

Matjaž Mihelj · Tadej Bajd
Aleš Ude · Jadran Lenarčič
Aleš Stanovnik · Marko Munič
Jure Rejc · Sebastjan Šlajpah

Robotik

Zweite Ausgabe

Robotik

Matjaž Mihelj • Tadej Bajd • Aleš Ude
Jadran Lenarčič • Aleš Stanovnik
Marko Munih • Jure Rejc • Sebastjan Šlajpah

Robotik

Zweite Ausgabe

123

Matjaž Mihelj
Fakultät für Elektrotechnik
Universität Ljubljana
Ljubljana, Slowenien

Aleš Stanovnik
Fakultät für Elektrotechnik
Universität Ljubljana
Ljubljana, Slowenien

Tadej Bajd
Fakultät für Elektrotechnik
Universität Ljubljana
Ljubljana, Slowenien

Marko Munih
Fakultät für Elektrotechnik
Universität Ljubljana
Ljubljana, Slowenien

Aleš Ude
Abteilung für Automatik,
Biokybernetik und Robotik
Jožef-Stefan-Institut
Ljubljana, Slowenien

Jure Rejc
Fakultät für Elektrotechnik
Universität Ljubljana
Ljubljana, Slowenien

Jadran Lenarčič
Jožef-Stefan-Institut
Ljubljana, Slowenien

Sebastjan Šljapah
Fakultät für Elektrotechnik
Universität Ljubljana
Ljubljana, Slowenien

ISBN 978-3-319-72910-7

ISBN 978-3-319-72911-4 (eBook)

<https://doi.org/10.1007/978-3-319-72911-4>

Kontrollnummer der Library of Congress: 2018946678

1. Auflage: © Springer Science+Business Media BV 2010 2. Auflage: ©

Springer International Publishing AG, Teil von Springer Nature 2019 Dieses Werk unterliegt dem

Urheberrecht. Alle Rechte bleiben dem Herausgeber vorbehalten, unabhängig davon, ob es sich um das gesamte Material oder Teile davon handelt, insbesondere das Recht der Übersetzung, des Nachdrucks, der Wiederverwendung von Abbildungen, der Vorlesung, der Ausstrahlung, der Reproduktion auf Mikrofilmen oder auf andere physische Weise sowie der Übertragung oder Speicherung von Informationen und Abruf, elektronische Anpassung, Computersoftware oder durch ähnliche oder unterschiedliche Methoden, die jetzt bekannt sind oder später entwickelt werden.

Die Verwendung von allgemein beschreibenden Namen, eingetragenen Namen, Warenzeichen, Dienstleistungsmarken usw. in dieser Veröffentlichung bedeutet, auch ohne ausdrückliche Angabe, nicht, dass diese Namen von den einschlägigen Schutzgesetzen und -vorschriften ausgenommen und daher frei für die allgemeine Nutzung sind verwenden.

Der Herausgeber, die Autoren und die Herausgeber können davon ausgehen, dass die Ratschläge und Informationen in diesem Buch zum Zeitpunkt der Veröffentlichung wahr und korrekt sind. Weder der Herausgeber noch die Autoren oder Herausgeber geben eine Garantie, weder ausdrücklich noch stillschweigend, in Bezug auf das hierin enthaltene Material oder für etwaige Fehler oder Auslassungen, die möglicherweise gemacht wurden. Der Verlag bleibt hinsichtlich der Zuständigkeitsansprüche in veröffentlichten Karten und institutionellen Zugehörigkeiten neutral.

Gedruckt auf säurefreiem Papier

Dieses Springer-Imprimatur wird von der eingetragenen Firma Springer International Publishing AG, Teil von Springer Nature, herausgegeben

Die eingetragene Firmenadresse lautet: Gewerbestrasse 11, 6330 Cham, Schweiz

Vorwort

Es ist vielleicht schwierig, sich darüber zu einigen, was ein Roboter ist, aber die meisten Menschen, die in der Robotik arbeiten, würden wahrscheinlich den „Vater der Robotik“, Joseph F. Engelberger (1925–2015), einen Pionier der Industrierobotik, mit den Worten zitieren: „Das kann ich nicht.“ Definiere einen Roboter, aber ich erkenne einen, wenn ich einen sehe.“

Das Wort Roboter stammt nicht aus einem wissenschaftlichen oder technischen Vokabular, sondern wurde erstmals im tschechischen Drama „RUR“ (Rossums Universalroboter) von Karel Čapek verwendet, das 1921 in Prag uraufgeführt wurde. Das Wort selbst wurde von seinem Bruder erfunden Josef. In dem Drama ist der Roboter ein künstlicher Mensch, ein brillanter Arbeiter, dem alle „unnötigen Eigenschaften“ wie Emotionen, Kreativität und die Fähigkeit, Schmerz zu empfinden, fehlen. Im Prolog des Dramas wird folgende Definition von Robotern gegeben: „Roboter sind keine Menschen (Roboti nejsou lidé). Sie sind mechanisch perfekter als wir, sie haben eine erstaunliche intellektuelle Kapazität, aber sie haben keine Seele. Die Schöpfung eines Ingenieurs ist technisch raffinierter als das Produkt der Natur.“

Das Buch „Robotik“ entstand durch jahrzehntelange Lehrtätigkeit in Robotik an der Fakultät für Elektrotechnik der Universität Ljubljana, Slowenien, wo 1980 das erste Lehrbuch über Industrierobotik veröffentlicht wurde (A. Kralj und T. Bajd, „Industrijska robotika“). Die Art und Weise, dieses recht anspruchsvolle Thema darzustellen, wurde mit mehreren Generationen von Studenten im Grundstudium erfolgreich getestet.

Die zweite Ausgabe des Buches führt das Erbe der ersten Ausgabe fort, die 2011 von der Bibliothekszeitschrift CHOICE mit der Auszeichnung „Outstanding Academic Title“ ausgezeichnet wurde. Das Hauptmerkmal des Buches bleibt seine Einfachheit. Das Einführungskapitel deckt nun umfassend verschiedene Roboterklassen ab, wobei der Schwerpunkt auf Industrierobotern liegt. Die Position, Ausrichtung und Verschiebung eines Objekts werden durch homogene Transformationsmatrizen beschrieben. Diese Matrizen, die die Grundlage für jede Analyse von Robotermechanismen bilden, werden durch einfache geometrische Überlegungen eingeführt. Geometrische Modelle des Robotermechanismus werden mit Hilfe einer originellen, benutzerfreundlichen Vektorbeschreibung erklärt. Da die Welt des Robotikers sechsdimensional ist, wurde der Ausrichtung von Roboter-Endeffektoren in dieser Ausgabe mehr Aufmerksamkeit gewidmet.

Roboterkinematik und -dynamik werden über einen Mechanismus mit nur zwei Rotationsfreiheitsgraden eingeführt, der jedoch ein wichtiger Bestandteil der beliebtesten Industrieroboterstrukturen ist. Die Darstellung der Roboterdynamik basiert ausschließlich auf der Kenntnis des Newtonschen Gesetzes und wurde zusätzlich vereinfacht, um das Verständnis dieser relativ komplexen Materie zu erleichtern. Der Arbeitsbereich spielt eine wichtige Rolle bei der Auswahl eines für die geplante Aufgabe geeigneten Roboters. Die Kinematik paralleler Roboter unterscheidet sich deutlich von der Kinematik serieller Manipulatoren und verdient zusätzliche Aufmerksamkeit.

Die in dieser Ausgabe vorgestellten Robotersensoren sind nicht nur für industrielle Manipulatoren relevant, sondern auch für komplexe Systeme wie humanoide Roboter. Robot Vision spielt in industriellen Anwendungen eine immer wichtigere Rolle und die Planung der Robotertrajektorie ist eine Voraussetzung für eine erfolgreiche Robotersteuerung. Es werden grundlegende Steuerungsschemata erläutert, die entweder zur gewünschten Endpunktbahn oder zur Kraft zwischen dem Roboter und seiner Umgebung führen. Roboterumgebungen werden durch Produktmontageprozesse veranschaulicht, bei denen Roboter Teil einer Produktionslinie sind oder als völlig unabhängige Einheiten arbeiten. Außerdem werden Robotergreifer, Werkzeuge und Zuführgeräte beschrieben.

Da die Fabrikhalle immer komplexer wird, wird die Interaktion zwischen Menschen und Robotern unumgänglich sein. Kollaborative Roboter sind für eine sichere Mensch-Roboter-Interaktion konzipiert. Durch den Einsatz mobiler Radroboter kann die Flexibilität der Produktion weiter erhöht werden. Einen Blick in die Zukunft, in der Mensch und Roboter Begleiter sein werden, bietet das Kapitel über humanoide Robotik, deren Komplexität fortgeschrittenere Kenntnisse der Mathematik erfordert. Für Anwender von Industrierobotern ist das Kapitel zur Standardisierung und Messung von Genauigkeit und Wiederholbarkeit interessant.

Das Buch erfordert minimale fortgeschrittene Kenntnisse in Mathematik und Physik. Es eignet sich daher für Einführungskurse in die Robotik an ingenieurwissenschaftlichen Fakultäten (Elektrotechnik, Mechanik, Informatik, Bauwesen). Es könnte auch für Ingenieure interessant sein, die keine Robotik studiert haben, aber im Arbeitsumfeld mit Robotern in Berührung gekommen sind und sich auf einfache und schnelle Weise Grundkenntnisse aneignen möchten.

Ljubljana, Slowenien
April 2018

Matjaž Mihelj
Tadej Bajd

Inhalt

1 Einleitung	1
1.1 Roboter manipulator	4
1.2 Industrierobotik	6
2 Homogene Transformationsmatrizen	11
2.1 Translationale Transformation 11	
2.2 Rotationstransformation 12	
2.3 Pose und Verschiebung	16
2.4 Geometrisches Robotermodell	19
3 Geometrische Beschreibung des Robotermechanismus	27
3.1 Vektorparameter eines kinematischen Paares 27	
3.2 Vektorparameter des Mechanismus	31
4 Orientierung	39
5 Zwei-Segment-Roboter manipulator	49
5.1 Kinematik 5.2	49
Statik	54
5.3 Arbeitsbereich	55
5.4 Dynamik	60
6 Parallelroboter	69
6.1 Eigenschaften von Parallelrobotern	69
6.2 Kinematische Anordnungen paralleler Roboter	73
6.3 Modellierung und Design paralleler Roboter	78
7 Robotersensoren	85
7.1 Prinzipien der Wahrnehmung	85
7.2 Bewegungssensoren	86
7.2.1 Platzierung der Sensoren	86
7.2.2 Potentiometer	87

7.2.3 Optischer Encoder	88
7.2.4 Magnetischer Encoder	92
7.2.5 Drehzahlmesser	93
7.2.6 Inertiale Messeinheit	94
7.3 Kontaktsensoren	96
7.3.1 Taktile Sensor	96
7.3.2 Endschalter und Stoßstange	98
7.3.3 Kraft- und Drehmomentsensor	98
7.3.4 Gelenkdrehmomentsensor	101
7.4 Näherungs- und Entfernungssensoren	102
7.4.1 Ultraschall-Entfernungsmesser	102
7.4.2 Laser-Entfernungsmesser und Laserscanner	103
8 Robot Vision	107
8.1 Systemkonfiguration	107
8.2 Vorwärtsprojektion	108
8.3 Rückprojektion	113
8.3.1 Einzelkamera	113
8.3.2 Stereovision	115
8.4 Bildverarbeitung	118
8.5 Objektpose aus Bild	118
8.5.1 Kamerakalibrierung	118
8.5.2 Objekthaltung	120
9 Flugbahnplanung	123
9.1 Interpolation der Trajektorie zwischen zwei Punkten	123
9.2 Interpolation durch Verwendung von Via-Punkten	126
10 Robotersteuerung	133
10.1 Steuerung des Roboters in internen Koordinaten	134
10.1.1 PD-Positionskontrolle	135
10.1.2 PD-Positionsregelung mit Schwerkraftkompensation ...	136
10.1.3 Steuerung des Roboters auf Basis der Umkehrdynamik	137
10.2 Steuerung des Roboters in externen Koordinaten	141
10.2.1 Steuerung basierend auf der transponierten Jacobi-Matrix	142
10.2.2 Steuerung basierend auf der inversen Jacobi-Matrix	143
10.2.3 PD-Positionsregelung mit Schwerkraftkompensation ...	144
10.2.4 Steuerung des Roboters basierend auf inverser Dynamik	144
10.3 Regelung der Kontaktkraft	147
10.3.1 Linearisierung eines Robotersystems durch Inverse Dynamik	148
10.3.2 Kraftsteuerung	149

11 Roboterumgebung	153
11.1 Robotersicherheit	153
11.2 Roboterperipherie in Montageprozessen	158
11.2.1 Konfigurationen der Montage-Produktionslinie	158
11.3 Zuführgeräte	160
11.4 Förderer	164
11.5 Robotergreifer und Werkzeuge	165
12 Kollaborative Roboter	173
12.1 Kollaboratives Industrierobotersystem	173
12.2 Kollaborativer Roboter	175
12.3 Kollaborativer Betrieb.....	177
12.3.1 Sicherheitsbewerteter überwachter Stopp	178
12.3.2 Handführung	178
12.3.3 Geschwindigkeits- und Abstandsüberwachung	180
12.3.4 Leistungs- und Kraftbegrenzung	181
12.4 Kollaborative Robotergreifer	184
12.5 Anwendungen kollaborativer Robotersysteme	185
13 Mobile Roboter	189
13.1 Mobile Roboterkinematik 13.2	190
Navigation	197
13.2.1 Lokalisierung	197
13.2.2 Pfadplanung	201
13.2.3 Pfadsteuerung	202
14 Humanoide Robotik	209
14.1 Fortbewegung des Zweibeiners 211.....	
14.1.1 Nullmomentpunkt 211	
14.1.2 Generierung von Laufmustern	213
14.2 Nachahmungslernen	216
14.2.1 Beobachtung menschlicher Bewegung und ihrer Übertragung zu Humanoid Robot Motion 14.2.2	217
Dynamische Bewegungsgrundelemente	221
14.2.3 Konvergenzeigenschaften der linearen Dynamik Systeme	222
14.2.4 Dynamische Bewegungsgrundelemente für Punkt-zu-Punkt Bewegungen	223
14.2.5 Schätzung von DMP-Parametern aus einer einzigen Demonstration	225
14.2.6 Modulation von DMPs	227
15 Genauigkeit und Wiederholbarkeit industrieller Manipulatoren	231
Ableitung der Beschleunigung bei Kreisbewegungen	243
Index	247

Kapitel 1

Einführung



Die heutige Robotik kann als eine Wissenschaft beschrieben werden, die sich mit der intelligenten Bewegung verschiedener Robotermechanismen beschäftigt, die in die folgenden vier Gruppen eingeteilt werden können: Roboter manipulatoren, Roboterfahrzeuge, Mensch-Roboter-Systeme und biologisch inspirierte Roboter (Abb. 1.1). Die am häufigsten anzutreffenden Roboter manipulatoren sind serielle Robotermechanismen. Der Roboter manipulator wird durch eine serielle Kette starrer Körper, sogenannte Robotersegmente, dargestellt, die durch Gelenke verbunden sind. Serielle Roboter manipulatoren werden im nächsten Abschnitt dieses Kapitels ausführlicher beschrieben. Parallelroboter sind sowohl in der Wissenschaft als auch in der Industrie von großem Interesse. Dabei sind Roboterbasis und Plattform durch parallele Segmente, sogenannte Beine, miteinander verbunden. Die Segmente sind mit translatorischen Aktuatoren ausgestattet, während die Gelenke an Basis und Plattform passiv sind. Parallelroboter werden überwiegend für Pick-and-Place-Aufgaben eingesetzt. Sie zeichnen sich durch hohe Beschleunigungen, Wiederholgenauigkeit und Genauigkeit aus. Da die Roboter manipulatoren bei verschiedenen Produktionsaufgaben den menschlichen Bediener ersetzen, ähneln sie häufig der Größe eines menschlichen Arms. Hersteller können auch bis zu zehnmal größere Roboter manipulatoren anbieten, die komplette Autokarosserien manipulieren können. Im Gegensatz dazu kommen in den Bereichen Biotechnologie und neue Materialien Mikro- und Nanoroboter zum Einsatz. Nanoroboter ermöglichen das Schieben, Ziehen, Pick-and-Place-Manipulationen, das Ausrichten, Biegen und Rillen auf der Skala von Molekülen und Partikeln. Der am weitesten verbreitete Nanomanipulator basiert auf dem Prinzip der Autonomie.

Autonome Roboterfahrzeuge gibt es an Land, im Wasser und in der Luft. Die landgestützten mobilen Roboter werden am häufigsten in künstlichen Umgebungen wie Wohnungen, Krankenhäusern, Kaufhäusern oder Museen eingesetzt, sind aber zunehmend auch auf Autobahnen und sogar auf unwegsamem Gelände zu finden. Dennoch werden die meisten mobilen Roboter auf ebenem Untergrund mit Fortbewegung über Räder eingesetzt, wobei drei Räder für die nötige Stabilität sorgen. Oftmals sind die Räder speziell so konstruiert, dass sie omnidirektionale Bewegungen ermöglichen. Roboterfahrzeuge finden sich als Staubsauger, autonome Rasenmäher, intelligente Führer durch Kaufhäuser oder Museen, Begleiter

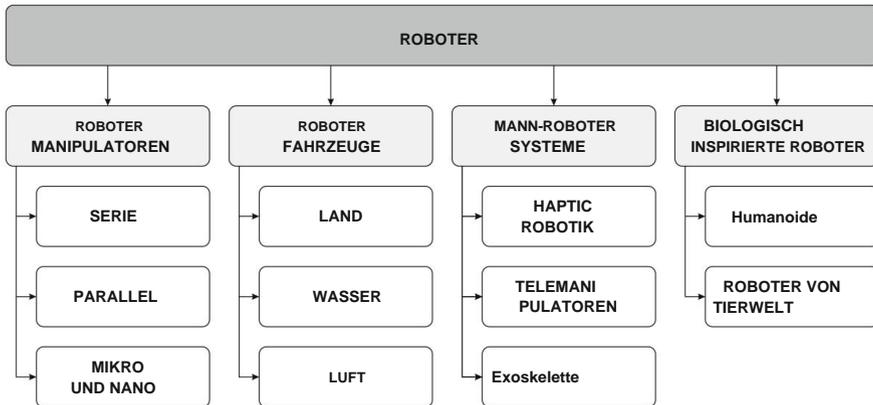


Abb. 1.1 Klassifizierung von Robotern

in klinischen Zentren, Raumfahrzeugen oder autonomen Autos. Die Schüler können bei verschiedenen Wettbewerben, zum Beispiel Fußball- oder Rettungsspielen, mit kleinen mobilen Robotern spielerisch lernen. Unter den Luftfahrzeugen scheinen kleine Quadcopter am beliebtesten zu sein. Sie haben einen sehr einfachen mechanischen Aufbau, was sie vergleichsweise kostengünstig macht. Quadcopter fliegen mit vier Rotoren und sind mit Sensoren wie Gyroskopen, Beschleunigungsmessern und Kameras ausgestattet und werden meist zu Überwachungszwecken eingesetzt. Größere autonome Versionen werden für militärische Aufklärungseinsätze eingesetzt. Wasserbasierte Roboter schwimmen entweder auf der Oberfläche oder arbeiten unter Wasser. Die Unterwasserversionen können die Form kleinerer autonomer U-Boote haben. Sie können häufig mit einem Roboterarm ausgestattet und in der Meeresforschung, zur Beobachtung des Meeresbodens oder von Schiffswracks oder als Begleiter auf Ölplattformen eingesetzt werden. Für meeresökologische Untersuchungen werden autonome schwimmende Roboter eingesetzt.

Neue Erkenntnisse im Bereich der Robotersteuerung beeinflussen stark die Entwicklung von Mensch-Roboter-Systemen wie haptischen Robotern, Telemanipulatoren und Exoskeletten. Der Einsatz haptischer Roboter bezieht sich auf virtuelle Umgebungen, die üblicherweise auf Computerbildschirmen angezeigt werden. Frühe virtuelle Umgebungen boten dem Betrachter Bild und Ton, jedoch keinen Tastsinn. Haptische Roboter vermitteln dem Benutzer das Gefühl von Berührung, begrenzter Bewegung, Nachgiebigkeit, Reibung und Textur in einer virtuellen Umgebung. Eine wichtige Rolle spielen haptische Roboter in der Rehabilitationsrobotik, wo kleine haptische Roboter zur Erfassung und Bewertung von Bewegungen der oberen Extremitäten bei gelähmten Personen eingesetzt werden. Stärkere haptische Systeme können das Handgelenk einer gelähmten Person halten und den Armendpunkt entlang der gewünschten Bahn führen, die dem Probanden in einer virtuellen Umgebung auf dem Computerbildschirm angezeigt wird. Der haptische Roboter übt zwei Arten von Kräften auf das Handgelenk des Probanden aus. Wenn der Patient nicht in der Lage ist, eine Bewegung entlang des ihm in der virtuellen Umgebung gezeigten Pfads auszuführen, schiebt der Roboter das Handgelenk entlang der erforderlichen Flugbahn und hilft dem Patienten, die Aufgabe zu erfüllen. Der Roboter hilft nur in dem Maße, wie es notwendig ist, damit der Patient den Zielpunkt erreicht. Wenn sich die gelähmte Extremität des Patienten vom Körper wegbewegt

In der geplanten Kurve schiebt der Roboter das Handgelenk in die Nähe der gewünschten Flugbahn. Telemanipulatoren sind Roboter, die von einem menschlichen Bediener gesteuert werden, wenn zwischen dem Telemanipulator und dem menschlichen Bediener eine Barriere besteht. Die Barriere zwischen dem Bediener und der Arbeitsumgebung ist normalerweise entweder die Entfernung (z. B. im Weltraum) oder die Gefährlichkeit (z. B. innerhalb eines Kernkraftwerks). Telemanipulatoren halten auch in der Medizin Einzug und werden in der Chirurgie (Telemedizin) eingesetzt. Exoskelette sind aktive Mechanismen, die an den oberen oder unteren Extremitäten des Menschen befestigt sind. Sie werden hauptsächlich zu Rehabilitationszwecken eingesetzt. Exoskelette der unteren Gliedmaßen können die Kraft gesunder Personen steigern oder gelähmten Personen die Umschulung des Gehens ermöglichen. Im Vergleich zu haptischen Rehabilitationsrobotern üben Exoskelette für die oberen Extremitäten Kräfte auf alle Segmente des gelähmten Armes aus.

Biologisch inspirierte Roboter können in humanoide Roboter und Roboter aus der Tierwelt unterteilt werden. Beispiele aus der Tierwelt sind verschiedene Arten von Roboterschlängen, Fischen, Vierbeinern, sechs- oder achtbeinigen Laufrobotern. Humanoide Roboter sind mit Abstand die fortschrittlichsten Robotersysteme in der Gruppe der biologisch inspirierten Roboter. Sie sind darauf ausgelegt, in einer menschlichen Umgebung zu leben und zu arbeiten. Die auffälligste Eigenschaft humanoider Roboter ist ihre Fähigkeit, auf zwei Beinen zu gehen. Sie gehen entweder mit statisch stabilem oder dynamisch stabilem Gang, sie können im Stand auf einem Bein balancieren, sie bewegen sich im Einklang mit menschlichen Kollegen und sie können sogar laufen. Die aktuellen Probleme in der humanoiden Robotik hängen mit künstlichem Sehen, Wahrnehmung und Analyse der Umgebung, Verarbeitung natürlicher Sprache, menschlicher Interaktion, kognitiven Systemen, maschinellem Lernen und Verhalten zusammen. Einige Roboter lernen auch aus Erfahrung, indem sie natürliche Prozesse wie Versuch und Irrtum und „Learning by Doing“ nachahmen, so wie ein kleines Kind lernt. Auf diese Weise erlangt der humanoide Roboter ein gewisses Maß an Autonomie, was darüber hinaus bedeutet, dass sich humanoide Roboter in manchen Situationen auf eine Weise verhalten können, die für ihre menschlichen Konstrukteure unvorhersehbar ist. Humanoide Roboter kommen in unsere Häuser und werden zu unseren Partnern. Sie könnten bald Begleiter älterer Menschen und Kinder, Assistenten von Krankenschwestern, Ärzten, Feuerwehrleuten und Arbeitern sein. Es entsteht die Notwendigkeit, Ethik in einem Roboter zu verkörpern, was als Robo-Ethik bezeichnet wird. Robo-Ethik ist eine angewandte Ethik, deren Ziel es ist, wissenschaftliche/kulturelle/technische Werkzeuge zu entwickeln, die von verschiedenen sozialen Gruppen und Glaubensrichtungen gemeinsam genutzt werden können. Diese Instrumente zielen darauf ab, die Entwicklung der Robotik zum Fortschritt der menschlichen Gesellschaft und des Einzelnen zu fördern und zu verhindern, dass sie Schaden anrichtet. Später, im Jahr 1983, fügte er das vierte Gesetz hinzu, das als nulltes Gesetz bekannt ist: Kein Roboter darf der Menschheit Schaden zufügen oder durch Untätigkeit zulassen, dass der Menschheit Schaden zugefügt wird. Die neue Generation humanoider Roboter wird Partner sein, die mit Menschen koexistieren, sie sowohl physisch als auch psychisch unterstützen und zur Verwirklichung einer sicheren und friedlichen Gesellschaft beitragen. Sie werden möglicherweise ethischer sein.

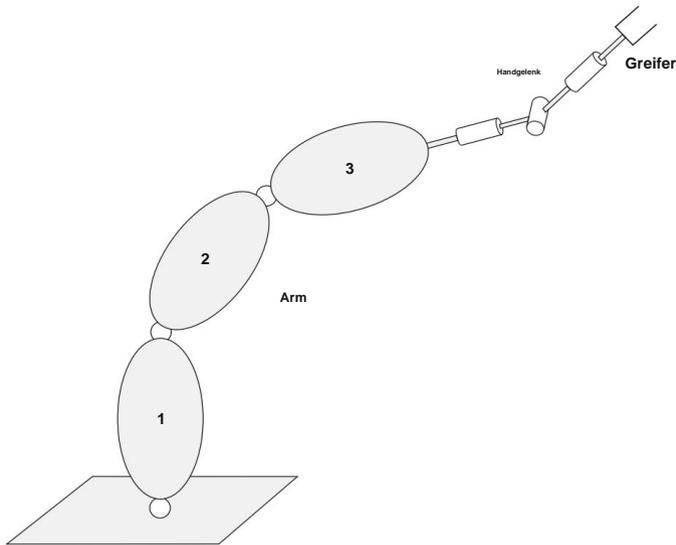


Abb. 1.2 Roboterarm

1.1 Roboterarm

Heutzutage sind die Industrieroboter-Manipulatoren die nützlichsten und effizientesten Robotersysteme, die den menschlichen Arbeiter bei schwierigen oder eintönigen Arbeiten oder dort, wo ein Mensch andernfalls mit gefährlichen Bedingungen konfrontiert wäre, ersetzen können. Der Roboterarm besteht aus Roboterarm, Handgelenk und Greifer (Abb. 1.2). Der Roboterarm ist eine serielle Kette aus drei relativ langen starren Segmenten, die für die Positionierung des Greifers im Arbeitsraum sorgen. Benachbarte Segmente eines Roboterarms werden durch ein Robotergelenk verbunden, das (Abb. 1.3) entweder translatorisch (prismatisch) oder rotatorisch (revolut) ist. Das Drehgelenk hat die Form eines Scharniers und begrenzt die Bewegung zweier benachbarter Segmente auf eine Drehung um die Gelenkachse. In der Robotik werden die Gelenkwinkel mit dem griechischen Buchstaben γ bezeichnet. In den vereinfachten Diagrammen wird das Drehgelenk durch einen Zylinder dargestellt. Das Translationsgelenk beschränkt die Bewegung zweier benachbarter Segmente auf Translation. Die relative Position zwischen zwei Segmenten wird als Abstand gemessen. Das Symbol des Translationsgelenks ist ein Prisma, während der Abstand mit dem Buchstaben d bezeichnet wird. Robotergelenke werden entweder von Elektro- oder Hydraulikmotoren angetrieben. Die Sensoren in den Gelenken messen den Winkel oder Abstand, die Geschwindigkeit und das Drehmoment.

Das Roboterhandgelenk besteht üblicherweise aus drei Drehgelenken. Die Aufgabe des Roboterhandgelenks besteht darin, die erforderliche Ausrichtung des vom Robotergreifer erfassten Objekts zu ermöglichen. Der Zwei- oder Mehrfinger-Robotergreifer wird am Endpunkt des Roboters platziert. Am Endpunkt können auch verschiedene Werkzeuge angebracht werden, um Bohr-, Spritzlackier- oder Schweißgeräte zu ermöglichen. Manipulatoren von Industrierobotern ermöglichen in der Regel eine Beweglichkeit in s

1.1 Roboterarm

5

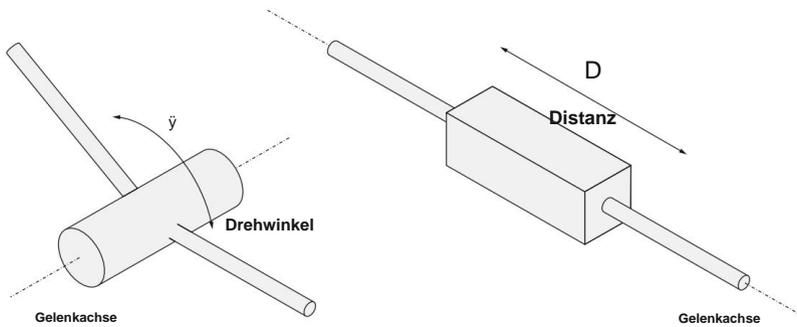


Abb. 1.3 Rotations- (links) und translatorisches (rechts) Robotergelenk

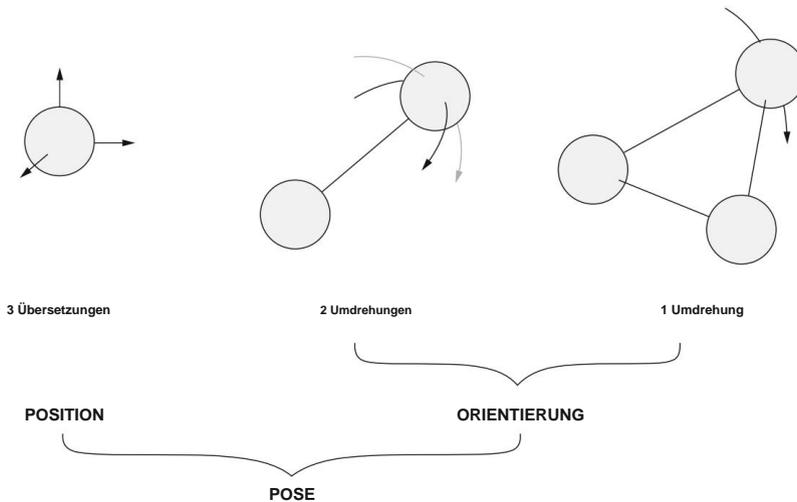


Abb. 1.4 Freiheitsgrade eines starren Körpers

Freiheit, was bedeutet, dass der Robotermechanismus sechs Gelenke und auch sechs Aktuatoren hat. Auf diese Weise kann der Roboterarm ein Objekt an einer beliebigen Stelle im Roboterarbeitsraum positionieren, während der Greifer das Objekt um alle drei Achsen eines rechteckigen Koordinatenrahmens drehen kann.

Um den Begriff Freiheitsgrad zu verdeutlichen, betrachten wir zunächst einen starren Körper, der üblicherweise das vom Industrieroboter manipulierte Objekt darstellt. Der einfachste starre Körper besteht aus drei Masseteilchen (Abb. 1.4). Ein einzelnes Masseteilchen hat drei Freiheitsgrade, die durch drei Verschiebungen entlang der Achsen eines rechteckigen Rahmens beschrieben werden. Die Verschiebung entlang einer Linie nennt man Translation. Wir fügen dem ersten ein weiteres Massenteilchen hinzu, so dass zwischen ihnen ein konstanter Abstand besteht. Das zweite Teilchen kann sich nur auf der Oberfläche einer Kugel bewegen, die das erste Teilchen umgibt. Seine Position auf der Kugel kann durch zwei Kreise beschrieben werden, die an Meridiane und Breitengrade auf einem Globus erinnern. Die Verschiebung entlang einer Kreislinie beträgt

Rotation genannt. Das dritte Masseteilchen wird so hinzugefügt, dass die Abstände zu den ersten beiden Teilchen konstant bleiben. Auf diese Weise kann sich das dritte Teilchen entlang des Kreises, einer Art Äquator, um die Achse bewegen, die durch die ersten beiden Teilchen bestimmt wird. Ein starrer Körper hat also sechs Freiheitsgrade: drei Translationen und drei Rotationen. Die ersten drei Freiheitsgrade beschreiben die Position des Körpers, während die anderen drei Freiheitsgrade seine Ausrichtung bestimmen. Der Begriff „Pose“ umfasst sowohl die Position als auch die Ausrichtung. Es wird oft gesagt, dass die Welt um uns herum dreidimensional ist, die Welt eines Robotikers jedoch sechsdimensional.

Moderne Industrieroboter manipulatoren sind unprogrammierbar und vielseitig einsetzbar. In der modernen industriellen Produktion ist es nicht mehr wirtschaftlich, große Material- oder Produktbestände vorzuhalten. Dies nennt man: „Just-in-time“-Produktion. Infolgedessen kann es vorkommen, dass sich am selben Tag verschiedene Arten eines bestimmten Produkts auf derselben Produktionslinie befinden. Dieses Problem, das für stationäre Automatisierungsgeräte am unangenehmsten ist, kann durch den Einsatz von in der Industrie unprogrammierbaren Roboter manipulatoren effizient gelöst werden. Mit unprogrammierbaren Robotern können wir per Knopfdruck von der Produktion eines Produkttyps zu einem anderen wechseln. Darüber hinaus ist der Roboter manipulator ein Mehrzweckmechanismus. Der Robotermechanismus ist eine grobe Nachahmung des menschlichen Arms. So wie wir unseren Arm sowohl für präzise als auch für schwere Arbeiten einsetzen, können wir denselben Roboter manipulator für verschiedene Aufgaben einsetzen. Dies ist umso wichtiger, wenn man bedenkt, dass die wirtschaftliche Lebensdauer eines Industrieroboters recht lang ist (12–16 Jahre). Daher kann es vorkommen, dass ein zu Schweißzwecken angeschaffter Roboter manipulator einer Pick-and-Place-Aufgabe zugewiesen wird. Roboterarme haben eine weitere wichtige Eigenschaft: Die Achsen zweier benachbarter Gelenke sind entweder parallel oder senkrecht. Da der Roboterarm nur über drei Freiheitsgrade verfügt, ist die Anzahl der möglichen Strukturen von Roboterarmen begrenzt. Zu den am häufigsten verwendeten gehören der anthropomorphe Roboterarm und der sogenannte SCARA-Roboterarm (Selective Compliant Articulated Robot for Assembly). Der anthropomorphe Roboterarmtyp (Abb. 1.5) verfügt über alle drei Gelenke des Rotationstyps und ähnelt daher dem menschlichen Arm weitestgehend. Die zweite Gelenkachse steht senkrecht zur ersten, während die dritte Gelenkachse parallel zur zweiten verläuft. Der Arbeitsbereich des anthropomorphen Roboterarms, der alle vom Roboterendpunkt erreichbaren Punkte umfasst, hat eine Kugelform. Der SCARA-Roboterarm taucht erst relativ spät in der Entwicklung der Industrierobotik auf (Abb. 1.6) und wird überwiegend für industrielle Montageprozesse eingesetzt. Zwei der Gelenke sind rotatorisch und eines translatorisch. Die Achsen aller drei Gelenke sind parallel. Der Arbeitsbereich des SCARA-Roboterarms ist zylindrisch. Auf dem Markt finden wir auch drei weitere kommerziell erhältliche Strukturen der Roboterarme: zylindrisch, kartesisch und in geringerem Maße kugelförmig.

1.2 Industrierobotik

Industrielle Roboter manipulatoren sind aus der heutigen Industrie nicht mehr wegzudenken und lassen sich in drei verschiedene Gruppen einteilen. In die erste Gruppe klassifizieren wir die Industrieroboter, die in einer Roboterzelle die Rolle des Masters übernehmen. Normalerweise eine Roboterzelle

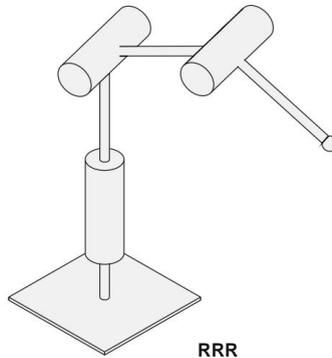


Abb. 1.5 Antropomorpher Roboterarm

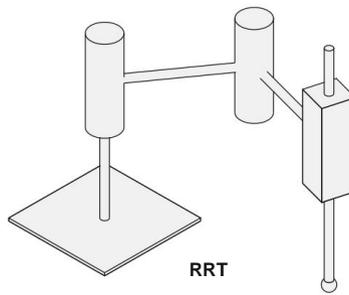


Abb. 1.6 SCARA-Roboterarm

beeinträchtigt einen oder mehrere Roboter, Arbeitsstationen, Lagerpuffer, Transportsysteme und numerisch gesteuerte Maschinen. In der zweiten Gruppe gibt es die Roboter, die Sklaven innerhalb der Roboterzelle sind. In die dritte Gruppe zählen wir die Industrieroboter, die in speziellen Anwendungen eingesetzt werden (Abb. 1.7).

Robotermeister in einer Roboterzelle sind in folgenden Produktionsprozessen zu finden: Schweißen, Lackieren, Beschichten und Versiegeln, Zerspanen und Montieren. Roboterschweißen (Punkt-, Lichtbogen-, Laserschweißen) stellt die häufigsten Roboteranwendungen dar. Es zeichnet sich durch Schnelligkeit, Präzision und Genauigkeit aus. Roboterschweißen ist besonders wirtschaftlich, wenn es im Dreischichtbetrieb durchgeführt wird. Heute treffen wir auf die größte Anzahl an Schweißrobotern in der Automobilindustrie. Dort beträgt das Verhältnis von menschlichen Arbeitern und Robotern 6:1. Industrieroboter werden häufig in aggressiven oder gefährlichen Umgebungen eingesetzt, beispielsweise beim Spritzlackieren. Die Roboterspritzlackierung bedeutet eine Materialeinsparung bei gleichzeitig höherer Qualität der lackierten Oberflächen. In einer toxischen Umgebung kann die soziale Motivation für die Einführung von Robotern die wirtschaftlichen Faktoren überwiegen. Bei Bearbeitungsanwendungen hält der Roboter normalerweise entweder ein Werkstück oder eine angetriebene Spindel und führt Bohren, Schleifen, Entgraten oder andere ähnliche Anwendungen aus. Roboter manipulieren zunehmend Einzel-

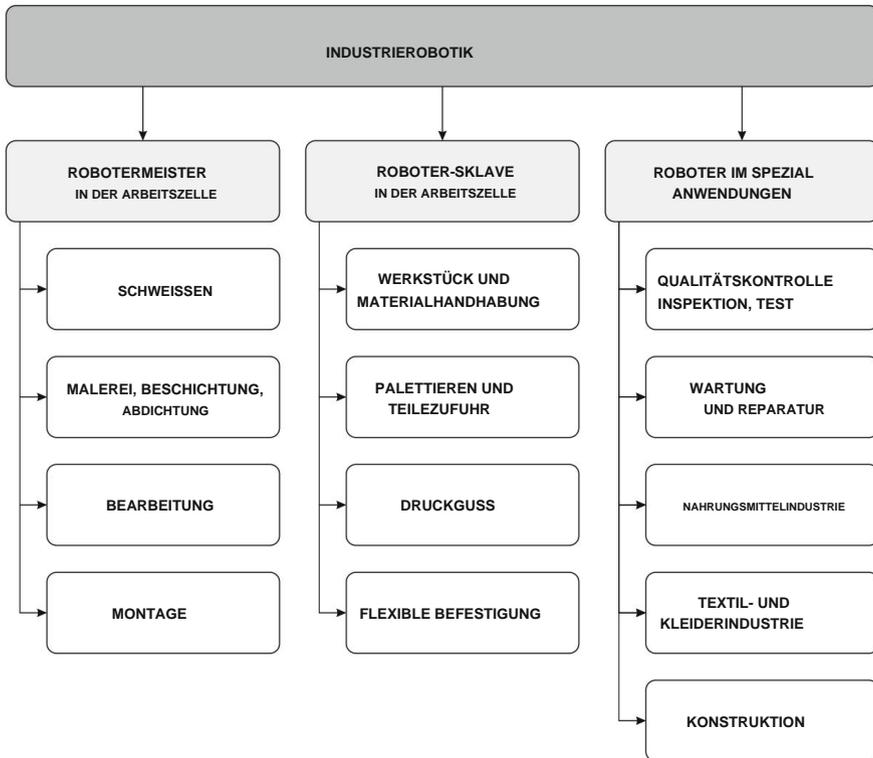


Abb. 1.7 Klassifizierung von Industrierobotern

ein wichtiges Einsatzgebiet für Montageroboter. Auch in der Automobilindustrie gibt es attraktive Montagevorgänge, bei denen ein Roboter Klebstoff auf die Windschutzscheibe aufträgt, während ein anderer Roboter die Windschutzscheibe hält und in die Öffnung der Fahrzeugkarosserie einführt.

Der Roboter übernimmt die Rolle eines Sklaven in den folgenden industriellen Anwendungen: Werkstück- und Materialhandhabung, Palettierung und Teilezuführung, Druckguss und flexible Vorrichtungen. In dieser Situation kann einer numerisch gesteuerten Maschine in der Roboterzelle die Rolle eines Masters übertragen werden. Pick-and-Place-Roboter stellen den häufigsten Einsatz von Robotern in der Materialhandhabung dar, wo Aufgaben oft mühsam oder sich wiederholend und potenziell gefährlich sind (z. B. Beladen von Pressen). Oftmals werden Industrieroboter bei Aufgaben eingesetzt, bei denen sie Punkt-zu-Punkt-Bewegungen ausführen. Solche Beispiele findet man beim Palettieren, also dem Ordnen von Werkstücken oder Produkten zum Zwecke der Verpackung oder Übergabe an eine Maschine. Besonders dann, wenn es sich um schwere Gegenstände handelt (z. B. Fässer in einer Brauerei), ist die Roboterpalettierung sinnvoll und willkommen. Druckgussvorgänge sind heiß, schmutzig und gefährlich und bieten eine unangenehme Umgebung für menschliche Arbeiter. Beim Roboterhandlung werden die Druckgussteile in der Druckguss

Einsatz flexibler Befestigungssysteme. Die Flexibilität einer Roboterzelle wird durch servogesteuerte programmierbare Positionierer erreicht, wodurch der Fertigungsprozess schneller und geschickter durchgeführt werden kann.

Besondere Einsatzgebiete der Industrieroboter sind: Qualitätssicherung, Inspektion und Prüfung, Wartung und Reparatur, Roboter in der Lebensmittel-, Textil- und Bekleidungsindustrie sowie im Baugewerbe. Qualitätssicherung, Inspektion und Prüfung werden häufig in der Elektronikindustrie eingesetzt, wo elektrische Parameter (z. B. Spannung, Strom, Widerstand) beim Zusammenbau elektronischer Schaltkreise getestet werden. In dieser Situation führt der Roboter die notwendigen Messungen am Objekt durch (dimensional, elektrisch), während er es ergreift und in eine neue Position bringt. Bei der Wartung und Reparatur von Robotern werden teleoperierte und autonome Roboter für verschiedene Anwendungen in der Nuklearindustrie, auf Autobahnen, Eisenbahnen, bei der Wartung von Stromleitungen und bei der Flugzeugwartung eingesetzt. Roboter halten auch Einzug in die Lebensmittelindustrie, wo sie neben Handhabungs- und Verpackungsanwendungen in der Lebensmittelverarbeitung auch für Aufgaben wie die Zubereitung von Speisen oder sogar das Dekorieren von Pralinen eingesetzt werden. Die Textil- und Bekleidungsindustrie stellt besondere Probleme dar, da die Werkstücke schlaff sind und die Handhabung von Textilien oder ähnlichen Materialien äußerst kompliziert ist. Auf der ganzen Welt wurden viele verschiedene Arten von Baurobotern entwickelt, jedoch wurden nur sehr wenige kommerziell

Die zentralen Herausforderungen der heutigen Robotik sind die Mensch-Roboter-Interaktion und die Mensch-Roboter-Kollaboration. Die Entwicklung der sogenannten Soft-Robotik ermöglicht die Interaktion und Zusammenarbeit von Menschen und Robotern in industriellen Umgebungen, im Service- und Alltagsbereich. Bei der Entwicklung kollaborativer Roboter, kurz Co-Bots, muss die Sicherheit der Mensch-Roboter-Interaktion gewährleistet sein. Als erster Schritt der kollaborativen Roboterforschung war daher die Analyse menschlicher Verletzungen durch stumpfe oder scharfe Werkzeugstöße erforderlich. Basierend auf zahlreichen Studien zu Mensch-Roboter-Kollisionen wurden die sicheren Robotergeschwindigkeiten für gegebene Roboterträglichkeitseigenschaften ermittelt. Eine sichere Mensch-Roboter-Interaktion wird außerdem durch neuartige Steuerungssysteme gewährleistet, die das Drehmoment in jedem Roboterelenk messen, kleinste Kontakte zwischen dem Roboter und dem menschlichen Bediener erkennen und den Roboter sofort stoppen. Voraussetzung für die effiziente Drehmomentregelung ist es, um den Roboter manipulator im Kontakt mit dem menschlichen Bediener gefügig zu machen, wird auch ein biologisch inspirierter Ansatz verwendet. Durch die Speicherung der Energie in den Federelementen in den Roboterarmgelenken wird die Bewegungssteuerung effizient und natürlich. Komplexe Co-Bots, die oft als mehrarmige Robotersysteme eingesetzt werden, können nicht auf die gleiche Weise programmiert werden wie gewöhnliche Industrieroboter-Manipulatoren. Es müssen kognitive Robotikansätze eingeführt werden, die auf Techniken der künstlichen Intelligenz basieren, wie z. B. Nachahmungslernen, Lernen aus Demonstrationen, verstärkendes Lernen oder Lernen aus Belohnungen. Auf diese Weise können Co-Bots Aufgaben in unbekanntem und unstrukturierten Umgebungen ausführen. Besondere Aufmerksamkeit muss auch den Roboterhänden gewidmet werden. Bei der Zusammenarbeit mit einem menschlichen Bediener muss die Roboterhand humanoid sein, um Werkzeuge und Geräte bedienen zu können, die für die menschliche Hand konzipiert sind. Außerdem muss die Roboterhand die ausgeübten Kräfte messen, um einen sanften Griff zu gewährleisten. Die heutigen Industrieroboter arbeiten aus Sicherheitsgründen immer noch hinter den Zäunen. Die zaunlose industrielle Soft-Robotik hat das Potenzial, neuartige unvorhergesehene Anwendungen zu eröffnen und

Kapitel 2

Homogene Transformationsmatrizen



2.1 Translationale Transformation

Wie bereits erwähnt, verfügen Roboter entweder über Translations- oder Rotationsgelenke. Um den Grad der Verschiebung in einem Gelenk zu beschreiben, benötigen wir eine einheitliche mathematische Beschreibung der translatorischen und rotatorischen Verschiebungen. Die translatorische Verschiebung d , gegeben durch den Vektor

$$d = ai + bj + ck, \quad (2.1)$$

kann auch durch die folgende homogene Transformationsmatrix H beschrieben werden

$$H = \text{Trans}(a, b, c) = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} x \\ y \\ z \\ 1 \end{matrix} \quad (2.2)$$

Bei Verwendung homogener Transformationsmatrizen hat ein beliebiger Vektor die folgende 4×1 -Form

$$q = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = xyz \begin{matrix} x \\ y \\ z \\ 1 \end{matrix} \quad (2.3)$$

Eine translatorische Verschiebung des Vektors q für einen Abstand d erhält man durch Multiplikation des Vektors q mit der Matrix H

$$v = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x+a \\ y+b \\ z+c \\ 1 \end{pmatrix} \quad (2.4)$$

Die Übersetzung, die durch Multiplikation mit einer homogenen Matrix dargestellt wird, entspricht der Summe der Vektoren \mathbf{q} und \mathbf{d}

$$\mathbf{v} = \mathbf{q} + \mathbf{d} = (xi + yj + zk) + (ai + bj + ck) = (x + a)i + (y + b)j + (z + c)k. \quad (2.5)$$

In einem einfachen Beispiel wird der Vektor $1i + 2j + 3k$ translatorisch um die Distanz $2i + 5j + 4k$ verschoben

$$\mathbf{v} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 7 \\ 7 \end{pmatrix}$$

Das gleiche Ergebnis erhält man durch Addition der beiden Vektoren.

2.2 Rotationstransformation

Rotationsverschiebungen werden in einem rechtsdrehenden rechteckigen Koordinatensystem beschrieben, wobei die Rotationen um die drei Achsen, wie in Abb. 2.1 dargestellt, als positiv betrachtet werden. Positive Drehungen um die ausgewählte Achse erfolgen gegen den Uhrzeigersinn, wenn man vom positiven Ende der Achse auf den Ursprung O blickt. Die positive Drehung kann auch durch die sogenannte Rechte-Hand-Regel beschrieben werden, bei der der Daumen entlang der Achse auf ihr positives Ende gerichtet ist, während die Finger dies ze...

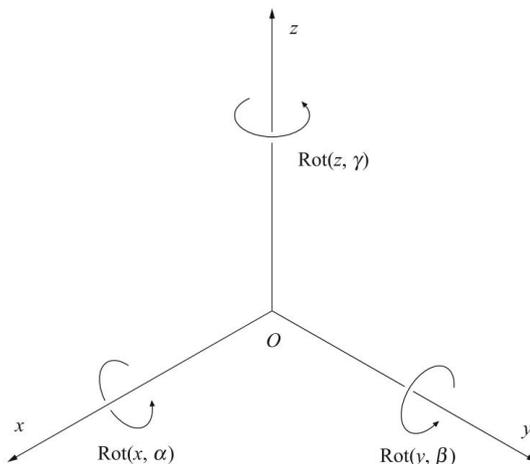


Abb. 2.1 Rechter rechteckiger Rahmen mit positiven Drehungen

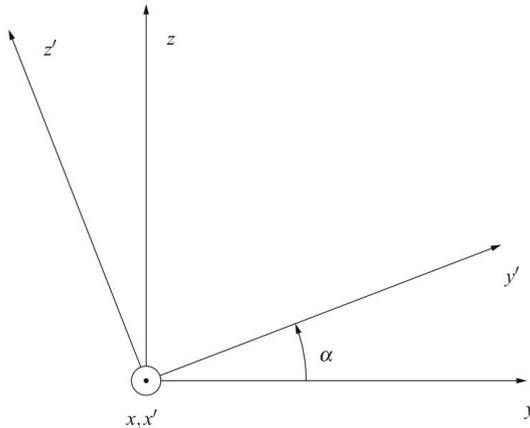


Abb. 2.2 Drehung um die x-Achse

positive Richtung der Rotationsverschiebung. Auch die Laufrichtung von Sportlern in einem Stadion ist ein Beispiel für eine positive Rotation.

Schauen wir uns zunächst die Drehung um die x-Achse genauer an. Die Koordinate

Der in Abb. 2.2 gezeigte Rahmen $x\text{-}y\text{-}z$ wurde durch Drehen des Referenzrahmens $x\text{-}y\text{-}z$ in positiver Richtung um die x-Achse um den Winkel \tilde{y} erhalten. Die Achsen x' und x kollinear.

Die Rotationsverschiebung wird ebenfalls durch eine homogene Transformationsmatrix beschrieben. Die ersten drei Zeilen der Transformationsmatrix entsprechen den x-, y- und z-Achsen des Referenzrahmens, während sich die ersten drei Spalten auf die x' , y' und z' x-Achsen des gedrehten Rahmens beziehen. Die oberen linken neun Elemente der Matrix H repräsentieren die 3×3 -Rotationsmatrix. Die Elemente der Rotationsmatrix sind Kosinuswerte der Winkel zwischen den Achsen, die durch die entsprechende Spalte und Zeile gegeben sind

$$\begin{aligned}
 \text{Rot}(x, \tilde{y}) &= \begin{matrix} & \begin{matrix} \tilde{y}x & \tilde{y}y & \tilde{y}z \end{matrix} \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{pmatrix} \cos 0\tilde{y} & 0 & 0 \\ 0 & \cos 90\tilde{y} & \sin 90\tilde{y} \\ \cos 90\tilde{y} & \sin 90\tilde{y} & \cos 90\tilde{y} \end{pmatrix} \end{matrix} \\
 &= \begin{matrix} & \begin{matrix} \tilde{y}x & \tilde{y}y & \tilde{y}z \end{matrix} \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \tilde{y} & \sin \tilde{y} \\ 0 & \sin \tilde{y} & \cos \tilde{y} \end{pmatrix} \end{matrix} \quad (2.6)
 \end{aligned}$$

Der Winkel zwischen x' und die x-Achsen sind $0\tilde{y}$ also haben wir $\cos 0\tilde{y}$ im dem x- Schnittpunktspalte und der x- Zeile. Der Winkel zwischen y' und die y- Achsen

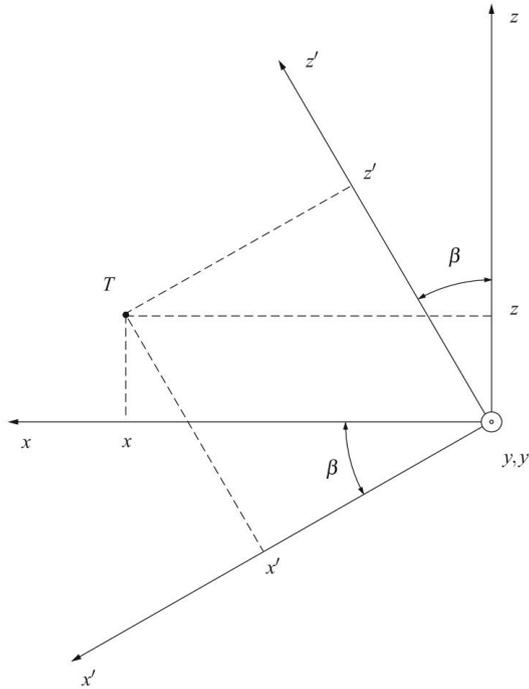


Abb. 2.3 Drehung um die y -Achse

ist 90° , wir setzen $\cos 90^\circ$ in den entsprechenden Schnittpunkt. Der Winkel zwischen dem y' und die y - Achse ist \tilde{y} , das entsprechende Matrixelement ist $\cos \tilde{y}$.

Um mit Rotationsmatrizen besser vertraut zu werden, werden wir mithilfe von Abb. 2.3 die Matrix ableiten, die eine Drehung um die y - Achse beschreibt . Die kollinearen Achsen sind y und y'

$$y = y' \quad (2.7)$$

Unter Berücksichtigung der Ähnlichkeit der Dreiecke in Abb. 2.3 ist es nicht schwierig, die folgenden zwei Gleichungen abzuleiten

$$\begin{aligned} \tilde{y} x &= x & \cos \tilde{y} + z' & \sin \tilde{y} \\ \tilde{y} z &= \tilde{y} x & \sin \tilde{y} + z' & \cos \tilde{y}. \end{aligned} \quad (2.8)$$

Alle drei Gleichungen. (2.7) und (2.8) können in Matrixform umgeschrieben werden

$$\text{Rot}(y, \tilde{y}) = \begin{pmatrix} \tilde{y} & x & z \\ y \cos \tilde{y} & 0 \sin \tilde{y} & 0 \\ 0 & 100 & \tilde{y} \\ \tilde{y} \sin \tilde{y} & 0 \cos \tilde{y} & 0 \\ 0 & 001 & \tilde{y} \end{pmatrix} \quad (2.9)$$

Die Drehung um die z-Achse wird durch die folgende homogene Transformationsmatrix beschrieben

$$\text{Rot}(z, \tilde{y}) = \begin{pmatrix} \tilde{y} & x & z \\ \cos \tilde{y} & \tilde{y} \sin \tilde{y} & 0 \\ \sin \tilde{y} & \cos \tilde{y} & 0 \\ 0 & 01 & 0 \\ 0 & 00 & 1 \end{pmatrix} \quad (2.10)$$

In einem einfachen numerischen Beispiel möchten wir den Vektor w bestimmen, den wir erhalten, indem wir den Vektor $u = 14i + 6j + 0k$ um 90° im Gegenuhrzeigersinn (also positiv) um die z-Achse drehen. Da $\cos 90^\circ = 0$ und $\sin 90^\circ = 1$ ist, ist es nicht schwierig, die Matrix zu bestimmen, die $\text{Rot}(z, 90^\circ)$ beschreibt, und sie mit dem Vektor u zu multiplizieren

$$w = \begin{pmatrix} 0 & \tilde{y} & 100 \\ \tilde{y} & 1 & 0 \\ 0 & 0 & 10 \\ \tilde{y} & 0 & 01 \end{pmatrix} \begin{pmatrix} 14 \\ 6 \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{y} & 14 \\ 0 & 0 \\ \tilde{y} & 1 \end{pmatrix}$$

Die grafische Darstellung der Drehung des Vektors u um die z-Achse ist in Abb. 2.4 dargestellt.

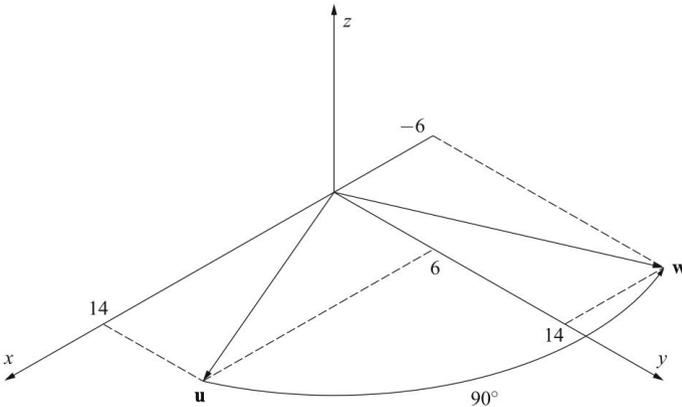


Abb. 2.4 Beispiel einer Rotationstransformation

2.3 Pose und Verschiebung

Im vorherigen Abschnitt haben wir gelernt, wie ein Punkt um die Achsen des kartesischen Rahmens verschoben oder gedreht wird. Im weiteren Verlauf werden wir uns für Verschiebungen von Objekten interessieren. Wir können immer einen Koordinatenrahmen an ein starres Objekt anhängen, das wir betrachten. In diesem Abschnitt beschäftigen wir uns mit der Pose und der Verschiebung rechteckiger Rahmen. Hier sehen wir, dass eine homogene Transformationsmatrix entweder die Pose eines Frames in Bezug auf einen Referenzframe beschreibt oder die Verschiebung eines Frames in eine neue Pose darstellt. Im ersten Fall stellt die obere linke 3×3 -Matrix die Ausrichtung des Objekts dar, während die rechte 3×1 -Spalte seine Position beschreibt (z. B. die Position seines Massenschwerpunkts). Die letzte Zeile der homogenen Transformationsmatrix wird immer durch $[0001]$ dargestellt. Im Falle einer Objektverschiebung entspricht die obere linke Matrix der Drehung und die rechte Spalte der Verschiebung des Objekts. Wir werden beide Fälle anhand einfacher Beispiele untersuchen. Lassen Sie uns zunächst die Bedeutung der homogenen Transformation betrachten. Betrachten wir das folgende Produkt homogener Matrizen, das eine neue homogene Transformationsmatrix H ergibt

$$\begin{aligned}
 H &= \text{Trans}(8, \ddot{y}6, 14)\text{Rot}(y, 90\ddot{y})\text{Rot}(z, 90\ddot{y}) \\
 &= \begin{pmatrix} 1 & 0 & 0 & 8 \\ \ddot{y} & 0 & 1 & 0 \\ 0 & 0 & 1 & 14 \\ \ddot{y} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \ddot{y} & \ddot{y} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \ddot{y} & \ddot{y} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & \ddot{y} & 1 & 0 \\ \ddot{y} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 0 & 1 & 8 \\ \ddot{y} & 1 & 0 & 0 \\ 0 & 1 & 0 & 14 \\ \ddot{y} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \ddot{y} & \ddot{y} & 0 & 0 \\ \ddot{y} & \ddot{y} & 0 & 0 \\ \ddot{y} & \ddot{y} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & \ddot{y} & 1 & 0 \\ \ddot{y} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.11)
 \end{aligned}$$

Bei der Definition der homogenen Matrix, die die Drehung darstellt, haben wir gelernt, dass die ersten drei Spalten die Drehung des Rahmens $x\text{-}y\text{-}z$ in Bezug auf den Referenzrahmen $x\text{-}y\text{-}z$ beschreiben

$$\begin{pmatrix} x' & y' & z' \\ \ddot{y} & 0 & \ddot{y} & \ddot{y} & 1 & \ddot{y} & 8 \\ \ddot{y} & 1 & 0 & 0 & \ddot{y} & \ddot{y} \\ \ddot{y} & 0 & \ddot{y} & 1 & \ddot{y} & 0 & \ddot{y} \\ \ddot{y} & 14 & 0 & 0 & 0 & 1 & \ddot{y} \end{pmatrix} \quad (2.12)$$

Die vierte Spalte stellt die Position des Ursprungs des Rahmens $x\text{-}y\text{-}z$ in Bezug auf den Referenzrahmen $x\text{-}y\text{-}z$ dar. Mit diesem Wissen können wir den durch die homogene Transformationsmatrix (2.11) beschriebenen Rahmen $x\text{-}y\text{-}z$ relativ zum Referenzrahmen $x\text{-}y\text{-}z$ (Abb. 2.5) grafisch darstellen. Die x -Achse zeigt

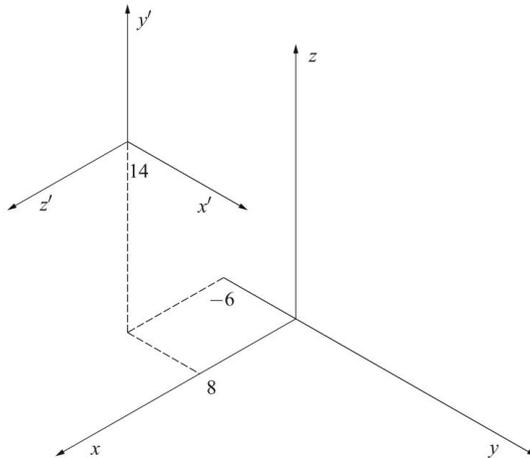


Abb. 2.5 Die Lage eines beliebigen Rahmens $x'-y'-z'$ in Bezug auf den Referenzrahmen $x-y-z$

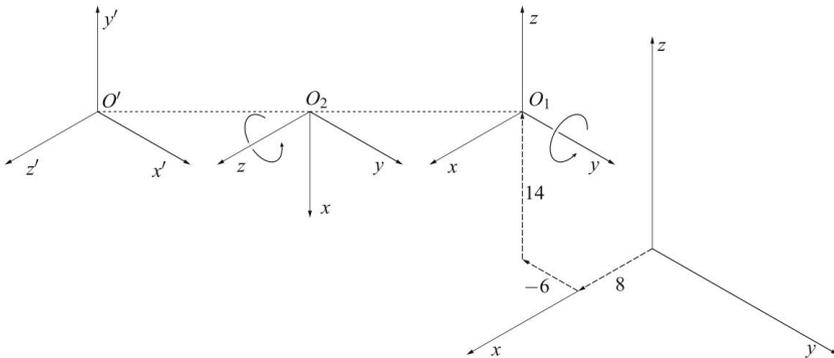


Abb. 2.6 Verschiebung des Referenzrahmens in eine neue Pose (von rechts nach links). Die Ursprünge sind durch den Punkt

Richtung der y' -Achse des Referenzrahmens, der y' -Achse liegt in Richtung der z -Achse, und der z' -Achse liegt in x -Richtung.

