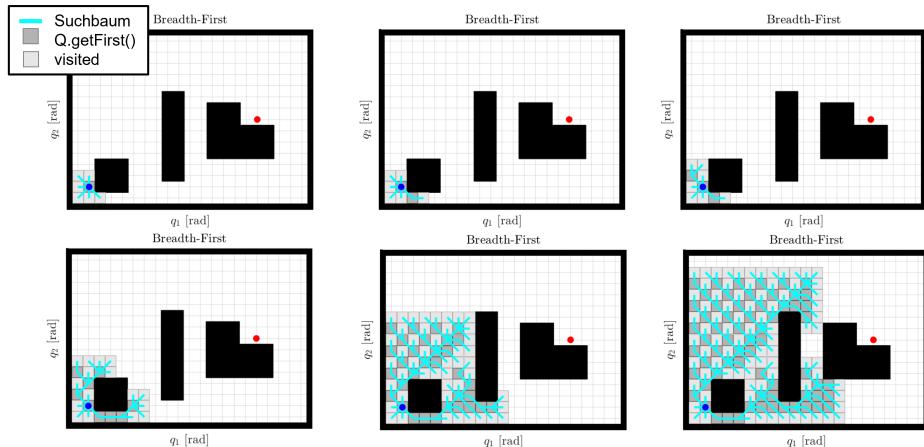


Bild 6.30: Beispielverlauf einer Breadth-First Suche (Die Code-Umgebung wird im Rahmen der Veranstaltung zur Verfügung gestellt, sodass die einzelnen Schritte der Implementierung nachvollziehbar sind)

Depth-First

Die Depth-First-Suche verfolgt einen anderen Ansatz als die Breadth-First-Suche: Sie durchsucht den Zustandsraum **in die Tiefe**, indem sie jeweils einem Pfad so weit wie möglich folgt, bevor sie alternative Wege betrachtet.

Die Depth-First-Suche kann als Vorwärtssuche nach dem Alg. 1 implementiert werden, indem ein Stapelspeicher \mathcal{C} nach dem LIFO-Prinzip (Last-In-First-Out) verwendet wird.

Der Algorithmus zeichnet sich durch eine vergleichsweise einfache Implementierung sowie eine effizientere Speicherverwaltung im Vergleich zur Breadth-First-Suche aus, da weniger Zustände gleichzeitig gespeichert werden müssen. Allerdings bringt dieses Vorgehen wesentliche Einschränkungen mit sich: Die Depth-First-Suche ist **nicht optimal** in Bezug auf Pfadlänge oder -kosten, da sie keine Garantie liefert, den kürzesten oder besten Pfad zu finden. Zudem kann sie in Sackgassen laufen und dabei unnötig lange Pfade erkunden, bevor eine Lösung gefunden wird.

Die Funktionsweise des Algorithmus wird in Bild 6.31 durch einen Suchbaum illustriert, in dem die Abfolge der expandierten Knoten dargestellt ist. In Bild 6.32 ist ein exemplarischer Suchverlauf innerhalb eines zweidimensionalen Gitterraums abgebildet, bei dem die sukzessive Exploration einzelner Pfade deutlich wird. Dabei zeigt sich, dass die Depth-First-Suche sehr schnell in tiefe Strukturen des Zustandsraums eindringen kann, ohne zunächst alle alternativen Nachbarn des Startknotens zu berücksichtigen.

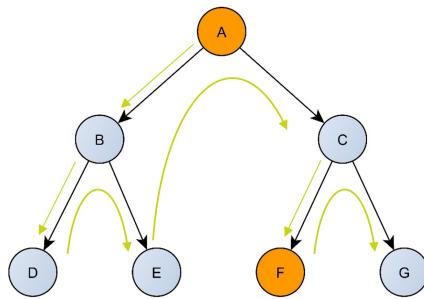


Bild 6.31: Beispiel der Pfadknoten der Depth-First-Suche

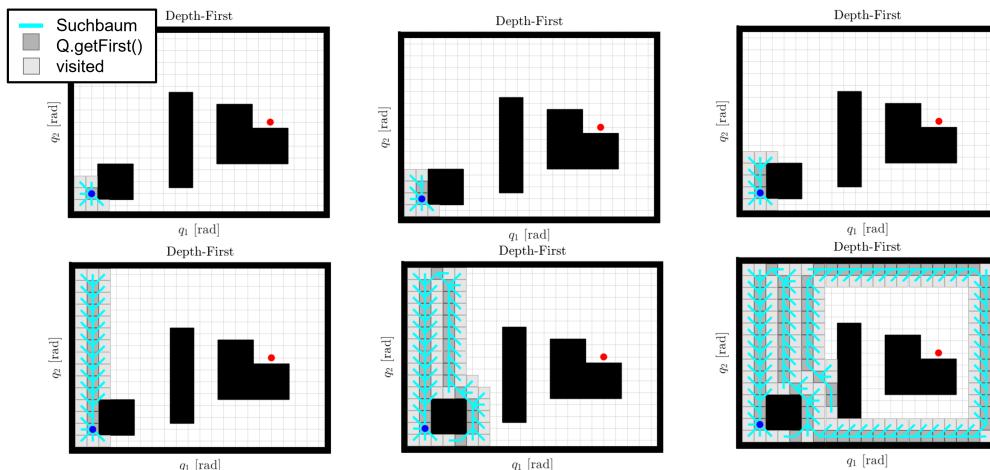


Bild 6.32: Beispielverlauf einer Depth-First-Suche (Die Code-Umgebung wird im Rahmen der Veranstaltung zur Verfügung gestellt, sodass die einzelnen Schritte der Implementierung nachvollziehbar sind)

Dijkstra-Algorithmus

Der Dijkstra-Algorithmus ist ein klassischer Suchalgorithmus zur Lösung diskreter Planungsprobleme, der stets den **kostengünstigsten Pfad** vom Start- zum Zielzustand findet. Im Gegensatz zu Breadth-First und Depth-First berücksichtigt er die Pfadkosten zu jedem Knoten und trifft seine Entscheidungen systematisch auf Basis dieser Information. Der Algorithmus durchsucht den Zustandsraum, indem er immer den Knoten mit den **geringsten bisher bekannten Pfadkosten** vom Start auswählt und dessen Nachbarn analysiert. Sobald der Zielzustand erreicht ist, ist garantiert, dass der gefundene Pfad der kostengünstigste ist. Im Folgenden ist eine Übersicht der Hauptschritte:

1. Allen Knoten wird zunächst eine unendliche Distanz zugewiesen, außer dem Startknoten (Distanz = 0).
2. Eine Prioritätswarteschlange verwaltet die Knoten nach aktueller Distanz zum Start.
3. In jedem Schritt wird der Knoten mit der geringsten Distanz entfernt („besucht“).
4. Die Nachbarn dieses Knotens werden aktualisiert, wenn über den aktuellen Knoten ein günstigerer Weg gefunden wurde.
5. Dieser Vorgang wiederholt sich, bis der Zielknoten besucht wurde.

Der Dijkstra-Algorithmus bezeichnet sich durch seine **Optimalität**, da er immer den Pfad mit minimalen Gesamtkosten findet, sowie seine **Vollständigkeit**, da eine Lösung garantiert gefunden wird, sofern sie existiert. Nachteile bestehen jedoch in der hohen **Rechenintensität**, insbesondere

bei großen oder gleichmäßig bewerteten Zustandsräumen, da jeder Knoten explizit ausgewertet wird. Zudem handelt es sich um ein Verfahren **ohne Heuristik**, da ausschließlich bekannte Pfadkosten berücksichtigt werden. Eine Besonderheit liegt darin, dass Dijkstra im Fall identischer Pfadkosten äquivalent zur Breadth-First-Suche arbeitet, allerdings mit deutlich höherem **Rechenaufwand**, da jeder Knoten exakt ausgewertet wird. Die Funktionsweise wird in Bild 6.33 durch einen gewichteten Suchbaum verdeutlicht. Darüber hinaus wird in Bild 6.34 der Ablauf in einem zweidimensionalen Gitterraum illustriert.

Eine besondere Eigenschaft des Dijkstra-Algorithmus zeigt sich dann, wenn alle Pfadkosten identisch sind, z. B. bei einem mobilen Roboter mit gleich langen Schritten. In diesem Fall entspricht Dijkstra in seiner Funktionsweise effektiv einer Breadth-First-Suche. Allerdings ist der Rechenaufwand deutlich höher, da jeder Knoten exakt ausgewertet werden muss.

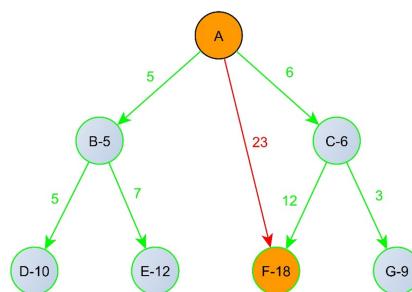


Bild 6.33: Beispiel der Pfadknoten und -kosten des Djikstra-Algorithmus

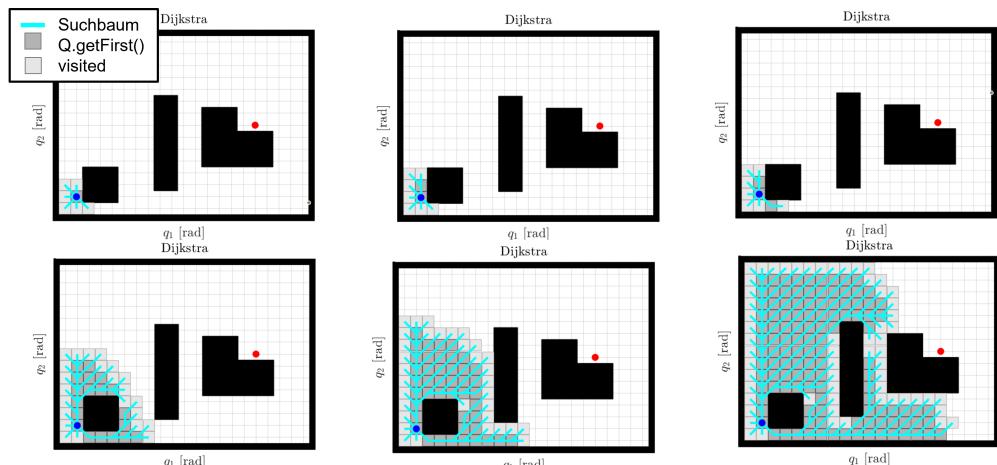


Bild 6.34: Beispielverlauf des Djikstra-Algorithmus (Die Code-Umgebung wird im Rahmen der Veranstaltung zur Verfügung gestellt, sodass die einzelnen Schritte der Implementierung nachvollziehbar sind)

Greedy-Suche

Die Greedy-Suche ist ein heuristischer Algorithmus, der bei der Lösung diskreter Planungsprobleme besonders auf **Schnelligkeit** ausgelegt ist. Sie folgt stets der Richtung, die **scheinbar am schnellsten** zum Ziel führt, basierend auf einer Schätzung der verbleibenden Distanz eines Knotens zum Ziel. Die Grundidee besteht darin, anstatt alle bisherigen Pfadkosten zu berücksichtigen, lediglich zu bewerten, **wie nah ein Knoten am Ziel liegt**. Der jeweils zielnächste Knoten wird als nächstes untersucht, unabhängig vom bisher zurückgelegten Weg. Der Verlauf des Algorithmus wird wie folgt dargestellt:

1. Eine Heuristikfunktion schätzt die Entfernung jedes Knotens zum Ziel.
2. Eine Prioritätswarteschlange verwaltet die offenen Knoten, sortiert nach der Heuristikfunktion.
3. In jedem Schritt wird der Knoten mit dem **kleinsten Heuristikkwert** ausgewählt.
4. Nachbarn dieses Knotens werden bewertet und zur Warteschlange hinzugefügt.
5. Der Algorithmus bricht ab, sobald das Ziel erreicht wurde.

Charakteristisch für die Greedy-Suche ist die **hohe Geschwindigkeit**, da die Exploration direkt in Richtung des Ziels gelenkt wird. Gleichzeitig weist sie jedoch erhebliche Einschränkungen auf. Das Verfahren ist **nicht vollständig**, da es in Sackgassen enden oder potenziell endlos suchen kann, und **nicht optimal**, da es nicht garantiert den kürzesten oder kostengünstigsten Pfad findet. Ein Vorteil besteht im **geringen Speicherbedarf**, da im Gegensatz zur Breadth-First-Suche oder Dijkstra nur wenige Knoten gleichzeitig in der Warteschlange gehalten werden müssen. Die Funktionsweise ist in Bild 6.35 anhand eines Gitterraums veranschaulicht. Der Suchverlauf zeigt, dass die Greedy-Suche konsequent in Richtung des Zielknotens voranschreitet, jedoch Umwege oder Sackgassen nicht effizient vermeidet. In Bild 6.36 ist ein Vergleich zwischen der Greedy-Suche und dem Dijkstra-Algorithmus dargestellt. Dies verdeutlicht, dass der Dijkstra-Algorithmus im Gegensatz zur Greedy-Suche stets den kürzesten Pfad garantiert.

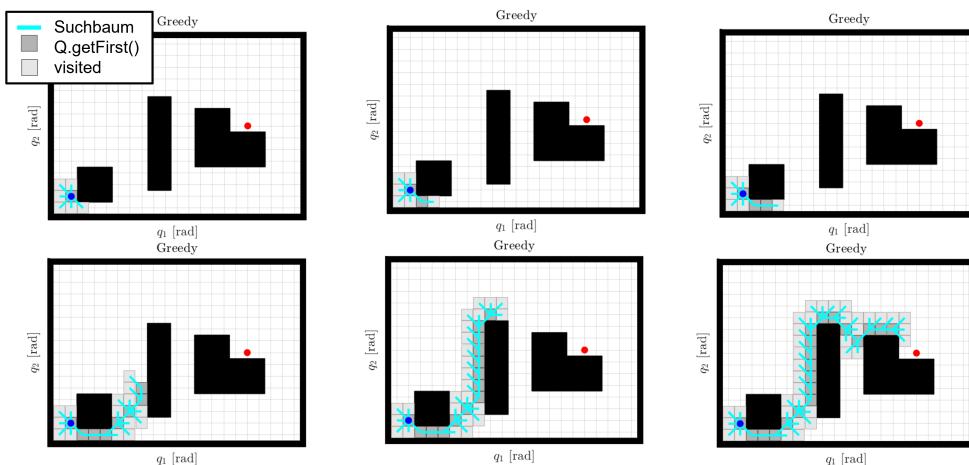


Bild 6.35: Beispielverlauf der Greedy-Suche (Die Code-Umgebung wird im Rahmen der Veranstaltung zur Verfügung gestellt, sodass die einzelnen Schritte der Implementierung nachvollziehbar sind)

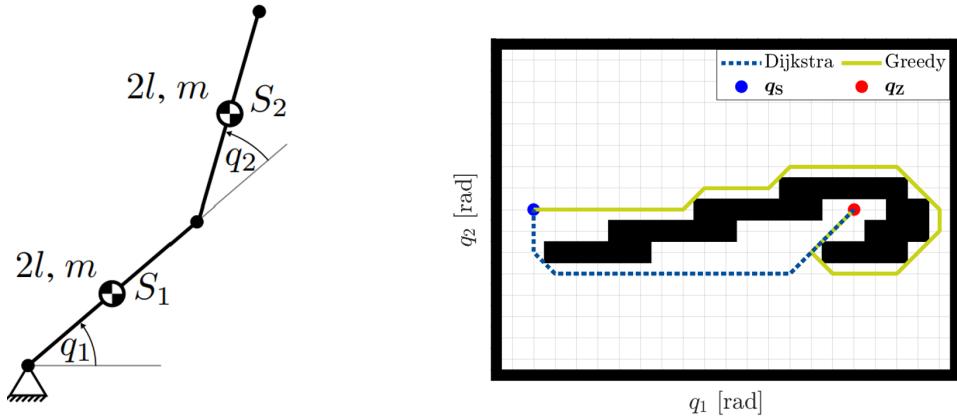


Bild 6.36: Rechts ist ein Vergleich der Planung der Greedy-Suche und des Djikstra-Algorithmus mit q_S als Startkonfiguration und q_Z als Zielkonfiguration. Dabei sind q_i die Gelenkwinkel des RR-Roboters (siehe links).

A*-Algorithmus

Der A*-Algorithmus kombiniert die Vorteile von Dijksta und Greedy zu einem leistungsstarken, heuristikbasierten Suchverfahren. Er berücksichtigt sowohl die **bisherigen Pfadkosten** als auch eine **geschätzte Restdistanz** zum Ziel. Dadurch findet A* oft schnell und zuverlässig **optimale Pfade**. Die Grundidee von A* besteht darin, jeden Knoten n mit einer Kostenfunktion zu bewerten.

$$\pi(n) = \xi(n) + \eta(n) \quad (6.48)$$

Dabei repräsentiert $\xi(n)$ die tatsächliche Kosten vom Start bis zu diesem Knoten und $\eta(n)$ die heuristische Schätzung der Kosten vom Knoten zum Ziel. Der Algorithmus wählt stets den Knoten mit dem niedrigsten $\pi(n)$ -Wert zur weiteren Exploration. Der Ablauf ist wie folgt beschrieben:

1. Der Startknoten wird mit $\pi=\xi + \eta$ bewertet und in eine Prioritätswarteschlange eingefügt.
2. In jeder Iteration wird der Knoten mit dem geringsten $\pi(n)$ -Wert entfernt und untersucht.
3. Seine Nachbarn werden analysiert und ihre ξ - und η -Werte neu berechnet.
4. Der Prozess endet, sobald der Zielknoten erreicht wurde

Der A*-Algorithmus ist **optimal**, sofern die verwendete Heuristik $\eta(n)$ zulässig ist, d. h., die tatsächlichen Kosten werden nicht unterschätzt. Er ist zudem **vollständig**, da eine Lösung immer gefunden wird, wenn die Zustandsmenge endlich ist. Im Vergleich zu Dijksta erweist sich A* als deutlich **effizienter**, da die Heuristik die Suche auf die relevanten Bereiche in der Zielrichtung fokussiert. Der **Rechenaufwand** hängt jedoch stark von der Wahl der Heuristik ab. Bild 6.37 präsentiert ein Beispiel eines gewichteten Suchbaums des A*-Algorithmus, während Bild 6.38 den Ablauf des Algorithmus darstellt.

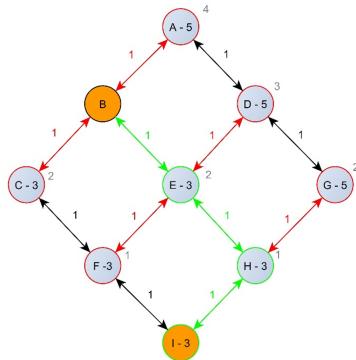


Bild 6.37: Beispiel der Pfadknoten und -kosten des A*-Algorithmus.

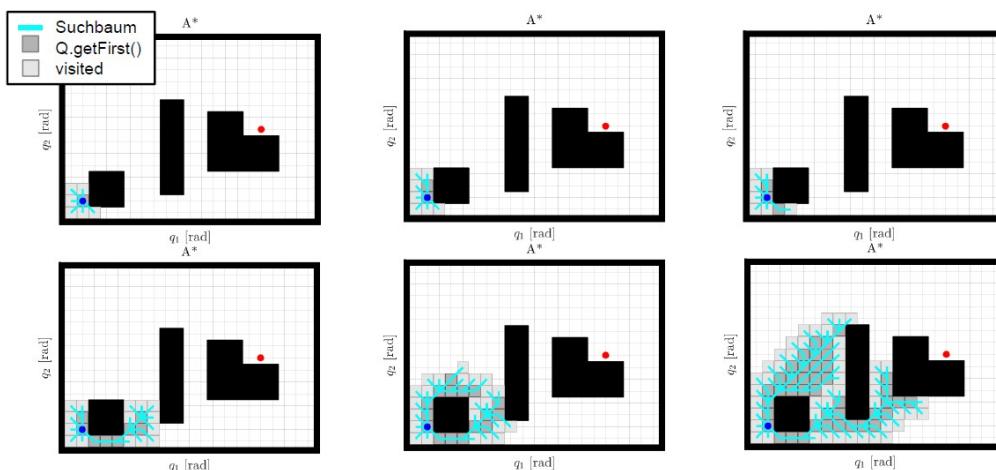


Bild 6.38: Beispielverlauf des A*-Algorithmus (Die Code-Umgebung wird im Rahmen der Veranstaltung zur Verfügung gestellt, sodass die einzelnen Schritte der Implementierung nachvollziehbar sind)

Abschließend zeigt Bild 6.39 den Vergleich der Algorithmen. Die Depth-First-Suche verfolgt einen Pfad konsequent bis zum Ende, bevor alternative Pfade betrachtet werden, erzeugt dabei jedoch sehr lange und ineffiziente Wege. Im Gegensatz dazu findet Breadth-First zwar stets einen Pfad, ist jedoch ineffizient bei großen Räumen. Darüber hinaus garantiert Dijkstra den kostenminimalen Pfad, benötigt dafür aber ebenfalls eine hohe Rechenzeit. Demgegenüber arbeitet Greedy deutlich schneller, liefert jedoch suboptimale Bahnen, da nur die momentane Zielnähe berücksichtigt wird. Schließlich kombiniert der A*-Algorithmus die Vorteile von Dijkstra und Greedy, indem er sowohl die zurückgelegten Kosten als auch eine Heuristik einbezieht, und findet dadurch effizient eine nahezu optimale Bahn.

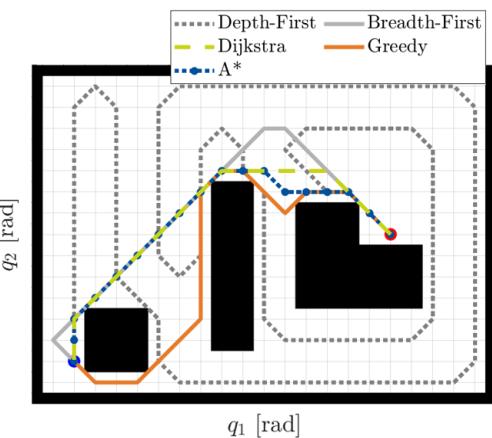


Bild 6.39: Vergleich der Bahnplanung aller Algorithmen.

7 Dynamik: Newton-Euler und Lagrange

Dieses Kapitel widmet sich dem Themengebiet der Roboterdynamik im Allgemeinen sowie den hierfür erforderlichen Werkzeugen und Methoden zur Analyse und Beschreibung. Nach einleitenden Begriffsdefinitionen und Grundlagen (Abschnitt 7.1) werden die dynamischen Bewegungsgleichungen eingeführt, deren Lösungen das Zeitverhalten eines Roboters eindeutig beschreiben.

Mit der Methode nach Newton-Euler¹ (Abschnitt 7.2) sowie den Langrange'schen Gleichungen zweiter Art (Abschnitt 7.3) werden zwei Ansätze zum Aufstellen der Bewegungsgleichungen detailliert vorgestellt. Nach einer bis zu diesem Zeitpunkt überwiegend geometrischen Analyse von Robotern werden jetzt also zusätzlich Kräfte und Drehmomente berücksichtigt. In der Praxis ist in Bezug auf die Roboterdynamik meistens folgende Frage zu beantworten: „Wie berechne ich für eine vorgegebene Bewegung die notwendigen Kräfte und/oder Momente meiner Roboterantriebe?“

Als weiterführende Literatur wird empfohlen:

- L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Berlin, Deutschland: Springer, 2000, S. 119–168
- B. Heimann, W. Gerth und K. Popp. *Mechatronik: Komponenten – Methoden – Beispiele*. München, Deutschland: Carl Hanser Verlag, 2006, S. 195–210
- W. Khalil und E. Dombre. *Modeling, Identification & Control of Robots*. New York, USA: Routledge, 2002, S. 191–256
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 187–226

7.1 Begriffsdefinitionen und Grundlagen

Die Dynamik umschließt die Lehre von der Bewegung massebehafteter Körper unter der Einwirkung von Kräften und Momenten. In der Dynamik wird somit die Verknüpfung von kinematischen Größen mit den Kraftgrößen beschrieben.

Definition
Dynamik

Strenggenommen beinhaltet die Dynamik auch die Statik, den Sonderfall von Systemen, die sich in Ruhe befinden. Die Beschreibung des Bewegungsverhaltens eines Roboters mit n Freiheitsgraden führt auf einen Satz von n Differentialgleichungen, den sogenannten Bewegungsgleichungen. Zusammengenommen bilden diese Bewegungsgleichungen das dynamische Robotermodell. Man unterscheidet bei der Beschreibung von Robotern zwei Arten von Dynamikgleichungen:

- direkte Dynamik (Vorwärtsdynamik),
- inverse Dynamik (Rückwärtssubstitution).

Direkte Dynamik (Vorwärtssubstitution)

¹I. Newton, englischer Mathematiker, Physiker, Astronom, Alchemist und Philosoph, 1643-1727;
L. Euler, schweizer Mathematiker, 1707-1783

Definition
 Direkte Dynamik

Bei der direkten Dynamik ist die Gelenkbeschleunigung $\ddot{q}(t)$ gesucht, die aus gegebenen Aktorkräften und/oder -momenten $\tau(t)$, den Gelenkwinkeln $q(t)$ und -geschwindigkeiten $\dot{q}(t)$ sowie einer am Endeffektor angreifenden Kraft \mathcal{F} resultiert (vgl. Bild 7.1).

Durch Integrieren folgen aus der Gelenkbeschleunigung direkt die Gelenkwinkel und -geschwindigkeiten. Anwendung findet die direkte Dynamik (in Kombination mit der direkten Kinematik) insbesondere bei der Simulation des Bewegungsverhaltens eines Roboters. Im Hinblick auf eine realitätsnahe Simulation ist hierbei ein möglichst exaktes Modell erforderlich.

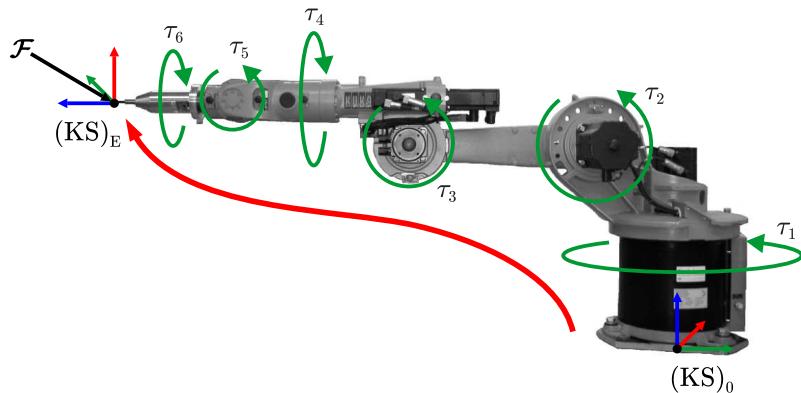


Bild 7.1: Direkte Dynamik

Bei der direkten Dynamik ist also folgender funktioneller Zusammenhang gesucht:

$$\ddot{q}(t) = f_D(q(t), \dot{q}(t), \tau(t), \mathcal{F}(t)). \quad (7.1)$$

Die Verwendung der direkten Dynamik ist in der Praxis jedoch eher unüblich, da in vielen Fällen $\ddot{q}(t)$ bereits aus der Bahnplanung bekannt ist.

Inverse Dynamik (Rückwärtodynamik)

Definition
 Inverse Kinematik

Bei der inversen Dynamik sind die Aktorkräfte und/oder momente $\tau(t)$ gesucht, die aus einer gegebenen Gelenkbewegung $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ sowie einer externen Kraft \mathcal{F} resultieren (vgl. Bild 7.2).

Anwendung findet die inverse Dynamik bei der Auslegung von Roboterstrukturen, beispielsweise zur Dimensionierung der Gelenke und Aktoren. Des Weiteren ist sie in der Robotersteuerung und -regelung von großem Nutzen.

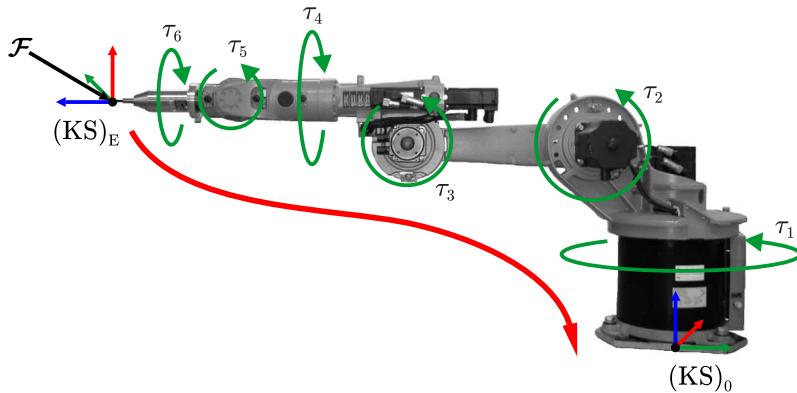
Die inverse Dynamik lautet in allgemeiner Form:

$$\tau(t) = g_D(q(t), \dot{q}(t), \ddot{q}(t), \mathcal{F}(t)). \quad (7.2)$$

Die folgenden Ausführungen beschränken sich auf die Herleitung der inversen Dynamik. Sie bildet die Grundlage für die später beschriebenen modellbasierten Regelverfahren.

Allgemeine Form der dynamischen Gleichungen

Das Differentialgleichungssystem zur Darstellung der inversen Dynamik eines Roboters mit n Freiheitsgraden (n generalisierten Koordinaten q_i) hat folgende Form (n verkoppelte Differentialgleichungen zweiter Ordnung):

**Bild 7.2:** Inverse Dynamik

$$\tau = M(q) \ddot{q} + c(q, \dot{q}) + g(q) + h(q, \dot{q})$$

↗ (n x 1) Vektor der Nichtlinearitäten (Reibung u.ä.)
 (im Folgenden vernachlässigt, später mehr)
 ↗ (n x 1) Vektor der Gravitationsanteile
 ↗ (n x 1) Vektor der Zentrifugal- und Coriolisterme
 ↗ (n x n) Trägheits- bzw. Massenmatrix
 (symmetrisch und positiv definit)
 ↗ (n x 1) Vektor der Gelenkkräfte/-momente

Die Kopplung der Differentialgleichungen ist aufgrund der Struktur des Roboters (kinematische Kette) plausibel. Im Folgenden werden zwei mögliche Methoden zum Aufstellen der Bewegungsgleichungen von Robotern vorgestellt:

- **Newton-Euler-Methode** (synthetische Methode) – Abschnitt 7.2:
Ausgangspunkt der Newton-Euler-Methode sind die freigeschnittenen Teilkörper/Glieder des Roboters. Durch Anwendung des Impuls- sowie des Drallsatzes auf jeden Teilkörper i erhält man ein System von $6n$ Gleichungen, die im Anschluss über die kinematischen Zwangsbedingungen der Struktur miteinander verknüpft werden.
- **Lagrange'sche Methode** (analytische Methode) – Abschnitt 7.3:
Im Unterschied zur Newton-Euler-Methode, bei der Kräfte und Momente zur dynamischen Modellierung herangezogen werden, bilden bei der Lagrange'schen Methode Energie- und Arbeitsbilanzen die Basis zur Aufstellung der Bewegungsgleichungen.

Im Folgenden werden starre Roboterglieder und spielfreie Gelenke angenommen.

7.2 Bewegungsgleichung nach Newton-Euler

In den nächsten Abschnitten ist das Vorgehen zur Aufstellung der dynamischen Gleichungen der Form

$$\tau = M(q) \ddot{q} + c(q, \dot{q}) + g(q) \quad (7.3)$$

nach Newton-Euler beschrieben. Es gliedert sich in sechs Teilschritte:

1. Roboter mit n Freiheitsgraden in n Teilkörper zerlegen und die Schwerpunkte S_i sämtlicher Teilkörper i im $(KS)_0$ bestimmen.

2. Ortsvektoren der Schwerpunkte $(0)\mathbf{r}_{S_i}$ nach der Zeit ableiten, um Schwerpunktsgeschwindigkeiten $(0)\dot{\mathbf{r}}_{S_i}$ und -beschleunigungen $(0)\ddot{\mathbf{r}}_{S_i}$ zu erhalten (oder alternativ in Schritt 3*).
3. Bestimmung der Winkelgeschwindigkeiten $(0)\omega_i$ und -beschleunigungen $(0)\dot{\omega}_i$ jedes Teilkörpers i , (je nach Vorgehensweise wird dieser Schritt auch mit 3* bezeichnet).
4. Jeden Teilkörper i freischneiden, seinen Massenträgheitstensor $(0)\mathbf{J}_i^{(S_i)}$ bestimmen sowie den Impuls- und Drallsatz (Drehimpulssatz) aufstellen.
5. Rekursive Berechnung der Antriebskräfte und/oder -momente (beginnend beim letzten Teilkörper n).
6. Umformen in Matrizenform (wenn gewünscht bzw. gefordert).

Schritt 1 und Schritt 2

Ein Roboter mit n Freiheitsgraden ist in n Teilkörper zu zerlegen und die jeweiligen Schwerpunkte S_i der Teilkörper i im $(KS)_0$ sind zu bestimmen (Bild 7.3). Die Berechnung der Ortsvektoren $(0)\mathbf{r}_{S_i}(\mathbf{q})$ kann dabei komfortabel unter Verwendung der homogenen Transformationsmatrizen bzw. der Denavit-Hartenberg Notation erfolgen (vgl. Kapitel 3). Anschließend müssen noch die

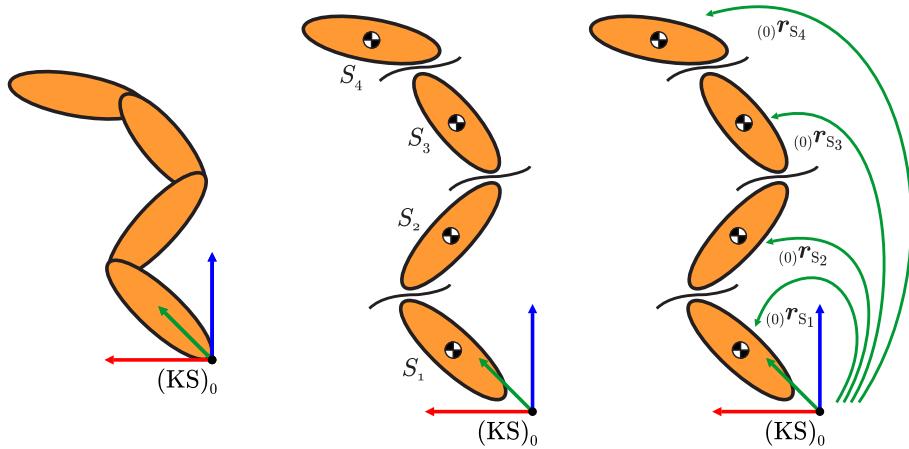


Bild 7.3: Roboter mit n Freiheitsgraden in n Teilkörper zerlegen und die Schwerpunkte S_i bestimmen

Ortsvektoren $(0)\mathbf{r}_{S_i}(\mathbf{q})$ nach der Zeit abgeleitet werden, um die Schwerpunktgeschwindigkeiten $(0)\dot{\mathbf{r}}_{S_i}(\mathbf{q})$ und -beschleunigungen $(0)\ddot{\mathbf{r}}_{S_i}(\mathbf{q})$ zu erhalten (eine Alternative zur Berechnung der Schwerpunktgeschwindigkeiten und -beschleunigungen ist in Schritt 3* dargestellt).

Zur Durchführung von Schritt 3 existieren zwei Möglichkeiten (Schritt 3, Schritt 3*), die nachstehend aufgeführt sind:

Schritt 3

Bestimmung der Winkelgeschwindigkeiten $(0)\omega_i$ und -beschleunigungen $(0)\dot{\omega}_i$ jedes Teilkörpers i im Inertialkoordinatensystem $(KS)_0$:

Der Ansatz basiert auf der zeitlichen Ableitung der Rotationsmatrizen:

$$\mathbf{S}_{((0)\omega_i)} = {}_{(0)} \left(\begin{array}{ccc} 0 & -\omega_{z_i} & \omega_{y_i} \\ \omega_{z_i} & 0 & -\omega_{x_i} \\ -\omega_{y_i} & \omega_{x_i} & 0 \end{array} \right) = {}^0 \dot{\mathbf{R}}_i {}^0 \mathbf{R}_i^T. \quad (7.4)$$

Zur Herleitung von Gleichung (7.4) zunächst eine kurze Wiederholung der **Eulerschen Differenziationsregel**:

Gegeben sei der in Abbildung 7.4 dargestellte Starrkörper. Basierend auf dem Zusammenhang

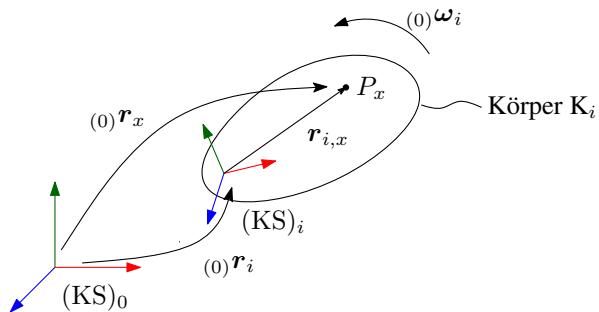


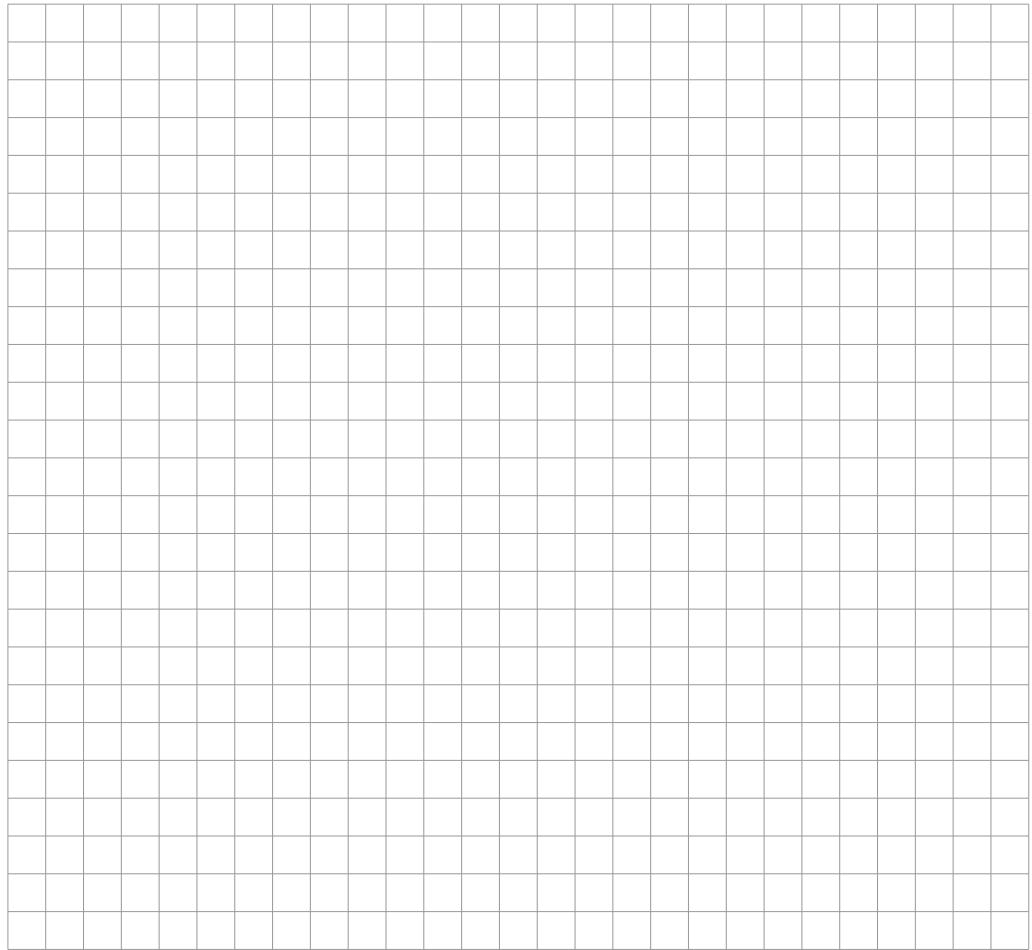
Bild 7.4: Bewegung eines Punktes P_x auf einem Starrkörper K_i in $(KS)_0$

$${}^{(0)}\boldsymbol{r}_x = {}^{(0)}\boldsymbol{r}_i + {}^{(0)}\boldsymbol{r}_{i,x} \quad (7.5)$$

gilt für die Geschwindigkeit von Punkt P_x in $(KS)_0$ nach Euler Folgendes:

$${}^{(0)}\dot{\boldsymbol{r}}_x = {}^{(0)}\dot{\boldsymbol{r}}_i + {}^{(0)}\boldsymbol{\omega}_i \times {}^{(0)}\boldsymbol{r}_{i,x}, \quad (7.6)$$

wobei	$(0)\dot{r}_x$	Geschwindigkeit von Punkt P_x in $(KS)_0$
	$(0)\dot{r}_i$	Geschwindigkeit des körperfesten $(KS)_i$ ausgedrückt im $(KS)_0$;
	$(0)r_{i,x}$	Vektor vom Ursprung $(KS)_i$ zu P_x ; in $(KS)_0$ ausgedrückt



Somit lässt sich die gesuchte Winkelgeschwindigkeit $(0)\omega_i$ wie folgt darstellen:

$$(0)\omega_i = ((0)\omega_{x_i}, (0)\omega_{y_i}, (0)\omega_{z_i})^T. \quad (7.7)$$

Abschließend ergeben sich durch Ableiten von $(0)\omega_i$ die Winkelbeschleunigungen $(0)\dot{\omega}_i$.

Die Rotationsmatrizen 0R_i sind unter Verwendung der Denavit-Hartenberg Notation leicht bestimmbar (vgl. Kapitel 3). Die in Schritt 3 betrachteten Koordinatensysteme sind Bild 7.5 zu entnehmen.

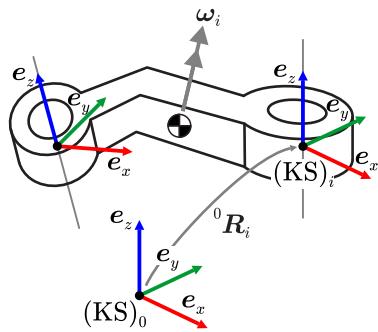


Bild 7.5: Bestimmung der Winkelgeschwindigkeiten und -beschleunigungen

Alternative: Schritt 3*

Rekursive Bestimmung der Geschwindigkeiten und Beschleunigungen sämtlicher Teilkörper i , beginnend in der Basis, die Koordinatensysteme seien nach Denavit-Hartenberg festgelegt. Die notwendigen Größen und Koordinatensysteme sind in Bild 7.6 exemplarisch für das Segment i dargestellt.

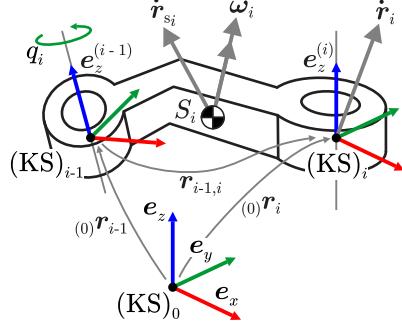


Bild 7.6: Rekursive Bestimmung der Geschwindigkeiten und Beschleunigungen

Für die Roboterbasis gilt:

$$(0)\omega_0 = (0)\dot{\omega}_0 = \mathbf{0}, \quad (0)\dot{r}_0 = (0)\ddot{r}_0 = \mathbf{0}. \quad (7.8)$$

Für Drehgelenke ergeben sich folgende Zusammenhänge (siehe auch [Kes97, S. 40ff]):

$$(0)\omega_i = (0)\omega_{i-1} + \dot{q}_i (0)e_z^{(i-1)}, \quad (7.9)$$

$$(0)\dot{\omega}_i = (0)\dot{\omega}_{i-1} + \ddot{q}_i (0)e_z^{(i-1)} + \dot{q}_i (0)\omega_{i-1} \times (0)e_z^{(i-1)}, \quad (7.10)$$

$$(0)r_i = (0)r_{i-1} + (0)r_{i-1,i}, \quad (7.11)$$

$$(0)\dot{r}_i = (0)\dot{r}_{i-1} + (0)\omega_i \times (0)r_{i-1,i}, \quad (7.12)$$

$$(0)\ddot{r}_i = (0)\ddot{r}_{i-1} + (0)\dot{\omega}_i \times (0)r_{i-1,i} + (0)\omega_i \times ((0)\omega_i \times (0)r_{i-1,i}). \quad (7.13)$$

Bei den Zeitableitungen ist die Eulersche Differentiationsregel (vgl. Tafelanschrieb) zu beachten.

Für Schubgelenke gelten folgende Zusammenhänge:

$$(0)\omega_i = (0)\omega_{i-1}, \quad (7.14)$$

$$(0)\dot{\omega}_i = (0)\dot{\omega}_{i-1}, \quad (7.15)$$

$$(0)r_i = (0)r_{i-1} + (0)r_{i-1,i}, \quad (7.16)$$

$$(0)\dot{r}_i = (0)\dot{r}_{i-1} + (0)\omega_i \times (0)r_{i-1,i} + \dot{q}_i (0)e_z^{(i-1)} \quad (7.17)$$

$$(0)\ddot{r}_i = (0)\ddot{r}_{i-1} + \ddot{q}_i (0)e_z^{(i-1)} + (0)\omega_i \times ((0)\omega_i \times (0)r_{i-1,i}) + 2\dot{q}_i (0)\omega_i \times (0)e_z^{(i-1)} + (0)\dot{\omega}_i \times (0)r_{i-1,i}. \quad (7.18)$$

Zur Erinnerung: Das Koordinatensystem $(KS)_{i-1}$ ist „feststehend“ bezüglich Teilkörper $i-1$, das Koordinatensystem $(KS)_i$ bewegt sich in Abhängigkeit von q_i .

Herleitung der Gleichungen (7.17) und (7.18); Schritt 3*

Nach Abbildung 7.6 gilt folgender Zusammenhang:

$$(0)r_i = (0)r_{i-1} + (0)r_{i-1,i}. \quad (7.19)$$

Zeitliches Ableiten führt zu:

$$(0)\ddot{\mathbf{r}}_i = \underbrace{(0)\dot{\mathbf{r}}_{i-1} + (0)\boldsymbol{\omega}_i \times (0)\mathbf{r}_{i-1,i}}_{\text{Eulersche-Diff.-Regel}} + \underbrace{\dot{q}_i(0)\mathbf{e}_z^{(i-1)}}_{\mathbf{v}_R}, \quad (7.20)$$

wobei der im Vergleich zu Gleichung (7.6) zusätzliche Geschwindigkeitsterm \mathbf{v}_R (Relativgeschwindigkeit) aus der gelenkwinkelabhängigen Relativbewegung zwischen $(KS)_i$ und $(KS)_{i-1}$ resultiert (kein „Starrkörper“).

Eine erneute zeitliche Ableitung der Gleichung (7.20) ergibt:

$$(0)\ddot{\mathbf{r}}_i = (0)\ddot{\mathbf{r}}_{i-1} + (0)\dot{\boldsymbol{\omega}}_i \times (0)\mathbf{r}_{i-1,i} + (0)\boldsymbol{\omega}_i \times (0)\dot{\mathbf{r}}_{i-1,i} + \ddot{q}_i(0)\mathbf{e}_z^{(i-1)} + \dot{q}_i(0)\dot{\mathbf{e}}_z^{(i-1)}. \quad (7.21)$$

Der Vektor $(0)\dot{\mathbf{r}}_{i-1,i}$ lässt sich aus der Beziehung (vgl. Abbildung 7.6)

$$(0)\mathbf{r}_{i-1,i} = (0)\mathbf{r}_i - (0)\mathbf{r}_{i-1} \quad (7.22)$$

durch zeitliches Differenzieren bestimmen:

$$(0)\dot{\mathbf{r}}_{i-1,i} = (0)\dot{\mathbf{r}}_i - (0)\dot{\mathbf{r}}_{i-1} \quad (7.23)$$

$$= \underbrace{(0)\dot{\mathbf{r}}_{i-1} + (0)\boldsymbol{\omega}_i \times (0)\mathbf{r}_{i-1,i} + \dot{q}_i(0)\mathbf{e}_z^{(i-1)}}_{(0)\dot{\mathbf{r}}_i \text{ (vgl. Gleichung (7.20))}} - (0)\dot{\mathbf{r}}_{i-1} \quad (7.24)$$

$$= (0)\boldsymbol{\omega}_i \times (0)\mathbf{r}_{i-1,i} + \dot{q}_i(0)\mathbf{e}_z^{(i-1)}. \quad (7.25)$$

Die Änderung $(0)\dot{\mathbf{e}}_z^{(i-1)}$ von Achse $(0)\mathbf{e}_z^{(i-1)}$ im $(KS)_0$ kann analog zu Schritt 3 ermittelt werden:

Aus der Beziehung

$$(0)\mathbf{e}_z^{(i-1)} = {}^0\mathbf{R}_{i-1(i-1)}\mathbf{e}_z^{(i-1)} \quad (7.26)$$

folgt nach Ableitung:

$$(0)\dot{\mathbf{e}}_z^{(i-1)} = {}^0\dot{\mathbf{R}}_{i-1(i-1)}\mathbf{e}_z^{(i-1)} + {}^0\mathbf{R}_{i-1}\underbrace{{}^{(i-1)}\dot{\mathbf{e}}_z^{(i-1)}}_{=0} \quad (7.27)$$

$$= {}^0\dot{\mathbf{R}}_{i-1(i-1)}\mathbf{e}_z^{(i-1)} \quad (7.28)$$

$$= \underbrace{{}^0\mathbf{R}_{i-1(i-1)}\mathbf{e}_z^{(i-1)}}_{(0)\mathbf{e}_z^{(i-1)}} = (0)\boldsymbol{\omega}_i \times (0)\mathbf{e}_z^{(i-1)} \quad (\text{vgl. Gleichung (??)}). \quad (7.29)$$

Nach Einsetzen von $(0)\dot{\mathbf{r}}_{i-1,i}$ und $(0)\dot{\mathbf{e}}_z^{(i-1)}$ in Gleichung (7.21) resultiert die Beschleunigung $(0)\ddot{\mathbf{r}}_i$ zu:

$$(0)\ddot{\mathbf{r}}_i = \overbrace{(0)\ddot{\mathbf{r}}_{i-1} + (0)\dot{\boldsymbol{\omega}}_i \times (0)\mathbf{r}_{i-1,i} + (0)\boldsymbol{\omega}_i \times ((0)\boldsymbol{\omega}_i \times (0)\mathbf{r}_{i-1,i})}^{\mathbf{a}_F} \quad (7.30)$$

$$+ \underbrace{\ddot{q}_i(0)\mathbf{e}_z^{(i-1)}}_{\mathbf{a}_R} + \underbrace{2(0)\boldsymbol{\omega}_i \times \dot{q}_i(0)\mathbf{e}_z^{(i-1)}}_{\mathbf{a}_C}, \quad (7.31)$$

mit

\mathbf{a}_F : Führungsbeschleunigung,

\mathbf{a}_R : Relativbeschleunigung,

\mathbf{a}_C : Coriolisbeschleunigung.

Abschließend erfolgt die Transformation der translatorischen Geschwindigkeiten ${}_{(0)}\dot{\mathbf{r}}_i$ sowie der Beschleunigungen ${}_{(0)}\ddot{\mathbf{r}}_i$ auf die Schwerpunktlage S_i der Teilkörper i . Die für die Umrechnung der Geschwindigkeiten und Beschleunigungen erforderlichen Größen sind in Bild 7.7 dargestellt.

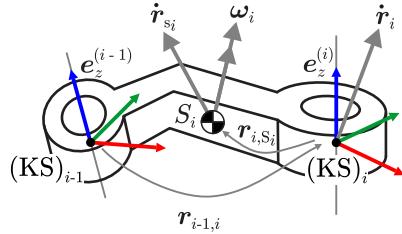


Bild 7.7: Transformation der Geschwindigkeiten und Beschleunigungen auf die Schwerpunktlage

Für die Schwerpunktgeschwindigkeiten ergeben sich somit:

$${}_{(0)}\dot{\mathbf{r}}_{S_i} = {}_{(0)}\dot{\mathbf{r}}_i + {}_{(0)}\omega_i \times {}_{(0)}\mathbf{r}_{i,S_i}, \quad (7.32)$$

die Schwerpunktbeschleunigungen berechnen sich zu:

$${}_{(0)}\ddot{\mathbf{r}}_{S_i} = {}_{(0)}\ddot{\mathbf{r}}_i + {}_{(0)}\dot{\omega}_i \times {}_{(0)}\mathbf{r}_{i,S_i} + {}_{(0)}\omega_i \times ({}_{(0)}\omega_i \times {}_{(0)}\mathbf{r}_{i,S_i}). \quad (7.33)$$

Schritt 4

Jeden Teilkörper i freischneiden (vgl. Bild 7.8), seinen Massenträgheitstensor ${}_{(0)}\mathbf{J}_i^{(S_i)}$ bestimmen sowie den Impuls- und Drehimpulssatz (Drallsatz) aufstellen:

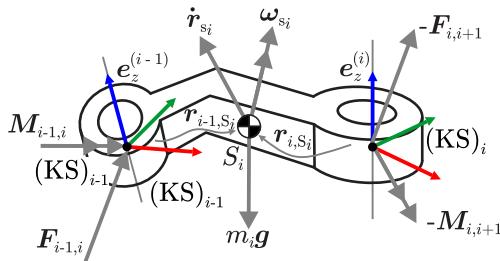


Bild 7.8: Freigeschalteter Teilkörper i

Schritt 4.1

Zunächst wird das Vorgehen zum Aufstellen des Massenträgheitstensors beschrieben:

Der Massenträgheitstensor des Körpers i um Schwerpunkt S_i in $(KS)_i$ lautet:

$${}_{(i)}\mathbf{J}_i^{(S_i)} = {}_{(i)}\begin{pmatrix} J_{xx_i}^{(S_i)} & J_{xy_i}^{(S_i)} & J_{xz_i}^{(S_i)}, \\ J_{yx_i}^{(S_i)} & J_{yy_i}^{(S_i)} & J_{yz_i}^{(S_i)}, \\ J_{zx_i}^{(S_i)} & J_{zy_i}^{(S_i)} & J_{zz_i}^{(S_i)}. \end{pmatrix} = \text{konst.} \quad (7.34)$$

Der Massenträgheitstensor gibt die Trägheit eines starren Körpers gegenüber einer Änderung seiner Rotationsbewegung an. In der Regel erhält man den Massenträgheitstensor aus CAD-Daten. Ist dies nicht möglich, so können die Elemente des Massenträgheitstensors auch wie folgt berechnet werden (vgl. TM 3 bzw. TM 2 für Elektrotechnik).