Achse. Um uns von der Richtigkeit des in Abb. 2.6 gezeichneten Rahmens zu überzeugen, überprüfen wir die in Gl. (2.11) enthaltenen Verschiebungen. Das Bezugssystem wird zunächst in den Punkt $(8, -6, 14)$ verschoben, anschließend um 90° um die neue y -Achse gedreht und schließlich um 90° um die neueste z -Achse gedreht (Abb. 2.6). Die drei Verschiebungen des Referenzrahmens führen zu derselben Endposition wie in Abb. 2.5.

Im weiteren Verlauf dieses Kapitels wollen wir die zweite Bedeutung der homogenen Transformationsmatrix erläutern, nämlich eine Verschiebung eines Objekts oder Koordinatensystems in eine neue Pose (Abb. 2.7). Zunächst möchten wir den Koordinatenrahmen $x-y-z$ um 90° gegen den Uhrzeigersinn um die z -Achse drehen. Dies kann durch die folgende Nachmultiplikation der Matrix H erreicht werden, die die Anfangsposition des beschreibt

Koordinatensystem $x-y-z$

$$H1 = H \cdot \text{Rot}(z, 90^\circ). \quad (2.13)$$

Die Verschiebung führte zu einer neuen Pose des Objekts und einem neuen Rahmen $x-y-z$, wie in Abb. 2.7 dargestellt. Wir werden diesen neuen Rahmen um \tilde{y}_1 entlang der xy -Achse, 3 Einheiten entlang der z -Achse und \tilde{y}_3 entlang der z -Achse verschieben.

$$H2 = H1 \cdot \text{Trans}(\tilde{y}_1, 3, \tilde{y}_3). \quad (2.14)$$

Nach der Übersetzung wird eine neue Pose des Objekts zusammen mit einem neuen Rahmen $x-y-z$ erhalten. Dieser Rahmen wird schließlich um 90° um die y -Achse in positiver Richtung gedreht.

$$H3 = H2 \cdot \text{Rot}(y, 90^\circ). \quad (2.15)$$

Die Gleichungen (2.13), (2.14) und (2.15) können nacheinander ineinander eingefügt werden.

$$H3 = H \cdot \text{Rot}(z, 90^\circ) \cdot \text{Trans}(\tilde{y}_1, 3, \tilde{y}_3) \cdot \text{Rot}(y, 90^\circ) = H \cdot D. \quad (2.16)$$

In Gl. (2.16) stellt die Matrix H die Anfangsposition des Rahmens dar, $H3$ ist die Endposition und D repräsentiert die Verschiebung.

$$\begin{aligned} D &= \text{Rot}(z, 90^\circ) \cdot \text{Trans}(\tilde{y}_1, 3, \tilde{y}_3) \cdot \text{Rot}(y, 90^\circ) \\ &= \begin{pmatrix} 0 & \tilde{y}_1 & 100 & 0 \\ \tilde{y} & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ \tilde{y} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 100 & \tilde{y}_1 & 0 & 0 \\ 0 & 10 & 3 & 0 \\ 0 & 0 & 1 & \tilde{y}_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 10 \\ \tilde{y} & \tilde{y} & 0 & 100 \\ \tilde{y} & \tilde{y} & 0 & 1000 \\ \tilde{y} & \tilde{y} & 0 & 001 \end{pmatrix} \\ &= \begin{pmatrix} 0 & \tilde{y}_1 & 0 & \tilde{y}_3 \\ \tilde{y} & 0 & 0 & \tilde{y}_1 \\ \tilde{y} & 100 & \tilde{y}_3 & 0 \\ \tilde{y} & 0 & 0 & 001 \end{pmatrix} \end{aligned} \quad (2.17)$$

Abschließend führen wir die Nachmultiplikation durch, die die neue relative Lage des Objekts beschreibt.

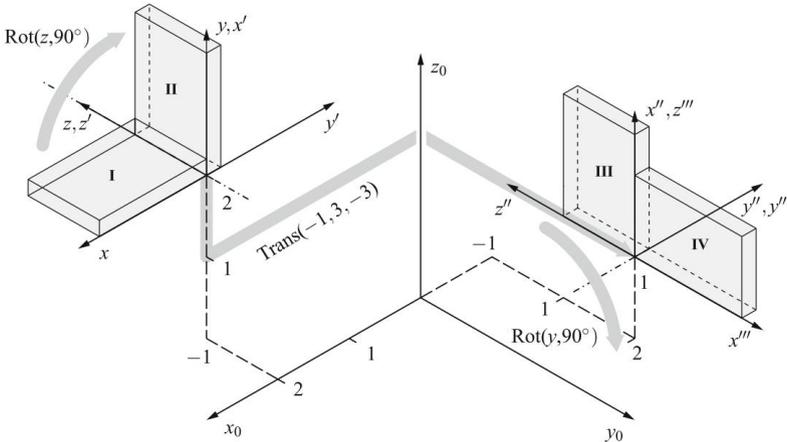


Abb. 2.7 Verschiebung des Objekts in eine neue Pose

$$\begin{aligned}
 H_3 = H \cdot D &= \begin{pmatrix} 1 & 0 & 0 & 2 \\ \ddot{y} & 0 & 0 & \ddot{y}_1 & \ddot{y}_1 \\ & 0 & 1 & 0 & 2 & 1 \\ \ddot{y} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & \ddot{y}_1 & 0 & \ddot{y}_3 \\ \ddot{y} & \ddot{y} & 0 & 0 & 1 & \ddot{y}_1 \\ \ddot{y} & \ddot{y} & 100 & \ddot{y}_3 & 0 & 0 & 0 & 1 \\ \ddot{y} & \ddot{y} & \ddot{y} & \ddot{y} & \ddot{y} & \ddot{y} \end{pmatrix} \\
 &= \begin{pmatrix} \ddot{y} & 0 & \ddot{y}_1 & 0 & \ddot{y}_1 \\ \ddot{y} & 100 & 2 & \ddot{y} & x_0 \\ & 0 & 0 & 1 & 1 & \ddot{y} & y_0 \\ \ddot{y} & 0 & 0 & 0 & 1 & \ddot{y} & z_0 \\ & \ddot{y} & 0 & 0 & 0 & 1 & \ddot{y} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ \ddot{y} \end{pmatrix} \tag{2.18}
 \end{aligned}$$

Wie im vorherigen Beispiel werden wir die Korrektheit der Matrix (2.18) grafisch überprüfen. Die drei Verschiebungen des Rahmens x–y–z: Drehung um 90° gegen den Uhrzeigersinn um die z-Achse, Translation um \ddot{y}_1 entlang der x-Achse, 3 Einheiten entlang der y-Richtung sind in Abb. 2.7 dargestellt. Das Ergebnis ist die endgültige Pose des Objekts. Die Achse zeigt in die positive Richtung der y_0 -Achse, in die positive Richtung der x_0 -Achse und in die negative Richtung der z_0 -Achse des Referenzrahmens. Die Richtungen der Achsen des endgültigen Rahmens entsprechen den ersten drei Spalten der Matrix H3. Es besteht auch Übereinstimmung zwischen der Position des Ursprungs des endgültigen Frames in Abb. 2.7 und der vierten Spalte der Matrix H3.

2.4 Geometrisches Robotermodell

Unser Endziel ist das geometrische Modell eines Roboterarmes. Ein geometrisches Robotermodell ergibt sich aus der Beschreibung der Pose des letzten Segments des Roboters (Endeffektor), ausgedrückt im Referenzrahmen (Basisrahmen). Das Wissen, wie man das beschreiben kann, ist das Ziel dieses Kapitels.

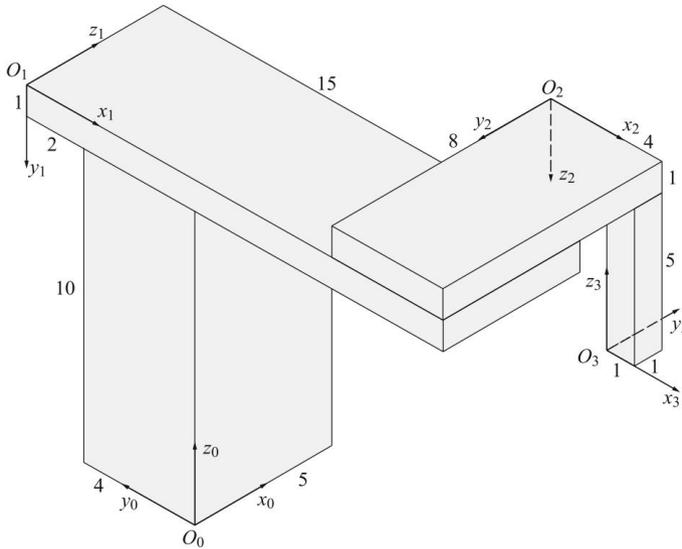


Abb. 2.8 Mechanische Montage

Die Pose eines Objekts mithilfe homogener Transformationsmatrizen wird zunächst auf den Montageprozess angewendet. Hierzu wird ein mechanischer Aufbau aus vier Blöcken betrachtet, wie er in Abb. 2.8 dargestellt ist. Eine Platte mit den Maßen $(5 \times 15 \times 1)$ wird über einen Block $(5 \times 4 \times 10)$ gelegt. Eine weitere Platte $(8 \times 4 \times 1)$ wird senkrecht zur ersten positioniert und hält einen weiteren kleinen Block $(1 \times 1 \times 5)$.

An jedem der vier Blöcke ist ein Rahmen befestigt, wie in Abb. 2.8 dargestellt. Unsere Aufgabe wird es sein, die Lage des Rahmens $x_3-y_3-z_3$ in Bezug auf den Referenzrahmen $x_0-y_0-z_0$ zu berechnen. Im letzten Kapitel haben wir gelernt, dass die Pose eines verschobenen Rahmens in Bezug auf den Referenzrahmen mithilfe der homogenen Transformationsmatrix H ausgedrückt werden kann. Die Pose des Rahmens $x_1-y_1-z_1$ in Bezug auf den Rahmen $x_0-y_0-z_0$ wird mit 0H_1 bezeichnet. Auf die gleiche Weise repräsentiert 1H_2 die Pose von Frame $x_2-y_2-z_2$ in Bezug auf Frame $x_1-y_1-z_1$ und 2H_3 die Pose von $x_3-y_3-z_3$ in Bezug auf Frame $x_2-y_2-z_2$. Wir haben auch gelernt, dass die aufeinanderfolgenden Verschiebungen durch Nachmultiplikationen (aufeinanderfolgende Multiplikationen von links nach rechts) homogener Transformationsmatrizen ausgedrückt werden können. Der Montageprozess kann durch Postmultiplikation der entsprechenden Matrizen beschrieben werden. Die Pose des vierten Blocks kann in Bezug auf den ersten durch die folgende Matrix geschrieben werden

$${}^0H_3 = {}^0H_1 {}^1H_2 {}^2H_3. \quad (2.19)$$

Die Blöcke wurden senkrecht zueinander positioniert. Auf diese Weise ist es nicht notwendig, die Sinus- und Cosinuswerte der Drehwinkel zu berechnen. Die Matrizen können direkt aus Abb. 2.8 ermittelt werden. Die x -Achse des Rahmens $x_1-y_1-z_1$ zeigt in die negative Richtung der y -Achse im Rahmen $x_0-y_0-z_0$. Die y -Achse des Rahmens $x_1-y_1-z_1$ zeigt nach innen

negative Richtung der z-Achse im Rahmen $x_0-y_0-z_0$. Die z-Achse des Rahmens $x_1-y_1-z_1$ hat die gleiche Richtung wie die x-Achse des Rahmens $x_0-y_0-z_0$. Die beschriebenen geometrischen Eigenschaften der Baugruppenstruktur werden in die ersten drei Spalten der homogenen Matrix geschrieben. Die Position des Ursprungs des Rahmens $x_1-y_1-z_1$ in Bezug auf den Rahmen $x_0-y_0-z_0$ wird in die vierte Spalte geschrieben

$${}^0H_1 = \begin{matrix} & \begin{matrix} 0 & 1 \\ - & - \end{matrix} \\ & \begin{matrix} x & y & z \end{matrix} \\ \begin{matrix} \ddot{y} & \ddot{y} & 1 & 0 & 0 & 6 \\ 0 & \ddot{y} & 1 & 0 & 1 & 1 \\ \ddot{y} & 0 & 0 & 0 & 1 & \ddot{y} \end{matrix} & \begin{matrix} \ddot{y} & \ddot{y} & 0 & 0 \\ j & \ddot{y} & 0 & 0 \\ z & \ddot{y} & & \end{matrix} \end{matrix} \quad (2.20)$$

Auf die gleiche Weise werden die beiden anderen Matrizen bestimmt

$${}^1H_2 = \begin{matrix} & \begin{matrix} 1 & 0 & 0 & 1 & 1 \\ \ddot{y} & 0 & 0 & 1 & \ddot{y} & 1 \\ 0 & \ddot{y} & 1 & 0 & 8 \\ \ddot{y} & 0 & 0 & 0 & 1 & \ddot{y} \end{matrix} \end{matrix} \quad (2.21)$$

$${}^2H_3 = \begin{matrix} & \begin{matrix} 1 & 0 & 0 & 3 \\ \ddot{y} & 0 & \ddot{y} & 1 & 0 & 1 \\ 0 & 0 & \ddot{y} & 1 & 6 \\ \ddot{y} & 0 & 0 & 0 & 1 & \ddot{y} \end{matrix} \end{matrix} \quad (2.22)$$

Die Position und Ausrichtung des vierten Blocks in Bezug auf den ersten wird durch die 0H_3 -Matrix angegeben, die durch sukzessive Multiplikation der Matrizen (2.20), (2.21) und (2.22) erhalten wird.

$${}^0H_3 = \begin{matrix} & \begin{matrix} 0 & 1 & 0 & 7 \\ \ddot{y} & \ddot{y} & 1 & 0 & 0 & \ddot{y} & 8 \\ 0 & 0 & 1 & 6 & 0 \\ \ddot{y} & 0 & 0 & 1 & \ddot{y} \end{matrix} \end{matrix} \quad (2.23)$$

Die vierte Spalte der Matrix 0H_3 [7, $\ddot{y}8$, 6, 1] stellt die Position des Ursprungs des Rahmens $x_3-y_3-z_3$ in Bezug auf den Referenzrahmen $x_0-y_0-z_0$ dar. Die Genauigkeit der vierten Spalte kann anhand von Abb. 2.8 überprüft werden. Der Rotationsteil der Matrix 0H_3 repräsentiert die Ausrichtung des Rahmens $x_3-y_3-z_3$ in Bezug auf den Referenzrahmen $x_0-y_0-z_0$.

Stellen wir uns nun vor, dass sich die erste horizontale Platte relativ zum ersten vertikalen Block um die Achse 1 um den Winkel $\ddot{y}1$ dreht. Die zweite Platte dreht sich ebenfalls um die vertikale Achse 2 um den Winkel $\ddot{y}2$. Der letzte Block wird entlang des dritten um die Distanz

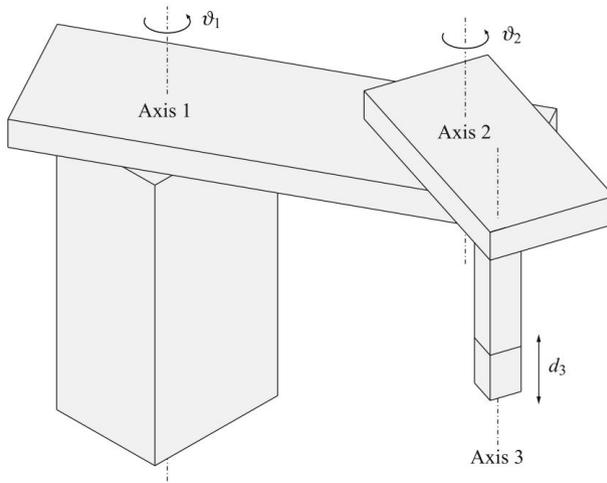


Abb. 2.9 Verschiebungen der mechanischen Baugruppe

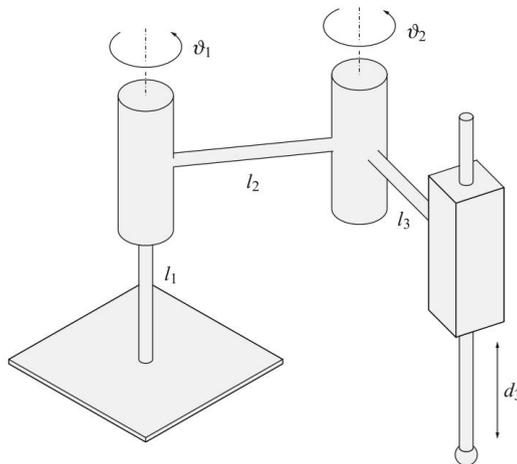


Abb. 2.10 SCARA-Roboterarm in beliebiger Pose

Achse. Auf diese Weise erhielten wir einen Roboterarm vom SCARA-Typ, wie im Einführungskapitel erwähnt.

Unser Ziel ist es, ein geometrisches Modell des SCARA-Roboters zu entwickeln. Blöcke und Platten aus Abb. 2.9 werden durch Symbole für Rotations- und Translationsgelenke ersetzt, die wir aus der Einleitung kennen (Abb. 2.10).

Das erste vertikale Segment mit der Länge l_1 beginnt an der Basis (wo der Roboter am Boden befestigt ist) und endet durch das erste Drehgelenk. Das zweite Segment mit der Länge l_2 ist horizontal und dreht sich um das erste Segment. Die Drehung im ersten Gelenk wird mit dem Winkel ϑ_1 bezeichnet. Das dritte Segment mit der Länge l_3 ist ebenfalls

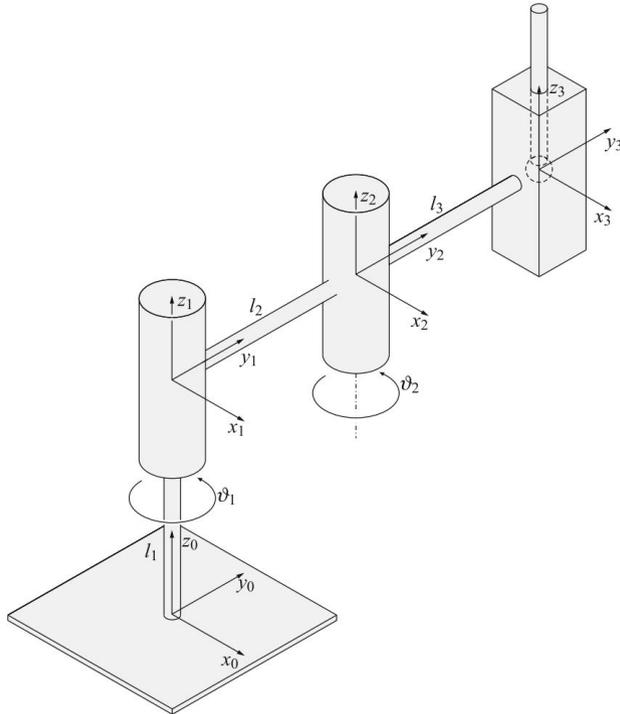


Abb. 2.11 Der SCARA-Roboterarm in der Ausgangsposition

horizontal und dreht sich am Ende des zweiten Segments um die vertikale Achse. Der Winkel wird mit ϑ_2 bezeichnet. Am Ende des dritten Segments befindet sich ein Translationsgelenk. Dadurch kann sich der Endeffektor des Roboters der Arbeitsebene nähern, auf der die Roboteranwendung ausgeführt wird. Das Translationsgelenk wird von der Anfangslänge Null auf die durch die Variable d_3 beschriebene Länge verschoben.

Der Robotermechanismus wird zunächst in die Ausgangsposition gebracht, die auch „Home-Position“ genannt wird. In der Ausgangsstellung müssen zwei benachbarte Segmente entweder parallel oder senkrecht sein. Die translatorischen Gelenke befinden sich in ihrer Ausgangsstellung $d_i = 0$. Die Ausgangsstellung des SCARA-Manipulators ist in Abb. 2.11 dargestellt.

Zunächst müssen die Koordinatensysteme in den in Abb. 2.11 dargestellten SCARA-Roboter eingezeichnet werden. Der erste (Referenz-)Koordinatenrahmen x_0 - y_0 - z_0 wird auf der Basis des Roboters platziert. Im letzten Kapitel werden wir erfahren, dass Roboterstandards erfordern, dass die z_0 -Achse senkrecht von der Basis weg zeigt. In diesem Fall wird es am ersten Segment ausgerichtet. Die anderen beiden Achsen werden so ausgewählt, dass die Robotersegmente parallel zu einer der Achsen des Referenzkoordinatensystems liegen, wenn sich der Roboter in seiner anfänglichen Ausgangsposition befindet. In diesem Fall richten wir die y_0 -Achse an. Der Koordinatenrahmen muss rechtshändig sein. Die restlichen Rahmen werden in die Robotersegmente eingesetzt. Die Ursprünge der Rahmen werden in der Mitte jedes Gelenks eingezeichnet. Eins

der Rahmenachsen müssen mit der Gelenkachse fluchten. Der einfachste Weg, das geometrische Modell eines Roboters zu berechnen, besteht darin, alle Rahmen in den Roboterjunkten parallel zum Referenzrahmen auszurichten (Abb. 2.11).

Das geometrische Modell eines Roboters beschreibt die Haltung des am Endeffektor befestigten Rahmens in Bezug auf den Referenzrahmen auf der Roboterbasis. Ebenso wie im Fall der mechanischen Baugruppe erhalten wir das geometrische Modell durch sukzessive Multiplikation (Postmultiplikation) homogener Transformationsmatrizen. Der Hauptunterschied zwischen der mechanischen Baugruppe und dem Roboter manipulator besteht in der Verschiebung der Roboterjunkte. Zu diesem Zweck folgt auf jede Matrix ${}^i\mathbf{H}_i$, die die Pose eines Segments beschreibt, eine Matrix D_i , die die Verschiebung entweder des Translations- oder des Rotationsgelenks darstellt. Unser SCARA-Roboter hat drei Gelenke. Die Lage des Endrahmens $x_3-y_3-z_3$ in Bezug auf den Basisrahmen $x_0-y_0-z_0$ wird durch die folgende Postmultiplikation von drei Paaren homogener Transformationsmatrizen

$${}^0\mathbf{H}_3 = ({}^0\mathbf{H}_1 D_1) \cdot ({}^1\mathbf{H}_2 D_2) \cdot ({}^2\mathbf{H}_3 D_3). \quad (2.24)$$

In Gl. (2.24) beschreiben die Matrizen ${}^0\mathbf{H}_1$, ${}^1\mathbf{H}_2$ und ${}^2\mathbf{H}_3$ die Stellung jedes Gelenkrahmens in Bezug auf den vorhergehenden Rahmen auf die gleiche Weise wie beim Zusammenbau der Blöcke. Aus Abb. 2.11 ist ersichtlich, dass die D_1 -Matrix eine Drehung um die positive z_1 -Achse darstellt. Das folgende Produkt zweier Matrizen beschreibt die Pose und die Verschiebung im ersten Gelenk

$${}^0\mathbf{H}_1 D_1 = \begin{array}{ccc} & \begin{array}{ccc} 1 & 0 & 0 & 0 \\ \ddot{y} & 0 & 1 & 0 \\ \vdots & 0 & 0 & 1 & 1 \\ \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \begin{array}{ccc} c_1 \ddot{y}_s 100 & & \\ \ddot{y} \ddot{y} & \begin{array}{ccc} s_1 & c_1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 \\ \ddot{y} \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \ddot{y} \end{array} = \begin{array}{ccc} & \begin{array}{ccc} c_1 \ddot{y}_s 100 & & \\ \ddot{y} & \begin{array}{ccc} s_1 & c_1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 1 \\ \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \ddot{y} \end{array} \end{array}$$

In den obigen Matrizen wurde die folgende kürzere Notation verwendet: $\sin \ddot{y}_1 = s_1$ und $\cos \ddot{y}_1 = c_1$.

Im zweiten Gelenk erfolgt eine Drehung um die z_2 -Achse

$${}^1\mathbf{H}_2 D_2 = \begin{array}{ccc} & \begin{array}{ccc} 1 & 0 & 0 & 0 \\ \ddot{y} & 0 & 1 & 0 & l_2 \\ \vdots & 0 & 0 & 1 & 0 \\ \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \begin{array}{ccc} c_2 \ddot{y}_s 200 & & \\ \ddot{y} \ddot{y} & \begin{array}{ccc} s_2 & c_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 \\ \ddot{y} \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \ddot{y} \end{array} = \begin{array}{ccc} & \begin{array}{ccc} c_2 \ddot{y}_s 200 & & \\ \ddot{y} & \begin{array}{ccc} s_2 & c_2 & 0 & l_2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \ddot{y} \end{array} \end{array}$$

Im letzten Gelenk erfolgt eine Translation entlang der z_3 -Achse

$${}^2\mathbf{H}_3 D_3 = \begin{array}{ccc} & \begin{array}{ccc} 1 & 0 & 0 & 0 \\ \ddot{y} & 0 & 1 & 0 & l_3 & 0 & 0 & 1 \\ \vdots & \vdots \\ \ddot{y} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} & \begin{array}{ccc} 100 & 0 & \\ \ddot{y} \ddot{y} & \begin{array}{ccc} 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \ddot{y} d_3 \\ \ddot{y} \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \ddot{y} \end{array} = \begin{array}{ccc} & \begin{array}{ccc} 100 & 0 & \\ \ddot{y} & \begin{array}{ccc} 0 & 1 & 0 & l_3 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \ddot{y} d_3 \\ \ddot{y} & 0 & 0 & 0 & 1 \end{array} & \ddot{y} \end{array} \end{array}$$

Das geometrische Modell des SCARA-Roboter manipulators wird durch Postmultiplikation der drei oben abgeleiteten Matrizen erhalten

$${}^0H_3 = \begin{pmatrix} c12 \ddot{y}s12 & 0 & \ddot{y}l3s12 & \ddot{y}l2s1 \\ \ddot{y} s12 & c12 & 0 & l3c12 + l2c1 & 0 \\ \ddot{y} & 0 & 1 & l1 & \ddot{y} & d3 & 0 & 0 & 0 & 1 \\ \ddot{y} & & & & & & & & & \ddot{y} \end{pmatrix}$$

Bei der Multiplikation der drei Matrizen wurde die Abkürzung $c12 = \cos(\ddot{y}1 + \ddot{y}2) = c1c2 \ddot{y} s1s2$ und $s12 = \sin(\ddot{y}1 + \ddot{y}2) = s1c2 + c1s2$ eingeführt.

Kapitel 3

Geometrische Beschreibung des Roboters Mechanismus



Die geometrische Beschreibung des Robotermechanismus basiert auf der Verwendung translatorischer und rotatorischer homogener Transformationsmatrizen. An der Roboterbasis und an jedem Segment des Mechanismus ist ein Koordinatenrahmen angebracht, wie in Abb. 3.1 dargestellt. Anschließend werden die entsprechenden Transformationsmatrizen zwischen den aufeinanderfolgenden Frames bestimmt. Ein in einem der Frames ausgedrückter Vektor kann durch sukzessive Multiplikation von Zwischentransformationsmatrizen in einen anderen Frame transformiert werden.

Der Vektor a in Abb. 3.1 wird relativ zum Koordinatenrahmen $x_3-y_3-z_3$ ausgedrückt, während der Vektor b im Rahmen $x_0-y_0-z_0$ angegeben wird, der zur Roboterbasis gehört. Eine mathematische Beziehung zwischen den beiden Vektoren wird durch die folgende homogene Transformation erhalten

$$\begin{bmatrix} b \\ 1 \end{bmatrix} = \begin{bmatrix} 0H1 & 1H2 & 2H3 \\ & & \end{bmatrix} \begin{bmatrix} A \\ 1 \end{bmatrix} \quad (3.1)$$

3.1 Vektorparameter eines kinematischen Paares

Vektorparameter werden zur geometrischen Beschreibung eines Robotermechanismus verwendet. Der Einfachheit halber beschränken wir unsere Betrachtung auf Mechanismen mit entweder parallel oder senkrecht aufeinanderfolgenden Gelenkachsen. Solche Mechanismen kommen in der Industrierobotik mit Abstand am häufigsten vor.

In Abb. 3.2 ist ein kinematisches Paar dargestellt, das aus zwei aufeinanderfolgenden Segmenten eines Robotermechanismus besteht, Segment $i-1$ und Segment i . Die beiden Segmente sind durch das Gelenk i verbunden, das sowohl Translation als auch Rotation umfasst. Die relative Stellung des Gelenks wird durch den Segmentvektor b_{i-1} und den Einheitsgelenkvektor e_i bestimmt, wie in Abb. 3.2 dargestellt. Das Segment i kann bezüglich des Segments $i-1$ entlang des Vektors e_i um den Abstand d_i verschoben und um den Winkel θ_i um e_i gedreht werden. Die Koordinate

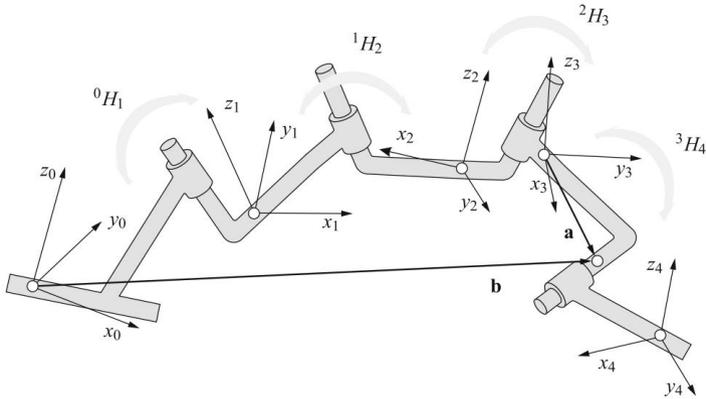


Abb. 3.1 Robotermechanismus mit an seinen Segmenten befestigten Koordinatenrahmen

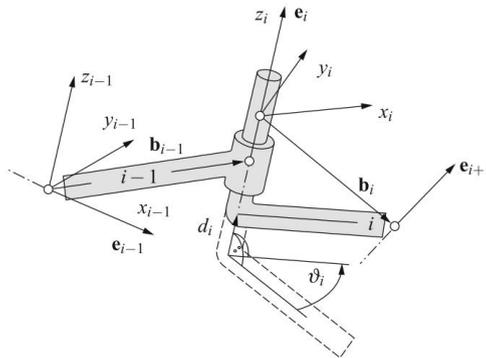


Abb. 3.2 Vektorparameter eines kinematischen Paares

Rahmen $x_i-y_i-z_i$ ist an das Segment i angehängt, während der Rahmen $x_{i-1}-y_{i-1}-z_{i-1}$ zum Segment $i-1$ gehört.

Der Koordinatenrahmen $x_i-y_i-z_i$ wird so in die Achse des Gelenks i gelegt, dass er parallel zum vorherigen Rahmen $x_{i-1}-y_{i-1}-z_{i-1}$ verläuft, wenn sich das kinematische Paar in seiner Ausgangsposition befindet (beide Gelenkvariablen sind Null ($\dot{y}_i = 0$ und $d_i = 0$)).

Die geometrischen Beziehungen und die relative Verschiebung zweier benachbarter Segmente Die Funktionen eines Robotermechanismus werden durch folgende Parameter bestimmt:

e_i – Einheitsvektor, der entweder die Rotationsachse oder die Translationsrichtung im Gelenk i beschreibt und als eine der Achsen des Rahmens $x_i-y_i-z_i$ ausgedrückt wird. Seine Komponenten sind die folgenden

$$e_i = \begin{matrix} 1 & 0 & 0 \\ \ddot{y} & 0 & \ddot{y} \\ \ddot{y} & 0 & \ddot{y} \end{matrix} \text{ oder } \begin{matrix} 0 & 1 & 0 \\ \ddot{y} & 1 & \ddot{y} \\ \ddot{y} & 0 & \ddot{y} \end{matrix} \text{ oder } \begin{matrix} 0 & 0 & 1 \\ \ddot{y} & 0 & \ddot{y} \\ \ddot{y} & 1 & \ddot{y} \end{matrix};$$

b_{i-1} – Segmentvektor, der das Segment $i-1$ beschreibt, ausgedrückt im Rahmen x_{i-1} – y_{i-1} – z_{i-1} . Seine Komponenten sind die folgenden

$$b_{i-1} = \begin{pmatrix} b_{i-1,x} \\ b_{i-1,y} \\ b_{i-1,z} \end{pmatrix};$$

θ_i – Rotationsvariable, die den Winkel darstellt, der um die e_i -Achse in der Ebene senkrecht zu e_i gemessen wird (der Winkel ist Null, wenn sich das kinematische Paar in der Ausgangsposition

findet); d_i – translatorische Variable, die den entlang der Richtung von e_i gemessenen Abstand darstellt (der Abstand ist gleich Null, wenn sich das kinematische Paar in der Ausgangspos

Wenn das Gelenk nur rotatorisch ist (Abb. 3.3 oben), wird die Gelenkvariable durch den Winkel θ_i dargestellt, während $d_i = 0$. Wenn sich der Robotermechanismus in seiner Ausgangsposition befindet, ist der Gelenkwinkel gleich Null $\theta_i = 0$ und die Koordinatensysteme x_{i-1} – y_{i-1} – z_{i-1} und x_i – y_i – z_i sind parallel. Wenn das Gelenk nur translatorisch ist (Abb. 3.3 unten), ist die Gele

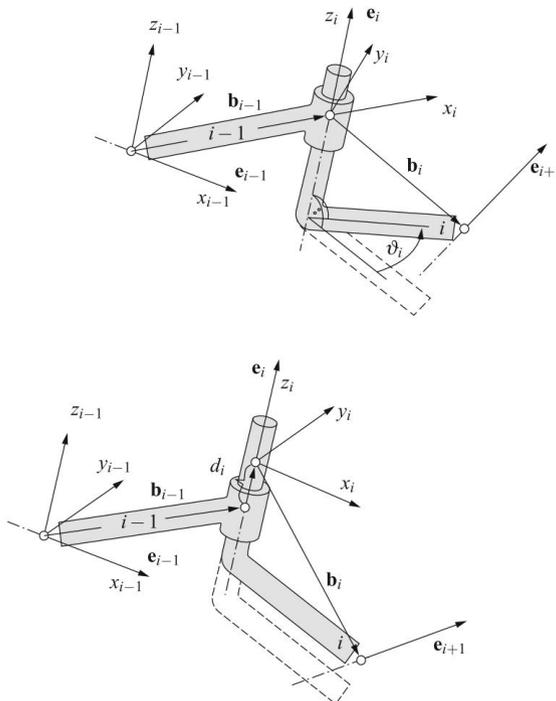


Abb. 3.3 Vektorparameter eines kinematischen Paares

während $\ddot{y}_i = 0$. Wenn sich das Gelenk in seiner Ausgangsposition befindet, dann ist $d_i = 0$. In diesem Fall sind die Koordinatensysteme $x_{i-1}-y_{i-1}-z_i$ und $x_{i\ddot{y}1}-y_{i\ddot{y}1}-z_{i\ddot{y}1}$ parallel, unabhängig vom Wert von translatorische Variable d_i .

Durch Ändern des Wertes der Drehgelenkvariablen \ddot{y}_i wird das Koordinatensystem $x_{i-1}-y_{i-1}-z_i$ zusammen mit dem Segment i gegenüber dem vorhergehenden Segment $i-1$ und dem entsprechenden System $x_{i\ddot{y}1}-y_{i\ddot{y}1}-z_{i\ddot{y}1}$ gedreht. Durch Ändern der translatorischen Variablen d_i ist die Verschiebung translatorisch, wobei sich nur der Abstand zwischen den beiden benachbarten Frames ändert.

Die Transformation zwischen den Koordinatensystemen $x_{i\ddot{y}1}-y_{i\ddot{y}1}-z_{i\ddot{y}1}$ und $x_{i-1}-y_{i-1}-z_i$ wird durch die homogene Transformationsmatrix bestimmt, die eine der drei möglichen Formen bezüglich der Richtung des Gelenkvektors e_i annimmt. Wenn der Einheitsvektor e_i parallel zur x_i -Achse ist, ist dies der Fall

$${}_{i\ddot{y}1}H_i = \begin{pmatrix} 1 & 0 & 0 & d_i + b_{i\ddot{y}1,x} & 0 & \cos \ddot{y}_i \\ \ddot{y}_i & \sin \ddot{y}_i & b_{i\ddot{y}1,y} & b_{i\ddot{y}1,z} & 1 & \ddot{y}_i \\ 0 & \sin \ddot{y}_i & \cos \ddot{y}_i & 0 & 0 & 0 \\ \ddot{y}_i & 0 & 0 & 0 & 0 & \ddot{y}_i \end{pmatrix}, \quad (3.2)$$

Wenn e_i parallel zur Y_i -Achse ist, haben wir die folgende Transformationsmatrix

$${}_{i\ddot{y}1}H_i = \begin{pmatrix} \cos \ddot{y}_i & 0 & \sin \ddot{y}_i & b_{i\ddot{y}1,x} \\ \ddot{y}_i & 0 & 10 & d_i + b_{i\ddot{y}1,y} & \ddot{y}_i \sin \ddot{y}_i \\ \ddot{y}_i & 0 & \cos \ddot{y}_i & b_{i\ddot{y}1,z} & 0 & 0 & 1 \\ \ddot{y}_i & 0 & 0 & 0 & 0 & 0 & \ddot{y}_i \end{pmatrix} \quad (3.3)$$

Wenn e_i parallel zur Z_i -Achse ist, hat die Matrix die folgende Form

$${}_{i\ddot{y}1}H_i = \begin{pmatrix} \cos \ddot{y}_i & \ddot{y}_i \sin \ddot{y}_i & 0 & b_{i\ddot{y}1,x} \\ \ddot{y}_i & \sin \ddot{y}_i & \cos \ddot{y}_i & 0 & b_{i\ddot{y}1,y} \\ 0 & 0 & 1 & d_i + b_{i\ddot{y}1,z} & 0 & 0 & 1 \\ \ddot{y}_i & 0 & 0 & 0 & 0 & 0 & \ddot{y}_i \end{pmatrix}. \quad (3.4)$$

In der Ausgangsstellung sind die Koordinatensysteme $x_{i\ddot{y}1}-y_{i\ddot{y}1}-z_{i\ddot{y}1}$ und $x_{i-1}-y_{i-1}-z_i$ parallel ($\ddot{y}_i = 0$ und $d_i = 0$) und nur für den Vektor $b_{i\ddot{y}1}$ verschoben

$${}_{i\ddot{y}1}H_i = \begin{pmatrix} 1 & 0 & 0 & b_{i\ddot{y}1,x} \\ \ddot{y}_i & 0 & 1 & b_{i\ddot{y}1,y} \\ 0 & 0 & 1 & b_{i\ddot{y}1,z} \\ \ddot{y}_i & 0 & 0 & 1 \end{pmatrix}. \quad (3.5)$$

3.2 Vektorparameter des Mechanismus

Die Vektorparameter eines Robotermechanismus werden in den folgenden vier Schritten bestimmt:

Schritt 1: Der Robotermechanismus wird in die gewünschte Ausgangsposition (Referenzposition) gebracht. Die Gelenkachsen müssen parallel zu einer der Achsen des an der Roboterbasis angebrachten Referenzkoordinatensystems $x_0-y_0-z_0$ sein. In der Referenzstellung sind alle Werte der Gelenkvariablen gleich Null, $\ddot{y}_i = 0$ und $d_i = 0$,

$i = 1, 2, \dots, n$; **Schritt 2** – die Mittelpunkte der Gelenke $i = 1, 2, \dots, n$ werden ausgewählt. Der Mittelpunkt des Gelenks i kann irgendwo entlang der entsprechenden Gelenkachse liegen. Ein lokaler Koordinatenrahmen $x_i-y_i-z_i$ wird so in die Gelenkmitte gelegt, dass seine Achsen parallel zu den Achsen des Referenzrahmens $x_0-y_0-z_0$ sind. Der lokale Koordinatenrahmen $x_i-y_i-z_i$ wird zusammen mit dem Segment i verschoben;

Schritt 3 – Der Einheitsgelenkvektor e_i wird jeder Gelenkachse $i = 1, 2, \dots, n$ zugewiesen. Es ist entlang einer der Achsen des Koordinatensystems $x_i-y_i-z_i$ gerichtet. In Richtung dieses Vektors wird die translatorische Größe d_i gemessen, während die rotatorische Größe \ddot{y}_i um den Gelenkvektor e_i ermittelt wird; **Schritt**

4 – Die Segmentvektoren b_i werden zwischen den Ursprüngen der Frames $x_i-y_i-z_i$, $i = 1, 2, \dots, n$ gezeichnet. Der Segmentvektor b_n verbindet den Ursprung des Rahmens $x_n-y_n-z_n$ mit dem Roboterendpunkt.

Manchmal wird ein zusätzlicher Koordinatenrahmen im Referenzpunkt eines Greifers positioniert und mit $x_{n+1}-y_{n+1}-z_{n+1}$ bezeichnet. Zwischen den Frames $x_n-y_n-z_n$ und $x_{n+1}-y_{n+1}-z_{n+1}$ besteht kein Freiheitsgrad, da beide Frames an das gleiche Segment gebunden sind. Die Transformation zwischen ihnen ist daher konstant.

Der Ansatz zur geometrischen Modellierung von Robotermechanismen wird anhand eines Beispiels eines Robotermechanismus mit vier Freiheitsgraden veranschaulicht, der in Abb. 3.4 dargestellt ist. Die gewählte Ausgangsstellung des Mechanismus zusammen mit den markierten Positionen der Gelenkzentren ist in Abb. 3.5 dargestellt. Die entsprechenden Vektorparameter und Gelenkvariablen sind in Tabelle 3.1 zusammengestellt.

Die Rotationsgrößen \ddot{y}_1 , \ddot{y}_2 und \ddot{y}_4 werden in den Ebenen senkrecht zu den Gelenkachsen e_1 , e_2 und e_4 gemessen, während die translatorische Größe d_i entlang der Achse e_3 gemessen wird. Ihre Werte sind Null, wenn sich der Robotermechanismus in seiner Ausgangsposition befindet. In Abb. 3.6 ist der Robotermechanismus in einer Pose dargestellt, in der alle vier Variablen positiv und ungleich Null sind. Die Variable \ddot{y}_1 repräsentiert den Winkel zwischen der anfänglichen und der momentanen y_1 -Achse, die Variable \ddot{y}_2 den Winkel zwischen der anfänglichen und der momentanen z_2 -Achse, die Variable d_3 ist der Abstand zwischen der anfänglichen und tatsächlichen Position der x_3 -Achse, während \ddot{y}_4 den Winkel zwischen der anfänglichen und momentanen z_4 -Achse darstellt.

Die ausgewählten Vektorparameter des Robotermechanismus werden in die homogenen Transformationsmatrizen (3.2)–(3.4) eingefügt.

$${}^{0H1} = \begin{pmatrix} c_1 \ddot{y}_1 & s_1 & 0 & 0 \\ \ddot{y}_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & h_0 \\ \ddot{y}_1 & 0 & 0 & 1 \end{pmatrix}$$

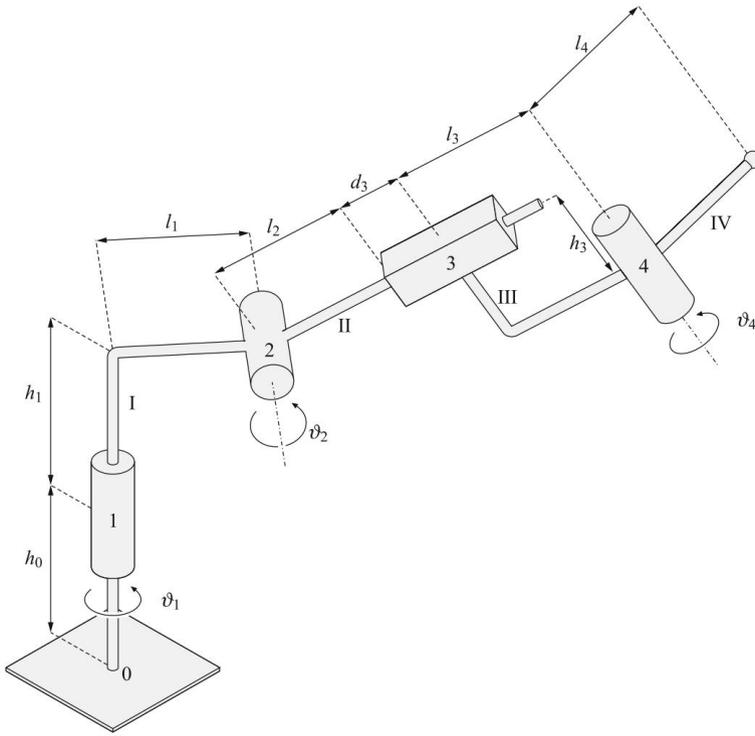


Abb. 3.4 Robotermechanismus mit vier Freiheitsgraden

$${}^1H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & \dot{y}s_2 & l_1 \\ 0 & s_2 & c_2 & h_1 \\ \dot{y} & 0 & 0 & 1 \end{pmatrix},$$

$${}^2H_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \dot{y} & 0 & 1 & d_3 + l_2 \\ 0 & 0 & 1 & 0 \\ \dot{y} & 0 & 0 & 1 \end{pmatrix},$$

$${}^3H_4 = \begin{pmatrix} c_4 & \dot{y}s_4 & 0 & 0 \\ s_4 & c_4 & 0 & l_3 \\ 0 & 0 & 1 & \dot{y}h_3 \\ \dot{y} & 0 & 0 & 1 \end{pmatrix}.$$

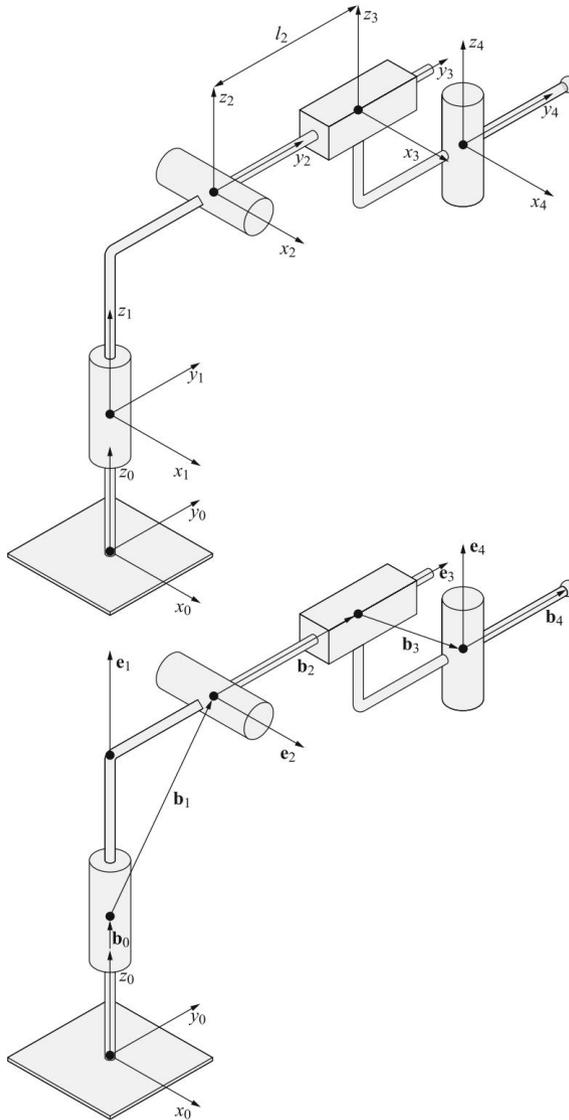


Abb. 3.5 Positionierung der Koordinatensysteme für den Robotermechanismus mit vier Freiheitsgraden

Eine zusätzliche homogene Matrix beschreibt die Position der Greiferreferenz Punkt, an dem der Koordinatenrahmen x_5 – y_5 – z_5 zugeordnet werden kann

$${}^4H_5 = \begin{matrix} \mathbf{1000} \\ \ddot{y} & \mathbf{010} & l_4 \\ & \mathbf{0010} \\ \ddot{y} & \mathbf{0001} \end{matrix} \begin{matrix} \ddot{y} \\ \vdots \\ \ddot{y} \end{matrix}$$

Tabelle 3.1 Vektorparameter und Gelenkvariablen für den Robotermechanismus in Abb. 3.5

ich 1234	
$\ddot{y}_i \dot{y}_1 \dot{y}_2 0 \dot{y}_4$	
di 0 0 d3 0	
ich 1234	
0100	
ei 0010	
1001	
..	1 23 4 5
0 00 0 0	
bi \dot{y}_1	0 l1 l2 l3 l4 h0 h1
	0 \dot{y}_h 3 0

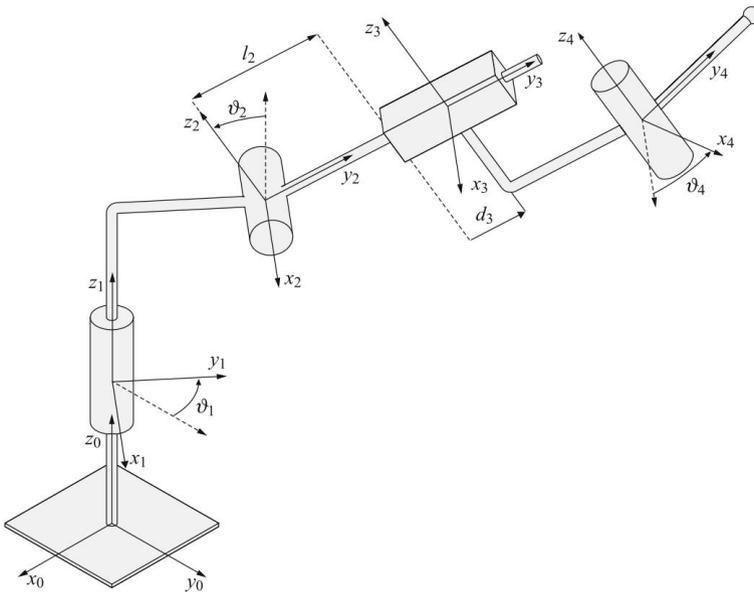


Abb. 3.6 Bestimmung der Rotations- und Translationsgrößen für den Robotermechanismus mit vier Freiheitsgraden

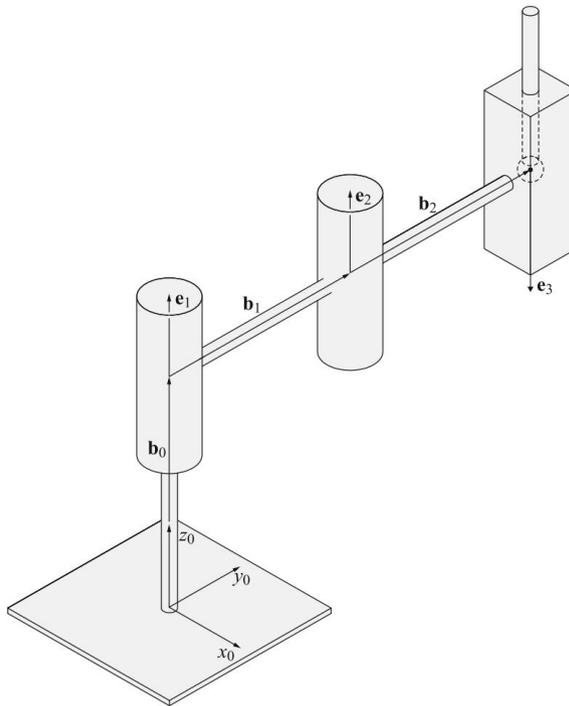


Abb. 3.7 Der SCARA-Roboterarm in der Ausgangsposition

Diese letzte Matrix ist konstant, da die Rahmen x_4 - y_4 - z_4 und x_5 - y_5 - z_5 parallel und um den Abstand l_4 verschoben sind. Normalerweise ist dieser zusätzliche Rahmen nicht einmal an der Robotermechanik befestigt, da die Position und Ausrichtung des Greifers im Rahmen x_4 - y_4 - z_4 beschrieben werden kann.

Bei der Bestimmung der Ausgangsposition (Ausgangsposition) des Robotermechanismus müssen wir darauf achten, dass die Gelenkachsen parallel zu einer der Achsen des Referenzkoordinatensystems liegen. Die Ausgangshaltung sollte so gewählt werden, dass sie einfach und leicht zu untersuchen ist, den zu erwartenden Roboteraufgaben gut entspricht und die Anzahl der erforderlichen mathematischen Operationen in den Transformationsmatrizen minimiert.

Als weiteres Beispiel betrachten wir den SCARA-Roboterarm, dessen geometrisches Modell bereits im vorherigen Kapitel entwickelt wurde und in Abb. 2.10 dargestellt ist.

Der Robotermechanismus sollte zunächst so in die Ausgangsposition gebracht werden, dass die Gelenkachsen parallel zu einer der Achsen des Bezugssystems x_0 - y_0 - z_0 liegen.

Auf diese Weise sind die beiden benachbarten Segmente entweder parallel oder senkrecht. Das Translationsgelenk muss sich in seiner Ausgangsposition befinden ($d_3 = 0$). Der SCARA-Roboter in der gewählten Ausgangshaltung ist in Abb. 3.7 dargestellt.

Die gemeinsamen Koordinatensysteme x_i - y_i - z_i sind alle parallel zum Referenzsystem. Deshalb zeichnen wir nur den Referenzrahmen und lassen die Punkte die Gelenkzentren anzeigen. In den Mittelpunkten beider Drehgelenke sind die Einheitsvektoren e_1 und e_2 platziert

Tabelle 3.2 Vektorparameter und Gelenkvariablen für den SCARA-Roboterarm

	ich 1234
	$\ddot{y}_1 \ddot{y}_2 0 \ddot{y}_4$
	di 0 0 d3 0
	ich 1234
	0100
ei	0010
	1001
	1 23 4 5
	0 00 0 0
bi \ddot{y}_1	0 l1 l2 l3 l4 h0 h1
	0 \ddot{y}_h 3 0

entlang der Gelenkachsen. Die Drehung um den e_1 - Vektor wird durch die Variable \ddot{y}_1 beschrieben , während \ddot{y}_2 den Winkel um den e_2 - Vektor darstellt . Der Vektor e_3 wird entlang der Translationsachse des dritten Gelenks platziert. Seine Übersetzungsvariable wird durch d_3 beschrieben. Das erste Gelenk ist über den Vektor b_0 mit der Roboterbasis verbunden . Vektor b_1 verbindet das erste und zweite Gelenk und Vektor b_2 das zweite und dritte Gelenk. Die Variablen und Vektoren sind in den drei Tabellen zusammengefasst (Tabelle 3.2).

In unserem Fall sind alle e_i - Vektoren parallel zur z_0 - Achse, die homogenen Transformationsmatrizen werden daher nach Gl. geschrieben. (3.4). Für beide Drehgelenke erhält man ähnliche Matrizen.

$${}^0H_1 = \begin{matrix} & c_1 \ddot{y}_s 100 & & \\ \ddot{y} & s_1 c_1 00 & \ddot{y} & \\ & 0 01 l_1 & & \\ \ddot{y} & 0 0 01 & & \ddot{y} \end{matrix}$$

$${}^1H_2 = \begin{matrix} & c_2 \ddot{y}_s 200 & & \\ \ddot{y} & s_2 c_2 0 l_2 & \ddot{y} & \\ & 0 0 10 & & \\ \ddot{y} & 0 0 01 & & \ddot{y} \end{matrix}$$

Für das translatorische Gelenk muss $\ddot{y}_3 = 0$ in Gl. eingesetzt werden. (3.4), Geben

$${}^2H_3 = \begin{pmatrix} 1 & 0 & 0 \\ \ddot{y} & 0 & 1 \\ 0 & 0 & 1 \\ \ddot{y} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ l_3 \\ \ddot{y}d_3 \\ 1 \end{pmatrix}$$

Durch Nachmultiplikation aller drei Matrizen erhält man das geometrische Modell des SCARA-Roboters

$${}^0H_3 = {}^0H_1 {}^1H_2 {}^2H_3 = \begin{pmatrix} c_{12} \ddot{y}s_{12} & 0 & \ddot{y}l_3s_{12} & \ddot{y}l_2s_1 \\ \ddot{y}s_{12} & c_{12} & 0 & l_3c_{12} + l_2c_1 \\ 0 & 1 & \ddot{y}d_3 & 0 \\ \ddot{y} & 0 & 0 & 1 \end{pmatrix}$$

Wir haben das gleiche Ergebnis wie im vorherigen Kapitel erhalten, jedoch auf viel einfachere und klarere Weise.

Kapitel 4

Orientierung



Wir beschreiben unsere Umwelt oft als eine dreidimensionale Welt. Die Welt des Robotikers ist jedoch sechsdimensional. Er muss nicht nur die Position eines Objekts berücksichtigen, sondern auch seine Ausrichtung. Wenn sich ein Robotergreifer oder Endeffektor einem zu greifenden Objekt nähert, sind die Raumwinkel zwischen Greifer und Objekt von größter Bedeutung.

Um die Position und Ausrichtung eines Objekts im Raum vollständig zu beschreiben, sind sechs Parameter erforderlich. Drei Parameter beziehen sich auf die Position und die anderen drei auf die Ausrichtung des Objekts. Es gibt drei Möglichkeiten, die Ausrichtung des Objekts mathematisch zu beschreiben. Die erste Möglichkeit ist eine Rotations-/Orientierungsmatrix bestehend aus neun Elementen. Die Matrix stellt eine redundante Beschreibung der Orientierung dar. Eine nichtredundante Beschreibung erfolgt durch RPY- oder Euler-Winkel. In beiden Fällen haben wir drei Winkel. Die RPY-Winkel werden um die Achsen eines festen Koordinatenrahmens definiert, während die Euler-Winkel die Ausrichtung um einen relativen Koordinatenrahmen beschreiben. Die dritte mögliche Beschreibung der Orientierung wird durch vier Parameter der Quaternion ermöglicht.

Im zweiten Kapitel haben wir bereits Rotationsmatrizen um die x-, y- und z-Achse eines rechteckigen Rahmens kennengelernt. Wir fanden sie bei der Entwicklung des geometrischen Modells eines Robotermechanismus nützlich. Es ist nicht schwer zu verstehen, dass es auch eine Matrix gibt, die die Drehung um eine beliebige Achse beschreibt. Dies kann in der folgenden Form ausgedrückt werden

$${}^0R_1 = \begin{pmatrix} 1 & 0 & 0 \\ \tilde{y} & 1 & 0 \\ \tilde{z} & 0 & 1 \end{pmatrix} \quad (4.1)$$

Die Matrix der Dimension 3×3 repräsentiert nicht nur die Drehung, sondern auch die Orientierung des Rahmens $x_1-y_1-z_1$ bezüglich des Rahmens $x_0-y_0-z_0$, wie aus Abb. 4.1 ersichtlich ist. Das Referenzsystem $x_0-y_0-z_0$ wird durch die Einheitsvektoren beschränkt.

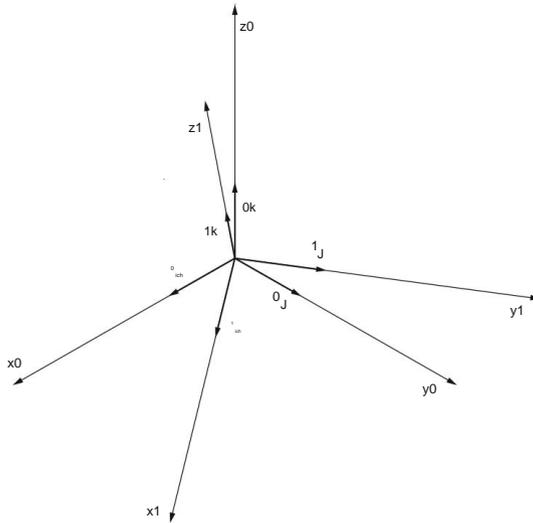


Abb. 4.1 Ausrichtung des Koordinatensystems $x_1-y_1-z_1$ in Bezug auf das Referenzkoordinatensystem $x_0-y_0-z_0$

0_{i0} , 0_j und 0_k und der gedrehte Rahmen $x_1-y_1-z_1$ mit den Einheitsvektoren 1_i und 1_k . Beide Koordinatensysteme fallen im selben Ursprung zusammen. Da es sich um die Einheitsvektoren handelt, sind die Elemente der Rotations-/Orientierungsmatrix einfach die Kosinusse der Winkel, die zu jedem Achsenpaar gehören.

Betrachten wir das Beispiel aus Abb. 4.2 und berechnen wir die Matrix, die die Ausrichtung des Rahmens $x_1-y_1-z_1$ darstellt, der um den Winkel $+\tilde{y}$ gegenüber dem Rahmen $x_0-y_0-z_0$ gedreht ist.

Wir haben es mit den folgenden von Null verschiedenen Produkten der Einheitsvektoren zu tun

$$\begin{aligned}
 0_{i0} 1_i &= 1, \\
 0_j 1_j &= \cos \tilde{y}, \\
 0_k 1_k &= \cos \tilde{y}, \\
 0_j 1_k &= \tilde{y} \sin \tilde{y}, \quad 0_k 1_j = -\tilde{y} \sin \tilde{y}, \\
 1_i 1_j &= \sin \tilde{y}.
 \end{aligned} \tag{4.2}$$

Die Matrix, die die Ausrichtung des Rahmens $x_1-y_1-z_1$ in Bezug auf $x_0-y_0-z_0$ beschreibt, lautet daher

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ \tilde{y} \sin \tilde{y} & \cos \tilde{y} & \tilde{y} \sin \tilde{y} \\ \tilde{y} \sin \tilde{y} & \tilde{y} \sin \tilde{y} & \cos \tilde{y} \end{pmatrix} \tag{4.3}$$

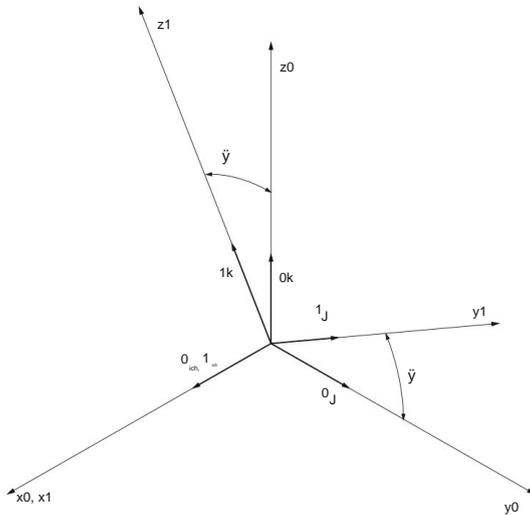
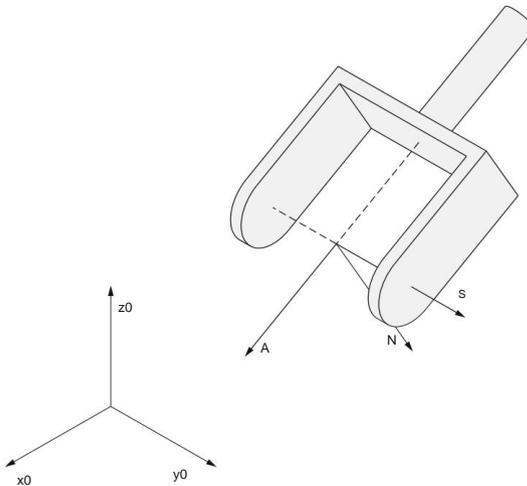
Abb. 4.2 Zwei um die x_0 -Achse gedrehte Koordinatensysteme

Abb. 4.3 Ausrichtung des Robotergreifers

Die Matrix (4.3) kann auch als Rotationsmatrix um die x -Achse interpretiert werden, die wir bereits als Teil der homogenen Matrix (2.6) aus dem zweiten Kapitel kennen.

Der Begriff der Orientierung bezieht sich in der Robotik meist auf die Ausrichtung des Robotergreifers. Zwischen zwei Fingern eines einfachen Robotergreifers wird ein Koordinatensystem mit drei Einheitsvektoren n , s und a platziert, die die Ausrichtung des Greifers beschreiben (Abb. 4.3).

Der z- Achsenvektor liegt in der Richtung der Annäherung des Greifers an das Objekt. Er wird daher mit dem Vektor \mathbf{a} (Ansatz) bezeichnet. Der an der Y -Achse ausgerichtete Vektor beschreibt die Gleitrichtung der Finger und wird als \mathbf{s} (slide) bezeichnet. Der dritte Vektor vervollständigt den rechtshändigen Koordinatenrahmen und wird als Normal bezeichnet. Dies kann als $\mathbf{n} = \mathbf{s} \times \mathbf{a}$ dargestellt werden. Die Matrix, die die Ausrichtung des Greifers in Bezug auf den Referenzrahmen $x_0-y_0-z_0$ beschreibt, hat die folgende Form

$$\mathbf{R} = \begin{pmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{pmatrix} \quad (4.4)$$

Das Element n_x der Matrix (4.3) bezeichnet die Projektion des Einheitsvektors \mathbf{n} auf die x_0 -Achse des Referenzrahmens. Er entspricht dem Kosinus des Winkels zwischen den Achsen \mathbf{i} und \mathbf{x}_0 und hat die gleiche Bedeutung wie die Elementmatrix⁰ der Rotation/Orientierung \mathbf{x} (4.1). Dasselbe gilt für die acht anderen Elemente der Orientierungsmatrix \mathbf{R} (4.3).

Um die Orientierung eines Objekts zu beschreiben, benötigen wir keine neun Elemente der Matrix. Der Vektor der linken Spalte ist das Kreuzprodukt der Vektoren \mathbf{s} und \mathbf{a} . Die Vektoren \mathbf{s} und \mathbf{a} sind Einheitsvektoren, die senkrecht zueinander stehen, sodass gilt

$$\begin{aligned} \mathbf{s} \cdot \mathbf{s} &= 1, \\ \mathbf{a} \cdot \mathbf{a} &= 1, \\ \mathbf{s} \cdot \mathbf{a} &= 0. \end{aligned} \quad (4.5)$$

Zur Beschreibung der Orientierung genügen also drei Elemente. Diese Orientierung wird oft durch die folgende Rotationsfolge beschrieben

R – Rollen – um die Z-
Achse, P – Nicken – um die Y
-Achse, Y – Gieren – um die X- Achse.

Diese Beschreibung wird meist verwendet, wenn die Ausrichtung eines Schiffs oder Flugzeugs beschrieben wird. Stellen wir uns vor, dass das Flugzeug entlang der Z- Achse fliegt und dass der Koordinatenrahmen in der Mitte des Flugzeugs positioniert ist. Dann stellt R die Drehung \tilde{y} um die z -Achse dar, P bezieht sich auf die Drehung \tilde{y} um die y -Achse und Y auf die Drehung \tilde{y} um die x- Achse, wie in Abb. 4.4 dargestellt.

Die Verwendung der RPY-Winkel für einen Robotergreifer ist in Abb. 4.5 dargestellt. Wie aus den Abb. ersichtlich ist. 4.4 und 4.5 wird die RPY-Ausrichtung in Bezug auf einen festen Koordinatenrahmen definiert. Bei der Entwicklung des geometrischen Modells des SCARA-Roboter manipulators im zweiten Kapitel haben wir die homogenen Transformationsmatrizen, die die Rotation (oder Translation) jedes einzelnen Gelenks beschreiben, nachmultipliziert. Die Position und Ausrichtung jedes Gelenkrahmens wurde in Bezug auf den vorhergehenden Rahmen definiert und bezieht sich auf die Gelenkachse, die nicht festgelegt ist. In diesem Fall, als

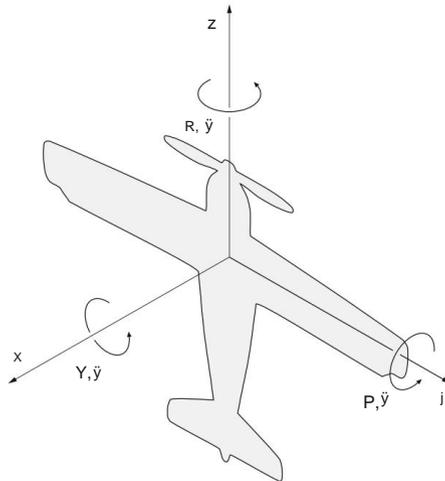


Abb. 4.4 RPY-Winkel für den Fall eines Flugzeugs

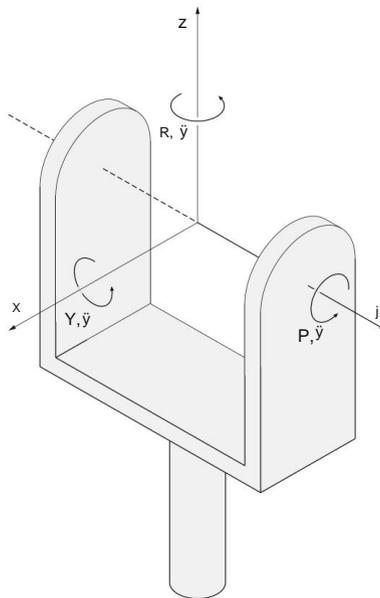


Abb. 4.5 RPY-Winkel für den Fall eines Robotergreifers

Wie wir gesehen haben, multiplizieren wir die Matrizen von links nach rechts. Wenn wir handeln bei aufeinanderfolgenden Drehungen um die Achsen desselben Koordinatensystems verwenden wir der Vormultiplikation der Rotationsmatrizen. Mit anderen Worten, die Multiplikationen werden in umgekehrter Reihenfolge von rechts nach links ausgeführt.

Wir beginnen mit der Drehung \tilde{y} um die z-Achse, fahren mit der Drehung \tilde{y} um die y-Achse fort und enden mit der Drehung \tilde{y} um die x-Achse. Die umgekehrte Reihenfolge der Drehungen ergibt sich auch aus der Benennung der RPY-Winkel. Die Orientierungsmatrix, die zu den RPY-Winkeln gehört, wird durch die folgende Multiplikation der Rotationsmatrizen

$$\begin{aligned}
 R(\tilde{y}, \tilde{y}, \tilde{y}) &= \text{Rot}(z, \tilde{y})\text{Rot}(y, \tilde{y})\text{Rot}(x, \tilde{y}) = c\tilde{y} \tilde{y}s\tilde{y} \mathbf{0} \\
 &= \begin{pmatrix} c\tilde{y} & \mathbf{0} & s\tilde{y} & \mathbf{0} \\ s\tilde{y} & c\tilde{y} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \tilde{y} & \tilde{y} & \tilde{y} \\ \tilde{y} & \tilde{y} & \tilde{y} \\ \tilde{y} & \tilde{y} & \tilde{y} \\ \tilde{y} & \tilde{y} & \tilde{y} \end{pmatrix} = \begin{pmatrix} c\tilde{y}c\tilde{y} & c\tilde{y}s\tilde{y}s\tilde{y} & \tilde{y} & s\tilde{y}c\tilde{y} \\ s\tilde{y}c\tilde{y} & s\tilde{y}s\tilde{y}s\tilde{y} & \tilde{y} & c\tilde{y}c\tilde{y} \\ \tilde{y} & \tilde{y} & c\tilde{y} & \tilde{y} \\ \tilde{y} & \tilde{y} & \tilde{y} & s\tilde{y} \end{pmatrix} \quad (4.6)
 \end{aligned}$$

Gleichung (4.6) berechnet die Rotationsmatrix aus den entsprechenden RPY-Winkeln.

Wir haben gelernt, dass Rotation und Orientierung entweder durch Rotationsmatrizen oder durch RPY-Winkel beschrieben werden können. Im ersten Fall benötigen wir 9 Parameter, während im letzteren Fall nur 3 Parameter erforderlich sind. Während Matrizen für Berechnungen praktisch sind, liefern sie jedoch kein schnelles und klares Bild beispielsweise der Ausrichtung eines Robotergreifers innerhalb eines Raums. RPY- und Euler-Winkel stellen die Ausrichtung eines Greifers zwar gut dar, sind für Berechnungen jedoch nicht geeignet. In diesem Kapitel werden wir lernen, dass Quaternionen entweder zur Berechnung oder Beschreibung der Orientierung geeignet sind.

Die Quaternionen stellen eine Erweiterung der komplexen Zahlen dar

$$z = a + ib, \quad (4.7)$$

wobei i die Quadratwurzel von -1 bedeutet, also $i^2 = -1$. Die komplexen Zahlen können geometrisch in einer Ebene dargestellt werden, indem ein rechteckiger Rahmen mit der Achse \tilde{y}_e (real) und \tilde{y}_m (imaginär) eingeführt wird. Beim Übergang von der Ebene in den Raum müssen zwei Einheitsvektoren j und k zu dem bereits vorhandenen i hinzugefügt werden. Es gilt auch die folgende Gleichung $i^2 = j^2 = k^2 = -1$. Die Quaternion hat die folgende Form

$$q = q_0 + q_1i + q_2j + q_3k. \quad (4.8)$$

In der Gl. (4.8) q_i sind reelle Zahlen, während i , j und k den Einheitsvektoren entlang der Achsen des rechteckigen Koordinatensystems entsprechen.

Bei der Beschreibung der Orientierung durch die RPY-Winkel waren die Multiplikationen der Rotationsmatrizen erforderlich. Auf ähnliche Weise müssen wir die Quaternionen multiplizieren

$$pq = (p_0 + p_1i + p_2j + p_3k)(q_0 + q_1i + q_2j + q_3k). \quad (4.9)$$

Tabelle 4.1 Regeln für Quaternion-Multiplikationen

*	1	i	j	k
1	1	i	j	k
i	i	-1	jk	-ji
j	j	ji	-1	ik
k	k	ki	ik	-1

Die Multiplikation von Quaternionen ist nicht kommutativ. Bei der Multiplikation zweier Quaternionen verwenden wir die Tabelle 4.1. Lassen Sie uns zwei Quaternionen multiplizieren

$$\begin{aligned}
 (2 + 3i \hat{y} j + 5k)(3 \hat{y} 4i + 2j + k) &= \\
 = 6 + 9i \hat{y} 3j + 15k\hat{y} & \\
 2 \hat{y} 8i \hat{y} 12i + 4ji \hat{y} 20ki + & \\
 + 4j + 6ij \hat{y} 2j &^2 + 10kj + \\
 + 2k + 3ik \hat{y} jk + 5k &^2 = \\
 = 6 + 9i \hat{y} 3j + 15k\hat{y} & \\
 \hat{y} 8i + 12 \hat{y} 4k \hat{y} 20j + & \\
 + 4j + 6k + 2 \hat{y} 10i + & \\
 + 2k \hat{y} 3j \hat{y} i \hat{y} 5 = & \\
 = 15 \hat{y} 10i \hat{y} 22j + 19k. &
 \end{aligned} \tag{4.10}$$

Der folgende Ausdruck einer Quaternion ist besonders geeignet, um das zu beschreiben Orientierung im Raum

$$q = \cos \frac{\hat{y}}{2} + \text{Sünde} \frac{\hat{y}}{2} \tag{4.11}$$

In der Gl. (4.11) ist ein Einheitsvektor, der an der Rotationsachse ausgerichtet ist, während \hat{y} der Winkel von ist Drehung. Die Orientierungsquaternion kann aus den RPY-Winkeln ermittelt werden. Drehung R wird durch die Quaternion beschrieben

$$qz\hat{y} = \cos \frac{\hat{y}}{2} + \text{Sünde} \frac{\hat{y}}{2} k. \tag{4.12}$$

Die folgende Quaternion gehört zur Rotation P

$$qy\hat{y} = \cos \frac{\hat{y}}{2} + \text{Sünde} \frac{\hat{y}}{2} j, \tag{4.13}$$

während Drehung Y wie folgt geschrieben werden kann

$$2 \quad \frac{\ddot{y}}{q_{xy}} = \cos \quad (4.14)$$

Nach der Multiplikation der oben genannten drei Quaternionen (4.12–4.14) erhält man die resultierende Orientierungsquaternion wird erhalten

$$q(\ddot{y}, \ddot{y}, \ddot{y}) = q_z \ddot{y} q_y \ddot{y} q_x \ddot{y} . \quad (4.15)$$

Lassen Sie uns die drei Beschreibungen der Orientierung, dh RPY-Winkel, Rotationsmatrix und Quaternionen, anhand eines Beispiels für die Beschreibung der Greiferorientierung veranschaulichen. Um das Beispiel klar und einfach zu machen, wird die Ebene des Zweifingergreifers in die x_0 - y_0 -Ebene des Referenzrahmens gelegt (Abb. 4.6). Die RPY-Winkel können aus Abb. 4.6 abgelesen werden. Die Drehungen um die z - und y -Achse sind gleich Null. Die Drehung um $\ddot{y}60\ddot{y}$ um die x -Achse ist aus Abb. 4.6 ersichtlich. Die Ausrichtung des Greifers kann daher durch den folgenden Satz von RPY-Winkeln besch

$$\ddot{y} = 0, \ddot{y} = 0, \ddot{y} = \ddot{y}60\ddot{y} . \quad (4.16)$$

Aus Abb. 4.6 können wir auch die Winkel zwischen den Achsen des Referenz- und Greiferkoordinatensystems ablesen. Ihre Kosinuswerte stellen die Orientierungs-/Rotationsmatrix R dar

$$\begin{aligned} n_x &= \cos 0\ddot{y}, s_x = \cos 90\ddot{y}, a_x = \cos 90\ddot{y} \quad n_y, \\ &= \cos 90\ddot{y}, s_y = \cos 60\ddot{y}, a_y = \cos 30\ddot{y} \quad n_z = , \\ \cos 0\ddot{y}, s_z &= \cos 150\ddot{y}, a_z = \cos 60\ddot{y} . \end{aligned} \quad (4.17)$$

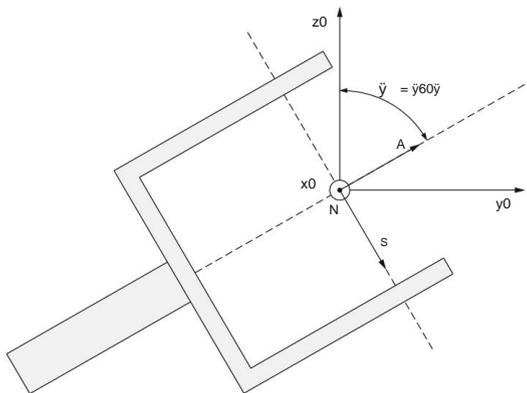


Abb. 4.6 Ausrichtung des Robotergreifers

Die Matrix R kann auch durch Einsetzen der bekannten RPY-Winkel in die Gleichung berechnet werden. (4.6)

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & \sin \gamma \\ 0 & -\sin \gamma & \cos \gamma \end{pmatrix} \quad (4.18)$$

Auf diese Weise wurde die Richtigkeit unserer Ableitung der Winkel aus Abb. 4.6 überprüft. Wir werden die Orientierungsquaternion berechnen, indem wir die RPY-Winkel in die Gleichungen einfügen. (4.12–4.14)

$$\begin{aligned} q_z \hat{y} &= 1 + 0\mathbf{k}, \\ q_y \hat{y} &= 1 + 0\mathbf{j}, \\ q_x \hat{y} &= 0,866 \hat{y} + 0,5\mathbf{i}. \end{aligned} \quad (4.19)$$

Die Orientierungsquaternion erhält man durch Multiplikation der drei obigen Quaternionen (4.15)

$$q_0 = 0,866, q_1 = 0,5, q_2 = 0, q_3 = 0. \quad (4.20)$$

Die Gleichungen (4.16), (4.18) und (4.20) demonstrieren drei verschiedene Beschreibungen der gleichen Greiferausrichtung.

Kapitel 5

Zwei-Segment-Roboter manipulator



5.1 Kinematik

Die Kinematik ist ein Teil der klassischen Mechanik, die Bewegung untersucht, ohne die Kräfte zu berücksichtigen, die für diese Bewegung verantwortlich sind. Bewegung wird im Allgemeinen durch Trajektorien, Geschwindigkeiten und Beschleunigungen beschrieben. In der Robotik interessieren uns vor allem Trajektorien und Geschwindigkeiten, da beides durch die Gelenksensoren gemessen werden kann. Bei Roboterjunkten werden die Trajektorien entweder als Winkel bei einem Rotationsgelenk oder als Distanz bei einem Translationsgelenk gemessen. Die Gelenkvariablen werden auch interne Koordinaten genannt. Bei der Planung und Programmierung einer Roboter Aufgabe ist die Flugbahn des Roboterendpunkts von größter Bedeutung. Die Position und Ausrichtung des Endeffektors werden durch externe Koordinaten beschrieben. Die Berechnung externer Variablen aus internen Variablen und umgekehrt ist das zentrale Problem der Roboterkinematik.

In diesem Kapitel beschränken wir unser Interesse auf einen planaren zweisegmentigen Roboter manipulator mit zwei Drehgelenken (Abb. 5.1). Nach der im Einleitungskapitel gegebenen Definition kann ein solcher Mechanismus kaum als Roboter bezeichnet werden. Dennoch ist dieser Mechanismus ein wichtiger Bestandteil der SCARA- und anthropomorphen Roboterstrukturen und wird es uns ermöglichen, mehrere charakteristische Eigenschaften der Bewegung von Robotermechanismen zu untersuchen.

Man unterscheidet zwischen direkter und inverser Kinematik. Direkte Kinematik stellt im Fall eines Zweisegmentroboters die Berechnung der Position des Roboterendpunkts aus den bekannten Gelenkwinkeln dar. Die inverse Kinematik berechnet die Gelenkvariablen aus der bekannten Position des Roboterendpunkts. Die direkte Kinematik stellt das einfachere Problem dar, da wir eine einzige Lösung für die Position des Roboterendpunkts haben. Die Lösungen der inversen Kinematik hängen maßgeblich von der Struktur des Roboter manipulators ab. Wir beschäftigen uns häufig mit mehreren Lösungen für die Gelenkvariablen, die zur gleichen Position des Roboterendpunkts führen, während in einigen Fällen keine analytische Lösung der inversen Kinematik existiert.

Die kinematische Analyse umfasst auch die Beziehungen zwischen der Geschwindigkeit des Roboterendpunkts und den Geschwindigkeiten einzelner Gelenke. Wir werden feststellen, dass die inverse Kinematik für Geschwindigkeiten einfacher ist als die inverse Kinematik für Trajektorien. Wir werden zuerst

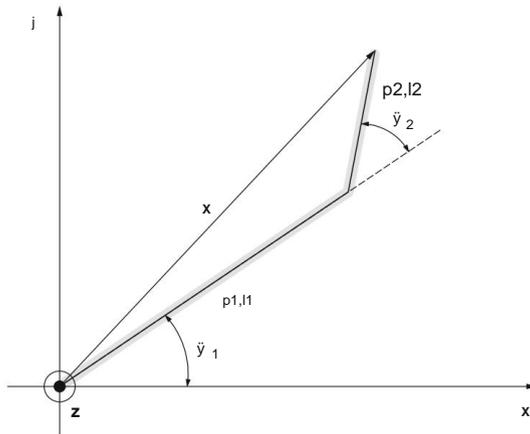


Abb. 5.1 Planarer Roboterarm mit zwei Segmenten

Lösung der direkten Kinematik für die Trajektorien. Durch Differentiation erhalten wir dann die Gleichungen, die die direkte Kinematik für die Geschwindigkeiten beschreiben. Durch einfache Matrixinversion kann die inverse Kinematik für Geschwindigkeiten berechnet werden. Betrachten wir nun den in Abb. 5.1 dargestellten planaren Roboterarm mit zwei Segmenten.

Die Rotationsachse des ersten Gelenks wird durch die vertikale Z-Achse dargestellt aus dem Flugzeug. Der Vektor p_1 ist entlang des ersten Segments gerichtet

$$p_1 = l_1 \begin{pmatrix} \cos \dot{y}_1 \\ \sin \dot{y}_1 \end{pmatrix} \quad (5.1)$$

Der Vektor p_2 liegt zusammen mit dem zweiten Segment. Seine Bestandteile sind aus Abb. 5.1 ablesbar

$$p_2 = l_2 \begin{pmatrix} \cos(\dot{y}_1 + \dot{y}_2) \\ \sin(\dot{y}_1 + \dot{y}_2) \end{pmatrix} \quad (5.2)$$

Der Vektor x verbindet den Ursprung des Koordinatenrahmens mit dem Endpunkt des Roboters

$$x = p_1 + p_2. \quad (5.3)$$

Wir haben also die Position des Roboterendpunkts

$$x = \begin{pmatrix} x \\ j \end{pmatrix} = \begin{pmatrix} l_1 \cos \dot{y}_1 + l_2 \cos(\dot{y}_1 + \dot{y}_2) \\ l_1 \sin \dot{y}_1 + l_2 \sin(\dot{y}_1 + \dot{y}_2) \end{pmatrix} \quad (5.4)$$

Durch die Definition des Vektors der Gelenkwinkel

$$q = \dot{y}_1 \ \dot{y}_2^T, \quad (5.5)$$

die Gl. (5.4) kann in der folgenden kürzeren Form geschrieben werden

$$\mathbf{x} = \mathbf{k}(\mathbf{q}), \quad (5.6)$$

wobei $\mathbf{k}(\cdot)$ die Gleichungen der direkten Kinematik darstellt.

Die Beziehung zwischen den Geschwindigkeiten des Roboterendpunkts und den Gelenkgeschwindigkeiten ist durch Differenzierung erhalten. Die Koordinaten des Endpunkts sind Funktionen des Gelenkwinkel, die wiederum Funktionen der Zeit sind

$$\begin{aligned} \mathbf{x} &= \mathbf{x}(\dot{y}_1(t), \dot{y}_2(t)) \\ \mathbf{y} &= \mathbf{y}(\dot{y}_1(t), \dot{y}_2(t)). \end{aligned} \quad (5.7)$$

Durch die Berechnung der Zeitableitungen von Gl. (5.7) und sie in Matrixform anordnen, wir können schreiben

$$\begin{matrix} \mathbf{X} \\ \mathbf{y}\ddot{\mathbf{y}} \end{matrix} = \begin{matrix} \frac{\dot{\mathbf{y}}\mathbf{x}}{\dot{y}_1} & \frac{\dot{\mathbf{y}}\mathbf{x}}{\dot{y}_2} \\ \frac{\dot{\mathbf{y}}\mathbf{y}}{\dot{y}_1} & \frac{\dot{\mathbf{y}}\mathbf{y}}{\dot{y}_2} \end{matrix} \begin{matrix} \ddot{y}_1 \\ \ddot{y}_2 \end{matrix}. \quad (5.8)$$

Für unseren zweisegmentigen Roboter manipulator erhalten wir den folgenden Ausdruck

$$\begin{matrix} \mathbf{X} \\ \mathbf{y}\ddot{\mathbf{y}} \end{matrix} = \begin{matrix} \dot{y}_1 s_1 \dot{y}_2 l_2 s_{12} \dot{y}_2 l_2 s_{12} \\ l_1 c_1 + l_2 c_1 l_2 c_{12} \end{matrix} \begin{matrix} \ddot{y}_1 \\ \ddot{y}_2 \end{matrix}. \quad (5.9)$$

Die Matrix, die in unserem Fall zweiter Ordnung ist, wird Jacobi-Matrix genannt $\mathbf{J}(\mathbf{q})$. Die Beziehung (5.9) kann in Kurzform geschrieben werden als

$$\mathbf{x}\ddot{\mathbf{y}} = \mathbf{J}(\mathbf{q})\mathbf{q}\ddot{\mathbf{y}}. \quad (5.10)$$

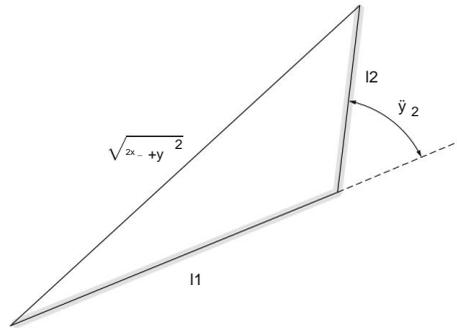
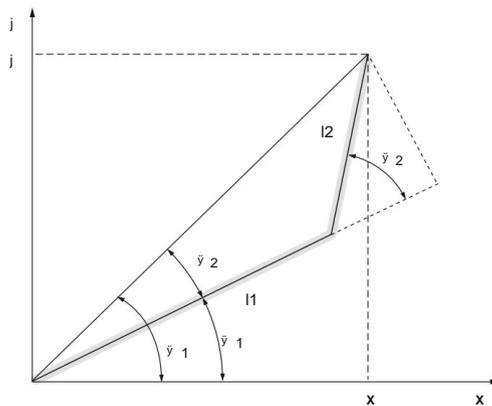
Auf diese Weise werden die Probleme der direkten Kinematik für Trajektorien und Geschwindigkeiten gelöst. Bei der Lösung der inversen Kinematik berechnen wir die Gelenkwinkel aus dem Bekannten Position des Roboterendpunkts. Abbildung 5.2 zeigt nur die Parameter des Zwei-Segment-Robotermechanismus, die für die Berechnung des \dot{y}_2 -Winkels relevant sind. Der

Es wird die Kosinusregel verwendet

$$z_{x_2} + J^2 = l_1^2 + l_2^2 \dot{y}_2^2 + 2l_1 l_2 \cos(180\dot{y}_1 - \dot{y}_2), \quad (5.11)$$

wobei $\dot{y}_2 \cos(180\dot{y}_1 - \dot{y}_2) = \cos(\dot{y}_2)$. Der Winkel des zweiten Segments des Zwei-Segment-Manipulators wird als umgekehrte trigonometrische Funktion berechnet

$$\dot{y}_2 = \arccos \frac{z_{x_2} + J^2 - l_1^2 - l_2^2 \dot{y}_1^2}{2l_1 l_2}. \quad (5.12)$$

Abb. 5.2 Berechnung des \ddot{y}_2 -WinkelsAbb. 5.3 Berechnung des \ddot{y}_1 -Winkels

Der Winkel des ersten Segments wird mit Hilfe von Abb. 5.3 berechnet. Es wird erhalten als Differenz der Winkel \ddot{y}_1 und \ddot{y}_2

$$\ddot{y}_1 = \ddot{y}_1 - \ddot{y}_2.$$

Der Winkel \ddot{y}_1 ergibt sich aus dem rechtwinkligen Dreieck aus der Horizontalen x und vertikale y -Koordinaten des Roboterendpunkts. Der Winkel \ddot{y}_2 ergibt sich durch Streckung das Dreieck aus Abb. 5.2 in das rechtwinklige Dreieck um, wie in Abb. 5.3 dargestellt. Wieder wir Nutzen Sie die inversen trigonometrischen Funktionen

$$\ddot{y}_1 = \text{Arctan} \frac{y}{x} - \text{arctan} \frac{l_2 \sin \ddot{y}_2}{l_1 + l_2 \cos \ddot{y}_2}. \quad (5.13)$$

Bei der Berechnung des \ddot{y}_2 -Winkels haben wir zwei Lösungen: Ellenbogen nach oben und Ellenbogen nach unten, wie in Abb. 5.4 dargestellt. Eine entartete Lösung wird durch den Endpunkt dargestellt

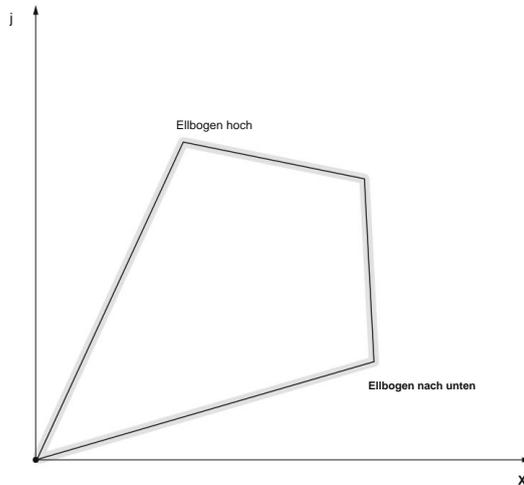


Abb. 5.4 Zwei Lösungen der inversen Kinematik

Position $x = y = 0$, wenn beide Segmente die gleiche Länge $l_1 = l_2$ haben. In diesem Fall kann die $\arctan \frac{y}{x}$ ist nicht definiert. Wenn der Winkel $\dot{y}_2 = 180^\circ$, Basis des einfachen Zwei-Segment-Mechanismus in einem beliebigen Winkel \dot{y}_1 erreicht werden. Wenn jedoch ein Punkt (x, y) außerhalb des Arbeitsbereichs des Manipulators liegt, kann das Problem der inversen Kinematik nicht gelöst werden.

Die Beziehung zwischen den Gelenkgeschwindigkeiten und der Geschwindigkeit des Endpunkts wird durch Invertieren der Jacobi-Matrix $J(q)$ erhalten.

$$\dot{q}\ddot{y} = J^{-1}(q)\dot{x}\ddot{y}. \quad (5.14)$$

Die Matrizen der Ordnung 2×2 können wie folgt invertiert werden

$$A = \begin{matrix} A & B \\ C & D \end{matrix} \quad A^{-1} = \frac{1}{ad - bc} \begin{matrix} d & -b \\ -c & a \end{matrix}$$

Für unseren Zwei-Segment-Manipulator können wir schreiben

$$\begin{matrix} \dot{y}_1 \\ \dot{y}_2 \end{matrix} = \frac{1}{l_1 l_2 s_2} \begin{matrix} l_2 c_2 l_2 s_2 \dot{y}_1 c_1 \dot{y}_2 \\ c_2 \dot{y}_1 s_1 \dot{y}_2 l_2 s_2 \end{matrix} \begin{matrix} x\dot{y} \\ y\dot{y} \end{matrix} \quad (5.15)$$

In allgemeinen Beispielen für Roboter-Manipulatoren ist es nicht notwendig, dass die Jacobi-Matrix die quadratische Form hat. In diesem Fall wird die sogenannte Pseudoinverse Matrix $J^+ (JJ^T)^{-1} J^T$ berechnet. Für einen Roboter mit sechs Freiheitsgraden ist die Jacobi-Matrix quadratisch, wird jedoch nach der Invertierung eher unpraktisch. Wenn der Manipulator nah ist

Für singuläre Posen (z. B. wenn der Winkel \dot{y}_2 für den einfachen Roboter mit zwei Segmenten nahe Null liegt) ist die inverse Jacobi-Matrix schlecht definiert. Wir werden die Jacobi-Matrix verwenden, wenn wir die Robotersteuerung untersuchen.

5.2 Statik

Lassen Sie uns nach dem Ende des Abschnitts zur Roboterkinematik einen kurzen Sprung zur Roboterstatik machen.

Nehmen wir an, dass der Endpunkt des zweisegmentigen Roboterarmes gegen ein Hindernis stößt (Abb. 5.5). Auf diese Weise übt der Roboter eine Kraft gegen das Hindernis aus. Die horizontale Kraftkomponente wirkt in positiver Richtung der x-Achse, während die vertikale Komponente entlang der y-Achse gerichtet ist. Die Kraft gegen das Hindernis wird durch die Motoren in den Roboterjunkten erzeugt. Der Motor des ersten Gelenks erzeugt das Drehmoment M_1 , während M_2 das Drehmoment im zweiten Gelenk ist.

Die positiven Richtungen beider Gelenkdrehmomente sind gegen den Uhrzeigersinn. Da sich der Roboter nicht bewegt, ist die Summe der äußeren Drehmomente gleich Null. Das bedeutet, dass das Drehmoment M_1 im ersten Gelenk gleich dem Drehmoment der äußeren Kraft ist bzw. gleich dem Drehmoment ist, das der Manipulator auf das Hindernis ausübt

$$M_1 = \dot{y}F_x y + F_y x. \quad (5.16)$$

Die Endpunktkoordinaten x und y , berechnet nach Gl. (5.4) werden in Gl. eingesetzt. (5.16)

$$M_1 = \dot{y}F_x (l_1 \sin \dot{y}_1 + l_2 \sin(\dot{y}_1 + \dot{y}_2)) + F_y (l_1 \cos \dot{y}_1 + l_2 \cos(\dot{y}_1 + \dot{y}_2)). \quad (5.17)$$

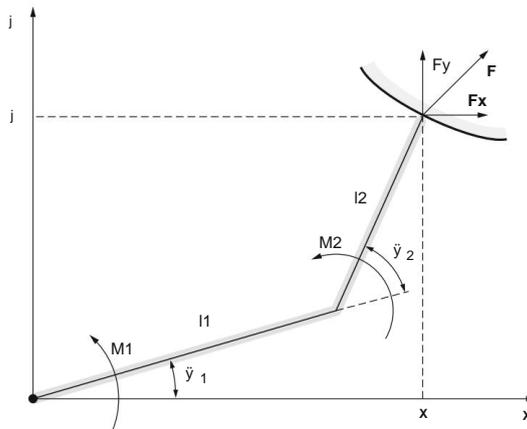


Abb. 5.5 Zweisegmentiger Roboterarm im Kontakt mit der Umgebung

Auf ähnliche Weise wird das Drehmoment im zweiten Gelenk ermittelt

$$M_2 = \dot{y} F_x l_2 \sin(\dot{y}_1 + \dot{y}_2) + F_y l_2 \cos(\dot{y}_1 + \dot{y}_2). \quad (5.18)$$

Die Gleichungen (5.17) und (5.18) können in Matrixform geschrieben werden

$$\begin{matrix} M_1 \\ M_2 \end{matrix} = \begin{matrix} \dot{y}_1 l_1 s_1 \dot{y}_2 l_2 s_{12} l_1 c_1 + l_2 c_{12} \dot{y}_2 \\ s_{12} l_2 c_{12} \end{matrix} \begin{matrix} F_x \\ F_y \end{matrix} \quad (5.19)$$

Die Matrix in Gl. (5.19) ist eine transponierte Jacobi-Matrix. Die transponierte Matrix der Ordnung 2×2 hat die folgende Form

$$A = \begin{matrix} A & B \\ C & D \end{matrix} \quad A^T = \begin{matrix} a & c \\ b & d \end{matrix}$$

Auf diese Weise erhielten wir einen wichtigen Zusammenhang zwischen den Gelenkdrehmomenten und den Kräften am Roboter-Endeffektor

$$\ddot{y} = J^T(q) f, \quad (5.20)$$

Wo

$$\ddot{y} = \begin{matrix} M_1 \\ M_2 \end{matrix} \quad f = \begin{matrix} F_x \\ F_y \end{matrix}$$

Gleichung (5.20) beschreibt die Roboterstatik. Es wird zur Steuerung eines Roboters eingesetzt, der mit der Umwelt in Kontakt steht.

5.3 Arbeitsbereich

Der Roboterarbeitsbereich besteht aus allen Punkten, die vom Roboterendpunkt erreicht werden können. Es spielt eine wichtige Rolle bei der Auswahl eines Industrieroboters für eine erwartete Aufgabe. Unser Ziel ist es, einen Ansatz zur Bestimmung des Arbeitsbereichs eines ausgewählten Roboters zu beschreiben. Wir betrachten noch einmal das Beispiel des einfachen planaren Zweisegmentroboters mit Drehgelenken. Unsere Untersuchung des Roboterarbeitsraums wird daher in einer Ebene stattfinden und wir werden es tatsächlich mit einer Arbeitsfläche zu tun haben. Unabhängig von den durch die Ebene auferlegten Einschränkungen werden wir uns der wichtigsten charakteristischen Eigenschaften der Roboterarbeitsräume bewusst. Industrieroboter verfügen in der Regel über die Fähigkeit, sich um die erste vertikale Gelenkachse zu drehen. Wir werden daher die Arbeitsfläche um die vertikale Achse des Referenzkoordinatensystems drehen und so eine Vorstellung von realistischen dreidimensionalen Roboterarbeitsräumen erhalten.

Betrachten wir den planaren Roboterarm mit zwei Segmenten, wie in Abb. 5.6 dargestellt. Die Rotationsfreiheitsgrade werden mit \dot{y}_1 und \dot{y}_2 bezeichnet. Die Längen der Segmente l_1 und l_2 werden als gleich betrachtet. Die Koordinaten des Roboterendpunkts können wie in (5.4) mit den folgenden zwei Gleichungen ausgedrückt werden:

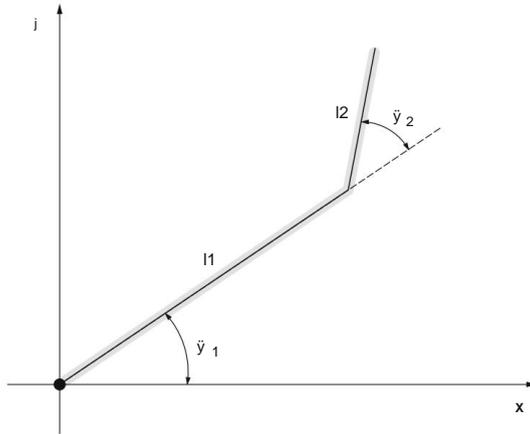


Abb. 5.6 Zweisegmentiger Roboterarm

$$\begin{aligned} x &= l_1 \cos \dot{y}_1 + l_2 \cos(\dot{y}_1 + \dot{y}_2) \\ y &= l_1 \sin \dot{y}_1 + l_2 \sin(\dot{y}_1 + \dot{y}_2) \end{aligned} \quad (5.21)$$

Wenn Gl. (5.21) zunächst quadriert und dann summiert, erhält man die Gleichungen eines Kreises

$$\begin{aligned} (x - l_1 \cos \dot{y}_1)^2 + (y - l_1 \sin \dot{y}_1)^2 &= l_2^2 \\ x^2 + y^2 - 2x l_1 \cos \dot{y}_1 - 2y l_1 \sin \dot{y}_1 + l_1^2 &= l_2^2 \end{aligned} \quad (5.22)$$

Die erste Gleichung hängt nur vom Winkel \dot{y}_1 ab, während in der zweiten Gleichung nur \dot{y}_2 vorkommt. Das Netz der für verschiedene Werte \dot{y}_1 und \dot{y}_2 aufgetragenen Kreise ist in Abb. 5.7 dargestellt. Die erste Gleichung beschreibt die Kreise, die in Abb. 5.7 mit $\dot{y}_1 = 0$ bezeichnet sind, 30° , 60° , 90° , 120° , 150° , und 180° . Ihre Radien sind gleich der Länge des zweiten Segments l_2 , die Mittelpunkte der Kreise hängen vom Winkel \dot{y}_1 ab und verlaufen entlang eines Kreises mit dem Mittelpunkt im Ursprung des Koordinatensystems und dem Radius l_1 . Die Mittelpunkte der Kreise der zweiten Gleichung liegen alle im Ursprung des Koordinatensystems, während ihre Radien von der Länge beider Segmente und dem Winkel \dot{y}_2 zwischen ihnen abhängen.

Das Netz in Abb. 5.7 dient der einfachen grafischen Darstellung der Arbeitsfläche eines Zweisegmentroboters. Es ist nicht schwierig, die Arbeitsfläche für den Fall zu bestimmen, dass \dot{y}_1 und \dot{y}_2 im gesamten Bereich von 0° bis 360° variieren. Für den Zwei-Segment-Manipulator mit gleicher Länge beider Segmente ist dies einfach ein Kreis mit dem Radius $l_1 + l_2$. Wenn der Bewegungsbereich der Robotergelenke eingeschränkt wird, was normalerweise der Fall ist, erhält man deutlich unregelmäßigere Formen von Arbeitsbereichen. Ein Teil der Arbeitsfläche, in dem sich \dot{y}_1 von 0° auf 60° und \dot{y}_2 von 60° auf 120° ändert, ist in Abb. 5.7 schraffiert dargestellt.

Bei der Darstellung der Arbeitsflächen des Zwei-Segment-Manipulators gingen wir davon aus, dass die Längen beider Segmente gleich sind. Diese Annahme wird nun durch gestützt

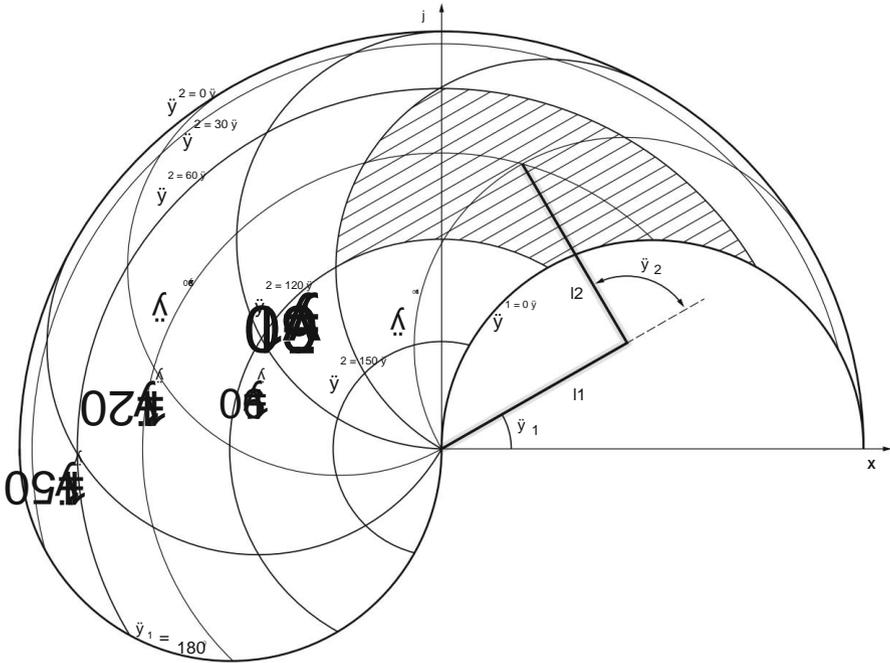


Abb. 5.7 Arbeitsbereich eines planaren zweisegmentigen Roboterarmes ($l_1 = l_2$, $0 \leq \gamma_1 \leq 180^\circ$, $0 \leq \gamma_2 \leq 180^\circ$)

ein ausreichender Beweis. Es ist nicht schwer zu erkennen, dass die Segmente industrieller SCARA und anthropomorpher Roboter gleich lang sind. Betrachten wir einen Roboter mit zwei Segmenten, bei dem das zweite Segment kürzer als das erste ist und die Winkel γ_1 und γ_2 zwischen 0° und 360° variieren (Abb. 5.8). Der Arbeitsbereich eines solchen Manipulators ist ein Ring mit Innenradius $R_i = l_1 \cdot l_2$ und Außenradius $R_o = l_1 + l_2$. Unser Ziel ist es, das Verhältnis der Segmentlängen l_1 und l_2 zu ermitteln, das bei konstanter Summe der Längen beider Segmente R_o den größten Arbeitsbereich ergibt. Der Arbeitsbereich des beschriebenen zweisegmentigen Roboterarmes beträgt

$$R_o^2 - R_i^2 = 2 A = \gamma \tag{5.23}$$

Durch Einfügen des Ausdrucks für den Innenradius in Gl. (5.23)

$$R_o^2 - (l_1 \cdot l_2)^2 = (2l_1 \cdot l_2)^2 \tag{5.24}$$

wir können schreiben

$$A = \gamma R_o^2 - \gamma (2l_1 \cdot l_2)^2 \tag{5.25}$$

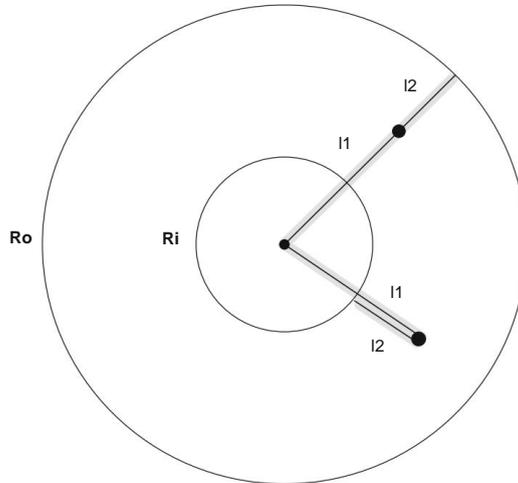


Abb. 5.8 Arbeitsbereich des Manipulators mit zwei Segmenten, wobei das zweite Segment kürzer ist

Für eine maximale Fläche sollte die Ableitung nach der Segmentlänge l_1 gleich sein null

$$-\dot{A} = 2\dot{y}(2l_1 - R_o) = 0 \cdot \dot{y}l_1 \quad (5.26)$$

Die Lösung ist

$$l_1 = \frac{R_o}{2}, \quad (5.27)$$

geben

$$l_1 = l_2. \quad (5.28)$$

Der größte Arbeitsbereich des Zwei-Segment-Mechanismus entsteht bei gleicher Länge beider Segmente.

Die Fläche der Arbeitsfläche hängt von den Segmentlängen l_1 und l_2 sowie von den minimalen und maximalen Werten der Winkel \dot{y}_1 und \dot{y}_2 ab. Durch Ändern der Verhältnisse l_1/l_2 können wir verschiedene Formen der Roboterarbeitsfläche erhalten. Die Fläche einer solchen Arbeitsfläche entspricht immer der in Abb. 5.9 dargestellten. In dieser Abbildung bezieht sich \ddot{y}_1 auf die Differenz zwischen dem maximalen und minimalen Gelenkwinkelwert $\ddot{y}_1 = (\dot{y}_1^{\max} - \dot{y}_1^{\min})$. Die Fläche der Arbeitsfläche ist die Fläche eines Ringsegments

$$A = \frac{\ddot{y}_1 \dot{y}}{360} (r_1^2 - r_2^2) \quad (5.29)$$

für \ddot{y}_1 in Winkelgraden angegeben.

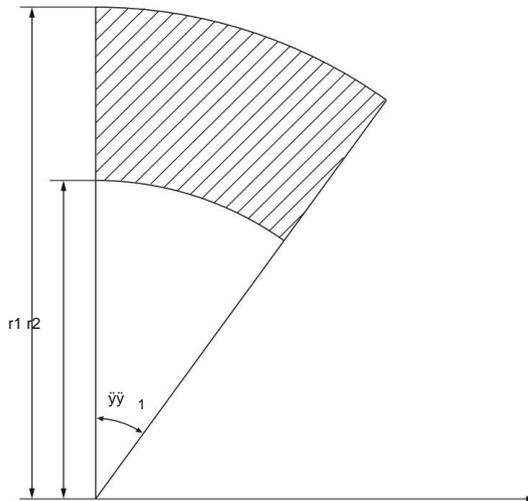


Abb. 5.9 Arbeitsfläche eines Zwei-Segment-Manipulators

In Gl. (5.29) ergeben sich die Radien r_1 und r_2 nach der Kosinusregel

$$r_1 = \sqrt{l_1^2 + l_2^2 + 2l_1l_2 \cos \gamma_{2\min}} \quad r_2 = \sqrt{l_1^2 + l_2^2 + 2l_1l_2 \cos \gamma_{2\max}} \quad (5.30)$$

Die Fläche der Arbeitsfläche ist ebenso wie ihre Form vom Verhältnis l_2/l_1 und von den Randbedingungen der Gelenkwinkel abhängig. Der Winkel γ_1 bestimmt die Lage der Arbeitsfläche zum Bezugssystem und hat keinen Einfluss auf deren Form. Untersuchen wir den Einfluss des zweiten Winkels γ_2 auf die Fläche der Arbeitsfläche. Wir gehen davon aus, dass $l_1 = l_2 = 1$ und γ_1 sich von 0° bis 180° ändert. Für gleiche Bereiche des Winkels γ_2 (30°) und für unterschiedliche Werte von $\gamma_{2\max}$ und $\gamma_{2\min}$ erhalten wir unterschiedliche Werte der Arbeitsbereiche

$$\begin{aligned} 0^\circ \leq \gamma_2 \leq 30^\circ & A = 0,07 \\ 30^\circ \leq \gamma_2 \leq 60^\circ & A = 0,19 \\ 60^\circ \leq \gamma_2 \leq 90^\circ & A = 0,26 \\ 90^\circ \leq \gamma_2 \leq 120^\circ & A = 0,26 \\ 120^\circ \leq \gamma_2 \leq 150^\circ & A = 0,19 \\ 150^\circ \leq \gamma_2 \leq 180^\circ & A = 0,07. \end{aligned}$$

Bisher haben wir unter dem Begriff „Workspace“ den sogenannten erreichbaren Roboter-Arbeitsraum verstanden. Dazu gehören alle Punkte in der Roboterumgebung, die vom Roboterendpunkt erreicht werden können. Oftmals ist dieser sogenannte geschickte Arbeitsbereich von größerer Bedeutung. Der geschickte Arbeitsbereich umfasst alle Punkte, die mit jeder beliebigen Ausrichtung des Roboter-Endeffektors erreicht werden können.

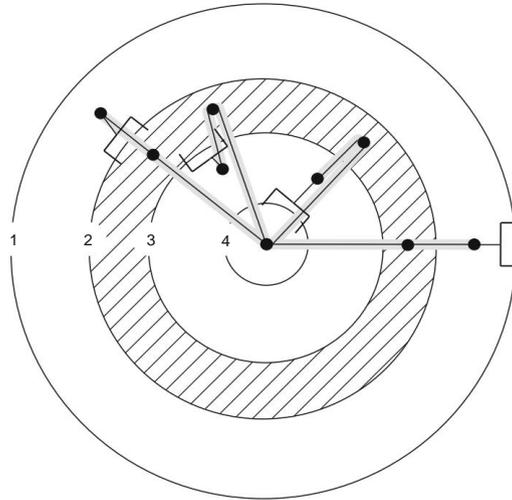


Abb. 5.10 Erreichbarer und geschickter Arbeitsbereich eines Zwei-Segment-Manipulators mit Endeffektor

kleiner als der erreichbare Arbeitsbereich. Der Arbeitsbereich für die Fingerfertigkeit ist größer, wenn das letzte Segment (Endeffektor) kürzer ist. Die erreichbaren und geschickten Arbeitsbereiche eines Zweisegmentroboters mit Endeffektor sind in Abb. 5.10 dargestellt. Der zweite und dritte Kreis werden erhalten, wenn der Endeffektor des Roboters auf den Bereich ausgerichtet ist durch die beiden Kreise eingeschränkt. Diese beiden Kreise stellen die Grenzen des geschickten Arbeitsbereichs dar. Der erste und der vierte Kreis schränken den erreichbaren Arbeitsbereich ein. Die Punkte zwischen dem ersten und dem zweiten sowie dem dritten und dem vierten Kreis können mit einer beliebigen Ausrichtung des Endeffektors nicht erreicht werden.

Für Roboter mit mehr als drei Gelenken ist der beschriebene grafische Ansatz nicht geeignet. Dabei nutzen wir numerische Methoden und Computeralgorithmen.

5.4 Dynamik

Zur Veranschaulichung betrachten wir den planaren Roboterarm mit zwei Segmenten, wie in Abb. 5.11 dargestellt. Die Segmente der Länge l_1 und l_2 können sich in der vertikalen x - y -Ebene bewegen, wobei ihre Positionen durch Winkel in Bezug auf die horizontale (x)-Achse beschrieben werden; \ddot{y}_1 und $\ddot{y} = \ddot{y}_1 + \ddot{y}_2$. Aktuatoren an den Gelenken liefern Drehmomente M_1 und M_2 , deren positive Richtung durch zunehmende Winkel definiert wird, also entlang der positiven Richtung der z -Achse unseres Referenzkoordinatensystems.

Wir approximieren nun die Segmente durch Punktmassen m_1 und m_2 in den Mittelpunkten starrer, aber ansonsten masseloser Stäbe (siehe Abb. 5.12). Sei r_1 die Position der Punktmasse m_1 in Bezug auf das erste Gelenk, das im Ursprung unseres Koordinatensystems liegt. Sei r_2 die Position der Punktmasse m_2 in Bezug auf das zweite Gelenk, das sich an der Verbindungsstelle der beiden Segmente befindet.

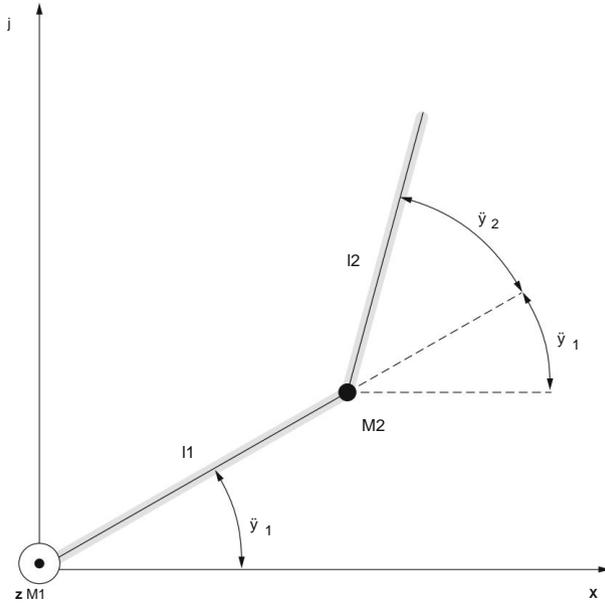


Abb. 5.11 Parameter des planaren, zweisegmentigen Roboterarmes, der sich in der vertikalen x - y -Ebene bewegt

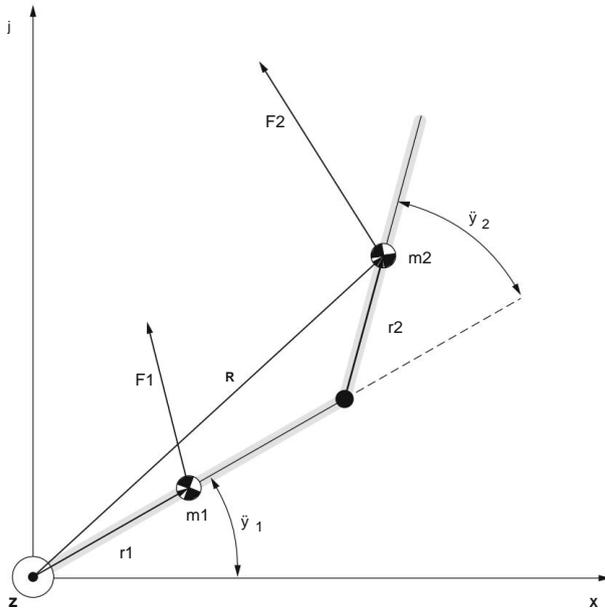


Abb. 5.12 Planarer Roboterarm, angenähert durch Punktmassen $m1$ und $m2$

Auf die Punktmassen m_1 und m_2 wirken die Kräfte, die von den masselosen Stäben übertragen werden, sowie die Schwerkraft. Das Newtonsche Gesetz besagt, dass die Vektorsumme aller auf ein Teilchen einwirkenden Kräfte gleich dem Produkt aus Teilchenmasse und Beschleunigung ist. Daher,

$$F_1 = m_1 a_1 \text{ und } F_2 = m_2 a_2, \quad (5.31)$$

wobei F_1 und F_2 die Summen aller Kräfte (d. h. der Stabkraft und der Schwerkraft) darstellen, die auf jede der Punktmassen m_1 und m_2 wirken, während a_1 und a_2 ihre Beschleunigungen in Bezug auf den Ursprung der Punktmassen sind Koordinatenrahmen.