Massenträgheitsmomente (Diagonalelemente):

$${}_{(i)}J_{xx_i}^{(S_i)} = \int_{K_i} (({}_i)y^2 + {}_{(i)}z^2) dm_i, \quad (7.35)$$

$${}_{(i)}J_{yy_i}^{(S_i)} = \int_{K_i} (({}_i)z^2 + {}_{(i)}x^2) dm_i, \quad (7.36)$$

$${}_{(i)}J_{zz_i}^{(S_i)} = \int_{K_i} (({}_i)x^2 + {}_{(i)}y^2) dm_i. \quad (7.37)$$

Die Massenträgheitsmomente geben die Trägheit eines starren Körpers K_i gegenüber einer Änderung seiner Rotationsbewegung um die Hauptträgheitsachsen an.

Deviationsmomente (Nebendiagonalelemente):

$${}_{(i)}J_{xy_i}^{(S_i)} = {}_{(i)}J_{yx_i}^{(S_i)} = -\int_{K_i} (({}_i)x {}_{(i)}y) dm_i, \quad (7.38)$$

$${}_{(i)}J_{yz_i}^{(S_i)} = {}_{(i)}J_{zy_i}^{(S_i)} = -\int_{K_i} (({}_i)y {}_{(i)}z) dm_i, \quad (7.39)$$

$${}_{(i)}J_{zx_i}^{(S_i)} = {}_{(i)}J_{xz_i}^{(S_i)} = -\int_{K_i} (({}_i)z {}_{(i)}x) dm_i. \quad (7.40)$$

Die Deviationsmomente sind immer dann ungleich null, wenn die Hauptträgheitsachsen nicht als körperfeste Koordinatenachsen gewählt werden.

Abschließend erfolgt die Transformation des im Koordinatensystem $(KS)_i$ ausgedrückten Massenträgheitstensors ${}_{(i)}J_i^{(S_i)}$ des Teilkörpers i um seinen Schwerpunkt S_i in das Inertialsystem $(KS)_0$ (vgl. Bild 7.9):

$${}_{(0)}J_i^{(S_i)} = \underbrace{{}_{(0)}R_i}_{\text{Basiswechsel}} {}_{(i)}J_i^{(S_i)} {}_{(0)}R_i^T. \quad (7.41)$$

Dies entspricht einem Basiswechsel (nur Rotation, keine Translation), also der Änderung der Achsen des Koordinatensystems, in dem der Massenträgheitstensor definiert ist. Dieser Schritt ist erforderlich, da eine Beschreibung des Systems im $(KS)_0$ angestrebt wird.

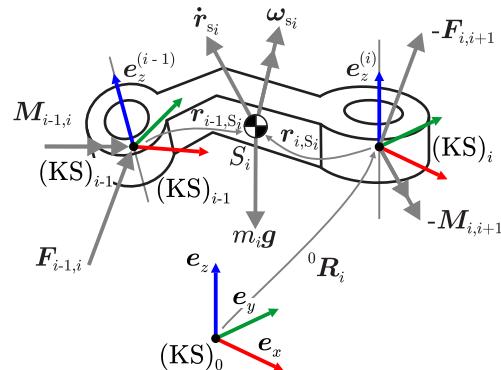


Bild 7.9: Transformation des Massenträgheitstensors in das $(KS)_0$

Schritt 4.2

Ist der Massenträgheitstensor bekannt, können Impuls- und Drallsatz für sämtliche Teilkörper aufgestellt werden. Ausgedrückt in vektorieller Schreibweise lauten diese wie folgt:

Impulssatz (Newton-Gleichung):

$$\begin{aligned} {}^{(0)}\mathbf{F}_i &= m_i {}^{(0)}\ddot{\mathbf{r}}_{S_i} \\ &= {}^{(0)}\mathbf{F}_{i-1,i} - {}^{(0)}\mathbf{F}_{i,i+1} + m_i {}^{(0)}\mathbf{g}. \end{aligned} \quad (7.42)$$

Diese Gleichung drückt aus, dass sowohl die vom Teilkörper $i - 1$ auf i einwirkende $({}^{(0)}\mathbf{F}_{i-1,i})$ als auch die vom Glied i auf das Glied $i + 1$ ausgeübte Kraft $({}^{(0)}\mathbf{F}_{i,i+1})$ zu berücksichtigen sind. Zudem ist zu beachten, dass der Gravitationsvektor $({}^{(0)}\mathbf{g})$ im Basiskoordinatensystem ausgedrückt wird.

Drallsatz (Euler-Gleichung):

$$\begin{aligned} {}^{(0)}\mathbf{M}_i^{(S_i)} &= {}^{(0)}\mathbf{J}_i^{(S_i)} {}^{(0)}\dot{\omega}_i + {}^{(0)}\boldsymbol{\omega}_i \times \left({}^{(0)}\mathbf{J}_i^{(S_i)} {}^{(0)}\boldsymbol{\omega}_i \right) \\ &= {}^{(0)}\mathbf{M}_{i-1,i} - {}^{(0)}\mathbf{M}_{i,i+1} + {}^{(0)}\mathbf{r}_{i,S_i} \times {}^{(0)}\mathbf{F}_{i,i+1} - {}^{(0)}\mathbf{r}_{i-1,S_i} \times {}^{(0)}\mathbf{F}_{i-1,i}. \end{aligned} \quad (7.43)$$

Auch beim Drallsatz müssen, analog zum Impulssatz, sämtliche durch benachbarte Teilkörper hervorgerufenen Momente sowie Kräfte, die nicht am Schwerpunkt angreifen, berücksichtigt werden. Es folgt die rekursive Berechnung der unbekannten Kräfte und Momente, ausgehend von der Spitze bzw. vom Endeffektor des Roboters hin zur Basis (Schritt 5).

Schritt 5

Rekursive Berechnung der Antriebskräfte und/oder -momente (beginnend beim letzten Teilkörper n). Hierfür sind im ersten Schritt die Impulssätze aufzustellen und anschließend rekursiv alle unbekannten Kräfte zu eliminieren:

Ein praktikables Vorgehen besteht darin, die unbekannte Kraft zwischen Teilkörper $n - 1$ und Teilkörper n (also $({}^{(0)}\mathbf{F}_{n-1,n})$) als erstes zu berechnen. Diese kann dann in den Impulssatz bezüglich des Teilkörpers $n - 1$ eingesetzt werden, woraufhin diese Gleichung wiederum nach der unbekannten Kraft $({}^{(0)}\mathbf{F}_{n-2,n-1})$ umgestellt wird. Da dieses Verfahren schematisch von der Roboterspitze bzw. dem Endeffektor zur Basis durchgeführt wird, spricht man von einer rekursiven Berechnung. Zu beachten ist, dass es sich bei $({}^{(0)}\mathbf{F}_{n,n+1})$ um eine externe, am Endeffektor angreifende Kraft handelt. Dies wird durch den Index $n + 1$ verdeutlicht (der Roboter hat nur n Teilkörper). Das Verfahren ist nachstehend nochmals dargestellt:

$$\begin{aligned} m_n {}^{(0)}\ddot{\mathbf{r}}_{S_n} &= {}^{(0)}\mathbf{F}_{n-1,n} - \overbrace{{}^{(0)}\mathbf{F}_{n,n+1}}^{\text{externe Kraft}} + m_n {}^{(0)}\mathbf{g} \\ \Rightarrow {}^{(0)}\mathbf{F}_{n-1,n} &= m_n {}^{(0)}\ddot{\mathbf{r}}_{S_n} + {}^{(0)}\mathbf{F}_{n,n+1} - m_n {}^{(0)}\mathbf{g}, \end{aligned}$$

$$\begin{aligned} m_{n-1} {}^{(0)}\ddot{\mathbf{r}}_{S_{n-1}} &= {}^{(0)}\mathbf{F}_{n-2,n-1} - {}^{(0)}\mathbf{F}_{n-1,n} + m_{n-1} {}^{(0)}\mathbf{g} \\ \Rightarrow {}^{(0)}\mathbf{F}_{n-2,n-1} &= m_{n-1} {}^{(0)}\ddot{\mathbf{r}}_{S_{n-1}} + {}^{(0)}\mathbf{F}_{n-1,n} - m_{n-1} {}^{(0)}\mathbf{g}, \end{aligned}$$

$$\begin{aligned} m_{n-2} {}^{(0)}\ddot{\mathbf{r}}_{S_{n-2}} &= {}^{(0)}\mathbf{F}_{n-3,n-2} - {}^{(0)}\mathbf{F}_{n-2,n-1} + m_{n-2} {}^{(0)}\mathbf{g} \\ \Rightarrow {}^{(0)}\mathbf{F}_{n-3,n-2} &= \dots, \end{aligned}$$

...

Das gleiche Verfahren findet auch bezüglich der aufgestellten Drallsätze zur Bestimmung der unbekannten Momente Anwendung. Hier sind zunächst die Drallsätze aufzustellen und anschließend rekursiv alle unbekannten Kräfte und Momente zu eliminieren:

$$\begin{aligned} & {}^{(0)}\boldsymbol{J}_n^{(\text{S}_n)} {}^{(0)}\dot{\boldsymbol{\omega}}_n + {}^{(0)}\boldsymbol{\omega}_n \times \left({}^{(0)}\boldsymbol{J}_n^{(\text{S}_n)} {}^{(0)}\boldsymbol{\omega}_n \right) \\ & = {}^{(0)}\boldsymbol{M}_{n-1,n} - \underbrace{{}^{(0)}\boldsymbol{M}_{n,n+1}}_{\text{externes Moment}} + {}^{(0)}\boldsymbol{r}_{n,\text{S}_n} \times \underbrace{{}^{(0)}\boldsymbol{F}_{n,n+1}}_{\text{externe Kraft}} - {}^{(0)}\boldsymbol{r}_{n-1,\text{S}_n} \times {}^{(0)}\boldsymbol{F}_{n-1,n} \\ & \Rightarrow {}^{(0)}\boldsymbol{M}_{n-1,n} = \dots, \end{aligned}$$

$$\begin{aligned} & {}^{(0)}\boldsymbol{J}_{n-1}^{(\text{S}_{n-1})} {}^{(0)}\dot{\boldsymbol{\omega}}_{n-1} + {}^{(0)}\boldsymbol{\omega}_{n-1} \times \left({}^{(0)}\boldsymbol{J}_{n-1}^{(\text{S}_{n-1})} {}^{(0)}\boldsymbol{\omega}_{n-1} \right) \\ & = {}^{(0)}\boldsymbol{M}_{n-2,n-1} - {}^{(0)}\boldsymbol{M}_{n-1,n} + {}^{(0)}\boldsymbol{r}_{n-1,\text{S}_{n-1}} \times {}^{(0)}\boldsymbol{F}_{n-1,n} - {}^{(0)}\boldsymbol{r}_{n-2,\text{S}_{n-1}} \times {}^{(0)}\boldsymbol{F}_{n-2,n-1} \\ & \Rightarrow {}^{(0)}\boldsymbol{M}_{n-2,n-1} = \dots, \end{aligned}$$

...

Auf diese Weise lassen sich sämtliche unbekannten Kräfte und Momente zwischen den Teilkörpern bestimmen:

$${}^{(0)}\boldsymbol{F}_{i-1,i} = \dots, \quad \forall i = 1, \dots, n, \quad (7.44)$$

$${}^{(0)}\boldsymbol{M}_{i-1,i} = \dots, \quad \forall i = 1, \dots, n. \quad (7.45)$$

Um abschließend zu ermitteln, welchen Anteil der Kräfte bzw. Momente der jeweilige Antrieb zur Verfügung stellen muss, erfolgt eine Abbildung auf die Achse, an der der Antrieb eines Teilkörpers angreift. Dabei wird unterschieden zwischen Rotations und Linearantrieben. Geht man von Koordinatensystemen nach Denavit-Hartenberg aus, bewegt der Motor des Teilkörpers i diesen entlang oder um die Achse \mathbf{z}_{i-1} . Das Skalarprodukt bringt die geforderte Abbildung auf diese Achse zum Ausdruck. Gelenk i ist ein Rotationsantrieb:

$$\tau_i = {}^{(0)}\boldsymbol{M}_{i-1,i}^T {}^{(0)}\boldsymbol{e}_z^{(i-1)}. \quad (7.46)$$

Gelenk i ist ein Linearantrieb:

$$\tau_i = {}^{(0)}\boldsymbol{F}_{i-1,i}^T {}^{(0)}\boldsymbol{e}_z^{(i-1)}. \quad (7.47)$$

Schritt 6

Umformen in Matrizenform (wenn gewünscht bzw. gefordert):

Abschließend können die Antriebskräfte und/oder -momente in die allgemeine Form der Bewegungsgleichung eines Roboters (Matrizenform) überführt werden. Man erhält dann:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}). \quad (7.48)$$

7.3 Bewegungsgleichung nach Lagrange

Die Lagrange'schen Gleichungen zweiter Art ermöglichen, im Gegensatz zum Vorgehen nach Newton-Euler, das Aufstellen der Bewegungsgleichung ausgehend von der Energie eines Systems. Insbesondere bei komplexen Systemen erfordert diese Methode geringeren Aufwand als das

bisher vorgestellte Verfahren, ist jedoch deutlich abstrakter. Zu beachten ist, dass beim Vorgehen nach Newton-Euler externe, am Roboter angreifende Kräfte und Momente implizit berücksichtigt werden. Bei Anwendung der Lagrange'schen Gleichungen zweiter Art müssen diese explizit in einem zusätzlichen Schritt (8) eingebracht werden.

Wie zuvor, das allgemeine Vorgehen zur Aufstellung der dynamischen Gleichungen

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \mathbf{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \mathbf{g}(\boldsymbol{q}) - \mathbf{J}^T(\boldsymbol{q}) \mathcal{F} \quad (7.49)$$

nach Lagrange in knapper Zusammenfassung der Teilschritte. Es sei darauf hingewiesen, dass das im Folgenden vorgestellte Verfahren bis einschließlich des vierten Teilschritts ähnlich zur Methode nach Newton-Euler ist:

1. Roboter mit n Freiheitsgraden in n Teilkörper zerlegen und die Schwerpunkte S_i sämtlicher Teilkörper i im $(KS)_0$ bestimmen
2. Ortsvektoren der Schwerpunkte $(0)\mathbf{r}_{S_i}$ nach der Zeit ableiten, um Schwerpunktsgeschwindigkeiten $(0)\dot{\mathbf{r}}_{S_i}$ zu erhalten (oder alternativ in Schritt 3*)
3. Bestimmung der Winkelgeschwindigkeiten $(0)\boldsymbol{\omega}_i$ jedes Teilkörpers i
4. Massenträgheitstensor $(0)\mathbf{J}_i^{(S_i)}$ jedes Teilkörpers i bestimmen

→ bis hierher: siehe **Newton-Euler**

5. Berechnung der Systemenergie (potentielle Energie U und kinetische Energie T) in Abhängigkeit der generalisierten Koordinaten \boldsymbol{q}
6. Anwendung der Lagrange'schen Gleichungen zweiter Art
7. Umformen in Matrizenform (wenn gewünscht bzw. gefordert)
8. Gegebenenfalls Berücksichtigung externer Kräfte und Momente \mathcal{F}

Im Folgenden werden die Schritte 5 bis 8 detailliert beschrieben.

Schritt 5

Berechnung der Systemenergie (potentielle Energie U und kinetische Energie T) in Abhängigkeit der generalisierten Koordinaten \boldsymbol{q} :

Die kinetische Energie T des Gesamtsystems besteht aus einem Translations- und einem Rotationsanteil. Sie setzt sich zusammen aus den kinetischen Energien jedes Teilkörpers i :

$$T = \frac{1}{2} \sum_{i=1}^n \left(\underbrace{m_i (0) \dot{\mathbf{r}}_{S_i}^T (0) \dot{\mathbf{r}}_{S_i}}_{\text{Translationsanteil}} + \underbrace{(0) \boldsymbol{\omega}_i^T (0) \mathbf{J}_i^{(S_i)} (0) \boldsymbol{\omega}_i}_{\text{Rotationsanteil}} \right). \quad (7.50)$$

Analog dazu setzt sich auch die potentielle Energie U des Gesamtsystems aus den potenziellen Energien jedes Teilkörpers i zusammen:

$$U = - \sum_{i=1}^n (m_i (0) \mathbf{g}^T (0) \mathbf{r}_{S_i}). \quad (7.51)$$

Schritt 6

Anwendung der Lagrange'schen Gleichungen zweiter Art zur Berechnung der Antriebskräfte und/oder -momente:

Zunächst muss die Lagrange'sche Funktion L aufgestellt werden, die die kinetische und potentielle Energie des Systems beinhaltet. Die Lagrange'sche Funktion L lautet:

$$L = T - U. \quad (7.52)$$

Die gesuchten Antriebskräfte und/oder -momente resultieren dann aus der Ableitung dieser skalaren Funktion nach der Zeit sowie nach den generalisierten Koordinaten und deren Zeitableitung.

Die Antriebskräfte und/oder -momente berechnen sich wie folgt zu:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau. \quad (7.53)$$

Dieses Verfahren kann auch für einen einzelnen Antrieb i durchgeführt werden. Das Antriebsmoment τ_i berechnet sich nach:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i. \quad (7.54)$$

Schritt 7

Abschließend können die Antriebskräfte und/oder -momente in die allgemeine Form der Bewegungsgleichung eines Roboters (Matrizenform) überführt werden, wenn dies gewünscht bzw. gefordert ist:

$$\tau = M(\mathbf{q}) \ddot{\mathbf{q}} + c(\mathbf{q}, \dot{\mathbf{q}}) + g(\mathbf{q}). \quad (7.55)$$

Schritt 8

Im Gegensatz zu Newton-Euler müssen bei Lagrange externe Kräfte und Momente \mathcal{F} , d. h. die Wechselwirkung des Endeffektors mit der Umwelt, explizit berücksichtigt werden (Bild 7.10). Dies ist durch eine reine Energiebetrachtung nicht möglich. Aus Kapitel 4 ist jedoch folgende

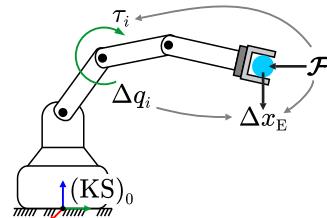


Bild 7.10: Externer Kraftangriff am Endeffektor

Beziehung bekannt:

$$\boldsymbol{\tau}_{\text{ext}}(\mathbf{q}) = \mathbf{J}^T(\mathbf{q}) \mathcal{F}. \quad (7.56)$$

Die Berücksichtigung bzw. Kompensation der externen Kräfte und Momente erfolgt bei Verwendung der Lagrange'schen Methode durch einfache Subtraktion des Vektors $\boldsymbol{\tau}_{\text{ext}}$:

$$\boldsymbol{\tau} = M(\mathbf{q}) \ddot{\mathbf{q}} + c(\mathbf{q}, \dot{\mathbf{q}}) + g(\mathbf{q}) - \boldsymbol{\tau}_{\text{ext}}(\mathbf{q}). \quad (7.57)$$

Beispiel

Planarer RR-Roboter

Folgendes Beispiel veranschaulicht das Vorgehen zum Aufstellen der Bewegungsgleichung

mit Hilfe der Langrange'schen Gleichungen zweiter Art an einem planaren RR-Roboter (siehe Bild 7.11).

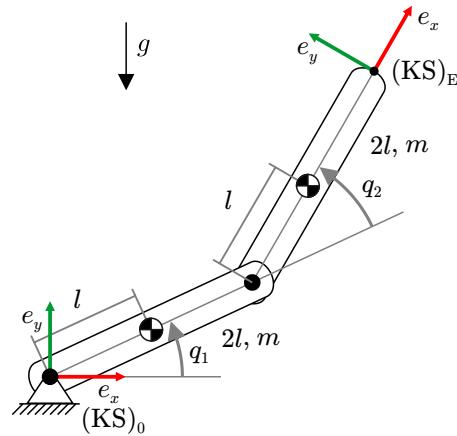
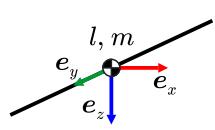
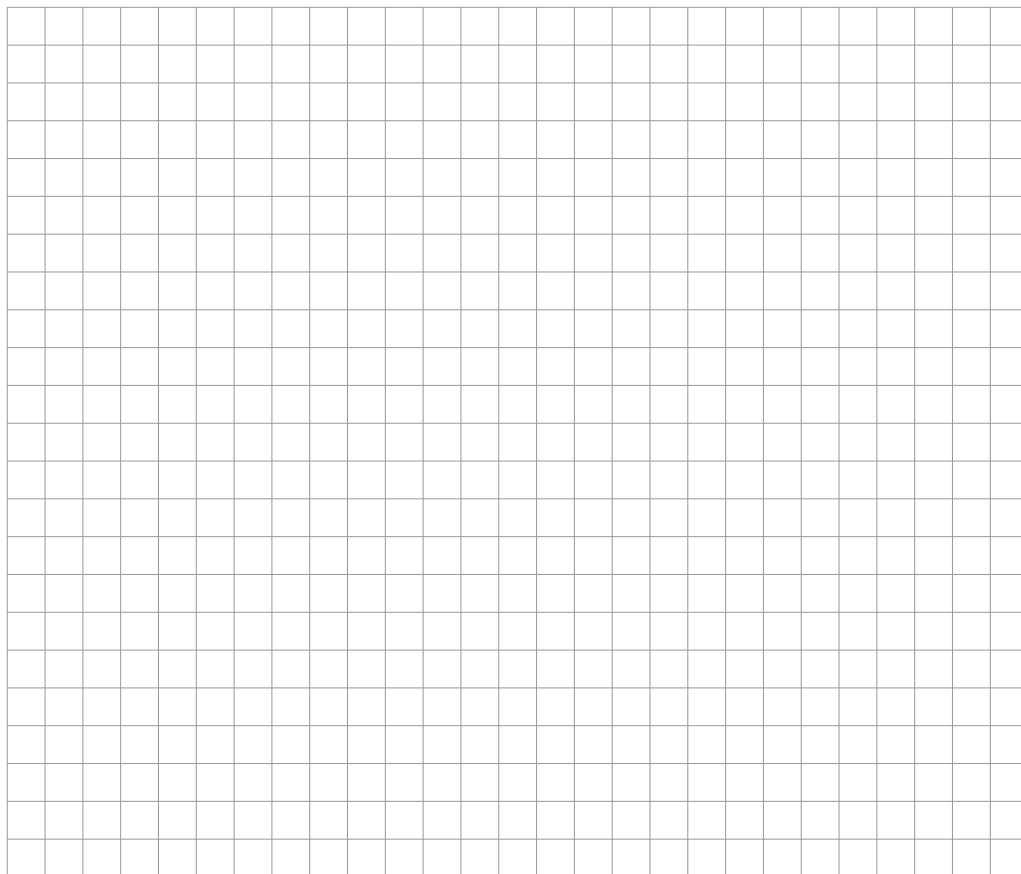


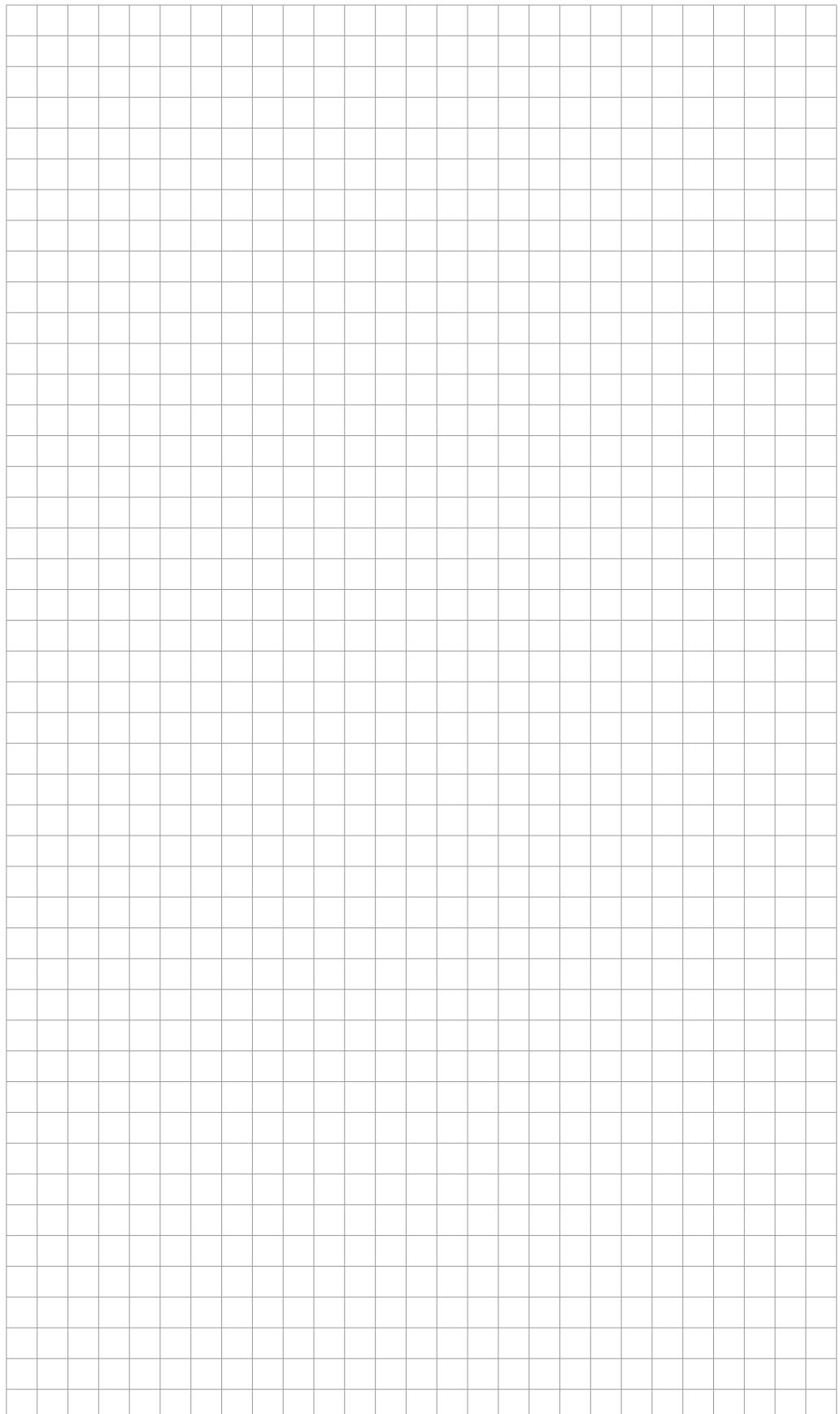
Bild 7.11: Planarer RR-Roboter

Die Roboterglieder sind vereinfacht als homogene Stäbe zu betrachten, mit nachstehenden Elementen des Massenträgheitstensors:



$$\begin{aligned} J_{xx}^{(S)} &= J_{zz}^{(S)} = \frac{1}{12} m l^2, \\ J_{yy}^{(S)} &= 0, \\ J_{xy}^{(S)} &= J_{yz}^{(S)} = J_{zx}^{(S)} = J_{yx}^{(S)} = J_{zy}^{(S)} = J_{xz}^{(S)} = 0. \end{aligned}$$





Somit ergibt sich für die kinetische Energie:

$$T = \frac{1}{2} \left(l^2 m (s_1^2 \dot{q}_1^2 + c_1^2 \dot{q}_1^2) + l^2 m (2s_1 \dot{q}_1 + s_{12} (\dot{q}_1 + \dot{q}_2))^2 + l^2 m (2c_1 \dot{q}_1 + c_{12} (\dot{q}_1 + \dot{q}_2))^2 + \frac{1}{3} ml^2 (\dot{q}_1^2 + (\dot{q}_1 + \dot{q}_2)^2) \right) \quad (7.58)$$

$$= \frac{1}{2} ml^2 \left(\dot{q}_1^2 + 4s_1^2 \dot{q}_1^2 + 4s_1 s_{12} \dot{q}_1 (\dot{q}_1 + \dot{q}_2) + s_{12}^2 (\dot{q}_1 + \dot{q}_2)^2 + 4c_1^2 \dot{q}_1^2 + 4c_1 c_{12} \dot{q}_1 (\dot{q}_1 + \dot{q}_2) + c_{12}^2 (\dot{q}_1 + \dot{q}_2)^2 \right) + \frac{1}{6} ml^2 (\dot{q}_1^2 + \dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \quad (7.59)$$

$$= \frac{1}{2} ml^2 \left(\dot{q}_1^2 + 4\dot{q}_1^2 + (\dot{q}_1 + \dot{q}_2)^2 + 4\dot{q}_1 (\dot{q}_1 + \dot{q}_2) (s_1 s_{12} + c_1 c_{12}) \right) + \frac{1}{6} ml^2 (2\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \quad (7.60)$$

$$= \frac{1}{2} ml^2 \left(5\dot{q}_1^2 + \dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2 + 4\dot{q}_1 (\dot{q}_1 + \dot{q}_2) c_2 \right) + \frac{1}{6} ml^2 (2\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \quad (7.61)$$

$$= ml^2 \left(\dot{q}_1^2 \left(3 + \frac{1}{3} + 2c_2 \right) + \dot{q}_2^2 \left(\frac{1}{2} + \frac{1}{6} \right) + \dot{q}_1 \dot{q}_2 \left(1 + 2c_2 + \frac{1}{3} \right) \right) \quad (7.62)$$

$$= ml^2 \left(\left(\frac{10}{3} + 2c_2 \right) \dot{q}_1^2 + \frac{2}{3} \dot{q}_2^2 + \left(\frac{4}{3} + 2c_2 \right) \dot{q}_1 \dot{q}_2 \right). \quad (7.63)$$

Die Lagrange'sche Funktion L lautet:

$$L = T - U \quad (7.64)$$

$$= ml^2 \left(\left(\frac{10}{3} + 2c_2 \right) \dot{q}_1^2 + \frac{2}{3} \dot{q}_2^2 + \left(\frac{4}{3} + 2c_2 \right) \dot{q}_1 \dot{q}_2 \right) - mgl (3s_1 + s_{12}). \quad (7.65)$$

Die Bewegungsgleichung für q_1 berechnet sich nach

$$\tau_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1}, \quad (7.66)$$

die Zwischenschritte

$$\frac{\partial L}{\partial \dot{q}_1} = ml^2 \left(\left(\frac{10}{3} + 2c_2 \right) 2\dot{q}_1 + \left(\frac{4}{3} + 2c_2 \right) \dot{q}_2 \right) \quad (7.67)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} = ml^2 \left(\left(\frac{20}{3} + 4c_2 \right) \ddot{q}_1 + 4\dot{q}_1 (-s_2) \dot{q}_2 + \left(\frac{4}{3} + 2c_2 \right) \ddot{q}_2 + 2\dot{q}_2 (-s_2) \dot{q}_2 \right) \quad (7.68)$$

$$= ml^2 \left(\left(\frac{20}{3} + 4c_2 \right) \ddot{q}_1 - 4\dot{q}_1 \dot{q}_2 s_2 + \left(\frac{4}{3} + 2c_2 \right) \ddot{q}_2 - 2\dot{q}_2^2 s_2 \right) \quad (7.69)$$

$$\frac{\partial L}{\partial q_1} = -mgl (3c_1 + c_{12}) \quad (7.70)$$

führen zu:

$$\tau_1 = ml^2 \left(\left(\frac{20}{3} + 4c_2 \right) \ddot{q}_1 - 4\dot{q}_1 \dot{q}_2 s_2 + \left(\frac{4}{3} + 2c_2 \right) \ddot{q}_2 - 2\dot{q}_2^2 s_2 \right) + mgl (3c_1 + c_{12}). \quad (7.71)$$

Die Berechnung der Bewegungsgleichung für q_2 ist analog.

7.4 Berücksichtigung von Motor und Getriebe

Das folgende Unterkapitel beschreibt, in welcher Weise Motoren und Getriebe die Dynamik eines Roboters beeinflussen. Es wird erläutert, mit welchen Mitteln diese (parasitären) Effekte

mathematisch modelliert werden können und auf welche Art das Robotermodell erweitert werden muss, um es dem Verhalten eines realen Roboters bestmöglich anzupassen.

Im Folgenden wird dargelegt, wie eine Getriebeübersetzung die auftretenden Motordrehmomente ändert (Abschnitt 7.4.1). Die mit dem Einsatz von Getrieben unweigerlich verbundenen Reibungseffekte und deren Modellierung werden anschließend in Abschnitt 7.4.2 behandelt. Die Betrachtung der Trägheiten von Motor und Getriebe (Abschnitt 7.4.3) schließt das Kapitel ab.

7.4.1 Getriebeübersetzung

Das auf der Abtriebsseite anliegende Moment τ_i kann mittels der Getriebeübersetzung u_{G_i} auf die Antriebsseite τ_{M_i} (die Motorseite) umgerechnet werden. Es gilt:

$$\tau_{M_i} = \frac{1}{u_{G_i}} \tau_i. \quad (7.72)$$

Somit ergibt sich für das Motordrehmoment, unter Berücksichtigung der Dynamik:

$$\begin{aligned} \boldsymbol{\tau}_M &= \text{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) \boldsymbol{\tau} \\ &= \text{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) (\boldsymbol{M}(q) \ddot{q} + \boldsymbol{c}(q, \dot{q}) + \boldsymbol{g}(q)). \end{aligned} \quad (7.73)$$

Der dazugehörige Antriebsstrang einer Roboterachse ist in Abbildung Bild 7.12 vereinfacht dargestellt, er besteht im Wesentlichen aus Motor, Getriebe und anschließendem Roboterglied.

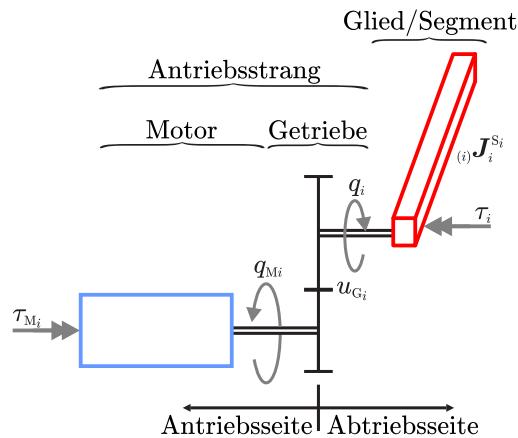


Bild 7.12: Antriebsstrang einer Roboterachse (Rotationsantrieb)

7.4.2 Reibeffekte

Jedes Gelenk und jeder Antrieb ist mit Reibung behaftet. Diese Reibung spielt eine zentrale Rolle bei der dynamischen Modellierung eines Roboters. Die Nichtbeachtung vorhandener, nichtlinearer Reibeffekte resultiert in einer negativen Beeinträchtigung des gesamten Systemverhaltens (statische Positionierfehler, unerwünschte Verzögerungen, usw.). Reibung spielt insbesondere bei langsamten Roboterbewegungen eine Rolle (Vermeidung des Slip-Stick-Effekts). In der Regel werden zwei Arten von Reibung unterschieden:

- Coulomb'sche Reibung (Gleitreibung) und
- viskose Reibung (Dämpfung).

Diese werden nun im Detail erläutert.

Für jedes Gelenk i ergibt sich die Gesamtreibung h_i als Addition der Coulomb'schen Reibung h_{c_i} und der viskosen, geschwindigkeitsabhängigen Reibung h_{v_i} :

$$h_i(q_i, \dot{q}_i) = \underbrace{r_{c_i} \operatorname{sign}(\dot{q}_i)}_{h_{c_i}} + \underbrace{r_{v_i} \dot{q}_i}_{h_{v_i}}. \quad (7.74)$$

Die zwei Arten von Reibung können mathematisch wie folgt modelliert werden:

- Coulomb'sche Reibung: $h_{c_i} = r_{c_i} \operatorname{sign}(\dot{q}_i)$ (siehe Bild 7.13(a)),
- viskose Reibung: $h_{v_i} = r_{v_i} \dot{q}_i$ (siehe Bild 7.13(b)),
- Gesamtreibung: $h_i = h_{c_i} + h_{v_i}$ (siehe Bild 7.13(c)).

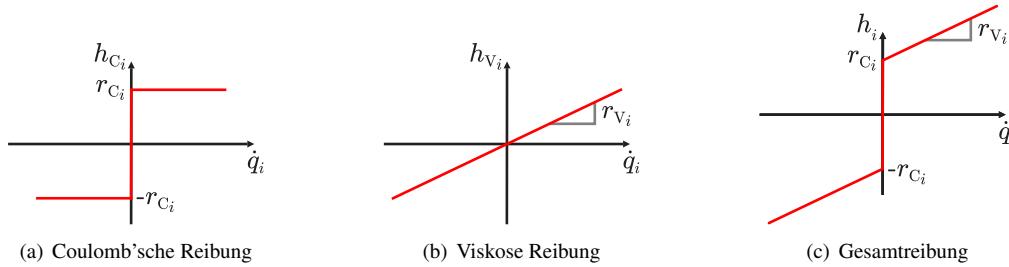


Bild 7.13: Reibung

Die Gesamtreibung resultiert aus einer linearen Superposition der viskosen Reibung sowie der Coulomb'schen Reibung.

Bei der Beschreibung der Reibeffekte eines Roboters mit n Freiheitsgraden (n generalisierte Koordinaten q_i) ergibt sich in vektorieller Schreibweise:

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \operatorname{diag}(\operatorname{sign}(\dot{\mathbf{q}})) \mathbf{r}_c + \operatorname{diag}(\dot{\mathbf{q}}) \mathbf{r}_v, \quad (7.75)$$

mit

$$\mathbf{r}_v = (r_{v_1}, r_{v_2}, \dots, r_{v_n})^T, \quad (7.76)$$

$$\mathbf{r}_c = (r_{c_1}, r_{c_2}, \dots, r_{c_n})^T. \quad (7.77)$$

Zu beachten ist, dass in diesem Fall der Vektor der Nichtlinearitäten lediglich von der Geschwindigkeit \dot{q}_i und nicht zusätzlich von der Position q_i abhängt.

Der Hauptanteil der Reibung tritt im Getriebe auf. Die Getriebeübersetzung hat demnach keinen Einfluss auf \mathbf{h} (vereinfachte Betrachtung). Für das Motormoment erhält man in diesem Fall:

$$\boldsymbol{\tau}_M = \operatorname{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) (\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})) + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}). \quad (7.78)$$

7.4.3 Motor- und Getriebeträgheiten

Zusätzlich zu den Robotersegmenten haben auch die Motoren der Antriebe von null verschiedene Massenträgheitsmomente (siehe Bild 7.14). Diese können zusätzlich bei der Beschreibung der Dynamik des Roboters berücksichtigt werden.

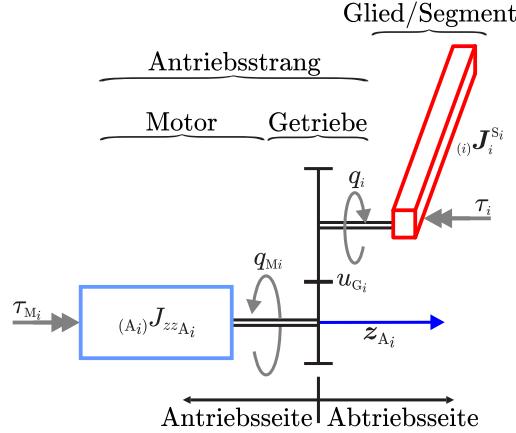


Bild 7.14: Antriebsstrang einer Roboterachse (Rotationsantrieb) mit Antriebsträgheit

Zur Berechnung des tatsächlich aufzubringenden Antriebsmoments τ_{M_i} muss zusätzlich die Trägheit des Antriebsstrangs $(A_i) J_{zz A_i}$ (Motor und Getriebe) um die Antriebsachse z_{A_i} berücksichtigt werden (hier zunächst ohne Reibanteil):

$$\tau_{M_i} = \frac{1}{u_{G_i}} \tau_i + (A_i) J_{zz A_i} \ddot{q}_{M_i}. \quad (7.79)$$

Mit $\ddot{q}_{M_i} = u_{G_i} \ddot{q}_i$ folgt daraus:

$$\tau_{M_i} = \frac{1}{u_{G_i}} \tau_i + u_{G_i} (A_i) J_{zz A_i} \ddot{q}_i. \quad (7.80)$$

In der Praxis werden beispielsweise Motoren mit Harmonic-Drive-Getrieben oder Zykloidgetriebe eingesetzt. Diese besitzen oft eine Übersetzung von $u_G \geq 50$.

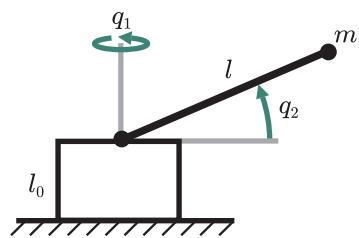
Das vollständige, dynamische Modell eines Roboters (bezogen auf das antriebsseitige Motormoment) hat demnach unter Berücksichtigung sämtlicher bisher erwähnter Faktoren (Getriebeübersetzung, Reibung, Antriebsstrangträgheit) folgende Form:

$$\begin{aligned} \boldsymbol{\tau}_M &= \text{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) (\boldsymbol{M}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})) + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ &\quad + \text{diag}(u_{G_1} (A_1) J_{zz A_1}, \dots, u_{G_n} (A_n) J_{zz A_n}) \ddot{\boldsymbol{q}}. \end{aligned} \quad (7.81)$$

Fragen zur Selbstkontrolle

1. Erläutern Sie die Newton-Euler Methode als klassischen Ansatz zur Bestimmung der Bewegungsgleichungen von Robotern und beschreiben Sie das allgemeine Vorgehen zur Aufstellung der dynamischen Gleichungen unter Verwendung dieser Methode!
2. Nennen Sie die Ihnen bekannten Reibeffekte und erläutern Sie deren Berücksichtigung in den Bewegungsgleichungen eines Roboters!

3. Bestimmen Sie die Lagrange'sche Funktion L sowie die Kraft τ_1 (erstes Gelenk) für den dargestellten reibungsfreien RR-Roboter (Endeffektormasse m , Gravitation g nach unten) als Funktion der Gelenkvariablen q_i und ihrer zeitlichen Ableitungen. Das Segment der Länge l ist als masselos anzunehmen. Sämtliche eingezeichneten Größen sind als bekannt vorauszusetzen!





8 Einzelachsregelung

In diesem Kapitel werden verschiedene Vorgehensweisen und Methoden zur Einzelachsregelung eines Roboters beschrieben. Zu Beginn erfolgt eine Wiederholung elementarer Begriffe (vgl. Abschnitt 8.1), allgemeiner Rückkopplungsstrukturen, sowie der als P-, PI- und PID-Regler bekannten, klassischen Rückführungen (Abschnitt 8.2).

Der Schwerpunkt des Kapitels ist die in Abschnitt 8.3 dargestellte Einzelachsregelung im Joint-space. Hierfür ist insbesondere das mechanische und elektrische Modell einer einzelnen Roboterachse von Interesse, welche die Grundlagen für die Kaskadenregelung eines Antriebsstrangs darstellen. Aufbauend auf der Übertragungsfunktion des geschlossenen Regelkreises werden in Abschnitt 8.4 dessen grundlegende Eigenschaften wie Stabilität und bleibende Regelabweichung analysiert. Gegenstand von Abschnitt 8.5 sind Verfahren zur Reglerauslegung, um ein gewünschtes (dynamisches und statisches) Verhalten der Einzelachsregelung zu erzeugen. Das aufgestellte Modell einer Einzelachse kann nicht nur zur Reglerauslegung sondern ebenfalls zur Vorsteuerung eingesetzt werden, um die Regelabweichung insbesondere bei hochdynamischen Bewegungen zu reduzieren (Abschnitt 8.6). Die Regelung einzelner Achsen im Taskspace ist in Abschnitt 8.7 dargestellt und schließt das Kapitel ab. Folgende Frage soll im weiteren Verlauf des Kapitels beantwortet werden: „Wie bringe ich meine Antriebe dazu, sich möglichst genau so zu bewegen, wie ich es vorschreibe?“

Als weiterführende Literatur wird empfohlen:

- L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Berlin, Deutschland: Springer, 2000, S. 199–213
- B. Heimann, W. Gerth und K. Popp. *Mechatronik: Komponenten – Methoden – Beispiele*. München, Deutschland: Carl Hanser Verlag, 2006, S. 230–293
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 299–310
- M. Horn und N. Douroumas. *Regelungstechnik*. München, Deutschland: Pearson Studium, 2004, S. 223–228
- Z. Vukic und O. Kuljaca. *Lectures on PID Controllers*. Techn. Ber. Zagreb, Kroatien: Faculty of Electrical Engineering and Computing, University of Zagreb, 2002, S. 29–43
- H.-D. Karl, M. Tinebor und J. Krietemeier. *EU Patentmeldung (EP0369119) – Verfahren und Schaltungsanordnung zur Erzeugung von digitalen Drehzahl- und Drehwinkelinformationen mittels eines Funktionsdrehmeters*. EP0369119. Europäische Patentanmeldung. 1990

8.1 Einleitung

Zum Einstieg in die Einzelachsregelung erfolgt die Abgrenzung der Begriffe Steuerung (Abschnitt 8.1.1) und Regelung (Abschnitt 8.1.2). Die Unterscheidung ist für die weiterführenden Kapitel dieser Vorlesung essenziell.

8.1.1 Robotersteuerung

Definition
Robotersteuerung

Damit der Endeffektor eines Roboters einer geforderten Sollbahn $\mathbf{x}_{E,s}(t)$, $\dot{\mathbf{x}}_{E,s}(t)$, und $\ddot{\mathbf{x}}_{E,s}(t)$ folgt, müssen die Aktoren der entsprechenden Gelenkwinkelbahn $\mathbf{q}_s(t)$, $\dot{\mathbf{q}}_s(t)$, $\ddot{\mathbf{q}}_s(t)$ folgen. Unter Verwendung der inversen Dynamik können die für die Sollbewegung notwendigen Antriebskräfte und/oder -momente $\tau_M(t)$ berechnet werden (vgl. Kapitel 7).

Der Zusammenhang ist in Bild 8.1 schematisch dargestellt.

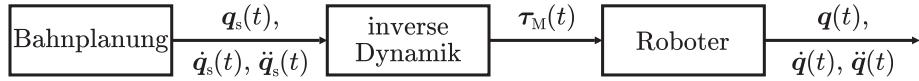


Bild 8.1: Robotersteuerung (schematische Darstellung)

Man spricht auch von einer sogenannten offenen Steuerkette bzw. einem offenen Regelkreis. Da in der Praxis jedoch kein exaktes dynamisches (und kinematisches) Robotermode vorliegt und externe Störgrößen auf den Roboter wirken (z. B. variable Lasten), ist dieser offene Kreis nicht exakt, es ergeben sich Abweichungen zwischen Soll- und Istverlauf:

$$\mathbf{q}_s(t) \neq \mathbf{q}(t), \quad \dot{\mathbf{q}}_s(t) \neq \dot{\mathbf{q}}(t), \quad \ddot{\mathbf{q}}_s(t) \neq \ddot{\mathbf{q}}(t), \quad (8.1)$$

und damit ($\mathbf{x}_E = f(\mathbf{q})$, ...):

$$\mathbf{x}_{E,s}(t) \neq \mathbf{x}_E(t), \quad \dot{\mathbf{x}}_{E,s}(t) \neq \dot{\mathbf{x}}_E(t), \quad \ddot{\mathbf{x}}_{E,s}(t) \neq \ddot{\mathbf{x}}_E(t). \quad (8.2)$$

Um Soll- und Istgrößen möglichst gut in Übereinstimmung zu bringen, ist die Berechnung der Abweichung (also des Fehlers) und dessen Rückführung erforderlich. Dies wird als Regelung bezeichnet (siehe Abschnitt 8.1.2).

8.1.2 Regelung

Die Norm DIN 19226 definiert den Begriff der Regelung wie folgt:

Definition
Regelung

Das Regeln, die Regelung, ist ein Vorgang, bei dem fortlaufend eine Größe, die Regelgröße, erfasst, mit einer anderen Größe, der Führungsgröße, verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird.

Vereinfacht auf die Regelung eines Roboters übertragen bedeutet dies, dass der aktuelle „Systemzustand“ $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, $\ddot{\mathbf{q}}(t)$ bzw. $\mathbf{x}_E(t)$, $\dot{\mathbf{x}}_E(t)$, $\ddot{\mathbf{x}}_E(t)$ erfasst und zurückgeführt wird. Durch Vergleichen mit der vorgegebenen Solltrajektorie (Führungsgröße) $\mathbf{q}_s(t)$, $\dot{\mathbf{q}}_s(t)$, $\ddot{\mathbf{q}}_s(t)$ bzw. $\mathbf{x}_{E,s}(t)$, $\dot{\mathbf{x}}_{E,s}(t)$ und $\ddot{\mathbf{x}}_{E,s}(t)$ können anschließend der aktuelle Fehler $e(t) = \mathbf{q}_s(t) - \mathbf{q}(t)$ bzw. $e(t) = \mathbf{x}_{E,s}(t) - \mathbf{x}_E(t)$ und die jeweiligen Ableitungen $\dot{e}(t)$, $\ddot{e}(t)$ sowie der daraus resultierende Steuervektor $\mathbf{y}(t)$ (die Antriebskräfte und/oder -momente) berechnet werden. Im Idealfall wird die Regeldifferenz $e(t)$ zu null geregelt.

Das Blockschaltbild des beschriebenen allgemeinen Regelkreises stellt sich wie in Bild 8.2 gezeigt dar. Die Zeitableitungen von $\mathbf{q}(t)$ bzw. $\mathbf{x}_E(t)$ werden mittels Verfahren zur numerischen Differentiation berechnet. Dabei ist zu beachten, dass sowohl aufgrund der begrenzten Sensorauflösung als auch aufgrund weiterer Effekte (z. B. thermische) der eigentlichen Messgröße ein Rauschanteil überlagert ist. Durch numerische Differentiation wird dieser im Allgemeinen in Bezug auf das eigentliche Signal noch verstärkt (das Signal-zu-Rausch-Verhältnis verschlechtert sich). Dies

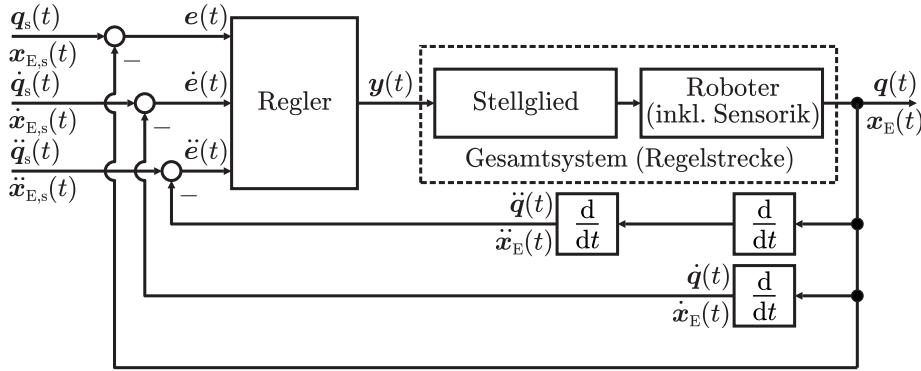


Bild 8.2: Allgemeine Struktur einer Roboterregelung

gilt insbesondere bei langsam Bewegungen und Beschleunigungen. Zur Unterdrückung der hochfrequenten Rauschanteile kann ein Tiefpassfilter eingesetzt werden. Diese Maßnahme führt jedoch zu einer Verzögerung des Signals im Zeitbereich (Phasenverschiebung im Frequenzbereich) und somit zu einer Verschlechterung der Performanz des geschlossenen Regelkreises.

Der Übersichtlichkeit halber beschränken sich die folgenden Darstellungen auf die Rückführung der Gelenkwinkel $q(t)$ bzw. der Lage $x_E(t)$.

8.2 Rückkopplungsstrukturen

Nachfolgend werden die Konzepte der dezentralen (Abschnitt 8.2.1) sowie der zentralen (Abschnitt 8.2.2) Regelung dargestellt und erläutert. Die klassischen Rückführungen sowie deren Kombination sind Gegenstand von Abschnitt 8.2.3. Eine kurze Wiederholung der Laplacetransformation in Abschnitt 8.2.4 schließt die Ausführungen.

8.2.1 Dezentrale Regelung

Bei der dezentralen Regelung ist für jede Achse des Roboters ein eigener Regler vorgesehen. Die nichtlinearen dynamischen Verkopplungen zwischen den einzelnen Achsen bleiben unberücksichtigt. Bei einem Roboter bzw. einem System mit n Paaren von Ein- und Ausgängen resultieren also insgesamt n Eingrößen-Regelkreise. Der schematische Aufbau einer dezentralen Regelung mit zwei Paaren von Ein- und Ausgängen $z_{s_i}(t)$ und $z_i(t)$ ist in Bild 8.3 als Beispiel dargestellt. Die diagonalen Pfeile zwischen Regelstrecke 1 und Regelstrecke 2 stellen die Verkopplung dieser beiden Teilsysteme dar. Die Regler 1 und Regler 2 berücksichtigen diese nicht. Auf dem Gebiet

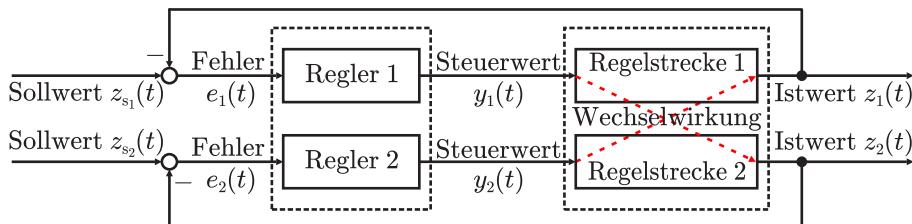


Bild 8.3: Schematischer Aufbau einer dezentralen Regelung mit zwei Paaren von Ein- und Ausgängen $z_{s_i}(t)$ und $z_i(t)$

der Industrierobotik findet dieses Verfahren überwiegend Anwendung.