Eine Berechnung der Beschleunigungen läuft also auf die Bestimmung der Kräfte hinaus, die auf die beiden

Die Position von m_1 in Bezug auf den Referenzrahmenursprung ist durch r_1 gegeben, während die Position von m_2 durch $r = 2r_1 + r_2$ gegeben ist (siehe Abb. 5.12). Die entsprechenden Beschleunigungen sind daher $a_1 = \ddot{r}_1$ und $a_2 = \ddot{r}$, wobei die beiden Punkte über dem Vektorsymbol zweite Ableitungen nach der Zeit bezeichnen. Daher,

$$a_1 = \ddot{r}_1 \text{ und } a_2 = \ddot{r} = 2\ddot{r}_1 + \ddot{r}_2. \quad (5.32)$$

Da r_1 und r_2 nun starre Stäbe darstellen, sind ihre Längen fest. Daher können diese Vektoren nur rotieren. Erinnern wir uns an die grundlegende Physik, die besagt, dass ein rotierender Vektor ein Teilchen in Kreisbewegung beschreibt. Eine solche Bewegung kann zwei Beschleunigungskomponenten haben (Abb. 5.13, siehe auch Anhang ??). Die erste Komponente ist die Radial- oder Zentripetalbeschleunigung a_r , die auf das Rotationszentrum gerichtet ist. Sie beruht nur auf der Änderung der Geschwindigkeitsrichtung und ist daher auch bei gleichförmiger Kreisbewegung vorhanden. Es ist durch den Ausdruck gegeben

$$a_r = -\dot{\theta}^2 r, \quad (5.33)$$

wobei $\dot{\theta}$ die Winkelgeschwindigkeit $\dot{\theta} = \dot{\theta}$ ist. Die zweite Komponente ist die Tangentialbeschleunigung, die entlang der Tangente an den Kreis gerichtet ist (Abb. 5.13). Sie entsteht durch die Änderung des Geschwindigkeitsbetrags und tritt nur bei Kreisbewegungen mit der Winkelbeschleunigung $\ddot{\theta} = \ddot{\theta}$ auf. Es ist gegeben durch

$$a_t = \ddot{\theta} \times r, \quad (5.34)$$

Dabei ist $\ddot{\theta}$ der Vektor der Winkelbeschleunigung, der senkrecht zur Bewegungsebene verläuft, also entlang der z-Achse unseres Referenzkoordinatensystems. Die Gesamtbeschleunigung ist offensichtlich

$$a = a_r + a_t = -\dot{\theta}^2 r + \ddot{\theta} \times r. \quad (5.35)$$

Berechnen wir nun die zweiten Ableitungen der Vektoren r_1 und r_2 nach der Zeit. Wie oben erwähnt, hat jede dieser Ableitungen zwei Komponenten, die der Radial- und der Tangentialbeschleunigung entsprechen. Also

$$\ddot{r}_1 = -\dot{\theta}_1^2 r_1 + \ddot{\theta}_1 \times r_1 \text{ und } \ddot{r}_2 = -\dot{\theta}_2^2 r_2 + \ddot{\theta}_2 \times r_2. \quad (5.36)$$

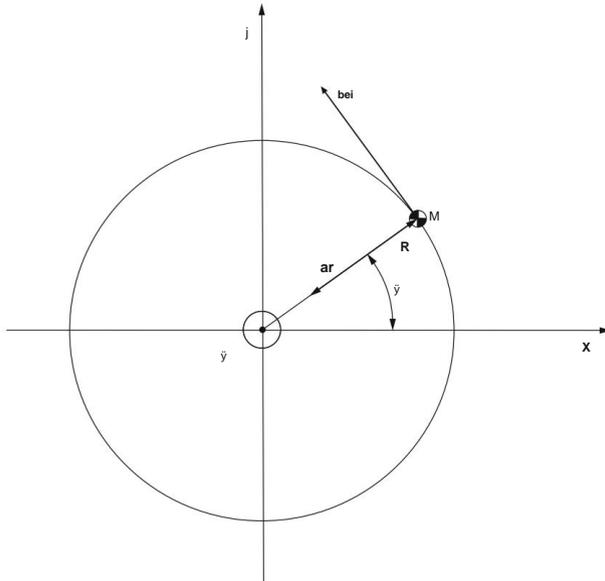


Abb. 5.13 Ein rotierender Vektor r fester Länge beschreibt ein Teilchen in Kreisbewegung

Der Betrag der Winkelgeschwindigkeit $\dot{\theta}_1$ und der Vektor der Winkelbeschleunigung $\ddot{\theta}_1$ des ersten Segments sind

$$\dot{\theta}_1 = \dot{\theta}_1 \quad \text{und} \quad \ddot{\theta}_1 = \ddot{\theta}_1 \mathbf{k}, \quad (5.37)$$

wobei \mathbf{k} der Einheitsvektor entlang der z-Achse ist. Die Winkelgeschwindigkeit $\dot{\theta}_2$ und die Winkelbeschleunigung $\ddot{\theta}_2$ des zweiten Segments betragen

$$\dot{\theta}_2 = \dot{\theta}_2 = \dot{\theta}_1 + \dot{\theta}_2 \quad \text{und} \quad \ddot{\theta}_2 = \ddot{\theta}_2 \mathbf{k} = (\ddot{\theta}_1 + \ddot{\theta}_2) \mathbf{k}. \quad (5.38)$$

Hier haben wir $\dot{\theta} = \dot{\theta}_1 + \dot{\theta}_2$ verwendet (siehe Abb. 5.11 und 5.12). Die zweiten Ableitungen der Vektoren r_1 und r_2 können geschrieben werden als

$$\ddot{r}_1 = \ddot{\theta}_1 r_1 + \dot{\theta}_1 \times r_1 = \ddot{\theta}_1 r_1 + \dot{\theta}_1 \times r_1 \quad (5.39)$$

Und

$$\begin{aligned} \ddot{r}_2 &= \ddot{\theta}_2 r_2 + \dot{\theta}_2 \times r_2 = \ddot{\theta}_2 r_2 + (\dot{\theta}_1 + \dot{\theta}_2) \times r_2 = \ddot{\theta}_2 r_2 + \dot{\theta}_1 \times r_2 + \dot{\theta}_2 \times r_2 \\ &= \ddot{\theta}_2 r_2 + (\ddot{\theta}_1 + \ddot{\theta}_2) \mathbf{k} \times r_2. \end{aligned} \quad (5.40)$$

Mit diesen Ausdrücken können wir nun die Beschleunigungen der beiden Punktmassen m_1 und m_2 berechnen, die unserem Zweisegmentroboter entsprechen. Die Beschleunigung a_1 von m_1 beträgt

$$\mathbf{a}_1 = \ddot{r}_1 = \ddot{\theta}_1 r_1 + \dot{\theta}_1 \times r_1. \quad (5.41)$$

Die Beschleunigung a_2 von m_2 beträgt

$$\begin{aligned} a_2 = \ddot{r} &= 2\ddot{r}_1 + \ddot{r}_2 = \\ &= \ddot{y}_1 \ddot{y}_2 \mathbf{1} + 2\ddot{y}_1 \dot{\gamma} (\dot{\gamma} \times \mathbf{r}_1) \dot{\gamma} (\ddot{y}_1 + \ddot{y}_2) \mathbf{2} r_2 + (\ddot{y}_1 + \ddot{y}_2) (\dot{\gamma} \times \mathbf{r}_2). \end{aligned} \quad (5.42)$$

Aus diesen Beschleunigungen erhalten wir die Gesamtkräfte, die auf die Teilchen m_1 und m_2 wirken

$$F_1 = m_1 a_1 \text{ und } F_2 = m_2 a_2. \quad (5.43)$$

Wir können nun die Drehmomente dieser Kräfte in Bezug auf den Ursprung des Koordinatensystems berechnen

$$\tau = \mathbf{r}_1 \times F_1 = \mathbf{r}_1 \times m_1 a_1 \text{ und } \tau_2 = \mathbf{r}_2 \times F_2 = (2\mathbf{r}_1 + \mathbf{r}_2) \times m_2 a_2. \quad (5.44)$$

Wir fügen Ausdrücke für a_1 und a_2 ein, wie oben abgeleitet, erinnern uns an das Doppelvektorprodukt $[\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})]$ und führen geduldig die langwierige Algebra durch erhalten

$$\begin{aligned} \tau_1 &= m_1 r_1^2 \ddot{y}_1 \mathbf{k} \\ \text{Und} \\ \tau_2 &= \ddot{y}_1 \mathbf{1} (4m_2 r_1^2 + 4m_2 r_1 r_2 \cos \gamma_2 + m_2 r_2^2) + \\ &\quad + \ddot{y}_2 (m_2 r_1^2 + 2m_2 r_1 r_2 \cos \gamma_2) \dot{\gamma} \\ &\quad + \ddot{y}_1 \dot{\gamma} \mathbf{2} 4m_2 r_1 r_2 \sin \gamma_2 \dot{\gamma} \mathbf{2} 2m_2 r_1 r_2 \sin \gamma_2 \mathbf{k}. \end{aligned} \quad (5.45)$$

Die Summe beider Drehmomente auf die beiden „Teilchen“ unseres Systems beträgt offensichtlich $\tau = \tau_1 + \tau_2$.

Andererseits können wir unser Zweisegmentensystem, bestehend aus zwei Punktmassen und zwei masselosen Stäben, aus einem anderen Blickwinkel betrachten. Als Folge des dritten Newtonschen Gesetzes (Auf jede Aktion gibt es eine gleiche, aber entgegengesetzte Reaktion) haben wir einen Satz, der besagt, dass sich interne Drehmomente in einem System aufheben, sodass nur Drehmomente externer Kräfte relevant sind. Die Drehmomente externer Kräfte auf unser Robotersystem sind die Drehmomente der Schwerkraft und das Drehmoment, das von der Basis ausgeübt wird, auf die Basis. Das Drehmoment der Basis ist gleich dem Drehmoment M_1 des Aktuators im ersten Gelenk. Die Summe dieser Drehmomente externer Kräfte (Basis + Schwerkraft) muss gleich $\tau_1 + \tau_2$ sein (oben abgeleitet), da beide Ergebnisse zwei Sichtweisen auf das Gesamtdrehmoment im selben System darstellen. Also

$$M_1 + \mathbf{r}_1 \times m_1 \mathbf{g} + \mathbf{r}_2 \times m_2 \mathbf{g} = \tau_1 + \tau_2. \quad (5.46)$$

Mit $\mathbf{r} = 2\mathbf{r}_1 + \mathbf{r}_2$ ergibt sich das Drehmoment des Aktuators im ersten Gelenk

$$M_1 = \ddot{y}_1 + \ddot{y}_2 \dot{\gamma} \mathbf{r}_1 \times m_1 \dot{\gamma} (2\mathbf{r}_1 + \mathbf{r}_2) \times m_2 \mathbf{g}. \quad (5.47)$$

Wenn man bedenkt, dass g vertikal nach unten zeigt (also entlang \ddot{y}), und die obigen Ergebnisse für \ddot{y}_1 und \ddot{y}_2 einsetzt, erhält man schließlich

$$\begin{aligned} M_1 = & \ddot{y}_1 (m_1 r_1^2 + m_2 r_2^2 + 4m_2 r_1^2 + 4m_2 r_1 r_2 \cos \dot{y}_2) + \\ & + \ddot{y}_2 (2m_2 r_1^2 + 2m_2 r_1 r_2 \cos \dot{y}_2) \ddot{y} \\ & \ddot{y} \ddot{y}_1 \ddot{y}_2 24m_2 r_1 r_2 \sin \dot{y}_2 \ddot{y} \ddot{y}_2 2m_2 r_1 r_2 \sin \dot{y}_2 + \\ & + m_1 g r_1 \cos \dot{y}_1 + 2m_2 g r_1 \cos \dot{y}_1 + m_2 g r_2 \cos(\dot{y}_1 + \dot{y}_2). \end{aligned} \quad (5.48)$$

Um das Drehmoment M_2 des Aktors im zweiten Gelenk zu erhalten, betrachten wir zunächst die Gesamtkraft F_2 , die auf die Punktmasse m_2 wirkt. Die Kraft F_2 ist eine Summe zweier Beiträge. Das eine ist die Schwerkraft $m_2 g$, das andere ist die Kraft F , die der masselose und starre Stab des zweiten Segments auf m_2 ausübt. Also

$$F_2 = F_2 + m_2 g. \quad (5.49)$$

Auf diese Gleichung wenden wir ein Vektorprodukt von r_2 von links an und erhalten

$$r_2 \times F_2 = r_2 \times F_2 + r_2 \times m_2 g. \quad (5.50)$$

Der erste Term auf der rechten Seite ist das Vektorprodukt von r_2 mit der Kraft F_2 auf m_2 durch den masselosen und starren Stab ausgeübt. Dieser Term entspricht dem Drehmoment M_2 des Aktors im zweiten Gelenk. (Beachten Sie, dass der Stab auch eine Kraft auf m_2 ausüben kann, die entlang des Stabes gerichtet ist, aber das Vektorprodukt dieser Komponente mit r_2 verschwindet.) Wir erhalten also

$$M_2 = r_2 \times F_2 - r_2 \times m_2 g. \quad (5.51)$$

Das Ersetzen von F_2 durch $m_2 a_2$ und den zuvor abgeleiteten Ausdruck für a_2 führt zu

$$\begin{aligned} M_2 = & \ddot{y}_1 (m_2 r_1^2 + 2m_2 r_1 r_2 \cos \dot{y}_2) + \ddot{y}_2 (2m_2 r_1^2 + 2m_2 r_1 r_2 \cos \dot{y}_2) \\ & + \ddot{y}_1 \ddot{y}_2 2m_2 r_1 r_2 \sin \dot{y}_2 + m_2 r_2 g \cos(\dot{y}_1 + \dot{y}_2). \end{aligned} \quad (5.52)$$

Die Ausdrücke für M_1 (5.48) und M_2 (5.52) scheinen relativ kompliziert zu sein. Lassen Sie uns daher einige einfache und bekannte Fälle untersuchen. Nehmen wir zunächst $\dot{y}_1 = 90^\circ$ und kein Drehmoment im zweiten Gelenk $M_2 = 0$ an (Abb. 5.14 links). Die Gleichung für M_2 reduziert sich

$$\ddot{y}_2 2m_2 r_1^2 = \ddot{y}_2 m_2 g r_2 \sin \dot{y}_2. \quad (5.53)$$

Dies ist die Gleichung eines einfachen Pendels mit Masse m_2 , Trägheitsmoment $J_2 = m_2 r_2^2$ das sich mit der Winkelbeschleunigung \ddot{y}_2 um das zweite Gelenk dreht (Abb. 5.14 links). Die linke Seite ist also $J_2 \ddot{y}_2$ und auf der rechten Seite haben wir das Drehmoment aufgrund der Schwerkraft. Dies ist also ein Beispiel für die einfache Gleichung $M = J \ddot{y}$, zu der unsere

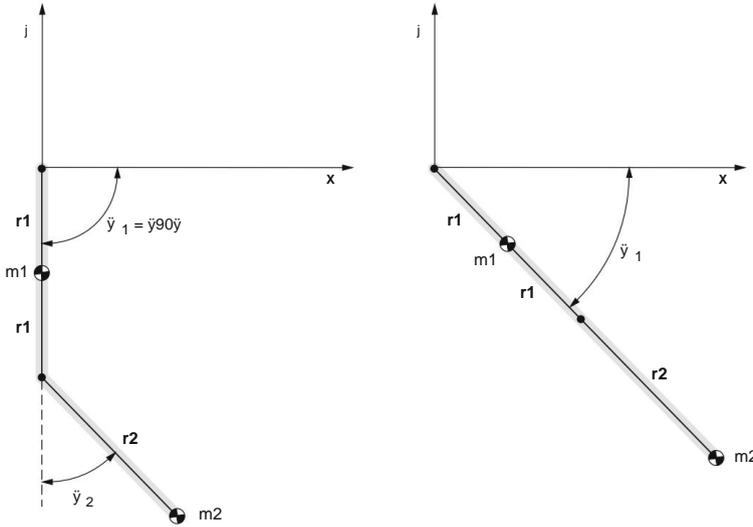


Abb. 5.14 Zwei einfache Beispiele des Zwei-Segment-Roboterarmes: $\ddot{y}_1 = \ddot{y}_1$ (links) und $\ddot{y}_2 = 0$ (rechts)

komplizierter Ausdruck wurde reduziert. Für kleine Schwingungen ($\ddot{y}_2 \ll \ddot{y}_1$) gilt $\sin \ddot{y}_2 \approx \ddot{y}_2$ und die Gleichung lautet

$$g \ddot{y}_1 + \frac{m_2}{m_1} \ddot{y}_2 = 0. \tag{5.54}$$

Dies ist die Gleichung des einfachen Pendels mit der Kreisfrequenz $\ddot{y}_0 =$

$$\frac{g}{r_2} \text{ Und}$$

Schwingungsperiode $T = 2\pi \sqrt{\frac{r_2}{g}}$.

Nehmen wir als nächstes $\ddot{y}_2 = 0$ an, sodass wir einen starren Stab haben, der um ein Ende rotiert und im Ursprung des Koordinatensystems fixiert ist (Abb. 5.14 rechts). Wenn wir zusätzlich die Schwerkraft „ausschalten“ ($g = 0$), erhalten wir für das Drehmoment im ersten Gelenk

$$\begin{aligned} M_1 &= \ddot{y}_1 (m_1 r_1^2 + m_2 r_2^2 + 4m_2 r_1^2 + 4m_2 r_1 r_2) = \\ &= \ddot{y}_1 [m_1 r_1^2 + m_2 (2r_1 + r_2)^2] = J_{12} \ddot{y}_1, \end{aligned} \tag{5.55}$$

wobei $\ddot{y}_1 = \ddot{y}_1$ ist die Winkelbeschleunigung und J_{12} ist das kombinierte Trägheitsmoment der beiden Massen. Alternativ könnte man das Drehmoment im ersten Gelenk gleich Null annehmen, die Schwerkraft einbeziehen und so ein relativ einfaches Pendel mit zwei Punktmassen auf einem masselosen starren Stab erhalten.

Erwähnen wir, dass die obigen vollständigen Gleichungen für M_1 und M_2 (Gl. (5.48) und (5.52)) mit geringfügigen Anpassungen der Notation für ein Doppelpendel mit gültig sind

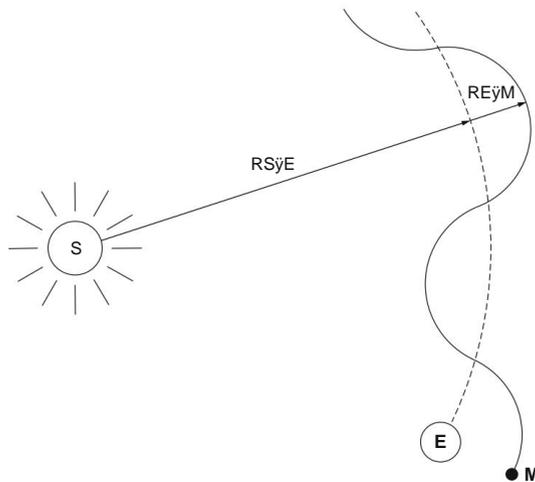


Abb. 5.15 Schematische Flugbahn (nicht maßstabsgetreu) der Erde (gestrichelte Kurve) und des Mondes (vollständige Kurve) im Bezugssystem der Sonne

Reibung. Die Drehmomente der Aktoren werden dabei durch die Reibungsmomente in den Gelenken ersetzt.

Eine amüsante Übung wäre es, die Flugbahnen der Endpunkte der beiden Segmente unseres einfachen Roboters mit den Flugbahnen der Erde und des Mondes zu vergleichen, wie sie vom Bezugssystem der Sonne aus gesehen werden. Nähern wir uns Erde und Mond als Punktteilchen (m_E \dot{y} m_M) auf koplanaren Kreisbahnen. Da die Gravitationskraft nur entlang der Verbindungslinie zwischen den beiden Teilchen wirkt, kann sie keine Drehmomente übertragen, sodass die Winkelbeschleunigungen Null und die Winkelgeschwindigkeiten konstant sind. Die Umlaufgeschwindigkeit der Erde um die Sonne ($RSyE \dot{y} 150 \cdot 10^6$ km, $T = 365$ Tage, $v_E \dot{y} 2,6 \cdot 10^6$ km/Tag) ist viel größer als die Umlaufgeschwindigkeit des Mondes um die Erde ($REyM \dot{y} 0,38 \cdot 10^6$ km, $T = 28$ Tage, $v_M \dot{y} 0,08 \cdot 10^6$ km/Tag), sodass die Flugbahn des Mondes im Bezugssystem der Sonne ungefähr eine Sinuskurve wäre, die der kreisförmigen Umlaufbahn der Erde um die Sonne überlagert ist (Abb. 5.15).

Bei unserem Zwei-Segment-Roboter könnte man höhere Winkelgeschwindigkeiten des zweiten Segments haben, was zu unterschiedlichen Formen der Flugbahn seines Endpunkts führen würde (z. B. die Epizykel des Ptolemäus).

Wenn wir zu unseren relativ komplizierten Gleichungen für die Drehmomente M_1 und M_2 (Gl. (5.48) und (5.52)) zurückkehren, die auf Aktoren in den Gelenken zurückzuführen sind, sehen wir, dass diese Gleichungen in einer Matrixform zusammengefasst werden können, die das dynamische Robotermodell wie folgt darstellt

$$\ddot{y} = B(q)\dot{q} + C(q, \dot{q})\dot{q} + g(q). \quad (5.56)$$

In dieser Gleichung vereint der Vektor \ddot{y} die Drehmomente beider Aktoren

$$\ddot{y} = \begin{pmatrix} M1 \\ M2 \end{pmatrix} \quad (5.57)$$

Die Vektoren q , \dot{q} und \ddot{q} gehören zu den Segmenttrajektorien, -geschwindigkeiten bzw. -beschleunigungen. Für den Zwei-Segment-Roboter haben wir

$$q = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix}, \quad \dot{q} = \begin{pmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{pmatrix}, \quad \ddot{q} = \begin{pmatrix} \ddot{\ddot{y}}_1 \\ \ddot{\ddot{y}}_2 \end{pmatrix}$$

Der erste Term der Gleichung für \ddot{y} wird als Inertialterm bezeichnet. In unserem Fall des planaren Roboterarmes mit zwei Segmenten mit $r_1 = r_2 = l$ und durch Vereinfachung der Notation mit $s_1 = \sin(\dot{y}_1)$, $c_{12} = \cos(\dot{y}_1 + \dot{y}_2)$ usw. erhalten wir

$$B(q) = \begin{pmatrix} \frac{1}{4}m_1l^2 + \frac{5}{4}m_2l^2 + m_2l^2c_2 & \frac{1}{4}m_2l^2 + \frac{1}{2}m_2l^2c_2 \\ \frac{1}{4}m_2l^2 + \frac{1}{2}m_2l^2c_2 & \frac{1}{4}m_2l^2 \end{pmatrix} \quad (5.58)$$

Der zweite Term dieser Matrixgleichung wird Coriolis-Term genannt und umfasst Geschwindigkeits- und Zentrifugaleffekte. Für den Zwei-Segment-Roboter haben wir die folgende Matrix

$$C(q, \dot{q}) = \begin{pmatrix} \dot{y}_2 m_2 l^2 s_2 \dot{y}_1 \dot{y}_2 & \frac{1}{2} m_2 l^2 s_2 \dot{y}_2^2 \\ \frac{1}{2} m_2 l^2 s_2 \dot{y}_1 \dot{y}_2 & 0 \end{pmatrix} \quad (5.59)$$

Die Gravitationsäule $g(q)$ hat in unserem Fall die folgende Form

$$g(q) = \begin{pmatrix} \frac{1}{2} m_1 g l c_1 + m_2 g l c_1 + \frac{1}{2} m_2 g l c_{12} \\ \frac{1}{2} m_2 g l c_{12} \end{pmatrix} \quad (5.60)$$

Kapitel 6

Parallele Roboter



In diesem Kapitel geht es um die immer beliebter werdenden und leistungsfähigeren Roboter, die sogenannten Parallelroboter. Standardmechanismen von Industrierobotern verfügen über serielle kinematische Ketten, in denen sich Glieder und Gelenke abwechseln, wie in Abb. 6.1 (links) dargestellt. Diese werden als serielle Roboter bezeichnet. In letzter Zeit haben wir eine erhebliche Weiterentwicklung paralleler Roboter erlebt. Sie umfassen geschlossene kinematische Ketten, ein Beispiel ist in Abb. 6.1 (rechts) dargestellt.

In der Industrie haben Parallelroboter in den letzten zwei Jahrzehnten zunehmend an Bedeutung gewonnen. Die ersten Entwicklungen gehen jedoch auf das Jahr 1962 zurück, als Gough und Whitehall einen Parallelroboter zum Testen von Autoreifen entwickelten. Etwa zur gleichen Zeit stellte Stewart einen ähnlichen Parallelroboter vor, um einen Flugsimulator zu entwerfen. Der Parallelroboter, bei dem eine mobile Plattform von sechs angetriebenen Beinen gesteuert wird, wird daher Stewart-Gough-Plattform genannt. Auch der Durchbruch der Parallelroboter war maßgeblich dem in den Achtzigerjahren von Clavel entwickelten Roboter zu verdanken. Sein Mechanismus wurde 1990 in den USA unter dem Namen Delta-Roboter patentiert. Die parallelen Mechanismen in der Robotik waren Anfang der achtziger Jahre Gegenstand systematischer wissenschaftlicher Forschung geworden. Diese Aktivitäten intensivierten sich in den neunziger Jahren erheblich und gipfelten in einigen wichtigen Errungenschaften in der Roboterkinematik im Allgemeinen.

6.1 Eigenschaften paralleler Roboter

Bei seriellen Robotern ist die Anzahl der Freiheitsgrade identisch mit der Gesamtzahl der Freiheitsgrade in Gelenken. Daher müssen alle Gelenke betätigt werden, und normalerweise werden nur einfache Translations- und Rotationsgelenke mit einem Freiheitsgrad verwendet. Bei Parallelrobotern ist die Anzahl der Freiheitsgrade geringer als die Gesamtzahl der Freiheitsgrade in den Gelenken, sodass viele Gelenke passiv sind. Passive Gelenke können komplexer sein; Typische Vertreter sind das Kardangelenk und das Kugelgelenk. Das Universalgelenk besteht aus zwei senkrechten Drehungen und drei senkrechten

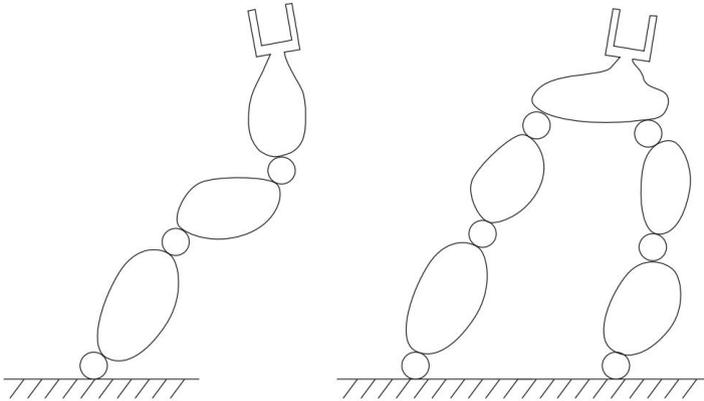


Abb. 6.1 Serielle kinematische Kette (links) und geschlossene kinematische Kette (rechts)

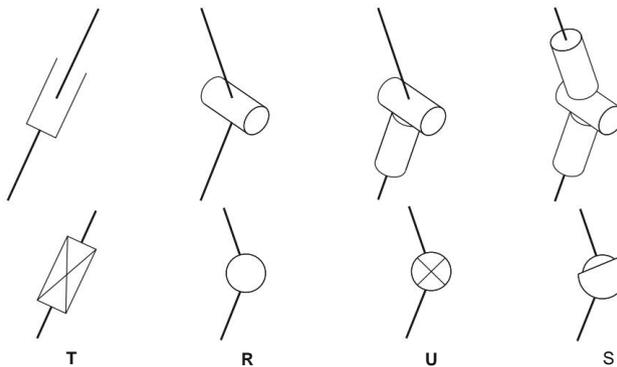


Abb. 6.2 Arten von Gelenken, die häufig in Parallelmechanismen verwendet werden

Rotationen bilden das Kugelgelenk, wie in Abb. 6.2 dargestellt. Hier werden die Buchstaben T, R, U und S verwendet, um das Translationsgelenk, das Rotationsgelenk, das Universalgelenk bzw. das Kugelgelenk zu kennzeichnen.

Bei Parallelrobotern ist das letzte (oberste) Glied der Mechanismen die sogenannte Plattform (Abb. 6.3). Die Plattform ist das aktive Glied, an dem der Endeffektor befestigt ist. Es ist über eine bestimmte Anzahl (normalerweise) serieller Mechanismen, sogenannte Beine, mit der festen Basis verbunden. Die gesamte Struktur enthält mindestens eine geschlossene kinematische Kette (mindestens zwei Beine). Die Verschiebungen in den Beinen bewirken eine Verschiebung der Plattform, wie in Abb. 6.3 dargestellt. Die Bewegungen der Plattform und der Beine sind durch oft sehr komplexe trigonometrische Ausdrücke (direkte und inverse Kinematik) verbunden, abhängig von der Geometrie des Mechanismus, der Art der Gelenke, der Anzahl der Beine und ihrer kinematischen Anordnung.

Leider gibt es keine eindeutigen und einheitlichen Bezeichnungen für Parallelroboter.

In dieser Arbeit wird ein Parallelroboter durch die Art der kinematischen Ketten bezeichnet, die er darstellt.

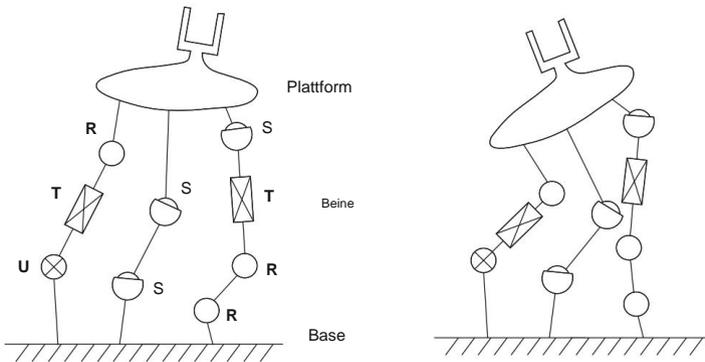


Abb. 6.3 Grundaufbau eines Parallelroboters

die Beine schicken. Daher wird der Roboter in Abb. 6.3 als UTR-SS-RRTS bezeichnet. Wenn Beine desselben Typs wiederholt werden, beispielsweise beim TRR-TRR-TRR-Roboter, kann die Bezeichnung vereinfacht als 3TRR verwendet werden.

Anzahl der Freiheitsgrade

Jedes Gelenk trägt zur Mobilität des Roboters bei, indem es eine bestimmte Anzahl von Freiheitsgraden oder alternativ eine entsprechende Anzahl von Einschränkungen einführt, die wie folgt definiert sind. Sei \tilde{y} die maximale Anzahl der Freiheitsgrade eines frei beweglichen Körpers (im Raum $\tilde{y} = 6$ und in der Ebene $\tilde{y} = 3$) und sei f_i die Anzahl der Freiheitsgrade des i -ten Gelenks. Die entsprechende Anzahl von Einschränkungen beträgt

$$c_i = \tilde{y} \tilde{y} f_i . \quad (6.1)$$

In der Roboterpraxis, in der serielle Roboter vorherrschen, betrachten wir Gelenke normalerweise als Elemente, die der Bewegung des Roboter-Endeffektors Freiheitsgrade verleihen. Bei Parallelrobotern hingegen ist es vorteilhafter, die Bewegung der Plattform (an der der Endeffektor befestigt ist) zu berücksichtigen und dabei die Anzahl der durch die Gelenke verursachten Einschränkungen zu berücksichtigen. Somit führt ein Universalgelenk U in einem Raum mit $\tilde{y} = 6$ $f_i = 2$ Freiheitsgrade und $c_i = \tilde{y} \tilde{y} f_i = 6 \cdot 6 \cdot 2 = 4$ Einschränkungen ein. Oder zum Beispiel führt ein Drehgelenk R in einer Ebene mit $\tilde{y} = 3$ $f_i = 1$ Freiheitsgrade und $c_i = \tilde{y} \tilde{y} f_i = 3 \cdot 3 \cdot 1 = 2$ Einschränkungen ein, während dasselbe Gelenk im Raum $c_i = \tilde{y} \tilde{y} f_i$ einführt = $6 \cdot 6 \cdot 1 = 5$ Einschränkungen. Beachten Sie, dass Rotations- und Translationsgelenke sowohl in einer Ebene als auch im Raum funktionieren können, wohingegen Kugel- und Universalgelenke nur räumliche Bewegungen erzeugen und nicht in planaren Robotern verwendet werden.

Die Anzahl der Freiheitsgrade eines Parallelroboters ist geringer als die Gesamtzahl der Freiheitsgrade, die die Robotergelenke beisteuern, anders als bei einem seriellen Roboter, bei dem diese beiden Zahlen identisch sind. Sei N die Anzahl der beweglichen Glieder des Roboters und n die Anzahl der Gelenke. Die Gelenke werden mit $i = 1, 2, \dots, n$ bezeichnet. Jedes Gelenk besitzt f_i - Freiheitsgrade und c_i - Einschränkungen. Die N frei beweglichen Verbindungen besitzen $N\tilde{y}$ Freiheitsgrade. Wenn sie zu einem Mechanismus zusammengefasst werden, ist

wird durch die durch Gelenke eingeführten Einschränkungen begrenzt, sodass die Anzahl der Freiheitsgrade eines Robotermechanismus beträgt

$$F = N\ddot{y} - \sum_{i=1}^N c_i . \quad (6.2)$$

Indem wir c_i durch $\ddot{y} - f_i$ ersetzen, erhalten wir die bekannte Grübler-Formel wie folgt

$$F = \ddot{y}(N - \ddot{y} n) + \sum_{i=1}^N f_i . \quad (6.3)$$

Wir dürfen nicht vergessen, dass die Anzahl der Motoren, die die Bewegung eines Roboters steuern, gleich F ist.

Beachten Sie, dass bei seriellen Robotern die Anzahl der beweglichen Glieder und die Anzahl der Gelenke identisch sind ($N = n$), sodass der erste Teil von Grüblers Formel immer Null ist ($\ddot{y}(N - \ddot{y} n) = 0$). Dies erklärt, warum die Anzahl der Freiheitsgrade bei seriellen Robotern einfach ist

$$F = \sum_{i=1}^N f_i . \quad (6.4)$$

Eine sehr praktische Form der Grübler-Formel zur Berechnung der Freiheitsgrade eines Parallelroboters erhält man wie folgt. Nehmen wir an, dass ein paralleler Mechanismus $k = 1, 2, \dots, K$ Zweige umfasst und dass jeder der Zweige \ddot{y}_k Freiheitsgrade und folglich $\ddot{y}_k = \ddot{y} - \ddot{y}_k$ Einschränkungen besitzt. Wenn die Plattform nicht mit den Beinen verbunden ist und sich frei im Raum bewegen kann, enthält sie \ddot{y} Freiheitsgrade. Die Anzahl der Freiheitsgrade einer verbundenen Plattform kann somit berechnet werden, indem die Summe der durch die Beine eingeführten Einschränkungen subtrahiert wird

$$F = \ddot{y} - \sum_{k=1}^K \ddot{y}_k . \quad (6.5)$$

Die Gleichungen (6.3) und (6.5) sind mathematisch identisch und können durch einfache algebraische Operationen ineinander umgewandelt werden.

Jetzt können wir die Freiheitsgrade für den in Abb. 6.3 gezeigten Roboter berechnen. Dieser Roboter verfügt über $N = 7$ bewegliche Glieder und $n = 9$ Gelenke. Die Gesamtzahl der Freiheitsgrade in Gelenken beträgt 16 (3 Rotationsgelenke, 2 Translationsgelenke, 1 Universal- und 3 Kugelgelenke). Unter Verwendung der in Gl. angegebenen Standard-Grübler-Formel. (6.3) erhalten wir

$$F = 6(7 - \ddot{y} 9) + 16 = 4.$$

Wenn wir nun die modifizierte Form der Grübler-Formel verwenden, müssen wir die durch jedes Bein eingeführten Einschränkungen berechnen. Dies ist ziemlich einfach, da wir nur die Anzahl der Freiheitsgrade jedes Beins von \ddot{y} subtrahieren müssen. Für den gegebenen Roboter (Beine

werden von links nach rechts gezählt) wir haben $\ddot{y}_1 = 2$, $\ddot{y}_2 = 0$ und $\ddot{y}_3 = 0$. Durch Einführung dieser Werte in Gl. (6.5) erhalten wir erwartungsgemäß

$$F = 6 \ddot{y}_2 = 4.$$

Vor- und Nachteile von Parallelrobotern

Beweggrund für die Einführung paralleler Roboter in der Industrie sind die zahlreichen wesentlichen Vorteile, die parallele Roboter im Vergleich zu seriellen Robotern haben. Die offensichtlichsten sind die folgenden:

Belastbarkeit, Steifigkeit und Genauigkeit. Die Tragfähigkeit von Parallelrobotern ist deutlich größer als die von Serienrobotern. Parallelroboter sind außerdem steifer und ihre Genauigkeit bei der Positionierung und Ausrichtung eines Endeffektors ist um ein Vielfaches besser als bei seriellen Robotern.

Hervorragende dynamische Eigenschaften. Die Plattform kann hohe Geschwindigkeiten und Beschleunigungen erreichen. Darüber hinaus ist die Resonanzfrequenz eines Parallelroboters um Größenordnungen höher.

Einfache Konstruktion. Mehrere passive Gelenke in Parallelrobotern ermöglichen eine kostengünstigere und einfachere mechanische Konstruktion. Beim Bau von Parallelrobotern können Standardlager, Spindeln und andere Maschinenelemente verwendet werden.

Der Einsatz paralleler Roboter ist jedoch begrenzt. Aufgrund der verwickelten Beine können Parallelroboter Schwierigkeiten haben, Hindernissen in ihrem Arbeitsbereich auszuweichen. Weitere wesentliche Nachteile sind:

Kleiner Arbeitsbereich. Parallelroboter haben deutlich kleinere Arbeitsräume als Serienroboter vergleichbarer Größe. Ihr Arbeitsbereich kann noch weiter eingeschränkt werden, da sich die Beine während der Bewegung der Plattform gegenseitig behindern können.

Komplexe Kinematik. Die Berechnung der Kinematik paralleler Roboter ist komplex und langwierig. Im Gegensatz zu seriellen Robotern, bei denen die Schwierigkeit bei der Lösung des Problems der inversen Kinematik auftritt, besteht bei Parallelmechanismen die Schwierigkeit bei der Lösung des Problems der direkten Kinematik.

Fatale kinematische Singularitäten. Serienroboter in kinematisch singulären Posen verlieren an Mobilität. Parallele Roboter in singulären Posen gewinnen Freiheitsgrade, die nicht kontrolliert werden können. Das ist eine fatale Situation, weil sie nicht gelöst werden kann.

6.2 Kinematische Anordnungen paralleler Roboter

Wir können eine immense Anzahl kinematischer Anordnungen paralleler Roboter erstellen. In der industriellen Praxis werden davon jedoch nur wenige eingesetzt. Die beliebteste und allgemeinste im kinematischen Sinne ist die Stewart-Gough-Plattform, wie in Abb. 6.4 dargestellt.

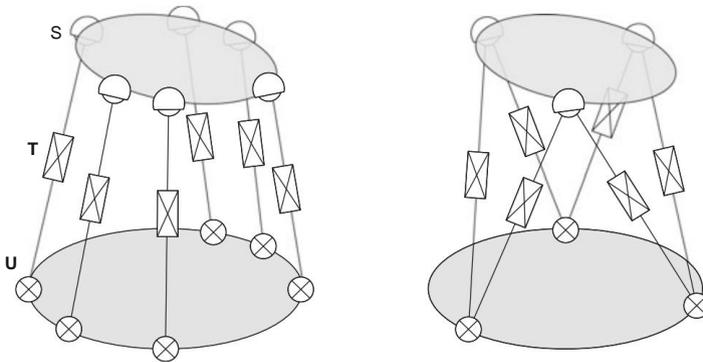


Abb. 6.4 Die Stewart-Gough-Plattform

Stewart-Gough-Plattform

Auf der linken Seite von Abb. 6.4 ist eine allgemeine Stewart-Gough-Plattform dargestellt. Gemäß unserer Bezeichnung handelt es sich um einen Mechanismus vom Typ 6UTS. Der Roboter enthält $n = 18$ Gelenke, $N = 13$ bewegliche Glieder und die Summe von $f_{i,i} = 1, 2, \dots, n$ beträgt 36. Dies ergibt das erwartete Ergebnis

$$F = 6(13 \dot{\vee} 18) + 36 = 6$$

Freiheitsgrade. Die Plattform dieses Roboters kann unter der Steuerung von sechs Motoren, typischerweise den sechs Translationsmotoren, räumlich positioniert und ausgerichtet werden. Durch Verkürzen oder Verbreitern der Beine (Änderung der Beinlänge) kann die Plattform in eine gewünschte Pose (Position und Ausrichtung) gebracht werden. Ein besonderer Vorteil der Stewart Gough-Plattform mit den UTS-Beinen besteht darin, dass auf die Plattform einwirkende Lasten in Form einer Längskraft in Richtung des Beins auf jedes einzelne Bein übertragen werden und keine Querbelastung auf die Beine ausgeübt wird. Diese Besonderheit ermöglicht hervorragende dynamische Leistungen.

Die Anzahl der Freiheitsgrade einer UTS-Strecke beträgt sechs und die Anzahl der Einschränkungen beträgt Null. Wenn wir Grüblers Formel (6.5) betrachten, lässt sich leicht überprüfen, dass die Anzahl der UTS-Beine keinen Einfluss auf die Anzahl der Freiheitsgrade des Roboters hat und dass die Mobilität der Stewart-Gough-Plattform nicht von der Anzahl der Beine abhängt.

Ein Roboter mit nur einem UTS-Bein, bei dem es sich um einen seriellen Roboter handelt, verfügt über sechs Freiheitsgrade, genau wie der völlig parallele ursprüngliche Stewart-Gough-Roboter mit sechs Beinen.

Der sechsbeinige Mechanismus auf der rechten Seite von Abb. 6.4 stellt schematisch die ursprüngliche Stewart-Gough-Plattform dar, die eine zentralsymmetrische Sternform aufweist. Bei dieser Anordnung werden jeweils zwei Beine an einem Punkt festgeklemmt, an dem zwei überlappende, zusammenfallende Kugelgelenke (oder Universalgelenke) angebracht sind. Daher beträgt die Anzahl der unabhängigen Kugelgelenke sechs und die gleiche Anzahl der Universalgelenke. Die überlappenden Gelenke vereinfachen nicht nur die Konstruktion, sondern ermöglichen auch einfachere Berechnungen der Roboterkinematik und -dynamik.

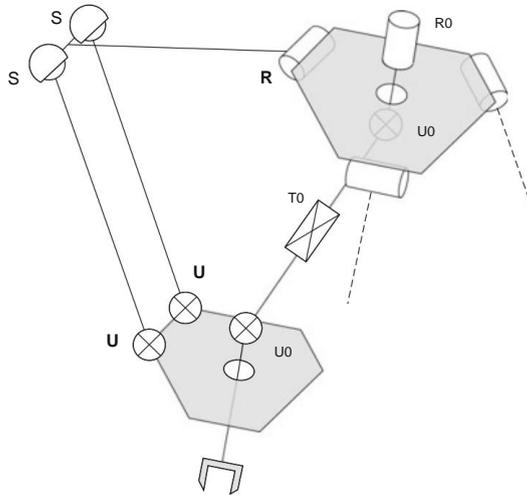


Abb. 6.5 Der Delta-Roboter

Delta-Roboter

Aufgrund seiner spezifischen Bewegungseigenschaften und seiner zahlreichen Anwendungen in der Industrie hat der Delta-Roboter seinen Platz unter den Roboterherstellern gefunden (siehe Abb. 6.5). Die Kinematik dieses Roboters ist sehr ausgefeilt. Das Hauptziel seines Schöpfers war die Entwicklung eines leichten Roboters mit extrem dynamischen Leistungen.

Die feste Basis des Roboters ist das obere Sechseck, während das untere Sechseck die bewegliche Plattform darstellt. Der Roboter hat drei seitliche Beine. In der Abbildung ist nur eines dargestellt, mit einem R-Gelenk, zwei S-Gelenken und zwei U-Gelenken; die anderen beiden Beine sind symbolisch mit einer gepunkteten Linie gezeichnet. Es gibt auch ein unabhängiges Mittelbein $R0U0T0U0$, das keinen Einfluss auf die Bewegung der Plattform hat. Zwischen der Mitte des Beins und der Basis befindet sich ein Parallelogrammmechanismus, der aus zwei Kugelgelenken S und zwei Universalgelenken U besteht. Jedes Bein verfügt also über 3 Glieder und 5 Gelenke. Ohne Berücksichtigung des Mittelschenkels beträgt die Anzahl der Freiheitsgrade des Mechanismus

$$F = 6(10 - 15) + 33 = 3.$$

Die Stellung der Plattform wird durch nur drei Variablen bestimmt. In der Originalversion des Delta-Roboters werden die drei Drehwinkel R in den Seitenbeinen motorisch gesteuert. Aufgrund der Parallelogrammstruktur der Beine führt die Plattform nur eine Translation aus und ist immer parallel zur Basis.

Der Zweck des Mittelbeins besteht darin, die Drehung $R0$ über die Plattform auf den Greifer am Endpunkt des Roboters zu übertragen. Es fungiert als teleskopierbare Antriebswelle zum Drehen des Greifers. Dieses Bein ist ein Kardangelenk mit zwei Universalgelenken $U0$, die durch ein Translationsgelenk $T0$ getrennt sind. Insgesamt verfügt der Mechanismus über vier Freiheitsgrade: drei translatorische Freiheitsgrade, die die räumliche Position des Greifers ermöglichen, und einen rotatorischen, der die

Drehung des Greifers um eine Achse senkrecht zur Plattform. Alle Aktuatoren des Delta-Mechanismus sind an der Basis befestigt und bewegen sich nicht. Daher ist der Mechanismus extrem leicht und die Plattform kann sich mit hohen Geschwindigkeiten und Beschleunigungen bewegen.

Planare Parallelroboter

Die folgenden Beispiele sind planare Parallelroboter, die in einer gegebenen Ebene mit $\ddot{y} = 3$ arbeiten. Das erste Beispiel ist in Abb. 6.6 links dargestellt. Der Roboter enthält drei Beine vom Typ RTR-RRR-RRR. Als Ergebnis haben wir $N = 7$ und $n = 9$ und die Gesamtzahl der Freiheitsgrade in Gelenken beträgt 9. Gemäß Gl. (6.3) beträgt die Anzahl der Freiheitsgrade dieses Roboters

$$F = 3(7 - 9) + 9 = 3.$$

Das Ergebnis ist zu erwarten, da alle Zweige keine Einschränkungen einführen (6.5). Dadurch kann die Plattform jede gewünschte Pose innerhalb des Arbeitsbereichs einnehmen. Beachten Sie, dass in der Ebene zwei Freiheitsgrade für die Position (Translationen in der x-y-Ebene) und ein Freiheitsgrad für die Ausrichtung (eine Drehung um die z-Achse) erforderlich sind. Um diesen Roboter zu aktivieren, werden drei Motoren benötigt. Zur Befestigung der Motoren können wir jedes der neun Gelenke auswählen. Normalerweise bevorzugen wir die an der Basis befestigten Gelenke, damit sich die Motoren nicht bewegen und ihr Gewicht die Roboterdynamik nicht beeinflusst. Im Einzelfall kann das Translationsgelenk auch über eine Elektroschraube oder einen Hydraulizylinder motorisiert werden.

In Abb. 6.6 rechts ist ein ähnlicher planarer Parallelroboter dargestellt, dessen Aufbau RTR-RR-RR ist. Hier können wir sehen, dass jeder der beiden RR-Zweige eine Einschränkung einführt. Nach Gl. (6.5) beträgt die Anzahl der Freiheitsgrade dieses Parallelroboters

$$F = 3 - 2 = 1.$$

Der Roboter wird mit einem Motor gesteuert. Die Plattform ist nur begrenzt beweglich und kann sich nur entlang einer Kurve in der x-y-Ebene bewegen. Wir können zum Beispiel entweder das positionieren

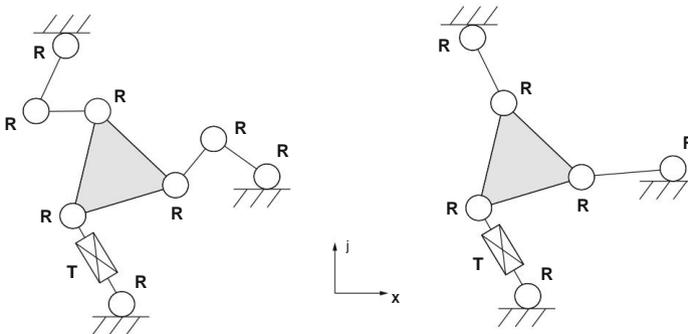


Abb. 6.6 Planare Parallelroboter

Plattform entlang der x-Achse, ohne Kontrolle über y und die Ausrichtung der Plattform zu haben, oder alternativ die Plattform ausrichten, ohne Kontrolle über ihre Position in x und y zu haben.

Parallele humanoide Schulter

Parallele Mechanismen kommen in der Natur, im menschlichen Körper oder bei Tieren sehr häufig vor. Es ist daher keine Überraschung, dass die Modelle paralleler Roboter effizient zur Simulation biomechanischer Eigenschaften von Menschen eingesetzt werden können, bei denen über die Gelenke gespannte Muskeln und Bänder verschiedene parallele kinematische Strukturen bilden. Beispielsweise kann der Schulterkomplex durch zwei Grundkompositionen dargestellt werden, das sogenannte Innengelenk, das die Bewegung des Schlüsselbeins und des Schulterblatts in Bezug auf den Rumpf umfasst, und das sogenannte Außengelenk, das damit verbunden ist Glenohumeralgelenk. In der heutigen humanoiden Robotik wird die Bewegung des Innengelenks aufgrund seiner mechanischen Komplexität typischerweise vernachlässigt. Dennoch ist sein Beitrag zur menschlichen Bewegung, Armreichweite und Dynamik von entscheidender Bedeutung.

In der Literatur wurde ein paralleler Schultermechanismus vorgeschlagen, der die innere Schulter darstellt. Seine Bewegung ist in Abb. 6.7 dargestellt. Die vorgeschlagene Struktur ist TS-3UTS. Es gibt einen zentralen Zweig T0S0 mit vier Freiheitsgraden und zwei Einschränkungen. Um die Achse des zentralen Beins sind drei UTS-Seitenbeine angebracht, die jeweils sechs Freiheitsgrade besitzen und deren Anzahl an Einschränkungen Null beträgt. Nach Gl. (6.5) beträgt die Anzahl der Freiheitsgrade des Roboters

$$F = 6 \cdot 3 - 2 = 16.$$

Der Roboter kann eine vollständige Ausrichtung der Plattform erzeugen (ungefähr drei Hauptausrichtungswinkel) und sich ähnlich wie die menschliche Schulter ausdehnen oder verkleinern. Der Arm ist über das Glenohumeralgelenk an dieser Plattform befestigt. Die vorgeschlagenen inneren Schultergelenke ahmen die Bewegung des Arms, einschließlich des Schulterzuckens

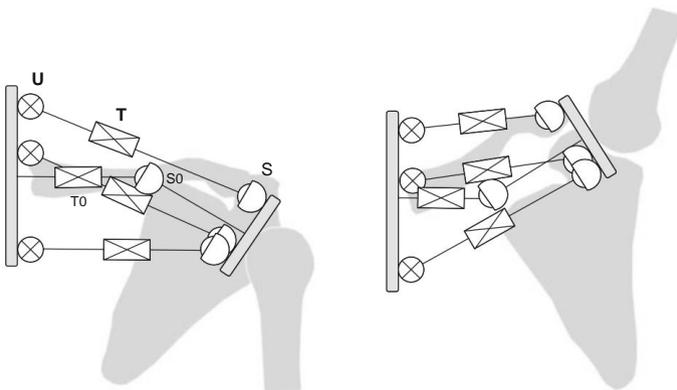


Abb. 6.7 Parallelroboter, der den inneren Schultermechanismus nachahmt

und vermeidet Kollisionen mit der Karosserie und bietet eine hervorragende statische Belastbarkeit und dynamische Fähigkeiten.

6.3 Modellierung und Design paralleler Roboter

Die meisten Parallelroboter, die in der Industrie oder in Forschungslaboren zum Einsatz kommen, verfügen über symmetrische kinematische Anordnungen. Aus konstruktiver Sicht ist es sinnvoll, dass sie aus den gleichen mechanischen Elementen bestehen. Symmetrie trägt auch dazu bei, ihre mathematische Behandlung zu vereinfachen.

Eine häufige Gruppe kinematischer Anordnungen wird durch den zuvor beschriebenen Schulterroboter repräsentiert. Diese Gruppe enthält ein zentrales Bein mit \check{y}_1 Freiheitsgraden, um das sich symmetrisch angeordnete seitliche Beine befinden, die häufig vom Typ UTS sind und $\check{y}_2, \check{y}_3, \dots, \check{y}_K = \check{y}$ Freiheitsgrade (und keine Einschränkungen) besitzen. Das zentrale Bein ist daher entscheidend für die Bestimmung der kinematischen Eigenschaften des gesamten Roboters, da die Anzahl der Freiheitsgrade des Roboters $F = \check{y}_1$ beträgt.

Die zweite Gruppe kinematischer Anordnungen wird durch die Stewart-Gough-Plattform repräsentiert, bei der alle Beine identisch sind und normalerweise vom Typ UTS sind, sodass $\check{y}_1, \check{y}_2, \dots, \check{y}_K = \check{y}$. Wenn $\check{y}_1, \check{y}_2, \dots, \check{y}_K < \check{y}$, sind nur wenige solcher Roboter beweglich, die meisten ihrer Strukturen weisen null oder negative Freiheitsgrade auf. Roboter mit einer negativen Anzahl an Freiheitsgraden werden als überbestimmt bezeichnet.

Betrachten Sie die zweite Gruppe von Robotern (Gough-Stewart-ähnliche kinematische Struktur) mit einem einzelnen Motor in jedem Bein. Ein solcher Roboter muss $K = F$ Beine haben, da ein Roboter mit $K < F$ nicht gesteuert werden kann. Es ist leicht zu überprüfen, dass nur die folgenden Roboter im Weltraum existieren können (wobei $\check{y} = 6$)

$$\begin{aligned} K = 1, \check{y}_1 &= 1 \\ K = 2, \check{y}_1 = \check{y}_2 &= 4 \\ K = 3, \check{y}_1 = \check{y}_2 = \check{y}_3 &= 5 \\ K = 6, \check{y}_1 = \check{y}_2 = \dots = \check{y}_6 &= 6 \end{aligned}$$

Roboter dieser Gruppe mit vier und fünf Beinen gibt es nicht. In der Ebene mit $\check{y} = 3$ können nur die folgenden Roboter existieren

$$\begin{aligned} K = 1, \check{y}_1 &= 1 \\ K = 3, \check{y}_1 = \check{y}_2 = \check{y}_3 &= 3 \end{aligned}$$

Im planaren Fall gibt es keine Roboter mit zwei Beinen.

Kinematische Parameter und Koordinaten paralleler Roboter

In Abb. 6.8 ist der Koordinatenrahmen x - y - z an der beweglichen Plattform befestigt, während x_0 - y_0 - z_0 an der Basis befestigt ist. Die Position der Plattform wird in Bezug auf den festen Koordinatenrahmen durch den Vektor r angegeben; seine Komponenten sind r_x, r_y, r_z . Die Au

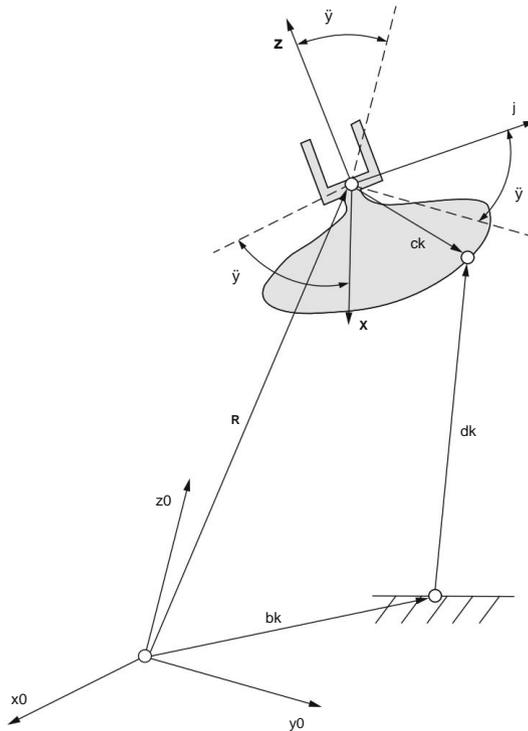


Abb. 6.8 Kinematische Parameter eines Parallelroboters

der Plattform kann durch ein ausgewähltes Triplet von Orientierungswinkeln \tilde{y} , \tilde{y} , \tilde{y} beschrieben werden, die zwischen beiden Koordinatensystemen auftreten (Einzelheiten siehe Kap. 4).

Der Vektor b_k definiert die Befestigung des Beins k an der Basis, ausgedrückt im Rahmen $x_0-y_0-z_0$, während der Vektor c_k die Befestigung desselben Beins an der Plattform im Rahmen $x-y-z$ definiert. Die Vektoren

$$d_k = r + R c_k \tilde{y} b_k, \quad k = 1, 2, \dots, K, \quad (6.6)$$

Beschreiben Sie die Geometrie der Roboterbeine, ausgedrückt im Koordinatensystem $x_0-y_0-z_0$. Hier ist $R = R(\tilde{y}, \tilde{y}, \tilde{y})$ die 3×3 -Rotationsmatrix, die den Koordinatenrahmen $x-y-z$ in $x_0-y_0-z_0$ transformiert. Gleichung (6.6) lässt sich auch in homogener Form wie folgt formulieren

$$d_k = H c_k \tilde{y} b_k \quad 0 \quad 0 \quad 0 \quad 1, \quad k = 1, 2, \dots, K, \quad (6.7)$$

wo die homogene Transformationsmatrix ist

$$H = \begin{bmatrix} R & r & \tilde{y} & b_k & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.8)$$

Wir gehen davon aus, dass die Beinlängen die Gelenkkordinaten des Roboters sind

$$q_k = d_k, \quad k = 1, 2, \dots, K, \quad (6.9)$$

wobei $\|\cdot\|$ die Vektornorm angibt. Sie sind Elemente des Vektors der Gelenkkordinaten

$$\mathbf{q} = (q_1, q_2, \dots, q_K)^T.$$

Die kinematischen Parameter des Roboters sind die Vektoren \mathbf{b}_k , $k = 1, 2, \dots, K$, ausgedrückt im Rahmen $x_0-y_0-z_0$, und die Vektoren \mathbf{c}_k , ausgedrückt im Rahmen $x-y-z$.

Nachdem wir die internen Koordinaten definiert haben, schauen wir uns die externen Koordinaten des Roboters an. Bei Parallelrobotern repräsentieren sie normalerweise einige Merkmale in der Bewegung der Plattform, an der der Endeffektor befestigt ist. In den meisten Fällen handelt es sich bei den gewählten externen Koordinaten um die Position und Ausrichtung der Plattform, die sogenannten kartesischen Koordinaten. Im Raum mit $\dot{y} = 6$ umfassen sie die drei Komponenten r_x, r_y, r_z des Ortsvektors in Abb. 6.8 und die drei Orientierungswinkel $\tilde{y}, \tilde{\alpha}, \tilde{\beta}$, so dass der Vektor der äußeren Koordinaten wie folgt definiert ist

$$\mathbf{p} = (r_x, r_y, r_z, \tilde{y}, \tilde{\alpha}, \tilde{\beta})^T.$$

Inverse und direkte Kinematik paralleler Roboter

Aus steuerungstechnischer Sicht ist die Beziehung zwischen den Außen- und Innenkoordinaten von größter Bedeutung. Ihre Beziehung wird, ähnlich wie bei seriellen Robotern, durch sehr komplizierte algebraische trigonometrische Gleichungen bestimmt.

Das Problem der inversen Kinematik paralleler Roboter erfordert die Bestimmung der inneren Koordinaten \mathbf{q} , also der Beinlängen, aus einem gegebenen Satz äußerer Koordinaten \mathbf{p} , die die Position und Ausrichtung der Plattform darstellen. Für einen gegebenen Satz externer Koordinaten \mathbf{p} können die internen Koordinaten durch einfaches Lösen von Gl. (6.7). Anders als bei seriellen Robotern ist es hier wichtig zu erkennen, dass die Werte der externen Koordinaten die Beinlängen des parallelen Roboters eindeutig definieren und die Berechnung unkompliziert ist.

Das Problem der direkten Kinematik paralleler Roboter erfordert die Bestimmung der Außenkoordinaten \mathbf{p} aus einem gegebenen Satz von Gelenkkordinaten \mathbf{q} (Abb. 6.9). Dieses Problem ist mathematisch äußerst kompliziert und die Berechnungsverfahren sind umständlich. Im Allgemeinen ist es nicht möglich, die externen Koordinaten als explizite Funktionen der internen Koordinaten auszudrücken, während dies bei seriellen Robotern recht einfach ist. In der Regel handelt es sich dabei um gekoppelte trigonometrische und quadratische Gleichungen, die nur in Sonderfällen in geschlossener Form gelöst werden können. Es gibt keine Regeln für den Umgang mit symbolischen Lösungen. Folgende Schwierigkeiten treten häufig auf:

Keine echte Lösung. Für einige Werte der internen Koordinaten gibt es keine echten Lösungen für die externen Koordinaten. In welchen Intervallen interner Koordinaten dies passieren kann, lässt sich nicht im Voraus vorhersagen.

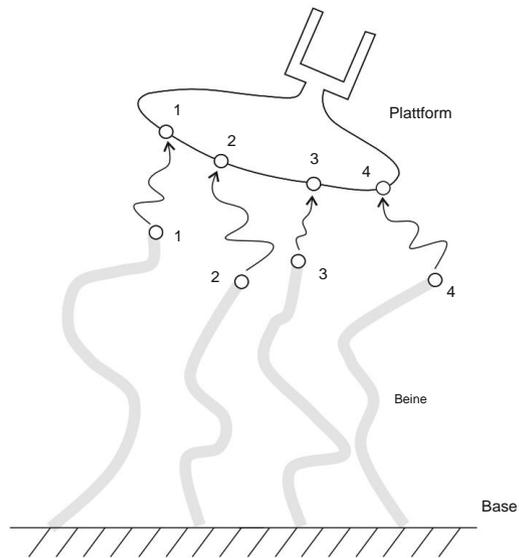


Abb. 6.9 Das Problem der direkten Kinematik besteht darin, die Stellung der Plattform entsprechend der Länge der Beine zu finden. Die Endpunkte der Beine müssen mit den entsprechenden Punkten auf der Plattform übereinstimmen (z. B. 1 $\dot{\gamma}$ 1).

Mehrere Lösungen. Für einen gegebenen Satz interner Koordinaten gibt es mehrere Lösungen für die externen Koordinaten. Die Anzahl der Lösungen für eine gegebene Kombination von Beinlängen hängt von der kinematischen Struktur des Mechanismus ab. Die allgemeine Stewart-Gough-Plattform bietet vierzig mögliche Lösungen für das Problem der direkten Kinematik. Für eine ausgewählte Kombination von Beinlängen gibt es vierzig verschiedene Posen der Plattform. Darüber hinaus ist es manchmal nicht möglich, zwischen zwei Posen der Plattform zu wechseln, da sich die Beine verheddern. In solchen Fällen könnte die Plattform nur durch Demontage der Beine in der ersten Pose und Wiederausbau in der neuen Pose von einer Pose in eine andere übergehen.

Keine geschlossenen Lösungen. Im Allgemeinen ist es für einen gegebenen Satz von Gelenkkkoordinaten nicht möglich, eine exakte Lösung für das Problem der direkten Kinematik zu finden, selbst wenn eine reale Lösung existiert. In solchen Fällen verwenden wir numerische Verfahren, die möglicherweise nicht unbedingt konvergieren und möglicherweise nicht alle Lösungen finden.

Entwurf von Parallelrobotern

Die Gestaltung paralleler Roboter hängt von der gewünschten Leistung, Flexibilität, Mobilität und Belastbarkeit sowie dem tatsächlichen Arbeitsbereich ab.