8.2.2 Zentrale Regelung

Im Unterschied zu der dezentralen Regelung (Eingrößenregelung) werden die (nichtlinearen) Verkopplungen bei einer zentralen Regelung nicht vernachlässigt, sondern im Regler explizit berücksichtigt (vgl. Bild 8.4). Die Regelung wirkt diesen Verkopplungen aktiv entgegen. Der

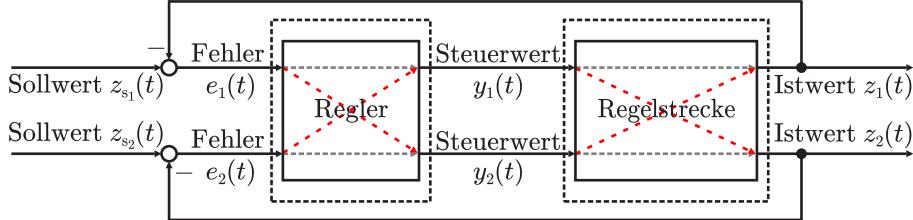


Bild 8.4: Schematischer Aufbau einer zentralen Regelung mit zwei Paaren von Ein- und Ausgängen $z_{s_i}(t)$ und $z_i(t)$ sowie unter Berücksichtigung der Verkopplungen von $y_1(t)$ auf $z_2(t)$ und $y_2(t)$ auf $z_1(t)$

Vorteil der zentralen Regelung gegenüber der dezentralen Regelung liegt vor allem in einer besseren Regelgüte, während als Nachteile der erhöhte Rechen- und Modellierungsaufwand zu nennen sind.

Sind sämtliche nichtlinearen Verkopplungen zwischen den einzelnen Regelstrecken ausreichend schwach, resultieren aus einer zentralen Regelung näherungsweise mehrere dezentrale Regelungen. In der Robotik haben sich bei den meisten Applikationen die vergleichsweise einfachen dezentralen Einzelachsregelungen als ausreichend herausgestellt. Erst bei hochdynamischen Roboterbewegungen sind die nichtlinearen dynamischen Verkopplungen zwischen den Achsen nicht mehr zu vernachlässigen und müssen explizit von der Regelung berücksichtigt werden.

Die folgenden Ausführungen beschränken sich vorerst auf die dezentrale Einzelachsregelung am Beispiel eines Antriebsstrangs im Jointspace (vgl. Bild 8.5 sowie Kapitel 7). Weiterführendes folgt in Kapitel 9 und Kapitel 10.

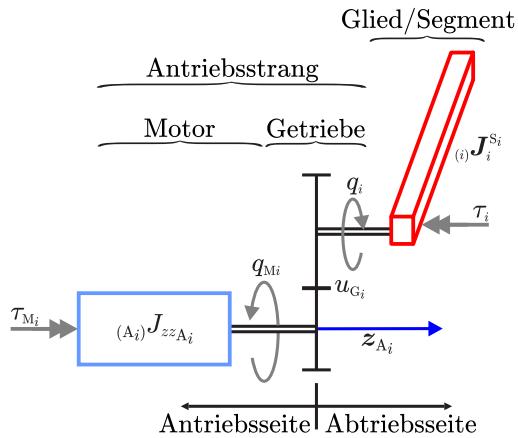


Bild 8.5: Modell einer Roboterachse

8.2.3 Klassische Rückführungen

Vor der Regelung der Einzelachsen des Roboters werden in diesem Abschnitt die bereits bekannten klassischen Rückführungen kurz wiederholt.

P-Glied

Das P-Glied besteht aus einem proportionalen Anteil mit einer Verstärkung K_P . Demnach ist die Stellgröße y (Ausgangssignal des Reglers) proportional zu der Regelabweichung e (Eingangssignal des Reglers). Das Regelgesetz lautet im Zeitbereich:

$$y(t) = K_P e(t). \quad (8.3)$$

Das zugehörige Blockschaltbild ist Bild 8.6 zu entnehmen.

I-Glied

Das I-Glied erzeugt durch zeitliche Integration der Regelabweichung e die Stellgröße y mit einer Gewichtung $\frac{1}{T_N}$, wobei T_N auch als Nachstellzeit bezeichnet wird:

$$y(t) = \frac{1}{T_N} \int_0^t e(t^*) dt^* + y_I(0), \quad (8.4)$$

siehe auch Bild 8.7 für die Darstellung im Zeitbereich. Der Anfangswert $y_I(0)$ berücksichtigt dabei die Historie des Integrators für $t < 0$. Für zu Beginn energiefreie Systeme ist $y_I(0) = 0$.

D-Glied

Das D-Glied ist ein Differenzierer. Er reagiert nicht auf die Regelabweichung e , sondern nur auf dessen zeitliche Änderung \dot{e} (Verwendung nur in Verbindung mit P- und/oder I-Regelung):

$$y(t) = T_V \dot{e}(t), \quad (8.5)$$

wobei T_V als Vorhaltezeit bezeichnet wird. In Bild 8.8 ist der D-Regler als Blockschaltbild im Zeitbereich dargestellt.

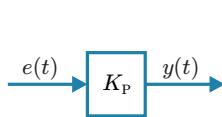


Bild 8.6: P-Glied

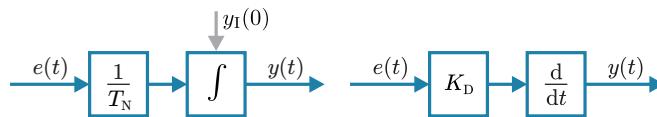


Bild 8.7: I-Glied

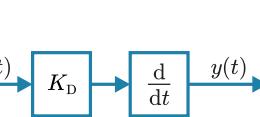
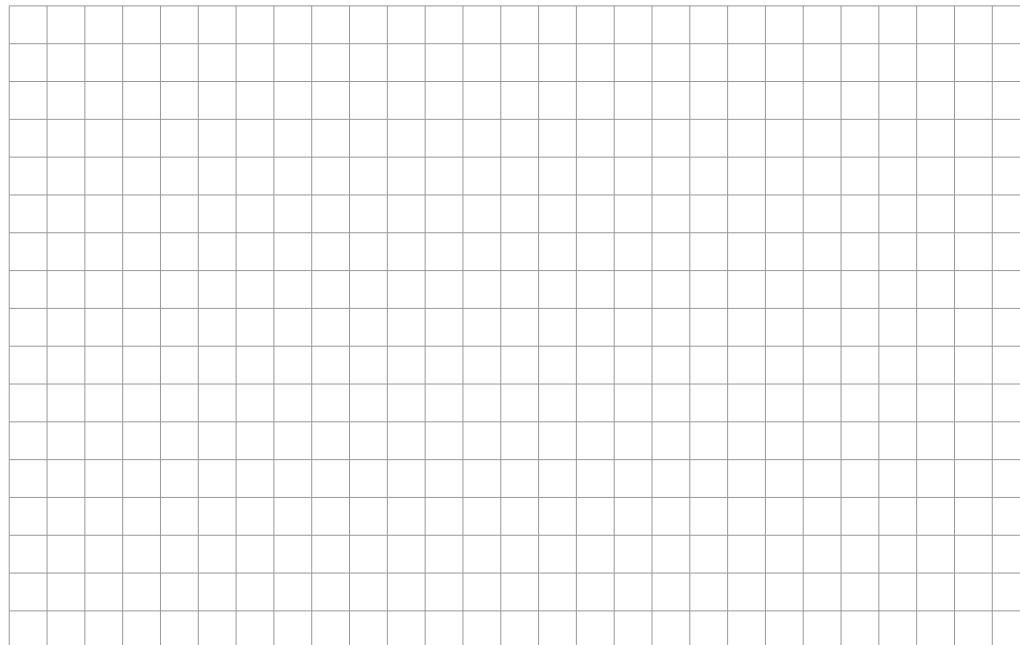


Bild 8.8: D-Glied





Kombination klassischer Rückführungen

Die klassischen Rückführungen können miteinander kombiniert werden, so dass beispielsweise die folgenden Glieder resultieren: PD-, PI- oder PID-Glieder. Bezüglich der Struktur handelt es sich dabei um Parallelschaltungen der elementaren Rückführungen. Beispielhaft sind der PD- und der PID-Regler Bild 8.9 zu entnehmen. In der Praxis werden häufig dezentralisierte Regler mit konstanten Regelparametern eingesetzt. Vorteil ist eine einfache Struktur, die wenig Rechenleistung erfordert.

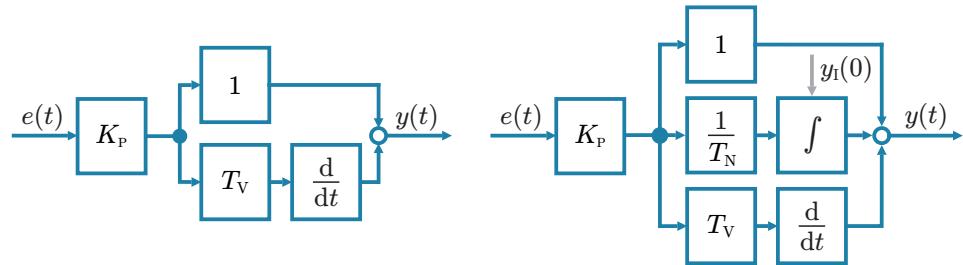


Bild 8.9: Struktur eines PD-Glieds (links) und eines PID-Glieds (rechts)

Die Gleichungen der in Abbildung 8.9 gezeigten Rückführungen lauten für ein

- PD-Glied:

$$y(t) = K_P (e(t) + T_V \dot{e}(t)) ,$$

- PID-Glied:

$$y(t) = K_P \left(e(t) + T_V \dot{e}(t) + \frac{1}{T_N} \int_0^t e(t^*) dt^* + \underbrace{y_0}_{y_I(0)} \right) .$$

Hinweis: Bei Verwendung der Einstellregeln (vgl. später) ist zu überprüfen, ob die vorhandene Reglerstruktur eine der hier dargestellten Strukturen entspricht.

Der Übersichtlichkeit halber werden im Folgenden die Regelgrößen wie folgt zusammengefasst:

$$K_I = K_P \frac{1}{T_N},$$

$$K_D = K_P T_V.$$

8.2.4 Frequenzbereich

Im Folgenden wird die Laplacetransformation¹ kurz wiederholt. Mittels Laplacetransformation wird ein System vom Zeit- in den Frequenzbereich überführt. Systeme, die im Zeitbereich durch lineare Differentialgleichungen mit konstanten Koeffizienten charakterisiert sind, können im Laplacebereich durch charakteristische algebraische Polynome beschrieben werden. Dies reduziert den rechnerischen Aufwand bei deren Analyse deutlich.

Das Verhalten bzw. die Eigenschaften, wie beispielsweise das Übertragungsverhalten und die Stabilität, physikalischer Systeme werden in der Regel im Frequenzbereich untersucht. Der Vorteil ist, dass gegenüber Untersuchungen im Zeitbereich eine explizite Lösung der System-Differentialgleichung nicht notwendig ist.

Anmerkung

Die Laplacetransformation (Zeitbereich $f(t) \circ \rightarrow$ Frequenzbereich $F(s)$) lautet:

$$F(s) = \mathcal{L}\{f(t)\}(s) = \int_0^{\infty} e^{-st} f(t) dt, \quad s \in \mathbb{C}. \quad (8.6)$$

Die Laplacevariable s ist Element der komplexen Zahlen \mathbb{C} .

Im Folgenden sei angenommen, dass alle Anfangswerte stets null sind (zu Beginn energiefreies System, d. h. $y_I(0) = 0$). Die Blockschatzbilder im Zeit- und Frequenzbereich eines PID-Reglers sind in Bild 8.10 gezeigt.

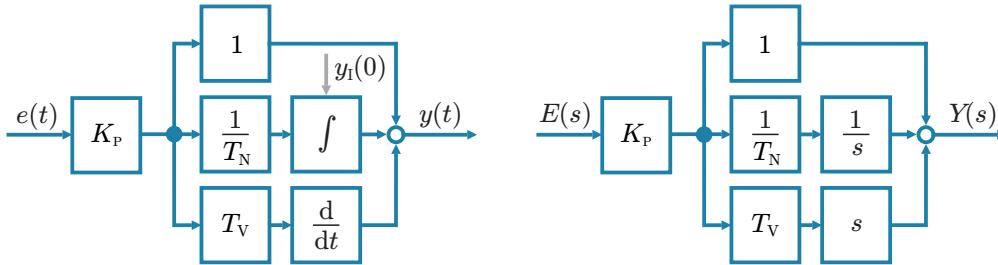


Bild 8.10: PID-Regler im Zeitbereich (links) und PID-Regler im Frequenzbereich (rechts)

Vereinfacht bedeutet dies, dass das Integral im Zeitbereich durch $1/s$ im Frequenzbereich und die Differentiation im Zeitbereich durch s im Frequenzbereich ausgedrückt wird.

Variables in Kleinbuchstaben stehen für die jeweiligen Größen im Zeitbereich, Variables in Großbuchstaben entsprechend im Frequenzbereich. Die in Abschnitt 8.2.3 eingeführten klassischen Rückführungen lauten im Frequenzbereich:

Anmerkung

- PD-Regler:

$$Y(s) = K_P (E(s) + T_V (s E(s) - e(0+))),$$

mit $e(0+) :$ Anfangswert (im Folgenden gleich null) ,

$$Y(s) = K_P (1 + s T_V) E(s),$$

- PID-Regler:

$$Y(s) = K_P \left(E(s) + s T_V E(s) + \frac{1}{T_N} \frac{1}{s} E(s) \right),$$

$$Y(s) = K_P \left(1 + s T_V + \frac{1}{s} \frac{1}{T_N} \right) E(s).$$

¹P.-S. Laplace, französischer Mathematiker und Astronom, 1749 - 1827

8.3 Einzelachsregelung im Jointspace

Nach der Erläuterung der grundlegenden Konzepte und Methoden, wird im folgenden Abschnitt auf die Einzelachsregelung eines Roboters im Jointspace eingegangen. Das vorliegende Unterkapitel gliedert sich in die Modellierung einer Roboterachse (Abschnitt 8.3.1), das vereinfachte Modell einer Roboterachse (Abschnitt 8.3.2) sowie die eingesetzten Achsregler (Abschnitt 8.3.3).

Prinzipieller Aufbau eines Roboters bzw. einer Roboterachse

Bevor detailliert auf die Einzelachsregelung im Jointspace eingegangen wird, soll zunächst der prinzipielle technische Aufbau einer Roboterachse vorgestellt werden. Diese besteht im Wesentlichen aus den in Bild 8.11 gezeigten (Antriebs-)Komponenten Motor mit Haltebremse,

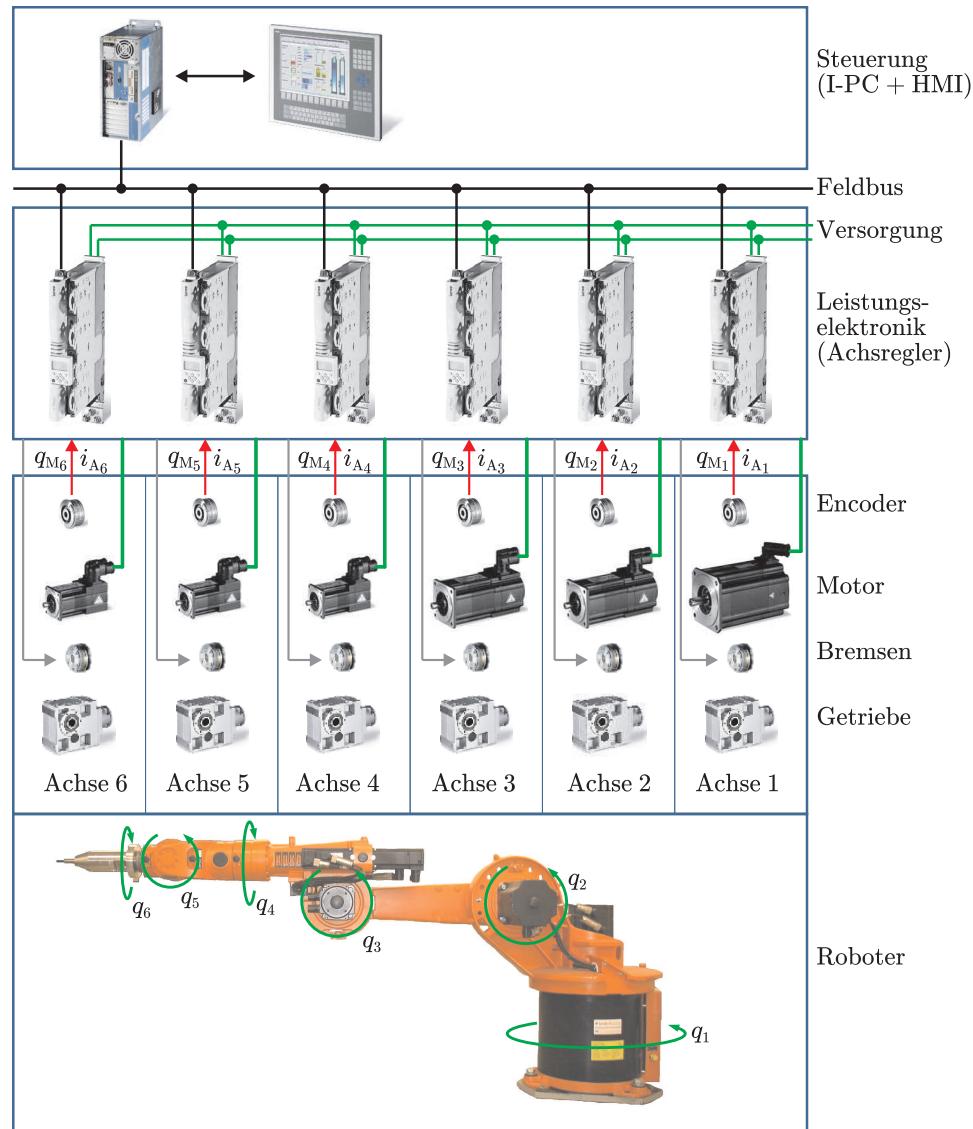


Bild 8.11: Komponenten eines Roboters (einer Roboterachse) am Beispiel eines Roboters mit sechs Freiheitsgraden, Quelle der Komponenten: Lenze GmbH

Getriebe und Winkelgeber (z. B. Encoder oder Resolver, vgl. Kapitel 11). Die Motoren werden von einer Leistungselektronik angesteuert und bewegen über ein Getriebe (Übersetzung u_{G_i}) das jeweilige Robotersegment. Auf der Leistungselektronik wird das Winkelgebersignal q_{M_i}

(Motorwinkel) ausgewertet und zur Einprägung eines drehmomentproportionalen Stroms i_{A_i} mittels Pulsweitenmodulation (PWM) eine Spannung an den Motorklemmen angelegt (vgl. Abschnitt 8.3.1). Die Gebersignale werden über einen Feldbus (z. B. EtherCAT) an den zentralen Prozessrechner (I-PC) gesendet, auf dem die Positions- und Drehzahlregelung des Roboters berechnet wird, deren Ausgangsgröße das Solldrehmoment τ_{M,s_i} bzw. der Sollstrom i_{A,s_i} ist. Über ein Bedienpanel bzw. Human-Machine-Interface (HMI) kann der Nutzer u. a. neue Sollbahnen oder Zielpunkte eingeben und die Parametrierung der Regelung vornehmen.

Blockschaltbild einer Einzelachsregelung im Jointspace

Die aktuellen (gemessenen) Motorwinkel $q_{M_i}(t)$ (vor Getriebe) können direkt mit den Sollwinkeln $q_{M,s_i}(t)$ verglichen und der Reglereingang $e_i(t)$ bestimmt werden. Der Regler berechnet anschließend in Abhängigkeit der Regelparameter einen entsprechenden Steuerwert $y_i(t)$ (häufig Soll-Motordrehmoment $\tau_{M,s_i}(t)$), der über die Motorkonstante k_M bzw. \hat{k}_M (vgl. später) in den korrespondierenden Steuerstrom $i_{A,s_i}(t)$ (Sollstrom des Ankers) umgerechnet werden kann. Der Zusammenhang ist Bild 8.12 zu entnehmen. Die Motorkonstante k_M wird als ein Proportionali-

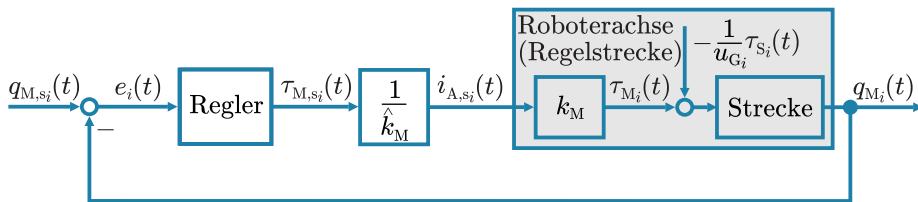


Bild 8.12: Blockschaltbild einer Einzelachsregelung im Jointspace

tätsfaktor angesehen, in den konstruktive Parameter einer elektrischen Maschine einfließen. Er setzt Motorstrom und Motormoment in Beziehung. Bei der Notation der Motorkonstante k_M ist zu beachten, dass es sich bei k_M um die tatsächliche Größe des physikalischen Systems handelt, während der Parameter mit Dach \hat{k}_M identifiziert oder aus Datenblättern übernommen wurde.

Die Berücksichtigung der Getriebetübersetzung ergibt:

$$\dot{q}_{M,s_i}(t) = u_{G_i} \dot{q}_{s_i}(t). \quad (8.7)$$

8.3.1 Modell einer Roboterachse

Im Folgenden wird das der Einzelachsregelung zugrunde liegende Modell einer von einem permanent erregten Synchronmotor angetriebenen Roboterachse hergeleitet. Es besteht aus drei Teilsystemen, die detailliert beschrieben werden:

- mechanisches Teilsystem,
- elektrisches Teilsystem,
- Stromregler (Bestandteil der Leistungselektronik).

Bild 8.13 zeigt die beiden miteinander verkoppelten Teilsysteme der Roboterachse (eine Erläuterung der einzelnen Größen erfolgt im Laufe der kommenden Abschnitte). Zur besseren Übersicht wird im Folgenden der Index i zur Kennzeichnung der Roboterachse i weggelassen. Zu beachten ist, dass es sich hierbei um eine vereinfachte Darstellung handelt, bei der nur der momentenbildende Strompfad betrachtet wird.

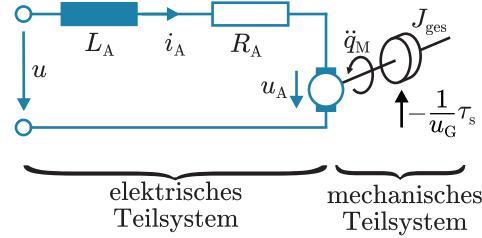


Bild 8.13: Vereinfachtes Modell einer Roboterachse

Mechanisches Teilsystem

Das mechanische Modell berücksichtigt die mechanischen Eigenschaften des Robotergliedes, wie beispielsweise Massenträgheit, Dämpfung und Getriebeübersetzung. Die Differentialgleichung zur Beschreibung lautet:

$$J_{\text{ges}} \ddot{q}_M + d_{\text{eff}} \dot{q}_M = \tau_M - \frac{1}{u_G} \tau_S, \quad \text{mit} \quad \tau_M = k_M i_A. \quad (8.8)$$

Dabei sind J_{ges} : auf Motorachse bezogenes Massenträgheitsmoment, \ddot{q}_M : Winkelbeschleunigung Motor, d_{eff} : Dämpfungskonstante, \dot{q}_M : Winkelgeschwindigkeit Motor, τ_M : Motormoment, k_M : Drehmomentkonstante, i_A : Ankerstrom, u_G : Getriebeübersetzung, τ_S : Störmoment.

Anmerkung

Im Vergleich zu der allgemeinen Form der dynamischen Gleichungen eines Roboters (siehe Gleichung (7.78)) werden bei der Einzelachs betrachtung sowie der anschließenden Regelung (vgl. Abschnitt 8.3.3) folgende Vereinfachungen angenommen und in dem Störmoment τ_S zusammengefasst:

- Vernachlässigung der nichtlinearen dynamischen Kopplungen (konfigurationsunabhängiges, konstantes Massenträgheitsmoment J_{ges}),
- Vernachlässigung der Gravitationseinflüsse sowie der Coriolis- und Zentrifugalterme,
- Zusammenfassung der Gesamtreibung in der Dämpfungskonstante d_{eff} bzw. Vernachlässigung der Coulomb'schen Reibung (wird in der Praxis (insbesondere bei langsam Bewegungen) häufig vorgesteuert).

Diese Vorgehensweise bzw. dieser Ansatz ist praktikabel, da der Großteil der (industriell) eingesetzten Roboter hochübersetzende Getriebe mit $u_G > 50$ verbaut. Die motorseitig wirkenden Störeinflüsse $\tau_{MS} = \frac{1}{u_G} \tau_S$ fallen dementsprechend wenig ins Gewicht.

Elektrisches Teilsystem

Stellt man die Maschengleichung für das elektrische Teilsystem (Ankerstromkreis) auf, ergibt sich folgende Differentialgleichung:

$$L_A \frac{di_A}{dt} + R_A i_A = u - u_A, \quad \text{mit} \quad u_A = k_A \dot{q}_M. \quad (8.9)$$

Dabei sind L_A : Ankerinduktivität, R_A : Ankerwiderstand, u : Eingangsspannung, u_A : induzierte Spannung im Ankerkreis (Gegenspannung), k_A : Spannungskonstante. Die beiden Teilsysteme sind über den Ankerstrom i_A sowie die Gegenspannung u_A (die sich aus der Winkelgeschwindigkeit des Roboters ergibt) miteinander gekoppelt.

Stromregler

Um das gewünschte Motordrehmoment $\tau_{M,s_i}(t)$ (Reglerausgang) zu erzeugen, muss über Variation der Eingangsspannung $u(t)$ (PWM-Leistungselektronik) der Ankerstrom $i_{A,s}(t)$ erzeugt werden. Die Größen Solldrehmoment und Sollankerstrom sind über die Drehmomentkonstante \hat{k}_M , die in der Regel aus dem Datenblatt des Motors entnommen werden kann, linear verkoppelt:

$$i_{A,s}(t) = \frac{1}{\hat{k}_M} \tau_{M,s}(t). \quad (8.10)$$

Zum Einstellen des gewünschten Ankerstroms $i_{A,s}(t)$ ist ein interner Stromregler erforderlich, der Teil der Leistungselektronik ist. Hierbei handelt es sich bei einem Servomotor in der Regel um einen PI-Regler, wie in Bild 8.14 dargestellt. Aus dem gewünschten Solldrehmoment $\tau_{M,s}$ wird der einzustellende Ankerstrom $i_{A,s}$ berechnet. Der PI-Regler (Parameter K_{I_P} und K_{I_I}) prägt diesen über die Spannung $u(t)$ in das elektrische Teilsystem des Motors (charakterisiert durch die Induktivität L_A und den Widerstand R_A) ein. Die Gegenspannung $u_A(t)$ bzw. $U_A(s)$ stellt eine Störgröße dar. Der tatsächlich fließende Ankerstrom $i_A(t)$ bzw. $I_A(s)$ erzeugt ein Drehmoment $\tau_M(t)$ bzw. $\mathcal{T}_M(s)$ (über die reale Motorkonstante k_M), das auf das mechanische System wirkt.

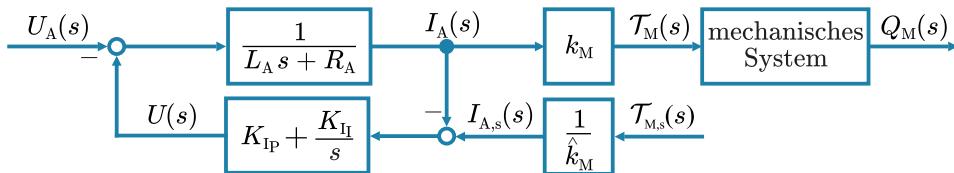


Bild 8.14: PI-Stromregelung eines permanent erregten Synchronmotors dargestellt im Frequenzbereich

Der interne Stromregler stellt den innersten Regelkreis einer Roboterregelung dar. Von seinem Verhalten hängt die Güte aller weiteren Regelungen ab, weshalb die Auslegung besonders sorgfältig erfolgen sollte. Das Blockschaltbild in Bild 8.15 zeigt das elektrische Teilsystem vereinfacht, wobei der elektrische Kreis durch ein PT_1 -Glied und der Stromregler durch ein PI-Glied ersetzt wurde.

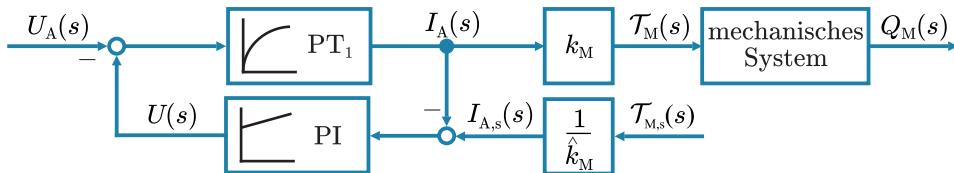


Bild 8.15: Blockschaltbild einer PI-Stromregelung (vereinfachte Darstellung)

Das elektrische Teilsystem besitzt zwei Zeitkonstanten, die Zeitkonstante des Stromreglers T_I sowie die Zeitkonstante der Motorspulen (elektrischer Kreis) T_S . Bei heutigen Antrieben sind typische Werte für diese Zeitkonstanten:

$$T_I = \frac{K_{I_P}}{K_{I_I}} \approx 1 \dots 5 \text{ ms}, \quad T_S = \frac{L_A}{R_A} \approx 1 \dots 5 \text{ ms}. \quad (8.11)$$

Durch die PI-Regelung des Ankerstroms $i_A(t)$ kann bei geschickter Wahl der Regelparameter (Pol-/Nullstellenkompensation) das elektrische Teilsystem insgesamt durch ein Verzögerungsglied erster Ordnung (PT_1 -Verhalten) mit der neuen Zeitkonstante T_e dargestellt werden. Dabei gilt: $T_e \approx 0,25 \text{ ms} \dots 0,5 \text{ ms} < T_I, T_S$. Das bedeutet, dass das geregelte elektrische System schneller ist als die jeweiligen Zeitkonstanten T_I des Stromreglers und T_S der Motorspulen vermuten

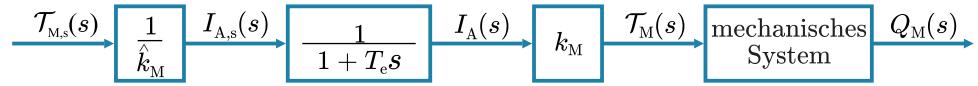
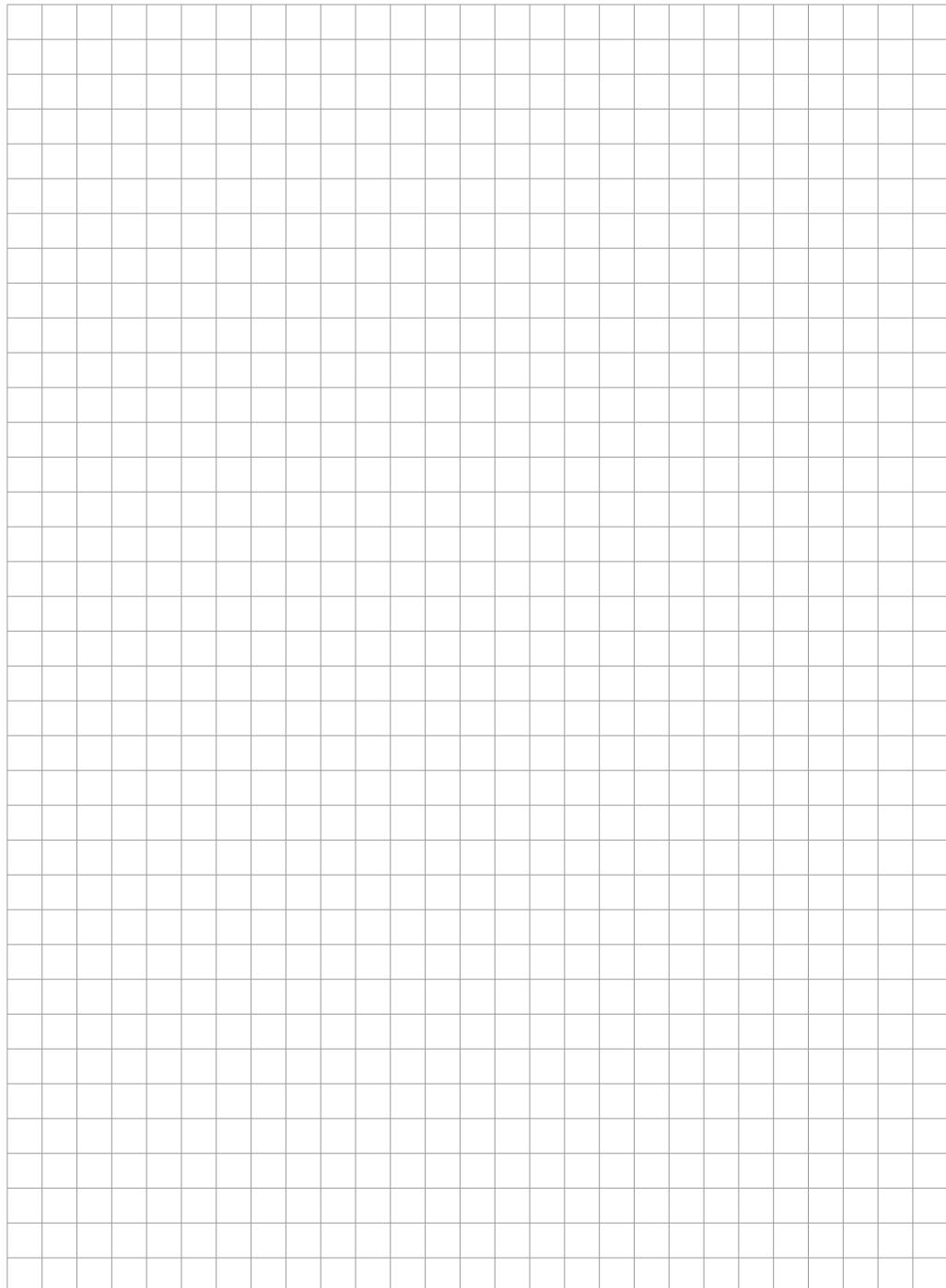


Bild 8.16: Vereinfachung des elektrischen Teilsystems zu einem PT₁-Glied

lassen. Das so vereinfachte Modell ist in Bild 8.16 dargestellt. Um dies zu erreichen, wird der Stromregler so ausgelegt, dass seine Nullstelle (Zähler der Übertragungsfunktion) mit der Polstelle des elektrischen Kreises (Nenner der Übertragungsfunktion) zusammenfällt. Diese sogenannte Kompensation hat zur Folge, dass das resultierende Gesamtsystem PT₁-Verhalten zeigt. Das System ist dynamischer als die Einzelkomponenten.





8.3.2 Vereinfachtes Modell einer Roboterachse

Betrachtet man nun die Zeitkonstante T_e des geregelten elektrischen Kreises und die Zeitkonstante T_m des gekoppelten mechanischen Kreises, fällt auf, dass die Zeitkonstante des elektrischen Teils sehr viel kleiner ist als die des mechanischen Teils:

$$T_e \ll T_m = \frac{J_{\text{ges}}}{d_{\text{eff}}}. \quad (8.12)$$

Das bedeutet, dass für das Verhalten der Roboterachse der mechanische Teil dominiert. Vereinfacht kann demnach angenommen werden, dass der Ankerstrom $i_A(t)$ dem Sollstrom des Ankers $i_{A,s}(t)$ entspricht. Somit ergibt sich das in Bild 8.17 gezeigte Modell der Roboterachse.

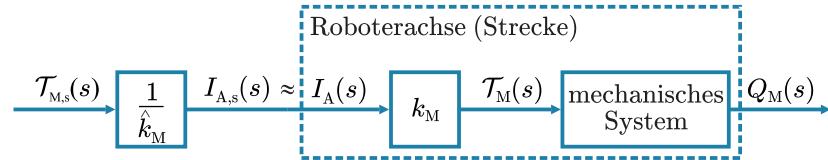


Bild 8.17: Vereinfachtes Modell (Vernachlässigung der Dynamik des elektrischen Teilsystems)

Aus vorangegangenen Überlegungen resultiert das Modell einer Roboterachse somit zu:

$$J_{\text{ges}} \ddot{q}_M(t) + d_{\text{eff}} \dot{q}_M(t) = k_M i_A(t) - \frac{1}{u_G} \tau_S(t). \quad (8.13)$$

Diesem liegt die Annahme zu Grunde, dass das Zeitverhalten des geregelten elektrischen Teilsystems deutlich schneller ist als das des mechanischen Teilsystems. Durch Integration von Gleichung (8.13) erhält man:

$$\begin{aligned} q_M(t) &= \iint \ddot{q}_M \, d^2 t + \int \dot{q}_M(0) \, dt + q_M(0) \\ &= \iint \frac{1}{J_{\text{ges}}} \left(k_M i_A(t) - \frac{1}{u_G} \tau_S(t) - d_{\text{eff}} \dot{q}_M(t) \right) \, d^2 t + \int \dot{q}_M(0) \, dt + q_M(0). \end{aligned} \quad (8.14)$$

Stellt man Gleichung (8.14) als Blockschaltbild dar, resultiert Bild 8.18.

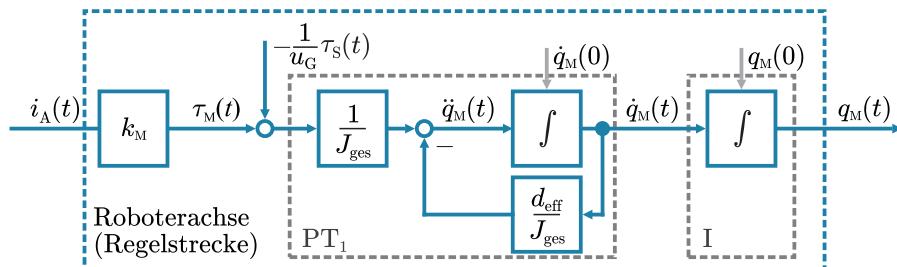


Bild 8.18: Blockschaltbild einer Roboterachse

Die Roboterachse entspricht demnach einem PT_1 -Glied mit nachgestelltem I-Glied. Unter Anwendung der Laplacetransformation lässt sich das Modell nach Gleichung (8.13) wie folgt in den Frequenzbereich umrechnen:

$$\begin{aligned} J_{\text{ges}} \ddot{q}_{\text{M}}(t) + d_{\text{eff}} \dot{q}_{\text{M}}(t) &= k_{\text{M}} i_{\text{A}}(t) - \frac{1}{u_{\text{G}}} \tau_{\text{S}}(t) \\ &\quad \circ \rightarrow \bullet \quad (8.15) \\ J_{\text{ges}} Q_{\text{M}}(s) s^2 + d_{\text{eff}} Q_{\text{M}}(s) s &= k_{\text{M}} I_{\text{A}}(s) - \frac{1}{u_{\text{G}}} \tau_{\text{S}}(s). \end{aligned}$$

Zu beachten ist, dass die Anfangswerte zu null gesetzt werden (vgl. auch nachstehende Regel für die Transformation der Ableitung im Zeitbereich in den Frequenzbereich):

$$\begin{aligned} f^{(n)}(t) &= \frac{d^n f(t)}{dt^n} \\ &\quad \circ \rightarrow \bullet \quad (8.16) \\ s^n F(s) - \left(s^{n-1} f^{(0)}(0+) - s^{n-2} f^{(1)}(0+) - \dots - s^0 f^{(n-1)}(0+) \right) . \end{aligned}$$

Somit ergibt sich das in Bild 8.19 gezeigte Blockschaltbild der Roboterachse im Frequenzbereich.

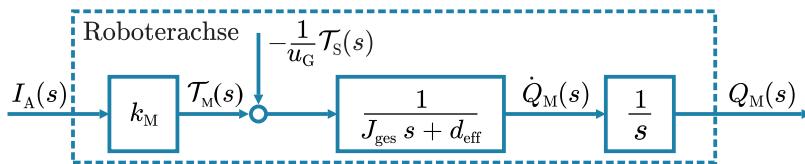


Bild 8.19: Blockschaltbild einer Roboterachse im Frequenzbereich

Nachdem das vollständige Modell einer Roboterachse hergeleitet wurde, wird im Folgenden auf die Einzelachsregelung eingegangen. Die Regelstrecke entspricht bei der Betrachtung einer einzelnen Achse dem bereits bekannten Modell.

8.3.3 Einzelachsregelung

Zur Regelung einer Roboterachse können verschiedene Regler eingesetzt werden. Der allgemeine Fall ist in Bild 8.20 gezeigt.

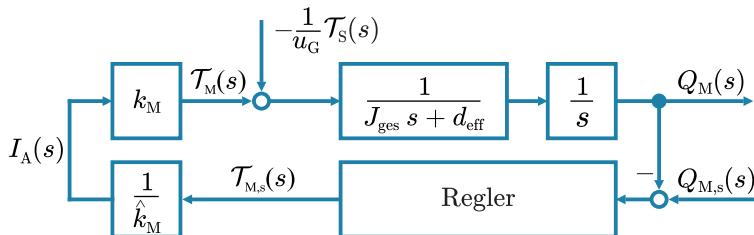
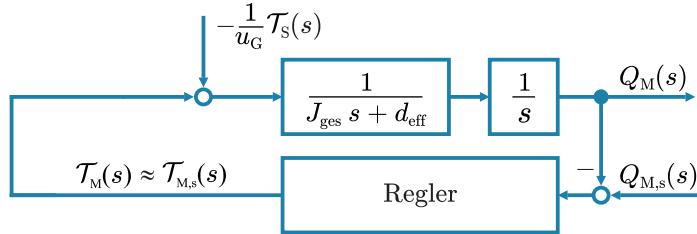


Bild 8.20: Blockschaltbild einer Einzelachsregelung im Frequenzbereich

Bevor jedoch konkrete Reglerstrukturen vorgestellt werden, wird noch folgende Annahme bezüglich der Struktur des Regelkreises getroffen: bei hinreichend guter Näherung der Drehmomentkonstanten gilt: $k_{\text{M}} \approx \hat{k}_{\text{M}}$. Somit ergibt sich das in Bild 8.21 dargestellte Blockschaltbild.

Bild 8.21: Blockschaltbild einer Einzelachsregelung mit $k_M \approx \hat{k}_M$

Ziel der Einzelachsregelung ist das Schließen des Regelkreises unter Verwendung eines geeigneten Reglers, so dass eine möglichst hohe Regelgüte (kleiner Regelfehler, schnelles Einschwingen, etc.) erreicht wird. Grundsätzlich sind dazu die gängigen Kombinationen von klassischen Rückkopplungsstrukturen geeignet (vgl. Abschnitt 8.2.3). Im Folgenden werden einige mögliche Realisierungen detailliert analysiert.

PID-Einzelachsregelung

Zur Regelung einer Roboterachse kann beispielsweise ein dezentraler PID-Regler mit konstanten Regelparametern K_P , K_I , K_D eingesetzt werden, siehe Bild 8.22. Vorteil ist seine einfache Struktur, die wenig Rechenleistung erfordert.

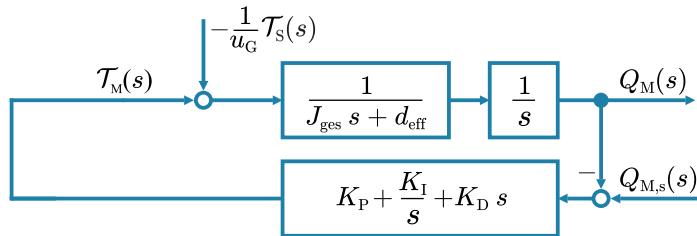


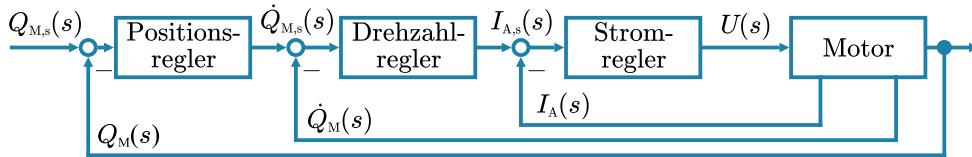
Bild 8.22: Blockschaltbild einer PID-Einzelachsregelung im Frequenzbereich

Nachteilig ist allerdings, dass die Regelparameter nicht unabhängig voneinander einstellbar sind. Des Weiteren ist eine Differenziation der verrauschten Positionsweite im Regler notwendig. Dies hat zur Folge, dass der Rauschanteil im Geschwindigkeitssignal verstärkt auftritt (vgl. auch Abschnitt 8.1.2). Alternativ zur klassischen PID-Regelung kann auch ein Kaskadenregler eingesetzt werden.

Kaskadenregelung

Bei der Kaskadenregelung erfolgt eine Schachtelung (Kaskadierung) einzelner Regelkreise. Es werden zunächst die Regelgrößen der inneren, schnellen Regelkreise geregelt, deren Sollwerte aus den Steuerwerten der äußeren, langsamten Regelkreise resultieren. Die Regelung einer Synchrongmaschine (einer Roboterachse) ist typischerweise als Kaskade aufgebaut. Der innerste und schnellste Regelkreis der in Bild 8.23 gezeigten Kaskadenregelung ist der Stromregler. Danach folgt der Drehzahlregler und, als langsamster sowie äußerster Regelkreis, der Positionsregler. Die Vorteile (gegenüber den zuvor vorgestellten Ansätzen) sind beispielsweise:

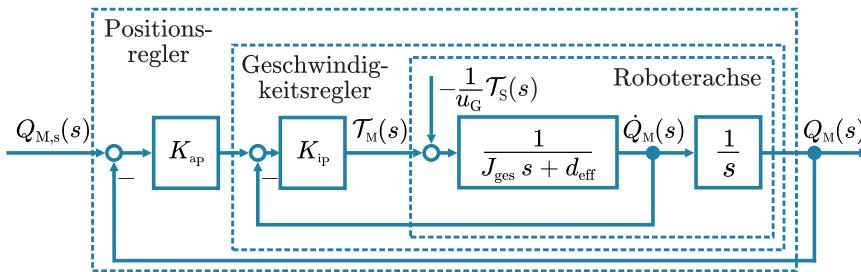
- komfortable und unabhängige Einstellung der Regelparameter von innen nach außen,
- schnellere Reaktion auf Regelfehler → höhere Regelgenauigkeit,

**Bild 8.23:** Typische Kaskadenregelung eines Synchronmotors

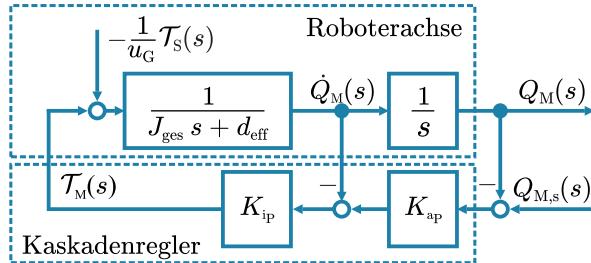
- Geschwindigkeitssignal nicht durch Differentiation im Regler erzeugt sondern intern vom Motor „bereitgestellt“ (siehe hierzu [KTK90]).

In der Praxis wird zunächst der innerste Regelkreis optimal ausgelegt. Für die Auslegung des folgenden, äußeren Reglers kann dieser dann als ideal angenommen werden.

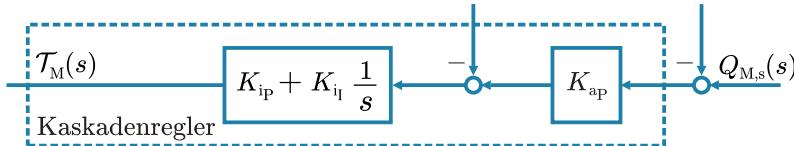
In Bild 8.24 ist eine P/P-Kaskadenregelung als Pendant zur PD-Regelung dargestellt. Sowohl der Positions- als auch der Geschwindigkeitsregler ist ein einfacher P-Regler (der Stromregler wird nach den in Abschnitt 8.3.2 vorgestellten Vereinfachungen nicht mehr explizit berücksichtigt).

**Bild 8.24:** P/P-Kaskadenregelung einer Roboterachse

Nach „Umzeichnen“ kann der Regelkreis in folgender kompakter Form dargestellt werden (siehe Bild 8.25). Diese Darstellungsform ist die Grundlage für die weiteren Ausführungen. Zu beachten ist, dass es sich lediglich um eine andere Darstellung handelt, die Struktur wurde nicht verändert.

**Bild 8.25:** P/P-Kaskadenregelung einer Roboterachse

Alternativ kann auch die sogenannte P/PI-Kaskadenregelung eingesetzt werden (vgl. Bild 8.26).

**Bild 8.26:** Alternativ P/PI-Kaskadenregelung einer Roboterachse

Im Folgenden werden die Übertragungsfunktionen eines Regelkreises diskutiert. Diese lassen bei geringem Aufwand zahlreiche Rückschlüsse auf die Eigenschaften und das Verhalten des Regelkreises zu.

8.4 Übertragungsfunktionen und deren Eigenschaften

Übertragungsfunktionen beschreiben in kompakter Form die Systemeigenschaften. In diesem Abschnitt werden die beiden wichtigsten, die Führungs- und Störgrößenübertragungsfunktion eingeführt. Darauf aufbauend erfolgen in Abschnitt 8.4.1 eine knappe Wiederholung wesentlicher Stabilitätseigenschaften durch Polbetrachtungen und in Abschnitt 8.4.2 eine Analyse der ggf. vorhandenen bleibenden Regelabweichung.

Führungs- und Störgrößenübertragungsfunktion

Die Übertragungsfunktion beschreibt das Systemverhalten im Frequenzbereich. In der Regelungstechnik ist sie die häufigste Darstellungsform für die Analyse des Eingangs- und Ausgangsverhaltens und dient beispielsweise zur Bestimmung der bleibenden Regelabweichung und für Stabilitätsuntersuchungen. Man unterscheidet zwei wichtige spezielle Übertragungsfunktionen:

- Führungsgrößenübertragungsfunktion (Führungsverhalten):

Die Führungsgrößenübertragungsfunktion gibt an, wie das System (Roboterachse) auf eine Änderung der Führungsgröße $Q_{M,s}$ (der Sollwerte) reagiert. Dabei wird das Störmoment \mathcal{T}_S zu null gesetzt:

$$G_F(s) = \frac{Q_M(s)}{Q_{M,s}(s)}, \quad \text{mit } \mathcal{T}_S(s) = 0. \quad (8.17)$$

- Störgrößenübertragungsfunktion (Störverhalten):

Die Störgrößenübertragungsfunktion gibt an, wie der Roboter auf eine Änderung der Störgröße \mathcal{T}_S reagiert. Hier wird der Systemeingang $Q_{M,s}$ zu null gesetzt:

$$G_S(s) = \frac{Q_M(s)}{\mathcal{T}_S(s)}, \quad \text{mit } Q_{M,s}(s) = 0. \quad (8.18)$$

Diese Übertragungsfunktionen werden am Beispiel der P/P-Kaskadenregelung hergeleitet. Die Grundlage für das Erstellen der **Führungsgrößenübertragungsfunktion** $G_F(s)$ ist das in dem Bild 8.27 gezeigte Blockschaltbild. Zu beachten ist, dass das Störmoment hierfür zu null gesetzt

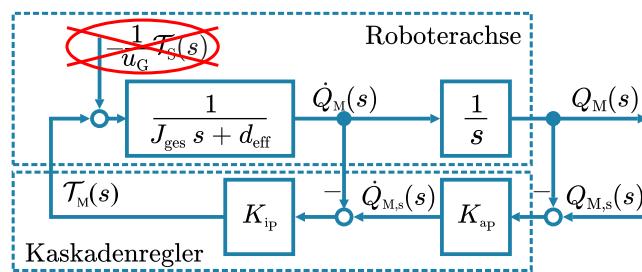
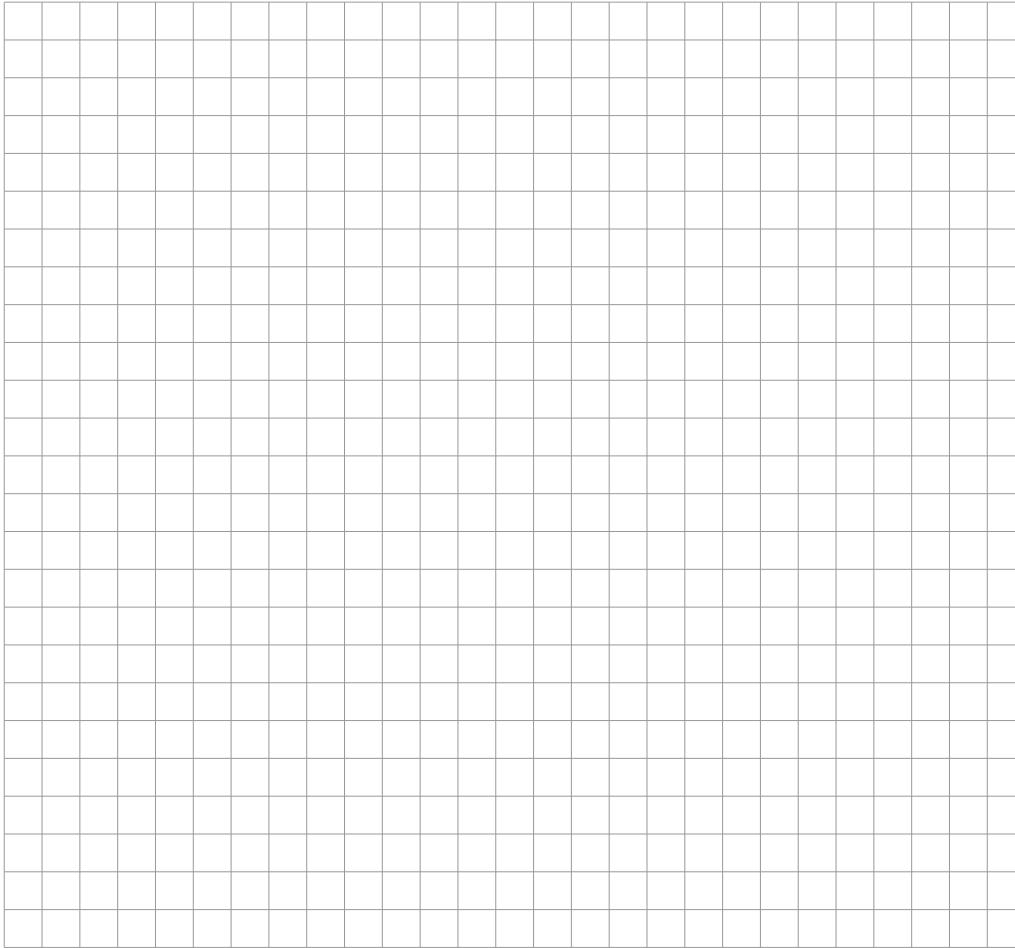


Bild 8.27: Führungsgrößenübertragungsfunktion einer Einzelachsregelung im Jointspace unter Verwendung einer P/P-Kaskadenregelung

ist.