Bei der Betrachtung der Arbeitsbereiche für parallele und serielle Roboter beziehen wir uns auf den erreichbaren Arbeitsbereich und den geschickten Arbeitsbereich. Einer der größten Nachteile von Parallelrobotern ist ihr kleiner Arbeitsbereich. Das Hauptziel der Arbeitsbereichsanalyse besteht daher darin, festzustellen, ob eine gewünschte Flugbahn innerhalb des Roboterarbeitsraums liegt. Die Größe des Arbeitsbereichs bei Parallelrobotern wird durch die Bereiche der Ver-

der Beine, Verschiebungen passiver Gelenke und insbesondere durch Interferenzen zwischen den Beinen des Roboters. Schon bei kleinen Bewegungen können die Beine miteinander kollidieren. Das Ineinandergreifen der Beine stellt in der Praxis ein großes Hindernis für die Bewegung und Erreichbarkeit eines Roboters dar. Die Bestimmung und Analyse des Roboterarbeitsraums ist im Allgemeinen ein langwieriger Prozess. Bei Parallelrobotern ist es meist noch komplexer, abhängig von der Anzahl der Freiheitsgrade und der Archi

Der Einfluss der Last wird bei seriellen Robotern üblicherweise in der Dynamik gesehen, die zu einem großen Teil die Trägheit der Glieder einschließt. Der Beitrag einer externen Kraft ist typischerweise geringer und kann in vielen Fällen vernachlässigt werden. Bei Parallelrobotern mit vielen Beinen, bei denen die Glieder sehr leicht sind und die Motoren typischerweise an der festen Basis befestigt sind, spielt die Roboterstatik eine wichtige Rolle. Die Berechnung der Roboterstatik basiert auf der bekannten Jacobi-Matrix, die die Transformation zwischen den äußeren und inneren Koordinaten darstellt. Dies würde den Rahmen unseres Buches sprengen, dem interessierten Leser steht jedoch umfangreiche Literatur, Artikel und Lehrbücher zur Verfügung.

In der Praxis können wir oft eine Stewart-Gough-Plattform sehen, die eine Struktur hat, wie sie in Abb. 6.10 dargestellt ist. Der Roboter enthält (anstelle von sechs Beinen vom Typ UTS) sechs Beine vom Typ STS. Kinematisch ist diese Architektur recht ungewöhnlich und überflüssig. Der Roboter hat zu viele Freiheitsgrade. Jedes Bein besitzt 7 Freiheitsgrade

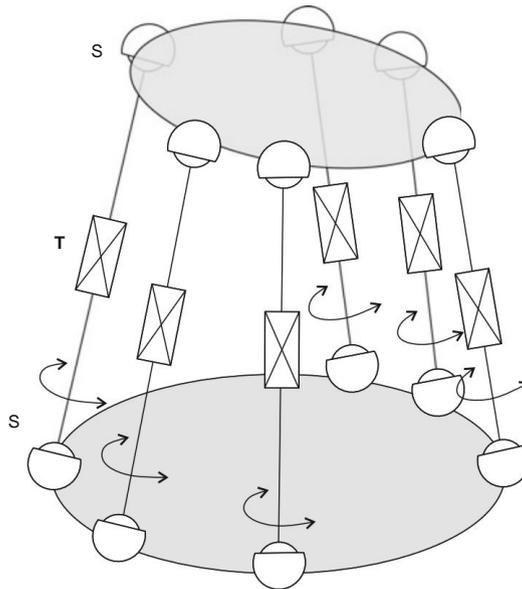


Abb. 6.10 Eine Modifikation der Stewart-Gough-Plattform

was \dot{y}_1 Einschränkungen entspricht. Nach Grüblers Formel (6.5) beträgt die Anzahl der Freiheitsgrade des Roboters

$$F = 6 - \dot{y}_6 = 12.$$

Es ist wichtig zu beachten, dass sechs der zwölf Freiheitsgrade sich in Drehungen der Beine um ihre eigene Achse manifestieren. Diese Drehungen haben keinen Einfluss auf die Bewegung der Plattform. Somit kann der Roboter immer noch mit nur sechs Motoren motorisiert werden, die die Länge der Beine verändern und sich auf die Translation T auswirken, während die Drehungen um die Beinachsen passiv bleiben und sich frei ändern können. Die Vorteile dieser Konstruktion bestehen darin, dass S-Gelenke einfacher zu bauen sind als U-Gelenke (und damit kostengünstiger) und dass die passiven Drehungen um die Beinachsen eine größere Flexibilität beim Anschluss von Strom- und Signalkabeln ermöglichen, da diese häufig entlang der Beine angeordnet sind die Basis zur

Kapitel 7

Robotersensoren



Das menschliche Sinnessystem umfasst Seh- und Hörsensoren, kinästhetische Sensoren (Bewegung, Kraft und Berührung) sowie Geschmacks- und Geruchssensoren. Diese Sensoren liefern Eingangssignale an das Gehirn, das anhand dieser sensorischen Informationen ein eigenes Bild der Umgebung erstellt und Entscheidungen für weitere Maßnahmen trifft. Ähnliche Anforderungen gelten auch für Robotermechanismen. Aufgrund der Komplexität der menschlichen Sensorik ist die Robotersensorik jedoch auf weniger Sensoren beschränkt.

Für den effizienten und präzisen Roboterbetrieb ist der Einsatz von Sensoren von entscheidender Bedeutung. Robotersensoren können im Allgemeinen unterteilt werden in: (1) propriozeptive Sensoren, die die internen Zustände des Robotermechanismus (Positionen, Geschwindigkeiten und Drehmomente in den Roboterjunkten) bewerten; und (2) exterozeptive Sensoren, die der Steuerung Informationen über die Roboterumgebung liefern (Kraft-, Tast-, Näherungs- und Distanzsensoren, Robotersicht).

7.1 Prinzipien der Wahrnehmung

Im Allgemeinen wandeln Sensoren die gemessene physikalische Größe in ein elektrisches Signal um, das in digitaler Form vom Computer ausgewertet werden kann. In der Robotik interessieren uns vor allem die folgenden Größen: Position, Geschwindigkeit, Kraft und Drehmoment. Durch den Einsatz spezieller Wandler können diese Größen in elektrische Signale wie Spannung, Strom, Widerstand, Kapazität oder Induktivität umgewandelt werden. Nach dem Umrechnungsprinzip lassen sich die Sensoren wie folgt einteilen:

- Elektrische Sensoren – die physikalische Variable wird direkt in ein elektrisches Signal umgewandelt; solche Sensoren sind beispielsweise Potentiometer oder Dehnungsmessstreifen;
- Elektromagnetische Sensoren – nutzen das Magnetfeld zur physikalischen Messung; eine häufige Bekehrung; ein Beispiel ist der Drehzahlmesser;

- **Optische Sensoren** – nutzen Licht zur Signalumwandlung; ein Beispiel dafür ist der optische Encoder.

7.2 Bewegungssensoren

Typische Sensoren für Roboterbewegungen sind Potentiometer, optische Encoder und Tachometer. Sie alle messen die Roboterbewegungen innerhalb des Roboterjoints. Es ist wichtig, wo im Gelenk der Bewegungssensor platziert wird und wie die Bewegungsparameter gemessen werden.

7.2.1 Platzierung der Sensoren

Betrachten wir zunächst einen Winkelverschiebungssensor. Unser Ziel ist es, den Winkel in einem Roboterjoint zu messen, das von einem Motor über ein Untersetzungsgetriebe mit dem Untersetzungsverhältnis k_r betätigt wird. Mithilfe eines Untersetzungsgetriebes verringern wir die Winkelgeschwindigkeit des Gelenks um den Faktor k_r gegenüber der Winkelgeschwindigkeit des Motors. Gleichzeitig erhöht sich das Gelenkdrehmoment um den gleichen Faktor. Es ist wichtig, ob der Bewegungssensor vor oder nach dem Reduzierstück platziert wird. Die Auswahl hängt von den Aufgabenanforderungen und dem verwendeten Sensor ab. Im Idealfall montieren wir den Sensor vor dem Reduzierstück (an der Seite des Motors), wie in [Abb. 7.1 dargestellt](#). Auf diese Weise messen wir direkt die Umdrehungen des Motors. Der Sensorausgang muss dann durch das Untersetzungsverhältnis dividiert werden, um den Gelenkwinkel zu erhalten.

Bezeichnen wir mit $\dot{\gamma}$ die Winkelposition des Gelenks, $\dot{\gamma}_m$ die Winkelposition des entsprechenden Motors und k_r das Untersetzungsverhältnis des Untersetzungsgetriebes. Wenn der Sensor

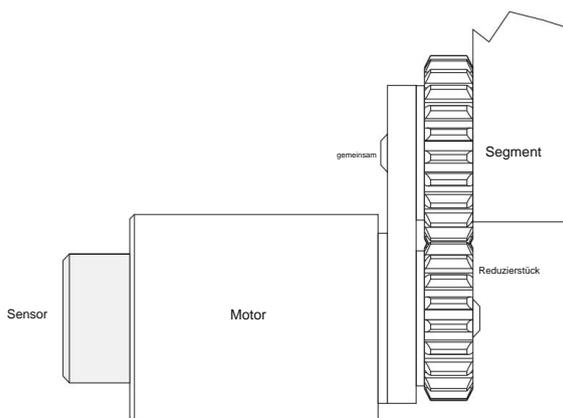


Abb. 7.1 Montage des Bewegungssensors vor dem Reduzierstück

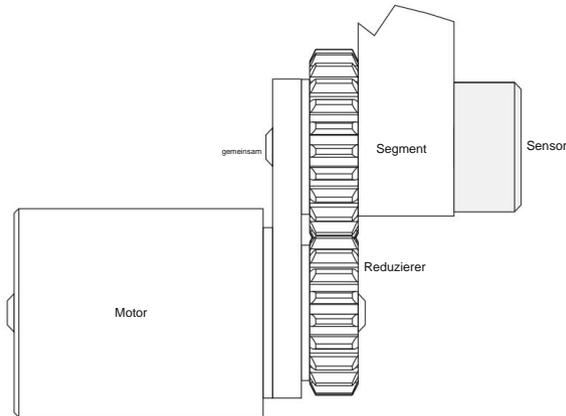


Abb. 7.2 Montage des Bewegungssensors hinter dem Reduzierstück

vor dem Reduzierstück platziert wird, ist sein Ausgang gleich dem Winkel $\dot{\gamma}_m$. Die Größe, die wir zur Steuerung benötigen, ist der Gelenkwinkel $\dot{\gamma}$, der durch das Verhältnis bestimmt wird

$$\dot{\gamma} = \frac{\dot{\gamma}_m}{k_r} \quad (7.1)$$

Durch Differenzierung der Gl. (7.1) bezüglich $\dot{\gamma}_m$ gilt

$$\frac{d\dot{\gamma}}{d\dot{\gamma}_m} = \frac{1}{k_r} \quad \text{1 also } d\dot{\gamma} = d\dot{\gamma}_m \cdot k_r \quad (7.2)$$

Das bedeutet, dass sich der Messfehler des Sensors um den Faktor k_r verringert. Der Vorteil der Platzierung des Sensors vor dem Reduzierstück besteht darin, dass genauere Informationen über die Winkelposition des Gelenks erhalten werden.

Eine weitere Möglichkeit der Sensormontage ist in Abb. 7.2 dargestellt. Hier wird der Sensor hinter dem Reduzierstück montiert. Auf diese Weise werden die Bewegungen des Gelenks direkt gemessen. Die Qualität des Regelsignals nimmt ab, da der nun nicht reduzierte Messfehler des Sensors direkt in den gemeinsamen Regelkreis einfließt. Da der Bewegungsbereich des Gelenks um den Faktor k_r kleiner ist als der des Motors, können Sensoren mit kleinerem Bewegungsbereich verwendet werden. Manchmal kommt man nicht umhin, den Bewegungssensor in die Gelenkachse einzubauen. Daher ist es wichtig, dass wir uns der Mängel einer solchen Unterbringung bewusst sind.

7.2.2 Potentiometer

Abbildung 7.3 zeigt ein Modell eines Drehpotentiometers und seiner Komponenten. Das Potentiometer besteht aus zwei Teilen: (1) Widerstandswicklung und (2) beweglichem Schleifer.

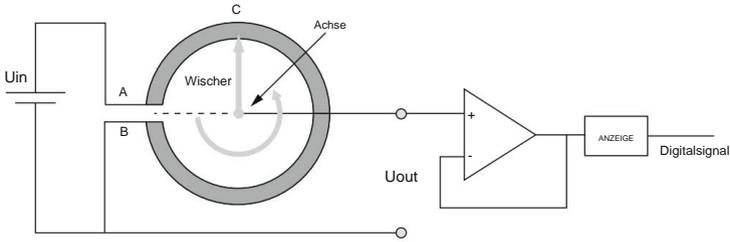


Abb. 7.3 Das Modell eines Potentiometers

Das Potentiometer stellt eine Kontaktmessmethode dar, da der Schleifer entlang der kreisförmigen Widerstandswicklung gleitet.

Potentiometer werden im Allgemeinen so hinter dem Reduzierstück platziert, dass die Potentiometerachse mit der Gelenkachse gekoppelt ist. Nehmen wir an, dass Punkt B die Referenzposition des zum Gelenk gehörenden Potentiometers darstellt. Der Widerstand des Potentiometers entlang der Wicklung AB ist gleich R , während r den Widerstand des CB- Teils der Wicklung darstellt. Der Winkel des Wischers in Bezug auf die Referenzposition B wird mit $\dot{\gamma}$ (im Bogenmaß) bezeichnet. Wenn der Widerstand entlang der kreisförmigen Wicklung des Potentiometers gleichmäßig ist und der Abstand zwischen den Punkten A und B vernachlässigbar ist, erhalten wir die folgende Gleichung

$$\frac{R}{R} = \frac{CB}{AB} = \frac{\dot{\gamma}}{2\dot{\gamma}}. \quad (7.3)$$

Nehmen wir an, dass das Potentiometer durch die Spannung U_{in} versorgt wird. Die am Schleifer gemessene Ausgangsspannung ist gleich

$$\frac{U_{out}}{U_{in}} = \frac{R}{R} = \frac{\dot{\gamma}}{2\dot{\gamma}}, \quad (7.4)$$

oder

$$U_{out} = \dot{\gamma} \cdot \frac{U_{in}}{2\dot{\gamma}} \quad (7.5)$$

Durch Messung der Ausgangsspannung U_{out} wird die Winkelposition $\dot{\gamma}$ ermittelt.

7.2.3 Optischer Encoder

Der Ansatz zur Kontaktmessung des Robotergelenkwinkels mithilfe von Potentiometern weist mehrere Mängel auf. Am wichtigsten ist die relativ kurze Lebensdauer aufgrund des hohen Verschleißes. Darüber hinaus ist die Platzierung am besten direkt in der Gelenkachse (hinter dem Reduzierstück) und nicht auf der Motorachse (vor dem Reduzierstück). Der

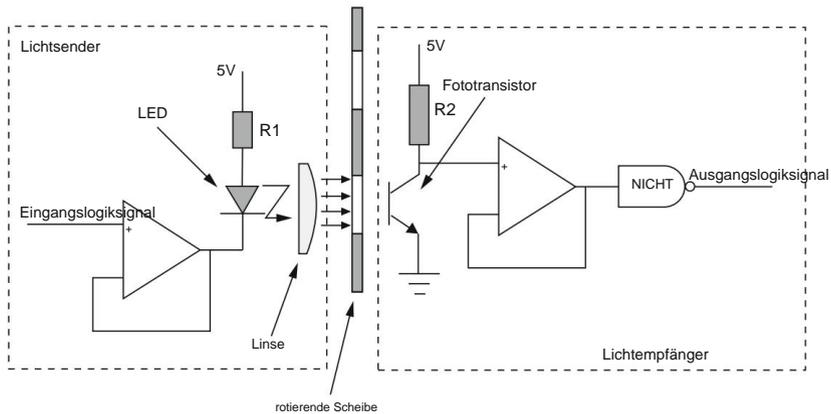


Abb. 7.4 Das Modell des optischen Encoders

Die am weitesten verbreiteten Bewegungssensoren in der Robotik sind daher optische Encoder, die eine berührungslose Messung ermöglichen.

Der optische Encoder basiert auf der Umwandlung der Gelenkbewegung in eine Reihe von Lichtimpulsen, die weiter in elektrische Impulse umgewandelt werden. Zur Erzeugung der Lichtimpulse wird eine Lichtquelle benötigt, meist eine Leuchtdiode. Die Umwandlung von Licht in elektrische Impulse erfolgt durch einen Fototransistor oder eine Fotodiode, die Licht in elektrischen Strom umwandelt.

Das Modell eines optischen Encoders zur Bestimmung der Gelenkwinkelstellung ist in [Abb. 7.4 dargestellt](#). Es besteht aus einer Lichtquelle mit Linse, einem Lichtdetektor und einer rotierenden Scheibe mit Schlitzen, die entweder mit einem Motor oder einer Gelenkachse verbunden ist. Auf der rotierenden Scheibe befindet sich eine Spur aus Schlitzen und Zwischenräumen, die abwechselnd das Licht von der Leuchtdiode zum Fototransistor weiterleiten oder blockieren. Der logische Ausgang des Sensors ist hoch, wenn das Licht durch den Schlitz fällt und auf den Fototransistor auf der anderen Seite der Drehplatte trifft. Wenn der Pfad zwischen der Leuchtdiode und dem Fototransistor durch den Zwischenraum zwischen zwei Steckplätzen blockiert ist, ist der logische Ausgang niedrig.

Die optischen Encoder werden in absolute und inkrementelle Encoder unterteilt. Im weiteren Text werden wir ihre wichtigsten Eigenschaften kennenlernen.

7.2.3.1 Absolutwertgeber

Der absolute optische Encoder ist ein Gerät, das die absolute Winkelposition eines Gelenks misst. Sein Ausgang ist ein digitales Signal. In einem digitalen System repräsentiert jede logische Signalleitung ein Informationsbit. Wenn alle diese Bits zu einer einzigen logischen Zustandsvariablen verbunden werden, bestimmt die Anzahl aller möglichen logischen Zustände die Anzahl aller absoluten Winkelpositionen, die vom Encoder gemessen werden können.

Nehmen wir an, wir möchten die Winkeldrehung von 360° mit dem Absolutwertgeber Auflösung von $0,1^\circ$ messen. Der Absolutwertgeber muss zwischen 3600 verschiedenen logischen Zuständen unterscheiden, was bedeutet, dass wir mindestens 12 Bit benötigen, um die Gelenkwinkel mit der erforderlichen Auflösung zu ermitteln. Mit 12 Bit können wir 4096 logische Zustände darstellen. Ein wichtiger Konstruktionsparameter von Absolutwertgebern ist daher die Anzahl der logischen Zustände, die von den Aufgabenanforderungen und der Platzierung des Gebers (vor oder nach dem Reduzierstück) abhängt. Wenn der Encoder vor einem Untersetzungsgetriebe mit dem Übersetzungsverhältnis k_r platziert wird, erhöht sich die Auflösung der Winkelmessung um den Faktor k_r . Wenn sich der Encoder hinter dem Reduzierstück befindet, wird die erforderliche Auflösung des Encoders direkt durch die erforderliche Auflösung der Gelenkwinkelmessung bestimmt. Alle logischen Zustände müssen einheitlich in die rotierende Scheibe des Encoders eingraviert werden. Ein Beispiel für einen Absolutwertgeber mit sechzehn logischen Zuständen ist in Abb. 7.5 dargestellt. Die sechzehn logischen Zustände können durch vier Bits dargestellt werden. Alle sechzehn logischen Zustände sind in die Oberfläche der rotierenden Scheibe eingraviert. Die Platte ist in radialer Richtung in vier Spuren unterteilt, die die vier Bits darstellen. Jede Spur ist entsprechend den logischen Zuständen in sechzehn Segmente unterteilt. Da die Informationen über die Winkelverschiebung durch vier Bits dargestellt werden, benötigen wir vier Paare von Leuchtdioden und Fototransistoren (ein Paar für jedes Bit). Mit der Drehung der Scheibe, die entweder mit dem Motor oder der Gelenkachse verbunden ist, ändert sich das Ausgangssignal entsprechend den logischen Zuständen, die durch die Reihenfolge der Segmente definiert sind (typischerweise

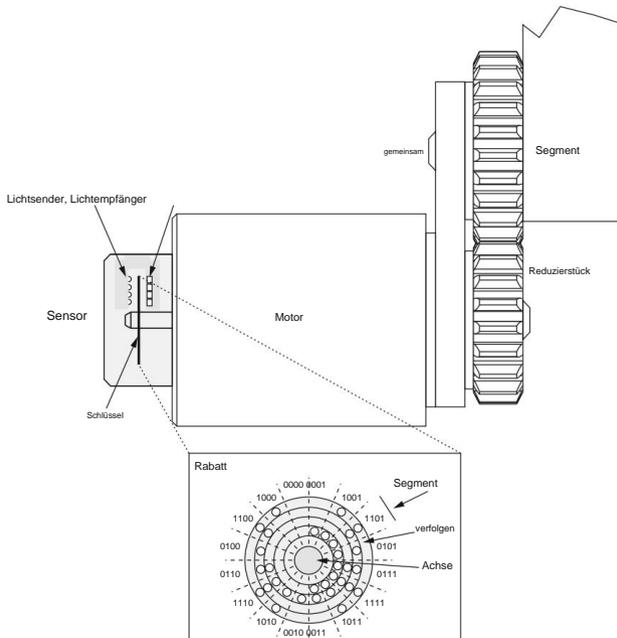


Abb. 7.5 Modell eines Absolutwertgebers

Der Absolutwertgeber ermittelt nicht nur die Winkelposition des Gelenks, sondern auch die Drehrichtung.

7.2.3.2 Inkrementalgeber

Im Gegensatz zu Absolutwertgebern liefern die Inkrementalgeber lediglich Informationen über die Änderungen der Winkelstellung des Gelenks. Die Vorteile von Inkrementalgebern im Vergleich zu Absolutwertgebern sind ihre Einfachheit, kleinere Abmessungen und (am wichtigsten) niedrige Kosten. Dies kann erreicht werden, indem die Anzahl der Spuren auf der rotierenden Scheibe auf nur eine einzige Spur verringert wird. Anstatt so viele Spuren zu haben, wie Bits zur Darstellung aller erforderlichen logischen Zustände erforderlich sind, haben wir jetzt nur noch eine Spur mit gleichmäßiger Abstufung der Schlitze entlang des Scheibenrandes. Abbildung 7.6 zeigt ein Modell eines Inkrementalgebers. Für eine einzelne Spur ist nur ein einziges Paar aus Leuchtdiode und Fototransistor (optisches Paar) erforderlich. Während der Drehung der kodierten Scheibe wird eine Reihe elektrischer Impulse erzeugt. Die Messung der Gelenkverschiebung basiert auf der Zählung dieser Impulse. Ihre Anzahl ist proportional zur Gelenkbewegung des Roboters. Der in Abb. 7.6 dargestellte Inkrementalgeber erzeugt bei jeder Umdrehung acht Impulse. Die Auflösung dieses Encoders beträgt

$$\ddot{y} = \frac{2\dot{y}}{8} = \frac{\dot{y}}{4}. \quad (7.6)$$

Durch die Erhöhung der Anzahl der Slots auf der Disc wird die Auflösung des Encoders erhöht. Indem man die Anzahl der Schlitze als n_c bezeichnet, kann eine allgemeine Gleichung für die Encoderauflösung geschrieben werden

$$\ddot{y} = \frac{2\dot{y}}{n_c}. \quad (7.7)$$

Der Encoder mit einer einzigen Spur ist nur in der Lage, die Änderung der Gelenkwinkelstellung zu beurteilen. Es kann keine Auskunft über die Drehrichtung oder die absolute Gelenkposition geben. Wenn wir Inkrementalgeber in der Robotersteuerung einsetzen wollen, müssen wir Folgendes bestimmen: (1) die Ausgangsposition, die die Referenz für die Messung der Änderung der Gelenkposition darstellt, und (2) die Drehrichtung.

Das Problem der Ausgangsposition wird durch das Hinzufügen eines zusätzlichen Referenzschlitzes auf der Scheibe gelöst. Dieser Referenzschlitz wird gegenüber der Schlitzspur, die die Winkelposition misst, radial verschoben. Zur Erkennung der Home-Position ist ein zusätzliches optisches Paar erforderlich. Bei der Suche nach dem Referenzschlitz wird der Roboter so programmiert, dass er sich mit niedriger Geschwindigkeit bewegt, solange der Referenzschlitz oder die Endposition des Gelenkbewegungsbereichs erreicht ist. Im letzteren Fall bewegt sich der Roboter in die entgegengesetzte Richtung zum Referenzschlitz.

Das Problem der Drehrichtungsbestimmung wird durch ein weiteres Paar aus Leuchtdiode und Fototransistor gelöst. Dieses zusätzliche optische Paar ist tangential und radial vom ersten optischen Paar versetzt, wie in Abb. 7.6 dargestellt. Wenn die Disc

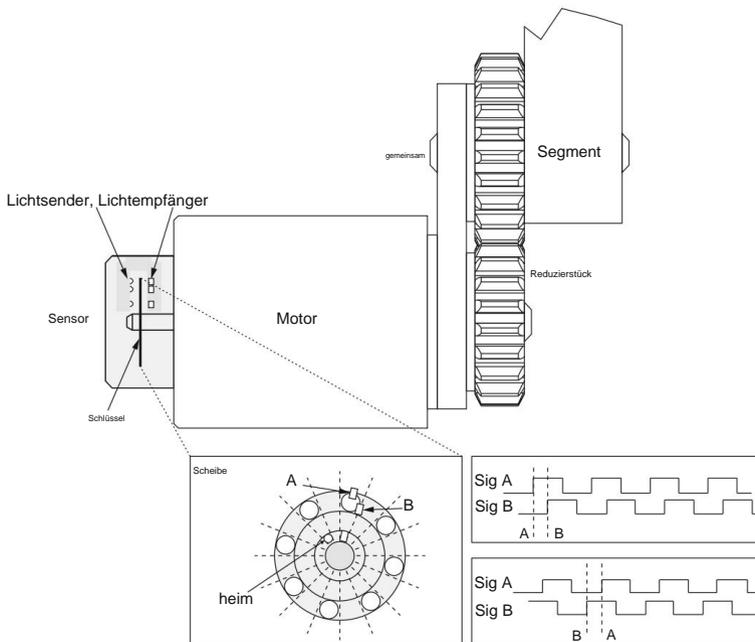


Abb. 7.6 Modell eines Inkrementalgebers. Die Impulsreihe für positive (oben) und negative (unten) Drehrichtung

rotiert, erhält man zwei Signale, die aufgrund der Verschiebung der optischen Paare phasenverschoben sind. Diese Phasenverschiebung entsteht dadurch, dass jeder Schlitz auf der Platte zuerst das erste optische Paar und nach einer kurzen Verzögerung auch das zweite Paar erreicht. Die optischen Komponenten werden üblicherweise so platziert, dass sich zwischen den beiden Signalen eine Phasenverschiebung von $\dot{y}/2$ ergibt. Bei der Drehung im Uhrzeigersinn ist das Signal B gegenüber dem Signal A um $\dot{y}/2$ phasenverzögert. Bei der Drehung gegen den Uhrzeigersinn ist das Signal B gegenüber dem Signal A um $\dot{y}/2$ phasenverzögert (Abb. 7.6). Die Drehrichtung des Encoders kann anhand der Phasenverschiebungen zwischen den Signalen A und B bestimmt werden. Ein weiterer Vorteil von zwei optischen Paaren besteht in der Möglichkeit, alle Änderungen sowohl im A- als auch im B- Signal zu zählen. Der als Quadraturdekodierung bekannte Ansatz ermöglicht eine Messauflösung, die dem Vierfachen der nominellen Encoderauflösung entspricht.

7.2.4 Magnetischer Encoder

Im Gegensatz zu optischen Encodern nutzt der magnetische Encoder ein Magnetfeld zur Positionsmessung. Dies kann durch die Verwendung einer Reihe von Magnetpolen (zwei oder mehr) am Sensorrotor erreicht werden, um die Encoderposition einem Magnetsensor anzuzeigen. Der Rotor c

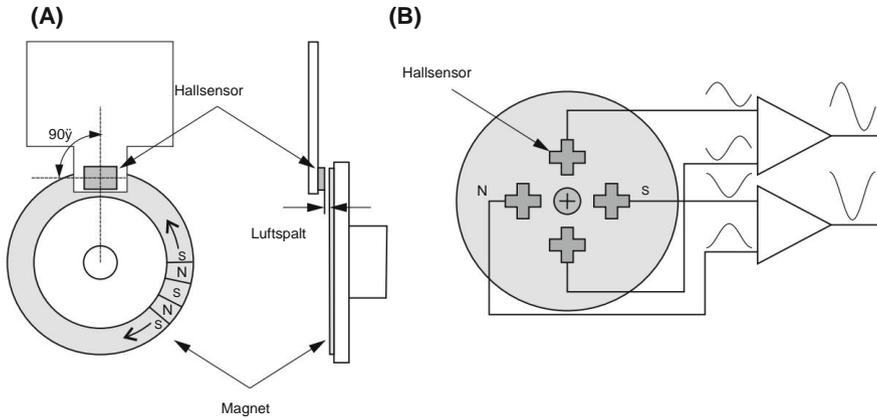


Abb. 7.7 Modell eines magnetischen Encoders: ein Hall-Sensor und abwechselnde Nord- und Südpole und b mehrere Hall-Sensoren mit diametral magnetisiertem rotierenden Magneten

mit dem Schaft und enthält abwechselnd gleichmäßig verteilte Nord- und Südpole um seinen Umfang. Der magnetische Sensor (typischerweise magnetoresistiv oder Hall-Effekt) liest die magnetischen Polpositionen. Hall-Sensoren erzeugen eine Ausgangsspannung proportional zur Stärke eines angelegten Magnetfelds. Magnetoresistive Sensoren erfassen Widerstandsänderungen, die durch ein Magnetfeld verursacht werden. Das Funktionsprinzip ist in Abb. 7.7a dargestellt.

Hall-Sensoren können zur Winkelmessung auch verwendet werden, wenn sie in der Nähe eines diametrisch magnetisierten Magneten platziert werden, der eine sinusförmige Wellenform erzeugt. Die Einschränkung dieser Methode ist die Mehrdeutigkeit bei Winkeln $>90^\circ$ in beide Richtungen vom Nulldurchgangspunkt. Um den Messbereich auf 360° zu erweitern, ist eine Methode erforderlich. Das Problem lässt sich lösen, indem man statt nur einem mehrere Hall-Sensoren verwendet und diese unter einem diametral magnetisierten rotierenden Magneten platziert, um mehrere Sinuswellenformen zu erzeugen. Abbildung 7.7b zeigt vier gleichmäßig beabstandete Hall-Sensoren, die vier Sinussignale erzeugen, die jeweils um 90° gegenüber ihrem Nachbarn phasenverschoben sind. Magnetische Encoder sind typischerweise robuster als optische.

7.2.5 Drehzahlmesser

Das Signal der Gelenkgeschwindigkeit kann durch numerische Differenzierung des Positionssignals gewonnen werden. Dennoch wird in der Robotik häufig die direkte Messung der Gelenkgeschwindigkeit mit Hilfe eines Tachometers eingesetzt. Der Grund ist das durch die numerische Differenzierung verursachte Rauschen, das die Qualität der Robotersteuerung stark beeinträchtigt.

Tachometer können unterteilt werden in: (1) Gleichstrom-Tachometer (DC) und (2) Wechselstrom-Tachometer (AC). In der Robotik kommen in der Regel einfachere Gleichstromtachometer zum Einsatz. Das Funktionsprinzip basiert auf einem Gleichstromgenerator, dessen Magnetfeld durch Permanentmagnete bereitgestellt wird. Da das Magnetfeld konstant ist, ist die Ausgangsspannung des Drehzahlmessers proportional zur Winkelgeschwindigkeit des Rotors. Durch den Einsatz von Kommutatoren in den DC-Tachometern entsteht eine leichte Welligkeit in der Ausgangsspannung, die nicht vollständig herausgefiltert werden kann. Dieser Mangel wird zusammen mit anderen Mängeln durch den Einsatz von Wechselstrom-Tachometern vermieden.

7.2.6 Trägheitsmesseinheit

Potentiometer und optische Encoder messen Gelenkbewegungen in Robotermechanismen. Betrachtet man beispielsweise ein Roboterluftfahrzeug oder einen Radroboter, liefern diese Sensoren keine Informationen über die Ausrichtung des Geräts im Raum.

Die Messung der Orientierung von Objekten (Robotern) im Raum basiert typischerweise auf dem magneto-inertialen Prinzip. Diese Methode kombiniert ein Gyroskop (Winkelgeschwindigkeitssensor), einen Beschleunigungsmesser (linearer Beschleunigungssensor) und ein Magnetometer (misst die Ausrichtung relativ zum Erdmagnetfeld und wird nicht als Trägheitssensor bezeichnet).

Die Methode soll am Beispiel eines starren Pendels veranschaulicht werden, das mit einem zweiachsigen Beschleunigungsmesser (misst Beschleunigungen entlang zweier senkrechter Achsen) und einem einachsigen Gyroskop ausgestattet ist (Abb. 7.8). Beide Sensoren geben die gemessenen Größen in ihren eigenen Koordinatenrahmen an, die in der Mitte des Sensors angebracht sind und deren Achsen parallel zu den x- und y-Achsen des am Pendel befestigten Koordinatenrahmens liegen. Abbildung 7.8a zeigt ein stationäres Pendel, während Abbildung 7.8b ein schwingendes Pendel zeigt. Uns interessiert die Ausrichtung des Pendels relativ zum Referenzkoordinatensystem $x_0-y_0-z_0$. Da das Pendel nur um die z-Achse schwingt, interessiert uns eigentlich nur der Winkel $\dot{\gamma}$.

Wir analysieren zunächst stationäre Bedingungen. Da die Winkelgeschwindigkeit eines stationären Pendels gleich Null ist, ist die Ausgabe des Gyroskops ebenfalls Null und das Gyroskop sagt uns nichts über die Ausrichtung des Pendels. Wir können jedoch sehen, dass der Beschleunigungsmesser immer noch die Erdbeschleunigung misst. Da sich der Beschleunigungsmesser in einem Winkel von $\dot{\gamma}$ relativ zum Gravitationsfeld befindet, werden zwei Beschleunigungskomponenten gemessen. Die Vektoren in beiden Komponenten ergibt die Erdbeschleunigung. Abbildung 7.8a zeigt, dass der Winkel zwischen den Vektoren \mathbf{g} und \mathbf{a}_y gleich $\dot{\gamma}$ ist. Da die Skalarwerte der Beschleunigungen a_x und a_y bekannt sind, können wir nun den Pendelwinkel bestimmen

$$\dot{\gamma} = \arctan \frac{a_x}{a_y} \quad (7.8)$$

Der Beschleunigungsmesser ermöglicht somit die Messung des Pendelwinkels im stationären Zustand. Aus diesem Grund werden Beschleunigungsmesser häufig als Neigungsmesser eingesetzt.

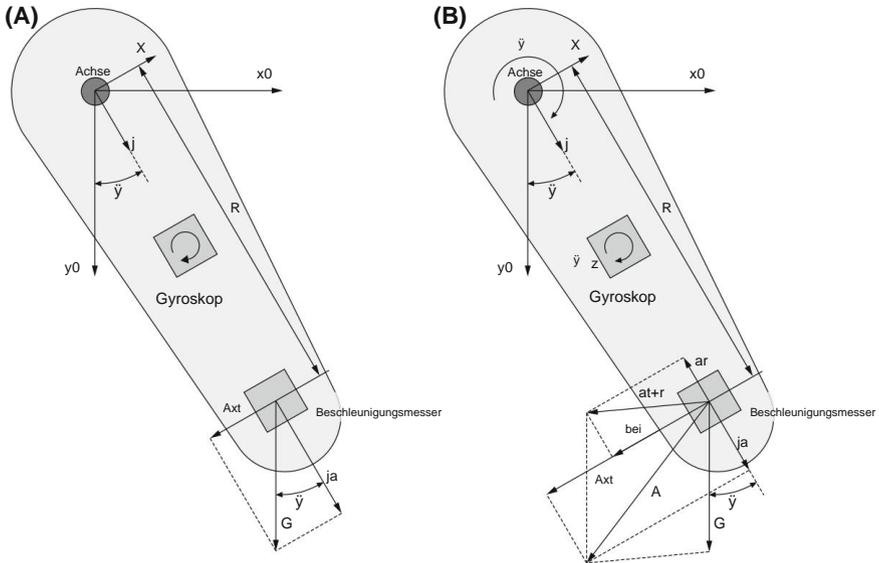


Abb. 7.8 Beispiel für die Verwendung eines Inertialmesssystems zur Messung des Winkels eines Pendels:
linke Abbildung – stationäres Pendel, rechte Abbildung – schwingendes Pendel

Ganz anders sind die Verhältnisse bei einem schwingenden Pendel. Da ist Schwingen
Bei einer beschleunigten Drehbewegung wird der Beschleunigungsmesser nicht nur durch die beeinflusst
Erdbeschleunigung g , aber auch durch Zentripetalbeschleunigung

$$a_r = \ddot{\gamma} \times (\dot{\gamma} \times r) \quad (7.9)$$

und Tangentialbeschleunigung

$$a_t = \ddot{\gamma} \times r. \quad (7.10)$$

Die auf den Beschleunigungsmesser wirkende Gesamtbeschleunigung beträgt somit

$$\mathbf{a} = \mathbf{g} + \mathbf{a}_r + \mathbf{a}_t. \quad (7.11)$$

Die Gleichung zur Berechnung des Winkels unter stationären Bedingungen (7.8) ist nicht mehr gültig.
Daher kann der Beschleunigungsmesser nicht zur Berechnung des Winkels eines schwingenden
Pendulums verwendet werden. Allerdings ist der Ausgang des Gyroskops, der die Winkelgeschwindigkeit misst
des Pendels, ist jetzt auch verfügbar. Da der Winkel des Pendels sein kann
Als zeitliches Integral der Winkelgeschwindigkeit berechnet, kann folgende Beziehung gelten
angegeben

$$\dot{\gamma} = \dot{\gamma}_0 + \dot{\gamma} dt, \quad (7.12)$$

wobei die Anfangsausrichtung des Pendels $\dot{\gamma}_0$ bekannt sein muss.

Das gegebene Beispiel macht deutlich, dass ein Beschleunigungsmesser für Orientierungsmessungen unter statischen oder quasistatischen Bedingungen geeignet ist, während ein Gyroskop für Orientierungsmessungen unter dynamischen Bedingungen geeignet ist. Allerdings müssen zwei Schwächen von Beschleunigungsmessern und Gyroskopen erwähnt werden. Ein Beschleunigungsmesser kann nicht zum Messen von Winkeln in einer horizontalen Ebene verwendet werden, da der Ausgang des Sensors Null ist, wenn seine Achse senkrecht zur Richtung der Schwerkraft ist.

Zu diesem Zweck können wir ein Magnetometer verwenden, das auch die Messung der Rotation um den Schwerkraftfeldvektor ermöglicht (denken Sie an die Funktionsweise eines Kompasses). Darüber hinaus ist weder die Ausgabe des Gyroskops noch des Beschleunigungsmessers ideal. Zusätzlich zur gemessenen Größe enthält die Ausgabe einen Offset und ein Rauschen. Die Integration des Offsets führt zu einer linearen Drift, daher ist Gl. (7.12) liefert keine genaue Messung der Pendelausrichtung. Aufgrund der Schwächen der einzelnen Sensoren ist es üblich, drei senkrechte Beschleunigungsmesser, drei senkrechte Gyroskope und drei senkrechte Magnetometer in einem einzigen System zu kombinieren, das als magneto-inertiale Messeinheit (IMU) bezeichnet wird. Die Kombination der besten Eigenschaften eines Beschleunigungsmessers, eines Gyroskops und eines Magnetometers kann eine genaue und zuverlässige Messung der räumlichen Orientierung ermöglichen.

Die vom Gyroskop gemessene Winkelgeschwindigkeit wird integriert und ergibt eine Schätzung der Orientierung. Messungen vom Beschleunigungsmesser und Magnetometer werden verwendet, um die Sensorausrichtung in Bezug auf die Schwerkraft- und Magnetfeldvektoren direkt zu berechnen. Dies wird durch Sensorfusion erreicht, die durch den Einsatz des Kalman-Filters erfolgen kann.

7.3 Kontaktsensoren

Die bisher betrachteten Sensoren liefern Informationen über Roboterhaltung und -bewegung. Sie ermöglichen das Schließen des Lage- und Geschwindigkeitsregelkreises. Bei einigen Roboteraufgaben ist der Kontakt des Endeffektors mit der Umgebung erforderlich. Typische Kontaktsensoren, die in der Robotik eingesetzt werden, sind taktile Sensoren sowie Kraft- und Drehmomentsensoren. Taktile Sensoren messen Parameter, die den Kontakt zwischen Sensor und Objekt definieren.

Die Erfassung besteht in der Messung einer punktuellen Kontaktkraft und der räumlichen Verteilung von Kräften senkrecht zu einer Fläche. Im Gegensatz dazu messen Kraft- und Drehmomentsensoren die Gesamtkräfte, die auf ein Objekt wirken.

7.3.1 Taktile Sensor

Roboter können auch durch Berührung Informationen über die Umgebung sammeln. Um die Manipulationsfähigkeiten von Robotern zu erhöhen, können taktile Sensoren in Roboterfingern eingesetzt werden, wie in Abb. 7.9a dargestellt. Der Sensor liefert Daten über die Kontaktkraftverteilung zwischen Finger und manipuliertem Objekt. Um die Robotersicherheit zu erhöhen (z. B. wenn

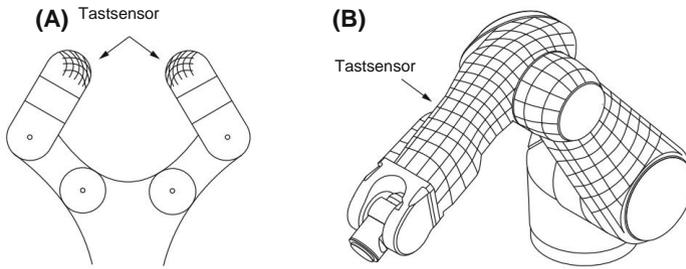


Abb. 7.9 Taktile Sensor im Roboterfinger (links) und als Roboterhaut (rechts)

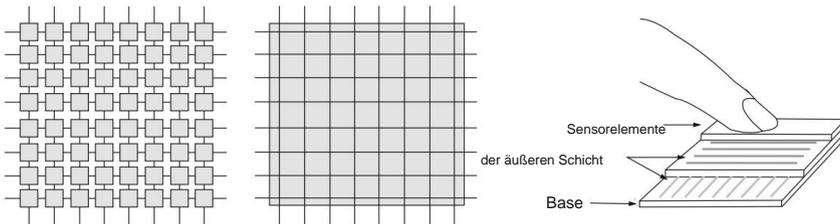


Abb. 7.10 Ein Modell eines taktilen Sensors

Beim Arbeiten mit einem Menschen können taktile Sensoren als künstliche Roboterhaut eingesetzt werden, die es dem Roboter ermöglicht, Kontakte mit Objekten in der Umgebung zu erfassen (Abb. 7.9b).

Die taktile Wahrnehmung basiert auf einer Reihe von Berührungssensoren, wie in Abb. 7.10 dargestellt. Der Folgende Sensorprinzipien können im Array implementiert werden:

- **verformungsbasierte Sensoren** – die Materialoberfläche verformt sich (ändert ihre Länge), wenn sie äußeren Kräften ausgesetzt wird; Die Verformung wird mit Dehnungsmessstreifen, die in einer Wheatstone-Brücke verbunden sind, in elektrische Signale umgewandelt.
- **Widerstandssensoren** – elektrischer Widerstand ändert sich mit dem Druck eines Materials, das zwischen zwei Elektroden platziert wird;
- **kapazitive Sensoren** – das Sensorelement ist ein Kondensator, dessen Kapazität sich mit der ausgeübten Kraft ändert; Kraft kann entweder eine Änderung des Abstands zwischen den Kondensatorplatten oder ihrer Fläche bewirken;
- **optische Sensoren** – die Erfassung basiert typischerweise auf der Messung der Lichtintensität; Die Lichtintensität kann durch Bewegungen eines Hindernisses oder einer reflektierenden Oberfläche in den Lichtweg moduliert werden. die Intensität des empfangenen Lichts ist eine Funktion der Verschiebung und damit der ausgeübten Kraft;
- **piezoelektrische Sensoren** – Materialien wie Quarz haben piezoelektrische Eigenschaften und können daher zur taktilen Erfassung verwendet werden; piezoelektrische Wandler sind für die statische Kraftübertragung nicht geeignet; Dieses Problem kann gelöst werden, indem der Sensor vibriert und der Unterschied in der Vibrationsfrequenz aufgrund der ausgeübten Kraft erfasst wird.

- **Magnetsensoren** – Änderungen der magnetischen Flussdichte oder der magnetischen Kopplung zwischen Schaltkreisen sind die am häufigsten verwendeten Prinzipien bei der magnetischen taktilen Erfassung;
- **mechanische Sensoren** – Sensorelemente sind mechanische Mikroschalter mit Einschaltfunktion und Aus-Zustände.

7.3.2 Endschalter und Stoßstange

Endschalter werden häufig zur Steuerung von Robotermechanismen verwendet. Sie können eine einzelne Position eines beweglichen Teils erfassen und sind daher geeignet, sicherzustellen, dass die Bewegung einen vordefinierten Grenzwert nicht überschreitet. Ein Stoßfängersensor, ein spezieller Endschalter, teilt dem Roboter beispielsweise mit, ob er Kontakt mit einem physischen Objekt hat oder nicht. Wenn der Sensor an der vorderen Stoßstange eines mobilen Roboters montiert wird, könnte der Roboter anhand dieser Informationen erkennen, ob er auf ein Hindernis, beispielsweise eine Wand, gestoßen ist (Abb. 7.11). Roboterstaubsauger verlassen sich in der Regel auf Stoßfängersensoren, um sich in der häuslichen Umgebung zurechtzufinden.

7.3.3 Kraft- und Drehmomentsensor

Im einfachsten Fall ermöglicht die Kraftmessung eine Abschaltung des Roboters, wenn die Kontaktkraft einen vorgegebenen Sicherheitsgrenzwert überschreitet. In einem anspruchsvolleren Fall verwenden wir Kraftsensoren zur Steuerung der Kraft zwischen dem Endeffektor des Roboters und der Umgebung. Es ist daher nicht schwer zu erkennen, dass der Kraftsensor im Handgelenk des Roboters platziert ist und daher oft als Handgelenksensor bezeichnet wird.

Zur Kraftmessung werden üblicherweise Dehnungsmessstreifen verwendet. Der Dehnungsmessstreifen ist an einem elastischen Balken befestigt, der sich unter der durch die ausgeübte Kraft verursachten Spannung verformt. Der Dehnungsmessstreifen verhält sich daher wie ein variabler Widerstand, dessen Widerstand sich proportional zu seiner Verformung ändert. Der Handgelenksensor darf das nicht beeinflussen

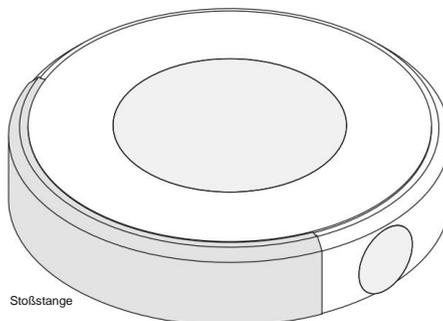


Abb. 7.11 Stoßfängersensoren für den Einsatz an einem mobilen Roboter

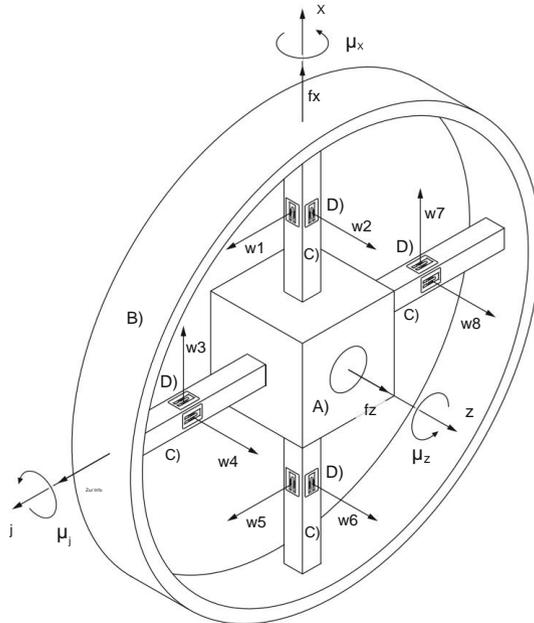


Abb. 7.12 Modell des Kraft- und Drehmomentsensors: ein starrer Körper, der mit dem Endeffektor des Roboters in Kontakt steht, b starrer Ring, der mit der Roboterumgebung in Kontakt steht, c elastische Balken und d Dehnungsmessstreifen

Interaktion des Roboters mit der Umgebung. Das bedeutet, dass der Handgelenksensor ausreichend steif sein muss. Die Roboter-Handgelenksensoren sind üblicherweise wie in Abb. 7.12 dargestellt. Die Struktur des Sensors basiert auf drei Komponenten: (a) starrem Innenteil, das mit dem Endeffektor des Roboters in Kontakt steht; (b) starrer Außenring, der mit der Roboterumgebung in Kontakt steht; und (c) elastische Balken, die den Außen- und den Innenring verbinden. Beim Kontakt des Roboters mit der Umgebung werden die Balken durch die äußeren Kräfte verformt, was zu einer Widerstandsänderung der Dehnungsmessstreifen führt.

Der Vektor der am Roboter-Endeffektor wirkenden Kräfte und Drehmomente liegt im dreidimensionalen Raum, der durch sechs Elemente, drei Kräfte und drei Drehmomente, dargestellt wird.

Der rechteckige Querschnitt eines Balkens (dargestellt in Abb. 7.12) ermöglicht die Messung von Verformungen in zwei Richtungen. Um die sechs Elemente des Kraft- und Drehmomentvektors zu messen, sind mindestens drei Balken erforderlich, die nicht kollinear sind. Im Beispiel in Abb. 7.12 werden vier Balken verwendet. An den senkrechten Flächen jedes Balkens sind zwei Dehnungsmessstreifen angebracht. Bei acht Dehnungsmessstreifen gibt es acht variable Widerstände R_1 bis R_8 . Als Folge der äußeren Kräfte und Momente treten elastische Verformungen w_1 bis w_8 auf, die zu Änderungen der Widerstände \dot{y}_{R1} bis \dot{y}_{R8} führen. Die kleinen Widerstandsänderungen werden mithilfe der Weizensteinbrücke in Spannungssignale umgewandelt (Abb. 7.13). Zu jeder der acht Variablen

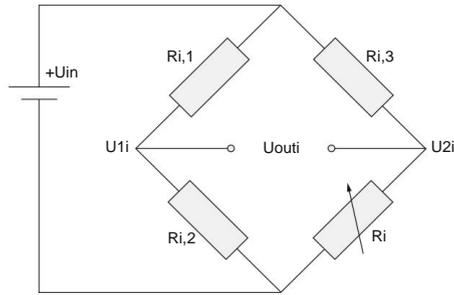


Abb. 7.13 Die Wheatstone-Brücke

Widerstände $\{R_1 \dots R_8\}$ werden drei zusätzliche Widerstände hinzugefügt. Die drei Widerstände werden zusammen mit dem Dehnungsmessstreifen in die Messbrücke geschaltet. Die Brücke wird mit der Spannung U_{in} versorgt, während die Ausgangsspannung U_{outi} durch die Differenz $U_{1i} - U_{2i}$ bestimmt wird.

$$U_{1i} = \frac{R_{i,2}}{R_{i,1} + R_{i,2}} U_{in}, \quad (7.13)$$

während die U_{2i} - Spannung ist

$$U_{2i} = U_{in} \frac{R_i}{R_i + R_{i,3}} \quad (7.14)$$

Die Ausgangsspannung ist gleich

$$U_{outi} = \frac{R_{i,2}}{R_{i,1} + R_{i,2}} - \frac{R_i}{R_i + R_{i,3}} U_{in}. \quad (7.15)$$

Durch Differenzierung der Gl. (7.15) In Bezug auf die Variable R_i kann der Einfluss der Änderung des DMS-Widerstands auf die Ausgangsspannung ermittelt werden

$$\dot{y}U_{outi} = \dot{y} \frac{R_{i,3}U_{in}}{(R_i + R_{i,3})^2} \dot{y}R_i. \quad (7.16)$$

Vor der Anwendung muss der Kraftsensor kalibriert werden, was die Ermittlung einer 6×8 -Kalibrierungsmatrix \mathbf{T} erfordert, die die sechs Ausgangsspannungen in die drei Kräfte f_x, f_y, f_z und drei Drehmomente μ_x, μ_y, μ_z

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \mathbf{K} \begin{bmatrix} U_{out1} \\ U_{out2} \\ U_{out3} \\ U_{out4} \\ U_{out5} \\ U_{out6} \\ U_{out7} \\ U_{out8} \end{bmatrix}^T, \quad (7.17)$$

Wo

$$K = \begin{pmatrix} 0 & 0 & K_{13} & 0 & 0 & 0 & K_{17} & 0 \\ \ddot{y} & K_{21} & 0 & 0 & 0 & 0 & K_{25} & 0 \\ 0 & K_{32} & 0 & K_{34} & 0 & K_{36} & 0 & K_{38} \\ 0 & 0 & 0 & K_{44} & 0 & 0 & 0 & K_{48} \\ 0 & K_{52} & 0 & 0 & 0 & K_{56} & 0 & 0 \\ \ddot{y} & K_{61} & 0 & K_{63} & 0 & K_{65} & 0 & K_{67} & 0 \end{pmatrix} \ddot{y} \quad (7.18)$$

ist die Kalibrierungsmatrix mit konstanten Werten K_{ij} .

7.3.4 Gelenkdrehmomentsensor

Oft ist es erforderlich oder vorzuziehen, Gelenkdrehmomente anstelle der Endeffektorkräfte des Roboters zu messen. In solchen Fällen muss ein Gelenkdrehmomentsensor verwendet werden. Durch die Messung der Gelenkdrehmomente kann der Roboter auf Kräfte reagieren, die an einer beliebigen Stelle seines Mechanismus wirken. Wenn das dynamische Modell des Roboters bekannt ist, ist es auch möglich, Endeffektorkräfte abzuschätzen.

Betrachten Sie als Beispiel Gl. (5.20). Die Umkehrung dieser Gleichung würde ergeben

$$f = J^{-T} (q) \ddot{y}. \quad (7.19)$$

Es ist zu beachten, dass die obige Gleichung nur unter statischen Bedingungen genaue Endeffektorkräfte liefern würde und wenn die Schwerkraft die Gelenkdrehmomente nicht beeinflusst.

Unter anderen Bedingungen muss das dynamische Robotermodell (5.56) berücksichtigt werden.

Das Funktionsprinzip des Drehmomentsensors ähnelt typischerweise dem des Handgelenksensors. Allerdings ist sein mechanischer Aufbau auf die Gelenkachse abgestimmt.

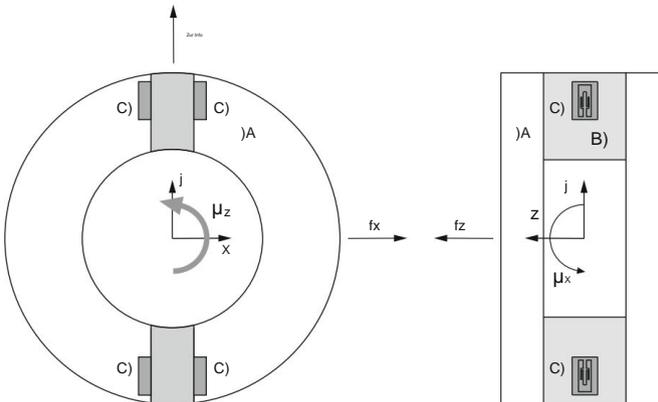


Abb. 7.14 Struktur des Gelenkdrehmomentsensors misst das Drehmoment μ_z : a Rahmen, b elastisches Element und c Dehnungsmessstreifen

Somit wird der Sensor zwischen Aktuator (und ggf. Getriebe) und Robotersegment integriert. Der Sensor muss eine hohe Empfindlichkeit gegenüber Torsion, eine geringe Empfindlichkeit gegenüber nicht-torsionellen Komponenten und eine hohe Steifigkeit in allen Kraft- und Momentenachsen gewährleisten. Die Verformung der mechanischen Struktur aufgrund des Gelenkdrehmoments wird mithilfe von Dehnungsmessstreifen gemessen. Eine schematische Darstellung des Gelenkdrehmomentsensors ist in [Abb. 7.14](#) dargestellt.

7.4 Näherungs- und Entfernungssensoren

Näherungs- und Entfernungssensoren erkennen die Anwesenheit von Objekten in der Nähe ohne jeglichen physischen Kontakt. Konsequenterweise ermöglichen sie die Unterscheidung von Hindernissen unterschiedlicher Form und Größe sowie eine effizientere Hindernisvermeidung als Kontaktsensoren. Um Hindernisse aus der Ferne zu erkennen, können unterschiedliche Methoden eingesetzt werden. Methoden, die auf magnetischen und kapazitiven Prinzipien basieren, ermöglichen typischerweise die Erkennung der Nähe eines Objekts, nicht jedoch seiner Entfernung. Wenn die Entfernung relevant ist, können aktive Methoden wie Ultraschall-Entfernungsmesser, Laser-Entfernungsmesser und Infrarot-Näherungssensor sowie passive Methoden auf Kamerabasis in Betracht gezogen werden. Alle Methoden zeichnen sich durch hohe Zuverlässigkeit und lange Lebensdauer aus, da sie ohne physischen Kontakt zwischen Sensor und erfasstem Objekt funktionieren.

7.4.1 Ultraschall-Entfernungsmesser

Ein Ultraschall-Entfernungsmesser misst mithilfe von Schallwellen die Entfernung zu einem Objekt. Die Entfernung wird gemessen, indem eine Schallwelle mit Ultraschallfrequenz ausgesendet wird (höhere Frequenzen eignen sich besser für kurze Entfernungen und hohe Präzisionsanforderungen) und darauf geachtet wird, ob diese Schallwelle zurückprallt ([Abb. 7.15a](#)). Die verstrichene Zeit zwischen der Erzeugung der Schallwelle und dem Zurückprallen der Schallwelle wird verwendet, um den Abstand zwischen dem Sensor und dem Objekt zu berechnen (unter Berücksichtigung der Tatsache, dass die Schallgeschwindigkeit in der Luft etwa 343 m/s beträgt).

Für eine erfolgreiche Objekterkennung und -vermeidung ist das Verständnis des Erkennungsbereichs wichtig. Die Strahlbreite eines Ultraschall-Entfernungsmessers wird typischerweise als Kegel mit einem bestimmten Winkel beschrieben. Dieser Winkel beschreibt den Bogen, in dem die Ultraschallwelle vom Wandler ausgeht. Ab einer bestimmten Entfernung beginnt die Ausdehnungsgeschwindigkeit jedoch abzuklingen, wie in [Abb. 7.15b](#) dargestellt. Eine Erweiterung des Messbereichs eines Ultraschall-Entfernungsmessers kann durch den Einsatz mehrerer Sensoreinheiten in unterschiedlichen Winkeln erreicht werden ([Abb. 7.15c](#)). In einem solchen Fall muss das Problem des Übersprechens berücksichtigt werden.

Verschiedene andere Faktoren beeinflussen die Leistung eines Ultraschall-Entfernungsmessers. Die Größe, Zusammensetzung, Form und Ausrichtung von Objekten müssen berücksichtigt werden. In den oberen Bildern in [Abb. 7.16](#) dargestellten Fällen sind die Messungen normalerweise korrekt, während in den unteren Bildern in [Abb. 7.16](#) dargestellten Fällen der Ultraschall-Entfernungsmesser falsche Ergebnisse liefern würde.

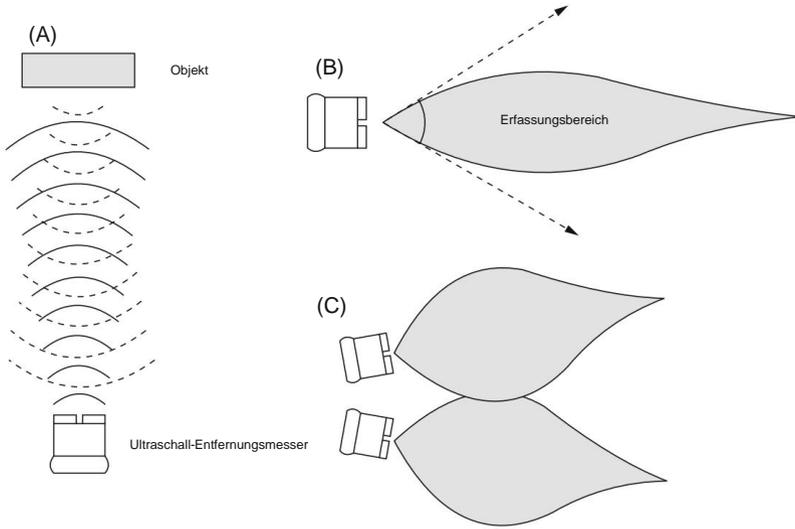


Abb. 7.15 Ultraschall-Entfernungsmesser: a Funktionsprinzip, b Erfassungsbereich und c Kombination mehrerer Sensoren

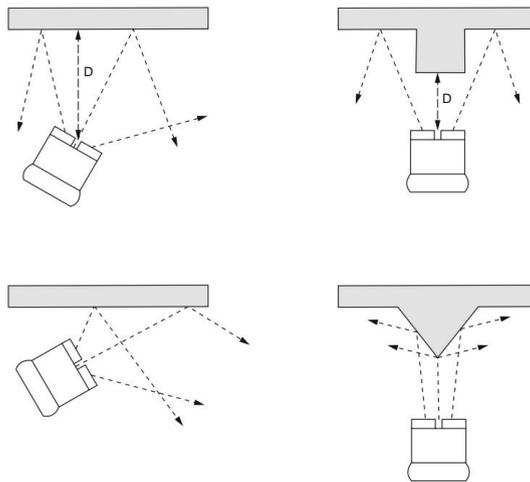


Abb. 7.16 Entfernungsmessung mit Ultraschall-Entfernungsmessern und Einschränkungen: korrekte Messungen der Entfernung d (obere Reihe) und falsche Ergebnisse (untere Reihe)

7.4.2 Laser-Entfernungsmesser und Laserscanner

Ein Laser-Entfernungsmesser verwendet einen Laserstrahl, um die Entfernung zu einem Objekt zu bestimmen. Die gebräuchlichste Form von Laser-Entfernungsmessern arbeitet nach dem Flugzeitprinzip.