Nach Aufstellen der Gleichungen und einigen Umformungen erhält man für die gesuchte Führungsgrößenübertragungsfunktion:

$$G_F(s) = \frac{K_{ip} K_{ap}}{J_{ges} s^2 + (d_{eff} + K_{ip}) s + K_{ip} K_{ap}}. \quad (8.19)$$



Das Vorgehen zum Aufstellen der **Störgrößenübertragungsfunktion** $G_S(s)$ ist analog. Wie in Bild 8.28 gezeigt, wird die Führungsgröße $Q_{M,s}$ zu null gesetzt. Als eigentliche Eingangsgröße dient das Störmoment \mathcal{T}_S .

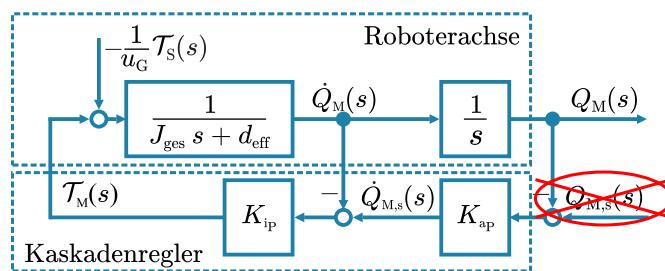


Bild 8.28: Störgrößenübertragungsfunktion einer Einzelachsregelung im Jointspace unter Verwendung einer P/P-Kaskadenregelung

Die Störgrößenübertragungsfunktion $G_S(s)$ für die P/P-Kaskadenregelung einer Roboterachse lautet:

$$G_S(s) = \frac{-1/u_G}{J_{\text{ges}} s^2 + (d_{\text{eff}} + K_{\text{ip}}) s + K_{\text{ip}} K_{\text{ap}}}. \quad (8.20)$$

8.4.1 Stabilitätsuntersuchung (Wiederholung)

Die Stabilität eines geschlossenen Regelkreises ist die Grundvoraussetzung für die Funktionalität eines Systems. Instabilitäten führen zu ungewolltem Aufschwingen, das zur Beschädigung des Mechanismus führen kann.

Anhand der Lage der Polstellen (Nullstellen des charakteristischen Polynoms (Nenner der Übertragungsfunktion(en)) in der komplexen Zahlebene (vgl. Bild 8.29) kann ermittelt werden, ob der geschlossenen Regelkreis stabil ist. Ein System ist:

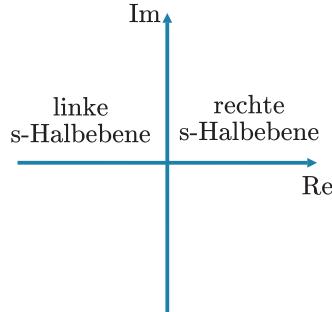


Bild 8.29: Komplexe Ebene nach Euler

- **instabil**, wenn mindestens einer der Pole in der offenen rechten s-Halbebene liegt oder mindestens ein (echter) mehrfacher Pol auf der Imaginärachse (Im) vorhanden ist,
- **grenzstabil**, wenn keiner der Pole in der offenen rechten s-Halbebene liegt und auf der Imaginärachse nur einfache Pole vorhanden sind,
- **stabil**, wenn alle Pole in der offenen linken s-Halbebene liegen,
- **schwingungsfähig (abklingend)**, wenn mindestens ein konjugiert komplexes Polpaar in der linken s-Halbebene existiert.

Für die Herleitung dieser Aussagen sei auf die Vorlesung Regelungstechnik verwiesen.

Beispiel

Einzelachsregelung im Jointspace unter Verwendung einer P/P-Kaskadenregelung

Anhand des Beispiels soll gezeigt werden, wie die Reglerparameter das Einschwingverhalten des geschlossenen Regelkreises beeinflussen. Dazu wird das charakteristische Polynom (Nenner) der Führungsgrößenübertragungsfunktion herangezogen². Zunächst sind dessen Nullstellen zu bestimmen. Ein Vergleich mit der Normalform eines PT₂-Gliedes (D : Dämpfung, ω_0 : Frequenz der ungedämpften Schwingung)

$$s^2 + 2D\omega_0 s + \omega_0^2 = 0 \quad (8.21)$$

erlaubt eine direkte Zuordnung der Regler- und Streckenparameter zu den Streckeneigenschaften Dämpfung D und Frequenz der ungedämpften Schwingung ω_0 :

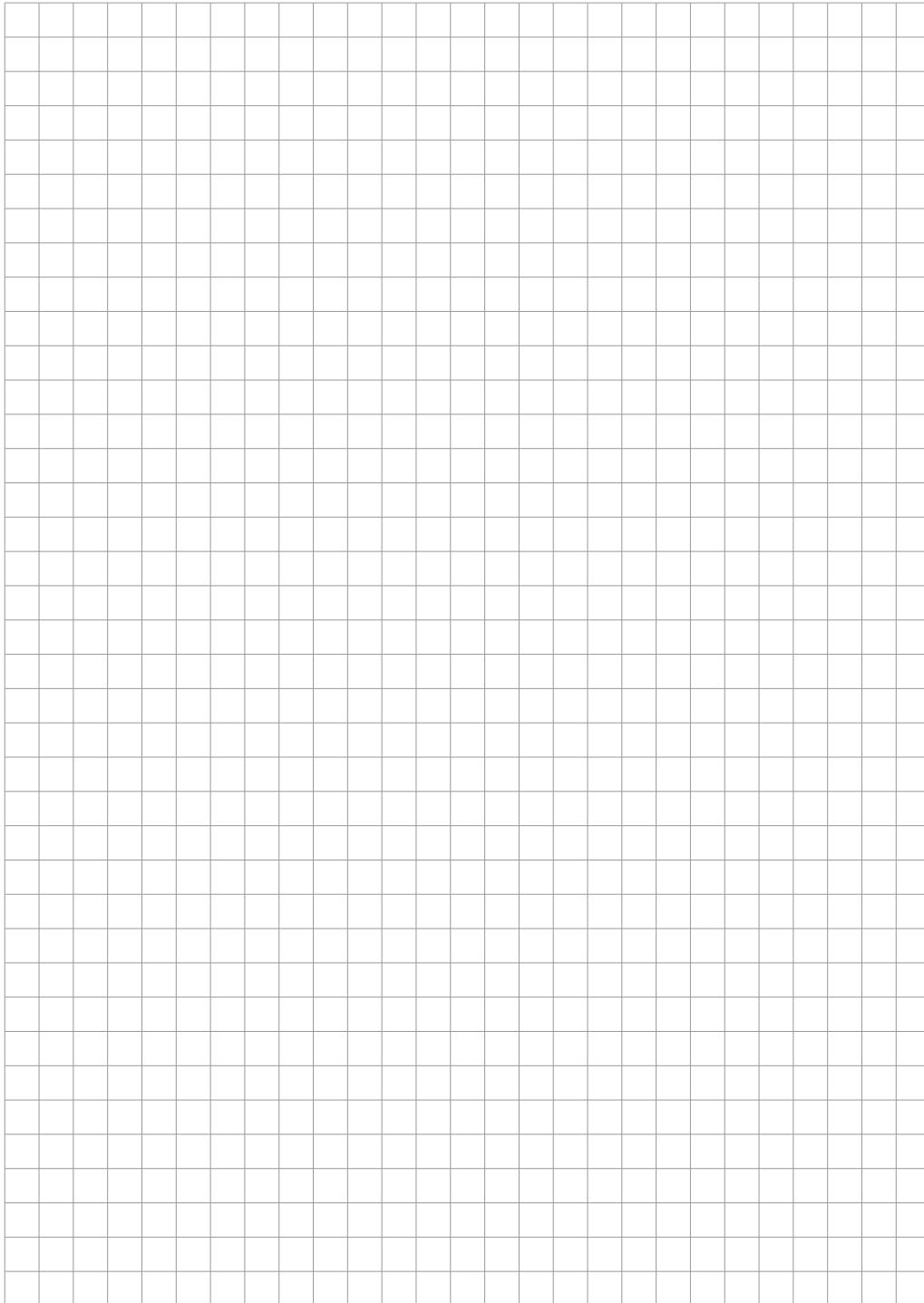
$$\underbrace{s^2 + \frac{d_{\text{eff}} + K_{\text{IP}}}{J_{\text{ges}}} s + \frac{K_{\text{IP}} K_{\text{AP}}}{J_{\text{ges}}}}_{2D\omega_0} \stackrel{!}{=} 0. \quad (8.22)$$

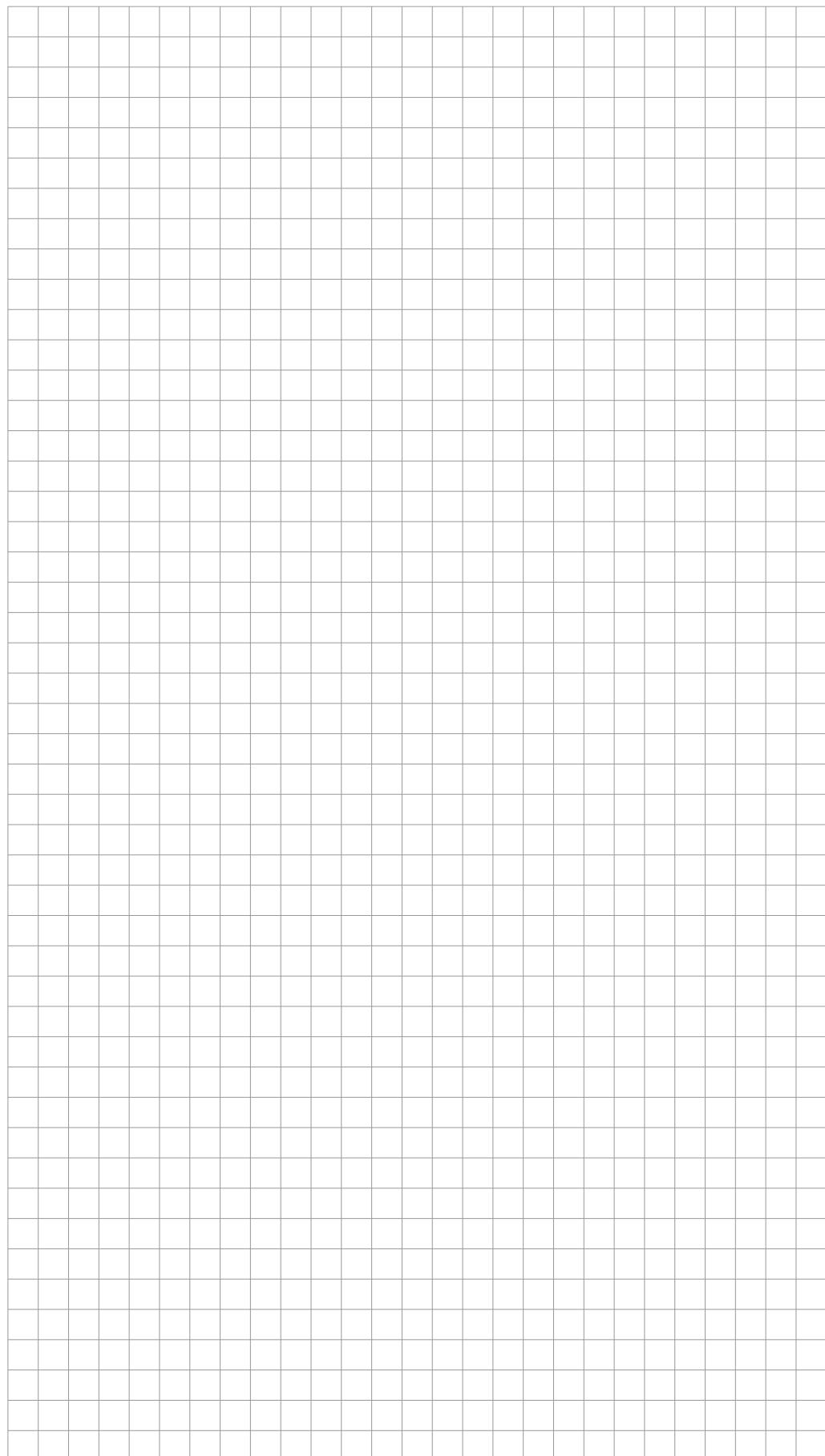
²Anmerkung: Die beiden Übertragungsfunktionen nach Gleichung (8.19) und Gleichung (8.20) weisen denselben Nenner auf und verfügen somit über dieselben Stabilitäts-eigenschaften.

Als Nullstellen ($\hat{=}$ Polstellen) erhält man:

$$\begin{aligned} s_{1,2} &= -\frac{d_{\text{eff}} + K_{\text{ip}}}{2 J_{\text{ges}}} \pm \sqrt{\left(\frac{d_{\text{eff}} + K_{\text{ip}}}{2 J_{\text{ges}}}\right)^2 - \frac{K_{\text{ip}} K_{\text{ap}}}{J_{\text{ges}}}} \\ &= -D \omega_0 \pm \sqrt{(D \omega_0)^2 - \omega_0^2} = -D \omega_0 \pm \omega_0 \sqrt{D^2 - 1}. \end{aligned} \quad (8.23)$$

Die Lage der Polstellen und somit das Einschwingverhalten hängen von den eingestellten Reglerparametern sowie von den Systemparametern ab. Ebenso können Dämpfung und Frequenz des geschlossenen Regelkreises abgelesen werden.





Zusammenfassung

Im Folgenden werden die Eigenschaften der hier behandelten Übertragungsfunktion in Stichpunkten zusammengefasst. Die Nullstellen des Nennerpolynoms

$$F(s) = s^2 + 2 D \omega_0 s + \omega_0^2 \quad (8.24)$$

lauten allgemein:

$$s_{1,2} = -D \omega_0 \pm \omega_0 \sqrt{D^2 - 1} = -\sigma_e \pm j \omega_e \quad (8.25)$$

und sind in Bild 8.30 dargestellt.

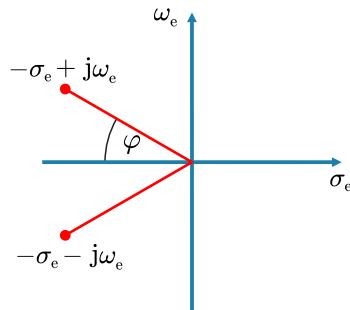


Bild 8.30: Lage der Pole in komplexer Zahlenebene

Die Eigenschaften eines Systems sind direkt aus den Polstellen „ablesbar“, wenn sie gedanklich in den Zeitbereich übertragen werden. Die Transformation der Sprungantwort eines PT_2 -Gliedes mit dem Nennerpolynom nach Gleichung (8.25) in den Zeitbereich liefert für ein schwach gedämpftes System ($D < 1$):

$$f(t) = \frac{1}{\omega_0^2} \left(1 - \frac{1}{\sqrt{1-D^2}} e^{-\omega_0 D t} \sin(\omega_0 \sqrt{1-D^2} t + \arccos(D)) \right), \quad (8.26)$$

mit der Amplitude $1/\omega_0^2$.

Die einzelnen Terme von Gleichung (8.26) können wie folgt interpretiert und mit der Lage der Polstellen in Zusammenhang gebracht werden:

- Je weiter „links“ sich die Pole befinden, desto kürzer ist die Einschwingzeit T :

$$T = \left| \frac{1}{\sigma_e} \right| = \left| \frac{1}{D \omega_0} \right| \quad (\text{Bedeutung im Zeitbereich: } e^{-\frac{t}{T}}). \quad (8.27)$$

- Je weiter „oben“ bzw. „unten“ sich die Pole befinden (betragsmäßig großes ω_e), desto höher ist die Frequenz beim Einschwingen (Bedeutung im Zeitbereich: $\sin(\omega_e t + \varphi)$).
- Je größer der Winkel φ , desto geringer ist die Schwingung gedämpft (Bedeutung im Zeitbereich: $D = \cos(\varphi)$).

8.4.2 Bleibende Regelabweichung

Die bleibende Regelabweichung e_∞ charakterisiert das Verhalten eines stabilen Regelkreises für $t \rightarrow \infty$. Im Folgenden wird gezeigt, wie mit Hilfe der Übertragungsfunktionen dessen bleibende

Regelabweichung berechnet werden kann. Dies ist ein wichtiges Kriterium, um einen Regler auszulegen. Zu diesem Zweck findet der Endwertsatz der Laplace-theorie Anwendung:

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s F(s). \quad (8.28)$$

Zunächst wird das System mit einem Führungs- bzw. Störgrößensprung beaufschlagt und danach, unter Berücksichtigung der jeweiligen Übertragungsfunktionen $G_F(s)$ bzw. $G_S(s)$, der Endwert für $t \rightarrow \infty$ berechnet. Dies ergibt:

- Führungsgrößensprung:

$$\underbrace{\lim_{t \rightarrow \infty} q_M(t)}_{\text{Endwertsatz}} = \underbrace{\lim_{s \rightarrow 0} s Q_M(s)}_{\text{Endwertsatz}} = \lim_{s \rightarrow 0} s G_F(s) \underbrace{Q_{M,s}(s)}_{\text{Einheitssprung } \frac{1}{s}} = \lim_{s \rightarrow 0} G_F(s). \quad (8.29)$$

- Störgrößensprung:

$$\underbrace{\lim_{t \rightarrow \infty} q_M(t)}_{\text{Endwertsatz}} = \underbrace{\lim_{s \rightarrow 0} s Q_M(s)}_{\text{Endwertsatz}} = \lim_{s \rightarrow 0} s G_S(s) \underbrace{T_S(s)}_{\text{Einheitssprung } \frac{1}{s}} = \lim_{s \rightarrow 0} G_S(s). \quad (8.30)$$

Es kommt zu einer bleibenden Regelabweichung, wenn für $t \rightarrow \infty$ der Soll- und der Istwert voneinander abweichen.

$$\lim_{t \rightarrow \infty} q_M(t) \neq \lim_{t \rightarrow \infty} q_{M,s}(t) \circ \bullet \lim_{s \rightarrow 0} s Q_M(s) \neq \lim_{s \rightarrow 0} s Q_{M,s}(s). \quad (8.31)$$

Folgendes Fehlermaß kann formuliert werden:

$$\begin{aligned} e_\infty &= \lim_{t \rightarrow \infty} q_{M,s}(t) - \lim_{t \rightarrow \infty} q_M(t) \\ &= \lim_{s \rightarrow 0} s Q_{M,s}(s) - \lim_{s \rightarrow 0} s Q_M(s). \end{aligned} \quad (8.32)$$

Dies ergibt für den Fall eines Führungs- bzw. Störgrößensprungs:

- bleibende Regelabweichung bei einem Führungsgrößensprung:

$$\begin{aligned} e_\infty &= \lim_{s \rightarrow 0} s Q_{M,s}(s) - \lim_{s \rightarrow 0} s Q_M(s) \\ &= \lim_{s \rightarrow 0} s \frac{1}{s} - \lim_{s \rightarrow 0} G_F(s) \\ &= 1 - \lim_{s \rightarrow 0} G_F(s), \end{aligned} \quad (8.33)$$

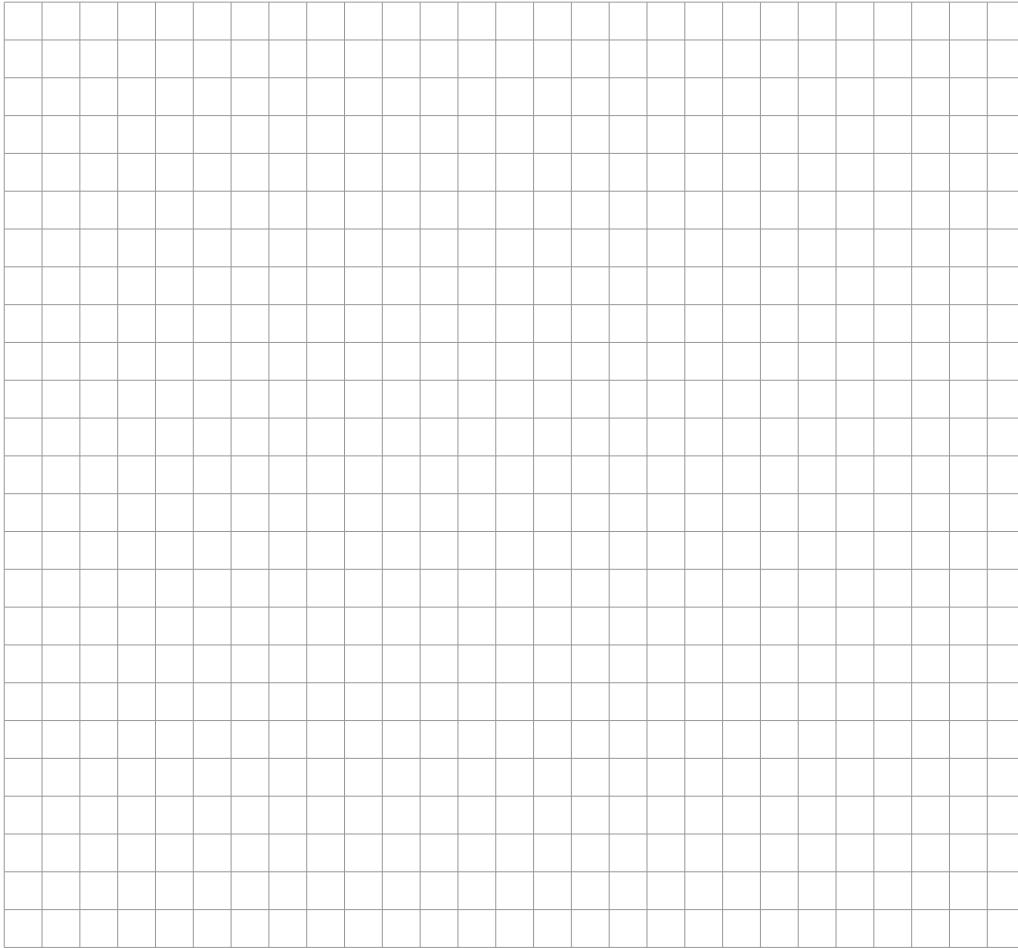
wobei hier $\lim_{t \rightarrow \infty} q_{M,s}(t) = 1$ (bzw. $Q_{M,s} = \frac{1}{s}$) gilt.

- bleibende Regelabweichung bei einem Störgrößenursprung:

$$\begin{aligned} e_\infty &= \lim_{s \rightarrow 0} s Q_{M,s} - \lim_{s \rightarrow 0} s Q_M(s) \\ &= 0 - \lim_{s \rightarrow 0} G_S(s), \end{aligned} \quad (8.34)$$

wobei hier $\lim_{t \rightarrow \infty} q_{M,s}(t) = 0$ (bzw. $Q_{M,s} = 0$) gilt.

Somit ist es ersichtlich, dass anhand der jeweiligen Übertragungsfunktionen die Regelabweichungen schnell und einfach bestimmt werden können.



8.5 Reglerauslegung

Nach einer Analyse der Eigenschaften der geregelten Strecke anhand der Übertragungsfunktionen(en) werden im Folgenden verschiedene Möglichkeiten zur Auslegung eines Reglers vorgestellt. Darunter versteht man Vorgehensweisen zur Bestimmung und Einstellung der Regelparameter unter Berücksichtigung der geforderten Eigenschaften an das zu regelnde System. Im Einzelnen sind dies die Polvorgabe (Abschnitt 8.5.1) und heuristische Einstellregeln (Abschnitt 8.5.2).

8.5.1 Polvorgabe

Bei der Polvorgabe werden die Reglerparameter so gewählt, dass die Nullstellen des Nennerpolynoms vorgegebenen Polen s_i entsprechen. Dies sei im Folgenden am Beispiel einer P/P-Kaskadenregelung für die Roboterachse erläutert.

- Das charakteristische Polynom lautet (siehe Abschnitt 8.4):

$$F(s) = s^2 + \frac{d_{\text{eff}} + K_{i_P}}{J_{\text{ges}}} s + \frac{K_{i_P} K_{a_P}}{J_{\text{ges}}} \quad (8.35)$$

- Die Parameter J_{ges} und d_{eff} sind systembedingt gegeben.
- Die Vorgabe/Auswahl geeigneter Polstellen s_i (hier: $i = 1, 2$) erfolgt mittels Definition einer gewünschten Systemdynamik, wozu die Ausführungen aus Abschnitt 8.4 heranzuziehen sind.

- Die Berechnung der unbekannten Parameter K_{i_p} , K_{a_p} geschieht mittels vergleichendem Polynom:

$$s^2 + \frac{d_{\text{eff}} + K_{i_p}}{J_{\text{ges}}} s + \frac{K_{i_p} K_{a_p}}{J_{\text{ges}}} = (s - s_1)(s - s_2). \quad (8.36)$$

Die Parameter K_{i_p} , K_{a_p} ergeben sich dann aus dem Koeffizientenvergleich.

A large grid of squares, approximately 20 columns by 30 rows, intended for students to work out their calculations for determining the parameters K_{i_p} and K_{a_p} .

8.5.2 Heuristische Einstellregeln

In der Praxis werden auch häufig heuristische Einstellregeln zur Bestimmung geeigneter Regelparameter verwendet. Diese Verfahren benötigen kein mathematisches Modell der Regelstrecke. Es müssen allerdings Experimente an dem zu regelnden System vorgenommen werden (dies ist nicht immer problemlos möglich).

Ein verbreitetes Verfahren zur Bestimmung der Regelparameter von P-, PI-, PD- und PID-Reglern ist die Methode von Ziegler und Nichols:

Der Regelkreis wird mit einem P-Regler geschlossen. Die Reglerverstärkung K_P wird solange erhöht, bis der Ausgang des Regelkreises bei konstantem Eingang eine Dauerschwingung mit der Periode T_{krit} ausführt (Reglerverstärkung $K_{P,\text{krit}}$). Die Regelparameter ergeben sich anschließend in Abhängigkeit der Reglerstruktur zu (siehe auch [ZN42]):

- P-Regler:

$$K_P = 0,5 K_{P,\text{krit}}. \quad (8.37)$$

- PI-Regler:

$$K_P = 0,45 K_{P,\text{krit}}, \quad T_N = 0,85 T_{\text{krit}}. \quad (8.38)$$

- PD-Regler:

$$K_P = 0,55 K_{P,\text{krit}}, \quad T_V = \frac{1}{0,15 T_{\text{krit}}}. \quad (8.39)$$

- PID-Regler:

$$K_P = 0,6 K_{P,\text{krit}}, \quad T_N = 0,5 T_{\text{krit}}, \quad T_V = \frac{1}{0,125 T_{\text{krit}}}. \quad (8.40)$$

Alternativ eignen sich für die heuristische Reglereinstellung beispielsweise die **Einstellregeln nach Chien, Hrones und Reswick** sowie das sogenannte **T-Summen-Verfahren** (siehe [HD04, S. 223–228] und [VK02, S. 29–43]). Diese Ansätze sind an dieser Stelle lediglich der Vollständigkeit halber genannt, eine detaillierte Vorstellung erfolgt nicht.

8.6 Vorsteuerung

Ein Problem von Reglerstrukturen ohne Vorsteuerung ist, dass diese lediglich auf eine bereits vorhandene Regelabweichung (Differenz zwischen der Soll- und der Istgröße) reagieren. Durch geeignete Vorsteuerung kann diese Abweichung (hier: Schleppfehler) reduziert und damit das Führungsverhalten des Systems verbessert werden. Je nach (verbleibenden) Regelfehler und Modellgüte erfolgt eine weitestgehende Kompensation der Nichtlinearitäten und somit eine Linearisierung der Regelstrecke im (zeitlich variablen) Arbeitspunkt. Dadurch kann der Regler in der Praxis „härtter“ ausgelegt werden (höhere Gesamtverstärkung des Reglers), ohne dass das System instabil wird.

Eine Vorsteuerung ermöglicht das Einbringen von Modellwissen mit dem Ziel, den Regelfehler klein zu halten. Verbreitete Ansätze zur Vorsteuerung einer Roboterachse sind:

- Geschwindigkeitsvorsteuerung
- Momentenvorsteuerung

Beide Ansätze werden im Folgenden kurz vorgestellt.

Geschwindigkeitsvorsteuerung

Grundlage für die Einführung der Geschwindigkeitsvorsteuerung ist das in Bild 8.31 gezeigte Blockschaltbild einer P/P-Kaskadenregelung einer Roboterachse.

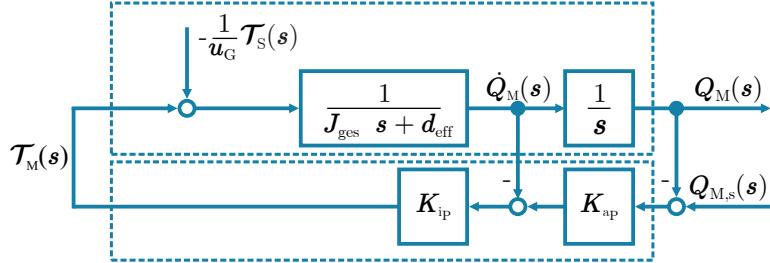


Bild 8.31: P/P-Kaskadenregelung einer Roboterachse

Zur Verbesserung des Führungsverhaltens (Reduktion des Schleppfehlers) kann die Sollgeschwindigkeit $\dot{Q}_{M,s}(s)$ gewichtet auf den Reglerausgang (der äußeren Kaskade) addiert werden. Somit bekommt der innere, die Geschwindigkeit einstellende Regelkreis die Sollgeschwindigkeit direkt als Vorgabe. Dazu addiert sich der aus dem Positionsfehler der Achse resultierende Ausgang des äußeren Regelkreises. Dabei erhält man $\dot{Q}_{M,s}(s)$ entweder durch Differentiation von $Q_{M,s}(s)$ (siehe Bild 8.32) oder durch direkte Vorgabe aus der Bahnplanung (siehe Bild 8.33).

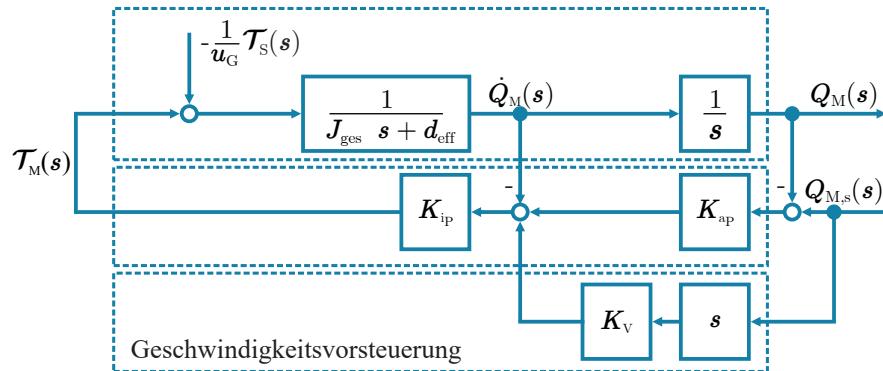


Bild 8.32: P/P-Kaskadenregelung einer Roboterachse mit Geschwindigkeitsvorsteuerung; Erzeugung der Sollgeschwindigkeit durch Differentiation der Führungsgröße

Momentenvorsteuerung

Alternativ zur Vorsteuerung der Geschwindigkeit der Einzelachsregelung kann zur Verbesserung des Führungsverhaltens (Reduktion des Schleppfehlers) auch eine Vorsteuerung des für die gewünschte Bewegung erforderlichen Motormoments erfolgen.

Dazu wird aus der Sollbahn $x_{E,s}(t), \dot{x}_{E,s}(t), \ddot{x}_{E,s}(t)$ bzw. $q_s(t), \dot{q}_s(t), \ddot{q}_s(t)$ das für die Bewegung notwendige Motormoment berechnet (inverse Dynamik) und auf den Reglerausgang addiert werden. Bei der dezentralen Regelung (separate Regelung jeder Roboterachse ohne Berücksichtigung nichtlinearer Verkopplungen) erfolgt dabei die Berechnung des Vorsteuermotormoments τ_{M,v_i} von Antrieb i unter Verwendung des Streckenmodells einer Roboterachse, vgl. Kapitel 8

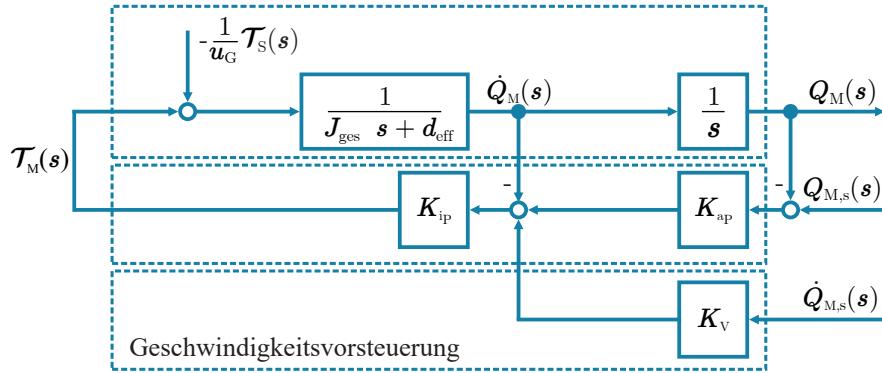


Bild 8.33: P/P-Kaskadenregelung einer Roboterachse mit Geschwindigkeitsvorsteuerung (Sollgeschwindigkeit direkt aus der Bahnplanung, kein Ableiten notwendig)

(Index i zur Kennzeichnung der Roboterachse i wird im Folgenden weggelassen, um die Übersichtlichkeit zu verbessern):

$$J_{\text{ges}} \ddot{q}_M(t) + d_{\text{eff}} \dot{q}_M(t) = \tau_M(t) - \frac{1}{u_G} \tau_S(t), \quad (8.41)$$

wobei J_{ges} die an- und abtriebsseitige Massenträgheit beinhaltet (siehe Abschnitt 7.4). Das Vorsteuermotormoment $\tau_{M,v}$ berechnet sich ausgehend von den ermittelten (identifizierten) Modellparametern \hat{J}_{ges} und \hat{d}_{eff} , die im Idealfall den physikalischen Parametern J_{ges} und d_{eff} entsprechen (in der Realität jedoch davon abweichen), wie folgt:

$$\tau_{M,v}(t) = \hat{J}_{\text{ges}} \ddot{q}_{M,s}(t) + \hat{d}_{\text{eff}} \dot{q}_{M,s}(t) + \underbrace{\frac{1}{u_G} \tau_S(t)}_{=0}. \quad (8.42)$$

Das Störmoment τ_S wird bei der Einzelachs betrachtung vernachlässigt und zu null gesetzt. Somit ergibt sich das aus Vorsteuermotormoment $\tau_{M,v}$ und Reglermoment $\tau_{M,r}$ zusammengesetzte Sollgesamtmoment $\tau_{M,\text{ges}}$:

$$\tau_{M,\text{ges}}(t) = \tau_{M,v}(t) + \tau_{M,r}(t). \quad (8.43)$$

Die Transformation des Vorsteuermoments nach Gleichung (8.42) (ohne Störmoment τ_S) in den Frequenzbereich ergibt:

$$\tau_{M,v}(t) = \hat{J}_{\text{ges}} \ddot{q}_{M,s}(t) + \hat{d}_{\text{eff}} \dot{q}_{M,s}(t) \circ \bullet \mathcal{T}_{M,v}(s) = \hat{J}_{\text{ges}} Q_{M,s}(s) s^2 + \hat{d}_{\text{eff}} Q_{M,s}(s) s. \quad (8.44)$$

Damit kann die aus Kapitel 8 bekannte P/P-Kaskadenregelung einer Einzelachse um den Vorsteueranteil ergänzt werden und man erhält das in Bild 8.34 dargestellte Signalflussdiagramm.

Alternativ können die Sollgeschwindigkeiten und -beschleunigungen auch direkt der Bahnplanung entnommen werden. In diesem Fall stellt sich das Blockschaltbild wie in Bild 8.35 dar.

Das Prinzip der Momentenvorsteuerung ist prinzipiell auf jeden beliebigen Regler anwendbar. Bei Verwendung eines PID-Reglers resultiert beispielsweise das in Bild 8.36 gezeigte Blockschaltbild. Dabei wurde die P/P-Kaskadenregelung durch einen PID-Regler ersetzt. Die Momentenvorsteuerung ist identisch zu Bild 8.35.

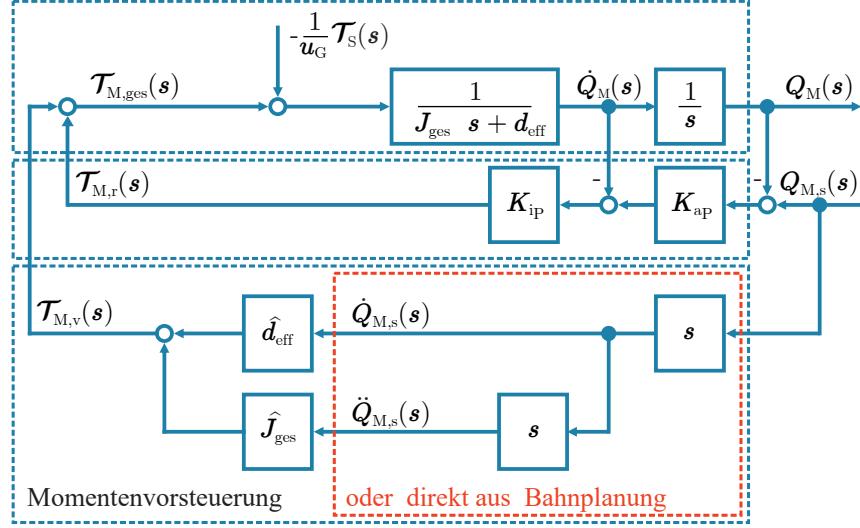


Bild 8.34: P/P-Kaskadenregelung einer Roboterachse mit Momentenvorsteuerung

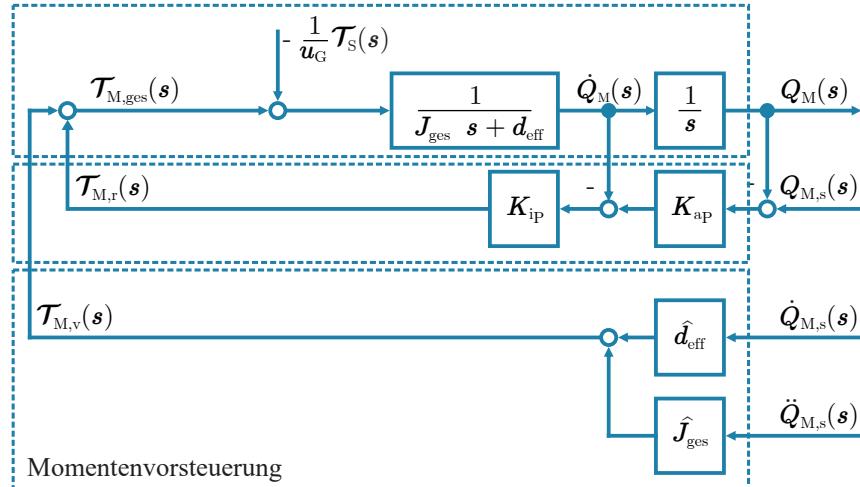


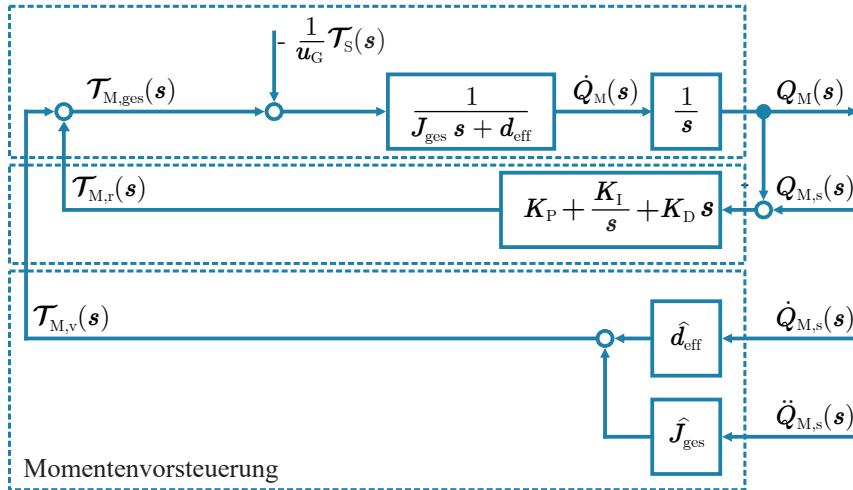
Bild 8.35: P/P-Kaskadenregelung einer Roboterachse mit Momentenvorsteuerung (Sollgeschwindigkeit und -beschleunigung aus der Bahnplanung)

Die Eigenschaften des Prinzips der Momentenvorsteuerung bei einer dezentralen Regelung lassen sich wie folgt zusammenfassen:

- Einfach zu implementieren und vielseitig einsetzbar,
- bei starker Abweichung zwischen Soll- und Istbahn ist auch eine Verschlechterung der Performance denkbar (da ausschließlich auf Sollwerten basierend),
- keine Berücksichtigung der nichtlinearen Verkopplungen zwischen den einzelnen Roboterachsen (da Einzelachsregelung),
- Identifikation der physikalischen Parameter J_{ges} und d_{eff} erforderlich.

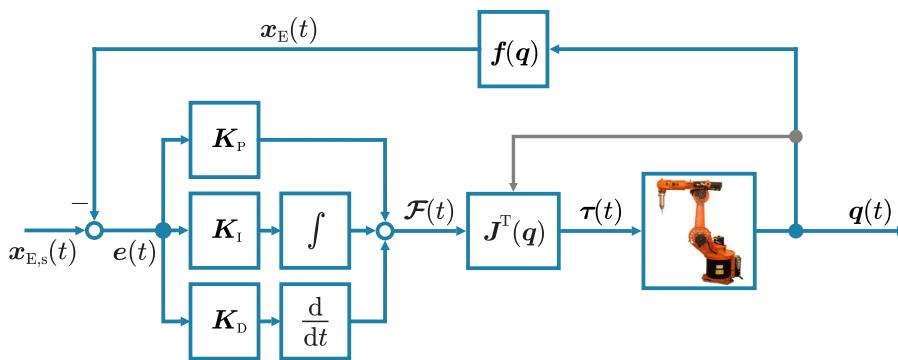
8.7 Einzelachsregelung im Taskspace

Die Regelung im Taskspace läuft ähnlich wie die Regelung im Jointspace ab. Lediglich die Gelenkvariablen q, \dot{q}, \ddot{q} sind sinngemäß durch die Endeffektorvariablen $x_E, \dot{x}_E, \ddot{x}_E$ zu ersetzen. Die Regelung ist hier im gesamten Taskspace des Roboters formuliert (Position und Orientierung).

**Bild 8.36:** PID-Regelung einer Roboterachse mit Momentenvorsteuerung

Die Reglerverstärkungen K_P , K_I und K_D sind deshalb Diagonalmatrizen. Die einzelnen Achsen werden jedoch voneinander unabhängig geregelt, wenn von der geometrischen Verknüpfung über die Kinematik abgesehen wird.

Das Beispiel einer PID-Regelung im Taskspace ist in Bild 8.37 gezeigt. In der hier vorgestellten

**Bild 8.37:** Einzelachsregelung im Taskspace

Form findet keine Beachtung von dynamischer Verkopplung der einzelnen Roboterachsen statt. Dies ist Gegenstand der Ausführungen in Kapitel 9. Zusätzlich ist die Berechnung der direkten Kinematik notwendig, was jedoch kein Problem darstellt (vgl. Kapitel 3).

Bei der gezeigten Reglerstruktur in Bild 8.37 wird der Reglerausgang als Kraft interpretiert. Anschaulich erscheint dies plausibel, wenn man bedenkt, dass eine Kraft den Roboter auf seine Sollbahn lenkt. Alternativ kann die Berechnung des D-Anteils auch unter Verwendung der Sollgeschwindigkeit aus der Bahnplanung erfolgen. Dies wird in Bild 8.38 dargestellt. Die Berechnung der kartesischen Istgeschwindigkeit geschieht dann auf Basis der Jacobi-Matrix.

Zwei Aspekte sind bei der Einzelachsregelung im Taskspace problematisch:

- Bei großen Abweichungen zwischen Soll- und Istwinkel bereitet die Regelung der Endeffektororientierung Probleme. Grund dafür ist, dass bei den meisten Winkelrepräsentationen die Orientierung im mitgedrehten Koordinatensystem angegeben wird. Die jeweiligen Winkel der Soll- und Istorientierung des Endeffektors sind deshalb in unterschiedlichen Koordinatensystemen definiert, so dass eine einfache Berechnung des Orientierungsfehlers durch Subtraktion bei starken Fehlern zwar rechnerisch möglich aber geometrisch falsch ist.

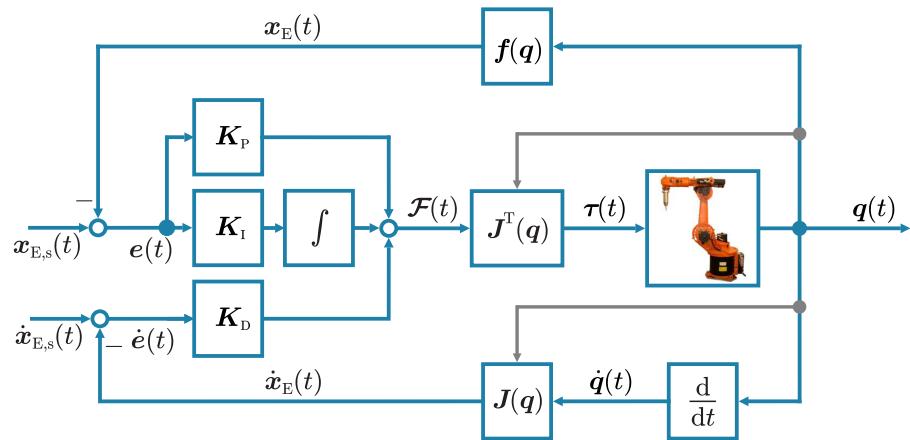


Bild 8.38: Einzelachsregelung im Taskspace (Sollgeschwindigkeit aus Bahnplanung)

- Der Reglerausgang wird mit der transponierten Jacobi-Matrix gewichtet. Diese hängt jedoch stark von der Konfiguration des Roboters ab, wodurch die Kraft stark unterschiedlich skaliert wird. Dies kann ggf. zu sehr großen Solldrehmomenten $\tau(t)$ führen.

Aus diesen Gründen finden die in diesem Abschnitt vorgestellten Konzepte zur Achsregelung im Taskspace kaum Verwendung.

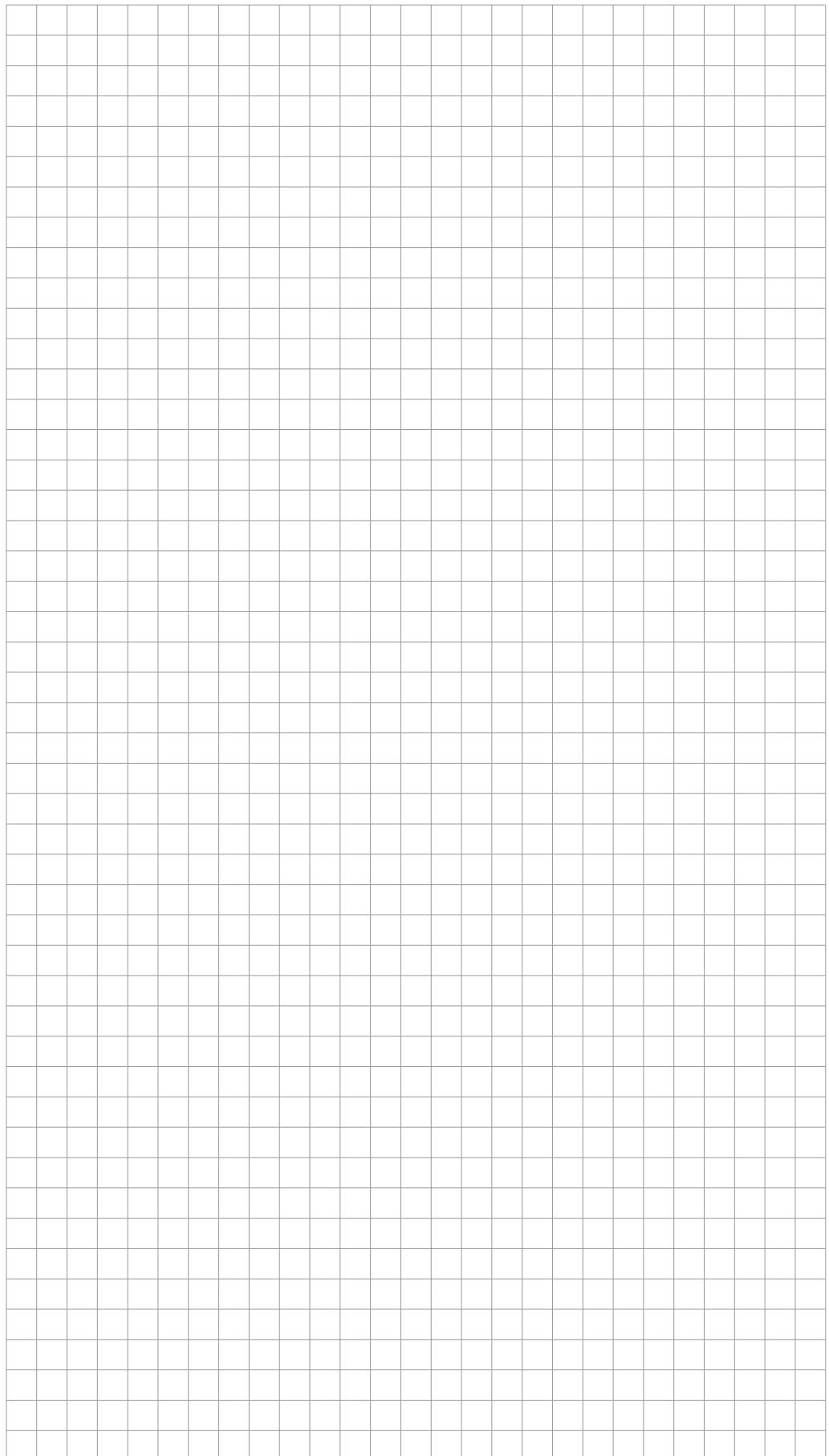
Fragen zur Selbstkontrolle

1. Nennen Sie die Ihnen bekannten klassischen Rückführungen und erläutern Sie die einzelnen Prinzipien! Skizzieren Sie zusätzlich zwei sinnvolle Kombinationen klassischer Rückführungen!
 2. Zeichnen Sie das Blockschaltbild einer klassischen Roboterachse im Frequenzbereich und geben Sie die Führungsgrößenübertragungsfunktion an!
 3. Erläutern Sie die Systemeigenschaften „stabil“, „instabil“, „grenzstabil“ und „schwingungsfähig“ basierend auf der Lage der Polstellen des charakteristischen Polynoms!
 4. Für einen Regler gelte folgende Führungsgrößenübertragungsfunktion:

$$G_F(s) = \frac{Q_M(s)}{Q_{M_s}(s)} = \frac{K_P K_D}{0.25 s^2 + (2 - K_P) s + K_P K_D}$$

Bestimmen Sie die Reglerkoeffizienten K_P und K_D so, dass zwei Pole des geschlossenen Regelkreises bei $s_{1,2} = -2$ liegen!





9 Mehrachsregelung

9.1 Einführung

Dieses Kapitel widmet sich der *zentralen* Mehrachsregelung. Im Gegensatz zur bereits bekannten *dezentralen* Einzelachsregelung berücksichtigt die Mehrachsregelung die dynamischen Verkopplungen der Roboterachsen explizit. Die zugrunde liegenden Konzepte basieren auf dem Prinzip der Momentenvorsteuerung, die für eine einzelne Achse im vorherigen Kapitel beschrieben wurde (Abschnitt 8.6). Die anschließenden Abschnitte behandeln zwei verschiedene zentrale, modellbasierte Ansätze: Die Feedforward-Steuerung (Abschnitt 9.2) und die Feedback-Regelung (Abschnitt 9.3).

Als weiterführende Literatur wird empfohlen:

- L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Berlin, Deutschland: Springer, 2000, S. 213–226
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 310–315

Bisher wurden Rückkopplungsstrukturen vorgestellt, bei denen jede Achse des Roboters separat geregelt wird (dezentrale Einzelachsregelung). Hierbei bleiben die nichtlinearen Verkopplungen zwischen den einzelnen Achsen i unberücksichtigt (bzw. werden als Störgröße τ_{S_i} modelliert). Anhand des bekannten dynamischen Modells

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \mathbf{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \mathbf{g}(\boldsymbol{q}) + \mathbf{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \quad (9.1)$$

ist allerdings zu sehen, dass es sich bei einem Roboter um ein verkoppeltes, nichtlineares System mit n Freiheitsgraden handelt (Multi-Input, Multi-Output: MIMO). Bei hochdynamischen Roboterbewegungen führt die Nichtbeachtung der Verkopplungen zu einem ungewollt großen Schleppfehler bzw. zu einer ungewollt großen Bahnabweichung. Bei der zentralen, modellbasierten Regelung werden die nichtlinearen Verkopplungen explizit berücksichtigt, wobei das inverse dynamische Modell nach Gleichung (9.1) die Grundlage bildet. Zu beachten ist, dass im Gegensatz zu Abschnitt 8.6 die Betrachtungen nun abtriebsseitig (also nach dem Getriebe) erfolgen – für die Getriebeübersetzung u_{G_i} gilt gedanklich $u_{G_i} = 1$ für alle Antriebe i .

Im Folgenden werden zwei verschiedene Ansätze vorgestellt: zum einen die Feedforward-Steuerung (Abschnitt 9.2) und zum anderen die Feedback-Regelung (Abschnitt 9.3).

9.2 Feedforward-Steuerung (Momentenvorsteuerung)

Bei der Feedforward-Steuerung werden das Vorsteuermotormoment $\tau_v(t)$ und der Reglerausgang $\tau_r(t)$ (Anteil Motormoment aus Regler) addiert (vgl. Bild 9.1). Das aus Abschnitt 8.6 bekannte Verfahren wird also lediglich auf den gesamten Roboter erweitert. Auch hier gilt, dass sich das Vorsteuermoment anhand identifizierter Modellparameter berechnet. Diese sind in Bild 9.1 mit einem Dach gekennzeichnet. Die Berechnung des Vorsteuermotormoments geschieht dabei unter Verwendung der inversen Dynamik in Abhängigkeit der Sollbahn $\boldsymbol{q}_s(t), \dot{\boldsymbol{q}}_s(t), \ddot{\boldsymbol{q}}_s(t)$. Es handelt sich demnach um eine Steuerung unter ausschließlicher Betrachtung der Sollgrößen.

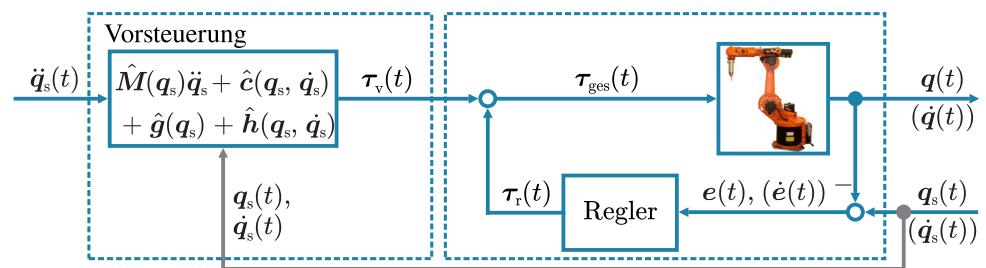


Bild 9.1: Feedforward-Steuerung (Momentenvorsteuerung)

Die Eigenschaften bzw. Voraussetzungen der Feedforward-Steuerung (Momentenvorsteuerung) lassen sich wie folgt zusammenfassen:

- Die Berechnung des Vorsteuermotormoments geschieht ausschließlich in Abhängigkeit der Sollgrößen $q_s(t)$, $\dot{q}_s(t)$, $\ddot{q}_s(t)$. Für eine gute Qualität muss die Abweichung zwischen Soll- und Istbahn hinreichend klein sein.
- Es ist eine hohe Modellgüte der inversen Dynamik erforderlich. Die identifizierten Parameter des inversen dynamischen Modells müssen gut mit der Realität übereinstimmen.

Die im Folgenden eingeführte Feedback-Regelung berücksichtigt zusätzlich die Istgrößen des Roboters und erlaubt dadurch eine direkte Beeinflussung der Fehlerdynamik.

9.3 Feedback-Regelung

Grundlage für die nun folgenden Ausführungen ist das in Bild 9.2 dargestellte Blockschaltbild. Es besteht aus einem inneren Teil, der auf der inversen Dynamik basiert, und einer Rückführung, die den eigentlichen Regler beinhaltet.

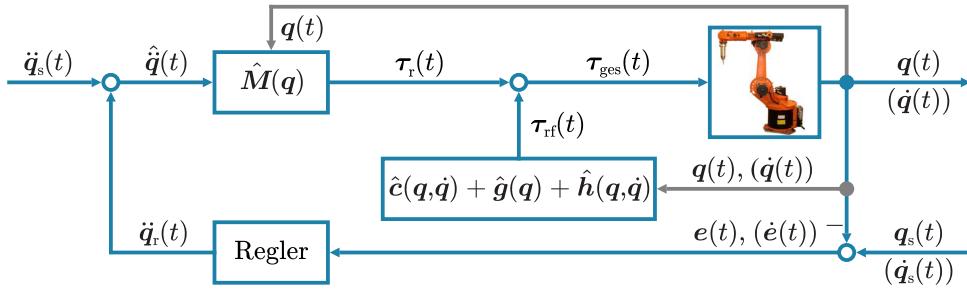


Bild 9.2: Feedback-Regelung

Bei der Feedback-Regelung erfolgt eine Linearisierung des Gesamtsystems durch Rückführung der nichtlinearen Verkopplungen $\hat{c}(q, \dot{q})$, $\hat{g}(q)$, $\hat{h}(q, \dot{q})$. Diese werden anhand der Istgrößen des Roboters bestimmt und nicht, wie bei der Feedforward-Steuerung, anhand der Sollgrößen. Das Gesamtmotormoment $\tau_{\text{ges}}(t)$ berechnet sich nach folgendem Zusammenhang:

$$\tau_{\text{ges}}(t) = \underbrace{\hat{M}(q) \hat{q}(t)}_{\tau_r(t)} + \underbrace{\hat{c}(q, \dot{q}) + \hat{g}(q) + \hat{h}(q, \dot{q})}_{\tau_{rf}(t)} \quad (9.2)$$

und besteht aus zwei Komponenten, wie in Bild 9.3 gezeigt. Der linke Teil τ_r basiert auf der geschätzten Massenmatrix \hat{M} und einer noch genauer zu definierenden Beschleunigung \hat{q} . Der rechte Teil τ_{rf} ist der Vorsteueranteil, der sich anhand der (identifizierten) nichtlinearen Verkopplungen sowie den Istgrößen des Roboters berechnet.

Die Eingangsgröße der Feedback-Regelung ist die Beschleunigung $\hat{q}(t)$, die sich aus dem Reglerausgang \ddot{q}_r und der Sollbeschleunigung \ddot{q}_s zusammensetzt:

$$\hat{q}(t) = \ddot{q}_s(t) + \ddot{q}_r(t). \quad (9.3)$$

Bei hinreichend gutem Modell (Übereinstimmung der Modellparameter mit den tatsächlichen physikalischen Größen) ist das System nach Bild 9.3 linear und entkoppelt bezüglich seiner Eingangsgröße \hat{q} . Es gilt:

$$\hat{q}(t) = \ddot{q}(t). \quad (9.4)$$

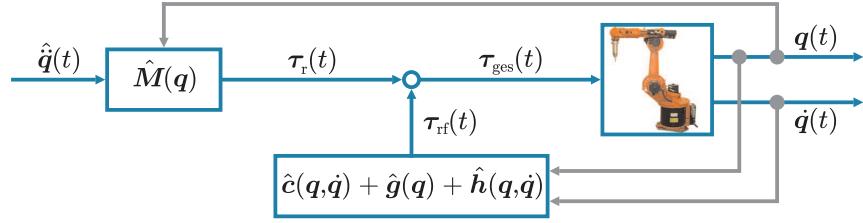


Bild 9.3: Feedback-Regelung (Rückführung nicht explizit dargestellt)

Eigenschaften der Feedback-Regelung

Die nichtlinearen Verkopplungen lassen sich modellbasiert berechnen (vgl. Gleichung (9.2)):

$$\tau_{\text{rf}} = \hat{c}(q, \dot{q}) + \hat{g}(q) + \hat{h}(q, \dot{q}). \quad (9.5)$$

Der in Abbildung 9.3 dargestellte linke Term berechnet sich zu

$$\tau_r = \hat{M}(q)(\ddot{q}_s + \ddot{q}_r) = \hat{M}(q) \underbrace{(\ddot{q}_s + F_r e(t))}_{\hat{q}}. \quad (9.6)$$

Für die gesamte Stellgröße gilt:

$$\tau_{\text{ges}} = \tau_{\text{rf}} + \tau_r. \quad (9.7)$$

Das inverse dynamische Modell des Roboters lautet:

$$\tau = M(q)\ddot{q} + c(q, \dot{q}) + g(q) + h(q, \dot{q}). \quad (9.8)$$

Aus der Forderung

$$\tau_{\text{ges}} \stackrel{!}{=} \tau \quad (9.9)$$

resultiert

$$\begin{aligned} & \hat{c}(q, \dot{q}) + \hat{g}(q) + \hat{h}(q, \dot{q}) + \hat{M}(q)(\ddot{q}_s + F_r e(t)) \\ & \stackrel{!}{=} M(q)\ddot{q} + c(q, \dot{q}) + g(q) + h(q, \dot{q}). \end{aligned} \quad (9.10)$$

Setzt man ein gutes Modell voraus ($\hat{M} \approx M$, $\hat{c} \approx c$, $\hat{g} \approx g$ und $\hat{h} \approx h$, so ergibt sich aus Gleichung (9.10)):

$$\begin{aligned} M(q)(\ddot{q}_s + F_r e(t)) &= M(q)\ddot{q} & | \rightarrow M^{-1}(q), \\ \underbrace{\ddot{q}_s - \ddot{q}}_{\ddot{e}} + F_r e(t) &= \mathbf{0}. \end{aligned} \quad (9.11)$$

Für einen PD-Regler

$$F_r = K_P + sK_D \quad (K_P = \text{diag}(k_P); \quad K_D = \text{diag}(k_D), \quad \text{mit positiven Einträgen}) \quad (9.12)$$

resultiert die (homogene) Fehlerdifferentialgleichung zweiter Ordnung:

$$\ddot{e} + K_D \dot{e} + K_P e = 0, \quad (9.13)$$

die für geeignete Reglerparameter gegen null konvergiert.

Für die Feedback-Regelung gelten folgende Eigenschaften:

- keine bleibende Regelabweichung,
- Gleichung (9.13) gilt auch für große Reglerfehler,
- keine Verkopplung der Achsenfehler (n entkoppelte Ein- und Ausgänge),
- Reglerdynamik über K_D und K_P in gewissen Grenzen frei einstellbar.

Anwendungsbeispiel – Auslegung der Reglerverstärkung

Aufgrund der Entkopplung können die Achsen einzeln betrachtet werden (der Index i wurde aus Gründen der Lesbarkeit weggelassen):

$$\ddot{e} + k_D \dot{e} + k_P e = 0 \quad (\text{Fehlerdifferentialgleichung}).$$

Das charakteristische Polynom lautet:

$$\lambda^2 + k_D \lambda + k_P = 0 \quad (\text{Ansatz: } e^{\lambda t}),$$

mit den Nullstellen:

$$\lambda_{1,2} = \frac{-k_D \pm \sqrt{k_D^2 - 4k_P}}{2}.$$

Daraus ergeben sich folgende Fälle:

$$1. \ k_D^2 - 4k_P > 0$$

zwei unterschiedliche reelle negative Nullstellen mit folgender Lösung:

$$e(t) = e_{01} e^{\lambda_1 t} + e_{02} e^{\lambda_2 t}.$$

$$2. \ k_D^2 - 4k_P = 0$$

zwei gleiche reelle negative Nullstellen ($\lambda_1 = \lambda_2 = \lambda$) mit der Lösung:

$$e(t) = (e_{01} + e_{02}t) e^{\lambda t}.$$

$$3. \ k_D^2 - 4k_P < 0$$

zwei konjugiert komplexe Nullstellen (unterschiedlich: $\lambda_1 = \overline{\lambda_2}$) mit der schwingungsfähigen Lösung:

$$\begin{aligned} e(t) &= (e_{01} \cos(\operatorname{Im}\{\lambda_1\}t) + e_{02} \sin(\operatorname{Im}\{\lambda_1\}t)) e^{\operatorname{Re}\{\lambda_1\}t} \\ &= A \sin(\operatorname{Im}\{\lambda_1\}t + \gamma), \end{aligned}$$

mit

$$\begin{aligned} A &= e^{\operatorname{Re}\{\lambda_1\}t} \sqrt{e_{01}^2 + e_{02}^2}, \\ \gamma &= \arctan(e_{01}, e_{02}). \end{aligned}$$

Die Koeffizienten e_{01} und e_{02} ergeben sich aus dem Anfangswertproblem. Je nach gewünschtem dynamischen Verhalten sind k_D und k_P geeignet zu wählen. Bei Einsatz eines PD-Reglers resultiert das in Bild 9.4 gezeigte Blockschaltbild.

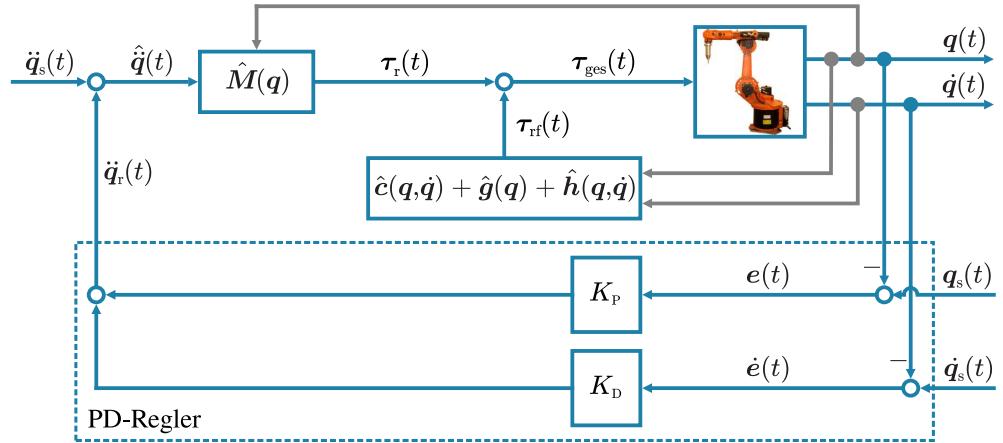


Bild 9.4: Feedback-Regelung unter Verwendung eines PD-Reglers

Wesentliche Eigenschaften bzw. Voraussetzungen der Feedback-Regelung sind:

- Die Berechnung der inversen Dynamik ist in Abhängigkeit von Istgrößen im Regelakt notwendig (unkritisch).
- Die Feedback-Regelung erfordert eine hohe Modellgüte. Es besteht die Gefahr einer Überkompensation, die unerwünschte Schwingungen zur Folge haben kann. Die geschätzten Parameter der inversen Dynamik müssen also eine hohe Güte haben (vgl. Vorlesung „Robotik II“).
- Aufgrund der speziellen Struktur ist eine direkte Beeinflussung der bezüglich der Freiheitsgrade entkoppelten Fehlerdynamik möglich.

9.4 Verzahnung der Vorlesungsinhalte

Wie zu Beginn der Veranstaltung erwähnt, stellt die von Ihnen gewählte Vorlesung Grundlagen und Methoden der modernen Robotik vor. Ziel ist die Vermittlung von Kenntnissen, die für die Modellierung und Regelung serieller Kinematiken erforderlich sind. Bei erneuter Betrachtung der Verzahnung der Vorlesungsinhalte in Form eines Regelkreises (Bild 9.5) wird deutlich, dass zu diesem Zeitpunkt die erforderlichen Kenntnisse für eine erfolgreiche Verwendung seriellkinematischer Mechanismen in der Praxis vermittelt wurden.

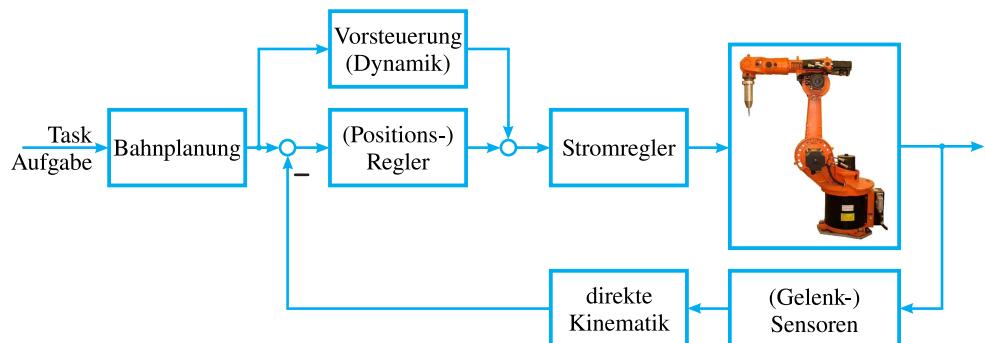


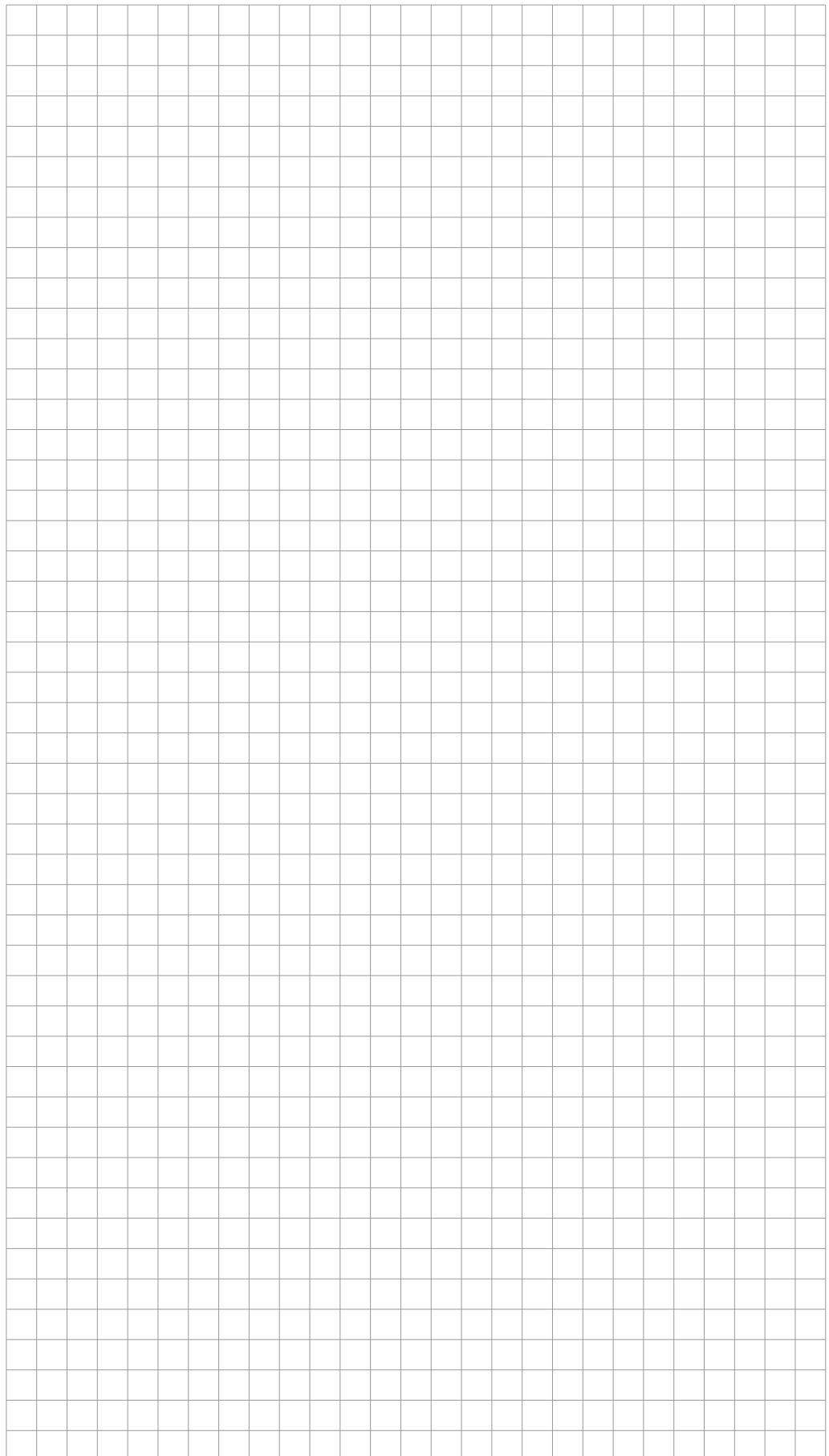
Bild 9.5: Verzahnung der Vorlesungsinhalte in Form eines Regelkreises

Auf diesen Grundlagen aufbauend behandelt das folgende Kapitel weiterführend fortgeschrittene Regelungsverfahren, wie beispielsweise Impedanz-, Admittanz- und hybride Kraft-/Lageregelung.

Fragen zur Selbstkontrolle

1. Erläutern Sie anhand eines geeigneten Blockschaltbildes das Prinzip einer dezentralen Einzelachsregelung mit Momentenvorsteuerung!
2. Welche Arten der zentralen modellbasierten Regelung sind Ihnen bekannt? Erläutern Sie die Eigenschaften (Unterschiede) und Voraussetzung der von Ihnen genannten Ansätze!
3. Skizzieren Sie das Blockschaltbild einer zentralen und modellbasierten Feedforward Steuerung (inkl. Regler) im Zeitbereich! Wie verändert sich das Schaltbild bzw. die implizierten Funktionen unter Berücksichtigung von Getriebeübersetzungen?

A large grid of squares, approximately 20 columns by 25 rows, intended for drawing a block diagram.



10 Fortgeschrittene Regelverfahren

Nachdem in Kapitel 8 und Kapitel 9 die Grundlagen der Einzel- und Mehrachsregelung hergeleitet wurden, sind in diesem Kapitel fortgeschrittene Regelverfahren von besonderem Interesse. Mit diesen ist die kontrollierte Interaktion eines Roboters mit seiner meist unbekannten Umgebung möglich. Im Folgenden werden die Steifigkeits- (Abschnitt 10.2 bis Abschnitt 10.4) und Impedanzregelung (Abschnitt 10.5), die direkte Kraftregelung (Abschnitt 10.6) sowie die hybride Kraft-/Lageregelung (Abschnitt 10.7) eingeführt. Im Rahmen der hybriden Kraft-/Lageregelung wird auch auf die benötigten natürlichen und künstlichen Randbedingungen eingegangen. In den folgenden Ausführungen ist die Antwort auf folgende Frage gesucht: „Was habe ich zu beachten bzw. welche Auswirkung hat es auf die Regelung, wenn die Wechselwirkungen zwischen Roboter und Umwelt nicht vernachlässigbar sind?“

Als weiterführende Literatur wird empfohlen:

- L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Berlin, Deutschland: Springer, 2000, S. 257–280
- J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004, S. 365–389
- W. Khalil und E. Dombre. *Modeling, Identification & Control of Robots*. New York, USA: Routledge, 2002, S. 377–393
- G. Zeng und A. Hemami. „An Overview of Robot Force Control“. In: *Robotica* 15.5 (1997), S. 473–482
- J.K. Salisbury. „Active Stiffness Control of a Manipulator in Cartesian Coordinates“. In: *Proceedings of the IEEE Conference on Decision and Control*. Albuquerque, USA, 1980, S. 95–100

10.1 Einleitung

Die bisher vorgestellten Regelungsverfahren eines Roboters sind für den Fall konzipiert, dass kein bzw. ein vernachlässigbarer Kontakt des Endeffektors mit seiner Umgebung besteht. Viele (industrielle) Anwendungen, wie beispielsweise Entgraten, Polieren, Schleifen und Fügen, erfordern jedoch einen (kontrollierten) Kontakt des Endeffektors mit seiner (z. T. unbekannten) Umgebung.

Aufgrund von Modellierungsfehlern und begrenzter Lagegenauigkeit eines Roboters sind Positionssregler (vgl. Kapitel 8 und Kapitel 9) für Aufgaben mit direktem Umweltkontakt nicht praktikabel. Ein unvorhergesehener Kontakt des „steifen“ Roboters mit einer „steifen“ Umgebung führt sehr schnell zu hohen Kräften, die in einer Beschädigung von Roboter und Umwelt resultieren. Aus diesem Grund sind weiterführende Regelungsverfahren notwendig, die Kontaktkräfte explizit berücksichtigen. Dies ist Bestandteil des vorliegenden Kapitels.

10.2 Passive Steifigkeitsregelung

Bei der Steifigkeitsregelung werden zwei unterschiedliche Ansätze behandelt: die rein passive Steifigkeitsregelung unter Zuhilfenahme einer zusätzlichen mechanischen Nachgiebigkeit (Abschnitt 10.2) und die aktive Steifigkeitsregelung (Abschnitt 10.3 bis Abschnitt 10.4). Die passive

Steifigkeitsregelung ist eine einfache Möglichkeit, die Kontaktkräfte zwischen Endeffektor und Umwelt ohne zusätzliche Sensoren zu reduzieren. Hierzu wird an dem Endeffektor des Roboters ein sogenanntes Remote Center of Compliance (RCC), im Wesentlichen bestehend aus Federn (und Dämpfern), montiert, siehe Bild 10.1. Dieses besitzt in bestimmte (vorgegebene) Raumrichtungen eine hohe Nachgiebigkeit/Elastizität, so dass unvermeidliche Lageungenauigkeiten gezielt ausgeglichen werden können (Vermeidung hoher Kontaktkräfte durch Reduktion der Steifigkeit). Anschaulich für den eindimensionalen Fall bedeutet dies für die Steifigkeit in Raumrichtung i :

$$\frac{1}{k_{\text{ges}_i}} = \frac{1}{k_{R_i}} + \frac{1}{k_{C_i}}, \quad (10.1)$$

wobei $\frac{1}{k_{\text{ges}_i}}$ die Gesamtnachgiebigkeit (Kehrwert der Steifigkeit) ist, $\frac{1}{k_{R_i}}$ die Nachgiebigkeit des Roboters und $\frac{1}{k_{C_i}}$ die Nachgiebigkeit des RCC. Die Raumrichtungen mit einer hohen Nachgiebigkeit bzw. Steifigkeit sind von der jeweiligen Aufgabe abhängig.

Beispiel

Einführen eines Bolzens in eine Bohrung

Das RCC gleicht Lageungenauigkeiten aus, so dass der Bolzen ohne Verkanten in die Bohrung eingeführt werden kann. Hohe Steifigkeit existiert lediglich in „Einführrichtung“, die anderen Richtungen sind weich, haben also eine geringe Steifigkeit. Der Einführvorgang ist in Bild 10.2 dargestellt. Zu erkennen ist, dass ein Verkanten des Bolzens aufgrund von Lagefehlern des Roboters (was zu hohen Kräften/Momenten führen würde) durch die mechanische Nachgiebigkeit vermieden wird.



Bild 10.1: Remote Centre of Compliance (ATI Industrial Automation)

Die wesentlichen Merkmale der passiven Steifigkeitsregelung sind:

- rein mechanischer Ansatz, keine komplexe Regelstrategie erforderlich
- RCC in der Regel nicht universell einsetzbar, da die Nachgiebigkeiten an die jeweilige Aufgabe angepasst werden müssen

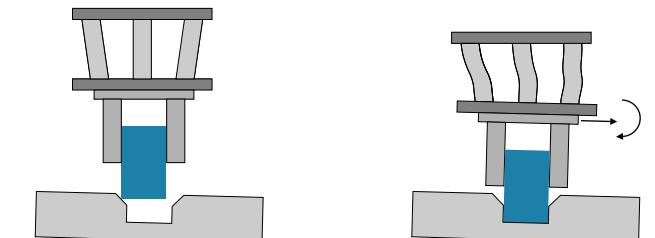


Bild 10.2: Einführen eines Bolzens in eine Bohrung mittels RCC

10.3 Aktive Steifigkeitsregelung

Die aktive Steifigkeitsregelung als Sonderfall der in Abschnitt 10.5 erläuterten Impedanzregelung erlaubt eine gleichzeitige Beeinflussung von Endeffektorlage x_E und Kraft \mathcal{F} durch Vorgabe gewünschter (kartesischer) Steifigkeiten

$$\mathbf{K}_E = \text{diag}(\mathbf{k}_E) \quad (10.2)$$

für die einzelnen Freiheitsgrade i des Endeffektors. Analog zur passiven Steifigkeitsregelung ist im Idealfall keine zusätzliche Sensorik notwendig. Abschnitt 10.3.1 erläutert hierzu die erforderlichen Grundlagen am vereinfachten Robotmodell. Da die Vernachlässigung des Eigengewichts des Roboters zu einem Lagefehler des Endeffektors führt, ist eine Gravitationskompensation nach Abschnitt 10.3.2 empfehlenswert. Die explizite Berücksichtigung des Kontakts zwischen (gravitationskompensierten) Roboter und Umgebung ist Gegenstand der Ausführungen von Abschnitt 10.3.3.

10.3.1 Grundlagen

Der Zusammenhang zwischen der Abweichung (auch Auslenkung der Feder) Δx_E und der (Kontakt-)Kraft \mathcal{F}_R wird bei der aktiven Steifigkeitsregelung im Allgemeinen als Hooke'sche Feder (linear) modelliert:

$$\mathcal{F}_R = \mathbf{K}_E (x_{E,s} - x_E) = \mathbf{K}_E \Delta x_E = \mathbf{K}_E \mathbf{J}(q) \Delta q. \quad (10.3)$$

Hierbei beschreibt $x_{E,s}$ die Solllage des Endeffektors und x_E die Istlage des Endeffektors. Bild 10.3 veranschaulicht den in Gleichung (10.3) beschriebenen Zusammenhang.

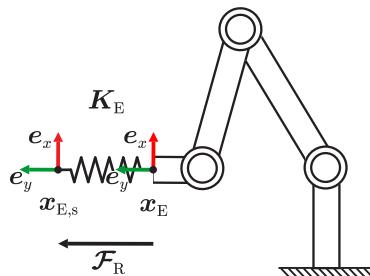


Bild 10.3: Modellierung des Roboterverhaltens als Hooke'sche Feder

In Abhängigkeit der gewählten Steifigkeiten k_{E_i} sowie der Lageabweichung Δx_E ergibt sich demnach eine entsprechende „virtuelle“ Federkraft \mathcal{F}_R . Diese kartesische Sollkraft wiederum wird von den Antrieben erzeugt und sorgt dafür, dass der Roboter versucht, die Lageabweichung Δx_E zu reduzieren. Je nach Steifigkeit resultiert ein hohes oder geringes Motordrehmoment. Mit der aus Kapitel 4 bekannten Gleichung $\tau = \mathbf{J}^T(q) \mathcal{F}$ lässt sich die Kraft \mathcal{F}_R in die entsprechenden Sollmomente $\tau_{M,s}$ umrechnen:

$$\begin{aligned} \tau_{M,s} &= \text{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) \mathbf{J}^T(q) \mathcal{F}_R \\ &= \text{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) \mathbf{J}^T(q) \mathbf{K}_E \mathbf{J}(q) \Delta q. \end{aligned} \quad (10.4)$$

Zum besseren Verständnis wird im Folgenden für die Getriebeübersetzung $u_{G_i} = 1$ angenommen.

Anmerkung

Damit gilt:

$$\tau = \tau_M \quad \text{und} \quad q = q_M. \quad (10.5)$$

Dadurch wird der Roboter rücktriebbar. Im Allgemeinen werden bei Robotern allerdings hochübertragende Getriebe eingesetzt, die nicht rücktriebbar (reibungsbehaftet) sind. Eine Ausnahme ist beispielsweise der auf Seilzügen basierende Whole Arm Manipulator (WAM, Barrett Technology), vgl. Bild 10.4. Dieser Aspekt wird im weiteren Verlauf (vgl. Abschnitt 10.3.3) nochmals genauer



Bild 10.4: Whole Arm Manipulator, Quelle Barrett Technology

erläutert.

Es ergibt sich das Sollmotormoment:

$$\tau_s = J^T(q) K_E J(q) \Delta q = K_q(q) \Delta q. \quad (10.6)$$

Die Matrix $K_q(q)$ wird als Gelenksteifigkeitsmatrix bezeichnet. Das dazugehörige Blockschaltbild ist Bild 10.5 zu entnehmen.

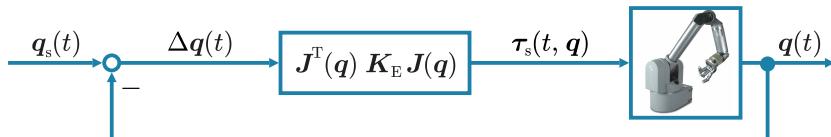


Bild 10.5: Blockschaltbild einer aktiven Steifigkeitsregelung im Jointspace

Bei der Wahl geeigneter Steifigkeiten $K_E = \text{diag}(k_E)$ ist Folgendes zu beachten:

- Steifigkeiten in Richtungen eines „kritischen“ Umweltkontakts sind möglichst klein zu wählen. Dies resultiert in einer Reduktion der Motordrehmomente bei einer Abweichung und damit in einer Reduktion der Kontaktkräfte.
- Steifigkeiten in Richtungen ohne „kritischen“ Umweltkontakt sind möglichst groß zu wählen. Dies resultiert in hohen Motordrehmomenten (auch bei kleinen Lageabweichungen) und damit in einer Reduktion des Bahnfehlers.

Folgende Faktoren wurden noch nicht berücksichtigt und sind Gegenstand nachfolgender Ausführungen:

- Eigengewicht des Roboters (Fehlerquelle), Gravitationskompensation notwendig (siehe Abschnitt 10.3.2),
- Wechselwirkung mit Umgebung noch nicht dargestellt (siehe Abschnitt 10.3.3),
- dynamische Kräfte im Allgemeinen unberücksichtigt, da zumeist langsame Bewegung,

- Aspekt der Rücktreibbarkeit der Antriebe.

10.3.2 Gravitationskompensation

Aufgrund des Eigengewichts des Roboters sowie des Gewichts des Handhabungsobjekts wirken lageabhängige Kräfte bzw. Momente τ_G auf die einzelnen Antriebe. Diese Kräfte bzw. Momente haben eine bleibende Regelabweichung zur Folge (vgl. Bild 10.6). Für eine exakte (genaue)

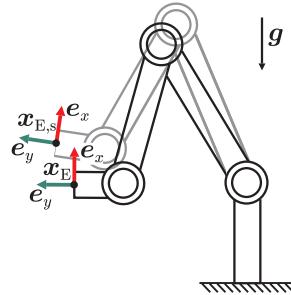


Bild 10.6: Endeffektorabweichung aufgrund der Gravitation

Regelung sind diese Kräfte/Momente auf die gewünschten/berechneten Sollkräfte τ_s zu addieren (Gravitationskompensation):

$$\tau_{\text{ges}}(t, \mathbf{q}) = \tau_s(t, \mathbf{q}) + \tau_G(\mathbf{q}). \quad (10.7)$$

Die Berechnung von $\tau_G = (\tau_{G_1}, \dots, \tau_{G_n})^T$ (für einen Roboter mit n Freiheitsgraden) geschieht unter Verwendung des statischen Kraft- und Momentenmodells. Nachstehend erfolgt eine kurze Zusammenfassung (Rückblick, vgl. Kapitel 7).

Gravitationskompensation – Statistisches Kraft- und Momentenmodell

Aus dem Impuls- und Drallsatz für die einzelnen Glieder lassen sich rekursiv vom Endeffektor zur Basis die notwendigen Aktorkräfte berechnen, um das System im Gleichgewicht zu halten. Die hierfür notwendigen Größen sind Bild 10.7 zu entnehmen.

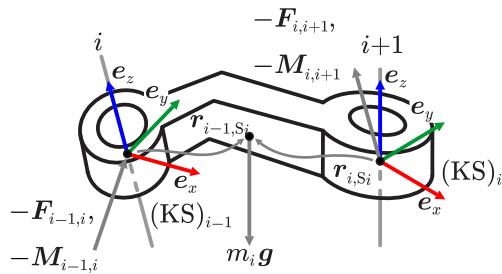


Bild 10.7: Am i -ten Roboterglied angreifende Kräfte und Momente

Die bereits bekannten Formeln zur Berechnung der Kräfte und Momente im $(KS)_{i-1}$ lauten (rein statische Betrachtung, d. h. Geschwindigkeiten und Beschleunigungen sind zu null gesetzt):

$${}_{(i-1)}\mathbf{F}_{i-1, i} = {}_{(i-1)}\mathbf{F}_{i, i+1} - m_i {}_{(i-1)}\mathbf{g}, \quad (10.8)$$

$$\begin{aligned} {}_{(i-1)}\mathbf{M}_{i-1, i} = & {}_{(i-1)}\mathbf{M}_{i, i+1} + {}_{(i-1)}\mathbf{r}_{S_i} \times {}_{(i-1)}\mathbf{F}_{i-1, i} \\ & + {}_{(i-1)}\mathbf{r}_{i, S_i} \times (-{}_{(i-1)}\mathbf{F}_{i, i+1}), \end{aligned} \quad (10.9)$$

wobei $(i-1)\mathbf{r}_{S_i}$, $(i-1)\mathbf{r}_{i,S_i}$ und $(i-1)\mathbf{g}$ häufig nicht im $(KS)_{i-1}$ gegeben sind, so dass eine Umrechnung mittels Basiswechsel erforderlich ist. So kann beispielsweise die in $(KS)_0$ gegebene Gravitation $(0)\mathbf{g}$ unter Verwendung von ${}^{i-1}\mathbf{R}_0$ (vgl. Kapitel 2) in dem gewünschten Koordinatensystem dargestellt werden.

Die zur Berechnung von Gleichung (10.8) und Gleichung (10.9) erforderlichen Kräfte und Momente, ausgeübt von Gelenk i auf das Gelenk $i+1$, sind in $(KS)_{i-1}$ anzugeben:

$$(i-1)\mathbf{F}_{i,i+1} = {}^{i-1}\mathbf{R}_i(i)\mathbf{F}_{i,i+1}, \quad (10.10)$$

$$(i-1)\mathbf{M}_{i,i+1} = {}^{i-1}\mathbf{R}_i(i)\mathbf{M}_{i,i+1}. \quad (10.11)$$

Es existieren zwei Sonderfälle:

- Reaktionen auf die Roboterbasis ($i=1$),
- Wechselwirkung des Endeffektors mit der Umwelt ($i=n$).

Sind die Kräfte $(i-1)\mathbf{F}_{i-1,i}$ sowie die Momente $(i-1)\mathbf{M}_{i-1,i}$ bekannt, kann der Vektor der Aktorkräfte und -momente zur Gravitationskompensation τ_G bestimmt werden:

- Linearaktorkräfte:

$$\tau_{G_i} = \underbrace{\left((i-1)\mathbf{e}_z^{(i-1)} \right)^T}_{(0,0,1)} (i-1)\mathbf{F}_{i-1,i}, \quad (10.12)$$

- Drehaktormomente:

$$\tau_{G_i} = \underbrace{\left((i-1)\mathbf{e}_z^{(i-1)} \right)^T}_{(0,0,1)} (i-1)\mathbf{M}_{i-1,i}. \quad (10.13)$$

Voraussetzung ist hierbei die Festlegung der Koordinatensysteme nach Denavit-Hartenberg. Der Freiheitsgrad q_i fällt mit der Gelenkkachse $i-1$ zusammen.

Das Skalarprodukt repräsentiert in diesem Fall eine Projektion der Kraft auf den Schubfreiheitsgrad bzw. des Moments auf den Rotationsfreiheitsgrad.

Aktive Steifigkeitsregelung inklusive Gravitationskompensation

Unter Verwendung der Gleichung (10.12) und Gleichung (10.13) kann die Vorsteuerung für die Gravitationskompensation der aktiven Steifigkeitsregelung leicht berechnet werden (vgl. Gleichung (10.7)):

$$\boldsymbol{\tau}_{\text{ges}}(t, \mathbf{q}) = \mathbf{J}^T(\mathbf{q}) \mathbf{K}_E \mathbf{J}(\mathbf{q}) \Delta \mathbf{q} + \boldsymbol{\tau}_G(\mathbf{q}). \quad (10.14)$$

Die Berücksichtigung der Gravitationskompensation führt auf das Blockschaltbild nach Bild 10.8. Eine zusätzliche Berücksichtigung von Reibeinflüssen und dynamischen nichtlinearen Verkoppungen ist ebenfalls durch einfache Addition der entsprechenden Terme möglich (vgl. Kapitel 7).

Anmerkung

Im Folgenden wird die Abhängigkeit des Soll- sowie des Gesamtmotormoments ($\boldsymbol{\tau}_s$ bzw. $\boldsymbol{\tau}_{\text{ges}}$) von den Gelenkwinkeln \mathbf{q} aus Übersichtlichkeitsgründen nicht weiter explizit angegeben.

Nachdem das Prinzip der Gravitationskompensation eingeführt wurde, wird nun die Modellierung des Kontakts von Roboter und Umgebung beschrieben.

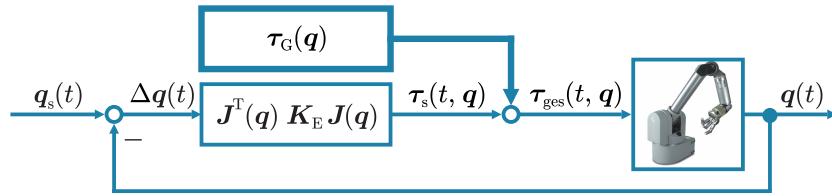


Bild 10.8: Blockschaltbild einer aktiven Steifigkeitsregelung im Jointspace inkl. Gravitationskompensation

10.3.3 Aktive Steifigkeitsregelung im Kontaktfall

Im Kontaktfall resultieren in Abhängigkeit von der Endeffektorlage x_E und der „Lage“ der Umgebung x_U die Kontaktkräfte \mathcal{F}_U (vgl. Bild 10.9). Diese Kontaktkräfte haben den im Blockschaibild

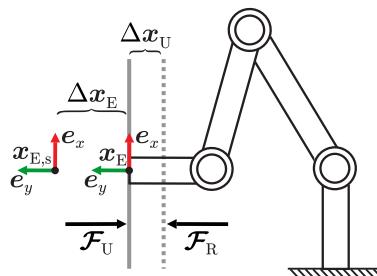


Bild 10.9: Bleibende Regelabweichung im Kontaktfall

(Bild 10.10) gezeigten Einfluss auf die Kräfte bzw. Momente der Antriebe. Das hervorgehobene physikalische System wirkt demnach im Kontaktfall den Aktorkräften des Roboters entgegen. Die Umgebung wird hierbei beispielweise als mechanisches System zweiter Ordnung modelliert:

$$\mathcal{F}_U = \mathbf{K}_U \Delta x_U + \mathbf{D}_U \Delta \dot{x}_U + \mathbf{M}_U \Delta \ddot{x}_U. \quad (10.15)$$

Die Kontaktkräfte (Reaktionskräfte) sind stark von der Roboterdynamik sowie der Umgebung bzw. den Umgebungsparametern (insbesondere \mathbf{K}_U) abhängig. Im statischen Fall gilt das Kräfte-/Momentengleichgewicht:

$$\mathcal{F}_U = \mathcal{F}_R \quad \text{bzw.} \quad \mathbf{K}_U \Delta x_U = \mathbf{K}_E \Delta x_E. \quad (10.16)$$

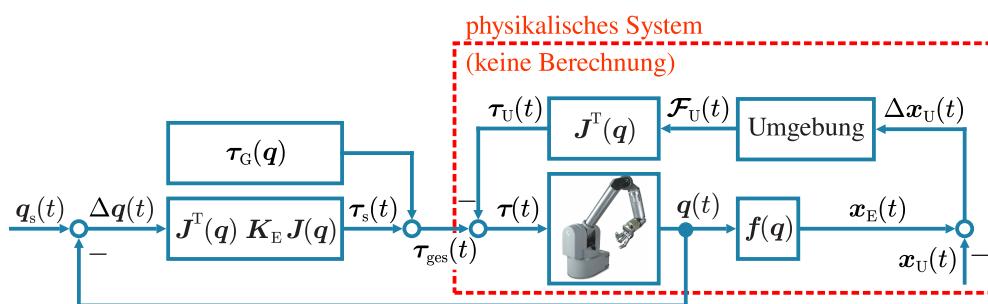


Bild 10.10: Blockschaltbild einer aktiven Steifigkeitsregelung im Jointspace im Kontaktfall

Berücksichtigung der Getriebeübersetzung

In den vorstehenden Abschnitten wurde die Getriebeübersetzung in den Gelenken u_{G_i} zu eins gesetzt. Dies führt dazu, dass die Gelenke rücktriebbar sind. Allerdings ist dies bei lediglich wenigen Robotern der Fall (vgl. Abschnitt 10.3.1). Der Großteil der (industriell) eingesetzten Roboter verbaut jedoch (hochübersetzende) Getriebe (z. B. mit $u_{G_i} > 50$), die im Wesentlichen aufgrund der Reibung nicht rücktriebbar sind.

Für den in Bild 10.9 bzw. Bild 10.10 betrachteten Kontaktfall folgt:

$$\tau_U = J^T \mathcal{F}_U. \quad (10.17)$$

Das aus der Kontaktsituation resultierende Motormoment ergibt sich somit zu:

$$\tau_M = \text{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) \tau = \text{diag}\left(\frac{1}{u_{G_1}}, \dots, \frac{1}{u_{G_n}}\right) J^T \mathcal{F}_U. \quad (10.18)$$

Aufgrund der im Allgemeinen nicht (oder nur sehr schwer) rücktriebbaren Getriebe liegt das Drehmoment τ_M jedoch nicht am Rotor des Motors (antriebsseitig) an – es erfolgt keine detektierbare Rückwirkung. Folglich ist eine sensorielle Erfassung der Umgebungs Kräfte, beispielsweise mittels gelenkintegrierter abtriebsseitiger Drehmomentsensoren, notwendig. So kann ein Drehmomentengelkreis auf Gelenkebene geschlossen werden und es resultiert eine geregelte Rücktriebbarkeit¹. Nachstehende Bild 10.11 zeigt schematisch den mittels gelenkintegrierter Drehmomentsensoren geschlossenen Regelkreis.

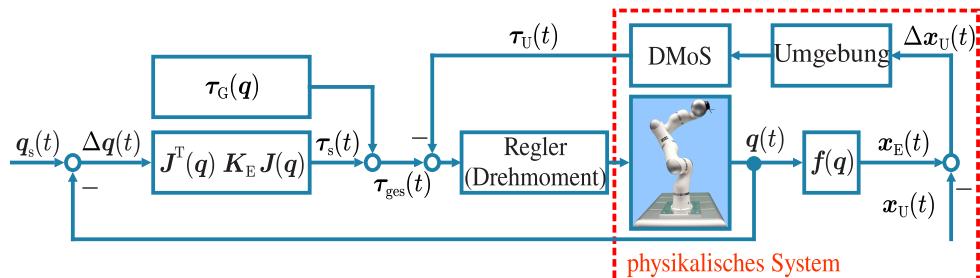


Bild 10.11: Aktive Steifigkeitsregelung mit Drehmomentregelkreis auf Gelenkebene unter Verwendung abtriebsseitig integrierter Drehmomentsensoren (DMoS)

Anmerkung

Die in Bild 10.11 dargestellte Gravitationskompensation ist in diesem Fall zwingend erforderlich, da die DMoS zusätzlich zur externen Kraft auch das Eigengewicht des Roboters erfassen.

Zu beachten ist, dass für kleine Regelfehler, z. B. unter Verwendung eines klassischen PD-Drehmomentreglers (vgl. Kapitel 8), lediglich geringe Solldrehmomente τ_s und folglich kleine Sollströme resultieren. Aufgrund der Gelenkreibung erfolgt nicht die gewünschte Endeffektorbewegung: Sind die Solldrehmomente betragsmäßig kleiner als die Reibung, so bleiben die jeweiligen Antriebe zeitweise „hängen“ und es resultiert ein „stick-slip“ ähnliches Verhalten. Dieser Effekt lässt sich durch geeignete Vorsteuerung der Gelenkreibung eliminieren (vgl. Abschnitt 7.4.2).

Alternativ kann die Erfassung der Umgebungs Kräfte über einen am Endeffektor montierten Kraft-Moment-Sensor (vgl. Abschnitt 10.4) erfolgen. Hierbei werden die externen Kräfte/Momente \mathcal{F} über die transponierte Jacobi-Matrix $J^T(q)$ in den Gelenkraum projiziert. Der entsprechende Regelkreis ist in Bild 10.12 dargestellt. Nachteil dieses Ansatzes ist, dass Kräfte und/oder Momente, die an der Struktur angreifen, unbeachtet bleiben, da diese sensoriell nicht erfasst werden.

Anmerkung

Im Gegensatz zu einer Integration abtriebsseitiger Drehmomentsensoren in die Roboterstruktur

¹Ein Beispiel für ein solches System ist der Leichtbauroboter (LBR) der KUKA Laboartories GmbH (vgl. Bild 10.11).

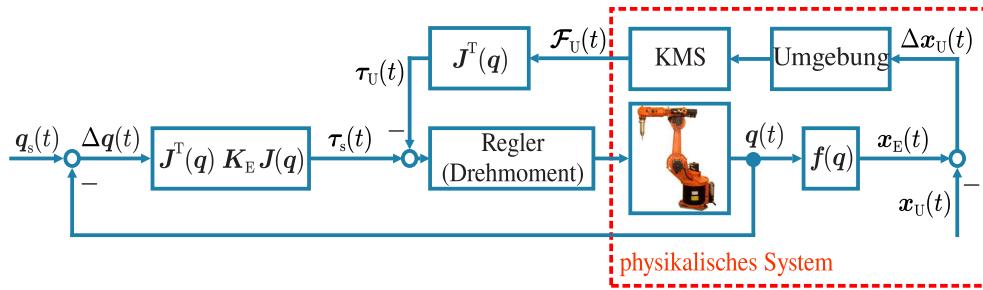


Bild 10.12: Aktive Steifigkeitsregelung mit Drehmomentregelkreis auf Gelenkebene und am Endeffektor montierten Kraft-Moment-Sensor (KMS)

ist eine explizite Gravitationskompensation unter Verwendung eines am Endeffektor montierten Kraft-/Momentsensor nicht zwingend erforderlich. Aufgrund reibungsbehafteter hochübersetzender Getriebe (nicht rücktreibbar) ist der Einfluss des Eigengewichts des Roboters auf die motorseitigen Antriebsmomente in der Regel vernachlässigbar.

Aktive Steifigkeitsregelung im Taskspace

Eine aktive Steifigkeitsregelung ist, ähnlich wie im oben beschriebenen Jointspace, auch im Taskspace zu realisieren. In diesem Fall stellt sich das Blockschaltbild des Regelkreises wie in Bild 10.13 dar. Wie bereits erwähnt, ist eine Gravitationskompensation (siehe Abschnitt 10.3.2) lediglich unter Verwendung abtriebsseitiger Drehmomentsensoren oder rücktreibbarer Antriebe erforderlich. Der Drehmomentregler ist der Übersichtlichkeit halber in den folgenden Abbildungen

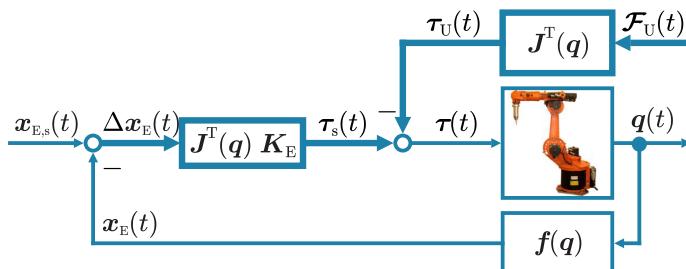


Bild 10.13: Blockschaltbild einer aktiven Steifigkeitsregelung im Taskspace mit Umgebungskontakt

nicht mehr explizit dargestellt.

Das Sollmotormoment berechnet sich ausgehend von der kartesischen Lageabweichung Δx_E zu:

$$\tau_s = \mathbf{J}^T(\mathbf{q}) \mathbf{K}_E \underbrace{\mathbf{J}(\mathbf{q}) \Delta \mathbf{q}}_{\Delta x_E} = \mathbf{J}^T(\mathbf{q}) \mathbf{K}_E \Delta x_E. \quad (10.19)$$

Die Merkmale der aktiven Steifigkeitsregelung (sowohl im Joint- als auch im Taskspace) lauten:

- Einfache Implementierung und für ungeregelt rücktreibbare Antriebe **in der Theorie** keine zusätzliche Sensorik erforderlich, vgl. Abschnitt 10.3.1,
- Online-Änderung der gewünschten Steifigkeiten problemlos möglich,
- in dieser Form keine Vorgabe einer gewünschten Dynamik möglich (nur Steifigkeit),
- bei kleinen Auslenkungen und/oder geringen Steifigkeiten (d. h. kleine Kräfte \mathcal{F}_R und somit geringe Motormomente τ_s) häufig bleibender Regel-/Lagefehler aufgrund unzureichender Modelle (z. B. Reibung in den Gelenken).