Die Entfernung kann bestimmt werden, indem die Zeit gemessen wird, die der Laserpuls benötigt, um zum Ziel zurückzu-

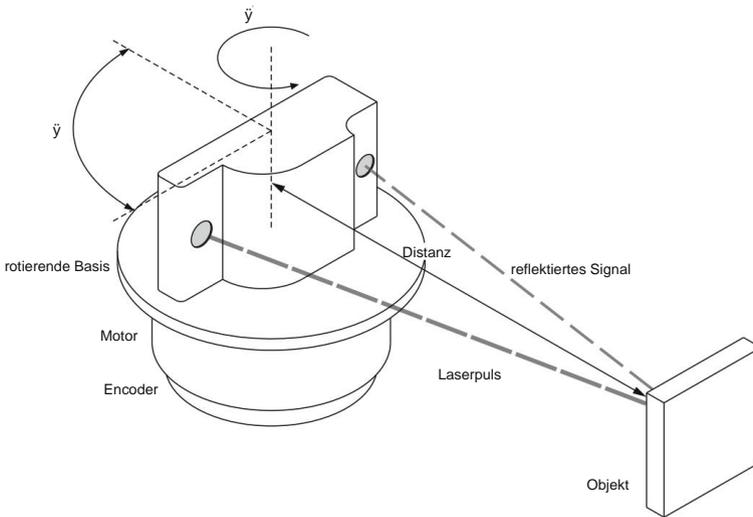


Abb. 7.17 Laserscanner

Sensor und erfordert eine präzise Zeitmessung. Wenn die Lichtgeschwindigkeit bekannt ist und die benötigte Zeit genau gemessen wird, kann die Entfernung berechnet werden. Eine andere Möglichkeit besteht darin, die Phasenverschiebung von Lichtwellen zu berechnen, indem das einfallende Licht analysiert und mit einem Referenzsignal verglichen wird. Die genaueste Methode zur Messung von Abstandsänderungen statt absoluter Abstände ist die Interferometrie.

Der Laser-Entfernungsmesser misst die Entfernung zu jeweils einem Objekt. Daher handelt es sich um einen eindimensionalen Sensor. Der Laserscanner verwendet einen Laser, der über das Sichtfeld des Sensors streicht. Wie der Name schon sagt, besteht das Instrument im Wesentlichen aus einem Laser und einem Scanner. Entfernungen werden wie mit dem Laser-Entfernungsmesser gemessen. Der Laserscanner erzeugt eine Reihe von Punkten, indem er die Umgebung mit hoher Geschwindigkeit abtastet. Dies wird typischerweise durch den Einsatz rotierender Baugruppen oder rotierender Spiegel erreicht, um die Umgebung um 360 Grad zu umrunden. Das Funktionsprinzip des Laserscanners ist in Abb. 7.17 dargestellt.

Die abgetasteten Punkte stellen Objektpositionen relativ zum Sensor dar. Die Erzeugung einer Punktmatrix ist in Abb. 7.18 dargestellt. Der Abstand d_L wird mit dem Laser gemessen und der Drehwinkel γ_L wird typischerweise mit einem Encoder an der rotierenden Baugruppe gemessen. Punkte werden daher in Polarkoordinaten definiert. Sie können mit in kartesische Koordinaten (x_L , y_L) relativ zum Sensor transformiert werden

$$x_L = d_L \cos \gamma_L \text{ und } y_L = d_L \sin \gamma_L . \quad (7.20)$$

Die abgetasteten Punkte können zur Erstellung einer Karte der Umgebung, zur Wegplanung und zur Vermeidung von Hindernissen verwendet werden. Ein dreidimensionaler (3-D) Scanner ermöglicht das Scannen von

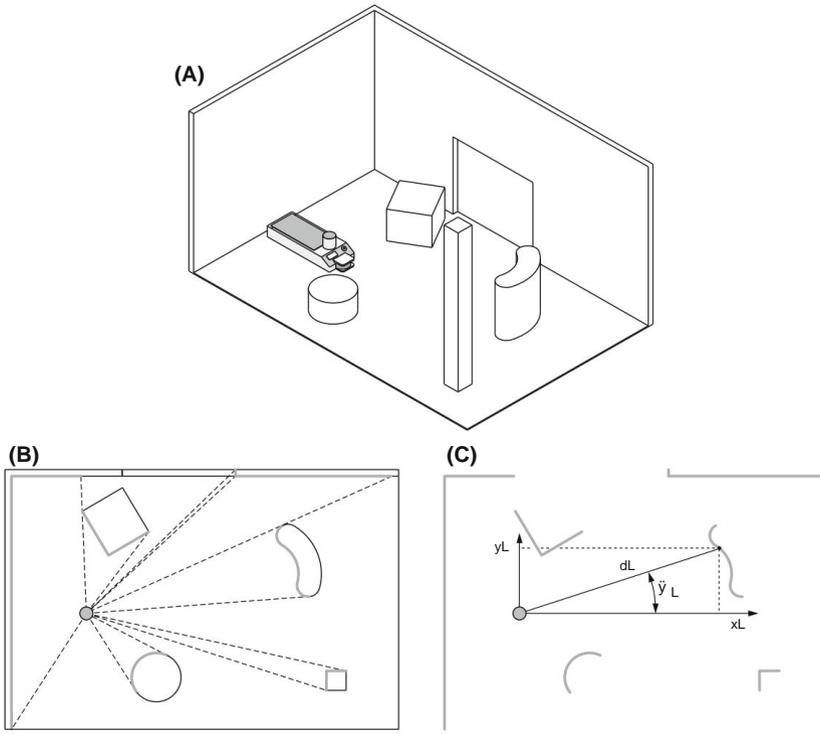


Abb. 7.18 Laserscanner zur Erstellung einer Umgebungskarte: a-Umgebung, b-Scanning und c-Karte

eines vollständigen 3D-Raums und Sammeln von 3D-Punktwolken Daten durch Nutzung eines weiteren Freiheitsgrads an der rotierenden Baugruppe. Diese 3D-Laserscanner werden üblicherweise als LiDAR (Light Detection And Ranging) bezeichnet und werden häufig in autonomen Fahrzeugen zum Scannen der Umgebung eingesetzt.

Kapitel 8

Robotervision



Die Aufgabe von Robot Vision besteht darin, die Geometrie des Roboterarbeitsraums anhand eines digitalen Bildes zu erkennen. Unser Ziel ist es, die Beziehung zwischen den Koordinaten eines Punktes im zweidimensionalen (2D) Bild und den Koordinaten des Punktes in der realen dreidimensionalen (3D) Roboterumgebung zu finden.

8.1 Systemkonfiguration

Das Roboter-Vision-System basiert auf der Verwendung von einer, zwei oder mehreren Kameras. Werden mehrere Kameras zur Beobachtung desselben Objekts eingesetzt, können daraus Informationen über die Tiefe des Objekts abgeleitet werden. In diesem Fall sprechen wir von 3D- oder Stereovision. Natürlich kann die 3D-Ansicht auch mit einer einzigen Kamera erreicht werden, wenn zwei Bilder des Objekts vorliegen, die aus unterschiedlichen Posen aufgenommen wurden. Wenn nur ein Bild verfügbar ist, kann die Tiefe anhand einiger zuvor bekannter geometrischer Eigenschaften des

Bei der Analyse der Konfiguration des Roboter-Vision-Systems ist es notwendig, zwischen möglichen Platzierungen der Kameras zu unterscheiden. Die Kameras können in einer festen Konfiguration, bei der sie starr in der Arbeitszelle montiert sind, oder in einer mobilen Konfiguration, bei der die Kamera an einem Roboter befestigt ist, platziert werden. In der ersten Konfiguration beobachtet die Kamera Objekte von einer festen Position in Bezug auf den Basiskoordinatenrahmen des Roboters. Das Sichtfeld der Kamera ändert sich während der Ausführung der Aufgabe nicht, sodass die Genauigkeit der Messung grundsätzlich konstant bleibt. Bei manchen Aufgaben lässt sich nur schwer verhindern, dass der Manipulator in das Sichtfeld der Kamera greift und dadurch die Objekte verdeckt. Daher ist es in einem solchen Fall notwendig, eine Kamera an einem Roboter anzubringen (in einer mobilen Konfiguration).

Die Kamera kann vor oder hinter dem Roboterhandgelenk angebracht werden. Im ersten Fall beobachtet die Kamera die Situation aus einer günstigen Position und der Manipulator verdeckt ihr Sichtfeld im Allgemeinen nicht. Im zweiten Fall ist die Kamera am Endeffektor des Roboters angebracht und beobachtet typischerweise nur das zu manipulierende Objekt.

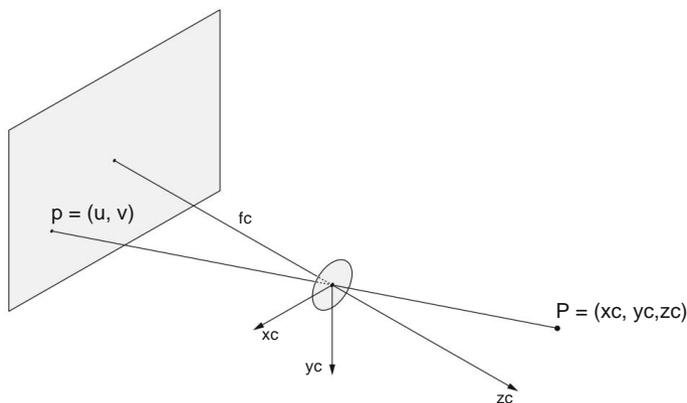


Abb. 8.1 Perspektivische Projektion

uliert. In beiden Fällen ändert sich das Sichtfeld der Kamera mit den Bewegungen des Manipulators. Wenn sich der Manipulator dem Objekt nähert, erhöht sich typischerweise die Genauigkeit der Messung.

8.2 Vorwärtsprojektion

Die Grundgleichungen der Optik bestimmen die Position eines Punktes in der Bildebene relativ zum entsprechenden Punkt im 3D-Raum (Abb. 8.1). Wir werden daher die geometrische Beziehung zwischen den Koordinaten des Punktes $P = (x_c, y_c, z_c)$ im Raum und den Koordinaten des Punktes $p = (u, v)$ im Bild finden.

Da die Blendenöffnung der Kameralinsen, durch die das Licht auf die Bildebene fällt, im Vergleich zur Größe der vom Roboter manipulierten Objekte klein ist, können wir die Linsen in unserem mathematischen Modell durch eine einfache Lochblende ersetzen. Bei der perspektivischen Projektion werden Punkte aus dem Raum durch Linien, die sich in einem gemeinsamen Punkt, dem Projektionszentrum, schneiden, auf die Bildebene projiziert. Beim Austausch einer echten Kamera durch eine Lochkamera liegt das Projektionszentrum

Bei der Untersuchung der Robotergeometrie und -kinematik haben wir an jedem starren Körper (z. B. an Robotersegmenten oder an vom Roboter manipulierten Objekten) einen Koordinatenrahmen angebracht. Bei der Robotersicht stellt die Kamera selbst einen starren Körper dar und ihr sollte ein Koordinatenrahmen zugeordnet werden. Die Pose der Kamera wird nun durch ein entsprechendes Koordinatensystem beschrieben. Die z_c - Achse des Kamerarahmens ist entlang der optischen Achse ausgerichtet, während der Ursprung des Rahmens in der Projektionsmitte liegt. Wir wählen einen rechtshändigen Rahmen, bei dem die x_c - Achse parallel zu den Zeilen des Bildsensors u

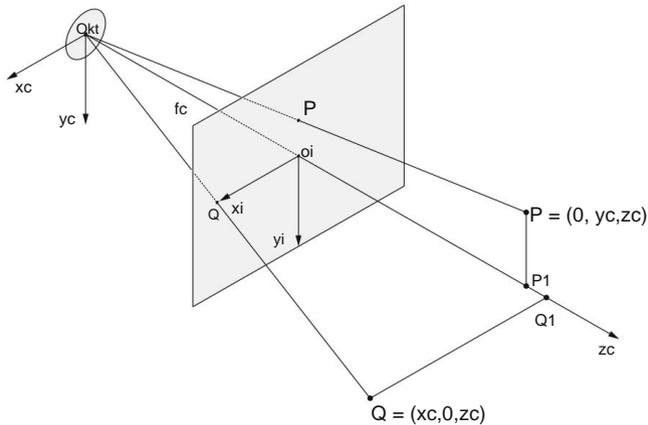


Abb. 8.2 Äquivalente Bildebene

Die Bildebene liegt in der Kamera, die hinter dem Projektionszentrum platziert ist. Der Abstand f_c zwischen dem Bild und dem Projektionszentrum wird Brennweite genannt. Im Kamerabild hat die Brennweite einen negativen Wert, da die Bildebene die negative z_c -Achse schneidet. Es ist bequemer, die äquivalente Bildebene zu verwenden, die auf einem positiven z_c -Wert platziert ist (Abb. 8.2). Die äquivalente Bildebene und die reale Bildebene sind symmetrisch in Bezug auf den Ursprung des Kamerarahmens. Die geometrischen Eigenschaften der Objekte sind in beiden Ebenen gleich und unterscheiden sich nur im Vorzeichen.

Von nun an nennen wir die äquivalente Bildebene einfach Bildebene. Auch die Bildebene kann als starrer Körper betrachtet werden, an dem ein Koordinatenrahmen befestigt werden sollte. Der Ursprung dieses Rahmens liegt im Schnittpunkt der optischen Achse mit der Bildebene. Die x_i - und y_i -Achsen verlaufen parallel zu den x_c - und y_c -Achsen des Kamerarahmens.

Auf diese Weise verfügt die Kamera über zwei Koordinatenrahmen, den Kamerarahmen und den Bildrahmen. Der Punkt P sei im Kamerabild ausgedrückt, während der Punkt p seine Projektion auf die Bildebene darstellt. Unser Ziel ist es, die Beziehungen zwischen den Koordinaten des Punktes P und den Koordinaten seines Bildes p zu finden.

Nehmen wir zunächst an, dass der Punkt P in der y_c - z_c -Ebene der Kamera liegt. Seine Koordinaten sind

$$P = \begin{pmatrix} 0 \\ \tilde{y}_{yc} \\ \tilde{y}_{zc} \end{pmatrix} \quad (8.1)$$

Der projizierte Punkt p liegt in diesem Fall in der y_i -Achse der Bildebene

$$p = \begin{pmatrix} 0 \\ y_i \end{pmatrix} \quad (8.2)$$

Aufgrund der Ähnlichkeit der Dreiecke $P P1O_c$ und $p oO_c$ können wir schreiben

$$\frac{y_c}{y_i} = \frac{z_c}{f_c}$$

oder

$$y_c y_i = \frac{z_c}{f_c} \cdot y_i \quad (8.3)$$

Betrachten wir auch den Punkt Q , der in der x_c - z_c -Ebene des Kamerarahmens liegt. Nach der perspektivischen Projektion des Punktes Q fällt sein Bild q auf die x_i -Achse des Bildrahmens. Wegen der ähnlichen Dreiecke $Q Q1O_c$ und $q oO_c$ gilt

$$\frac{x_c}{x_i} = \frac{z_c}{f_c}$$

oder

$$x_i = f_c z_c \frac{x_c}{z_c} \quad (8.4)$$

Auf diese Weise haben wir die Beziehung zwischen den Koordinaten (x_c, y_c, z_c) des Punktes P im Kamerabild und den Koordinaten (x_i, y_i) des Punktes p in der Bildebene erhalten. Die Gleichungen (8.3) und (8.4) stellen die mathematische Beschreibung der perspektivischen Projektion von einem 3D- auf einen 2D-Raum dar. Beide Gleichungen können in Form einer perspektivischen Matrixgleichung geschrieben werden

$$s \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} f_c & 0 & 0 \\ 0 & f_c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \quad (8.5)$$

In Gl. (8.5) s ist ein Skalierungsfaktor, während (x_i, y_i) die Koordinaten des projizierten Punktes im Bildrahmen und (x_c, y_c, z_c) die Koordinaten des ursprünglichen Punktes im Kamerarahmen sind.

Aus der Matrix Gl. (8.5) Es ist nicht schwer zu erkennen, dass wir die Koordinaten (x_i, y_i) und den Skalierungsfaktor s eindeutig bestimmen können, wenn wir (x_c, y_c, z_c) kennen. Im Gegenteil, wir können die Koordinaten (x_c, y_c, z_c) im Kamerarahmen nicht berechnen, wenn nur die Koordinaten (x_i, y_i) im Bildrahmen bekannt sind, nicht jedoch der Skalierungsfaktor. Gleichung (8.5) stellt die Vorwärtsprojektion in der Robotersicht dar. Die Berechnung von (x_c, y_c, z_c) aus (x_i, y_i) wird als Rückwärtsprojektion bezeichnet. Bei Verwendung einer einzelnen Kamera und ohne a priori Informationen über die Größe der Objekte in der Roboterumgebung kann keine eindeutige Lösung des inversen Problems gefunden werden.

Zur Vereinfachung der Programmierung ist es bequemer, Indizes zu verwenden, die die Position eines Pixels (dh des kleinsten Elements eines digitalen Bilds) in einem 2D-Bild markieren, anstelle von metrischen Einheiten entlang der x_i - und y_i -Achsen des Bildrahmens. Wir werden zwei Indizes verwenden, die wir Indexkoordinaten eines Pixels nennen (Abb. 8.3).

Dies sind der Zeilenindex und der Spaltenindex. Im Speicher, in dem das digitale Bild gespeichert ist, läuft der

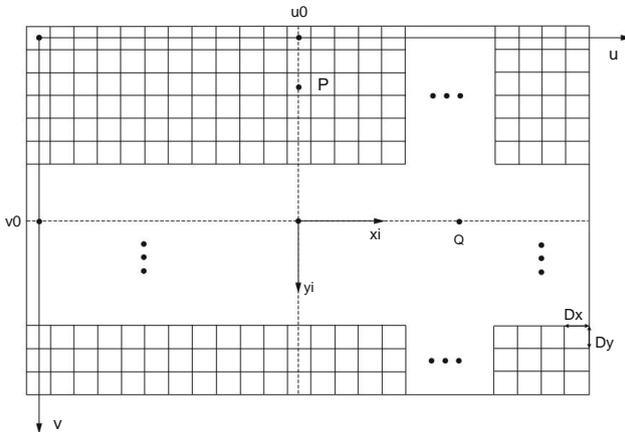


Abb. 8.3 Die Bildebene und der Indexkoordinatenrahmen

Der Spaltenindex beginnt links und endet am rechten Bildrand. Wir werden die u - Achse für die Spaltenindizes und die v - Achse für die Zeilenindizes verwenden. Auf diese Weise gehört der Indexkoordinatenrahmen $u-v$ zu jedem einzelnen Bild. Das obere linke Pixel wird entweder mit $(0, 0)$ oder $(1, 1)$ bezeichnet. Die Indexkoordinaten haben keine Maßeinheiten.

Im weiteren Text werden wir den Zusammenhang zwischen den Bildkoordinaten (x_i, y_i) und den Indexkoordinaten (u, v) finden. Nehmen wir an, dass das digitale Bild als direkte Ausgabe vom Bildsensor erhalten wurde (die A/D-Wandlung wurde am Ausgang des Bildsensors durchgeführt). In diesem Fall entspricht jedes Pixel einem bestimmten Element des Bildsensors. Wir gehen davon aus, dass die Fläche des Bildsensors rechteckig ist.

Der Ursprung des Bildrahmens liegt im Punkt (u_0, v_0) des Indexrahmens. Die Größe eines Pixels wird durch das Paar (D_x, D_y) dargestellt. Die Beziehung zwischen dem Bildrahmen x_i-y_i und dem Indexrahmen $u-v$ wird durch die folgenden zwei Gleichungen beschrieben

$$\begin{aligned} \frac{x_i}{D_x} &= u - u_0 \\ \frac{y_i}{D_y} &= v - v_0. \end{aligned} \quad (8.6)$$

Gleichungen (8.6) können umgeschrieben werden als

$$\begin{aligned} u &= u_0 + \frac{x_i}{D_x} \\ v &= v_0 + \frac{y_i}{D_y}. \end{aligned} \quad (8.7)$$

x_i In Gl. (8.7) stellen die Anzahl der digitalen Konvertierungen entlang der Zeile bzw. -Spalte dar. Gleichung (8.7) kann in die folgende Matrixform umgeschrieben werden

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad (8.8)$$

Mithilfe des Lochkameramodells können wir nun Gl. kombinieren. (8.5), wobei die Bildkoordinaten mit den Kamerakoordinaten in Beziehung gesetzt werden, und Gl. (8.8) beschreibt die Beziehung zwischen den Bild- und Indexkoordinaten

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} \frac{f_c}{D_x} & 0 & u_0 & 0 \\ 0 & \frac{f_c}{D_y} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \quad (8.9)$$

Die obige Matrix kann auch in der folgenden Form geschrieben werden

$$P = \begin{pmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \quad (8.10)$$

Die P-Matrix stellt die perspektivische Projektion vom Kamerarahmen in den entsprechenden Indexkoordinatenrahmen dar. Die Variablen

$$f_x = \frac{f_c}{D_x} \quad (8.11)$$

$$f_y = \frac{f_c}{D_y}$$

sind die Brennweiten der Kamera entlang der x_c - und y_c -Achse. Die Parameter f_x , f_y , u_0 und v_0 werden als intrinsische Parameter einer Kamera bezeichnet.

Im Allgemeinen sind die intrinsischen Parameter der Kamera nicht bekannt. Die Spezifikationen der Kamera und der Objektive sind nicht ausreichend genau. Die intrinsischen Parameter der Kamera werden daher durch den Kamerakalibrierungsprozess ermittelt. Wenn wir die intrinsischen Parameter der Kamera kennen, können wir die Indexkoordinaten (u, v) eindeutig aus den gegebenen Koordinaten (x_c, y_c, z_c) berechnen. Die Koordinaten (x_c, y_c, z_c) können nicht aus den bekannten (u, v) -Koordinaten bestimmt werden, ohne den Skalierungsfaktor zu kennen.

8.3 Rückprojektion

Das digitale Bild wird durch eine Pixelmatrix dargestellt. Da die Indexkoordinaten (u , v) keine Maßeinheiten haben, bedeutet dies, dass charakteristische Merkmale des Bildes eher qualitativ als quantitativ beschrieben werden. Wenn wir die Abstände in metrischen Einheiten ausdrücken möchten, müssen wir die Beziehung zwischen den Indexkoordinaten (u , v) und den Koordinaten (x_r , y_r , z_r) im 3D-Referenzrahmen kennen. Ohne die tatsächlichen Abmessungen oder die Geometrie der Szene zu kennen, ist es unmöglich, die Merkmale des Bildes zu erkennen.

8.3.1 Einzelne Kamera

Nehmen wir an, wir hätten ein Roboter-Vision-System mit einer einzigen Kamera. Das System erhält als Eingabe das Bild des Roboterarbeitsraums und muss als Ausgabe geometrische Messungen reproduzieren. Die notwendigen Transformationen zwischen den Koordinatensystemen sind aus Abb. 8.4 ersichtlich.

Nehmen wir an, dass wir nun in der Lage sind, den Punkt q im Bild zu erkennen. Unser Ziel ist es, die Koordinaten des realen Punktes Q aus den Koordinaten seines Bildes q zu bestimmen. Dies ist das Problem der Rückprojektion. Um das Problem zu lösen, müssen wir wissen, wie die Koordinaten des Punktes q mit den Koordinaten des realen Punktes Q im Referenzrahmen zusammenhängen, was das Problem der Vorwärtsprojektion dar-

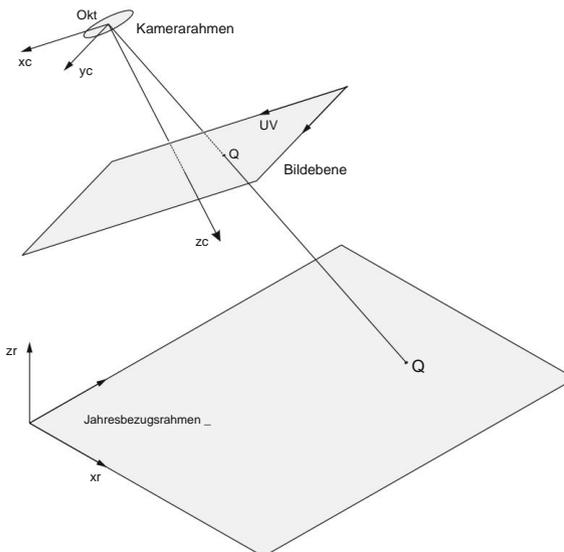


Abb. 8.4 Die Koordinatenrahmen in einem Roboter-Vision-System

Lösen wir zunächst das Problem der Vorwärtsprojektion. Der Punkt Q wird durch die Koordinaten (x_r, y_r, z_r) im Referenzkoordinatensystem angegeben. Wir möchten die Koordinaten seines Bildes $q = (u, v)$ bestimmen, ausgedrückt im Indexrahmen. Der Rahmen $x_c - y_c - z_c$ ist an der Kamera befestigt. Die Matrix M repräsentiert die Transformation von der Referenz in das Kamerabild

$$\begin{pmatrix} \tilde{y}^{x_c} \\ \tilde{y}^{y_c} \\ \tilde{y}^{z_c} \\ \tilde{y}^1 \end{pmatrix} = M \begin{pmatrix} \tilde{y}^{x_r} \\ \tilde{y}^{y_r} \\ \tilde{y}^{z_r} \\ \tilde{y}^1 \end{pmatrix} \quad (8.12)$$

Durch die Kombination der Gleichungen (8.12) und (8.9) erhalten wir

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = P N \begin{pmatrix} \tilde{y}^{x_r} \\ \tilde{y}^{y_r} \\ \tilde{y}^{z_r} \\ \tilde{y}^1 \end{pmatrix} \quad (8.13)$$

Die Beziehung (8.13) beschreibt die Vorwärtsprojektion. Die Elemente der P -Matrix sind die intrinsischen Parameter der Kamera, während die Elemente der M -Matrix ihre extrinsischen Parameter darstellen. Die 3×4 -Matrix

$$H = PM \quad (8.14)$$

wird als Kalibrierungsmatrix der Kamera bezeichnet. Es wird im Kalibrierungsprozess verwendet, um sowohl die intrinsischen als auch die extrinsischen Parameter der Kamera zu bestimmen.

Im weiteren Text werden wir die Rückwärtsprojektion betrachten. Unser Ziel ist es, aus den bekannten Koordinaten des Bildpunktes (u, v) und der Kalibrierungsmatrix H die Koordinaten (x_r, y_r, z_r) des realen Punktes Q zu bestimmen. Der Skalierungsfaktor s ist nicht bekannt. In (8.13) haben wir vier Unbekannte s, x_r, y_r und z_r und nur drei Gleichungen für einen einzelnen Punkt im Raum.

Versuchen wir es mit den drei Punkten A, B und C (Abb. 8.5). Wir kennen die Abstände zwischen diesen drei Punkten. Ihre Koordinaten im Referenzrahmen sind

$$(x_{rj}, y_{rj}, z_{rj}), j = 1, 2, 3.$$

Die Koordinaten der entsprechenden Bildpunkte sind

$$(u_j, v_j), j = 1, 2, 3.$$

Die Vorwärtsprojektion kann in der folgenden Form geschrieben werden

$$s_j \begin{pmatrix} u_j \\ v_j \\ 1 \end{pmatrix} = H \begin{pmatrix} \tilde{y}^{x_{rj}} \\ \tilde{y}^{y_{rj}} \\ \tilde{y}^{z_{rj}} \\ \tilde{y}^1 \end{pmatrix} \quad (8.15)$$

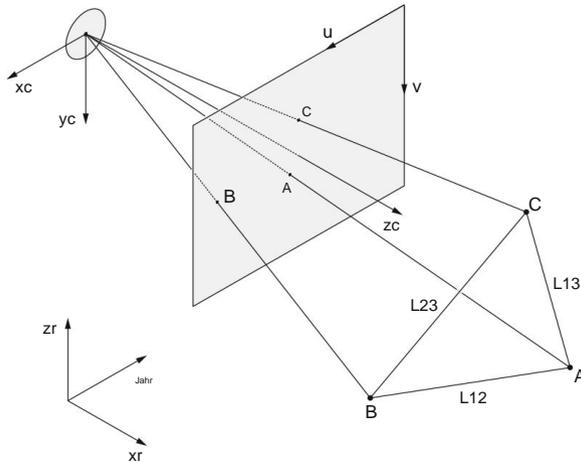


Abb. 8.5 Beispiel für die Projektion von drei Punkten

In Gl. (8.15) haben wir 12 Unbekannte und 9 Gleichungen. Zur Lösung des Problems benötigen wir zusätzlich drei Gleichungen. Diese Gleichungen können aus der Größe des Dreiecks ermittelt werden, das durch die Punkte A, B und C dargestellt wird. Wir bezeichnen die Dreiecksseiten AB, BC und CA als die Längen L12, L23 und L31

$$\begin{aligned}
 L_{12}^2 &= (x_{r1} - x_{r2})^2 + (y_{r1} - y_{r2})^2 + (z_{r1} - z_{r2})^2 \\
 L_{23}^2 &= (x_{r2} - x_{r3})^2 + (y_{r2} - y_{r3})^2 + (z_{r2} - z_{r3})^2 \\
 L_{31}^2 &= (x_{r3} - x_{r1})^2 + (y_{r3} - y_{r1})^2 + (z_{r3} - z_{r1})^2.
 \end{aligned} \tag{8.16}$$

Jetzt haben wir zwölf Gleichungen für die zwölf Unbekannten. Somit existiert die Lösung des inversen Problems. Es ist unpraktisch, dass die letzten drei Gleichungen nichtlinear sind und für die numerische Lösung der Gleichungen ein Computer erfordern. Der Ansatz wird als modellbasierte Rückwärtsprojektion bezeichnet.

8.3.2 Stereovision

Da das Modell des beobachteten Objekts meist nicht verfügbar ist oder sich das Objekt mit der Zeit verändert, müssen andere Lösungen für das Rückprojektionsproblem gefunden werden.

Eine mögliche Lösung ist der Einsatz von Stereovision: Erfassung auf Basis von zwei Kameras. Das Prinzip ähnelt der menschlichen visuellen Wahrnehmung, bei der sich die vom linken und rechten Auge gesehene Bilder aufgrund der Parallaxe geringfügig unterscheiden und das Gehirn die Unterschiede zwischen den Bildern verwendet, um die Entfernung zum beobachteten Objekt zu bestimmen.

Der Einfachheit halber gehen wir von zwei parallelen Kameras aus, die den Punkt Q beobachten, wie in Abb. 8.6 dargestellt. Punkt Q wird auf die Bildebene der linken und rechten Kamera projiziert. Die Bildebene der linken Kamera enthält die Projektion q_l mit den Koordinaten $x_{l,l}$ und $y_{l,l}$.

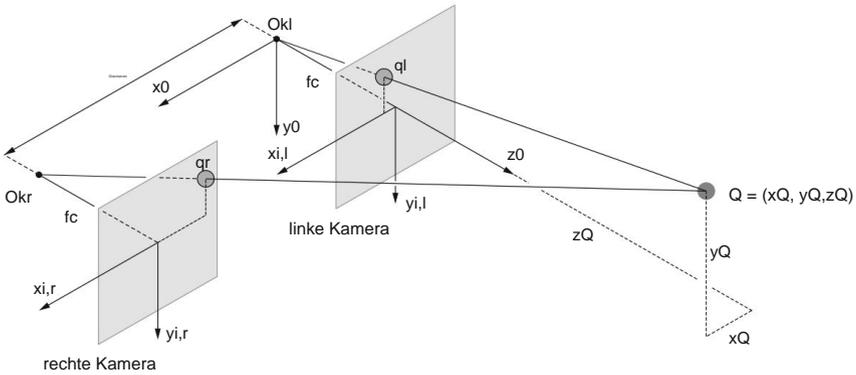


Abb. 8.6 Stereoansicht des Punktes Q mit zwei parallelen Kameras

während die Bildebene der rechten Kamera die Projektion q_r mit den Koordinaten $x_{i,r}$ und $y_{i,r}$ auf den Achsen des Vision-System-Koordinatenrahmens x_0 - y_0 - z_0 haben die gleichen Richtungen wie der Koordinatenrahmen der linken Kamera.

Abbildung 8.7a zeigt die Draufsicht, während Abbildung 8.7b die Seitenansicht der Situation in Abbildung 8.6 zeigt. Diese Ansichten helfen uns, die Koordinaten des Punktes Q zu berechnen. Aus der Geometrie in Abb. 8.7a können wir die folgenden Beziehungen extrahieren (Abstände x_Q , y_Q und z_Q beziehen sich auf den Koordinatenrahmen x_0 - y_0 - z_0).

$$\frac{z_Q}{fc} = \frac{x_Q}{x_{i,l}} \quad (8.17)$$

$$\frac{z_Q}{fc} = \frac{x_Q \cdot dc}{x_{i,r} \cdot fc}$$

Dabei ist dc der Abstand zwischen den Kameras. Aus der ersten Gleichung in (8.17) drücken wir aus

$$x_Q = z_Q \frac{x_{i,l}}{fc} \quad (8.18)$$

und in die zweite Gleichung einsetzen, um zu erhalten

$$\frac{x_{i,l} z_Q}{x_{i,r} fc} = \frac{z_Q}{fc} = \frac{dc}{x_{i,r}} \quad (8.19)$$

Wir können dann den Abstand z_Q zum Punkt Q bestimmen

$$z_Q = \frac{fc dc}{x_{i,l} \cdot x_{i,r}} \quad (8.20)$$

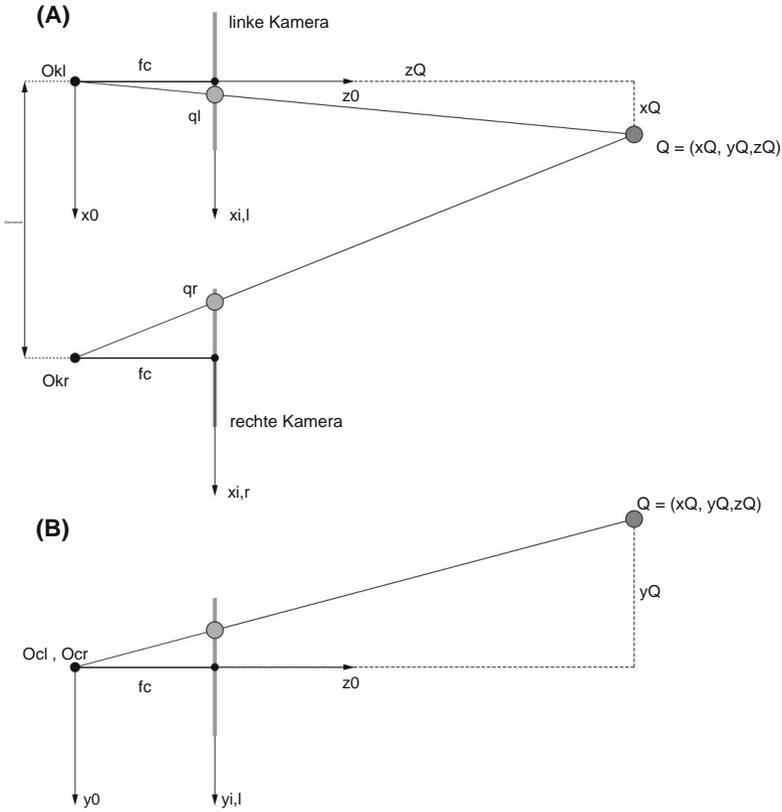


Abb. 8.7 Projektionen des Punktes Q auf die Ebenen der linken und rechten Kamera. Die obere Abbildung a zeigt eine Ansicht beider Kameras von oben, während die untere Abbildung b eine Seitenansicht der Kameras zeigt

Der Abstand x_Q lässt sich aus Gl. (8.18). Zur Bestimmung des Abstandes y_Q verweisen wir auf Abb. 8.7b. Aus der Geometrie können wir Beziehungen extrahieren

$$\frac{z_Q}{f_c} = \frac{y_Q}{y_{i,l}}, \quad (8.21)$$

So können wir die verbleibende Koordinate berechnen

$$y_Q = z_Q \cdot \frac{y_{i,l}}{f_c} \quad (8.22)$$

Die Verwendung von zwei Kameras ermöglicht die Berechnung der Position (und Ausrichtung) eines Objekt im Raum ohne ein genaues Modell des Objekts.

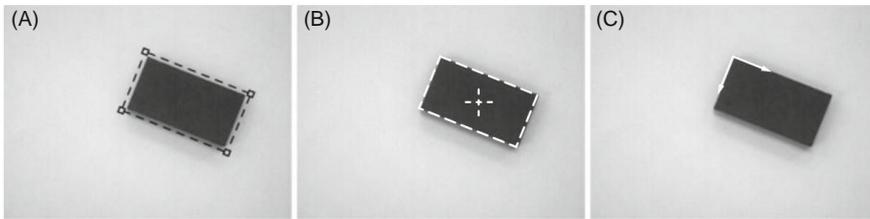


Abb. 8.8 a Modelldefinition, b erkannte Objektmerkmale, c lokalisiertes Objekt

8.4 Bildverarbeitung

Im Gegensatz zu den meisten anderen sensorischen Systemen liefern visuelle Systeme sehr weitreichende Informationen, die komplexe Verarbeitungsalgorithmen erfordern, bevor sie zur Robotersteuerung genutzt werden können. Das Ziel der Bildverarbeitung besteht darin, numerische Informationen aus dem Bild zu gewinnen, die eine robuste Beschreibung des Objekts in der Szene liefern. Ein Beispiel für das Ergebnis der Bildverarbeitung ist in [Abb. 8.8 dargestellt](#). Ein Objekt wird zunächst in der Szene identifiziert und dann seine Pose bestimmt, die mit dem Ko

Die Bildverarbeitung geht über den Rahmen dieses Buches hinaus und wird hier nicht speziell behandelt.

8.5 Objektpose aus dem Bild

Um den Roboter relativ zum interessierenden Objekt zu steuern, muss die Objekthaltung relativ zum Roboterkoordinatensystem $x-y-z$ definiert werden. Wie in [Abb. 8.8 dargestellt](#), ist die Pose des Objekts im Bildkoordinatenrahmen nach der Bildverarbeitung bekannt. Um seine Pose im Roboterrahmen zu bestimmen, muss die Transformation zwischen dem Bild und dem Roboterkoordinatenrahmen definiert werden, die das Ergebnis der Kamerakalibrierung ist. [Abbildung 8.9](#) zeigt einen einfachen Ansatz für das Kalibrierungsproblem, bei dem die Bildebene parallel zur horizontalen Ebene verläuft. Der Einfachheit halber befindet sich der Bildrahmen $x_i-y_i-z_i$ am selben Punkt wie der Indexrahmen $u-v$ (die z_i - Achse wurde dem Bildrahmen hinzugefügt, um die Drehung um die vertikale Achse hervorzuheben).

8.5.1 Kamerakalibrierung

Die Kamera ist an einer festen Position über dem Arbeitsbereich des Roboters montiert. Die Kalibrierung erfolgt mit dem Kalibrierungsmuster (Schachbrett) und der Kalibrierungsspitze am Roboter-Endeffektor. Das Kalibrierungsmuster kann mit einer Referenzmarkierung ergänzt werden, die im Bild erscheint und als Referenzpunkt oder Maß dient. Das Ziel des Kalibrierungsverfahrens besteht darin, die Transformationsmatrix H_i zwischen den Bildern zu finden

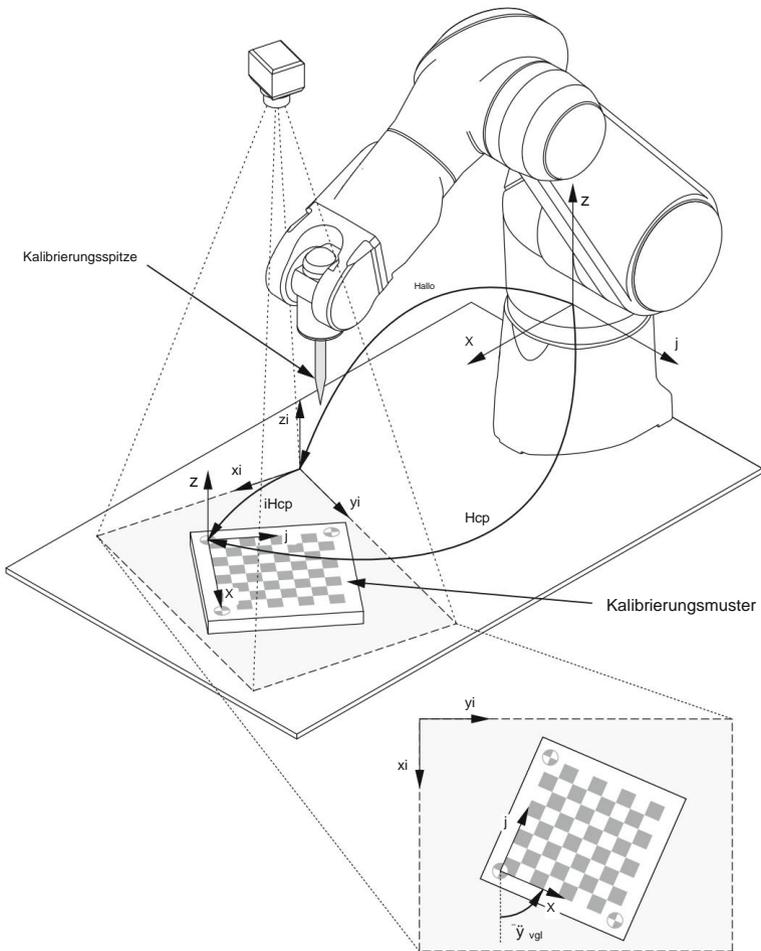


Abb. 8.9 Transformationen zur Kamerakalibrierung

und die Roboterkoordinatensysteme $x_i-y_i-z_i$ und $x-y-z$. Basierend auf den Beziehungen in Abb. 8.9 kann die folgende Gleichung geschrieben werden

$$H_{cp} = H_{allo} iH_{cp}, \quad (8.23)$$

Dabei sind H_{cp} und iH_{cp} die Posen des Kalibrierungsmusters, ausgedrückt im Roboter bzw. im Bildkoordinatenrahmen.

Die im Bildkoordinatenrahmen $x_i-y_i-z_i$ ausgedrückte Lage des Kalibrierungsmusters iH_{cp} ist das Ergebnis der Bildverarbeitung

$$iH_{cp} = \begin{pmatrix} \cos \tilde{y}_{cp} \tilde{y} & \sin \tilde{y}_{cp} \tilde{y} & 0 & \sin \tilde{y}_{cp} & x_{cp} \\ \sin \tilde{y}_{cp} \tilde{y} & \cos \tilde{y}_{cp} \tilde{y} & 0 & \cos \tilde{y}_{cp} & y_{cp} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad (8.24)$$

wobei \tilde{y}_{cp} bzw. \tilde{y} (x_{cp} , y_{cp}) sind die Ausrichtung und Position des Kalibrierungsmusters (relativ zur Bildebene. Die Position wird in metrischen Einheiten ausgedrückt als

$$\begin{pmatrix} x_{cp} \\ y_{cp} \end{pmatrix} = \tilde{y} \begin{pmatrix} u_{cp} \\ v_{cp} \end{pmatrix}, \quad (8.25)$$

Dabei sind (u_{cp}, v_{cp}) die Ursprungskordinaten des Kalibrierungsmusters in Pixeln und \tilde{y} das Verhältnis zwischen der in metrischen Einheiten ausgedrückten Position und den Pixeln auf dem Bild (das Verhältnis kann aus dem Kalibrierungsmuster mit der bekannten Größe der schwarzen und weißen Felder ermittelt werden). Die Matrix iH_{cp} repräsentiert eine Drehung um die z_i -Achse und eine Verschiebung entlang der x_i - und y_i -Achsen des Bildkoordinatenrahmens.

Die Pose des Kalibrierungsmusters H_{cp} , ausgedrückt im Roboterkoordinatensystem $x-y-z$, kann mit der Kalibrierungsspitze am Roboterendeffektor und den auf dem Kalibrierungsmuster markierten Kalibrierungspunkten bestimmt werden. Durch Platzieren der Kalibrierungsspitze auf dem Kalibrierungspunkt, Aufzeichnen der Endeffektorkoordinaten des Roboters und Wiederholen des Vorgangs für die drei Kalibrierungspunkte wird ein Koordinatensatz erhalten, der die Definition der Kalibrierungsmusterhaltung relativ zum Roboterkoordinatenrahmen als ermöglicht

$$H_{cp} = \begin{pmatrix} \cos \tilde{y}_{cp} \tilde{y} & \sin \tilde{y}_{cp} \tilde{y} & 0 & x_{cp} \\ \sin \tilde{y}_{cp} \tilde{y} & \cos \tilde{y}_{cp} \tilde{y} & 0 & y_{cp} \\ 0 & 0 & 1 & z_{cp} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (8.26)$$

Dabei sind \tilde{y}_{cp} und (x_{cp}, y_{cp}, z_{cp}) die Ausrichtung bzw. Position des Kalibrierungsmusters relativ zum Roboterrahmen.

Aus Gl. (8.23), (8.24) und (8.26) die Transformationsmatrix zwischen dem Bild und die Roboterkoordinatenrahmen können erhalten werden als

$$H_i = H_{cp} iH_{\tilde{y}1} \quad \text{vgl.} \quad (8.27)$$

8.5.2 Objektpose

Mit dem bekannten H_i kann die Objekthaltung H_o relativ zum Roboterkoordinatensystem wie in Abb. 8.10 dargestellt bestimmt werden.

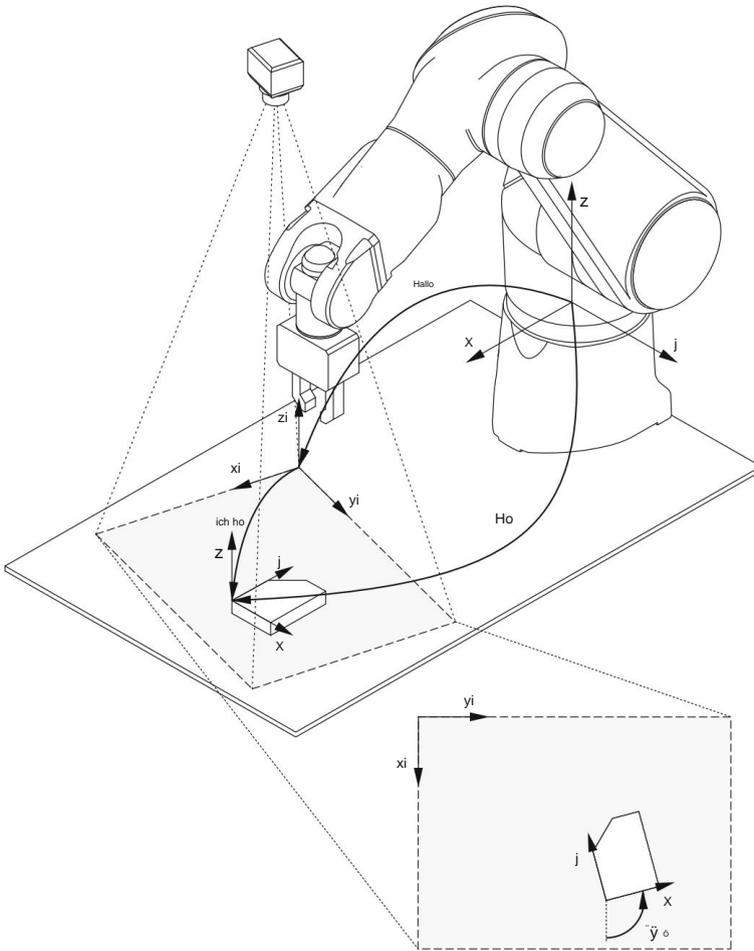


Abb. 8.10 Transformationen zur Berechnung der Objektposition

Die im Bildkoordinatensystem $x_i-y_i-z_i$ ausgedrückte **Pose des Objekts iH_o** ist das Ergebnis der Bildverarbeitung

$${}^iH_o = \begin{pmatrix} \cos i\gamma_o \ddot{y} & \sin i\gamma_o & 0 & \ddot{z}_o \\ \sin i\gamma_o \cos i\gamma_o & \cos i\gamma_o & 0 & \ddot{y}_o \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (8.28)$$

wobei $i\ddot{y}_o$ bzw. $i\ddot{x}_o, y_o$ sind die Ausrichtung und Position des Objekts relativ zum (Bildebene. Die Position wird in metrischen Einheiten ausgedrückt als

$$\begin{matrix} i\ddot{x}_o \\ i\ddot{y}_o \end{matrix} = \ddot{y} \begin{matrix} U_o \\ v_o \end{matrix}, \quad (8.29)$$

Dabei sind (u_o, v_o) die Objektsprungskordinaten in Pixel.

Schließlich kann H_o bestimmt werden als

$$H_o = \text{Hallo } iH_o. \quad (8.30)$$

Kapitel 9

Flugbahnplanung



In den vorherigen Kapiteln haben wir mathematische Modelle von Robotermechanismen untersucht. Zunächst interessierten wir uns für die Kinematik und Dynamik von Robotern. Bevor wir dieses Wissen auf die Robotersteuerung anwenden, müssen wir uns mit der Planung der Roboterbewegung vertraut machen. Das Ziel der Trajektorienplanung besteht darin, Referenzeingaben für das Robotersteuerungssystem zu generieren, die sicherstellen, dass der Endeffektor des Roboters der gewünschten Trajektorie folgt.

Die Roboterbewegung wird normalerweise im rechteckigen Weltkoordinatenrahmen definiert, der im Arbeitsbereich des Roboters am besten für die Roboter Aufgabe platziert ist. In der einfachsten Aufgabe definieren wir nur den Anfangs- und Endpunkt des Roboter-Endeffektors. Das inverse kinematische Modell wird dann verwendet, um die Gelenkvariablen zu berechnen, die der gewünschten Position des Roboter-Endeffektors entsprechen.

9.1 Interpolation der Trajektorie zwischen zwei Punkten

Beim Bewegen zwischen zwei Punkten muss der Roboter manipulator in einem bestimmten Zeitintervall t_f vom Anfangspunkt zum Endpunkt verschoben werden. Oft interessiert uns nicht die genaue Flugbahn zwischen den beiden Punkten. Dennoch müssen wir den zeitlichen Verlauf der Bewegung für jede Gelenkvariable bestimmen und die berechnete Trajektorie dem Steuereingang zur Verfügung stellen.

Die Gelenkgröße ist entweder der Winkel $\dot{\gamma}$ für das Rotationsgelenk oder die Verschiebung d für das Translationsgelenk. Bei der Interpolation der Trajektorie wird nicht zwischen Rotations- und Translationsgelenken unterschieden, sodass die Gelenkvariable allgemeiner mit q bezeichnet wird. Bei industriellen Manipulatoren, die sich zwischen zwei Punkten bewegen, wählen wir am häufigsten das sogenannte trapezförmige Geschwindigkeitsprofil. Die Roboterbewegung beginnt bei $t = 0$ mit konstanter Beschleunigung, gefolgt von der Phase konstanter Geschwindigkeit und endet mit der Phase konstanter Verzögerung (Abb. 9.1).

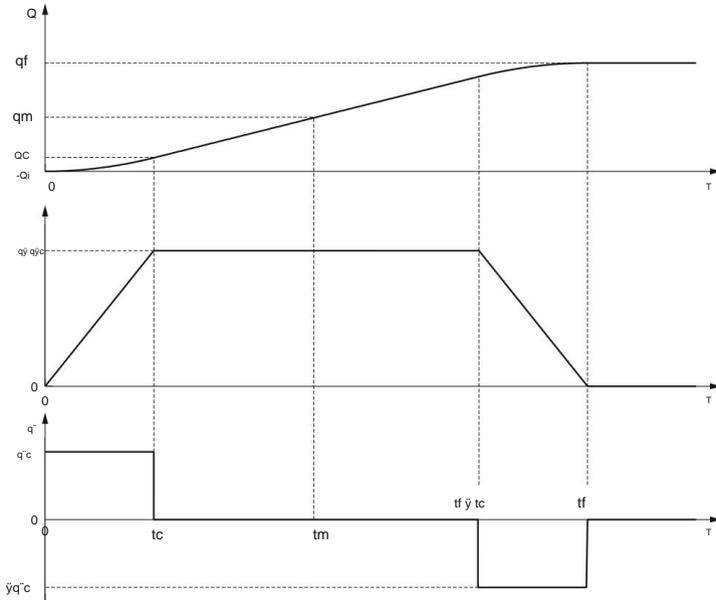


Abb. 9.1 Die Zeitabhängigkeit der Gelenkvariablen bei trapezförmigem Geschwindigkeitsprofil

Die resultierende Trajektorie des Gelenkwinkels oder der Gelenkverschiebung besteht aus dem zentralen linearen Intervall, das mit einem parabelförmigen Segment beginnt und endet. Die Anfangs- und Endgeschwindigkeiten der Bewegung zwischen den beiden Punkten sind Null. Die Dauer der Phase konstanter Beschleunigung entspricht dem Intervall mit konstanter Verzögerung. In beiden Phasen beträgt die Beschleunigung \ddot{q}^c . Auf diese Weise haben wir es mit einer symmetrischen Flugbahn zu tun, wobei

$$\dot{q}_m = \frac{\dot{q}_f + \dot{q}_i}{2} \quad \text{im Moment } t_m = \frac{t}{2}. \quad (9.1)$$

Die Flugbahn $q(t)$ muss mehrere Einschränkungen erfüllen, damit sich das Robotergelenk im erforderlichen Zeitintervall vom Anfangspunkt q_i zum Endpunkt q_f bewegt. Die Geschwindigkeit am Ende der anfänglichen Parabelphase muss gleich der Konstante sein \dot{q}_f . **Geschwindigkeit in der linearen Phase. Die Geschwindigkeit in der ersten Phase ergibt sich aus der Gleichung, die die Bewegung mit konstanter Beschleunigung beschreibt**

$$\ddot{q} = \ddot{q}^c t. \quad (9.2)$$

Am Ende der ersten Phase haben wir

$$\ddot{q}^c = \ddot{q}^c t_c. \quad (9.3)$$

9.1 Interpolation der Trajektorie zwischen zwei Punkten

125

Die Geschwindigkeit in der zweiten Phase kann mit Hilfe von Abb. 9.1 ermittelt werden

$$\dot{q} \ddot{q} = \frac{q_m \ddot{y} q_c}{t_m \ddot{y} t_c}, \quad (9.4)$$

Dabei stellt q_c den Wert der Gelenkvariablen q am Ende der anfänglichen parabolischen Phase (dh zum Zeitpunkt t_c) dar. Bis zu diesem Zeitpunkt findet die Bewegung mit konstanter Beschleunigung \ddot{q}^c statt, die Geschwindigkeit ergibt sich also aus Gl. (9.2). Die Zeitabhängigkeit der Gelenkposition erhält man durch Integration von Gl. (9.2)

$$\dot{q} dt^2 = \ddot{q}^c t dt = \ddot{q}^c q = \frac{1}{2} \ddot{q}^c t^2 + Q_i, \quad (9.5)$$

wobei die anfängliche Gelenkposition q_i als Integrationskonstante verwendet wird. Am Ende der ersten Phase haben wir

$$\dot{q}^c = q_i + 2 \frac{1}{\ddot{q}^c t_c^2}. \quad (9.6)$$

Die Geschwindigkeit am Ende der ersten Phase (9.3) ist gleich der konstanten Geschwindigkeit in der zweiten Phase (9.4)

$$\dot{q}^c t_c = \frac{q_m \ddot{y} q_c}{t_m \ddot{y} t_c}. \quad (9.7)$$

Durch Einfügen von Gl. (9.6) in Gl. (9.7) und unter Berücksichtigung des Ausdrucks (9.1) erhalten wir nach Umordnung die folgende quadratische Gleichung

$$\ddot{q}^c t_c^2 \ddot{y} \ddot{q}^c t_c + q_f \ddot{y} q_i = 0. \quad (9.8)$$

Die Beschleunigung \ddot{q}^c wird durch den gewählten Aktor und die dynamischen Eigenschaften der Robotermechanik bestimmt. Für ausgewählte q_i , q_f , \ddot{q}^c und t_f beträgt das Zeitintervall t_c

$$t_c = \frac{t_f}{2} \pm \frac{1}{2} \sqrt{\frac{4 q_f \ddot{y} q_i}{\ddot{q}^c}}. \quad (9.9)$$

Um die Bewegung zwischen der Anfangsposition q_i und der Endposition q_f zu erzeugen, muss in der ersten Phase das folgende Polynom generiert werden

$$q(t) = q_i + \ddot{q}^c t^2 - \frac{1}{2} \ddot{q}^c t^2. \quad (9.10)$$

In der zweiten Phase muss eine lineare Trajektorie beginnend im Punkt (t_c, q_c) mit der Steigung \dot{q}^c generiert werden

$$q(t) - q_c = \dot{q}^c (t - t_c). \quad (9.11)$$

Nach Umordnung erhalten wir

$$q(t) = q_i + \ddot{q} \frac{(t - t_c)^2}{2} - \ddot{q} (t - t_c) \quad (9.12)$$

In der letzten Phase muss die parabolische Flugbahn ähnlich wie in der ersten Phase erzeugt werden, nur dass nun der Extrempunkt in (t_f, q_f) liegt und die Kurve auf den Kopf gestellt wird

$$q(t) = q_f - \ddot{q} \frac{(t - t_f)^2}{2} + \ddot{q} (t - t_f) \quad (9.13)$$

Auf diese Weise erhielten wir analytisch die Zeitabhängigkeit des Winkels oder der Verschiebung des Rotations- oder Translationsgelenks, das sich von Punkt zu Punkt bewegt.

9.2 Interpolation durch Verwendung von Via-Punkten

Bei einigen Roboteraufgaben sind Bewegungen des Endeffektors erforderlich, die komplexer sind als Punkt-zu-Punkt-Bewegungen. Beim Schweißen müssen beispielsweise die gekrümmten Oberflächen der Objekte beachtet werden. Solche Trajektorien können erhalten werden, indem neben dem Anfangs- und dem Endpunkt auch die sogenannten Via-Punkte definiert werden, durch die sich der Roboter-Endeffektor bewegen muss.

In diesem Kapitel analysieren wir das Problem, bei dem wir die Trajektorie durch n über Punkte $\{q_1, \dots, q_n\}$ interpolieren wollen, die der Roboter in Zeitintervallen $\{t_1, \dots, t_n\}$ erreichen muss. Die Interpolation wird mit Hilfe trapezförmiger Geschwindigkeitsprofile durchgeführt. Die Flugbahn besteht aus einer Folge linearer Segmente, die die Bewegungen zwischen zwei Durchgangspunkten beschreiben, und parabolischen Segmenten, die die Übergänge durch die Durchgangspunkte darstellen. Um die Diskontinuität der ersten Ableitung zum Zeitpunkt t_k zu vermeiden, muss die Trajektorie $q(t)$ in der Nähe von q_k einen parabelförmigen Verlauf haben. Dadurch bleibt die zweite Ableitung im Punkt q_k (Beschleunigung) unstetig.

Die interpolierte Trajektorie, definiert als Folge linearer Funktionen mit parabolischen Übergängen durch die Durchgangspunkte (die Übergangszeit t_k), wird durch die folgenden Einschränkungen analytisch beschrieben

$$q(t) = \begin{cases} a_{1,k} (t - t_k) + a_{0,k} & t_k < t < t_{k+1} \\ b_{2,k} (t - t_k)^2 + b_{1,k} (t - t_k) + b_{0,k} & t_k < t < t_{k+1} \end{cases} \quad (9.14)$$

Die Koeffizienten $a_{0,k}$ und $a_{1,k}$ bestimmen die linearen Teile der Flugbahn, wobei k den Index des entsprechenden linearen Segments darstellt. Zu den parabelförmigen Übergängen gehören die Koeffizienten $b_{0,k}$, $b_{1,k}$ und $b_{2,k}$. Der Index k stellt die fortlaufende Nummer eines Parabelsegments dar.