10.4 Aktive Steifigkeitsregelung inklusive Rückführung

Die aktuellen Kontaktkräfte (Kräfte/Momente) können mittels Kraft-Moment-Sensor (KMS) gemessen und zurückgeführt werden (Funktionsweise des Sensors vgl. Kapitel 11). Diese zusätzliche Information kann anschließend beispielsweise zur Steigerung der Sicherheit und/oder zum Schließen des zusätzlichen Drehmomentregelkreis auf Gelenkebene (vgl. Abschnitt 10.3.3) verwendet werden. Die gemessenen Kontaktkräfte $\hat{\mathcal{F}}_U$ ergeben sich nach Bild 10.14. Im Folgenden sei angenommen, dass die gemessenen Kontaktkräfte $\hat{\mathcal{F}}_U$ den tatsächlichen Kontaktkräften \mathcal{F}_U entsprechen ($\hat{\mathcal{F}}_U \approx \mathcal{F}_U$) und dass die mechanische Steifigkeit des KMS signifikant größer als die der Umgebung ist.

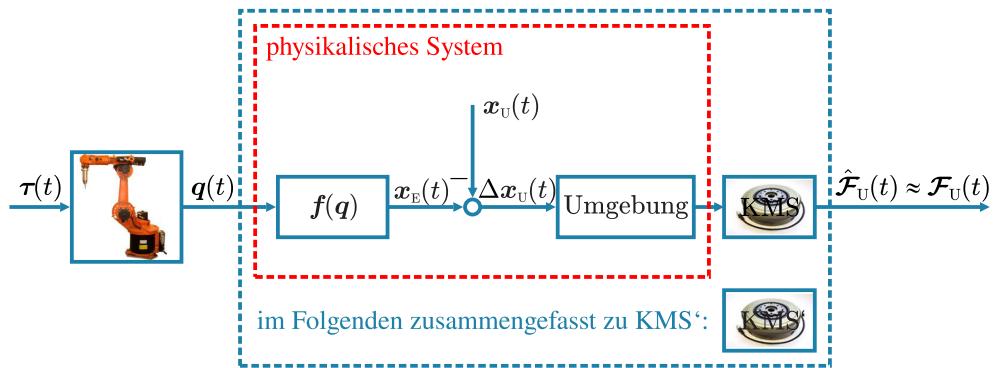


Bild 10.14: Messung der Kontaktkräfte/-momente durch Kraft-Moment-Sensor

Die Kontaktkräfte werden durch die Nachgiebigkeitsmatrix

$$C_F = \text{diag}(c_F) = \text{diag}\left(\frac{1}{k_{F_1}}, \dots, \frac{1}{k_{F_m}}\right) \quad (10.20)$$

in eine Lageänderung $\Delta x_{E,C}$ umgerechnet, um die Solllage $x_{E,s}$ in Abhängigkeit der gemessenen Kräfte zu modifizieren (siehe Bild 10.15).

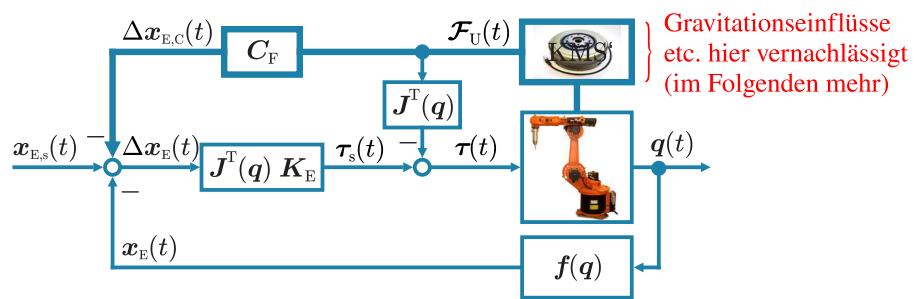


Bild 10.15: Blockschaltbild einer aktiven Steifigkeitsregelung im Taskspace inkl. Kraftrückführung

Die Anpassung der Solllage in Abhängigkeit der gemessenen Kontaktkräfte führt zu einer Steigerung der Sicherheit. Der Regelkreis verfügt über eine Möglichkeit zur schnellen Reaktion auf unerwartet hohe und damit gefährliche Kontaktkräfte.

Anmerkung

Oft ist der Kraft-Moment-Sensor zwischen dem letzten Glied des Roboters und dem Handhabungsobjekt angebracht. Dies führt, wie nun dargestellt, zu einer Verfälschung der gemessenen Kontaktkraft und bedarf einer Gravitationskompensation.

Eigengewichtskompensation des Handhabungsobjekts

Die im Sensorkoordinatensystem $(KS)_F$ gemessenen Kräfte und Momente ${}_{(F)}\mathcal{F}_{\text{ges}}$ enthalten neben den Kontaktkräften ${}_{(F)}\mathcal{F}_U$ einen zusätzlichen Anteil ${}_{(F)}\mathcal{F}_H$, resultierend aus der am Kraft-Moment-Sensor montierten Masse m_H (Handhabungsobjekt), vgl. Bild 10.16. Dieser Anteil

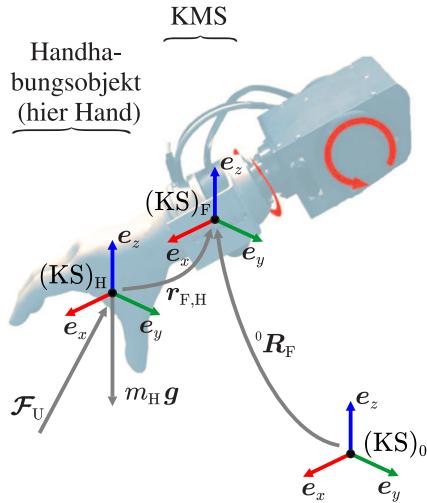


Bild 10.16: Darstellung der am Sensor angreifenden Kräfte und Momente

setzt sich aus einem statischen sowie dynamischen Teil zusammen, wobei der dynamische Teil im Folgenden vernachlässigt wird (hinreichend langsame Bewegung). Der statische Anteil ist von der Lage des Endeffektors und damit von den Gelenkwinkeln \mathbf{q} abhängig. Um die reinen Kontaktkräfte ${}_{(F)}\mathcal{F}_U$ zu erhalten, müssen die Gewichtskraft des Handhabungsobjekts $m_H \mathbf{g}$ sowie das Moment resultierend aus der Gewichtskraft $r_{F,H} \times m_H \mathbf{g}$ kompensiert werden:

$${}_{(F)}\mathcal{F}_U = {}_{(F)}\mathcal{F}_{\text{ges}} - {}_{(F)}\mathcal{F}_H = {}_{(F)}\mathcal{F}_{\text{ges}} - \begin{pmatrix} m_H {}_{(F)}\mathbf{g} \\ {}_{(F)}\mathbf{r}_H \times m_H {}_{(F)}\mathbf{g} \end{pmatrix}. \quad (10.21)$$

Die Gravitationsvektoren \mathbf{g} sowie $\mathbf{r}_{F,H}$ sind häufig nicht im Sensorkoordinatensystem $(KS)_F$ gegeben. Eine Umrechnung mittels Rotationsmatrix ist nötig (vgl. Kapitel 2) und ergibt beispielsweise (unter der Annahme, dass die Größen im Inertialkoordinatensystem $(KS)_0$ gegeben sind):

$$\begin{aligned} {}_{(F)}\mathcal{F}_U &= {}_{(F)}\mathcal{F}_{\text{ges}} - {}_{(F)}\mathcal{F}_H \\ &= {}_{(F)}\mathcal{F}_{\text{ges}} - \begin{pmatrix} m_H {}^0\mathbf{R}_F^T(\mathbf{q}) {}^{(0)}\mathbf{g} \\ {}^0\mathbf{R}_F^T(\mathbf{q}) {}^{(0)}\mathbf{r}_{F,H} \times m_H {}^0\mathbf{R}_F^T(\mathbf{q}) {}^{(0)}\mathbf{g} \end{pmatrix}. \end{aligned} \quad (10.22)$$

Anmerkungen:

- Masse m_H sowie Hebelarm $\mathbf{r}_{F,H}$ sind oft aus CAD-Daten bekannt (alternativ auch Kalibrierung möglich),
- Messdaten sind im Sensorkoordinatensystem $(KS)_F$ gegeben, für die Regelung ist eine Umrechnung in das Inertialkoordinatensystem $(KS)_0$ nötig.

Die Umrechnung der Kontaktkräfte in das Basiskoordinatensystem $(KS)_0$ ist problemlos unter Verwendung der entsprechenden Rotationsmatrix vom $(KS)_F$ in $(KS)_0$ möglich:

$$(0)\mathcal{F}_U = \begin{pmatrix} {}^0\mathbf{R}_F(q) & \mathbf{0} \\ \mathbf{0} & {}^0\mathbf{R}_F(q) \end{pmatrix} (F)\mathcal{F}_U = {}^0\mathbf{R}_F^*(q) (F)\mathcal{F}_U. \quad (10.23)$$

Im Folgenden werden aus Gründen der Übersicht der Kraft-Moment-Sensor, die Eigengewichtskompensation des Handhabungsobjekts sowie die Umrechnung in $(KS)_0$ zusammengefasst (vgl. Bild 10.17). Ebenso wird der Index (0) weggelassen. Die Kraft ist im $(KS)_0$ gegeben.

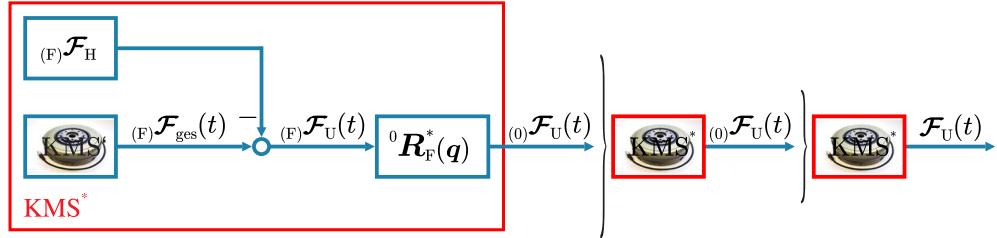


Bild 10.17: Kompakte Darstellung der Erfassung der Kontaktkräfte und -momente

Aktive Steifigkeitsregelung inklusive Kraftrückführung – vollständiger Regelkreis

Unter Berücksichtigung der Eigengewichtskompensation des Handhabungsobjekts resultiert das Blockschaltbild der aktiven Steifigkeitsregelung mit Kraftrückführung nach Bild 10.18.

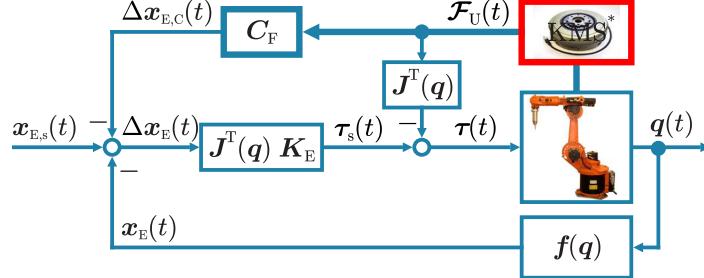


Bild 10.18: Blockschaltbild einer aktiven Steifigkeitsregelung inklusive Kraftrückführung im Taskspace

Der Regelkreis der aktiven Steifigkeitsregelung inklusive Kraftrückführung ist jetzt vollständig. Er beinhaltet:

- Rückwirkung im Kontaktfall (Umgebung),
- Eigengewichtskompensation des Handhabungsobjekts,
- Sensor kann zum Schließen der Drehmomentkaskade verwendet werden (über transponierte Jacobi-Matrix).

10.5 Impedanzregelung

Die Impedanzregelung ist eine Erweiterung der bekannten Steifigkeitsregelung. Bei der Impedanzregelung wird ein gewünschtes dynamisches Verhalten des Roboters vorgegeben (vgl. Bild 10.19) und nicht nur eine Sollsteifigkeit wie bei den bisherigen Betrachtungen.

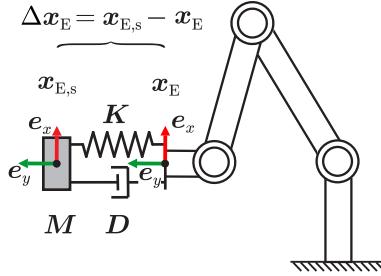


Bild 10.19: Roboter als einfaches Masse/Feder/Dämpfer-System

Der Roboter soll sich dabei gegenüber der Umgebung (also bzgl. einer Umgebungskraft) wie eine gewünschte mechanische Impedanz bestehend aus Massenmatrix M , Dämpfungsmatrix D und Steifigkeitsmatrix K – also wie ein Masse/Feder/Dämpfer-System zweiter Ordnung – verhalten. Somit muss für den geregelten Roboter mit $\Delta x_E = x_{E,s} - x_E$

$$M \Delta \ddot{x}_E(t) + D \Delta \dot{x}_E(t) + K \Delta x_E(t) = \mathcal{F}_U(t) \quad (10.24)$$

gelten. Für die Kraft im Frequenzbereich gilt analog

$$M \Delta X_E(s) s^2 + D \Delta X_E(s) s + K \Delta X_E(s) = \mathcal{F}_U(s). \quad (10.25)$$

Da obige Differentialgleichung (10.24) in Endeffektorkoordinaten erfüllt sein soll, wird zunächst das dynamische Modell des Roboters im Taskspace hergeleitet.

Roboterdynamik im Taskspace

Die Dynamik des Roboters im Jointspace liefert den Vektor τ der Gelenkmomente, der mit Hilfe der mit $\tau = J^T(q)\mathcal{F}_\tau \Leftrightarrow \mathcal{F}_\tau = J^{-T}(q)\tau$ in den verallgemeinerten Kraftvektor \mathcal{F}_τ am Endeffektor umgerechnet wird:

$$\tau = M(q)\ddot{q} + c(q,\dot{q}) + g(q) \quad (10.26)$$

$$J^{-T}(q)\tau = J^{-T}(q)M(q)\ddot{q} + J^{-T}(q)c(q,\dot{q}) + J^{-T}(q)g(q) \quad (10.27)$$

$$\mathcal{F}_\tau = J^{-T}(q)M(q)\ddot{q} + J^{-T}(q)c(q,\dot{q}) + J^{-T}(q)g(q) \quad (10.28)$$

Die Substitution von \ddot{q} durch einen entsprechenden Ausdruck in \ddot{x} erfolgt durch Ableitung der differenziellen Kinematik $\dot{x}_E = J(q)\dot{q}$:

$$\ddot{x}_E = \dot{J}(q)\dot{q} + J(q)\ddot{q} \quad (10.29)$$

$$\Leftrightarrow \ddot{q} = J^{-1}(q)\ddot{x}_E - J^{-1}(q)\dot{J}(q)\dot{q} \quad (10.30)$$

Durch Ersetzen von $\ddot{\mathbf{q}}$ folgt im Kartesischen Raum:

$$\mathcal{F}_\tau = \mathbf{J}^{-T}(\mathbf{q}) \mathbf{M}(\mathbf{q}) (\mathbf{J}^{-1}(\mathbf{q}) \ddot{\mathbf{x}}_E - \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{J}}(\mathbf{q}) \dot{\mathbf{q}}) + \mathbf{J}^{-T}(\mathbf{q}) \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}^{-T}(\mathbf{q}) \mathbf{g}(\mathbf{q}) \quad (10.31)$$

$$\begin{aligned} \mathcal{F}_\tau &= \underbrace{\mathbf{J}^{-T}(\mathbf{q}) \mathbf{M}(\mathbf{q}) \mathbf{J}^{-1}(\mathbf{q})}_{\mathbf{M}_K(\mathbf{q})} \ddot{\mathbf{x}}_E \\ &\quad + \underbrace{\mathbf{J}^{-T}(\mathbf{q}) (\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q}) \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{J}}(\mathbf{q}) \dot{\mathbf{q}})}_{\mathbf{c}_K(\mathbf{q}, \dot{\mathbf{q}})} + \underbrace{\mathbf{J}^{-T}(\mathbf{q}) \mathbf{g}(\mathbf{q})}_{\mathbf{g}_K(\mathbf{q}) = \mathcal{F}_g(\mathbf{q})} \end{aligned} \quad (10.32)$$

$$\mathcal{F}_\tau = \mathbf{M}_K(\mathbf{q}) \ddot{\mathbf{x}}_E + \mathbf{c}_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q}) \quad (10.33)$$

Der resultierende Zusammenhang zwischen der Dynamik im Gelenk- und Arbeitsraum lautet dann

$$\mathbf{M}_K(\mathbf{q}) = \mathbf{J}^{-T}(\mathbf{q}) \mathbf{M}(\mathbf{q}) \mathbf{J}^{-1}(\mathbf{q}), \quad (10.34)$$

$$\mathbf{c}_K(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}^{-T}(\mathbf{q}) (\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q}) \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{J}}(\mathbf{q}) \dot{\mathbf{q}}), \quad (10.35)$$

$$\mathcal{F}_g(\mathbf{q}) = \mathbf{J}^{-T}(\mathbf{q}) \mathbf{g}(\mathbf{q}) \quad (10.36)$$

und die Gelenkmomente, die für eine bestimmte Bewegung im Arbeitsraum erforderlich sind, lauten

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q}) (\mathbf{M}_K(\mathbf{q}) \ddot{\mathbf{x}}_E + \mathbf{c}_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q})) \quad (10.37)$$

Zusammenfassend kann man also folgendes für die Roboterdynamik im Taskspace angeben:

$$\mathbf{M}_K(\mathbf{q}) \ddot{\mathbf{x}}_E + \mathbf{c}_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q}) = \mathcal{F}_\tau \quad (10.38)$$

mit

$$\ddot{\mathbf{x}}_E = \mathbf{J}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}) \dot{\mathbf{q}} \quad (10.39)$$

$$\mathcal{F}_\tau = \mathbf{J}^{-T}(\mathbf{q}) \boldsymbol{\tau} \quad (10.40)$$

$$\mathcal{F}_g(\mathbf{q}) = \mathbf{J}^{-T}(\mathbf{q}) \mathbf{g}(\mathbf{q}) \quad (10.41)$$

$$\mathbf{M}_K(\mathbf{q}) = \mathbf{J}^{-T}(\mathbf{q}) \mathbf{M}(\mathbf{q}) \mathbf{J}^{-1}(\mathbf{q}) \quad (10.42)$$

$$\mathbf{c}_K(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}^{-T}(\mathbf{q}) \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}_K(\mathbf{q}) \dot{\mathbf{J}}(\mathbf{q}) \dot{\mathbf{q}} \quad (10.43)$$

Der Ansatz eines klassischen Impedanzreglers bei Auftreten externer Kräfte erfolgt dann wie nachfolgend beschrieben.

Impedanzregler im Taskspace

Das dynamische Modell des Roboters im Taskspace ($SE3$ -Koordinaten \mathbf{x}) mit der Stellgröße \mathcal{F}_R des Reglers und der externen (Kontakt-)Kraft \mathcal{F}_U lautet

$$\mathbf{M}_K(\mathbf{q}) \ddot{\mathbf{x}}_E + \mathbf{c}_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q}) = \mathcal{F}_R - \mathcal{F}_U \quad (10.44)$$

Der auf diesem dynamischen Modell basierende Reglerentwurf beinhaltet dann die folgenden Schritte, die eine Ähnlichkeit zur Feedback-Regelung aufweisen, um die nichtlinearen Terme der Dynamik zu kompensieren:

1. Kompensation des nichtlinearen Modellteils $c_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q})$
2. Kompensation der externen Kraft \mathcal{F}_U , die auf die „falsche“ Impedanz trifft
3. Gewünschte Dynamik für die externe Kraft \mathcal{F}_U mit

$$\mathbf{M}_K(\mathbf{q})\ddot{\mathbf{x}}_{E,s} + \mathbf{M}_K(\mathbf{q})\mathbf{M}^{-1}(\mathbf{K}\Delta\mathbf{x}_E + \mathbf{D}\Delta\dot{\mathbf{x}}_E - \mathcal{F}_U)$$
 hinzufügen

Es ergibt sich dann das Regelgesetz $\Delta\mathbf{x}_E = \mathbf{x}_{E,s} - \mathbf{x}_E$ in der Form

$$\mathcal{F}_R = \mathbf{M}_K(\mathbf{q})\ddot{\mathbf{x}}_{E,s} + \mathbf{M}_K(\mathbf{q})\mathbf{M}^{-1}(\mathbf{K}\Delta\mathbf{x}_E + \mathbf{D}\Delta\dot{\mathbf{x}}_E - \mathcal{F}_U) + c_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q}) + \mathcal{F}_U \quad (10.45)$$

Die Dynamik des geschlossenen Regelkreises, also des geregelten Roboters ergibt sich durch Einsetzen des Regelgesetzes (10.45) in die Dynamik des Roboters im Taskspace:

$$\begin{aligned} \mathbf{M}_K(\mathbf{q})\ddot{\mathbf{x}}_E + c_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q}) &= \mathbf{M}_K(\mathbf{q})\ddot{\mathbf{x}}_{E,s} \\ &+ \mathbf{M}_K(\mathbf{q})\mathbf{M}^{-1}(\mathbf{K}\Delta\mathbf{x}_E + \mathbf{D}\Delta\dot{\mathbf{x}}_E - \mathcal{F}_U) + c_K(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{F}_g(\mathbf{q}) + \mathcal{F}_U - \mathcal{F}_U \end{aligned} \quad (10.46)$$

Nach einigen Umformungen folgt die Dynamik des Roboters in der Form

$$\mathbf{M}_K(\mathbf{q})\ddot{\mathbf{x}}_E = \mathbf{M}_K(\mathbf{q})\ddot{\mathbf{x}}_{E,s} + \mathbf{M}_K(\mathbf{q})\mathbf{M}^{-1}(\mathbf{K}\Delta\mathbf{x}_E + \mathbf{D}\Delta\dot{\mathbf{x}}_E - \mathcal{F}_U) \quad (10.47)$$

$$\mathbf{M}_K(\mathbf{q})\Delta\ddot{\mathbf{x}}_E + \mathbf{M}_K(\mathbf{q})\mathbf{M}^{-1}(\mathbf{K}\Delta\mathbf{x}_E + \mathbf{D}\Delta\dot{\mathbf{x}}_E) = \mathbf{M}_K(\mathbf{q})\mathbf{M}^{-1}\mathcal{F}_U \quad (10.48)$$

$$\mathbf{M}\Delta\ddot{\mathbf{x}}_E + \mathbf{D}\Delta\dot{\mathbf{x}}_E + \mathbf{K}\Delta\mathbf{x}_E = \mathcal{F}_U \quad (10.49)$$

Gleichung (10.49) entspricht der gewünschten Dynamik des geregelten Roboters. Dieser verhält sich somit gegenüber externen Kräften \mathcal{F}_U wie eine mechanische Impedanz mit den Parametern \mathbf{M} , \mathbf{D} und \mathbf{K} .

Aus der Stellgröße \mathcal{F}_R ist schließlich das zugehörige Soll-Drehmoment $\tau_s = \mathbf{J}^T(\mathbf{q})\mathcal{F}_R(t)$ zu bestimmen und mit Hilfe eines unterlagerten Drehmomentreglers auf den Roboter anzuwenden. Es ergibt sich dann das in Bild 10.20 dargestellte Blockschaltbild, in dem der unterlagerte Drehmomentregler zur besseren Übersicht nicht dargestellt ist. Es wird $\tau_m \approx \tau_s$ angenommen. Da der Drehmomentregler im Wesentlichen ein sehr schneller Regler der Motorströme in den Gelenken ist, stellt diese Annahme keine wesentliche Einschränkung dar.

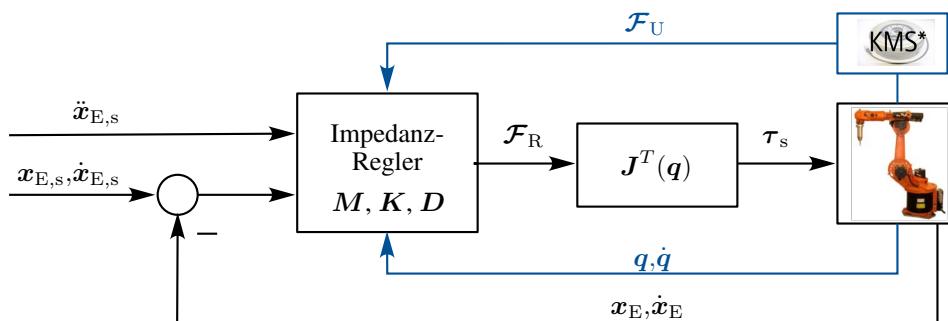


Bild 10.20: Blockschaltbild einer Impedanzregelung (der unterlagerte Drehmomentregler ist nicht dargestellt)

Setzt man gedanklich die virtuelle Massenträgheit \mathbf{M} und die virtuelle Dämpfung \mathbf{D} zu null, erkennt man, dass die Impedanzregelung in die Steifigkeitsregelung übergeht.

Eigenschaften der Impedanzregelung

Aus den vorherigen Ausführungen lassen sich zusammenfassend folgende Eigenschaften der Impedanzregelung angeben:

- Die Impedanzregelung ist eine Kaskadenregelung: Im äußereren Kreis wird die Position, im inneren Kreis das Drehmoment geregelt. Es handelt sich daher um eine sogenannte „Force based impedance control“
- Für die Berechnung von \mathcal{F}_R wird nur $\ddot{x}_{E,s}$, nicht aber \ddot{x}_E benötigt, dessen Bestimmung häufig problematisch ist.
- Einhalten einer gewünschten Kontaktkraft
 - Die Sollposition $x_{E,s}$ liegt im Inneren oder hinter einem Objekt, wenn eine bestimmte Kontaktkraft zwischen Roboter und Objekt gewünscht ist. Im stationären Zustand ergibt sich die Kontaktkraft in Abhängigkeit der bleibende Regeldifferenz und der Steifigkeit zu $\mathcal{F}_U = \mathbf{K} \Delta x_E$.
 - Daraus ergibt sich der Vorteil gegenüber einer reinen Kraftregelung, dass bei einem Kontaktverlust der Roboter lediglich zur Sollposition fährt und nicht dauerhaft mit der Sollkraft beschleunigt.
- Bei unveränderter Trägheit, also $M = M_K$ verschwindet \mathcal{F}_U im Regelgesetz; es ist dann kein Kraftsensor erforderlich.
- Wird die externe Kraft \mathcal{F}_U nicht im Regelgesetz berücksichtigt, ergibt sich für den geregelten Roboter nicht die gewünschte Impedanz gemäß Gleichung (10.49). Im Einzelnen folgt:

Ohne \mathcal{F}_U : Die rechte Seite von Gleichung (10.49) wäre $\mathbf{0}$ und somit $\Delta x_E(t \rightarrow \infty) = \mathbf{0}$. Dieses Verhalten ist bei Kontakt unerwünscht und entspricht bei $M = E$ der Feedback-Regelung.

Ohne \mathcal{F}_U : Auf der rechten Seite von Gleichung (10.49) ergäbe sich $(E + MM_K^{-1})\mathcal{F}_U$ und somit ein falsche Impedanz.

Ohne \mathcal{F}_U und \mathcal{F}_U : Die rechte Seite von Gleichung (10.49) wäre $MM_K^{-1}\mathcal{F}_U$, was ebenfalls zu einer falschen Impedanz führen würde.

Einstellempfehlungen

Bei Umgebungen mit erwartetem Kontakt lassen sich folgende Ziele angeben:

- Vermeidung hoher Kräfte bei unsicheren Abmaßen der Umgebung
- Komplementäre Anpassung der dynamischen Eigenschaften (insb. Steifigkeit) an die Umgebung
- Beispiel: Nachbildung des menschlichen Arms:
 - steif und schnell in freier Bewegung,
 - nachgiebig und langsam bei kritischen Bewegungen in Kontakt

Für die Diagonalelemente der diagonalen Matrizen der Impedanzregelung können dann folgende „Einstellregeln“ formuliert werden:

- M_i : Hohe Werte in Kartesischen Richtungen mit vermutetem Kontakt zur Begrenzung der Dynamik des Roboters. Daraus resultiert eine Kraft entgegen der Bewegungsrichtung und somit kleinere Kontaktkräfte. Kleine Werte in Richtungen ohne erwarteten Kontakt, um ein möglichst gutes (schnelles) Tracken der Solltrajektorie zu ermöglichen.

- K_i : Niedrige Werte in Kartesischen Richtungen mit vermutetem Kontakt (hohe Nachgiebigkeit) und hohe Werte in Richtungen ohne Kontakt, um die Genauigkeit der Positionsregelung zu erhöhen (gutes Tracking der gewünschten Trajektorie).
- D_i : Die Dämpfung (Dissipation von kinetischer Energie) dient der Einstellung des gewünschten transienten Verhaltens.

Zusammenfassung Impedanzregler

Die wesentlichen Merkmale der Impedanzregelung sind:

- Festlegung einer gewünschten Solldynamik des Roboters durch Vorgabe von M , D und K ,
- Vorgabe einer gewünschten Reaktion auf Kontaktkräfte möglich,
- keine direkte Vorgabe von Sollkontaktkräften,
- setzt ausreichende Kenntnis über die Umgebung voraus (zur geeigneten Wahl von M , D , und K),
- zusätzliche Sensorik erforderlich (nur bei Kraftrückführung oder beim Schließen des Drehmomentregelkreises auf Gelenkebene).

Alternativ zur Impedanzregelung ist auch das Verfahren der Admittanzregelung gebräuchlich (vgl. Abschnitt 10.6).

10.6 Direkte Kraftregelung

Nachfolgend werden zwei wichtige Fälle der direkten Kraftregelung beschrieben. Die Admittanzregelung (Abschnitt 10.6.1), die neben der Sollkraft auch die Lage des Roboters berücksichtigen kann und die kraftbasierte direkte Lageregelung (Abschnitt 10.6.2), welche nur auf der Vorgabe von Sollkräften aufbaut.

10.6.1 Lagebasierte direkte Kraftregelung (Admittanzregelung)

Bisher wurden Regelungsverfahren vorgestellt, die keine direkte Vorgabe einer Soll-Kontaktkraft \mathcal{F}_s beinhalten. Bei der lagebasierten direkten Kraftregelung (Admittanzregelung) erfolgt eine explizite Vorgabe der Soll-Kontaktkraft \mathcal{F}_s (im Gegensatz zur Impedanzregelung, bei der die Kontaktkraft \mathcal{F}_U indirekt über die Solllage $x_{E,s}$ vorgegeben wird). Diese führt zu einer Lage- bzw. Geschwindigkeitsänderung des Endeffektors. Basis ist die mechanische Admittanz, also der Kehrwert der mechanischen Impedanz.

Admittanzregelung ohne Sollwertvorgabe der Position

Wie bei der Impedanzregelung handelt es sich bei der Admittanzregelung um eine Kaskadenregelung: Im äußeren Kreis wird jetzt aber die Kraft und im inneren die Position geregelt. Es handelt sich um eine sogenannte „Position based impedance control“. Der eigentliche Admittanzregler hat daher Kräfte als Ein- und Positionsabweichungen bzw. Geschwindigkeiten als Ausgangsgrößen. Daraus ergibt sich das in Bild 10.21 dargestellte Blockschaltbild, in dem der unterlagerte Positionsregler für q_s ist nicht dargestellt ist.

Das Modell der gewünschten mechanischen Impedanz bleibt unverändert

$$M \Delta \ddot{x}_E + D \Delta \dot{x}_E + K \Delta x_E = \mathcal{F}_s, \quad (10.50)$$

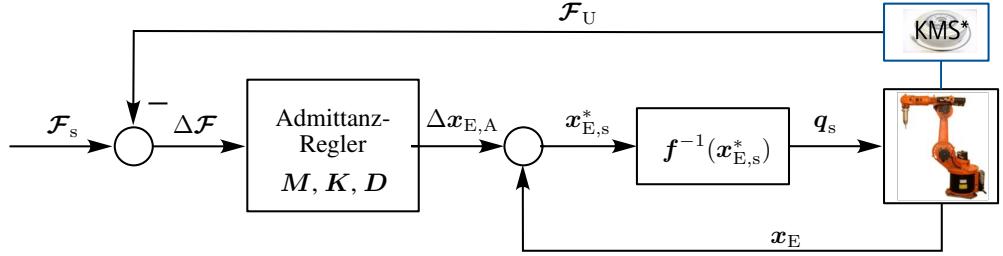


Bild 10.21: Blockschaltbild einer Admittanzregelung (der unterlagerte Positionsregler ist nicht dargestellt)

aus in Abhängigkeit der Regeldifferenz $\Delta\mathcal{F} = \mathcal{F}_s - \mathcal{F}_U$ der Kraft eine der gewünschten mechanischen Admittanz entsprechende Positionsänderung $\Delta\mathbf{x}_E$ des bestimmt werden muss. Im Bildbereich ergibt sich

$$\Delta\mathbf{x}_{E,A} \circ-\bullet \quad \Delta\mathbf{X}_{E,A}(s) = (\mathbf{M}s^2 + \mathbf{D}s + \mathbf{K})^{-1} \Delta\mathcal{F}(s) \quad (10.51)$$

Der aktuelle Sollwert $\mathbf{x}_{E,s}^*$ des unterlagerten Positionsregelkreises ergibt sich dann aus der aktuellen Position und $\Delta\mathbf{x}_{E,A}$ zu $\mathbf{x}_{E,s}^* = \mathbf{x}_E + \Delta\mathbf{x}_{E,A}$ und $\mathbf{q}_s = \mathbf{f}^{-1}(\mathbf{x}_{E,s}^*)$. Unter der Voraussetzung geringer Positionsänderungen $\Delta\mathbf{x}_{E,A}$ kann auf die Berechnung der inversen Kinematik \mathbf{f}^{-1} verzichtet werden, indem die Positionsdifferenz im Taskspace bestimmt und anschließend die differentielle inverse Kinematik in der Form

$$\Delta\mathbf{q}_A = \mathbf{J}(\mathbf{q})^{-1} \Delta\mathbf{x}_{E,A} \quad (10.52)$$

$$\mathbf{q}_{s,\text{neu}} = \mathbf{q}_{s,\text{alt}} + \Delta\mathbf{q}_A \quad (10.53)$$

für die Bestimmung des neuen Sollwerts der Gelenkwinkel genutzt wird.

Admittanzregler mit Sollwertvorgabe der Position

Mit der zusätzlichen Vorgabe einer Sollposition $\mathbf{x}_{E,s}$ für den Endeffektor wird diese durch $\Delta\mathbf{x}_{E,A}$ angepasst und es folgt $\mathbf{x}_{E,s}^* = \mathbf{x}_{E,s} + \Delta\mathbf{x}_{E,A}$. Es ergibt sich dann das in Bild 10.22 dargestellte Blockschaltbild

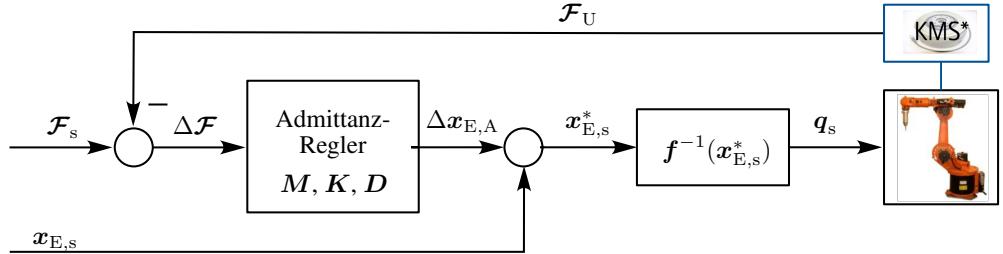


Bild 10.22: Blockschaltbild einer Admittanzregelung mit Positionssollwert (der unterlagerte Positionsregler ist nicht dargestellt)

Für die Bestimmung des vom unterlagerten Positionsreglers für \mathbf{q} gilt wieder, dass bei kleinen Positionsabweichungen die differentielle Kinematik Verwendung finden kann. Dann gilt:

$$\Delta\mathbf{q}_A = \mathbf{J}(\mathbf{q})^{-1} (\mathbf{x}_{E,s}^* - \mathbf{x}_{E,A}) \quad (10.54)$$

$$\mathbf{q}_{s,\text{neu}} = \mathbf{q}_{s,\text{alt}} + \Delta\mathbf{q}_A \quad (10.55)$$

Eigenschaften der Admittanzregelung

Aus den vorherigen Ausführungen lassen sich zusammenfassend folgende Eigenschaften der Admittanzregelung angeben:

- Die Admittanzregelung ist insbesondere für Industrieroboter gut geeignet, die i.d.R. mit Positionen oder Geschwindigkeitsreglern ausgestattet sind, aber keine Drehmomentschnittstelle aufweisen.
- Die Admittanzregelung erlaubt eine direkte Vorgabe einer Sollkraft \mathcal{F}_s .
- Für die Wahl von M , D und K gilt das bereits für die Impedanzregler gesagte.
- Wie beim Impedanzregler ist Sensorik für die Erfassung der Umgebungskraft \mathcal{F}_U erforderlich

Vergleich von Impedanz- und Admittanzregler

Zusammenfassen sollen hier noch einmal die Admittanz- der Impedanzregelung gegenübergestellt werden.

Admittanzregler Bei geringen gewünschten Steifigkeiten, also hohen Nachgiebigkeiten entstehen bei kleinen Abweichungen $\Delta\mathcal{F}$ der Ist- von der Sollkraft relativ große Positionsabweichungen, die der Positionsregler ggf. nicht ausreichend schnell ausregeln kann. Die gewünschte mechanische Admittanz stellt sich für den Roboter jedoch nur ein, wenn die erforderliche Positionsabweichung idealerweise unmittelbar umgesetzt wird. Für hohe Nachgiebigkeiten ist die Admittanzregelung daher eher ungeeignet. Für große bis mittlere Steifigkeiten wird hingegen ein gutes Verhalten erzielt.

Impedanzregler Bei hohen gewünschten Steifigkeiten führen kleine Änderungen der Positionsabweichungen zu großen Sollkräften und somit zu relativ großen Regelabweichungen in der Drehmomentregelung. Auch wenn diese schneller abgebaut werden können als Regeldifferenzen der inneren Positionsregelung bei der Admittanzregelung, ist der Impedanzregler daher für hohe Steifigkeiten eher ungeeignet. Für große und mittlere Nachgiebigkeiten ergibt sich hingegen ein gutes Verhalten.

10.6.2 Kraftbasierte direkte Kraftregelung

Im Gegensatz zur Admittanzregelung bleibt bei der kraftbasierten direkten Kraftregelung die Lage des Roboters unberücksichtigt. Eine Bewegung wird rein aus den gemessenen Kontaktkräften erzeugt (vgl. Bild 10.23). Anwendung findet dieses Verfahren beispielsweise bei sogenannten Teach-In-Verfahren (Führen des Roboters durch Kontakt am Endeffektor).

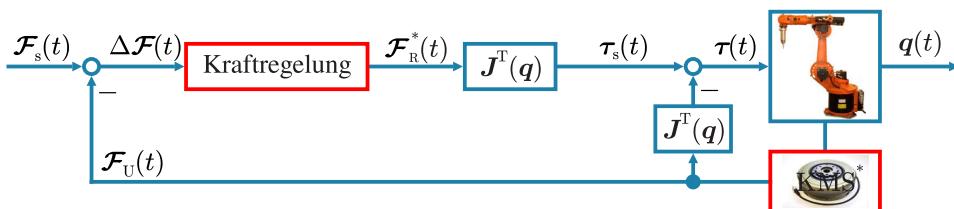


Bild 10.23: Blockschaltbild einer kraftbasierten direkten Kraftregelung im Taskspace

Als Regelgesetz für die Kraftregelung können die bekannten Rückführungen P, PD, PI oder PID (vgl. Kapitel 6) verwendet werden. In der Praxis werden häufig PI- und PID-Regler eingesetzt (gute Stabilität). Die wichtigsten Merkmale der kraftbasierten direkten Kraftregelung sind:

- Direkte Vorgabe von Sollkontaktkräften,
- zum „Führen“ des Roboters geeignet (keine Vorgabe von Solllagen),
- zusätzliche Sensorik erforderlich.

10.7 Hybride Kraft-/Lageregelung

Bei einigen Anwendungen bietet es sich an, eine hybride Kraft-/Lageregelung zu verwenden. In diesem Fall werden bestimmte Freiheitsgrade (festgelegt über die diagonale Auswahlmatrix S) lage- und die übrigen Freiheitsgrade kraftgeregt. Für diesen Ansatz ist beispielsweise das am Anfang des Kapitels vorgestellte Beispiel vom Einführen eines Bolzens in eine Bohrung gut geeignet. Die Gesamtreglerstruktur setzt sich aus einer Lage- sowie einer Kraftregelung (unabhängig voneinander) zusammen (vgl. Bild 10.24).

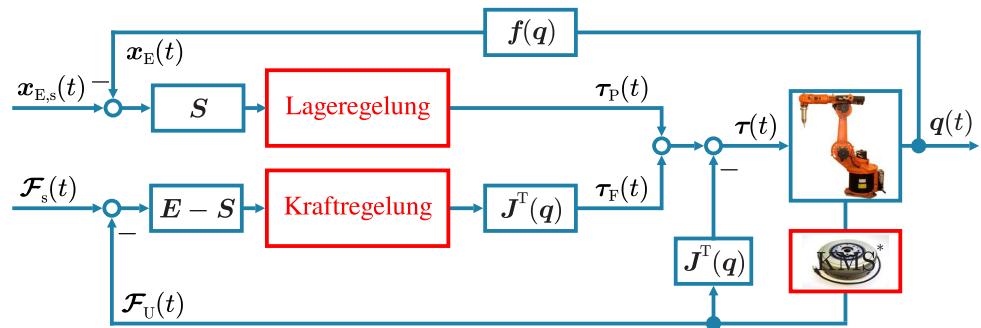


Bild 10.24: Blockschaltbild einer hybriden Kraft-/Lageregelung im Taskspace

Zur diagonalen Auswahlmatrix S : Mittels der Auswahlmatrix $S = \text{diag}(s_1, \dots, s_m)$ wird festgelegt, welche Endeffektorfreiheitsgrade (Gesamtanzahl m) kraft- und welche lagegeregt werden. Dabei bedeutet:

- $s_i = 0$: Freiheitsgrad x_{E_i} kraftgeregt,
- $s_i = 1$: Freiheitsgrad x_{E_i} lagegeregt.

Die Festlegung der Diagonalelemente der Auswahlmatrix S basiert auf den **natürlichen** und **künstlichen** Randbedingungen, die nachstehend erläutert werden.

Natürliche Randbedingungen

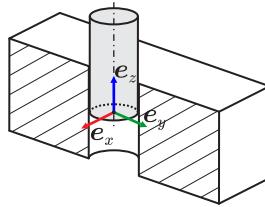
Definition
Natürliche
Randbedingungen

In Richtung jedes Freiheitsgrades des Taskspace existieren, festgelegt durch die Interaktion zwischen Roboter und Umwelt, lage- oder kraftbasierte Zwangsbedingungen. Diese sind direkt durch die geometrischen Randbedingungen der zu verrichtenden Aufgabe vorgegeben und daher nicht frei wählbar.

Beispiel

Reibungsfreies Einführen eines Bolzens in eine Bohrung

Das in Bild 10.25 gezeigte Beispiel dient zur Erläuterung der natürlichen Zwangsbedingungen. Dabei wird jeder Freiheitsgrad dahingehend analysiert, ob aufgrund der Vorgaben (der Bolzen kann reibungsfrei eingeführt werden) bzw. Geometrie, Bewegungen oder Kräfte möglich sind. So ist im Beispiel keine Bewegung in x_E - und y_E -Richtung möglich, so dass die dazugehörigen Geschwindigkeiten zu null gesetzt werden. Für die z_E -Richtung gilt, dass beim Einführen

**Bild 10.25:** Einführen eines Bolzens in eine Bohrung

keine Reibungskräfte auftreten und somit $F_z = 0$ gilt. Die Überlegungen für die rotatorischen Freiheitsgrade sind analog und in Tabelle 10.1 zusammengefasst.

Tabelle 10.1: Natürliche Randbedingungen beim Einführen eines Bolzens

Freiheitsgrad x_{E_i}	Natürliche Randbedingung
x_E	$\dot{x}_E = 0$
y_E	$\dot{y}_E = 0$
z_E	$F_z = 0$
α_E (Drehung um x_E)	$\dot{\alpha}_E = 0$
β_E (Drehung um y_E)	$\dot{\beta}_E = 0$
γ_E (Drehung um z_E)	$M_z = 0$

Künstliche Randbedingungen

Aufgrund der natürlichen Randbedingungen können nur für bestimmte Freiheitsgrade Vorgaben vom Benutzer gemacht und diese somit geregelt werden (künstliche Randbedingungen). Die entsprechenden Freiheitsgrade ergeben sich direkt aus den natürlichen Randbedingungen.

Definition
Künstliche
Randbedingungen

Für das Beispiel „reibungsfreies Einführen eines Bolzens in eine Bohrung“ sind die natürlichen und künstlichen Randbedingungen in Tabelle 10.2 zusammengefasst. Das Einführen soll mit der Sollgeschwindigkeit $\dot{z}_{E,s}$ durchgeführt werden.

Tabelle 10.2: Zusammenfassung der natürlichen und künstlichen Randbedingungen

Freiheitsgrad x_{E_i}	Natürliche Randbedingung	Künstliche Randbedingung
x_E	$\dot{x}_E = 0$	$F_x = 0$
y_E	$\dot{y}_E = 0$	$F_y = 0$
z_E	$F_z = 0$	$\dot{z}_E = \dot{z}_{E,s}$
α_E	$\dot{\alpha}_E = 0$	$M_x = 0$
β_E	$\dot{\beta}_E = 0$	$M_y = 0$
γ_E	$M_z = 0$	$\dot{\gamma}_E = 0$

Im Fall einer hybriden Kraft-/Lagerregelung ergibt sich die Steuermatrix S für das Beispiel aus den künstlichen Randbedingungen demnach zu:

$$S = \text{diag}(0, 0, 1, 0, 0, 1). \quad (10.56)$$

Die Freiheitsgrade $x_E, y_E, \alpha_E, \beta_E$ werden folglich kraftgeregelt und die Freiheitsgrade z_E, γ_E positionsgeregelt.

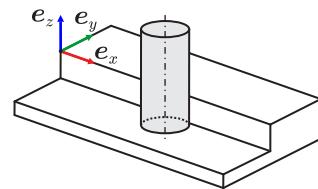
Abschließende Bemerkungen:

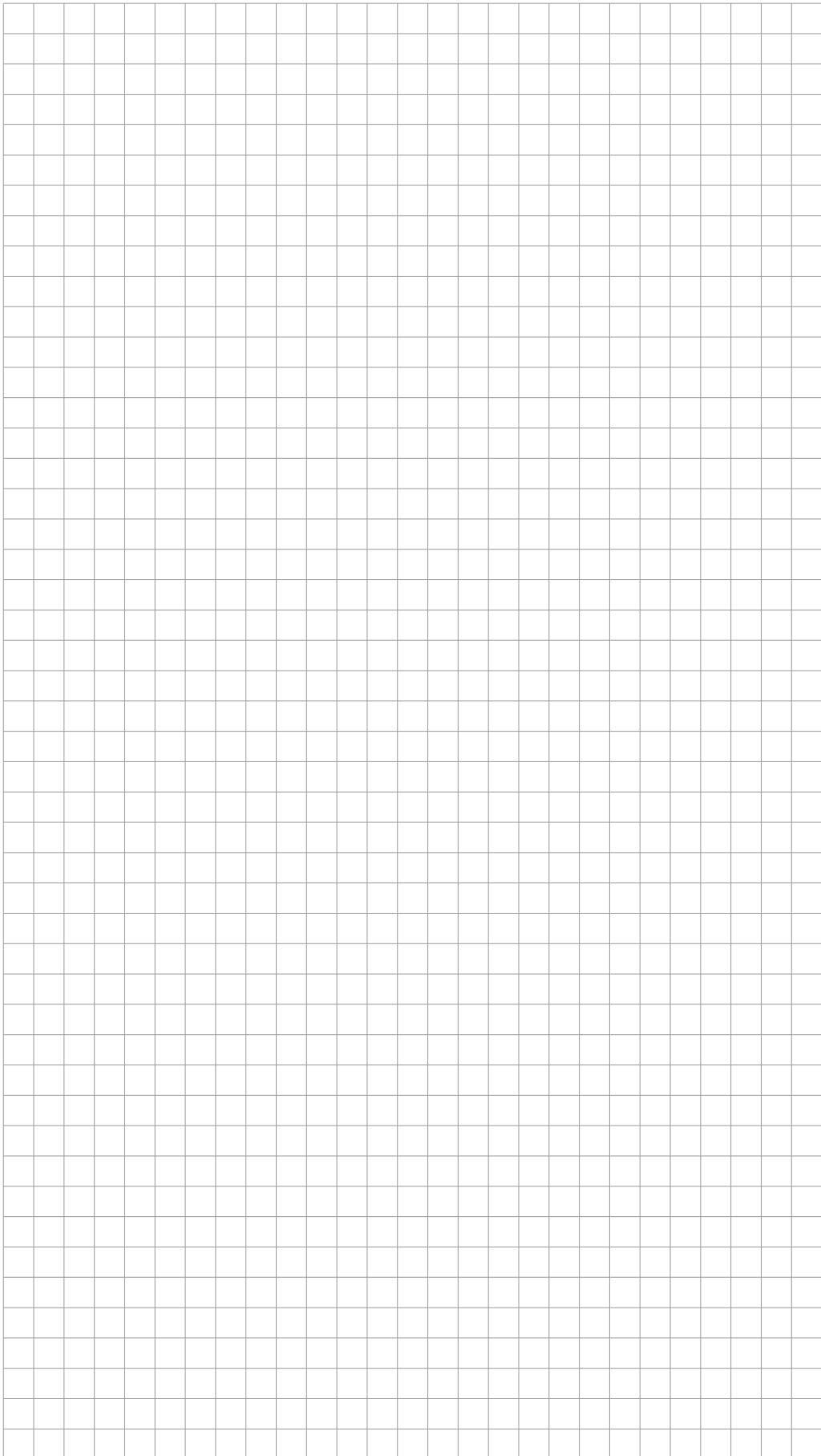
- Stabilitätsuntersuchungen (sowie Vorgehensweisen zur Findung geeigneter Regelparameter etc.) der vorgestellten Regelverfahren werden im Umfang dieser Vorlesung aufgrund ihrer Komplexität nicht behandelt. Interessierte finden Hinweise hierzu in „An Overview of Robot Force Control“ (vgl. Literaturhinweise am Anfang dieses Kapitels).

- Es existieren weitere, zahlreiche Variationen der hier behandelten Regelungsansätze. Die im Rahmen dieser Vorlesung gezeigten Verfahren repräsentieren lediglich eine Auswahl verbreiteter (grundlegender) Regelungsstrategien.
 - Grundsätzlich können die vorgestellten Ansätze durch die bereits gezeigten Vorsteuerungsmethoden ergänzt werden.

Fragen zur Selbstkontrolle

1. Welche rein mechanische Möglichkeit zur „Steifigkeitsregelung“ ist Ihnen bekannt? Erläutern Sie die Grundidee!
 2. Skizzieren Sie das Blockschaltbild einer hybriden Kraft-/Lageregelung und erläutern Sie das Prinzip!
 3. Der Endeffektor eines Roboters soll den abgebildeten rotationssymmetrischen Bolzen in x_0 -Richtung entlang einer Kante bewegen (vgl. Abbildung). Es ist anzunehmen, dass keine Reibung vorhanden sei. Geben Sie für diese Aufgabenstellung die künstlichen und natürlichen Randbedingungen an!







11 Sensoren

Dieses Kapitel behandelt häufige in Robotern eingesetzte Sensoren. Sie erfassen eine physikalische Messgröße und wandeln diese in ein elektrisches Signal um, das anschließend weiterverarbeitet wird (z. B. zum Schließen von Regelkreisen). Zur Charakterisierung und Bewertung von Sensoren sind verschiedene Kenngrößen wichtig, die in Abschnitt 11.1 eingeführt werden. Anschließend werden die Funktionsprinzipien gängiger Sensorarten detailliert beschrieben. Dies beinhaltet Winkelgeber (Abschnitt 11.2), Inertialsensoren (Abschnitt 11.3) und Kraft-Moment-Sensoren (Abschnitt 11.4). In den folgenden Abschnitten ist die Antwort auf die Frage „Wie erfasse ich die Rückführsignale, die ich beispielsweise in meiner Regelung benötige und verwende?“ gesucht.

Das letzte Kapitel der Robotik I Veranstaltung kann das Thema Sensoren lediglich überblicksartig behandeln und soll die vorherigen Kapitel (die den inhaltlichen Schwerpunkt bilden) abrunden. Für ein vertieftes Verständnis der Sensorik (insbesondere bezüglich Funktionsweise und Auswertung) sei auf die entsprechenden Spezialvorlesungen, wie beispielsweise „Elektronisch betriebene Kleinmaschinen“ des Instituts für Antriebssysteme und Leistungselektronik sowie „Messtechnik I und II“ des Instituts für Mess- und Regelungstechnik, verwiesen.

Als weiterführende Literatur wird empfohlen:

- E. Kiel. *Antriebslösungen: Mechatronik für Produktion und Logistik*. Heidelberg, Deutschland: Springer, 2007, S. 124–130
- S. Hesse und G. Schnell. *Sensoren für die Prozess- und Fabrikautomation: Funktion – Ausführung – Anwendung*. 6. Aufl. Wiesbaden, Deutschland: Vieweg+Teubner Verlag, 2014, S. 185–231
- E. Schrüfer. *Elektrische Messtechnik*. München, Deutschland: Carl Hanser Verlag, 2007

11.1 Kenngrößen

Ein Sensor (häufig auch als Aufnehmer bezeichnet) erfasst bestimmte physikalische oder chemische Größen, wie beispielsweise Temperatur, Position, Beschleunigung, mechanische Kräfte und Säuregrad. Diese werden auf unterschiedlichste Art und Weise (physikalisch/chemisch) quantitativ gemessen und in leicht übertragbare sowie weiterverarbeitbare (meist elektrische) Signale umgeformt. Bild 11.1 zeigt das prinzipielle Signalflussdiagramm eines Sensors.

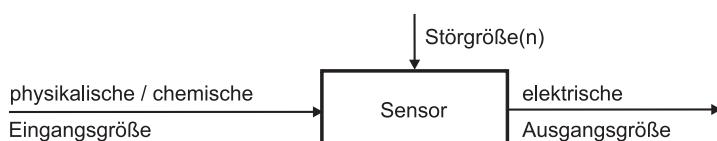


Bild 11.1: Signalflussdiagramm eines Sensors

Im Folgenden werden, angelehnt an DIN 1319, die wichtigsten **Kenngrößen von Sensoren** definiert. Sie bilden die Grundlage für die Bewertung von bzw. den Vergleich zwischen Sensoren:

Definition
Sensorkenngrößen

- Auflösung:
Die Auflösung gibt den kleinsten zuverlässig wahrnehmbaren Unterschied von zwei Eingangssignalen gleicher Dimension an.
- Messbereich:
Der Messbereich eines Messgeräts gibt die Grenzen (Minimum und Maximum) des messbaren Bereichs einer Messgröße an.
- Präzision (zufällige Abweichung):
Die Präzision ist ein Maß für die Übereinstimmung zwischen unabhängigen Messergebnissen unter konstanten Bedingungen (vgl. Kapitel 1, Wiederholgenauigkeit). Liegen die Messwerte nah beieinander, ist eine hohe Präzision vorhanden (vgl. Bild 11.2).
- Richtigkeit (systematische Abweichung):
Die Richtigkeit ist ein Maß für die Übereinstimmung zwischen dem Mittelwert mehrerer Messungen unter konstanten Bedingungen und dem anerkannten Referenzwert. Wenn der Mittelwert gut mit dem wahren Wert übereinstimmt, liegt eine hohe Richtigkeit vor (vgl. Bild 11.2).
- Messabweichung:
Die Messabweichung ist die Differenz zwischen Messwert und dem anerkannten Referenzwert.
- (Mess-)Genauigkeit:
Die Messgenauigkeit ist die maximal mögliche Abweichung eines aus Messungen gewonnenen Wertes von der wahren (exakten) Größe in einem Konfidenzintervall (meist 95%). Sie charakterisiert demnach die Summe aller möglichen Fehler eines Sensors. Der wahre Wert ist hier lediglich eine „theoretische“ Größe, da er im Allgemeinen unbekannt ist. Daher wird hierfür seit einiger Zeit der Ausdruck „anerkannter Referenzwert“ verwendet. Eine hohe Genauigkeit kann nur erreicht werden, wenn die Präzision und die Richtigkeit gut sind (vgl. Bild 11.2).

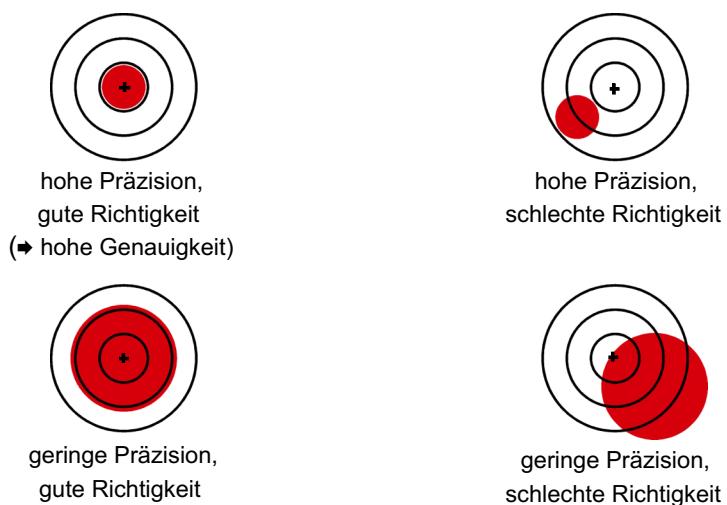


Bild 11.2: Anschauliche Verdeutlichung von Präzision, Richtigkeit und Genauigkeit (die roten Bereiche entsprechen den gemessenen Werten)

Anforderungen an Sensoren

Wichtige Anforderungen an Sensoren sind unter anderem:

- Eindeutige und reproduzierbare Abbildung der Eingangs- auf die Ausgangsgröße innerhalb des Messbereichs,
- Ausgangsgröße darf nur von der zu erfassenden Eingangsgröße abhängen, nicht von anderen Größen, wie z. B. der Temperatur,
- Messsystem muss vernachlässigbar kleine Rückwirkungen auf die Messgröße haben.

Wünschenswerte Eigenschaften von Sensoren

Neben den zuvor genannten Anforderungen an Sensoren, die für eine korrekte Erfassung der Messgröße unabdingbar sind, existieren eine Reihe weiterer, wünschenswerter Eigenschaften, wie:

- lineare Abbildung der Eingangs- auf die Ausgangsgröße,
- Unempfindlichkeit gegenüber elektromagnetischen Störungen,
- einfache Stromversorgung,
- Möglichkeiten zur Funktionskontrolle,
- kleine Baugröße,
- Schutz gegen (mechanische) Überlast.

11.2 Winkelmesssysteme

Die Messung von Winkeln oder Wegen ist in der Robotik essentiell und kann unter Verwendung verschiedenster Messprinzipien erfolgen. Im Folgenden werden zwei typische Verfahren (Sensortypen) vorgestellt: Die Resolvertechnik in Abschnitt 11.2.1 und die optischen Encoder in Abschnitt 11.2.2.

11.2.1 Resolver

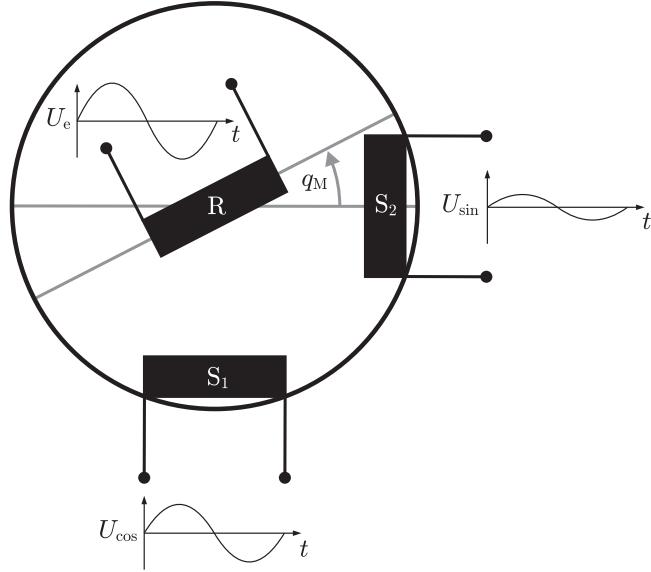
Als Resolver bezeichnet man im Allgemeinen einen induktiv arbeitenden Messumformer, der die Winkellage eines Rotors in eine elektrische Größe umwandelt. Hierbei wird auf elektrisch-induktiver Basis eine analoge Information über die Winkelstellung eines Motors erzeugt. Diese ist ggf. mit der inversen Getriebeübersetzung $1/u_G$ zu multiplizieren, um den abtriebsseitigen Gelenkwinkel q zu erhalten, wobei die Getriebelastizität unberücksichtigt bleibt.

Funktionsprinzip

Elektrisch betrachtet handelt es sich bei einem Resolver um einen generatorisch betriebenen Elektromotor. Im einfachsten Fall besteht er aus zwei um 90° versetzten ortsfesten (Stator) und einer mit der Motorachse mechanisch verbundenen Spule. Bei dem in Bild 11.3 dargestellten Beispiel wird im Rotor R eine sinusförmige Trägerfrequenzspannung U_e :

$$U_e = U_{e_0} \sin(\omega t), \quad (11.1)$$

mit der Amplitude U_{e_0} und einer Frequenz ω ($\omega \gg q_M$) erzeugt. Diese induziert (aufgrund der um 90° versetzten Statorstränge S_1 und S_2) in den Statorwicklungen der ortsfesten Spulen Spannungen mit einer kosinus- und sinusförmigen Amplitude U_{\cos} bzw. U_{\sin} . Die induzierten

Bild 11.3: Funktionsprinzip eines Resolvers (für konstantes q_M)

Spannungen sind dabei direkt von dem Drehwinkel q_M des Rotors bzw. Motors abhängig:

$$\begin{aligned} U_{\cos} &= U_e u_{rs} \cos(p q_M) \\ &= U_{e0} \sin(\omega t) u_{rs} \cos(p q_M) \\ &= \underbrace{U_{e0} u_{rs} \cos(p q_M)}_{U_{\cos}^*} \sin(\omega t), \end{aligned} \quad (11.2)$$

$$\begin{aligned} U_{\sin} &= U_e u_{rs} \sin(p q_M) \\ &= U_{e0} \sin(\omega t) u_{rs} \sin(p q_M) \\ &= \underbrace{U_{e0} u_{rs} \sin(p q_M)}_{U_{\sin}^*} \sin(\omega t). \end{aligned} \quad (11.3)$$

Hierbei entspricht u_{rs} dem Übersetzungsverhältnis der Rotor- zu den Statorwicklungen und p der Polpaarzahl. Zu beachten ist, dass die Abtastung bei maximaler Amplitude der Trägerfrequenzspannung erfolgt ($\omega t = 2n\pi + \pi/2, n \in \mathbb{N}$), um die Erregerfrequenz zu eliminieren. Somit gilt:

$$\begin{aligned} U_{\cos} &= U_{e0} u_{rs} \cos(p q_M) \\ &= U_{\cos}^*, \end{aligned} \quad (11.4)$$

$$\begin{aligned} U_{\sin} &= U_{e0} u_{rs} \sin(p q_M) \\ &= U_{\sin}^*. \end{aligned} \quad (11.5)$$

Für das dargestellte Beispiel eines Resolvers mit zwei um 90° versetzten Statorsträngen (häufigste Bauform) ist die Polpaarzahl gleich eins ($p = 1$). Durch Ergänzung beispielsweise einer weiteren Rotorwicklung (90° versetzt) und Modifikation der ortsfesten Spulenanzordnung (eingeschlossener Winkel $\pm 45^\circ$) resultiert eine Polpaarzahl von zwei ($p = 2$). Dies hat eine höhere Auflösung zur Folge, bedeutet aber gleichzeitig eine Verkleinerung des eindeutigen Messbereichs, da der (aufgrund der elektromagnetischen Induktion) gemessene Winkel in diesem Fall während einer mechanischen Umdrehung zweifach den Bereich von 0° bis 360° durchläuft (vgl. Eigenschaften). Dies ergibt sich aus der Tatsache, dass das durch die zweite, um 90° versetzte Spule erzeugte

Magnetfeld das der ersten Spule derart überlagert, dass ein um 180° rotationssymmetrisches Gesamtmagnetfeld erzeugt wird.

Aus dem Verhältnis der beiden Spannungen U_{\sin}^* und U_{\cos}^* lässt sich abschließend die Winkelstellung q_M des Motors ermitteln:

$$q_M = \frac{1}{p} \arctan2(U_{\sin}^*, U_{\cos}^*). \quad (11.6)$$

Zur Einprägung der Trägerfrequenzspannung U_e in die Rotorsspule kommen je nach Ausführung des Resolvers Schleifringe (Verschleißproblematik) oder berührungslose Ringtransformatoren zum Einsatz.

Neben dem beispielhaft beschriebenen Aufbau existieren zahlreiche weitere Bauformen, wie beispielsweise Differenzialwinkellagegeber, bürstenlose Winkelgeber, induktive Winkelgeber, etc (vgl. hierzu [Kie07, S. 124–130] sowie [N.N04]).

Folgende Eigenschaften zeichnen magnetische Resolver aus:

- liefern Winkelinformationen innerhalb $(1/p)$ -Umdrehungen absolut (aufgrund der Getriebeübersetzung ist der abtriebsseitige Gelenkwinkel nicht absolut, so dass eine Referenzierung erforderlich ist),
- im Vergleich zu optischen Encodern (siehe Abschnitt 11.2.2) robust gegenüber Temperaturschwankungen, Vibrationen, Spannungsstörungen, etc.,
- empfindliche Auswerteelektronik ist in der Regel im Schaltschrank untergebracht (kein Kontakt mit der Umwelt).

11.2.2 Optische Encoder

Optische Encoder werden in vielen (industriellen) Anwendungen zur Winkel- und Wegmessung eingesetzt.

Funktionsprinzip

Zur Messung wird ein Lichtstrahl auf einen Photodetektor ausgerichtet. Ein auf einer Scheibe angebrachtes codiertes Muster unterbricht periodisch den Lichtstrahl. Die einzelnen Unterbrechungen können anschließend gemessen („gezählt“) werden. Das Funktionsprinzip ist in Bild 11.4 gezeigt. Für weiterführende Informationen bezüglich optischer Encoder wird auf „E. Kiel (Lenze AG) – Antriebslösungen (Mechatronik für Produktion und Logistik), S. 124-130“ sowie auf die Vorlesung „Elektronisch betriebene Kleinmaschinen“ verwiesen.

Folgende Eigenschaften zeichnen optische Encoder aus:

- inkrementelles Verfahren, in der Regel ist jedoch eine Referenzmarke zur Absolutwertbestimmung vorhanden,
- für hohe Messauflösungen werden mehrere konzentrisch angeordnete Scheiben kombiniert),
- durch versetzte Abtastung der selben Teilscheibe lässt sich zudem die Drehrichtung erkennen.

11.3 Inertiale Messeinheit

Bei einer inertialen Messeinheit handelt es sich um eine Kombination aus Beschleunigungs-, Drehraten- und teilweise zusätzlichen Magnetfeldsensoren in einer kompakten Einheit. Magnet-

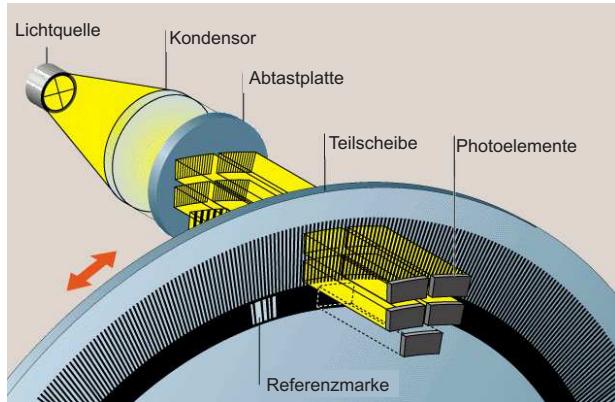


Bild 11.4: Funktionsprinzip eines optischen Encoders (Quelle: Heidenhain)

feldsensoren reagieren allerdings sehr empfindlich auf elektromagnetische Felder (der Motoren) und ferromagnetische Materialien (z. B. Robotersegmente). Aus diesem Grund besitzen die in der Robotik eingesetzten inertialen Messeinheiten in der Regel keine Magnetfeldsensoren.

Die zugehörige Sensorik einer inertialen Messeinheit setzt sich aus jeweils drei orthogonal zueinander angeordneten Beschleunigungs- und Drehratensensoren (Funktionsprinzipien im Folgenden Abschnitt 11.3.1 bzw. Abschnitt 11.3.2) mit ihren sensitiven Achsen in Richtung der körperfesten Koordinatenachsen zusammen (siehe Bild 11.5).

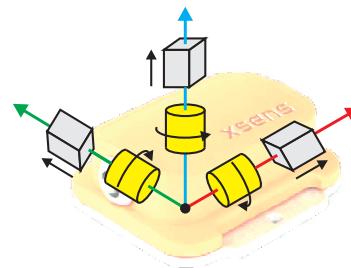


Bild 11.5: Inertiale Messeinheit (ohne Magnetfeldsensoren)

11.3.1 Beschleunigungssensoren

Als Maß für die Beschleunigung detektieren Beschleunigungssensoren eine auf eine Masse ausgeübte Kraft ($F = m a$). Grundsätzlich existieren hierfür verschiedene Methoden. Beispielsweise kann kapazitiv die Verschiebung einer seismischen Masse in Richtung der drei Koordinatenachsen gemessen werden (vgl. Bild 11.6). Anhand der gemessenen Werte wird anschließend die Beschleunigung (inklusive Gravitation) bestimmt.

Die nachstehenden Herleitungen beziehen sich auf den in Bild 11.7 gezeigten eindimensionalen Fall, bei dem eine Kraft F die federnd gelagerte (Federsteifigkeit k) Masse m auslenkt. Dabei gelten die folgenden grundlegenden Beziehungen:

$$F = m a, \quad (11.7)$$

$$\Delta d = k F, \quad (11.8)$$

wobei a der Beschleunigung der Masse m entspricht.

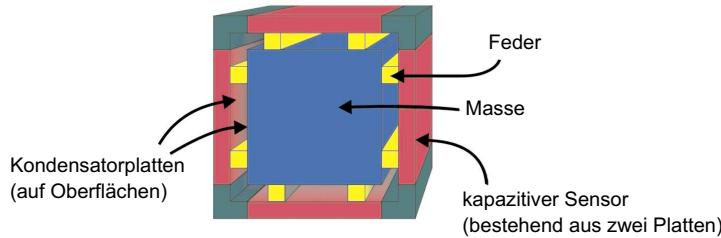


Bild 11.6: Funktionsprinzip eines Beschleunigungssensors (Quelle: Luinge, Inertial Sensing of Human Movement)

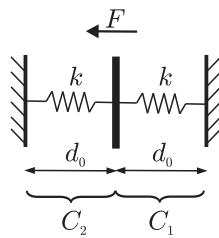


Bild 11.7: Prinzipskizze eines Beschleunigungssensors

Allgemein berechnet sich die elektrische Kapazität eines Plattenkondensators folgendermaßen:

$$C = \epsilon_0 \epsilon_r \frac{A}{d}, \quad (11.9)$$

wobei ϵ_0 die elektrische Feldkonstante ($\epsilon_0 \approx 8.86 \cdot 10^{-12} \text{ As/Vm}$) und ϵ_r die materialabhängige Dielektrizitätszahl sind. Die übrigen Größen sind die Fläche der Kondensatorplatten A sowie der Abstand der Platten d . Übertragen auf das in Bild 11.7 gezeigte Schema eines Beschleunigungssensors ergeben sich die Kapazitäten C_1 und C_2 zu:

$$C_1 = \epsilon_0 \epsilon_r \frac{A}{d + \Delta d}, \quad (11.10)$$

$$C_2 = \epsilon_0 \epsilon_r \frac{A}{d - \Delta d}. \quad (11.11)$$

Nach Umformung resultiert für C_1 :

$$C_1 = \epsilon_0 \epsilon_r \frac{A}{d} \frac{1}{1 + \frac{\Delta d}{d}} \approx \epsilon_0 \epsilon_r \frac{A}{d} \left(1 - \frac{\Delta d}{d}\right) \quad (11.12)$$

$$= \epsilon_0 \epsilon_r \frac{A}{d} - \epsilon_0 \epsilon_r \frac{A}{d} \frac{\Delta d}{d} = C_0 - \Delta C. \quad (11.13)$$

Analog folgt für C_2 :

$$C_2 = \epsilon_0 \epsilon_r \frac{A}{d} \frac{1}{1 - \frac{\Delta d}{d}} \approx \epsilon_0 \epsilon_r \frac{A}{d} \left(1 + \frac{\Delta d}{d}\right) \quad (11.14)$$

$$= \epsilon_0 \epsilon_r \frac{A}{d} + \epsilon_0 \epsilon_r \frac{A}{d} \frac{\Delta d}{d} = C_0 + \Delta C. \quad (11.15)$$

Bild 11.8 zeigt die Verschaltung der beiden Kondensatoren zu einer Halbbrücke, welche oft zur Auswertung der Sensorsignale eingesetzt wird. Die Spannung U_0 ist die Versorgungsspannung der Messbrücke und U_d die Messspannung.

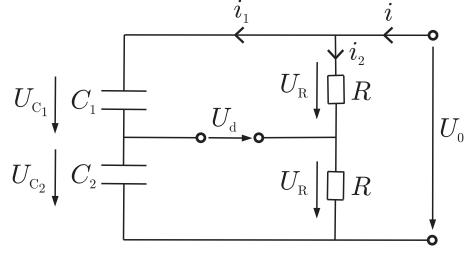


Bild 11.8: Verschaltung eines Beschleunigungssensors in einer Halbbrücke

Die an den Widerständen abfallende Spannung U_R lautet:

$$U_R = \frac{1}{2} U_0, \quad (11.16)$$

für die an einem Kondensator anliegende Spannung gilt:

$$U_{C_i} = \frac{1}{C_i} \int i_1 dt. \quad (11.17)$$

Die Anwendung der Maschenregel liefert:

$$U_{C_1} + U_{C_2} = U_0. \quad (11.18)$$

Die Gesamtkapazität C_{ges} berechnet sich zu:

$$\frac{1}{C_{\text{ges}}} = \frac{1}{C_1} + \frac{1}{C_2} \quad \Rightarrow \quad C_{\text{ges}} = \frac{C_1 C_2}{C_1 + C_2}. \quad (11.19)$$

Daraus folgt die Spannung U_0 :

$$U_0 = \frac{C_1 + C_2}{C_1 C_2} \int i_1 dt, \quad (11.20)$$

mit

$$\int i_1 dt = C_1 U_{C_1} \quad (11.21)$$

folgt weiterführend:

$$U_0 = \frac{C_1 + C_2}{C_1 C_2} C_1 U_{C_1}. \quad (11.22)$$

Das lässt sich anschließend nach der gesuchten Kondensatorspannung U_{C_1} auflösen:

$$U_{C_1} = U_0 \frac{C_2}{C_1 + C_2} \quad (11.23)$$

$$= U_0 \frac{C_0 + \Delta C}{C_0 - \Delta C + C_0 + \Delta C} \quad (11.24)$$

$$= U_0 \frac{C_0 + \Delta C}{2 C_0}. \quad (11.25)$$

Ebenfalls aus der Maschengleichung erhält man:

$$U_{C_1} + U_d - U_R = 0 \quad \Leftrightarrow \quad U_d = U_R - U_{C_1}. \quad (11.26)$$

Einsetzen von Gleichung (11.25) und Gleichung (11.16) in Gleichung (11.26) setzt abschließend die gemessenen Spannungen mit der Kraft (und damit mit der Beschleunigung) in Beziehung:

$$U_d = \frac{U_0}{2} - \frac{U_0}{2} \left(1 + \frac{\Delta C}{C_0} \right) \quad (11.27)$$

$$= -\frac{U_0}{2} \frac{\Delta C}{C_0} = -\frac{U_0}{2} \frac{\epsilon_0 \epsilon_r \frac{A}{d} \frac{\Delta d}{d}}{\epsilon_0 \epsilon_r \frac{A}{d}} \quad (11.28)$$

$$= -\frac{U_0}{2} \frac{\Delta d}{d} = -\frac{U_0}{2} \frac{k F}{d}. \quad (11.29)$$

Somit berechnet sich die gesuchte Beschleunigung a zu:

$$a = -2 \frac{U_d}{U_0} \frac{d}{k m} \quad (11.30)$$

Anzumerken ist an dieser Stelle, dass dies eine vereinfachte Betrachtung ist. In der Realität wirkt die Beschleunigung zunächst auf das Gehäuse des Sensors und erst im Anschluss auf die seismische Masse.

11.3.2 Drehratensensoren

Auch die Messung der Drehgeschwindigkeiten (Winkelgeschwindigkeiten) um die körperfesten Koordinatenachsen kann auf unterschiedlichen Prinzipien bzw. physikalischen Mechanismen basieren. In der Regel wird eine Masse mittels piezoelektrischer Elemente in Schwingungen versetzt. Bei Rotation treten durch die vorhandenen Vibratoren Corioliskräfte auf, die die Masse auslenken, vgl. Bild 11.9. Diese Auslenkung ist senkrecht zur Vibrationsbewegung und propor-

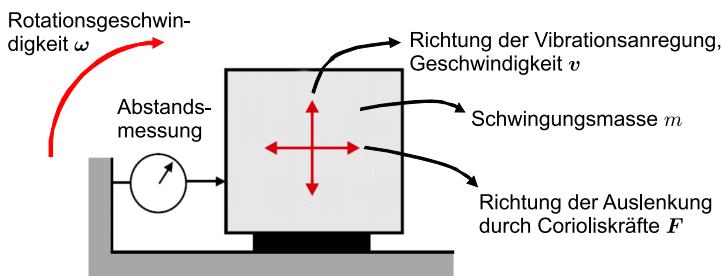


Bild 11.9: Funktionsprinzip eines Drehratensensors

tional zu der Winkelgeschwindigkeit. Die Corioliskräfte berechnen sich bekanntermaßen wie folgt:

$$\mathbf{F} = 2 m (\mathbf{v} \times \boldsymbol{\omega}). \quad (11.31)$$

Die Abstandsmessung kann beispielsweise kapazitiv erfolgen. In diesem Fall findet der in Abschnitt 11.3.1 dargestellte Formalismus Anwendung.

Abschließende Bemerkungen

Theoretisch könnte eine inertiale Messeinheit als Navigationssystem genutzt werden, denn mittels der gemessenen Daten (translatorische Beschleunigung und Winkelgeschwindigkeit) kann die aktuelle Lage relativ zu einer Ausgangsposition (x_0 bzw. α_0) bzw. -geschwindigkeit \dot{x}_0 berechnet

werden. Eine doppelte Integration der Beschleunigung \ddot{x}_i und eine einfache Integration der Drehraten $\dot{\alpha}_i$ liefert sowohl die Position x_i als auch die Orientierung α_i :

$$x_i(t) = \int \int \ddot{x}_i(t) dt dt + \dot{x}_{0_i} t + x_{0_i} + \frac{1}{2} a t^2, \quad (11.32)$$

$$\alpha_i(t) = \int \dot{\alpha}_i(t) dt + \alpha_{0_i} + \omega t. \quad (11.33)$$

Hierbei stellen a und ω Nullpunktfehler (Bias) dar.

Aufgrund des zufälligen Nullpunktfehlers (Bias) der Sensorik, der durch verschiedene Faktoren, wie beispielsweise Gravitationseinflüsse, Temperaturdrift, etc., beeinflusst wird, und des Signalrauschens ist es allerdings schwierig bis unmöglich, eine driftfreie Lage zu berechnen. Durch die Integration kommt es zu einer Verstärkung des Bias, welche den Einsatz von inertialen Messeinheiten zur Navigation ohne jegliche messtechnische Stützung in vielen Fällen verhindert.

11.4 Kraft-Moment-Sensor

Die mechanische Kraft bzw. das Moment kann meistens nicht direkt gemessen werden. Ordnet man jedoch im Kraftfluss einen Verformungskörper an, kann aus dessen Stauchung bzw. Dehnung auf die Größe der Kraft geschlossen werden. Ein Kraft-Moment-Sensor (vgl. Bild 11.10) erfordert eine besonders gestaltete Konstruktion, die in der Regel mit Dehnungsmessstreifen ausgestattet ist. Die Veränderungen dieses Verformungskörpers werden erfasst und in eine elektrische Ausgangsgröße umgewandelt.



Bild 11.10: 6-Achsen Kraft-Moment-Sensor (Quelle: SCHUNK)

Im Folgenden wird das diesen Sensoren zugrunde liegende Messverfahren detailliert beschrieben; zunächst für die Messung einer Kraft (Abschnitt 11.4.1) und anschließend für die Messung eines Drehmoments (Abschnitt 11.4.2).

11.4.1 Messung einer Kraft

Dehnungsmessstreifen (DMS) sind Messeinrichtungen zur Erfassung von Verformungen. Sie ändern bei Verformungen (Längenänderung) ihren elektrischen Widerstand und können demnach als Dehnungssensoren eingesetzt werden. In der Regel werden Dehnungsmessstreifen (siehe Bild 11.11) auf Bauteile geklebt, die sich unter Belastung verformen, den sogenannten Verformungskörpern. Deren Stauchung bzw. Dehnung führt dann zu einer messbaren Veränderung des Widerstands des DMS. Anhand dieser Widerstandsänderungen kann anschließend die angreifende Kraft F bestimmt werden.

Der Zusammenhang zwischen Dehnung ϵ und Widerstandsänderung ΔR eines Dehnungsmessstreifens lautet:

$$\frac{\Delta R}{R} = k \epsilon, \quad (11.34)$$

wobei k der so genannte Verstärkungsfaktor (für Folien DMS etwa $k = 2$) ist. Unter der Annahme einer elastischen Verformung nach dem Hooke'schen Gesetz ergibt sich:

$$\frac{\Delta R}{R} = k \epsilon = k \frac{F}{E A}. \quad (11.35)$$

Die Größen E und A sind hierbei Materialkonstanten (Elastizitätsmodul E , Querschnittsfläche A). Daraus ergibt sich für die zu bestimmende Kraft F :

$$F = \frac{\Delta R}{R} \frac{E A}{k}. \quad (11.36)$$

Die Widerstandsänderungen ΔR eines DMS sind recht klein, daher erfolgt die Auswertung in der Regel in Vollbrückenschaltungen (Wheatstone'sche Brücke¹). Damit ist auch eine Kompensation von Störungen, wie beispielsweise Thermospannungen möglich.

Wheatstone'sche Brücke mit vier Dehnungsmessstreifen

Die Messbrücke ist in Bild 11.12 dargestellt. Vier veränderliche Widerstände sind wie gezeigt verschaltet. Die Brücke wird mit der Spannung U_0 gespeist. Die Spannung U_d ist die sogenannte

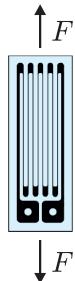


Bild 11.11: Folien DMS, Quelle: Wikipedia

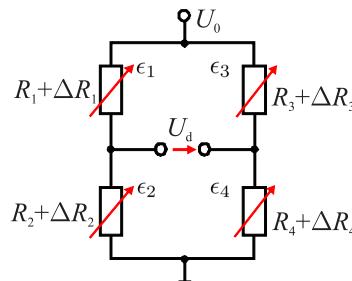


Bild 11.12: Wheatstone'sche Vollbrücke

Messspannung, die Ausgangsgröße. Im unbelasteten Fall gilt:

$$U_d = U_0 \frac{R_2 R_3 - R_1 R_4}{(R_1 + R_2)(R_3 + R_4)}. \quad (11.37)$$

Sind die Widerstände im unbelasteten Fall identisch ($R_i = R$), so gilt im Belastungsfall:

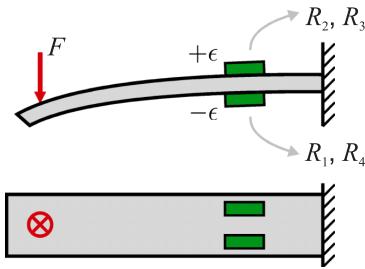
$$R_i = R + \Delta R_i, \quad (11.38)$$

mit $\Delta R_i \ll R$. Daraus folgt anschließend:

$$U_d \approx \frac{U_0}{4} \left(\frac{\Delta R_2}{R} + \frac{\Delta R_3}{R} - \frac{\Delta R_1}{R} - \frac{\Delta R_4}{R} \right) \approx \frac{k U_0}{4} (\epsilon_2 + \epsilon_3 - \epsilon_1 - \epsilon_4). \quad (11.39)$$

Eine Steigerung der Empfindlichkeit des Messaufbaus ist durch geschickte Anordnung der DMS auf Ober- und Unterseite eines Biegebalkens möglich. Die Dehnungen ϵ_2 und ϵ_3 müssen dasselbe Vorzeichen haben, ebenso ϵ_1 und ϵ_4 . Unterscheiden sich jetzt noch die Vorzeichen von ϵ_2 , ϵ_3 und ϵ_1 , ϵ_4 , nimmt Gleichung (11.40) betragsmäßig ein Maximum an. Dies erreicht man beispielsweise dadurch, dass R_2 und R_3 auf der Oberseite und R_1 und R_4 auf der Unterseite eines Biegebalkens angebracht sind (siehe Bild 11.13).

¹Ch. Wheatstone, englischer Physiker, 1802-1875

**Bild 11.13:** Kraftmessung am Biegebalken

Bei Dehnung erhält man:

$$\Delta R_2 = \Delta R_3 = \Delta R, \quad (11.40)$$

bei Stauchung gilt:

$$\Delta R_1 = \Delta R_4 = -\Delta R. \quad (11.41)$$

Damit folgt aus Gleichung (11.40) bzw. Gleichung (11.41):

$$U_d \approx \frac{U_0}{4} \left(\frac{4 \Delta R}{R} \right) \approx k U_0 \epsilon. \quad (11.42)$$

Somit erhält man für den Biegebalken:

$$\epsilon = \frac{U_d}{k U_0}, \quad (11.43)$$

mit Gleichung (11.35) ergibt sich für die Kraft:

$$F = \frac{U_d}{U_0 k} E A. \quad (11.44)$$

Ein weiterer Vorteil dieser Anordnung bzw. der Wheatstone'schen Vollbrücke im Allgemeinen ist, dass sich Widerstandsänderungen aufgrund von Temperaturschwankungen ΔT vollständig (theoretisch) ausgleichen, da sich alle Widerstände gleich erwärmen:

$$\Delta R_i = \Delta R(\Delta T). \quad (11.45)$$

Somit ergibt sich nach Gleichung (11.39):

$$U_m(\Delta T) = 0, \quad (11.46)$$

d. h. Widerstandsänderungen aufgrund von Temperaturschwankungen haben keinen Einfluss auf die Messspannung U_d .

11.4.2 Messung eines Moments

Die Messung von Momenten erfolgt analog zur Kraftmessung. In der Regel werden an einem Vollstab oder an einem Rohr vier Dehnungsmessstreifen im Winkel von 45° zur Aufnehmerachse angeordnet. Die Dehnung berechnet sich in diesem Fall folgendermaßen:

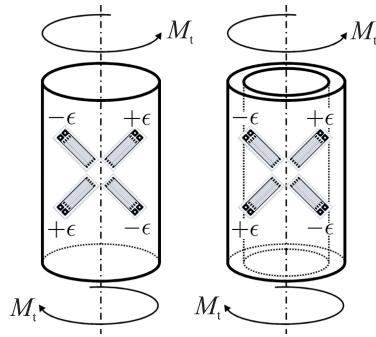


Bild 11.14: Verformungskörper zur Momentenmessung (links: Vollstab, rechts: Rohr)

Vollstab:

$$\epsilon = \frac{M_t}{\pi G r^3}. \quad (11.47)$$

Rohr:

$$\epsilon = \frac{M_t 8D}{\pi G (D^4 - d^4)}, \quad (11.48)$$

mit M_t : Torsionsmoment, G : Schubmodul, D : Außendurchmesser, d : Innendurchmesser. Das weitere Vorgehen ist analog zu Abschnitt 11.4.1 und für den Vollstab nochmals ausführlich dargestellt:

Die Widerstände R_2 und R_3 erfahren eine Dehnung $+ \epsilon$ die Widerstände R_1 und R_4 eine Stauchung $- \epsilon$. Im unbelasteten Zustand gelte:

$$R_1 = R_2 = R_3 = R_4 = R. \quad (11.49)$$

Somit gilt für die Messspannung nach Gleichung (11.42):

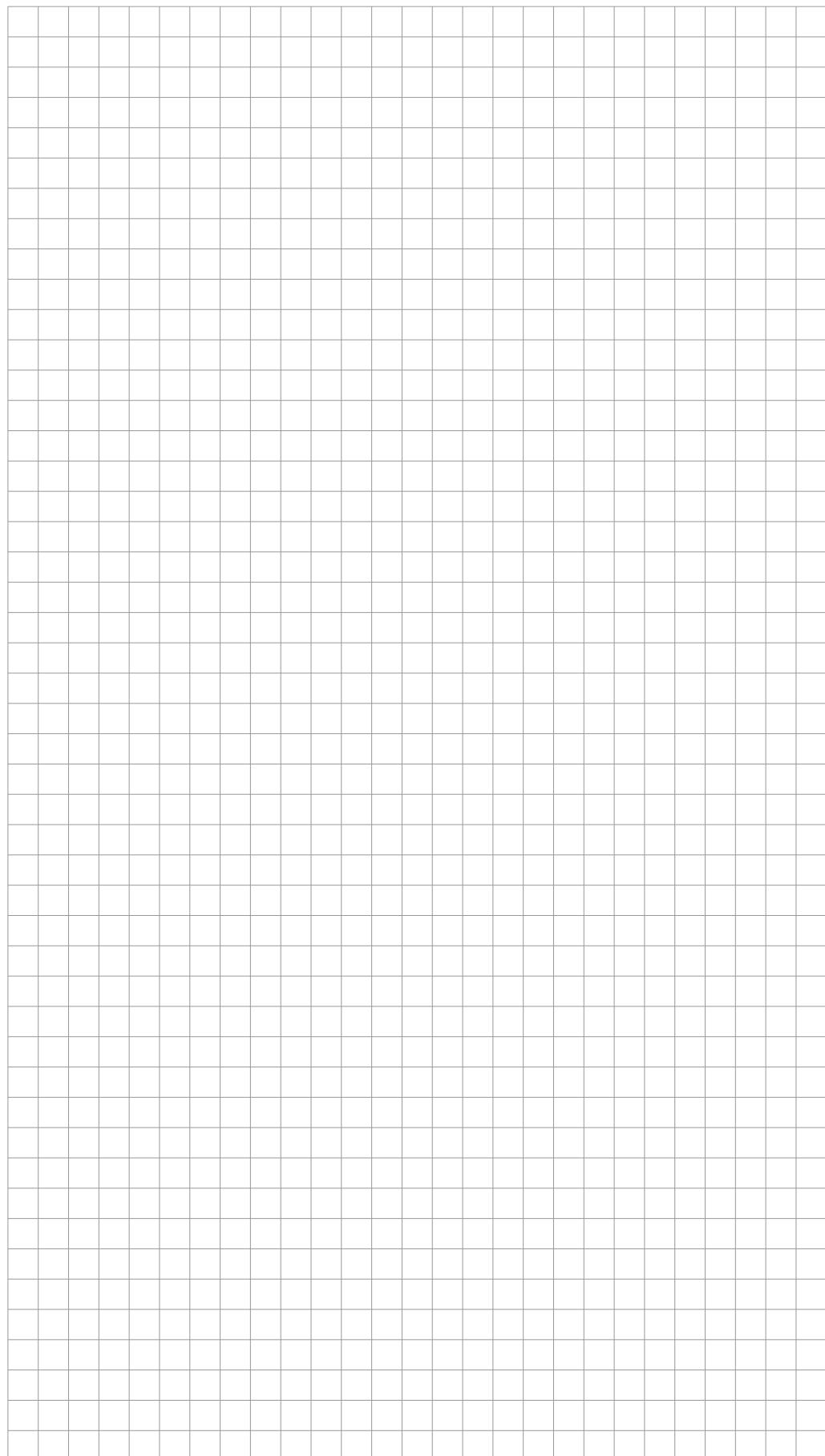
$$U_d = k U_0 \epsilon. \quad (11.50)$$

Kombiniert man Gleichung (11.50) mit der Dehnung des Vollstabes nach Gleichung (11.47), so ergibt sich abschließend:

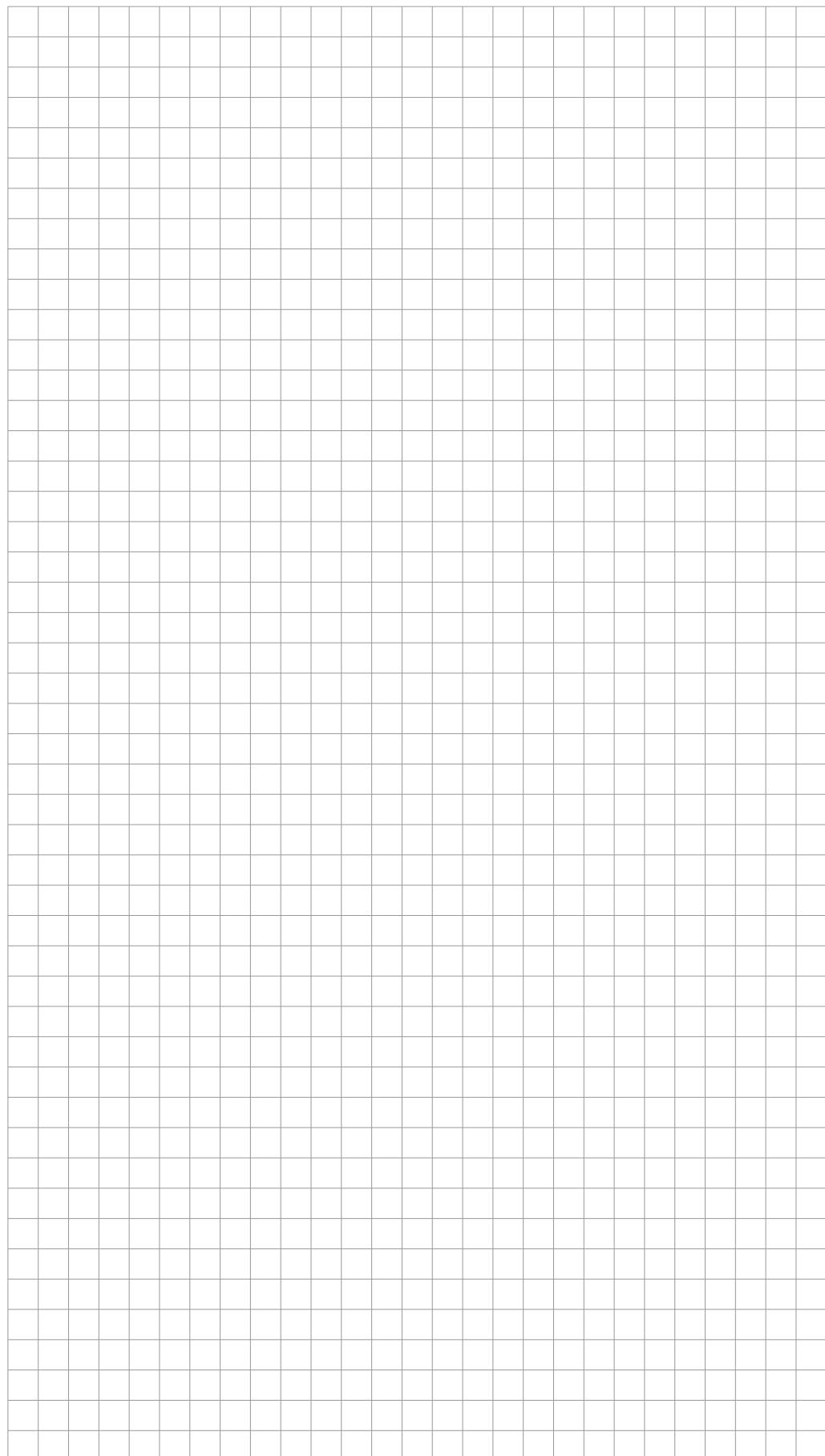
$$M_t = \frac{U_d}{k U_0} \pi G r^3. \quad (11.51)$$

Fragen zur Selbstkontrolle

1. Nennen Sie vier typische Kenngrößen für Sensoren und erläutern Sie die einzelnen Begriffe!
2. Woraus setzt sich eine inertiale Messeinheit zusammen, erläutern Sie die Funktionsprinzipien der implementierten Sensoren?
3. Erläutern Sie das Prinzip der Messung einer Kraft mittels Kraft-Moment-Sensor!







Literatur

- [AP19] T. Asfour und F. Paus. *Robotik I - Einführung in die Robotik, Vorlesung 14 (Grundlagen Bewegungsplanung)*. Karlsruher Institut für Technologie (KIT), 2019.
- [AS+07] A. Albu-Schäffer u. a. „The DLR Lightweight Robot – Design and Control Concepts for Robots in Human Environments“. In: *Industrial Robot: An International Journal* 34.5 (2007), S. 376–385.
- [Ang06] J. Angeles. „Is there a Characteristic Length of a Rigid-Body Displacement“. In: *Mechanism & Machine Theory* 41.8 (2006), S. 884–896.
- [BM08] L. Biagiotti und C. Melchiorri. *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.
- [CC00] D. Constantinescu und E. A. Croft. „Smooth and time-optimal trajectory planning for industrial manipulators along specified paths“. In: *Journal of Robotic Systems* 17.5 (2000), S. 233–249.
- [Cra04] J.J. Craig. *Introduction to Robotics: Mechanics and Control*. New Jersey, USA: Prentice Hall, 2004.
- [DH91] R. Dillmann und M. Huck. *Planungssysteme*. Springer, 1991.
- [DIN99] DIN EN ISO 9283. *Industrieroboter – Leistungskenngrößen und zugehörige Prüfmethoden*. Ausgabedatum 1999-05. 1999.
- [DKL98] E.B. Dam, M. Koch und M. Lillholm. *Quaternions, Interpolation and Animation*. Techn. Ber. Kopenhagen, Dänemark: Department of Computer Science, University of Copenhagen, 1998.
- [Die06] J. Diebel. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*. Techn. Ber. Stanford, USA: Stanford University, 2006.
- [Ebe10] D. Eberly. *Quaternion Algebra and Calculus*. Techn. Ber. Washington, USA: Geometric Tools, LLC, 2010.
- [HD04] M. Horn und N. Douroumas. *Regelungstechnik*. München, Deutschland: Pearson Studium, 2004.
- [HGP06] B. Heimann, W. Gerth und K. Popp. *Mechatronik: Komponenten – Methoden – Beispiele*. München, Deutschland: Carl Hanser Verlag, 2006.
- [HS14] S. Hesse und G. Schnell. *Sensoren für die Prozess- und Fabrikautomation: Funktion – Ausführung – Anwendung*. 6. Aufl. Wiesbaden, Deutschland: Vieweg+Teubner Verlag, 2014.
- [Hir+02] G. Hirzinger u. a. „DLR’s Torque-Controlled Light Weight Robot III – are we Reaching the Technological Limits now?“ In: *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*. Washington, USA, 2002, S. 1710–1716.
- [KD02] W. Khalil und E. Dombre. *Modeling, Identification & Control of Robots*. New York, USA: Routledge, 2002.
- [KK86] W. Khalil und J. Kleinfinger. „A new Geometric Notation for Open and Closed-Loop Robots“. In: *Proc. of the 1986 IEEE International Conference on Robotics and Automation*. San Francisco, USA, 1986, S. 1174–1179.

- [KTK90] H.-D. Karl, M. Tinebor und J. Krieteimeier. *EU Patentanmeldung (EP0369119) – Verfahren und Schaltungsanordnung zur Erzeugung von digitalen Drehzahl- und Drehwinkelinformationen mittels eines Funktionsdrehmelders*. EP0369119. Europäische Patentanmeldung. 1990.
- [Kes97] S. Kessel. „Skriptum zur Vorlesung Mechanik III für Studierende des Maschinenbaus“. Magisterarb. Dortmund, Deutschland: Lehrstuhl für Mechanik, Universität Dortmund, 1997.
- [Kie07] E. Kiel. *Antriebslösungen: Mechatronik für Produktion und Logistik*. Heidelberg, Deutschland: Springer, 2007.
- [Kre08] V.E. Kremer. *Quaternions and SLERP*. Techn. Ber. Saarbrücken, Deutschland: Department for Computer Science, University of Saarbrücken, 2008.
- [LaV06] S.M. LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [MV03] K. Meyberg und P. Vachenauer. *Höhere Mathematik 1: Differential- und Integralrechnung Vektor- und Matrizenrechnung*. Berlin, Deutschland: Springer-Verlag, 2003.
- [N.N04] N.N. *Synchro and Resolver Engineering Handbook*. Techn. Ber. Blacksburg, USA: MOOG Components Group, 2004.
- [PR87] F. Pfeiffer und E. Reithmeier. *Roboterdynamik*. Stuttgart, Deutschland: Teubner Verlag, 1987.
- [SS00] L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Berlin, Deutschland: Springer, 2000.
- [SSS98] L. Stocco, S. Salcudean und F. Sassani. „Matrix Normalization for Optimal Robot Design“. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 1998, S. 1346–1351.
- [SV96] Rolf-Dieter Schraft und Hansjörg Volz. *Serviceroboter: Innovative Technik in Dienstleistung und Versorgung*. Berlin, Deutschland: Springer, 1996.
- [Sal80] J.K. Salisbury. „Active Stiffness Control of a Manipulator in Cartesian Coordinates“. In: *Proceedings of the IEEE Conference on Decision and Control*. Albuquerque, USA, 1980, S. 95–100.
- [Sch07] E. Schrüfer. *Elektrische Messtechnik*. München, Deutschland: Carl Hanser Verlag, 2007.
- [VK02] Z. Vukic und O. Kuljaca. *Lectures on PID Controllers*. Techn. Ber. Zagreb, Kroatien: Faculty of Electrical Engineering und Computing, University of Zagreb, 2002.
- [ZH97] G. Zeng und A. Hemami. „An Overview of Robot Force Control“. In: *Robotica* 15.5 (1997), S. 473–482.
- [ZN42] J.G. Ziegler und N.B. Nichols. „Optimum Settings for Automatic Controllers“. In: *American Society of Mechanical Engineers – Transactions* 64.8 (1942), S. 759–765.

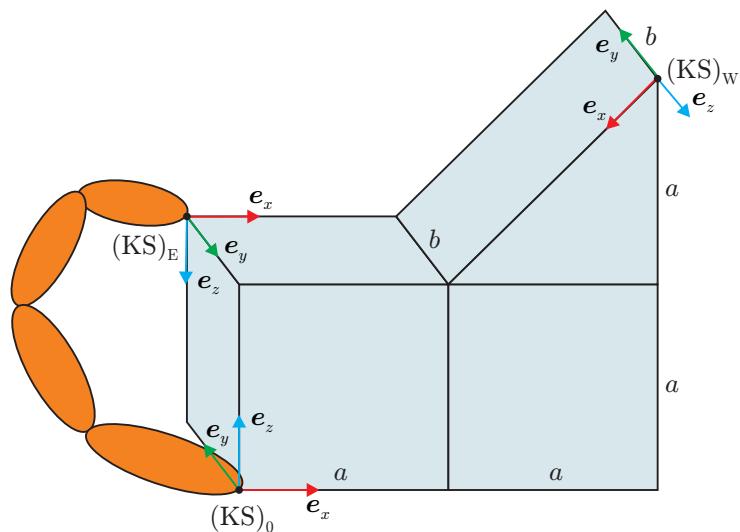
12 Übungen zur Vorlesung Robotik I

Übung 1:

Koordinatentransformation – Beispiel mit exakter Notation

Gegeben sind die dargestellten Koordinatensysteme sowie die eingezeichneten Längen.

Gegeben: a, b .

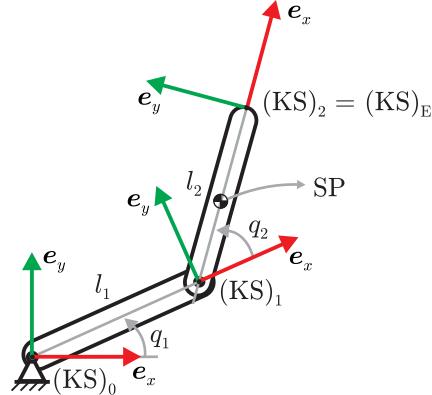


- Wie lautet der Vektor vom Endeffektor zum Werkstück im $(KS)_E$ und im $(KS)_0$?
- Wie lautet die homogene Transformation von $(KS)_W$ ins $(KS)_E$?
- Wo liegt der Mittelpunkt der schrägen Fläche bezüglich des Endeffektors im $(KS)_E$?

Übung 2:**Anwendung Koordinatentransformation auf einen planaren RR-Roboter**

Gegeben ist der abgebildete RR-Roboter mit zwei Drehantrieben sowie die eingezeichneten Größen.

Gegeben: l_1, l_2, q_1, q_2 .



- Bestimmen Sie die Schwerpunktslage von Glied 2 im Inertialkoordinatensystem $(KS)_0$ unter Verwendung homogener Transformationsmatrizen!
- Bestimmen Sie aufbauend auf dem vorangegangenen Aufgabenteil die Lage des Endeffektors (Position und Orientierung) im Inertialkoordinatensystem $(KS)_0$!

Einfache direkte Kinematik (inverser Zusammenhang)

Gegeben ist die Endeffektorlage eines Roboters mit drei (Gelenk-)Freiheitsgraden in Form einer homogenen Transformationsmatrix 0T_E :

$${}^0T_E = \left(\begin{array}{ccc|c} c_1 c_{23} & -s_1 & -c_1 s_{23} & c_1 (l_2 c_2 + l_3 c_{23}) \\ s_1 c_{23} & c_1 & -s_1 s_{23} & s_1 (l_2 c_2 + l_3 c_{23}) \\ s_{23} & 0 & c_{23} & l_1 + l_2 s_2 + l_3 s_{23} \\ \hline 0 & 0 & 0 & 1 \end{array} \right).$$

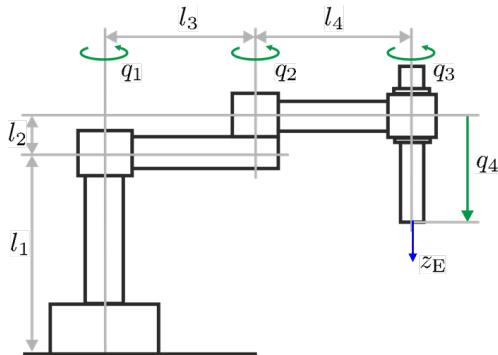
Bestimmen Sie unter Verwendung der gegebenen Transformationsmatrix die direkte Kinematik $\boldsymbol{x}_E = f(\boldsymbol{q})$ des gegebenen Roboters! Ermitteln Sie die Orientierung basierend auf Kardan-Winkeln (zusammengesetzte Drehungen)!

$$\text{Gegeben: } \boldsymbol{R}_{\text{KARD}} = \boldsymbol{R}_x(\alpha) \boldsymbol{R}_y(\beta) \boldsymbol{R}_z(\gamma) = \left(\begin{array}{ccc} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -s_\alpha c_\beta \\ s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma & s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta \end{array} \right).$$

Übung 3:**Denavit-Hartenberg Notation**

Gegeben ist der abgebildete SCARA RRRP-Roboter mit drei Dreh- und einem Schubgelenk. Die eingezeichneten Größen sind als bekannt vorauszusetzen.