Zunächst werden die Geschwindigkeiten in den linearen Segmenten aus den gegebenen Positionen und den entsprechenden Zeitintervallen berechnet, wie in Abb. 9.2 dargestellt. Wir gehen davon aus, dass die Anfangs- und Endgeschwindigkeit gleich Null sind. In diesem Fall haben wir

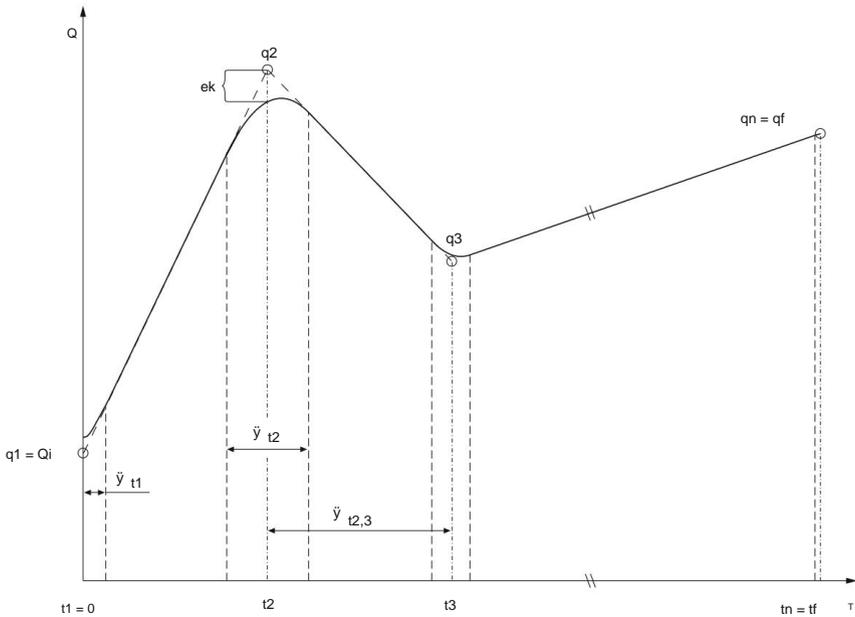


Abb. 9.2 Trajektorieninterpolation durch n Via-Punkte – lineare Segmente mit parabolischen Übergängen werden verwendet

$$\begin{aligned}
 & \ddot{y} = 0 & k = 1 \\
 & \ddot{y} = \frac{q_k \dot{y}_k - q_{k+1} \dot{y}_{k+1}}{t_{k+1} - t_k} & k = 2, \dots, n \\
 & \ddot{y} = 0 & k = n + 1.
 \end{aligned} \tag{9.15}$$

Weiterhin müssen wir die Koeffizienten der linearen Segmente $a_{0,k}$ und $a_{1,k}$ bestimmen. Der Koeffizient $a_{0,k}$ kann aus der linearen Funktion (9.14) unter Berücksichtigung ermittelt werden die bekannte Position zum Zeitpunkt t_k , wenn sich das Robotersegment dem Punkt nähert q_k

$$q(t_k) = q_k = a_{1,k} (t_k - \dot{y}_k) + a_{0,k} = a_{0,k}, \tag{9.16}$$

daher

$$t = t_k - \dot{y}_k a_{0,k} = q_k \quad k = 1, \dots, n - 1 \tag{9.17}$$

Der Koeffizient $a_{1,k}$ kann aus der zeitlichen Ableitung der linearen Funktion bestimmt werden (9.14)

$$\dot{q}(t) = a_{1,k}. \tag{9.18}$$

Durch Berücksichtigung der gegebenen Geschwindigkeiten im Zeitintervall $t_{k,k+1}$ erhalten wir

$$a_{1,k} = \dot{y}_{k,k+1} \quad k = 1, \dots, n - 1. \tag{9.19}$$

Auf diese Weise werden die Koeffizienten der linearen Segmente der Flugbahn bestimmt und wir können mit den Koeffizienten der Parabelfunktionen fortfahren. Wir gehen davon aus, dass die Übergangszeit als t_k vorgegeben ist. Wenn die Übergangszeit nicht vorgegeben ist, ist der Absolutwert der Beschleunigung $|\ddot{q}_k|$ im Zwischenpunkt muss zunächst definiert werden und anschließend wird aus den Beschleunigungen und Geschwindigkeiten vor und nach dem Zwischenpunkt die Übergangszeit berechnet. In diesem Fall muss nur das Vorzeichen der Beschleunigung bestimmt werden, indem das Vorzeichen der Geschwindigkeitsdifferenz im Zwischenpunkt berücksichtigt

$$\ddot{q}_k = \text{sign}(q_{\dot{y},k,k+1} - \dot{y}_{qk\dot{y},k}) |\ddot{q}_k|, \quad (9.20)$$

Dabei bedeutet $\text{sign}(\cdot)$ das Vorzeichen des Ausdrucks in den Klammern. Ausgehend von den Werten der Beschleunigungen in den Zwischenpunkten und den Geschwindigkeiten vor und nach dem Zwischenpunkt wird die Zeit der Bewegung durch den Zwischenpunkt t_k berechnet (Verzögerung und Beschleunigung)

$$t_k = \frac{q_{\dot{y},k,k+1} - \dot{y}_{qk\dot{y},k}}{\ddot{q}_k}. \quad (9.21)$$

Wir werden mit der Berechnung der Koeffizienten der quadratischen Funktionen fortfahren. Die erforderliche Kontinuität der Geschwindigkeit beim Übergang vom linearen in das parabolische Bahnsegment zum Zeitpunkt (t_k) und die erforderliche Geschwindigkeit

Die Kontinuität beim Übergang vom parabolischen in den linearen Abschnitt bei $(t_k + \Delta t)$ stellt den Ausgangspunkt für die Berechnung der Koeffizienten $b_{1,k}$ und $b_{2,k}$ dar. Zuerst berechnen wir die zeitliche Ableitung der quadratischen Funktion (9.14)

$$\dot{q}(t) = 2b_{2,k}(t - t_k) + b_{1,k}. \quad (9.22)$$

Unter der Annahme, dass die Geschwindigkeit zum Zeitpunkt (t_k) ist $\dot{q}_{k\dot{y},k}$, während bei $(t_k + \Delta t)$, Es ist $(t_k + \Delta t) \dot{q}_{k,k+1}$ ist, können wir schreiben

$$\begin{aligned} \dot{q}_{k\dot{y},k} &= 2b_{2,k} t_k + b_{1,k} & t &= t_k + \frac{\Delta t}{2} \\ \dot{q}_{k,k+1} &= 2b_{2,k} (t_k + \Delta t) + b_{1,k} & t &= t_k + \frac{\Delta t}{2} \end{aligned} \quad (9.23)$$

Durch Hinzufügen von Gl. (9.23) kann der Koeffizient $b_{1,k}$ bestimmt werden

$$b_{1,k} = \frac{\dot{q}_{k,k+1} + \dot{q}_{k\dot{y},k}}{2} \quad k = 1, \dots, n \quad (9.24)$$

und durch Subtrahieren von Gl. (9.23) wird der Koeffizient $b_{2,k}$ berechnet

$$b_{2,k} = \frac{\dot{q}_{k,k+1} - \dot{q}_{k\dot{y},k}}{2\Delta t} = \frac{\dot{q}_{k,k+1} - \dot{q}_{k\dot{y},k}}{2(t_{k+1} - t_k)} \quad k = 1, \dots, n \quad (9.25)$$

Unter Berücksichtigung der Kontinuität der Position zum Zeitpunkt $(t_k + \text{Koeffizient } \frac{t_k}{2})$, Die $b_{0,k}$ des quadratischen Polynoms kann berechnet werden. Bei $(t_k + \text{Position } q(t), \frac{t_k}{2})$, Die berechnet aus der linearen Funktion

$$q(t_k + \frac{t_k}{2}) = a_{1,k} t_k + \frac{t_k^2}{2} \ddot{y}(t_k) + a_{0,k} = \dot{y}(t_k) t_k + q_k \quad (9.26)$$

entspricht der Position $q(t)$, berechnet aus der quadratischen Funktion

$$\begin{aligned} q(t_k + \frac{t_k}{2}) &= b_{2,k} t_k^2 + b_{1,k} t_k + b_{0,k} \\ &= \frac{q\ddot{y}(t_{k+1}) - \dot{y}(t_k) \ddot{y}(t_k)}{2t_k} \frac{t_k^2}{2} + \frac{q\dot{y}(t_{k+1}) + \dot{y}(t_k) \ddot{y}(t_k)}{2} \cdot \frac{t_k}{2} + b_{0,k} \end{aligned} \quad (9.27)$$

Durch Gleichsetzung von (9.26) und (9.27) wird der Koeffizient $b_{0,k}$ bestimmt

$$b_{0,k} = q_k + (q\dot{y}(t_{k+1}) - \dot{y}(t_k) \ddot{y}(t_k)) \frac{t_k}{8} \quad (9.28)$$

Es kann überprüft werden, dass der berechnete Koeffizient $b_{0,k}$ auch die Kontinuität der Position zum Zeitpunkt $(t_k + \frac{t_k}{2})$ gewährleistet. Eine solche Wahl des Koeffizienten $b_{0,k}$ verhindert, dass die gemeinsame Trajektorie durch den Punkt Q_k verläuft. Der Roboter nähert sich diesem Punkt nur mehr oder weniger. Der Abstand der berechneten Trajektorie vom Referenzpunkt hängt hauptsächlich vom Verzögerungs- und Beschleunigungszeitintervall t_k ab, das durch die erforderliche Beschleunigung $|\ddot{y}|$ vorgegeben ist. Der Fehler e_k der berechneten Trajektorie kann abgeschätzt werden, indem man die gewünschte Position q_k mit der tatsächlichen Position $q(t)$ zum Zeitpunkt t_k vergleicht, die man durch Einsetzen von t_k in die quadratische Funktion (9.14) erhält.

$$e_k = q_k - q(t_k) = q_k - b_{0,k} = \dot{y}(t_k) \frac{t_k}{8} \quad (9.29)$$

Es ist zu erkennen, dass der Fehler e_k nur dann Null ist, wenn die Geschwindigkeiten der linearen Segmente vor und nach den Durchgangspunkten gleich sind oder wenn das Zeitintervall t_k Null ist, was eine unendliche Beschleunigung bedeutet (was in der Realität nicht möglich ist).

Der beschriebene Ansatz zur Trajektorieninterpolation weist einen kleinen Mangel auf. Aus Gl. (9.29) lässt sich beobachten, dass der Roboter den Zwischenpunkt nicht erreicht, sondern ihn umrundet. Da die Anfangs- und Endpunkte der Flugbahn auch als Durchgangspunkte betrachtet werden, führt dies zu einem Fehler in der Flugbahnplanung. Am Startpunkt der Trajektorie unterscheiden sich die tatsächliche und die gewünschte Position um den Fehler e_1 (Abb. 9.3, die Lichtkurve zeigt die Trajektorie ohne Korrektur), der sich aus Gl. (9.29) ergibt. Der Fehler stellt einen Schritt im Positionssignal dar, der in der Robotik unerwünscht ist. Um diese abrupte Positionsänderung zu vermeiden, müssen der erste und der letzte Trajektorienpunkt getrennt von den Viapunkten behandelt werden.

Die erforderlichen Geschwindigkeiten am Start- und Endpunkt sollten Null sein. Die Geschwindigkeit am Ende des Zeitintervalls t_1 muss gleich der Geschwindigkeit im ersten sein

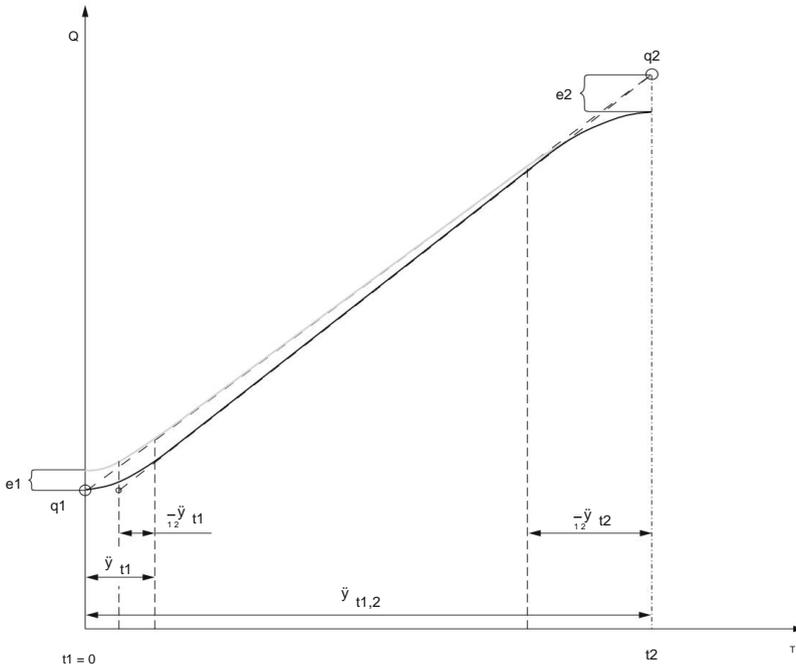


Abb. 9.3 Trajektorieninterpolation – vergrößerte Darstellung des ersten Segments der in Abb. 9.2 gezeigten Trajektorie. Die hellere Kurve stellt die Flugbahn ohne Korrektur dar, während die dunklere Kurve die korrigierte Flugbahn zeigt

lineares Segment. Zuerst berechnen wir die Geschwindigkeit im linearen Teil

$$\dot{q}_{1,2} = \frac{q_2 - q_1}{t_2 - t_1} \quad (9.30)$$

Gleichung (9.30) ähnelt Gl. (9.15), nur dass jetzt t_1 im Nenner subtrahiert wird, da sich in dem kurzen Zeitintervall (Beginn des Parabelsegments in Abb. 9.3) die Position des Roboters nur in sehr geringem Maße ändert. Dadurch wird eine höhere Geschwindigkeit im linearen Abschnitt der Flugbahn erreicht. Am Ende des Beschleunigungsintervalls t_1 haben wir

$$\dot{q}_{1,2} = \frac{q_2 - q_1}{t_2 - t_1} = \ddot{q} t_1 \quad (9.31)$$

Wir müssen auch die Beschleunigung \ddot{q} am Startpunkt der Flugbahn bestimmen. Unter der Annahme, dass sein absoluter Wert $|\ddot{q}|$ vorgegeben wurde, muss lediglich das Vorzeichen entsprechend gewählt werden. Die Wahl des Vorzeichens erfolgt auf Basis der Positionsdivergenz. Grundsätzlich ist der Geschwindigkeitsunterschied zu be-

Bei der Bestimmung des Vorzeichens der Beschleunigung ist jedoch die Anfangsgeschwindigkeit Null und das Vorzeichen kann daher von der Positionsdifferenz abhängen.

$$q''1 = \text{sign}(q2 - q1) |q''1|. \quad (9.32)$$

Aus Gl. (9.31) wird das Zeitintervall $t1$ berechnet

$$1 (q2 - q1) = q''1 t1 (t2 - t1) + \frac{1}{2} q''1 t1^2. \quad (9.33)$$

Nach Umordnung erhalten wir

$$\frac{1}{2} q''1 t1^2 + q''1 (t2 - t1) t1 - (q2 - q1) = 0, \quad (9.34)$$

also das Zeitintervall $t1$

$$t1 = \frac{-q''1 (t2 - t1) \pm \sqrt{q''1^2 (t2 - t1)^2 - 2q''1 (q2 - q1)}}{q''1}, \quad (9.35)$$

und nach der Vereinfachung von Gl. (9.35)

$$t1 = (t2 - t1) \frac{1 - \sqrt{1 - \frac{2(q2 - q1)}{q''1 (t2 - t1)^2}}}{2}. \quad (9.36)$$

In Gl. (9.36) wurde das Minuszeichen vor der Quadratwurzel gewählt, da das Zeitintervall $t1$ kürzer sein muss als $(t2 - t1)$. Aus Gl. (9.30) kann die Geschwindigkeit im linearen Teil der Flugbahn berechnet werden. Wie aus Abb. 9.3 ersichtlich ist (die dunklere Kurve stellt die korrigierte Trajektorie dar), beseitigt die eingeführte Korrektur den Fehler in der Ausgangsposition.

Ebenso wie für das erste Segment muss die Korrektur auch für das letzte Segment zwischen den Punkten $qn-1$ und qn berechnet werden. Die Geschwindigkeit im letzten linearen Segment beträ

$$q_{n-1,n} = \frac{q_n - q_{n-1}}{t_n - t_{n-1} - \frac{1}{2} t_{2n}}. \quad (9.37)$$

Im Nenner von Gl. (9.37) wurde der Wert $\frac{1}{2} t_{2n}$ abgezogen, da sich unmittelbar vor dem vollständigen Stopp des Roboters dessen Position nur sehr wenig ändert. Beim Übergang vom letzten linearen Segment zum letzten parabelförmigen Segment sind die Geschwindigkeiten gleich

$$\frac{q_n - q_{n-1}}{t_n - t_{n-1} - \frac{1}{2} t_{2n}} = q_{n,n}. \quad (9.38)$$

Die Beschleunigung (Verzögerung) des letzten Parabelsegments wird auf der Grundlage der Positionsdifferenz

$$q_n = \text{sign}(q_{n-1} - q_n) |q_n|. \quad (9.39)$$

Durch Einsetzen der obigen Gleichung in Gl. (9.38) berechnen wir auf ähnliche Weise wie für das erste Parabelsegment, auch die Dauer des letzten Parabelsegments

$$t_n = (t_n - t_{n-1}) + \frac{2(q_n - q_{n-1})}{q_n}. \quad (9.40)$$

Aus Gl. (9.37) kann die Geschwindigkeit des letzten linearen Segments bestimmt werden. Durch die Berücksichtigung der Korrekturen am Anfang und am Ende der Trajektorie wird der zeitliche Verlauf ermittelt durch die Via-Punkte wird berechnet. Auf diese Weise wurde die gesamte Flugbahn interpoliert an den n Punkten.

Kapitel 10

Robotersteuerung



Das Problem der Robotersteuerung kann als Berechnung der Kräfte oder Drehmomente erklärt werden, die von den Aktoren erzeugt werden müssen, um die Aufgabe des Roboters erfolgreich zu erfüllen. Sowohl im Übergangszeitraum als auch im stationären Zustand müssen entsprechende Arbeitsbedingungen gewährleistet sein. Die Roboter Aufgabe kann entweder als Ausführung der Bewegungen im freien Raum dargestellt werden, wo eine Positionskontrolle erfolgt, oder in Kontakt mit der Umgebung, wo eine Kontrolle der Kontaktkraft erforderlich ist. Zunächst untersuchen wir die Positionskontrolle eines Robotermechanismus, der nicht mit seiner Umgebung in Kontakt steht. Anschließend werden wir im weiteren Text die Positionsregelung um die Kraftregelung erweitern.

Das Problem der Robotersteuerung ist kein Einzelfall. Es gibt verschiedene Methoden, die sich in ihrer Komplexität und Wirksamkeit der Roboteraktionen unterscheiden. Die Wahl der Steuerungsmethode hängt von der Roboter Aufgabe ab. Ein wichtiger Unterschied besteht beispielsweise zwischen der Aufgabe, bei der der Endeffektor des Roboters genau der vorgeschriebenen Flugbahn folgen muss (z. B. Laserschweißen), und einer anderen Aufgabe, bei der es nur erforderlich ist, dass der Endeffektor des Roboters die gewünschte Endposition erreicht. Die Details der Flugbahn zwischen dem Anfangs- und dem Endpunkt sind nicht wichtig (z. B. Palettierung). Auch der mechanische Aufbau des Robotermechanismus hat Einfluss auf die Auswahl der geeigneten Steuerungsmethode. Die Steuerung eines kartesischen Roboter manipulators unterscheidet sich im Allgemeinen von der Steuerung eines anthropom

Die Robotersteuerung erfolgt üblicherweise im Weltkoordinatensystem, das vom Benutzer definiert wird und auch als Koordinatensystem der Roboter Aufgabe bezeichnet wird. Anstelle des Weltkoordinatenrahmens verwenden wir oft einen kürzeren Ausdruck, nämlich externe Koordinaten. Uns interessiert vor allem die Pose des Roboter-Endeffektors ausgedrückt in den Außenkoordinaten und seltener die Gelenkpositionen, die auch Innenkoordinaten genannt werden. Dennoch müssen wir uns darüber im Klaren sein, dass wir in jedem Fall die internen Koordinaten (dh Gelenkwinkel oder Verschiebungen) direkt steuern. Die Endeffektorhaltung wird nur indirekt gesteuert. Sie wird durch das kinematische Modell des Robotermechanismus und die vorgegebenen Werte der internen Koordinaten bestimmt.

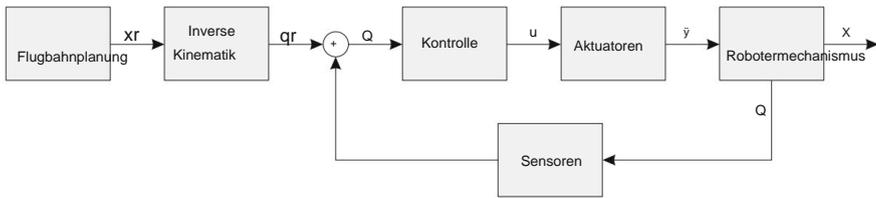


Abb. 10.1 Ein allgemeines Robotersteuersystem

Abbildung 10.1 zeigt ein allgemeines Robotersteuersystem. Die Eingabe in das Steuerungssystem ist die gewünschte Pose des Roboter-Endeffektors, die mithilfe der im vorherigen Kapitel vorgestellten Trajektorieninterpolationsmethoden ermittelt wird. Die Variable x_r repräsentiert die gewünschte (dh die Referenzhaltung) des Roboter-Endeffektors. Der x -Vektor, der die tatsächliche Pose des Roboter-Endeffektors beschreibt, umfasst im Allgemeinen sechs Variablen. Drei davon definieren die Position des Roboterendpunkts, während die anderen drei die Ausrichtung des Roboterendeffektors bestimmen. Wir

Die Position des Roboter-Endeffektors wird durch den Vektor vom Ursprung des Weltkoordinatenrahmens zum Roboter-Endpunkt bestimmt. Die Ausrichtung des Endeffektors kann auf verschiedene Arten dargestellt werden. Eine der möglichen Beschreibungen ist die sogenannte RPY-Notation, die aus der Luftfahrt stammt und in Abb. 4.4 dargestellt ist. Die Ausrichtung wird durch den Winkel $\dot{\gamma}$ um die z -Achse (Roll), den Winkel $\dot{\gamma}$ um die y -Achse (Pitch) und den Winkel $\dot{\gamma}$ um die x -Achse (Yaw) bestimmt.

Mithilfe des inversen Kinematikalgorithmus werden die internen Koordinaten q_r berechnet, die der gewünschten Endeffektorhaltung entsprechen. Die Variable q_r repräsentiert die Gelenkposition (dh den Winkel $\dot{\gamma}$ für das Rotationsgelenk und den Abstand d für das Translationsgelenk). Die gewünschten internen Koordinaten werden mit den tatsächlichen internen Koordinaten im Robotersteuersystem verglichen. Auf Basis des Positionsfehlers $q^- = q_r - q$ wird die Regelleistung u berechnet. Der Ausgang u wird von einem digitalen in ein analoges Signal umgewandelt, verstärkt und an die Aktuatoren des Roboters weitergeleitet. Die Aktoren sorgen für die für die erforderliche Roboterbewegung notwendigen Kräfte bzw. Drehmomente. Die Roboterbewegung wird durch die Sensoren ausgewertet, die im Kapitel über Robotersensoren beschrieben wurden.

10.1 Steuerung des Roboters in internen Koordinaten

Der einfachste Robotersteuerungsansatz basiert auf Steuerungen, bei denen der Regelkreis für jeden einzelnen Freiheitsgrad separat geschlossen wird. Solche Regler eignen sich zur Regelung unabhängiger Systeme zweiter Ordnung mit konstanten Trägheits- und Dämpfungsparametern. Dieser Ansatz eignet sich weniger für Robotersysteme, die durch nichtlineares und zeitlich variierendes Verhalten gekennzeichnet sind.

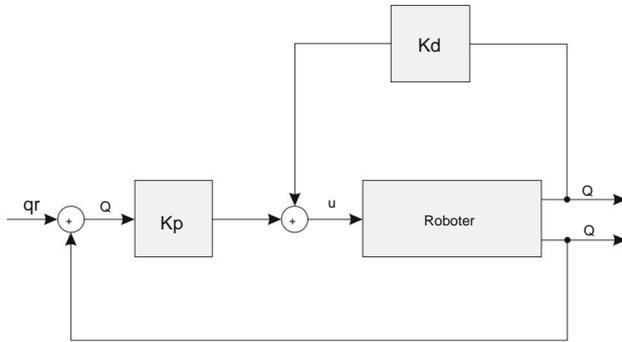


Abb. 10.2 PD-Lageregelung mit hoher Dämpfung

10.1.1 PD-Positionskontrolle

Zunächst wird ein einfacher Proportional-Differential-Regler (PD) analysiert. Das grundlegende Steuerungsschema ist in Abb. 10.2 dargestellt. Die Steuerung basiert auf der Berechnung des Positionsfehlers und der Bestimmung von Steuerparametern, die eine Reduzierung oder Unterdrückung des Fehlers ermöglichen. Der Positionsfehler wird für jedes Gelenk separat reduziert, was bedeutet, dass so viele Regler entwickelt werden müssen, wie es Freiheitsgrade gibt. Die Referenzpositionen q_r werden mit den tatsächlichen Positionen der Roboterelkenke q verglichen

$$\tilde{q} = q_r - q. \quad (10.1)$$

Der Positionsfehler \tilde{q} wird durch die proportionale Positionsverstärkung K_p verstärkt. Da ein Roboter manipulator über mehrere Freiheitsgrade verfügt, wird der Fehler \tilde{q} als Vektor ausgedrückt, während K_p eine Diagonalmatrix der Verstärkungen aller Gelenkregler ist. Der berechnete Steuereingang provoziert eine Bewegung des Roboters in Richtung einer Verringerung des Positionsfehlers. Da die Ansteuerung der Roboter motoren proportional zum Fehler ist, kann es vorkommen, dass der Roboter überschießt, anstatt in der gewünschten Position anzuhalten. Solche Überschinger sind in der Robotik nicht erlaubt, da sie zu Kollisionen mit Objekten in der Roboterumgebung führen können. Um sichere und stabile Roboteraktionen zu gewährleisten, wird ein Geschwindigkeitsregelkreis mit negativem Vorzeichen eingeführt. Der geschlossene Geschwindigkeitskreislauf bringt Dämpfung in das System. Sie wird durch die tatsächlichen Gelenkgeschwindigkeiten \dot{q} und die tatsächlichen Beschleunigungen \ddot{q} dargestellt. Das Dämpfungsglied multipliziert den Geschwindigkeitsfehler mit dem Geschwindigkeitsgewinn K_d dargestellt.

$$u = K_p(q_r - q) - \dot{q} + K_d\dot{q}, \quad (10.2)$$

Dabei stellt u die Steuereingaben (dh die gemeinsamen Kräfte oder Drehmomente) dar, die von den Aktoren bereitgestellt werden müssen. Aus Gl. (10.2) können wir feststellen, dass bei höheren Geschwindigkeiten der Roboterbewegungen der Geschwindigkeitsregelkreis die Gelenk betätigung reduziert und durch Dämpfung des Systems die Stabilität des Roboters gewährleistet.

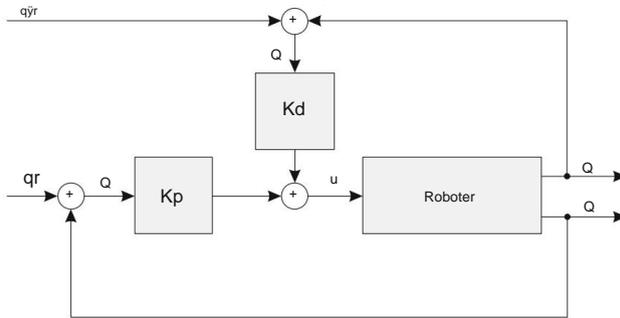


Abb. 10.3 PD-Lageregelung

Das in Abb. 10.2 dargestellte Steuerungsverfahren sorgt für eine hohe Dämpfung des Systems im schnellsten Teil der Flugbahn, was normalerweise nicht erforderlich ist. Ein solches Verhalten des Reglers kann durch die Aufrüstung des PD-Reglers mit dem Referenzgeschwindigkeitssignal vermieden werden. Dieses Signal wird als numerische Ableitung der gewünschten Position erhalten. Der Geschwindigkeitsfehler wird als Steuereingang verwendet

$$\ddot{y}q^{\sim} = q\ddot{y}r - \dot{y}q\ddot{y} \quad (10.3)$$

Der in Abb. 10.3 gezeigte Steueralgorithmus kann wie folgt geschrieben werden:

$$u = K_p(q_r - q) + K_d(\dot{q}_r - \dot{q}) \quad (10.4)$$

Da anstelle der Gesamtgeschwindigkeit \dot{q} die Differenz zwischen der Referenzgeschwindigkeit \dot{q}_r und \dot{q} verwendet wird, verringert sich die Dämpfungswirkung. Für einen positiven Unterschied kann der Regelkreis die Roboterbewegung sogar beschleunigen.

Die Synthese des PD-Lagereglers besteht aus der Bestimmung der Matrizen K_p und K_d .

Für eine schnelle Reaktion müssen die K_p - Gewinne hoch sein. Durch die richtige Wahl der K_d - Verstärkungen wird eine kritische Dämpfung der Robotersysteme erreicht. Die kritische Dämpfung gewährleistet eine schnelle Reaktion ohne Überschwingen. Solche Regler müssen für jedes Gelenk separat gebaut werden. Das Verhalten jedes Controllers ist völlig unabhängig von den Controllern, die zu den anderen Gelenken des Robotermechanismus gehören.

10.1.2 PD-Positionskontrolle mit Schwerkraftkompensation

Im Kapitel über Roboterdynamik haben wir festgestellt, dass der Robotermechanismus unter dem Einfluss von Trägheits-, Coriolis-, Zentripetal- und Gravitationskräften steht (5.56). Im Allgemeinen müssen auch Reibungskräfte, die in Robotergelenken auftreten, in das dynamische Modell des Roboters einbezogen werden. In einem etwas vereinfachten Modell wird nur die viskose Reibung berücksichtigt, die proportional zur Gelenkgeschwindigkeit ist (F_v ist eine Diagonalmatrix der).

Gelenkreibungskoeffizienten). Die aufgezählten Kräfte müssen von den Roboteraktoren überwunden werden, was aus der folgenden Gleichung, ähnlich zu Gl., ersichtlich ist. (5.56)

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + Fv\dot{q} + g(q) = \ddot{y}. \quad (10.5)$$

Bei der Entwicklung des PD-Controllers haben wir nicht auf die spezifischen Kräfte geachtet, die auf die Robotermechanik einwirken. Die Robotersteuerung berechnete die erforderlichen Betätigungskräfte allein auf Basis der Differenz zwischen Soll- und Ist-Gelenkposition. Eine solche Steuerung kann die Kraft, die zur Erzeugung der gewünschten Roboterbewegung erforderlich ist, nicht vorhersagen. Da die Kraft aus dem Positionsfehler berechnet wird, bedeutet dies, dass der Fehler im Allgemeinen nie gleich Null ist. Wenn wir das dynamische Robotermodell kennen, können wir die Kräfte vorhersagen, die für die Ausführung einer bestimmten Roboterbewegung erforderlich sind. Diese Kräfte werden dann unabhängig vom Positionsfehlersignal von den RoboterMotoren erzeugt.

Unter quasistatischen Bedingungen, wenn der Roboter stillsteht oder sich langsam bewegt, können wir von null Beschleunigungen $\ddot{q} \approx 0$ und Geschwindigkeiten $\dot{q} \approx 0$ ausgehen. Das dynamische Modell des Roboters wird wie folgt vereinfacht

$$\ddot{y} \approx g(q). \quad (10.6)$$

Nach Gl. (10.6) müssen die RoboterMotoren vor allem den Schwerkräfteeffekt kompensieren. Das Modell der Gravitationskräfte $\hat{g}(q)$ (der Zirkumflex bezeichnet das Robotermodell), das eine gute Näherung an die tatsächlichen Gravitationskräfte $g(q)$ darstellt, kann in den in Abb. 10.4 dargestellten Regelalgorithmus implementiert werden. Der in Abb. 10.2 dargestellte PD-Regler wurde um einen zusätzlichen Regelkreis erweitert, der die Gravitationskräfte aus der tatsächlichen Roboterposition berechnet und direkt dem Reglerausgang hinzufügt. Der in Abb. 10.4 dargestellte Regelalgorithmus kann wie folgt geschrieben werden

$$u = K_p(q_r - \hat{q}) + \dot{\hat{q}} + K_d\dot{\hat{q}} + \hat{g}(q). \quad (10.7)$$

Durch die Einführung der Schwerkraftkompensation wird der PD-Regler von der Last befreit, die durch die Schwerkraft verursachten Fehler zu reduzieren. Auf diese Weise werden die Fehler bei der Trajektorienverfolgung deutlich reduziert.

10.1.3 Steuerung des Roboters basierend auf inverser Dynamik

Bei der Untersuchung des PD-Reglers mit Schwerkraftkompensation haben wir das dynamische Robotermodell untersucht, um die Effizienz der Steuerungsmethode zu verbessern. Mit der auf Umkehrdynamik basierenden Regelungsmethode wird dieses Konzept weiter aufgeweitet. Aus den Gleichungen, die das dynamische Verhalten eines RoboterManipulators mit zwei Segmenten beschreiben (5.56), können wir deutlich erkennen, dass das Robotermodell nichtlinear ist. Ein linearer Regler wie der PD-Regler ist daher nicht die beste Wahl.

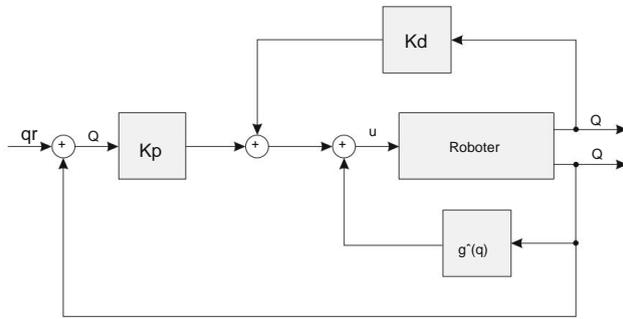


Abb. 10.4 PD-Regelung mit Schwerkraftkompensation

Wir werden das neue Steuerungsschema aus dem beschriebenen dynamischen Robotermodell ableiten nach Gl. (10.5). Nehmen wir an, dass die von den Motoren erzeugten Drehmomente \ddot{y} gleich sind an die Steuerausgänge u . Gleichung (10.5) kann umgeschrieben werden

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + Fv\ddot{y} + g(q) = u. \quad (10.8)$$

Im nächsten Schritt werden wir das direkte Roboterdynamikmodell bestimmen, das beschreibt Roboterbewegungen unter dem Einfluss der vorgegebenen Gelenkmomente. Zuerst drücken wir das aus Beschleunigung \ddot{q} aus Gl. (10.8)

$$\ddot{q} = B^{-1}(q)(u - (C(q, \dot{q})\dot{q} + Fv\ddot{y} + g(q))). \quad (10.9)$$

Durch Integration der Beschleunigung unter Berücksichtigung des Anfangsgeschwindigkeitswertes, Die Geschwindigkeit der Roboterbewegung wird ermittelt. Durch Integration der Geschwindigkeit während der Aufnahme Unter Berücksichtigung der Ausgangsposition berechnen wir die tatsächlichen Positionen in den Roboterjelenken. Das direkte dynamische Modell eines Robotermechanismus ist in Abb. 10.5 dargestellt.

Um die dynamischen Gleichungen zu vereinfachen, definieren wir eine neue Variable $n(q, \dot{q})$, umfasst alle dynamischen Komponenten außer der Trägheitskomponente

$$n(q, \dot{q}) = C(q, \dot{q})\dot{q} + Fv\ddot{y} + g(q). \quad (10.10)$$

Das dynamische Robotermodell kann mit der folgenden kürzeren Gleichung beschrieben werden

$$B(q)\ddot{q} + n(q, \dot{q}) = \ddot{y}. \quad (10.11)$$

Auf die gleiche Weise gilt Gl. (10.9) kann auch in kürzerer Form geschrieben werden

$$\ddot{q} = B^{-1}(q)(u - n(q, \dot{q})). \quad (10.12)$$

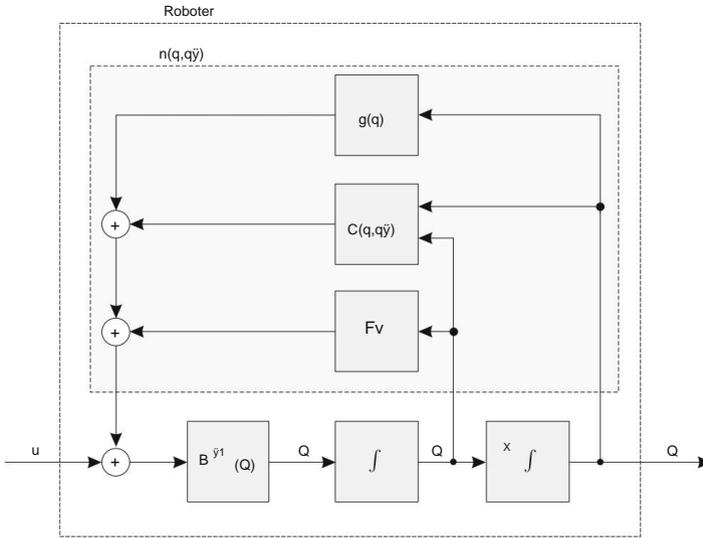


Abb. 10.5 Das direkte dynamische Modell eines Robotermechanismus

Nehmen wir an, dass das dynamische Modell des Roboters bekannt ist. Die Inertialmatrix $B^{\wedge}(q)$ ist eine Näherung der realen Werte $B(q)$, während $n^{\wedge}(q, \dot{q})$ eine Näherung von $n(q, \dot{q})$ wie folgt darstellt

$$n^{\wedge}(q, \dot{q}) = C^{\wedge}(q, \dot{q})\dot{q}\ddot{y} + F^{\wedge}v\dot{q}\ddot{y} + g^{\wedge}(q). \quad (10.13)$$

Der Reglerausgang u wird durch die folgende Gleichung bestimmt

$$u = B^{\wedge}(q)y + n^{\wedge}(q, \dot{q}), \quad (10.14)$$

wobei das ungefähre inverse dynamische Modell des Roboters verwendet wurde. Das System kombiniert die Gleichungen. (10.12) und (10.14) ist in Abb. 10.6 dargestellt .

Nehmen wir die Äquivalenz $B^{\wedge}(q) = B(q)$ und $n^{\wedge}(q, \dot{q}) = n(q, \dot{q})$ an. In Abb. 10.6 sehen wir, dass sich die Signale $n^{\wedge}(q, \dot{q})$ und $n(q, \dot{q})$ subtrahieren, da das eine mit positivem und das andere mit negativem Vorzeichen dargestellt wird. In ähnlicher Weise ist das Produkt der Matrizen $B^{\wedge}(q)^{-1}$ und $B(q)$ ergibt eine Einheitsmatrix, die weggelassen werden kann. Das vereinfachte System ist in Abb. 10.7 dargestellt . Durch die Implementierung der inversen Dynamik (10.14) wird das Regelsystem linearisiert, da zwischen dem Eingang y und dem Ausgang q nur zwei Integratoren vorhanden sind. Das System ist nicht nur linear, sondern auch entkoppelt (z. B. beeinflusst das erste Element des Vektors y nur das erste Element des Ortsvektors q). Aus Abb. 10.7 ist auch nicht schwer zu erkennen, dass die Variable y die Eigenschaften einer Beschleunigung hat, also

$$y = \ddot{q}. \quad (10.15)$$

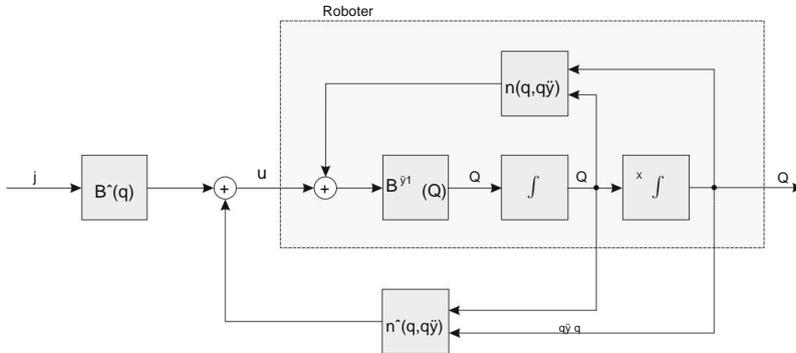


Abb. 10.6 Linearisierung des Regelsystems durch Implementierung des inversen dynamischen Modells

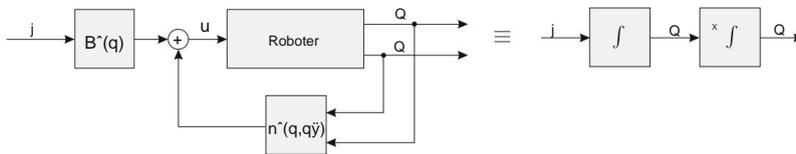


Abb. 10.7 Das linearisierte System

Im Idealfall würde es ausreichen, die gewünschten Gelenkbeschleunigungen als zweite Ableitung der gewünschten Gelenkpositionen zu ermitteln und das Steuerungssystem folgt den vorgegebenen Gelenkbahnen. Da wir nie über ein vollständig genaues dynamisches Modell des Roboters verfügen, wird es immer zu einem Unterschied zwischen der gewünschten und der tatsächlichen Gelenkposition kommen, der mit der Zeit zunimmt. Der Positionsfehler ist definiert durch

$$\mathbf{q}^- = \mathbf{q}_r - \mathbf{y}, \quad (10.16)$$

wobei \mathbf{q}_r die gewünschte Roboterposition darstellt. In ähnlicher Weise kann auch der Geschwindigkeitsfehler als Differenz zwischen der gewünschten und der tatsächlichen Geschwindigkeit definiert werden

$$\dot{\mathbf{y}}\mathbf{q}^- = \dot{\mathbf{q}}_r - \dot{\mathbf{y}}. \quad (10.17)$$

Der Vektor \mathbf{y} mit den Beschleunigungseigenschaften kann nun geschrieben werden als

$$\mathbf{y} = \mathbf{q}^- \ddot{\mathbf{r}} + K_p(\mathbf{q}_r - \mathbf{y}) + K_d(\dot{\mathbf{q}}_r - \dot{\mathbf{y}}). \quad (10.18)$$

Es besteht aus der Referenzbeschleunigung $\mathbf{q}^- \ddot{\mathbf{r}}$ und zwei beitragenden Signalen, die von den Fehlern von Position und Geschwindigkeit abhängen. Diese beiden Signale unterdrücken den Fehler, der aufgrund der unvollständig modellierten Dynamik entsteht. Das vollständige Steuerungsschema ist in Abb. 10.8 dargestellt.

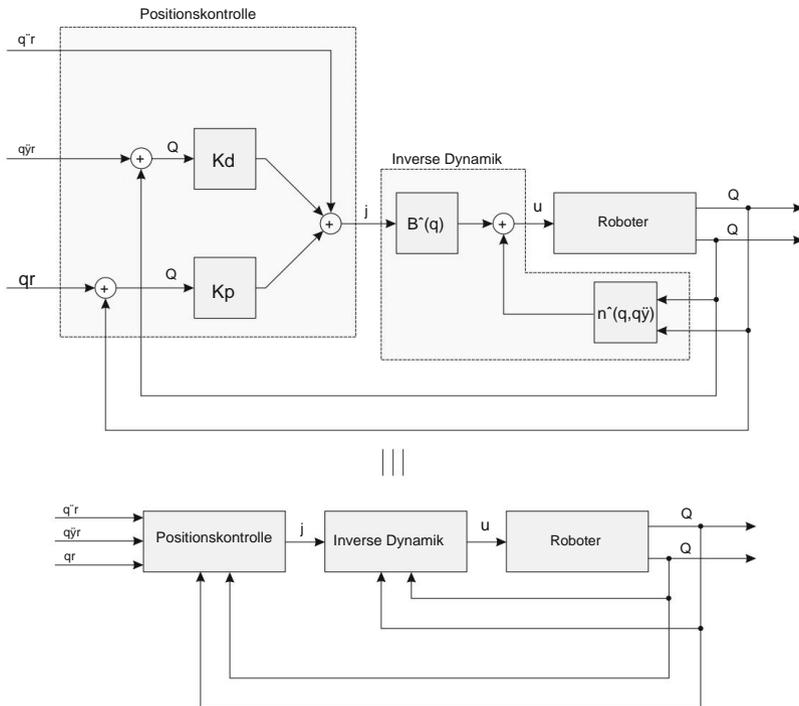


Abb. 10.8 Steuerung des Roboters auf Basis der Umkehrdynamik

Unter Berücksichtigung von Gl. (10.18) und der Gleichung $y = \ddot{q}$ kann die Differentialgleichung, die die Roboterdynamik beschreibt, geschrieben werden als

$$\ddot{q} + K_d \dot{y} \ddot{q} + K_p q = 0, \quad (10.19)$$

wobei der Beschleunigungsfehler $\ddot{q} = \ddot{q} - \dot{y} \ddot{q}$ eingeführt wurde. Die Differentialgleichung (10.19) beschreibt die Zeitabhängigkeit des Regelfehlers bei Annäherung an Null. Die Dynamik der Reaktion wird durch die Verstärkungen K_p und K_d bestimmt.

10.2 Steuerung des Roboters in externen Koordinaten

Alle bisher untersuchten Kontrollschemas basierten auf der Kontrolle der internen Koordinaten (dh der Gelenkpositionen). Die gewünschten Positionen, Geschwindigkeiten und Beschleunigungen wurden durch die Gelenkvariablen des Roboters bestimmt. Normalerweise sind wir mehr an der Bewegung des Endeffektors des Roboters interessiert als an den Verschiebungen bestimmter Robotergelenke.

An der Spitze des Roboters sind verschiedene Werkzeuge angebracht, um verschiedene Roboterarbeiten zu erledigen. Im weiteren Text konzentrieren wir uns auf die Robotersteuerung in den externen Koordinaten.

10.2.1 Steuerung basierend auf der transponierten Jacobi-Matrix

Die Regelungsmethode basiert auf der bereits bekannten Gl. (5.18) und verbindet die am Roboter-Endeffektor wirkenden Kräfte mit den Gelenkmomenten. Die Beziehung wird durch die Verwendung der transponierten Jacobi-Matrix definiert

$$\dot{y} = J^T(q)f, \quad (10.20)$$

wobei der Vektor \dot{y} die Gelenkdrehmomente darstellt und f die Kraft am Endpunkt des Roboters ist.

Unser Ziel ist es, die Pose des Roboter-Endeffektors zu steuern, wobei seine gewünschte Pose durch den Vektor x_r definiert wird und die tatsächliche Pose durch den Vektor x gegeben wird. Die Vektoren x_r und x bestehen im Allgemeinen aus sechs Variablen, von denen drei die Position des Roboterendpunkts bestimmen und drei die Ausrichtung des Endeffektors bestimmen, also $x = xy$. Roboter sind normalerweise nicht mit Sensoren ausgestattet, die die Haltung des Endeffektors beurteilen; Robotersensoren messen die Gelenkvariablen. Die Pose des Roboter-Endeffektors muss daher mithilfe der Gleichungen des direkten kinematischen Modells $x = k(q)$ bestimmt werden, das im Kapitel über Roboterkinematik (5.4) eingeführt wurde. Der Positionsfehler des Roboter-Endeffektors wird berechnet als:

$$\tilde{x} = x_r - \dot{y} x = x_r - \dot{y} k(q). \quad (10.21)$$

Der Positionsfehler muss auf Null reduziert werden. Ein einfaches proportionales Steuersystem mit der Verstärkungsmatrix K_p eingeführt wird

$$f = K_p \tilde{x}. \quad (10.22)$$

Bei der Analyse von Gl. (10.22) näher betrachten, stellen wir fest, dass es uns an die Gleichung erinnert, die das Verhalten einer Feder (in externen Koordinaten) beschreibt, wobei die Kraft proportional zur Federdehnung ist. Diese Überlegung hilft uns, das eingeführte Kontrollprinzip zu erklären. Stellen Sie sich vor, dass sechs Federn virtuell am Endeffektor des Roboters befestigt sind, eine Feder für jeden Freiheitsgrad (drei für die Position und drei für die Ausrichtung). Wenn sich der Roboter von der gewünschten Pose entfernt, werden die Federn gedehnt und ziehen den Endeffektor des Roboters mit einer Kraft proportional zum Positionsfehler in die gewünschte Pose. Die Kraft f drückt daher den Endeffektor des Roboters in die gewünschte Pose. Da die Bewegung des Roboters nur durch die Motoren in den Gelenken erzeugt werden kann, müssen die die Motoren steuernden Größen aus der Kraft f berechnet werden. Diese Berechnung wird mit Hilfe der transponierten Jacobi-Matrix durchgeführt, wie in Gl. (10.20)

$$u = J^T(q)f. \quad (10.23)$$

Der Vektor u repräsentiert die gewünschten Gelenkdrehmomente. Die Kontrollmethode basierend auf der transponierten Jacobi-Matrix ist in Abb. 10.9 dargestellt.

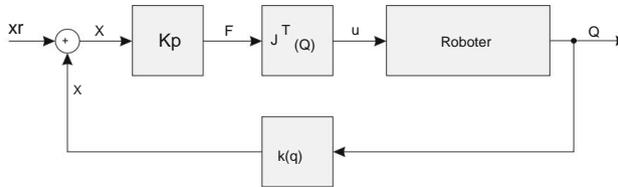


Abb. 10.9 Steuerung basierend auf der transponierten Jacobi-Matrix

10.2.2 Steuerung basierend auf der inversen Jacobi-Matrix

Die Steuerungsmethode basiert auf der Beziehung zwischen den Gelenkgeschwindigkeiten und den Geschwindigkeiten des Roboterendpunkts (5.10), die durch die Jacobi-Matrix gegeben ist. In Gl. (5.10) betonen wir die zeitlichen Ableitungen der äußeren Koordinaten x und inneren Koordinaten q

$$\dot{x}\dot{y} = J(q)q\dot{y}\dot{y} \quad \frac{dx}{dt} = J(q) \frac{dq}{dt} \quad (10.24)$$

Da dt im Nenner auf beiden Seiten von Gl. (10.24) kann darauf verzichtet werden. Auf diese Weise erhalten wir den Zusammenhang zwischen Änderungen der internen Koordinaten und Änderungen der Pose des Roboterendpunkts

$$dx = J(q)dq. \quad (10.25)$$

Gleichung (10.25) gilt nur für kleine Verschiebungen.

Wie bei der zuvor untersuchten Steuerungsmethode, die auf der transponierten Jacobi-Matrix basiert, können wir auch in diesem Fall zunächst den Fehler der Pose des Roboterendpunkts mithilfe von Gl. berechnen. (10.21). Wenn der Fehler in der Pose klein ist, können wir den Positionsfehler in den internen Koordinaten durch die umgekehrte Beziehung (10.25) berech

$$q\tilde{=} = J^{-1}(q)x\tilde{=} \quad (10.26)$$

Auf diese Weise wird das Steuerungsverfahren auf das bekannte Verfahren der Robotersteuerung in den internen Koordinaten übersetzt. Im einfachsten Beispiel, basierend auf dem Proportionalregler, können wir schreiben

$$u = Kp q\tilde{=} \quad (10.27)$$

Die Gleichung beschreibt das Verhalten einer Feder (in internen Koordinaten). Die auf der inversen Jacobi-Matrix basierende Steuerungsmethode ist in Abb. 10.10 dargestellt.

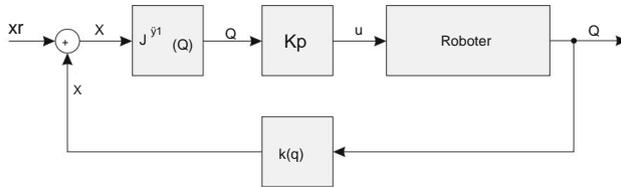


Abb. 10.10 Steuerung basierend auf der inversen Jacobi-Matrix

10.2.3 PD-Positionssteuerung mit Schwerkraftkompensation

Die PD-Lageregelung mit Schwerkraftkompensation wurde bereits eingehend untersucht für die internen Koordinaten. Jetzt leiten wir den analogen Steueralgorithmus ab die externen Koordinaten. Der Ausgangspunkt ist Gl. (10.21) und drückt den Fehler aus der Pose des Endeffektors. Die Geschwindigkeit des Roboterendpunkts wird mit berechnet mit Hilfe der Jacobi-Matrix aus den Gelenkgeschwindigkeiten

$$\dot{x}\tilde{y} = J(q)\dot{q}\tilde{y}. \quad (10.28)$$

Die Gleichung, die den PD-Regler in externen Koordinaten beschreibt, ist analog zu das in den internen Koordinaten (10.2) geschriebene

$$f = K_p x\tilde{y} - \dot{y} K_d x\tilde{y}. \quad (10.29)$$

In Gl. (10.29) wird der Posenfehler mit der Matrix der Positionsgewinne K_p multipliziert, während der Geschwindigkeitsfehler mit der Matrix K_d multipliziert wird. Das negative Vorzeichen des Geschwindigkeitsfehler führen zu einer Dämpfung im System. Die Gelenkmomente werden berechnet aus der Kraft f , die an der Spitze des Roboters wirkt, mit Hilfe der transponierten Jacobi-Funktion Matrix (ähnlich wie in Gleichung (10.23)) und durch Hinzufügen der kompensierenden Komponente Schwerkraft (wie in Gleichung (10.7)). Der Steueralgorithmus wird geschrieben als

$$u = J^T(q)f + g(q). \quad (10.30)$$

Das vollständige Steuerungsschema ist in Abb. 10.11 dargestellt.

10.2.4 Steuerung des Roboters basierend auf inverser Dynamik

Im Kapitel über die Steuerung von Robotern in den internen Koordinaten wird Folgendes erläutert Es wurde ein auf Umkehrdynamik basierender Regler eingeführt

$$u = B^{-1}(q)y + n(q, \dot{q}, \ddot{q}). \quad (10.31)$$

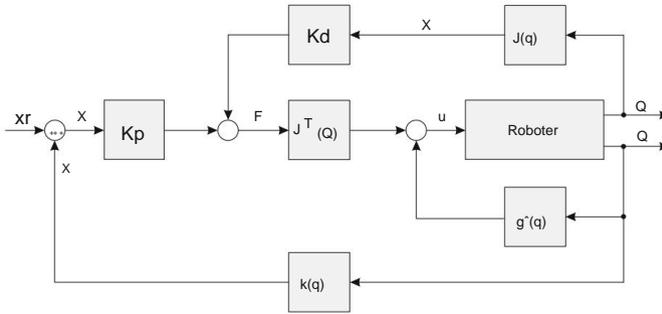


Abb. 10.11 PD-Regelung mit Schwerkraftkompensation in externen Koordinaten

Wir haben auch gelernt, dass der Vektor y die Eigenschaften einer Beschleunigung hat

$$y = \ddot{q}, \quad (10.32)$$

die so ermittelt wurde, dass der Roboter der gewünschten Flugbahn folgte ausgedrückt in den internen Koordinaten. Denn unser Ziel ist es, eine Kontrollmethode zu entwickeln Um die externen Koordinaten zu berücksichtigen, muss das y -Signal entsprechend angepasst werden. Gleichung (10.31), Die Linearisierung des Systems bleibt unverändert.

Wir beginnen wieder mit der Gleichung, die die Gelenkgeschwindigkeiten des Roboters in Beziehung setzt Endeffektorgeschwindigkeiten

$$\dot{x} = J(q)\dot{q}. \quad (10.33)$$

Durch Berechnung der Zeitableitung von Gl. (10.33) erhalten wir

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q}. \quad (10.34)$$

Der Fehler der Pose des Roboter-Endeffektors wird als Differenz zwischen ermittelt seine gewünschte und seine tatsächliche Pose

$$\tilde{x} = x_r - \dot{x} = x_r - \dot{J}(q)\dot{q}. \quad (10.35)$$

Auf ähnliche Weise wird der Geschwindigkeitsfehler des Roboter-Endeffektors bestimmt

$$\tilde{\dot{x}} = \dot{x}_r - \dot{x} = \dot{x}_r - \dot{J}(q)\dot{q}. \quad (10.36)$$

Der Beschleunigungsfehler ist die Differenz zwischen der gewünschten und der tatsächlichen Beschleunigung

$$\tilde{\ddot{x}} = \ddot{x}_r - \ddot{x}. \quad (10.37)$$

Bei der Entwicklung des inversen Dynamik-basierten Reglers in den internen Koordinaten, Gl. (10.19) wurde abgeleitet und beschreibt die Dynamik des Kontrollfehlers in der Form

$\ddot{q} + K_d \dot{q} + K_p q = 0$. Eine analoge Gleichung kann für den Fehler der Endeffektorhaltung geschrieben werden. Aus dieser Gleichung kann die Beschleunigung \ddot{x} des Roboter-Endeffektors ausgedrückt werden

$$\ddot{x} + K_d \dot{y}\ddot{x} + K_p x = \ddot{x}^r + K_d \dot{y}\ddot{x} + K_p x. \quad (10.38)$$

Aus Gl. (10.34) drücken wir \ddot{q} unter Berücksichtigung der Gleichung $y = q$ aus

$$\ddot{q} = J^{-1}(q) \ddot{x} - \dot{J}(q, \dot{q})\dot{q}. \quad (10.39)$$

Durch Ersetzen von \ddot{x} in Gl. (10.39) mit Ausdruck (10.38) wird der Steueralgorithmus basierend auf der inversen Dynamik in den externen Koordinaten erhalten

$$\ddot{q} = J^{-1}(q) \ddot{x}^r + K_d \dot{y}\ddot{x} + K_p x - \dot{J}(q, \dot{q})\dot{q}. \quad (10.40)$$

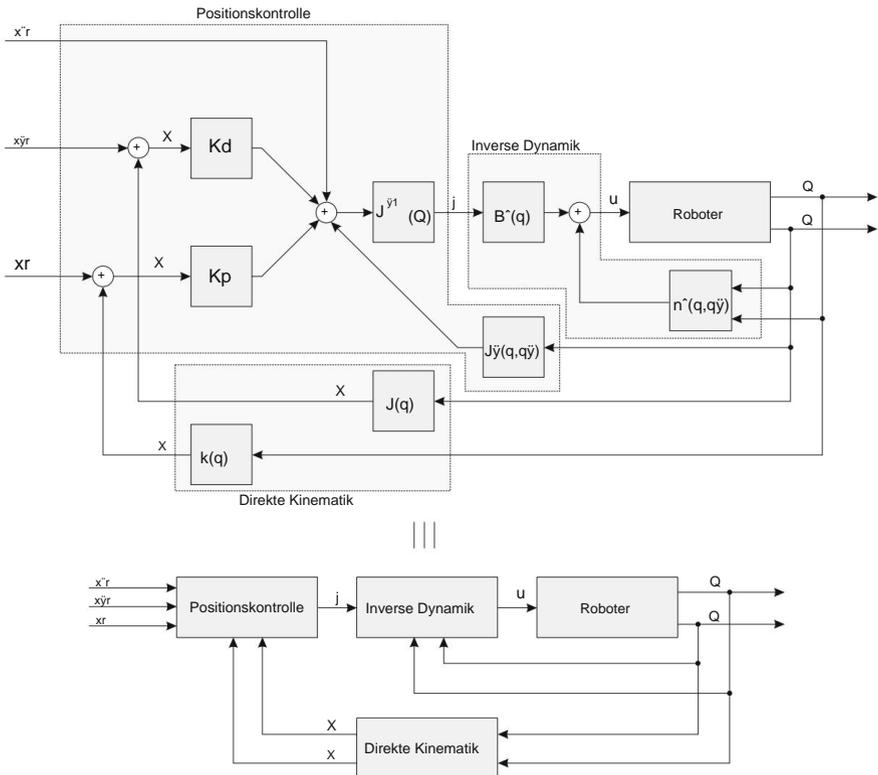


Abb. 10.12 Robotersteuerung basierend auf inverser Dynamik in externen Koordinaten

Das Regelungsschema, das die Linearisierung des Systems auf Basis der Umkehrdynamik (10.31) und die Regelung mit geschlossenem Regelkreis (10.40) umfasst, ist in Abb. 10.12 dargestellt.

10.3 Kontrolle der Kontaktkraft

Die Positionskontrolle reicht aus, wenn ein Roboter manipulator einer Trajektorie im freien Raum folgt. Wenn es zu Kontakt zwischen dem Endeffektor des Roboters und der Umgebung kommt, ist die Positionskontrolle kein geeigneter Ansatz. Stellen wir uns einen Roboter manipulator vor, der ein Fenster mit einem Schwamm reinigt. Da der Schwamm sehr nachgiebig ist, ist es möglich, die Kraft zwischen Roboter und Fenster durch Steuerung der Position zwischen Roboter greifer und Fenster zu steuern. Wenn der Schwamm ausreichend nachgiebig ist und wir die Position des Fensters genau genug kennen, wird der Roboter die Aufgabe angemessen erledigen.

Wenn die Nachgiebigkeit des Roboterwerkzeugs oder seiner Umgebung geringer ist, ist es nicht so einfach, Aufgaben auszuführen, die einen Kontakt zwischen dem Roboter und seiner Umgebung erfordern. Stellen wir uns nun einen Roboter vor, der mit einem steifen Werkzeug Farbe von einer glasigen Oberfläche kratzt. Jede Unsicherheit hinsichtlich der Position der Glasoberfläche oder eine Fehlfunktion des Robotersteuerungssystems verhindert eine zufriedenstellende Ausführung der Aufgabe. Entweder zerbricht das Glas, oder der Roboter wedelt nutzlos in der Luft.

Bei beiden Roboterarbeiten, also dem Reinigen eines Fensters oder dem Schaben einer glatten Oberfläche, ist es sinnvoller, statt der Position der Glasoberfläche die Kraft zu bestimmen, die der Roboter auf die Umgebung ausüben soll. Die meisten modernen Industrieroboter führen relativ einfache Aufgaben aus, wie zum Beispiel Punktschweißen, Spritzlackieren und verschiedene Punkt-zu-Punkt-Operationen. Einige Roboteranwendungen erfordern jedoch eine Steuerung der Kontaktkraft. Ein typisches Beispiel ist das Schleifen oder eine ähnliche Roboterbearbeitungsaufgabe. Ein wichtiger Bereich der Industrierobotik ist auch die Roboter montage, bei der mehrere Einzelteile zusammengebaut werden sollen. Bei solchen Roboterarbeiten ist die Erfassung und Steuerung der Kräfte von größter Bedeutung.

Für den effizienten Einsatz von Robotern bei einer Montageaufgabe ist der genaue Betrieb eines Roboter manipulators in einer unsicheren, nicht strukturierten und veränderlichen Umgebung erforderlich. Dabei müssen mehrere Bauteile mit hoher Genauigkeit zusammengeführt werden. Durch Messung und Regelung der Kontaktkräfte kann die erforderliche Positionsgenauigkeit des Roboter manipulators erreicht werden. Da bei der Roboter kraftsteuerung relative Messungen verwendet werden, sind die absoluten Fehler bei der Positionierung des Manipulators oder des Objekts nicht so kritisch wie bei der Roboter positionssteuerung. Beim Umgang mit steifen Objekten erzeugen bereits kleine Positionsänderungen große Kontaktkräfte. Die Messung und Steuerung dieser Kräfte kann zu einer deutlich höheren Positionsgenauigkeit der Roboterbewegung führen.

Wenn ein Roboter Kraft auf die Umgebung ausübt, haben wir es mit zwei Arten von Roboterarbeiten zu tun. Im ersten Fall möchten wir, dass der Endeffektor des Roboters in eine gewünschte Pose gebracht wird, während der Roboter mit der Umgebung in Kontakt steht. Dies ist der Fall bei der Roboter montage. Ein charakteristisches Beispiel ist das Einsetzen eines Pflocks in ein Loch. Die Bewegung des Roboters muss so beschaffen sein, dass die Kontaktkraft auf Null oder auf einen minimal zulässigen Wert reduziert wird. Bei der zweiten Art von Roboterarbeit benötigt

Endeffektor, der eine vorgegebene Kraft auf die Umgebung ausübt. Dies ist das Beispiel des Roboterschleifens. Dabei hängt die Roboterbewegung von der Differenz zwischen der gewünschten und der tatsächlich gemessenen Kontaktkraft ab.