Gegeben: $l_1, l_2, l_3, l_4, q_1, q_2, q_3, q_4$.



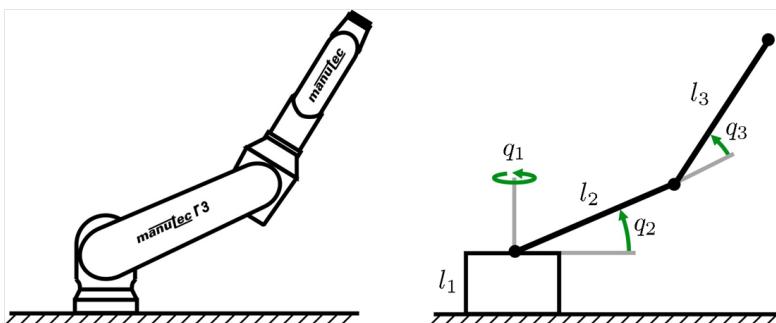
Bestimmen Sie die Denavit-Hartenberg Parameter des abgebildeten Roboters mit folgenden Vorgaben:

- Basiskoordinatensystem $(KS)_0$ im Fußpunkt des Roboters
- Endeffektorkoordinatensystem $(KS)_E$ an der „Armspitze“, z -Achse wie vorgegeben

Anwendung der Denavit-Hartenberg Notation auf einen RRR-Roboter

Gegeben ist der abgebildete RRR-Roboter mit drei Drehgelenken. Die eingezeichneten Größen sind als bekannt vorauszusetzen.

Gegeben: $l_1, l_2, l_3, q_1, q_2, q_3$.



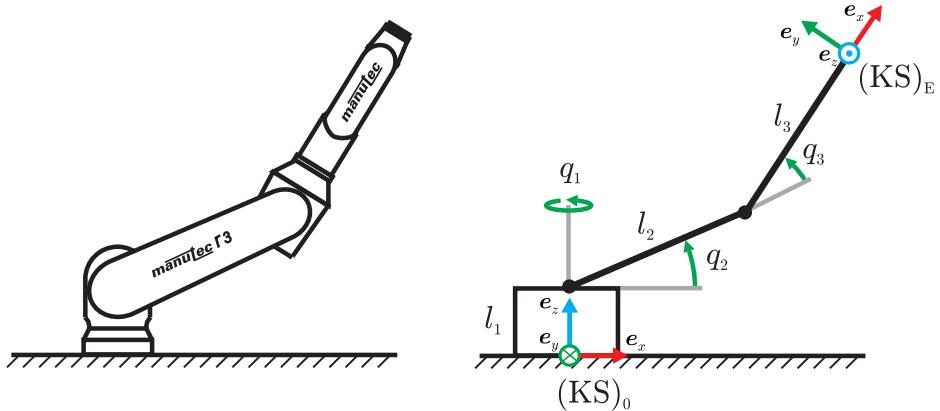
Bestimmen Sie die homogene Transformation 0T_3 mit Hilfe der Denavit-Hartenberg Parameter des abgebildeten Roboters mit folgenden Vorgaben:

- Basiskoordinatensystem $(KS)_0$ im Fußpunkt des Roboters
- Endeffektorkoordinatensystem $(KS)_E$ an der „Armspitze“

Übung 4:**Inverse Kinematik**

Gegeben ist der abgebildete RRR-Roboter mit drei Drehgelenken.

Gegeben: $l_1, l_2, l_3, (0)\boldsymbol{x}_E = ((0)x_E, (0)y_E, (0)z_E)^T$.

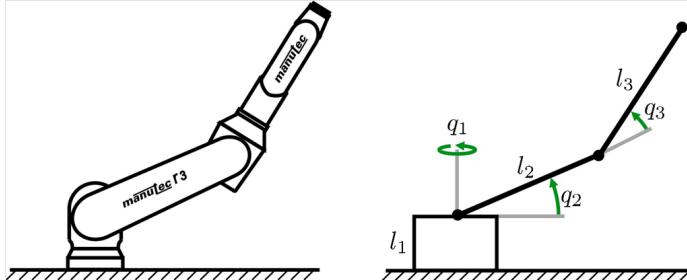


Bestimmen Sie die inverse Kinematik des abgebildeten Roboters als Funktion der translatorischen Endeffektorfreiheitsgrade (Roboterspitze) unter der Annahme, dass sich das Basiskoordinatensystem $(KS)_0$ wie dargestellt im Fußpunkt des Roboters befindet.

Übung 5:**Jacobi-Matrix – Anwendung auf RRR-Roboter**

Gegeben ist der abgebildete RRR-Roboter mit drei Drehantrieben.

Gegeben: l_1, l_2, l_3 .



Bestimmen Sie die **analytische** Jacobi-Matrix $\mathbf{J}(\mathbf{q})$ als Funktion der Gelenkwinkel \mathbf{q} unter Verwendung der direkten Kinematik:

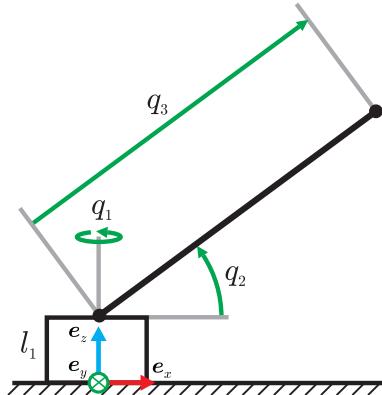
$$\begin{pmatrix} (0)x_E \\ (0)y_E \\ (0)z_E \end{pmatrix} = \mathbf{f}(\mathbf{q}) = \begin{pmatrix} c_1(l_2 c_2 + l_3 c_{23}) \\ s_1(l_2 c_2 + l_3 c_{23}) \\ l_1 + l_2 s_2 + l_3 s_{23} \end{pmatrix}.$$

Beschränken Sie sich bei der Berechnung auf den Zusammenhang von Gelenkgeschwindigkeiten und den translatorischen Endeffektorgeschwindigkeiten (ohne Rotationen).

Jacobi-Matrix – Anwendung auf RRP-Roboter

Gegeben ist der abgebildete RRP-Roboter mit zwei Drehantrieben und einem Linearantrieb.

Gegeben: l_1 .

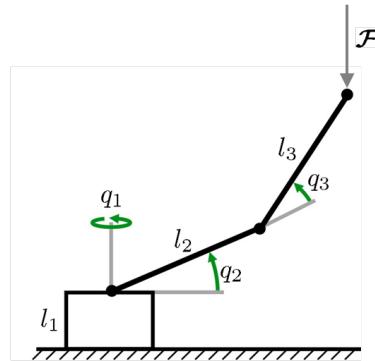


Bestimmen Sie die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ als Funktion der Gelenkwinkel \mathbf{q} für den abgebildeten Roboter! Nehmen Sie dabei an, dass sich das Basiskoordinatensystem $(KS)_0$, wie dargestellt, im Fußpunkt des Roboters befindet.

Übung 6:**Statisches Kraft- und Momentenmodell – Auswirkungen einer Gewichtskraft am Endeffektor**

Auf einen RRR-Roboter (drei rotatorische Freiheitsgrade) mit den Gelenksteifigkeiten $\mathbf{K}_q = \text{diag}(\mathbf{k})$, mit $\mathbf{k} = (k_1, k_2, k_3)^T$ wird in einer beliebigen Ruhelage \mathbf{q} am Endeffektor eine Gewichtskraft $\mathcal{F} = (0, 0, -F_G)^T$ aufgebracht.

Gegeben: $l_1, l_2, l_3, k_1, k_2, k_3, \mathbf{q}, F_G, \mathbf{J}(\mathbf{q}) = \begin{pmatrix} -s_1(l_2 c_2 + l_3 c_{23}) & -c_1(l_2 s_2 + l_3 s_{23}) & -l_3 c_1 s_{23} \\ c_1(l_2 c_2 + l_3 c_{23}) & -s_1(l_2 s_2 + l_3 s_{23}) & -l_3 s_1 s_{23} \\ 0 & l_2 c_2 + l_3 c_{23} & l_3 c_{23} \end{pmatrix}$.

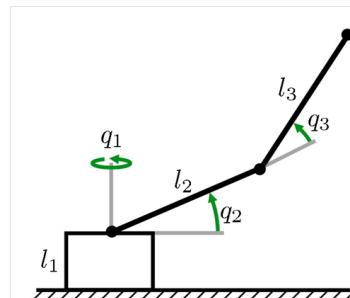


- Welche (zusätzlichen) Gelenkmomente sind notwendig, um den Roboter im Gleichgewicht zu halten?
- Bestimmen Sie die resultierende Änderung der Gelenkwinkel Δq !
- Bestimmen Sie die resultierende Auslenkung des Endeffektors $\Delta \mathbf{x}_E$!

Statisches Kraft- und Momentenmodell – Nachgiebigkeit in einem bestimmten Arbeitspunkt

Ein RRR-Roboter mit den Gelenksteifigkeiten $\mathbf{K}_q = \text{diag}(\mathbf{k})$, mit $\mathbf{k} = (k_1, k_2, k_3)^T$ befindet sich in einer bestimmten Ruhelage \mathbf{q}_R .

Gegeben: $l_i = k_i = 1, \mathbf{q}_R = \begin{pmatrix} 0 \\ \frac{\pi}{2} \\ -\frac{\pi}{2} \end{pmatrix}, \mathbf{J}(\mathbf{q}) = \begin{pmatrix} -s_1(l_2 c_2 + l_3 c_{23}) & -c_1(l_2 s_2 + l_3 s_{23}) & -l_3 c_1 s_{23} \\ c_1(l_2 c_2 + l_3 c_{23}) & -s_1(l_2 s_2 + l_3 s_{23}) & -l_3 s_1 s_{23} \\ 0 & l_2 c_2 + l_3 c_{23} & l_3 c_{23} \end{pmatrix}$.

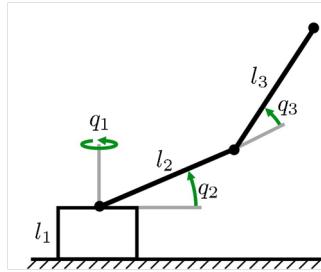


- Bestimmen Sie die Nachgiebigkeitsmatrix $\mathbf{C}_x(\mathbf{q}_R)$!
- In welchen Raumrichtungen ist der Roboter am „weichsten“ bzw. am „steifsten“?

Übung 7:**Singularitäten eines RRR-Roboters**

Gegeben ist der abgebildete RRR-Roboter mit insgesamt drei Drehantrieben. Die (Segment-)Längen seien gegeben.

Gegeben: $l_1 = l_2 = l_3 = 1$, $\mathbf{J}(\mathbf{q}) = \begin{pmatrix} -s_1(l_2 c_2 + l_3 c_{23}) & -c_1(l_2 s_2 + l_3 s_{23}) & -l_3 c_1 s_{23} \\ c_1(l_2 c_2 + l_3 c_{23}) & -s_1(l_2 s_2 + l_3 s_{23}) & -l_3 s_1 s_{23} \\ 0 & l_2 c_2 + l_3 c_{23} & l_3 c_{23} \end{pmatrix}$.

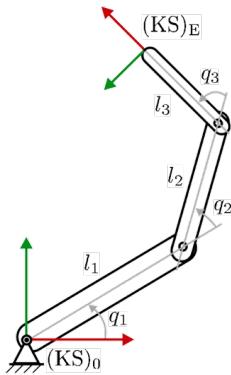


- Bestimmen Sie unter Verwendung der gegebenen Jacobi-Matrix die Singularitäten des abgebildeten RRR-Roboters!
- Skizzieren Sie die singulären Konfigurationen!

Nullraumbewegung

Die Konfiguration eines kinematisch redundanten RRR-Roboters mit zwei translatorischen Endeffektorfreiheitsgraden (Orientierung wird nicht betrachtet) soll bei einer konstanten Endeffektorposition optimiert werden. Hierzu wird eine Nullraumbewegung durchgeführt.

Gegeben: $l_1 = l_2 = l_3 = 1$, $\mathbf{J}(\mathbf{q}) = \begin{pmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{pmatrix}$.



Bestimmen Sie die homogene Lösung $\dot{\mathbf{q}}_h$, die sich dadurch auszeichnet, dass aus einer Änderung der Gelenkwinkel keine Endeffektorbewegung resultiert!

Übung 8:**Bahnplanung**

Eine Roboterachse bewegt sich mit einem typischen Geschwindigkeitstrapezprofil in der Zeit t_e von einer Startposition $q_S = q_1$ zu einer Zielposition $q_Z = q_2$. Die Roboterachse wird dabei aus der Ruhe heraus mit einer Beschleunigung $a < \ddot{q}_{\max}$ bis zum Erreichen ihrer maximalen Geschwindigkeit \dot{q}_{\max} beschleunigt bzw. mit einer Beschleunigung $-a$ bis zum Stillstand abgebremst.

Gegeben: $q_1, q_2, \dot{q}(0) = \dot{q}(t_e) = 0, a, \dot{q}_{\max} = v_{\max}$.

- a) Zeichnen Sie qualitativ den Beschleunigungs-, Geschwindigkeits- und Positionsverlauf $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$!
- b) Bestimmen Sie die Gesamtzeit t_e , die zum Abfahren der Trajektorie nötig ist!

Die Roboterachse soll nun ein Beschleunigungstrapezprofil abfahren. Die Voraussetzungen sind identisch zu Aufgabenteil a) und b).

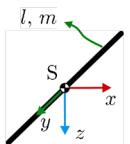
- c) Zeichnen Sie den neuen Beschleunigungs-, Geschwindigkeits- und Positionsverlauf der Bahnkoordinate $s(t), \dot{s}(t), \ddot{s}(t)$! Wie unterscheidet sich das Beschleunigungstrapezprofil vom Geschwindigkeitstrapezprofil?

Übung 9:**Dynamik (Newton-Euler) – Bewegungsgleichung eines RR-Roboters**

Bestimmen Sie unter Verwendung der Newton-Euler-Methode die Bewegungsgleichungen (inverse Dynamik) des abgebildeten RR-Roboters mit insgesamt zwei Drehgelenken! Nehmen Sie an, dass es sich bei dem Roboter um ein starres, reibungsfreies System handelt. Die Roboterglieder sind als massebehaftete homogene Stäbe zu modellieren.

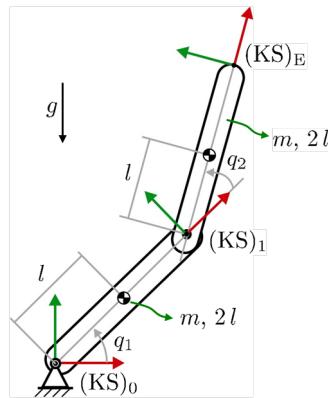
Gegeben: l, m .

Formelsammlung:



$$J_{xx}^{(S)} = J_{zz}^{(S)} = \frac{1}{12} ml^2$$

$$J_{yy}^{(S)} = 0$$

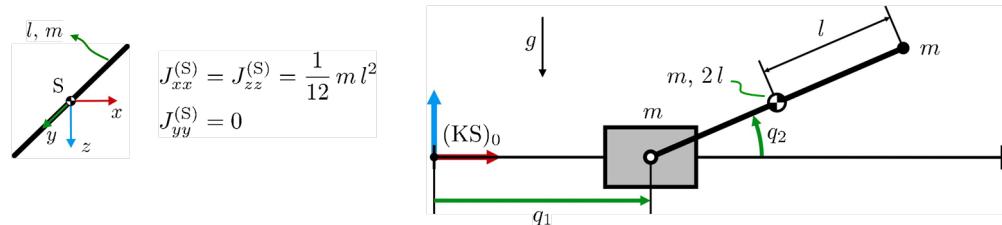


Übung 10:**Dynamik (Lagrange) – Bewegungsgleichung eines PR-Roboters**

Bestimmen Sie die Bewegungsgleichungen nach Lagrange (inverse Dynamik) des abgebildeten PR-Roboters mit einem Schub- und einem Drehgelenk! Nehmen Sie an, dass es sich bei dem Roboter um ein starres, reibungsfreies System handelt. Der Roboterarm ist als massebehafteter homogener Stab zu modellieren. Der Linearschlitten und der Endeffektor sind als Punktmassen zu betrachten.

Gegeben: l, m .

Formelsammlung:

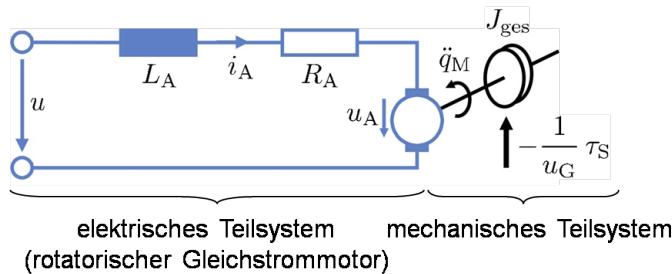


Übung 11:

Einzelachsregelung im Jointspace

Die Lage einer einzelnen starren und spielfreien Roboterachse mit rotatorischem Antrieb (auf Motorachse bezogenes Massenträgheitsmoment J_{ges} , Dämpfungskonstante d_{eff} , Getriebeübersetzung u_G , Drehmomentkonstante k_M) soll geregelt werden.

Gegeben: $J_{\text{ges}}, d_{\text{eff}}, u_G \gg 1, k_M, t_e \ll t_m$.



Untersuchen Sie die bleibende Regelabweichung des Gesamtsystems bei einem Führungs- und Störgrößensprung mit der Amplitude 1 für eine P/P-Kaskadenregelung!

Nehmen Sie an, dass die elektrische Zeitkonstante t_e sehr viel kleiner ist als die mechanische Zeitkonstante t_m .

Feedback-Regelung

Gegeben ist ein Roboter mit $n = 2$ Gelenken. Sein dynamisches Modell lautet

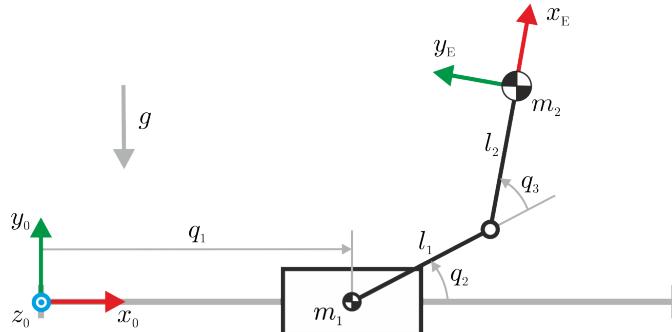
$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

mit $\mathbf{M}(\mathbf{q})$, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{g}(\mathbf{q})$ und $\boldsymbol{\tau}$ der Massematrix, den Coriolis- und Zentrifugaltermen, dem Gravitationsanteil sowie den Gelenkmomenten. Das Modell ist hinreichend genau bekannt und $\mathbf{M}(\mathbf{q})$, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ sowie $\mathbf{g}(\mathbf{q})$ können als bekannt vorausgesetzt werden. Der Roboter soll sich für $t \rightarrow \infty$ einer gewünschten Trajektorie $\mathbf{q}_s(t), \dot{\mathbf{q}}_s(t), \ddot{\mathbf{q}}_s(t)$ mit vorgegebener Dynamik annähern.

- Geben Sie das Regelgesetz für $\boldsymbol{\tau}$ für eine Feedback-Regelung mit PD-Regler sowie die resultierende Differentialgleichung der Regeldifferenz $e(t) = \mathbf{q}_s(t) - \mathbf{q}(t)$ in Abhängigkeit der Reglerparameter $\mathbf{K}_P = \text{diag}(k_{P1}, k_{P2})$ und $\mathbf{K}_D = \text{diag}(k_{D1}, k_{D2})$ an!
- Berechnen Sie die Reglerparameter, wenn die Dynamik des Regelfehlers $e_1(t)$ die Eigenwerte $\lambda_{11} = \lambda_{12} = -1$ und die des Regelfehlers $e_2(t)$ die Eigenwerte $\lambda_{21} = -1 - j$, $\lambda_{22} = -1 + j$ aufweisen sollen.
- Ist der geschlossene Regelkreis stabil?
- Treten im geschlossenen Regelkreis Schwingungen auf?

Übung 12:**Musterklausur (Aufgabe 1)**

Gegeben ist der abgebildete PRR-Portalroboter mit den Gelenkvariablen q_1 , q_2 und q_3 .



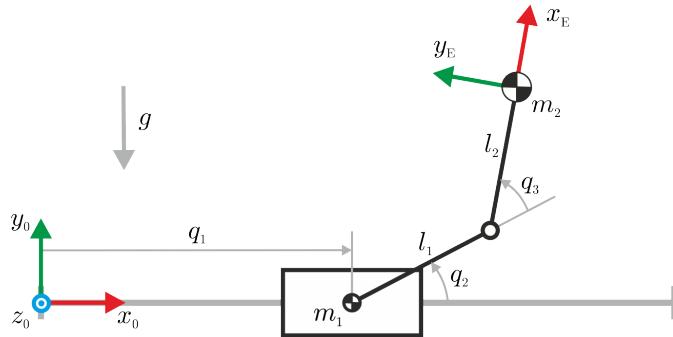
Der Schlitten 1 (Masse m_1) sowie der Endeffektor (Masse m_2) sind als Punktmassen zu betrachten. Alle anderen Teile sind masselos anzunehmen. Die Länge des ersten am Schlitten befestigten Armgliedes ist l_1 . Das zweite Armglied hat die Länge l_2 . Das reibungsfreie und starre System unterliegt wie skizziert der Erdbeschleunigung g . Die Umweltkoordinaten $(0)x_E$ und $(0)y_E$ bestimmen die Position des Endeffektors im Inertialkoordinatensystem $(KS)_0$. Die Orientierung soll unberücksichtigt bleiben und ist demnach nicht als Freiheitsgrad des Endeffektors zu betrachten.

Gegeben: $l_1, l_2, m_1, m_2, q_{1\max} > 2(l_1 + l_2), g$.

- Skizzieren Sie den Arbeitsraum für $0 \leq q_1 \leq q_{1\max}$, $0 \leq q_2 \leq \pi$ und $0 \leq q_3 \leq 2\pi$!
Nehmen Sie dazu die Abmaße des Schlittens zu null an! Welches sind die minimalen und maximalen Werte, die $(0)x_E$ und $(0)y_E$ annehmen können?
- Bestimmen Sie die direkte Kinematik!
- Kann die inverse Kinematik analytisch berechnet werden? Begründen Sie Ihre Aussage!
- Bestimmen Sie die analytische Jacobi-Matrix!
- Bestimmen Sie die Lagrange'sche Funktion L für das System als Funktion der Gelenkvariablen q_i und ihrer zeitlichen Ableitungen!
- Wie lautet die Gleichung zur Bestimmung der Bewegungsgleichungen unter Verwendung der Lagrange'schen Funktion L ?
- Bestimmen Sie die Kraft τ_1 für das erste Gelenk!

Musterklausur (Aufgabe 1 als exemplarische Single-Choice-Aufgabe)

Gegeben ist der abgebildete PRR-Portalroboter mit den Gelenkvariablen q_1 , q_2 und q_3 .



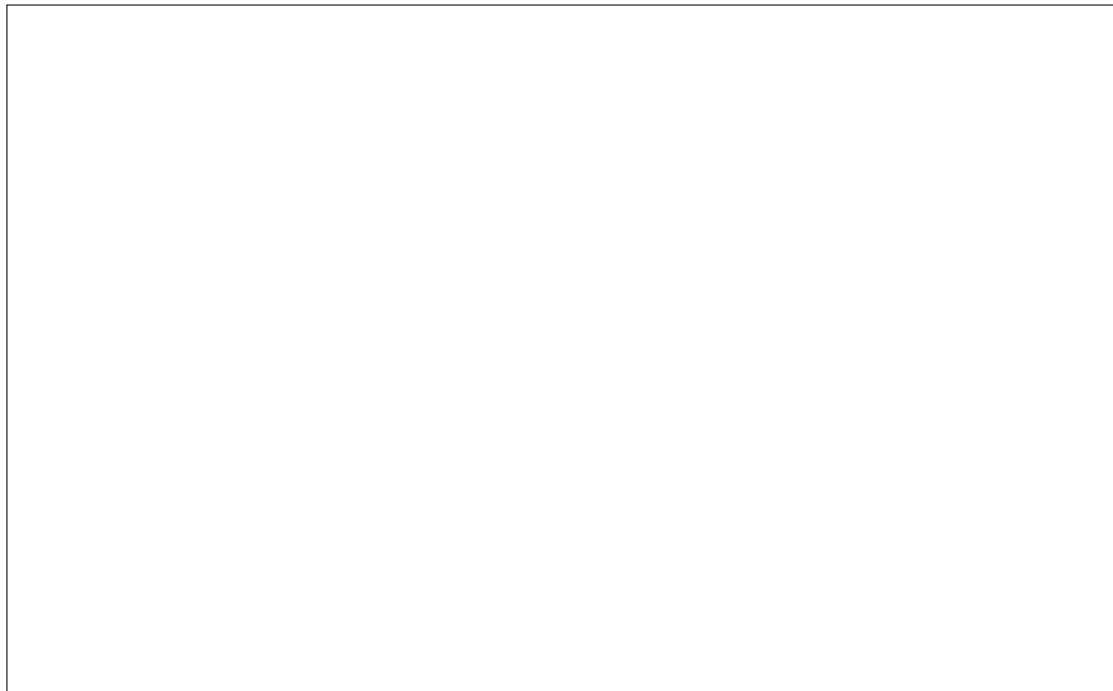
Der Schlitten 1 (Masse m_1) sowie der Endeffektor (Masse m_2) sind als Punktmassen zu betrachten. Alle anderen Teile sind masselos anzunehmen. Die Länge des ersten am Schlitten befestigten Armgliedes ist l_1 . Das zweite Armglied hat die Länge l_2 . Das reibungsfreie und starre System unterliegt wie skizziert der Erdbeschleunigung g . Die Umweltkoordinaten $(0)x_E$ und $(0)y_E$ bestimmen die Position des Endeffektors im Inertialkoordinatensystem $(KS)_0$. Die Orientierung soll unberücksichtigt bleiben und ist demnach nicht als Freiheitsgrad des Endeffektors zu betrachten.

Gegeben: $l_1, l_2, m_1, m_2, q_{1\max} > 2(l_1 + l_2), g$.

Hinweis: $c_1 = \cos(q_1)$, $s_1 = \sin(q_1)$, $c_{12} = \cos(q_1 + q_2)$ und $s_{12} = \sin(q_1 + q_2)$.

Die Gelenkkraumgrenzen sind wie folgt anzunehmen: $0 \leq q_1 \leq q_{1\max}$, $0 \leq q_2 \leq \pi$ und $0 \leq q_3 \leq 2\pi$

a.1) Skizzieren Sie den Arbeitsraum. Nehmen Sie dazu die Abmaße des Schlittens zu null an.



a.2) Was sind die minimalen und maximalen Werte die $(0)x_E$ für den abgebildeten PRR-Roboter annehmen kann?

- $-q_{1\max} - l_1 \leq (0)x_E \leq q_{1\max} + l_1$
- $-(l_1 + l_2) \leq (0)x_E \leq (l_1 + l_2)$
- $-q_{1\max} \leq (0)x_E \leq q_{1\max}$
- $-l_2 \leq (0)x_E \leq l_2$
- $0 \leq (0)x_E \leq q_{1\max} + l_1 + l_2$
- $-(l_1 - l_2) \leq (0)x_E \leq (l_1 - l_2)$
- Keine der Antworten ist korrekt

a.3) Was sind die minimalen und maximalen Werte die $(0)y_E$ für den abgebildeten PRR-Roboter annehmen kann?

- $-(l_1 + l_2) \leq (0)y_E \leq 0$
- $-l_2 \leq (0)y_E \leq l_1 + l_2$
- $-l_1 \leq (0)y_E \leq l_2$
- $-q_{1\max} - l_1 - l_2 \leq (0)y_E \leq l_1 + l_2$
- $-l_1 - l_2 \leq (0)y_E \leq l_1 + l_2 + q_{1\max}$
- $l_1 - l_2 \leq (0)y_E \leq l_1 + l_2$
- Keine der Antworten ist korrekt

b.1) Bestimmen Sie die für den abgebildeten PRR-Roboter die direkte Kinematik für $(0)x_E$.

- $(0)x_E = q_1 + l_1 s_2 + l_2 s_{23}$
- $(0)x_E = q_1 + l_1 c_2 + l_2 c_{23}$
- $(0)x_E = l_1 s_2 + l_2 s_{23}$
- $(0)x_E = l_1 c_2 + l_2 c_{23}$
- Keine der Antworten ist korrekt.

b.2) Bestimmen Sie die für den abgebildeten PRR-Roboter die direkte Kinematik für $(0)y_E$.

- $(0)y_E = q_1 + l_1 s_2 + l_2 s_{23}$
- $(0)y_E = q_1 + l_1 c_2 + l_2 c_{23}$
- $(0)y_E = l_1 s_1 + l_2 s_{12}$
- $(0)y_E = l_1 c_1 + l_2 c_{12}$
- Keine der Antworten ist korrekt.

c.1) Gibt es mehrere Lösungen der inversen Kinematik?

- Nein, es gibt keine Lösung für die inverse Kinematik.
- Ja, da $\dim(\mathbf{x}_E) - \dim(\mathbf{q}) < 0$.
- Nein, da $\dim(\mathbf{q}) < \dim(\mathbf{x}_E)$.
- Nein, da es am Endeffektor zwei betrachtete Freiheitsgrade gibt.
- Keine der Antworten ist korrekt.

d.1) Gegeben sei die direkte Kinematik eines RRR-Roboters

$$\begin{pmatrix} {}^{(0)}x_E \\ {}^{(0)}y_E \\ {}^{(0)}z_E \end{pmatrix} = \mathbf{f}(\mathbf{q}) = \begin{pmatrix} c_1(l_2 c_2 + l_3 c_{23}) \\ s_1(l_2 c_2 + l_3 c_{23}) \\ l_1 + l_2 s_2 + l_3 s_{23} \end{pmatrix}.$$

Bestimmen Sie die **analytische** Jacobi-Matrix $\mathbf{J}(\mathbf{q})$ als Funktion der Gelenkwinkel \mathbf{q} . Beschränken Sie sich bei der Berechnung auf den Zusammenhang von Gelenkgeschwindigkeiten und den translatorischen Endeffektorgeschwindigkeiten (ohne Rotationen).

- $\mathbf{J}(\mathbf{q}) = \begin{pmatrix} -s_1(l_2 c_2 + l_3 c_{23}) & c_1(l_2 c_2 + l_3 c_{23}) & 0 \\ -c_1(l_2 s_2 + l_3 s_{23}) & -s_1(l_2 s_2 + l_3 s_{23}) & l_2 c_2 + l_3 c_{23} \\ -c_1 l_3 s_{23} & -s_1 l_3 s_{23} & l_3 c_{23} \end{pmatrix}.$
- $\mathbf{J}(\mathbf{q}) = \begin{pmatrix} s_1(l_2 c_2 + l_3 c_{23}) & c_1(l_2 s_2 + l_3 s_{23}) & c_1 l_3 s_{23} \\ -c_1(l_2 c_2 + l_3 c_{23}) & s_1(l_2 s_2 + l_3 s_{23}) & s_1 l_3 s_{23} \\ 0 & -l_2 c_2 + l_3 c_{23} & -l_3 c_{23} \end{pmatrix}.$
- $\mathbf{J}(\mathbf{q}) = \begin{pmatrix} s_1(l_2 c_2 + l_3 c_{23}) & -c_1(l_2 c_2 + l_3 c_{23}) & 0 \\ c_1(l_2 s_2 + l_3 s_{23}) & s_1(l_2 s_2 + l_3 s_{23}) & -l_2 c_2 + l_3 c_{23} \\ c_1 l_3 s_{23} & s_1 l_3 s_{23} & -l_3 c_{23} \end{pmatrix}.$
- $\mathbf{J}(\mathbf{q}) = \begin{pmatrix} -s_1(l_2 c_2 + l_3 c_{23}) & -c_1(l_2 s_2 + l_3 s_{23}) & -c_1 l_3 s_{23} \\ c_1(l_2 c_2 + l_3 c_{23}) & -s_1(l_2 s_2 + l_3 s_{23}) & -s_1 l_3 s_{23} \\ 0 & l_2 c_2 + l_3 c_{23} & l_3 c_{23} \end{pmatrix}.$
- Keine der Antworten ist korrekt.

e.1) Bestimmen Sie die potentielle Energie U für das abgebildete PRR-Portalroboter-System als Funktion der Gelenkvariablen q_i und ihrer zeitlichen Ableitungen.

- $U = m_1 g q_1 + m_2 g {}_{(0)}y_E$
- $U = \frac{1}{2}m_1 \dot{q}_1^2 + \frac{1}{2}m_2 (({}_{(0)}\dot{x}_E^2 + {}_{(0)}\dot{y}_E^2))$
- $U = (m_1 + m_2) g {}_{(0)}x_E$
- $U = \frac{1}{2}m_1 \dot{q}_1^2 + \frac{1}{2}m_2 \dot{\mathbf{x}}_E^T \dot{\mathbf{x}}_E$
- Keine der Antworten ist korrekt.

e.2) Bestimmen Sie die kinetische Energie T für das abgebildete PRR-Portalroboter-System als Funktion der Gelenkvariablen q_i und ihrer zeitlichen Ableitungen.

- $T = m_1 \dot{q}_1^2 + m_2 (({}_{(0)}\dot{x}_E^2 + {}_{(0)}\dot{y}_E^2))$
- $T = m_2 \dot{q}_1^2 + m_1 (({}_{(0)}\dot{x}_E^2 + {}_{(0)}\dot{y}_E^2))$
- $T = m_1 \dot{q}_1^2 + \frac{1}{2}m_2 (({}_{(0)}\dot{x}_E^2 + {}_{(0)}\dot{y}_E^2))$
- $T = \frac{1}{2}m_1 \dot{q}_1^2 + m_2 (({}_{(0)}\dot{x}_E^2 + {}_{(0)}\dot{y}_E^2))$
- Keine der Antworten ist korrekt.

f.1) Geben Sie die Lagrange'sche Funktion L an.

- $L = -T - U$
- $L = TU + S$
- $L = T + U$
- $L = T - U$
- $L = -T - U - S$
- Keine der Antworten ist korrekt.

f.2) Wie lautet die Gleichung zur Bestimmung der Bewegungsgleichungen unter Verwendung der Lagrange'schen Funktion L ?

- $\boldsymbol{\tau} = \frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} + \frac{\partial L}{\partial \mathbf{q}}$
- $\boldsymbol{\tau} = \frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} + \frac{\partial L}{\partial \mathbf{q}}$
- $\boldsymbol{\tau} = \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}}$
- $\boldsymbol{\tau} = \frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}}$
- Keine der Antworten ist korrekt.

g.1) Gegeben sei die Lagrange'sche Funktion eines PR-Roboters

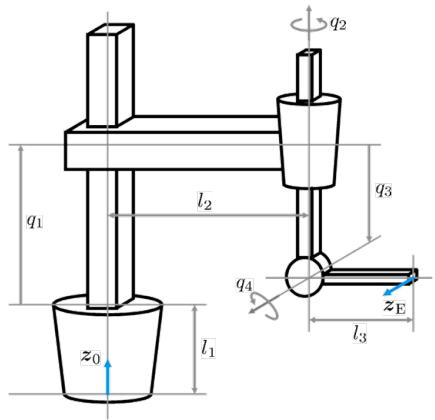
$$L = \frac{1}{6}m(9\dot{q}_1^2 - 18\dot{q}_1\dot{q}_2 l s_2 + 16\dot{q}_2^2 l^2) - 3l m g s_2.$$

Für das zweite Gelenk soll das Moment τ_2 bestimmt werden. Wie lautet dieses?

- $\tau_2 = \frac{1}{6}m(-18\ddot{q}_1 l s_2 - 18\dot{q}_1 \dot{q}_2 l c_2 + 32\ddot{q}_2 l^2 - 18\dot{q}_1 \dot{q}_2 l c_2) - 3l m g c_2$
- $\tau_2 = \frac{1}{6}m(-9\ddot{q}_1 l s_2 - 18\dot{q}_1 \dot{q}_2 l c_2 + 16\ddot{q}_2 l^2) - \frac{1}{6}m(18\dot{q}_1 \dot{q}_2 l c_2) - 3l m g c_2$
- $\tau_2 = \frac{1}{6}m(-18\ddot{q}_1 l s_2 + 32\ddot{q}_2 l^2) + 3l m g c_2$
- $\tau_2 = \frac{1}{6}m(-9\ddot{q}_1 l s_2 - 18\dot{q}_1 \dot{q}_2 l c_2 + 16\ddot{q}_2 l^2) - \frac{1}{6}m(18\dot{q}_1 \dot{q}_2 l c_2)$
- Keine der Antworten ist korrekt.

Musterklausur (Aufgabe 2)

Gegeben ist der abgebildete RPPR-Roboter mit den Gelenkvariablen q_1, q_2, q_3 und q_4 .



Die Drehrichtungen sind im mathematisch positiven Sinn um die gegebenen Achsen anzunehmen.

Gegeben: l_1, l_2, l_3 .

- Zeichnen Sie die körperfesten Koordinatensysteme sowie das Inertialkoordinatensystem $(KS)_0$ im Fußpunkt des Roboters mit Achsenbezeichnungen entsprechend der Denavit-Hartenberg Notation (DH Notation) in das gegebene Bild ein! Legen Sie den Ursprung des letzten Koordinatensystems an die Spitze des Endeffektors!
- Wie viele Möglichkeiten existieren, für die Platzierung von $(KS)_1$? Begründen Sie Ihre Aussage und zeichnen Sie ggf. eine Alternativlösung ein!
- Bestimmen Sie die DH Parameter!
- Geben Sie die allgemeine Form zur Berechnung der DH Matrix $A_i(q_i)$ in Abhängigkeit der DH Parameter an!
- Bestimmen Sie die DH Matrix $A_1(q_1)$!

Musterklausur (Aufgabe 3)

Gegeben ist die Laplace-Übertragungsfunktion einer einzelnen Roboterachse (rotatorischer Gleichstrommotor) durch:

$$\frac{Q_M(s)}{I_A(s)} = \frac{k_M}{J_{\text{ges}} s^2 + d_{\text{eff}} s}.$$

Dabei bezeichnen $Q_M(s)$ und $I_A(s)$ die Laplace-Transformierten der Motorlage $q_M(t)$ bzw. des Ankerstroms $i_A(t)$. Aus Vereinfachungsgründen sind im Folgenden alle Größen als dimensionslos zu betrachten.

Gegeben: k_M , d_{eff} , J_{ges} .

- Welche zusätzlichen Vernachlässigungen bzw. vereinfachenden Modellannahmen waren notwendig, um auf das gegebene Modell einer Roboterachse zu kommen? Nennen Sie zwei!
- Zeichnen Sie das Blockschaltbild für eine P/P-Kaskadenregelung mit Geschwindigkeitsvorsteuerung im Jointspace für eine einzelne Roboterachse, die wie oben angegeben modelliert ist! Berücksichtigen Sie zusätzlich die Störmomente unter der Annahme, dass die Getriebeübersetzung u_G sehr viel größer als eins ist!
- Was ist der Vorteil einer Regelung inklusive Vorsteuerung im Vergleich zu einer Regelung ohne Vorsteuerung?
- Ermitteln Sie die Führungs- sowie Störgrößenübertragungsfunktion des geschlossenen Regelkreises! Der Herleitungsweg muss klar erkennbar sein.
- Bestimmen Sie die bleibende Regelabweichung hinsichtlich eines Führungs- und Störgrößensprungs! Wie könnte der Regelkreis ggf. modifiziert werden, um eine bleibende Regelabweichung zu verhindern?
- Für einen ähnlichen Regler gilt folgende Führungsgrößenübertragungsfunktion:

$$G_F(s) = \frac{Q_M(s)}{Q_{M,s}(s)} = \frac{K_P K_D}{0.5 s^3 + (2 + K_P) s^2 + K_P K_D s}.$$

Bestimmen Sie die Reglerkoeffizienten K_P und K_D so, dass zwei Pole des geschlossenen Regelreises bei $s_{1,2} = -4$ liegen! Der Rechenweg muss klar erkennbar sein. Ist das System schwingungsfähig? Begründen Sie Ihre Antwort! Wie nennt man diesen Fall?

13 Musterklausur Robotik I

Klausur Robotik I

Titelblatt

Name: _____

Matrikelnummer: _____

Hinweise zur Bearbeitung:

- Kein eigenes Papier verwenden!
 - Bitte jedes Blatt mit Namen und Matrikelnummer versehen!
 - Bitte nicht in rot, grün oder mit Bleistift schreiben!
 - Bitte geben Sie vollständige Lösungswege an! Der Rechenweg muss klar erkennbar sein!
 - Alle niedergeschriebenen Antworten werden gewertet, außer sie sind erkennbar durchgestrichen.
 - Gegebenenfalls (zwischen-)definierte Größen müssen nicht explizit eingesetzt werden.
 - Die angegebene Punktzahl ist eine ungefähre, das endgültige Punkteschema kann abweichen.
 - Sollte der vorgesehene Platz nicht ausreichen, erhalten Sie weitere Blätter (bitte melden).
-

Punkte	Fragen:	
	Aufgaben:	
	Gesamt:	

Kurzfragenteil

Frage 1:

Nennen Sie jeweils einen Vor- und Nachteil von Parallelkinematiken gegenüber seriellen Robotern!

Frage 2:

Wie berechnet sich aus den Elementardrehmatrizen $\mathbf{R}_x(\alpha)$, $\mathbf{R}_y(\beta)$ und $\mathbf{R}_z(\gamma)$ die Rotationsmatrix \mathbf{R}_{KARD} (zusammengesetzte Drehung) unter Verwendung von KARDAN-Winkeln? Nennen Sie den wesentlichen Unterschied im Vergleich zu zusammengesetzten Drehungen unter Verwendung von RPY-Winkeln!

Gegeben: $\mathbf{R}_x(\alpha)$, $\mathbf{R}_y(\beta)$, $\mathbf{R}_z(\gamma)$.

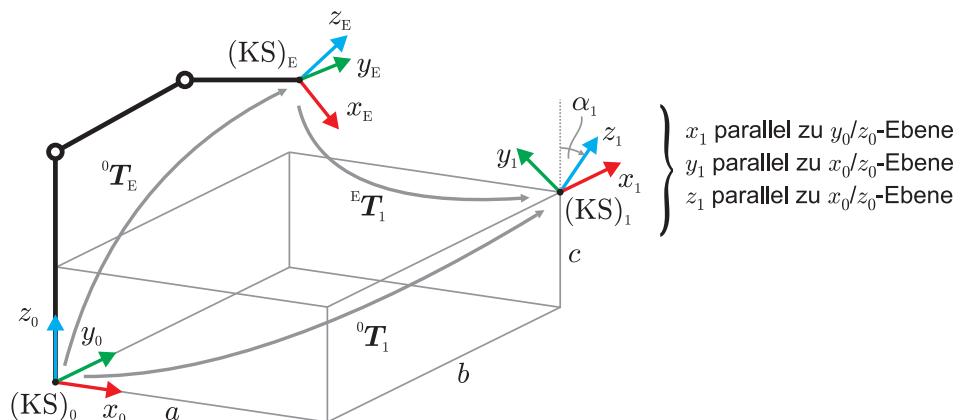
Hinweis: Die gegebenen Elementardrehungen müssen nicht ausmultipliziert oder explizit hingeschrieben werden.

Frage 3:

Nennen Sie zwei Nachteile von zusammengesetzten Drehung. Welche alternative Beschreibung von Rotationen bzw. Orientierungen schlagen Sie bezugnehmend auf Ihre genannten Nachteile vor?

Frage 4:

Bestimmen Sie die homogene Transformationsmatrix ${}^0\mathbf{T}_1$, die die Transformation von $(\text{KS})_1$ nach $(\text{KS})_0$ beschreibt! Bestimmen Sie anschließend unter Verwendung der bekannten Transformation ${}^E\mathbf{T}_1$ für die dargestellte Konfiguration des Roboters die Transformation der direkten Kinematik ${}^0\mathbf{T}_E$!



Gegeben: $a, b, c, \alpha_1 = 45^\circ$ (Drehung um x_1), ${}^E\mathbf{T}_1 = \begin{pmatrix} \cos(\beta) & \sin(\beta) & 0 & d \\ -\sin(\beta) & \cos(\beta) & 0 & 0 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

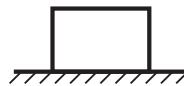
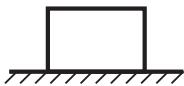
Hinweis (aus Vorlesung): $({}^0\mathbf{T}_B)^{-1} = \left(\begin{array}{c|c} ({}^0\mathbf{R}_B)^T & -({}^0\mathbf{R}_B)^T ({}^0\mathbf{r}_B) \\ \hline 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{array} \right)$.

Frage 5:

Geben Sie eine rekursive Vorschrift zur numerischen Lösung der inversen Kinematik eines seriellen Roboters an und benennen Sie die einzelnen Größen.

Frage 6:

- Wodurch zeichnet sich eine kinematische Singularität physikalisch aus?
- Wie lautet die mathematische Bedingung für eine kinematische Singularität?
- Skizzieren Sie zwei singuläre Konfigurationen eines planaren RR-Roboters! Skizzieren Sie dabei nur Konfigurationen, die sich nicht trivial aus den vorherigen ergeben!

**Frage 7:**

Eine Roboterachse soll sich in der Zeit t_e von $q(0) = q_S$ nach $q(t_e) = q_Z$ bewegen. Zu Beginn und am Ende der Bewegung soll die Geschwindigkeit stetig auf Null abklingen ($\dot{q}(0) = \dot{q}(t_e) = 0$). Gegeben: $q_S, q_Z, t_e, q_S < q_Z$.

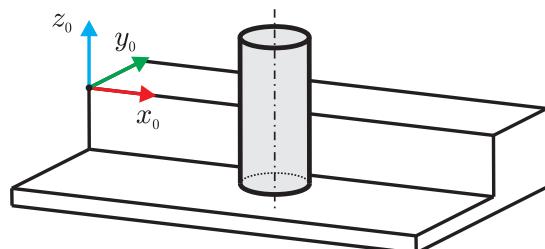
- Bestimmen Sie die Koeffizienten a_i eines Polynomansatzes dritter Ordnung, der diese Anforderungen erfüllt!
- Skizzieren Sie qualitativ den Geschwindigkeitsverlauf $\dot{q}(t)$!

Frage 8:

Wie lautet die allgemeine Form der dynamischen Gleichungen (das Differentialgleichungssystem) eines Roboters mit n Freiheitsgraden (n generalisierten Koordinaten q_i)? Benennen Sie die einzelnen Größen und geben Sie die Dimensionen der verwendeten Vektoren und/oder Matrizen an!

Frage 9:

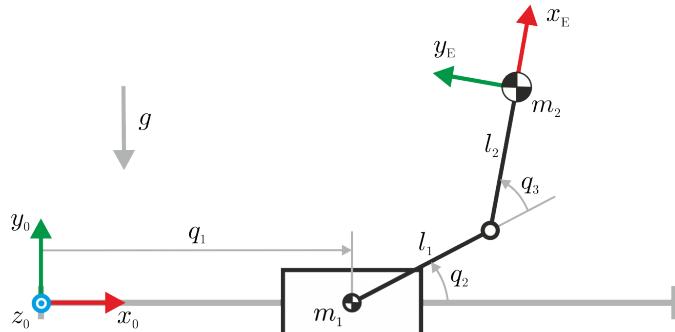
Der Endeffektor eines Industrieroboters soll den abgebildeten rotationssymmetrischen Bolzen in x_0 -Richtung entlang einer Kante bewegen. Es ist anzunehmen, dass keine Reibung vorhanden ist. Geben Sie für diese Aufgabenstellung die künstlichen und natürlichen Randbedingungen an!



Aufgabenteil

Aufgabe 1

Gegeben ist der abgebildete PRR-Portalroboter mit den Gelenkvariablen q_1 , q_2 und q_3 .



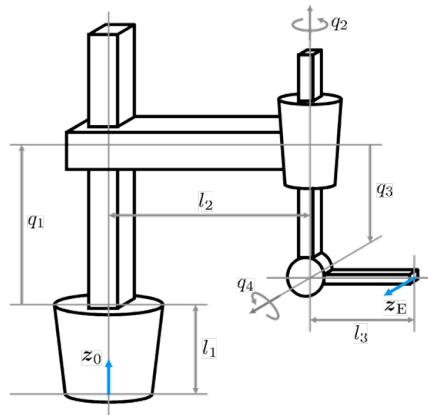
Der Schlitten 1 (Masse m_1) sowie der Endeffektor (Masse m_2) sind als Punktmassen zu betrachten. Alle anderen Teile sind masselos anzunehmen. Die Länge des ersten am Schlitten befestigten Armgliedes ist l_1 . Das zweite Armglied hat die Länge l_2 . Das reibungsfreie und starre System unterliegt wie skizziert der Erdbeschleunigung g . Die Umweltkoordinaten $(0)x_E$ und $(0)y_E$ bestimmen die Position des Endeffektors im Inertialkoordinatensystem $(KS)_0$. Die Orientierung soll unberücksichtigt bleiben und ist demnach nicht als Freiheitsgrad des Endeffektors zu betrachten.

Gegeben: $l_1, l_2, m_1, m_2, q_{1\max} > 2(l_1 + l_2), g$.

- Skizzieren Sie den Arbeitsraum für $0 \leq q_1 \leq q_{1\max}$, $0 \leq q_2 \leq \pi$ und $0 \leq q_3 \leq 2\pi$!
Nehmen Sie dazu die Abmaße des Schlittens zu null an! Welches sind die minimalen und maximalen Werte, die $(0)x_E$ und $(0)y_E$ annehmen können?
- Bestimmen Sie die direkte Kinematik!
- Kann die inverse Kinematik analytisch berechnet werden? Begründen Sie Ihre Aussage!
- Bestimmen Sie die analytische Jacobi-Matrix!
- Bestimmen Sie die Lagrange'sche Funktion L für das System als Funktion der Gelenkvariablen q_i und ihrer zeitlichen Ableitungen!
- Wie lautet die Gleichung zur Bestimmung der Bewegungsgleichungen unter Verwendung der Lagrange'schen Funktion L ?
- Bestimmen Sie die Kraft τ_1 für das erste Gelenk!

Aufgabe 2

Gegeben ist der abgebildete RPPR-Roboter mit den Gelenkvariablen q_1, q_2, q_3 und q_4 .



Die Drehrichtungen sind im mathematisch positiven Sinn um die gegebenen Achsen anzunehmen.

Gegeben: l_1, l_2, l_3 .

- Zeichnen Sie die körperfesten Koordinatensysteme sowie das Inertialkoordinatensystem $(KS)_0$ im Fußpunkt des Roboters mit Achsenbezeichnungen entsprechend der Denavit-Hartenberg Notation (DH Notation) in das gegebene Bild ein! Legen Sie den Ursprung des letzten Koordinatensystems an die Spitze des Endeffektors!
- Wie viele Möglichkeiten existieren, für die Platzierung von $(KS)_1$? Begründen Sie Ihre Aussage und zeichnen Sie ggf. eine Alternativlösung ein!
- Bestimmen Sie die DH Parameter!
- Geben Sie die allgemeine Form zur Berechnung der DH Matrix $\mathbf{A}_i(q_i)$ in Abhängigkeit der DH Parameter an!
- Bestimmen Sie die DH Matrix $\mathbf{A}_1(q_1)$!

Aufgabe 3

Gegeben ist die Laplace-Übertragungsfunktion einer einzelnen Roboterachse (rotatorischer Gleichstrommotor) durch:

$$\frac{Q_M(s)}{I_A(s)} = \frac{k_M}{J_{\text{ges}} s^2 + d_{\text{eff}} s}.$$

Dabei bezeichnen $Q_M(s)$ und $I_A(s)$ die Laplace-Transformierten der Motorlage $q_M(t)$ bzw. des Ankerstroms $i_A(t)$. Aus Vereinfachungsgründen sind im Folgenden alle Größen als dimensionslos zu betrachten.

Gegeben: k_M , d_{eff} , J_{ges} .

- a) Welche zusätzlichen Vernachlässigungen bzw. vereinfachenden Modellannahmen waren notwendig, um auf das gegebene Modell einer Roboterachse zu kommen? Nennen Sie zwei!
- b) Zeichnen Sie das Blockschaltbild für eine P/P-Kaskadenregelung mit Geschwindigkeitsvorsteuerung im Jointspace für eine einzelne Roboterachse, die wie oben angegeben modelliert ist! Berücksichtigen Sie zusätzlich die Störmomente unter der Annahme, dass die Getriebeübersetzung u_G sehr viel größer als eins ist!
- c) Was ist der Vorteil einer Regelung inklusive Vorsteuerung im Vergleich zu einer Regelung ohne Vorsteuerung?
- d) Ermitteln Sie die Führungs- sowie Störgrößenübertragungsfunktion des geschlossenen Regelkreises! Der Herleitungsweg muss klar erkennbar sein.
- e) Bestimmen Sie die bleibende Regelabweichung hinsichtlich eines Führungs- und Störgrößensprungs! Wie könnte der Regelkreis ggf. modifiziert werden, um eine bleibende Regelabweichung zu verhindern?
- f) Für einen ähnlichen Regler gilt folgende Führungsgrößenübertragungsfunktion:

$$G_F(s) = \frac{Q_M(s)}{Q_{M,s}(s)} = \frac{K_P K_D}{0.5 s^3 + (2 + K_P) s^2 + K_P K_D s}.$$

Bestimmen Sie die Reglerkoeffizienten K_P und K_D so, dass zwei Pole des geschlossenen Regelkreises bei $s_{1,2} = -4$ liegen! Der Rechenweg muss klar erkennbar sein. Ist das System schwingungsfähig? Begründen Sie Ihre Antwort! Wie nennt man diesen Fall?