Die Methode zur Steuerung der Roboterkraft basiert auf der Steuerung des Roboters mittels inverser Dynamik. Aufgrund der Interaktion des Roboters mit der Umgebung erscheint im inversen dynamischen Modell eine zusätzliche Komponente, die die Kontaktkraft f darstellt. Da die am Roboter-Endeffektor wirkenden Kräfte mithilfe der transponierten Jacobi-Matrix (5.18) in Gelenkdrehmomente umgewandelt werden, können wir das dynamische Robotermodell in der folgenden Form schreiben

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + Fv\dot{q} + g(q) = \ddot{y} \ddot{y} J^T(q)f. \quad (10.41)$$

Auf der rechten Seite der Gl. (10.5) haben wir die Komponente $\ddot{y} J^T(q)f$ repräsentieren Wechselwirkungskraft mit der Umgebung hinzugefügt. Es ist ersichtlich, dass die Kraft f durch die transponierte Jacobi-Matrix auf ähnliche Weise wirkt wie die Gelenkdrehmomente (dh sie versucht, eine Roboterbewegung zu erzeugen). Das Modell (10.41) kann durch Einführung in eine kürzere Form umgeschrieben werden

$$n(q, \dot{q}) = C(q, \dot{q})\dot{q} + Fv\dot{q} + g(q), \quad (10.42)$$

was uns das folgende dynamische Modell eines Roboters im Kontakt mit seiner Umgebung liefert

$$B(q)\ddot{q} + n(q, \dot{q}) = \ddot{y} \ddot{y} J^T(q)f. \quad (10.43)$$

10.3.1 Linearisierung eines Robotersystems durch inverse Dynamik

Bezeichnen wir die Stellgröße, die die gewünschten Betätigungsmomente in den Roboterjunkten darstellt, mit dem Vektor u . Gleichung (10.43) kann wie folgt geschrieben werden

$$B(q)\ddot{q} + n(q, \dot{q}) + J^T(q)f = u. \quad (10.44)$$

Aus Gl. (10.44) drücken wir das direkte dynamische Modell aus

$$\ddot{q} = B^{-1}(q) u - \ddot{y} n(q, \dot{q}) \ddot{y} J^T(q)f. \quad (10.45)$$

Gleichung (10.45) beschreibt die Reaktion des Robotersystems auf die Steuereingabe u . Durch Integration der Beschleunigung unter Berücksichtigung des Anfangsgeschwindigkeitswerts erhält man die tatsächliche Geschwindigkeit der Roboterbewegung. Durch Integration der Geschwindigkeit und unter Berücksichtigung der Ausgangsposition berechnen wir die tatsächlichen Positionen in den Roboterjunkten. Das beschriebene Modell wird durch den Block Roboter in A

Ähnlich wie bei der Entwicklung der auf inverser Dynamik basierenden Regelungsmethode werden wir das System linearisieren, indem wir das inverse dynamische Modell in den geschlossenen Regelkreis einbeziehen

$$\mathbf{u} = \mathbf{B}^{-1}(\mathbf{q})\mathbf{y} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}^T(\mathbf{q})\mathbf{f}, \quad (10.46)$$

Die Verwendung von Zirkumflex bezeichnet die geschätzten Parameter des Robotersystems. Der Unterschied zwischen Gl. (10.46) und (10.14), die die Steuerung basierend auf inverser Dynamik in internen Koordinaten darstellen, ist die $\mathbf{J}^T(\mathbf{q})\mathbf{f}$, Kompensation des Einflusses Komponente \mathbf{J} enz externer Kräfte auf den Robotermechanismus. Das Kontrollschema, das die Gleichungen kombiniert. (10.45) und (10.46) ist in Abb. 10.13 dargestellt. Unter der Annahme, dass die geschätzten Parameter den tatsächlichen Roboterparametern entsprechen, kann beobachtet werden, dass durch die Einführung des geschlossenen Regelkreises (10.46) das System linearisiert wird, da zwischen dem Eingang \mathbf{y} und dem Ausgang \mathbf{q} , wie bereits, nur zwei Integratoren vor

10.3.2 Kraftsteuerung

Nach der Linearisierung des Regelsystems muss der Eingangsvektor \mathbf{y} bestimmt werden. Die Kraftsteuerung wird in die Steuerung der Pose des Endeffektors übersetzt. Dies lässt sich vereinfacht mit folgender Überlegung erklären: Wenn wir möchten, dass der Roboter die auf die Umgebung ausgeübte Kraft erhöht, muss der Endeffektor des Roboters in die Richtung der Kraftwirkung verschoben werden. Jetzt können wir das Steuerungssystem verwenden, das entwickelt wurde, um den Roboter in den externen Koordinaten zu steuern (10.40).

Das Steuerungsschema des Roboter-Endeffektors einschließlich der Linearisierung unter Berücksichtigung der Kontaktkraft ist in Abb. 10.14 dargestellt.

Bis zu diesem Punkt haben wir hauptsächlich das Wissen über die Posensteuerung des Roboter-Endeffektors zusammengefasst, wie in den vorherigen Kapiteln erläutert. Im nächsten Schritt bestimmen wir die gewünschte Pose, Geschwindigkeit und Beschleunigung des Roboter-Endeffektors auf der Grundlage der zwischen dem Roboter-Endpunkt und seiner Umgebung gemessenen Kraft.

Nehmen wir an, wir wollen eine konstante gewünschte Kraft \mathbf{f}_r steuern. Mit dem Kraft-Handgelenk-Sensor wird die Kontaktkraft \mathbf{f} gemessen. Die Differenz zwischen gewünschter und gemessener Kraft stellt den Kraftfehler dar

$$\tilde{\mathbf{f}} = \mathbf{f}_r - \mathbf{f}. \quad (10.47)$$

Die gewünschte Roboterbewegung wird unter der Annahme berechnet, dass die Kraft $\tilde{\mathbf{f}}$ ein virtuelles Objekt mit der Trägheit B_c und der Dämpfung F_c verschieben muss. In unserem Fall ist das virtuelle Objekt tatsächlich der Endeffektor des Roboters. Zum leichteren Verständnis betrachten wir ein System mit nur einem Freiheitsgrad. Wenn auf ein solches System eine Kraft einwirkt, beginnt eine beschleunigte Bewegung. Die Bewegung wird durch die Kraft, die Masse des Objekts und die Dämpfung bestimmt. Der Roboter-Endeffektor verhält sich daher wie ein System bestehend aus einer Masse und einem Dämpfer, die unter dem Einfluss der

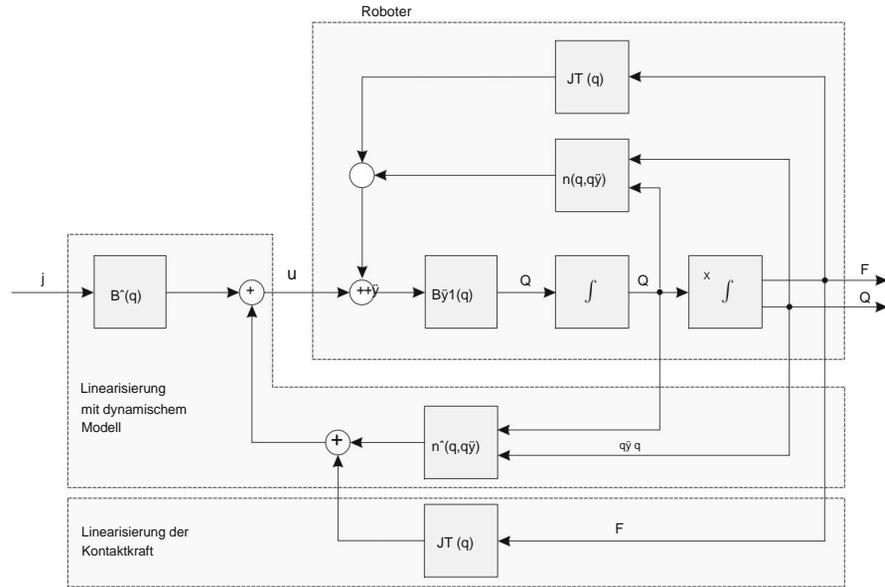


Abb. 10.13 Linearisierung des Regelsystems durch Implementierung des inversen dynamischen Modells und der gemessenen Kontaktkraft

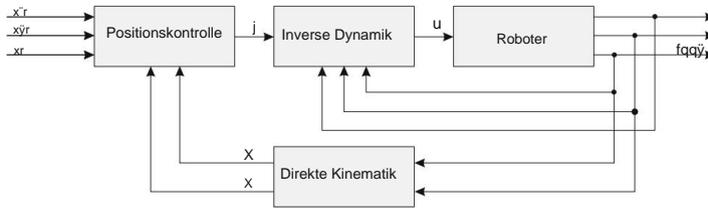


Abb. 10.14 Robotersteuerung basierend auf inverser Dynamik in externen Koordinaten einschließlich der Kontaktkraft

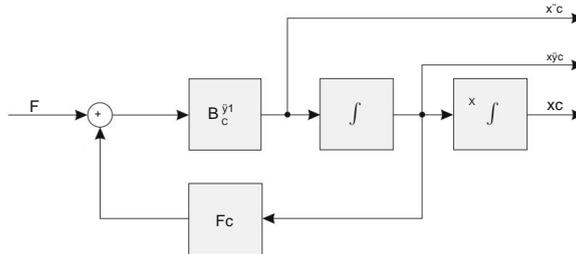


Abb. 10.15 Kraftsteuerung übersetzt in die Steuerung der Pose des Roboter-Endeffektors

Kraft \tilde{f} . Für mehr Freiheitsgrade können wir die folgende Differentialgleichung schreiben, die die Bewegung des Objekts beschreibt

$$\tilde{f} = B_c \ddot{x}_c + F_c \dot{x}_c. \quad (10.48)$$

Die Matrizen B_c und F_c bestimmen die Bewegung des Objekts unter dem Einfluss der Kraft \tilde{f} . Aus Gl. (10.48) kann die Beschleunigung des virtuellen Objekts berechnet werden

$$\ddot{x}_c = B_c^{-1} \tilde{f} - \dot{y}_c F_c \dot{x}_c. \quad (10.49)$$

Durch Integration der Gl. (10.49) werden die Geschwindigkeiten und die Lage des Objekts berechnet, wie in Abb. 10.15 dargestellt. Auf diese Weise werden aus dem Kraftfehler die Referenzlage x_c , die Referenzgeschwindigkeit \dot{x}_c und die Referenzbeschleunigung \ddot{x}_c ermittelt. Die berechneten Variablen sind Eingaben für das Steuerungssystem, dargestellt in Abb. 10.14. Auf diese Weise wurde die Kraftsteuerung in die bereits bekannte Robotersteuerung in externen K

Um gleichzeitig auch die Pose des Roboter-Endeffektors zu steuern, ist eine parallele Komposition enthalten. Bei der parallelen Komposition wird davon ausgegangen, dass die Referenzkontrollvariablen durch Summieren der Referenzen für die Kraftsteuerung (x_c , \dot{x}_c , \ddot{x}_c) und Referenzen für die Posensteuerung (x_d , \dot{x}_d , \ddot{x}_d) erhalten werden. Die parallele Zusammensetzung wird durch Gleichungen definiert

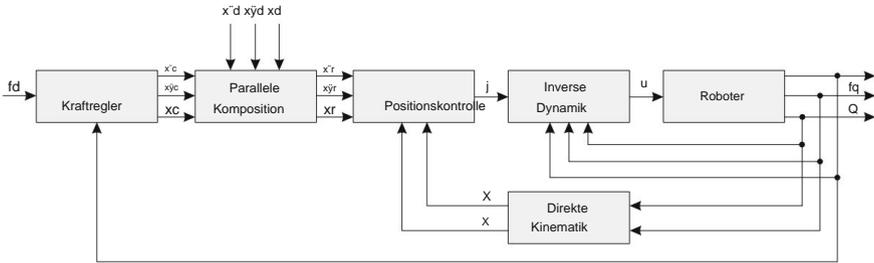


Abb. 10.16 Direkte Kraftsteuerung in den externen Koordinaten

$$\begin{aligned}
 x_r &= x_d + x_c \\
 \dot{x}_r &= \dot{x}_d + \dot{x}_c \\
 \ddot{x}_r &= \ddot{x}_d + \ddot{x}_c
 \end{aligned}
 \tag{10.50}$$

Das Steuerungssystem, das die Kontaktkraftsteuerung, die parallele Zusammensetzung und die Steuerung des Roboters auf der Grundlage der inversen Dynamik in externen Koordinaten umfasst, ist in Abb. 10.16 dargestellt. Die Kraftsteuerung erfolgt durch Auswahl

$$\begin{aligned}
 X\Gamma &= XC \\
 \dot{x}_r &= \\
 \ddot{x}_c \ddot{x}_r &= \ddot{x}_c
 \end{aligned}
 \tag{10.51}$$

Das beschriebene Steuerungsverfahren ermöglicht die Steuerung von Kräften. Es ermöglicht jedoch keine unabhängige Steuerung der Pose des Roboter-Endeffektors, da diese durch den Fehler im Kraftsignal bestimmt wird.

Kapitel 11

Roboterumgebung



In diesem Kapitel werden Roboterumgebungen veranschaulicht, am Beispiel von Produktmontageprozessen, bei denen Roboter Teil einer Produktionslinie oder als völlig unabhängige Einheiten sind. Das Beispiel lässt sich problemlos auch auf andere Aufgaben übertragen, beispielsweise Produktinspektion und -tests, Schweißen, Lackieren, Pick-and-Place-Vorgänge usw.

Tatsächlich stellen Roboter eine ideale Lösung für viele Arbeitssicherheits- und Gesundheitsprobleme dar, vor allem weil sie in der Lage sind, schwierige und ermüdende Aufgaben in einer gefährlichen Umgebung auszuführen. Schweiß- und Lackierroboter ermöglichen es menschlichen Arbeitern, giftige Dämpfe und Dämpfe zu vermeiden. Roboter beladen auch Kraftpressen, die in der Vergangenheit häufig zu Verletzungen von Arbeitern führten. Roboter arbeiten in Gießereien und radioaktiven Umgebungen. Mit der zunehmenden Anzahl von Robotern in industriellen Prozessen steigt jedoch auch die Gefahr, die von den Robotern selbst ausgeht. Daher ist die Berücksichtigung der Sicherheit bei der Konstruktion einer Roboterarbeitszelle von größter Bedeutung.

11.1 Robotersicherheit

Industrieroboter sind starke Geräte, die sich schnell in ihrem Arbeitsbereich bewegen. In den meisten Fällen kommt es erst dann zu einem Unfall, wenn ein menschlicher Arbeiter den Arbeitsbereich des Roboters betritt. Eine Person betritt die Roboterumgebung entweder versehentlich oder sogar unwissentlich oder mit dem Ziel, den Roboter neu zu programmieren oder zu warten. Für einen menschlichen Bediener ist es oft schwierig zu beurteilen, was der nächste Schritt des Roboters sein wird. Besonders gefährlich sind unerwartete Roboterbewegungen, die entweder auf einen Roboteranfall oder einen Programmierfehler zurückzuführen sind. Viele staatliche Organisationen und große Unternehmen haben zusammen mit Roboterherstellern Sicherheitsstandards entwickelt. Die Ansätze zur Gewährleistung einer sicheren Zusammenarbeit zwischen menschlichen Arbeitskräften und Industrierobotern lassen sich in drei Hauptgruppen einteilen: (1) Robotersicherheitsfunktionen, (2) Schutzmaßnahmen am Roboterarbeitsplatz und (3) Schulung und Überwachung des Personals.

Heutige Roboter verfügen über weitgehend integrierte Sicherheitsfunktionen für alle drei Betriebsarten: normale Arbeit, Programmierung und Wartung. Fehlervermeidungsfunktionen erhöhen die Zuverlässigkeit und Sicherheit des Roboters. Eine solche Funktion verhindert beispielsweise, dass der Roboter in die Presse greift, bevor diese geöffnet ist. Die in die Robotersteuereinheit integrierten Sicherheitsfunktionen ermöglichen normalerweise eine Synchronisierung zwischen dem Roboter und anderen Maschinen in der Roboterumgebung. Die Überprüfung der Signale, die anzeigen, wann ein Gerät bereit ist, eine aktive Rolle in der Roboterzelle einzunehmen, muss Teil einer sicheren Roboterprogrammierung sein. Bei der Zustandskontrolle von Maschinen im Roboterarbeitsbereich spielt der Einsatz zuverlässiger Sensoren eine wichtige Rolle. Wichtige Sicherheitsmerkmale jedes Robotersystems sind auch Software und elektrische Stopps.

Beim Programmieren oder Teachen eines Roboters muss sich der menschliche Bediener im Arbeitsbereich des Roboters befinden. In der Programmierphase muss die Geschwindigkeit der Roboterbewegungen deutlich geringer sein als bei normaler Arbeit. Die Geschwindigkeit des Roboters muss auf einen solchen Wert reduziert werden, dass der menschliche Bediener unerwartete Bewegungen des Roboters vermeiden kann. Die empfohlene maximale Geschwindigkeit des Roboters beträgt 0,25 m/s, wenn sich ein menschlicher Arbeiter im Arbeitsbereich befindet.

Das Programmiergerät kann eine entscheidende Komponente für den sicheren Roboterbetrieb sein. Programmierfehler beim Teachen eines Roboters führen häufig zu unerwarteten Roboterbewegungen. Das Design eines Programmiergeräts kann einen erheblichen Einfluss auf den sicheren Betrieb haben. Die Verwendung einer Joystick-Steuerung erwies sich als sicherer als die Verwendung von Steuerdrucktasten. Auch die Größe von Notruftasten hat einen wichtigen Einfluss auf die Reaktionszeit des menschlichen Bedieners.

Spezielle Sicherheitsfunktionen erleichtern die sichere Wartung des Roboters. Eine solche Funktion ist beispielsweise die Möglichkeit, die Steuerung einzuschalten, während der Roboterarm nicht mit Strom versorgt wird. Eine weitere Funktion ermöglicht die passive manuelle Bewegung der Robotersegmente, während die Roboteraktuatoren ausgeschaltet sind. Einige Roboterfunktionen bewirken, dass der Roboter so schnell wie möglich anhält, während andere es dem Steuerungssystem ermöglichen, den aktuellen Befehl auszuführen und anschließend anzuhalten.

Die meisten Roboterunfälle ereignen sich, wenn Personen absichtlich oder unvorsichtig den Arbeitsbereich des Roboters betreten. Die Absicherung des Roboterarbeitsplatzes verhindert ein solches Eindringen in die Roboterzelle. Es gibt drei Hauptansätze für den Schutz des Roboterarbeitsplatzes: (1) Barrieren und Zäune, (2) Anwesenheitserkennung und (3) Warnschilder, Signale und Licht.

Am häufigsten werden Metallbarrieren oder Zäune verwendet, um zu verhindern, dass unbefugte Arbeiter den Arbeitsbereich des Roboters betreten. Die Farbe des Zauns spielt eine wichtige Rolle, um nicht informiertes Personal wirksam zu warnen. Die Zäune stellen auch einen ausreichenden Schutz vor diversen Fahrzeugen dar, die zum Materialtransport in der Produktionshalle eingesetzt werden. Außerdem muss für eine sichere Öffnung der Tore gesorgt werden, die den Zutritt zum umzäunten Bereich ermöglichen. Ein menschlicher Bediener kann nur nach Abschalten des Robotersystems über ein Bedienfeld außerhalb der Schranken eintreten. Gut konzipierte Schutzbarrieren können auch Umstehende vor Gegenständen schützen, die aus dem Griff des Roboters herausfliegen.

Wichtige Sicherheitsmaßnahmen sind die Geräte, die die Anwesenheit einer Person im Arbeitsbereich des Roboters erkennen. Dies können druckempfindliche Fußmatten, Lichtvorhänge, Endeffektorsensoren, verschiedene Ultraschall-, kapazitive, Infrarot- oder Mikrowellensensoren innerhalb der Roboterzelle und Computer Vision sein. Instrumentierte Fußmatten oder Lichtvorhänge können das Eindringen einer Person in den Arbeitsbereich des Roboters erkennen. In einem solchen

In diesem Fall wird ein Warnsignal ausgelöst und der normale Roboterbetrieb kann gestoppt werden. Die Endeffektorsensoren erkennen unerwartete Kollisionen mit Objekten in der Roboterumgebung und lösen einen Notstopp aus. Berührungslose Sensoren und Computer Vision erkennen verschiedene Eingriffe in den Arbeitsbereich des Roboters.

Warnschilder, Signale und Lichter können den sicheren Betrieb von Roboterzellen erheblich erhöhen. Diese Warnschilder machen den Bediener auf das Vorliegen einer Gefahrensituation aufmerksam. Bedienungsanleitungen und eine entsprechende Schulung sind ebenfalls wichtig für den effektiven Einsatz von Warnschildern. Solche Schilder sind bei Personen, die unbeabsichtigt den Arbeitsbereich des Roboters betreten, wirksamer als bei Bedienern, die mit der Bedienung der Roboterzelle vertraut sind. Erfahrene Bediener vernachlässigen häufig die Warnungen und betreten absichtlich den Arbeitsbereich des Roboters, ohne den Roboter auszuschalten, um etwas Zeit zu sparen. Solche Bewegungen sind häufig Unfallursache. Auch Fehlalarme können die Wirksamkeit von Warnungen beeinträchtigen.

Die Auswahl qualifizierter Arbeitskräfte, Sicherheitsschulungen und eine ordnungsgemäße Aufsicht sind Voraussetzungen für sicheres Arbeiten mit Robotern. Besonders kritische Momente sind das An- und Abfahren einer Roboterzelle. Ebenso kann die Wartung und Programmierung von Robotern gefährlich sein. Einige Roboteranwendungen (z. B. Schweißen) beinhalten spezifische Gefahrensituationen, die den Arbeitern gut bekannt sein müssen. Wer im Roboterumfeld beschäftigt ist, muss für seinen Beruf sowohl körperliche als auch geistige Voraussetzungen erfüllen. Die Auswahl geeigneter Arbeitskräfte ist ein wichtiger erster Schritt. Der zweite, ebenso wichtige Schritt ist eine umfassende Sicherheitsschulung. Eine zufriedenstellende Sicherheit wird nur durch ständige Aufsicht der Mitarbeiter erreicht. Zusatzschulungen sind ein wichtiger Bestandteil beim Einsatz von Industrierobotern. In den Schulungen müssen die Arbeitnehmer mit den möglichen Gefahren und deren Schwere vertraut gemacht werden. Sie müssen lernen, Gefahrensituationen zu erkennen und zu vermeiden. Häufige Fehler, die Unfallursachen sind, sollten ausführlich erläutert werden. Solche Schulungen werden in der Regel mit Hilfe von Roboterherstellern vorbereitet.

Es wird erwartet, dass künftige Roboter nicht mehr hinter Schutzvorrichtungen mit verschlossenen Türen oder Lichtschranken arbeiten werden. Stattdessen werden sie eng mit Menschen zusammenarbeiten, was zu der grundsätzlichen Frage führt, wie eine sichere physische Mensch-Roboter-Interaktion gewährleistet werden kann. Die größten Fortschritte werden bei der Entwicklung leichter, flexibler Robotersegmente, nachgiebiger Gelenke, neuartiger Aktuatoren und fortschrittlicher

Die Roboterinstallation kann als einzelne Roboterzelle oder als Teil einer größeren industriellen Produktionslinie erfolgen. Industrieroboter sind positionsgesteuert und oft ohne Sensoren zur Erfassung ihrer Umgebung. Aus diesem Grund müssen die Roboter im Falle einer unsachgemäßen Aktivität des Roboters oder seiner Peripherieteile von der menschlichen Umgebung isoliert werden, um Verletzungen von Personen oder Kollisionen mit anderen Geräten in der Roboterarbeitszelle zu verhindern. Das Sicherheitsrisiko für jede einzelne Roboterzelle muss definiert werden, damit entsprechende Vorkehrungen getroffen werden können. Unsachgemäßes Roboterverhalten kann auf einen Fehler des Robotersystems oder menschliches Versagen zurückzuführen sein, wie zum Beispiel:

- unvorhersehbares Roboterverhalten aufgrund eines Fehlers im Steuerungssystem,
 - Kabelverbindungsfehler aufgrund der Roboterbewegung, •
- Datenübertragungsfehler, der zu unvorhersehbaren Roboterbewegungen führt, • Fehler am Roboterwerkzeug, z. B. Schweißpistole,

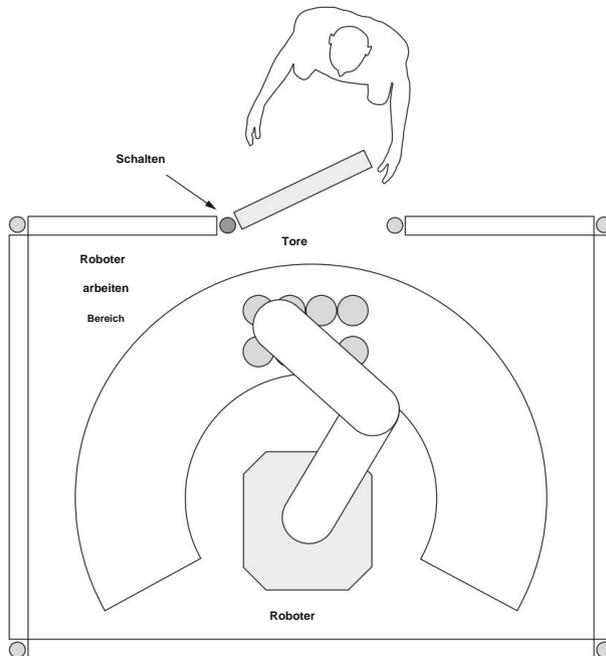


Abb. 11.1 Ebene 1: mechanischer Roboterzellschutz

• Softwarefehler, •

verschlossene mechanische Komponenten des Roboters.

Die potenziellen Gefahren von Systemfehlern, die sich aus diesen Fehlern ergeben, können unterteilt werden in drei Kategorien:

- Bei Kollisionsgefahr besteht die Möglichkeit, dass der sich bewegende Roboter oder das daran befestigte Werkzeug beschädigt wird
Roboter schlägt den Bediener.
- Einklemmgefahr liegt vor, wenn der Roboter während der Bewegung in der Nähe von Objekten in der Roboterzelle (z. B. Transportmechanismen) den Bediener einklemmt.
- Andere Gefahren, die für jede Roboteranwendung spezifisch sind, wie z. B. die Gefahr eines Stromschlags, des Aufpralls eines Schweißlichtbogens, Verbrennungen, giftiger Substanzen, Strahlung, übermäßiger Geräuschpegel.

Aus all diesen Gründen können die Sicherheitsanforderungen an Roboter in drei Ebenen unterteilt werden.

Level 1 ist das Schutzniveau der gesamten Roboterzelle. Dies wird normalerweise durch physischen Schutz durch eine Kombination aus mechanischen Zäunen, Geländern und Toren erreicht (Abb. 11.1). Neben dem physischen Schutz kann auch ein Anwesenheitssensor (z. B. Laservorhänge) installiert werden.

Stufe 2 umfasst eine Schutzstufe, während sich ein Bediener im Arbeitsbereich des Roboters aufhält. Normalerweise erfolgt der Schutz durch Anwesenheitssensoren. Im Gegensatz zur vorherigen Ebene, die hauptsächlich auf mechanischem Schutz basiert, basiert Ebene 2 auf der Wahrnehmung der Anwesenheit des Bedieners (Abb. 11.2).

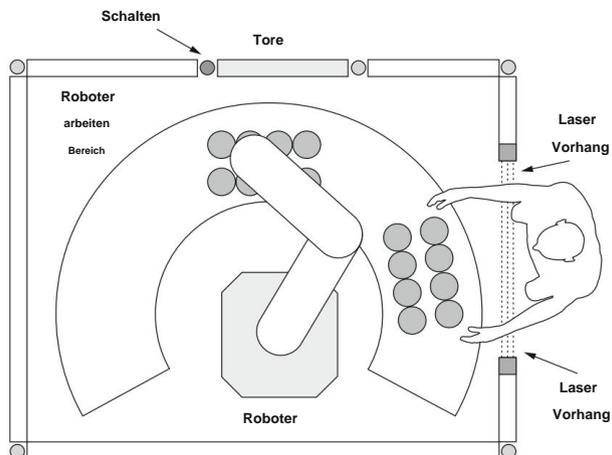


Abb. 11.2 Level 2: optoelektrischer Roboterzellenschutz

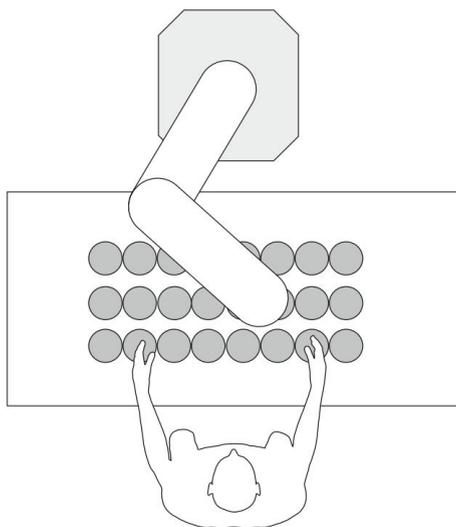


Abb. 11.3 Ebene 3: Zusammenarbeit von Mensch und Roboter

Stufe 3 ist die Schutzstufe, bei der Menschen mit dem Roboter in Kontakt kommen, der als kollaborativer Roboter bezeichnet wird. Die Sicherheit auf dieser Ebene erfolgt durch die Erkennung der Anwesenheit eines Menschen oder von Hindernissen in der Nähe des Roboters oder wenn der Roboter und der Mensch zusammenarbeiten (Abb. 11.3). In Risikosituationen muss das Robotersystem langsamer werden oder anhalten. Diese Systeme umfassen Sensoren zur Personenverfolgung, verschiedene Kraft- und Drehmomentsensoren sowie Kontakt- oder Berührungssensoren. Kollaborative Roboter werden in Kap. ausführlicher beschrieben. 12.

11.2 Roboterperipherie in Montageprozessen

Die in der Industrie installierten Robotersysteme sind meist Teil größerer, dedizierter Produktionslinien. Die Produktionslinien werden für die Massenproduktion von Teilen verwendet, bei denen mehrere Bearbeitungsvorgänge erforderlich sind. Die Produktionslinie ist in Arbeitsstationen unterteilt, an denen menschliche Arbeiter, spezielle Maschinen oder Roboter notwendige Aufgaben ausführen. Es können auch andere Peripheriegeräte integriert werden, um die Kapazitäten der Produktionslinie zu erhöhen. Die richtig ausgewählte Peripherie erhöht zudem die Zuverlässigkeit, Flexibilität und Effizienz des Systems.

11.2.1 Konfigurationen der Montage-Produktionslinie

Montagelinien in der Industrie bestehen aus Förderbändern, auf Förderbändern transportierten Paletten, Bildverarbeitungssystemen, Pneumatikzylindern, verschiedenen Sensoren und Robotern oder Manipulatoren. Die Paletten bieten die Möglichkeit, einzelne gefertigte Teile, die den Automatisierungsprozess durchlaufen, zu indexieren, zu lokalisieren und zu verfolgen. Die Roboter bieten Flexibilität und können in jede Konfiguration der Produktionslinie integriert werden. Die gebräuchlichsten Konfigurationen von Montagelinien mit Roboterunterstützung sind:

- Inline (direkt, L-förmig, U-förmig, kreisförmig, rechteckig), • rotierend,
- Hybrid.

In Abb. 11.4 ist ein Beispiel einer kreisförmigen Inline-Produktionslinie dargestellt.

Liniendarbeitsplätze werden von Menschen, speziellen Maschinen und Robotern bedient. Teile zur Montage

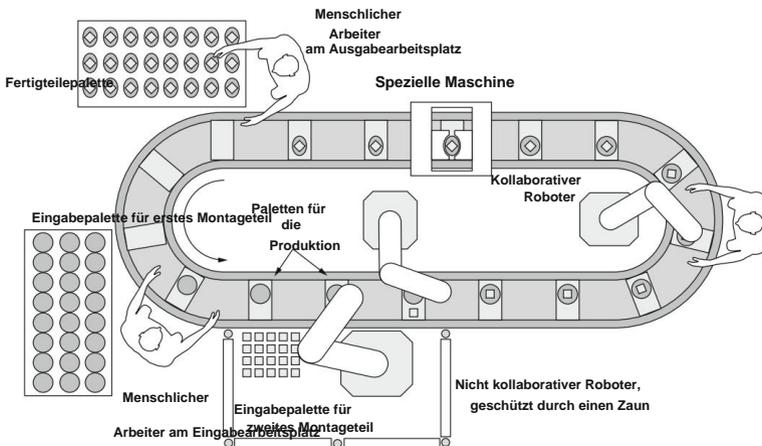


Abb. 11.4 Beispiel einer kreisförmigen Inline-Montageproduktionslinie mit Mensch-, Maschinen- und Roboterarbeitsplätzen

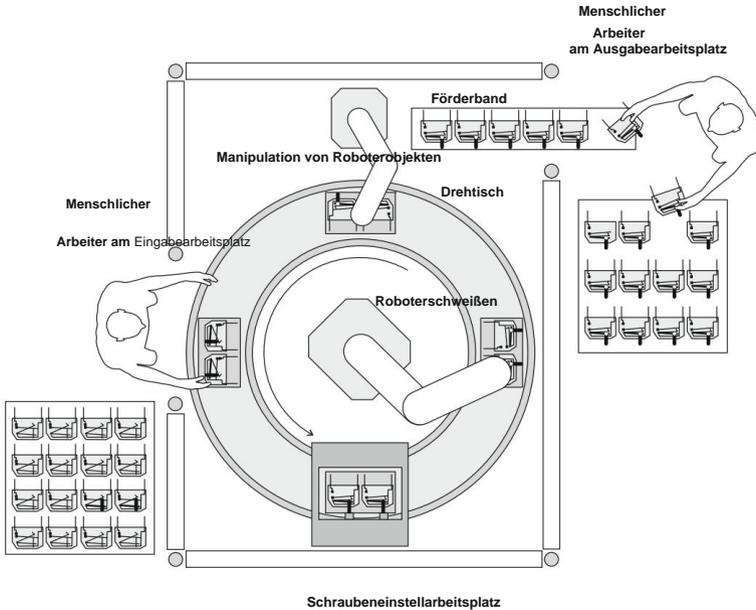


Abb. 11.5 Beispiel einer Drehtisch-Produktionslinie mit Mensch-, Maschinen- und Roboterarbeitsplätzen

Die Werkstücke werden von Hand oder vom Roboter manipuliert und über Paletten entlang eines Förderbands zwischen den Arbeitsstationen transportiert. Der Abstand zwischen den Paletten ist nicht unbedingt konstant und ihre Position wird durch Ortungssensoren, meist kapazitive oder induktive Anwesenheitssensoren, überwacht. Diese Sensoren sind notwendig, um den Robotern oder einer speziellen Maschine zu signalisieren, dass sich die Palette in der richtigen Position befindet und der Arbeitsstationsvorgang durchgeführt werden kann. Die Zykluszeit für den Transport des Teils von einer Arbeitsstation zur nächsten ist in der Regel konstant, sodass die Arbeitsstationen synchron sind. In bestimmten Fällen integrieren die Entwickler der Produktionslinie Teile, um die Paletten zu puffern, wodurch die Produktionslinie asynchron wird. Die Notwendigkeit eines Puffers entsteht in Fällen, in denen einige Arbeitsplätze variable Zykluszeiten haben; Mit einem Puffer wird die Gesamtzykluszeit der Produktionslinie nicht beeinflusst.

Eine weitere sehr häufige Konfiguration einer Montagelinie ist ein Dreh- oder Drehtisch (Abb. 11.5), der normalerweise durch Elektromotoren angetrieben wird. Die Geschwindigkeit und Wiederholgenauigkeit der Positionierung sind hoch. Der Drehtisch wird oft als Wählmaschine oder Indexiermaschine bezeichnet. Der Vorteil der Rotationskonfiguration besteht darin, dass sie weniger Stellfläche benötigt und häufig kostengünstiger ist als andere Produktionslinienkonfigurationen. Der Rundtisch führt stets einen synchronen Teiletransfer zwischen den Arbeitsstationen mit konstanter Taktzeit durch.

Wie im vorherigen Beispiel kann diese Konfiguration auch von Menschen, Robotern oder dedizierten Maschinen bedient werden. Der Drehtisch hat eine kreisförmige Form, um die sich die Paletten oder Teilehalter bewegen und der Reihe nach Teile in jede Hand transportieren

oder automatisierter Arbeitsplatz, an dem Produktionsvorgänge durchgeführt werden. Der Drehtisch kann in mehrere Arbeitsstationen (mindestens 2) aufgeteilt werden, sodass der Drehwinkel reduziert wird. Drehtische mit mehr als 2 Arbeitsstationen, z. B. 4, 5, 6 Arbeitsstationen. Die Größe des Drehtisches wird durch die Teilegröße, die Gerätegröße und die Anzahl der Arbeitsplätze der rotierenden Produktionslinie bestimmt. Es sind auch Plattenspieler mit Closed-Loop-Steuerung erhältlich.

Normalerweise wird eine Kombination der oben genannten Konfigurationen installiert und als hybride Produktionslinienkonfiguration bezeichnet. Mehrere Faktoren bestimmen die Gesamtkonfiguration der Produktionslinie, wie zum Beispiel:

- benötigter Platz für die Produktionslinie,
- Kosten für die Installation der Produktionslinie, •
- Zykluszeit der Produktionslinie.

11.3 Zuführgeräte

Die Aufgabe der Zuführeinrichtungen besteht darin, Teile oder Baugruppen so an den Roboter oder die entsprechende Maschine zu bringen, dass die Teilelage bekannt ist. In den Roboterzellen ohne Robotersicht ist der zuverlässige Betrieb der Zuführgeräte von größter Bedeutung. Die Position eines Teils muss genau sein, da sich der Endeffektor des Roboters immer auf derselben Flugbahn bewegt und erwartet wird, dass sich das Teil immer an der gleichen Stelle befindet.

Die Anforderungen an die Roboterzuführvorrichtungen sind deutlich strenger als bei der manuellen Montage, es sei denn, die Roboterzelle ist mit einem Roboter-Vision-System ausgestattet. Die Roboterzuführvorrichtungen dürfen die Teile nicht verformen, müssen zuverlässig arbeiten, die Teile genau positionieren, mit ausreichender Geschwindigkeit arbeiten, eine minimale Ladezeit erfordern und eine ausreichende Anzahl von Teilen enthalten.

Die Zuführeinrichtung darf keine Schäden an den gehandhabten Teilen verursachen, da beschädigte Teile anschließend vom Roboter in Baugruppen eingefügt werden, die nicht ordnungsgemäß funktionieren. Die Kosten für solche beschädigten Baugruppen sind höher als die Kosten für ein zuverlässigeres Zuführgerät. Die Zuführeinrichtung muss alle Teile, deren Abmessungen innerhalb der Toleranzgrenzen liegen, zuverlässig verarbeiten. Es muss außerdem schnell genug sein, um die Anforderungen der gesamten Zykluszeit der Produktionslinie zu erfüllen, und darf den Betrieb niemals verlangsamen. Darüber hinaus soll die Zuführeinrichtung möglichst wenig Zeit für die Bestückung mit den Teilen benötigen. Es ist wünschenswerter, eine große Menge an Teilen auf einmal in die Zuführvorrichtung zu füllen, als sie einzeln manuell einzulegen. Die Zuführgeräte sollten möglichst viele Teile enthalten. Auf diese Weise wird die Anzahl der pro Tag erforderlichen Ladungen reduziert.

Die einfachsten Zuführgeräte sind Paletten und Vorrichtungen; Ein alltägliches Beispiel sind die Karton- oder Kunststoffpaletten, die für Eier verwendet werden. Die Paletten lagern die Teile und bestimmen dabei deren Position und teilweise auch Ausrichtung. Im Idealfall wird dieselbe Palette für den Versand der Teile vom Lieferanten und für die spätere Verwendung in der Roboterzelle des Verbrauchers verwendet. Die Beladung der Paletten erfolgt entweder automatisch maschinell oder manuell. Zerbrechliche Teile, flexible Gegenstände oder Teile mit ungewöhnlichen Formen müssen

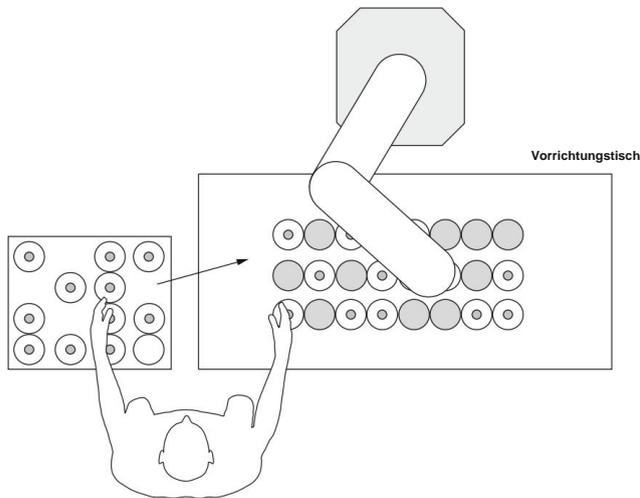


Abb. 11.6 Gleichzeitiges Laden einer Vorrichtungstabelle

Die Paletten stellen den schwächsten Punkt der Palettierung dar. Ein weiterer Nachteil von Paletten ist ihre relativ große Oberfläche, die im Arbeitsbereich des Roboters viel Platz einnimmt.

Die einfachste Möglichkeit, Teile in die Roboterzelle zu bringen, stellt ein Vorrichtungstisch dar. Der menschliche Bediener entnimmt ein Teil aus einem Behälter, in dem die Teile unsortiert sind, und legt es auf den Vorrichtungstisch im Roboterarbeitsraum (Abb. 11.6). Der Spanntisch muss spezielle Nuten enthalten, die eine zuverlässige Positionierung eines Teils im Roboterarbeitsraum gewährleisten. Ein solcher Spanntisch wird häufig beim Schweißen eingesetzt, wo die Bauteile vor dem Roboterschweißen ebenfalls auf dem Tisch aufgespannt werden müssen. Der Zeitaufwand für das Roboterschweißen ist erheblich länger als für das Be- und Entladen, was den Einsatz eines Vorrichtungstisches rechtfertigen kann.

Die Paletten können vorab an anderer Stelle beladen und anschließend in die Roboterzelle gebracht werden (Abb. 11.7). Dadurch wird eine lange Wartezeit des Roboters vermieden, während der menschliche Bediener die Paletten belädt. Der menschliche Arbeiter muss die Palette lediglich in den Arbeitsbereich des Roboters bringen und sie mithilfe spezieller Stifte im Arbeitstisch richtig positionieren. Es ist wichtig, dass die Palette eine ausreichende Anzahl an Teilen enthält, um einen kontinuierlichen Roboterbetrieb zu ermöglichen. Der Austausch der Paletten im Roboterarbeitsraum stellt ein Sicherheitsproblem dar, da der Bediener den Roboter abschalten muss oder die Roboterzelle mit anderen Sicherheitslösungen (z. B. Drehtisch oder kollaborativer Roboter) ausgestattet sein muss.

Auf einem Drehtisch kann eine größere Anzahl Paletten platziert werden (Abb. 11.8). Der Drehtisch ermöglicht die Beladung der Paletten auf einer Seite, während die Robotertätigkeiten auf der anderen Seite des Drehtisches stattfinden. Dadurch wird die Inaktivität der Roboterzelle erheblich reduziert und der menschliche Bediener vor den Bewegungen des Roboters geschützt.

Im Allgemeinen werden drei Arten von Paletten verwendet: vakuumgeformte oder spritzgegossene Kunststoff- und Metallpaletten. Da die Kosten für vakuumgeformte Paletten niedrig sind,

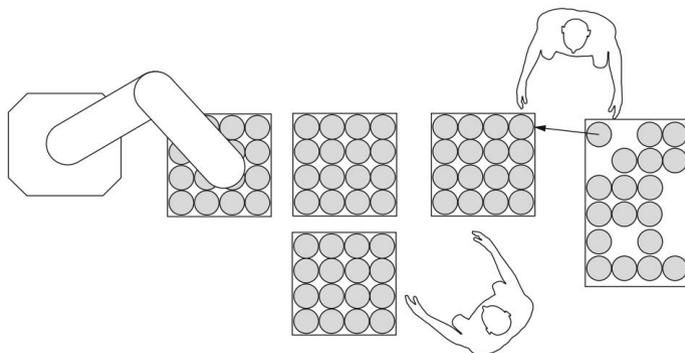


Abb. 11.7 Beladung der Paletten im Vorfeld

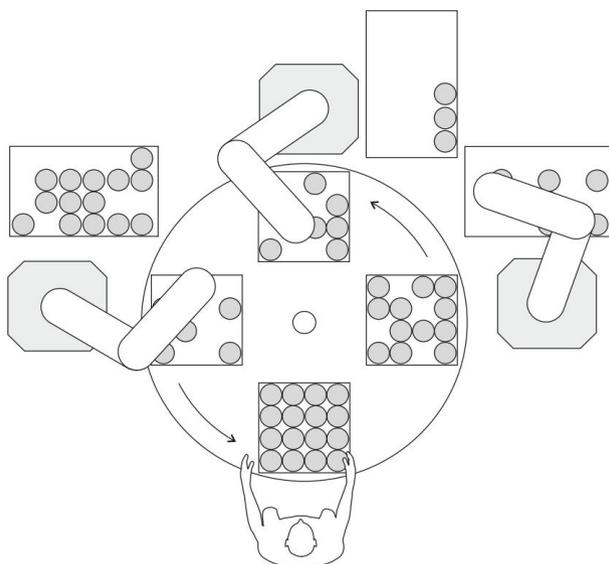


Abb. 11.8 Rundtisch mit Palettentischen

Sie dienen sowohl der Verpackung und dem Versand der Teile als auch dem Einsatz in der Roboterzelle. In alle Paletten müssen Referenzlöcher eingebaut werden, die zu den Stiften im Arbeitstisch passen, um eine einfache und schnelle Positionierung zu ermöglichen. Da die vakuumgeformten Paletten kostengünstig sind, ist es nicht schwer zu verstehen, dass sie nicht die genauesten, zuverlässigsten und langlebigsten sind. Sie bestehen aus einer dünnen Kunststoffolie, die erhitzt und über einer Form vakuumgeformt wird. Die Ungenauigkeit der Palette ist eine Folge ihrer geringen Steifigkeit. Spritzgegossene Kunststoffpaletten werden verwendet, wenn präzisere und haltbarere Paletten erforderlich sind. Die Herstellung der Form ist ziemlich teuer, während die Herstellungskosten einer einzelnen Palette nicht hoch sind. Wir müssen bedenken, dass die meisten Val

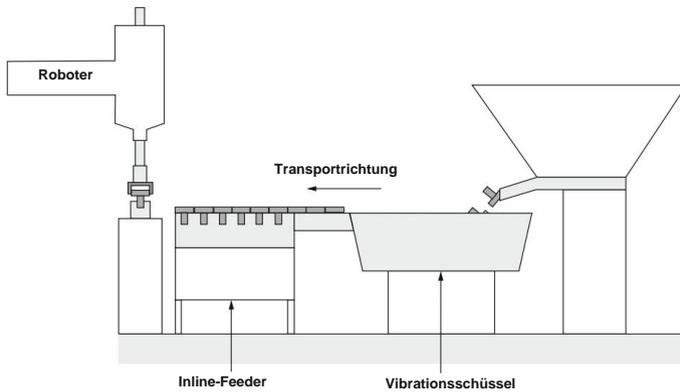


Abb. 11.9 Vibrationswendelförderer

die nicht brennbar sind. Sie werden durch verschiedene Bearbeitungsansätze hergestellt. Die Metallpaletten sind die zuverlässigsten und langlebigsten, während ihre Kosten höher sind als die der Kunststoffpaletten. Sie werden daher nur innerhalb des RoboterMontageprozesses eingesetzt.

Teilezuführungen stellen eine weitere interessante Familie von Zuführungsgeräten dar, die nicht nur zum Lagern von Teilen, sondern auch zum Positionieren und sogar Ausrichten dieser Teile in die für das Roboter greifen geeignete Pose verwendet werden. Am gebräuchlichsten sind Vibrationswendelförderer (Abb. 11.9). Dabei werden die Teile ungeordnet in die Schüssel geladen. Die Vibration der Schüssel und des Inline-Förderers wird durch einen Elektromagneten erzeugt, und die richtige Vibration wird durch die Befestigung der Vibrationsförderer an einer großen Masse, normalerweise einem dicken Stahltisch, erreicht. Durch die Vibrationen wandern die Teile aus der Schüssel. Speziell geformte spiralförmige Zäune zwingen sie in die erforderliche Ausrichtung. Derselbe Wendelförderer kann für verschiedene Teile verwendet werden, jedoch nicht gleichzeitig. Ein weiterer Vorteil besteht darin, dass die Schüssel eine große Anzahl von Teilen aufnehmen kann und dabei nur eine kleine Fläche im Roboterarbeitsraum einnimmt. Für Teile wie weiche Gummigegegenstände oder Federn sind Futterschalen nicht geeignet. Ein weiterer Nachteil sind mögliche Beschränkungen.

Ein einfacher Magazinzuführer besteht aus einem Rohr, in dem die Teile gespeichert sind, und der pneumatisch oder elektrisch betätigten Schiebeplatte, die die Teile einzeln aus dem Magazin entnimmt (Abb. 11.10). Das Magazin wird manuell beladen, so dass die Ausrichtung der Teile bekannt ist. Die Schwerkraft drückt die Teile in die Gleitplatte. Die Mechanik der Schiebeplatte muss so ausgelegt sein, dass sie ein Verklemmen der Teile verhindert, während immer nur ein Teil aus der Zuführung herausgeführt wird. Die Gleitplatte muss alle Teile außer dem unteren blockieren.

Magazinzuführungen sind hervorragende Lösungen für die Handhabung integrierter Schaltkreise (Abb. 11.11). Integrierte Schaltkreise werden bereits in Röhren geliefert, die für Speise Zwecke verwendet werden können. Die Magazinzuführung für integrierte Schaltkreise besteht üblicherweise aus einem Rohr, in dem die Teile gespeichert sind, und einer Schiebeplatte, die die Teile einzeln aus dem Magazin entnimmt.

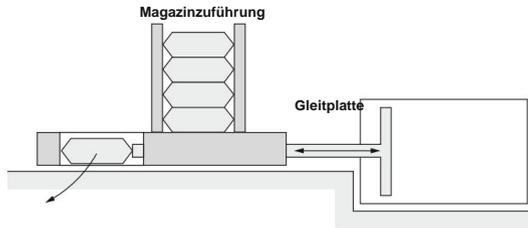


Abb. 11.10 Magazinzuführung

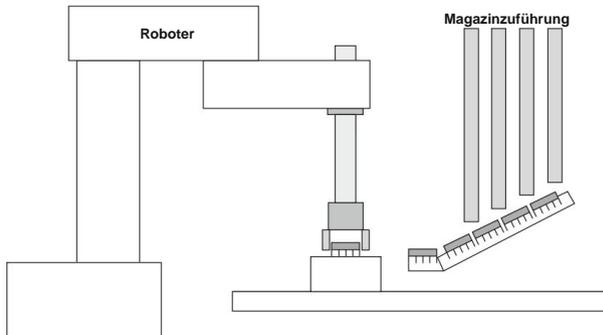


Abb. 11.11 Magazinzuführung mit integriertem Schaltkreis

mehrere Röhren. Die Rohre werden entlang einer vibrierenden Inline-Zuführung ausgerichtet. Der Hauptnachteil von Magazinzuführungen ist die manuelle Beladung. Auch für die Handhabung großer Gegenstände sind sie ungeeignet.

11.4 Förderer

Für den Transport von Teilen, Baugruppen oder Paletten zwischen den Roboterzellen werden Förderbänder eingesetzt. Der einfachste Förderer nutzt eine Kunststoff- oder Metallkette, die die Paletten entlang einer Metallführung schiebt (Abb. 11.12). Ein Elektromotor treibt die Kette mit konstanter Geschwindigkeit an. Die Antriebskraft wird durch die Reibung zwischen Kette und Palette dargestellt. Die Palette wird durch spezielle Stifte gestoppt, die von Pneumatikzylindern betätigt werden. Die Kette gleitet weiter gegen den Boden der Palette. Wenn eine weitere Palette ankommt, wird sie von der ersten gestoppt. Auf diese Weise entsteht eine Palettenschlange vor der Roboterzelle.

Die Drehung eines Förderers erfolgt durch Biegen der Metallführung. Die Vorteile des Gleitkettenförderers liegen in den geringen Kosten und der einfachen Handhabung der Paletten und der Durchführung der Wendungen. Der Nachteil besteht darin, dass keine senkrechten Schnittpunkte hergestellt werden können. Außerdem müssen die Kurven in weiten Bögen ausgeführt werden, wa

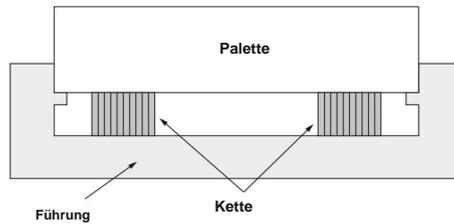


Abb. 11.12 Gleitkettenförderer (Endansicht)

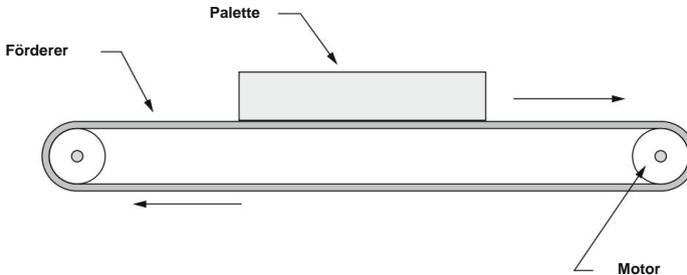


Abb. 11.13 Bandförderer

Platz in der Produktionsstätte. Der Gleitkettenförderer eignet sich am besten als Einzelschleifen-Zuführsystem.

Beim riemengetriebenen Förderer treibt der obere Teil des Riemens Paletten oder andere Gegenstände oder Materialien an (Abb. 11.13). Eine Wende oder Kreuzung erfolgt mit Hilfe einer speziellen Vorrichtung, die das Heben, Übertragen und Drehen von Paletten ermöglicht.

Ein Förderer kann auch aus Rollen bestehen, die von einer gemeinsamen Antriebswelle angetrieben werden (Abb. 11.14). Die Antriebswelle überträgt das Drehmoment über einen Antriebsriemen auf die Rollenwelle. Der Vorteil des Rollenförderers liegt in den geringen Kollisionskräften, die zwischen den Paletten oder den vom Förderer transportierten Gegenständen auftreten. Sie entstehen durch die geringe Reibung zwischen den Rollen und den Paletten. Die Drehungen werden mithilfe von Hebe- und Transfervorrichtungen durchgeführt. Die Nachteile der Rollenförderer sind hohe Kosten und geringe Beschleunigungen.

11.5 Robotergriefer und Werkzeuge

So wie Roboter manipulieren Kopien des menschlichen Arms sind, imitieren Robotergriefer die menschliche Hand. In den meisten Fällen sind Robotergriefer wesentlich einfacher als die menschliche Hand und umfassen Handgelenk und Finger, insgesamt 22 Freiheitsgrade.

Die Greifer von Industrierobotern unterscheiden sich stark voneinander, daher ist es nicht schwer zu verstehen, dass ihre Kosten von fast vernachlässigbar bis höher als die Kosten eines Roboter manipulators reichen.

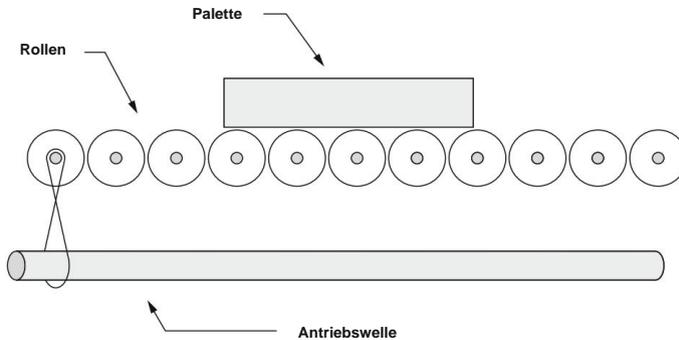


Abb. 11.14 Förderband mit Rollen

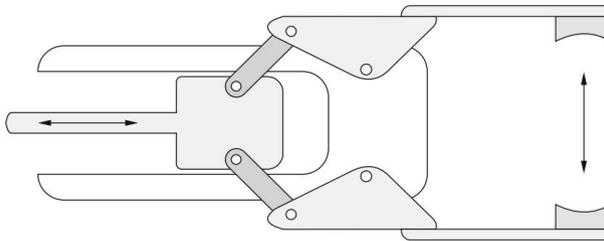


Abb. 11.15 Robotergrifer mit zwei Fingern

ulator. Obwohl viele verschiedene Robotergrifer im Handel erhältlich sind, ist es oft notwendig, einen speziellen Greifer zu entwickeln, um den Anforderungen einer bestimmten Roboteraufgabe gerecht zu werden.

Die charakteristischsten Robotergrifer sind solche mit Fingern. Sie lassen sich in Greifer mit zwei Fingern (Abb. 11.15) und Greifer mit mehreren Fingern unterteilen. Die meisten Mehrfingergrifer haben drei Finger (Abb. 11.16), um ein besseres Greifen zu erreichen. In industriellen Anwendungen treffen wir meist auf Greifer mit zwei Fingern. Die einfachsten Zweifingergrifer werden nur zwischen den beiden Zuständen offen und geschlossen gesteuert. Es sind auch zwei Fingergrifer erhältlich, bei denen der Abstand oder die Kraft zwischen den Fingern gesteuert werden kann. Mehrfingergrifer verfügen üblicherweise über drei Finger mit jeweils drei Segmenten. Ein solcher Greifer verfügt über 9 Freiheitsgrade, was mehr ist als ein Roboter manipulator. Die Kosten für solche Greifer sind hoch. Bei Mehrfingergrifern werden die Motoren oft nicht in den Fingergelenken platziert, da die Finger schwer oder nicht stark genug werden können. Stattdessen sind die Motoren alle in der Handfläche des Grifiers untergebracht, während Sehnen sie mit Riemenscheiben in den Fingergelenken verbinden. Neben Grifern mit Fingern gibt es in der Industrierobotik auch Vakuum-, Magnet-, Perforations- und Klebgrifer. Verschiedene Endeffektorwerkzeuge, die beim Spritzlackieren, Endbearbeiten oder Schweißen verwendet werden, gelten nicht als Robotergrifer.

Zweifingergrifer werden zum Greifen der Teile in einem Roboter montagesprozess eingesetzt. Ein Beispiel für einen solchen Greifer ist in Abb. 11.15 dargestellt. Verschiedene Endpunkte können

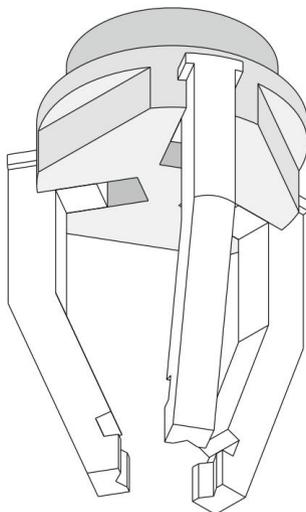


Abb. 11.16 Robotergriefer mit drei Fingern

an den Fingern befestigt werden, um den Robotergriff an die Form und Oberfläche des zu greifenden Teils oder der Baugruppe anzupassen. Bei Zweifinger-Robotergrifern kommen pneumatische, hydraulische oder elektrische Motoren zum Einsatz. Die hydraulische Betätigung ermöglicht höhere Greifkräfte und damit die Handhabung schwererer Gegenstände. Verschiedene Aufbauten von Zweifingergrifern sind in Abb. 11.17 dargestellt. Einfache kinematische Darstellungen ermöglichen die Auswahl eines passenden Greifers für die gewählte Aufgabe. Der Greifer auf der rechten Seite von Abb. 11.17 ermöglicht das parallele Fingergreifen.

In industriellen Prozessen werden Roboter manipulieren häufig zur Maschinenbeladung eingesetzt. In solchen Fällen ist der Roboter mit einem Doppelgreifer effizienter. Der Roboter kann gleichzeitig ein Rohteil in die Maschine bringen und gleichzeitig ein Fertigteil aus der Maschine entnehmen. In Abb. 11.18 ist ein Doppelgreifer dargestellt .

Zum Greifen heißer Gegenstände werden spezielle Greifer eingesetzt. Dabei sind die Aktoren weit entfernt von den Fingern platziert. Bei der Handhabung heißer Objekte wird Luftkühlung eingesetzt, während der Greifer im Rahmen des Manipulationszyklus häufig in Wasser eingetaucht wird. Von größter Bedeutung ist auch die Wahl des geeigneten Materials für die Finger.

Beim Greifen leichter und zerbrechlicher Gegenstände können Greifer mit Federfingern eingesetzt werden. Dadurch wird die maximale Greifkraft begrenzt und gleichzeitig ein einfaches Öffnen und Schließen der Finger ermöglicht. Ein Beispiel für einen einfachen Greifer mit zwei Federfingern ist in Abb. 11.19 dargestellt.

Die Form des Objekts erfordert eine sorgfältige Konstruktion eines Zweifinger-Robotergrifers. Ein sicherer Halt kann entweder durch Form- oder Kraftschluss der beiden Finger erreicht werden. Möglich ist auch die Kombination der beiden Greifmodi (Abb. 11.20).

Bei der Ausführung eines Robotergriffs mit zwei Fingern ist auch die Position der Finger in Bezug auf das Objekt wichtig. Die Greifkraft kann nur von außen aufgebracht werden

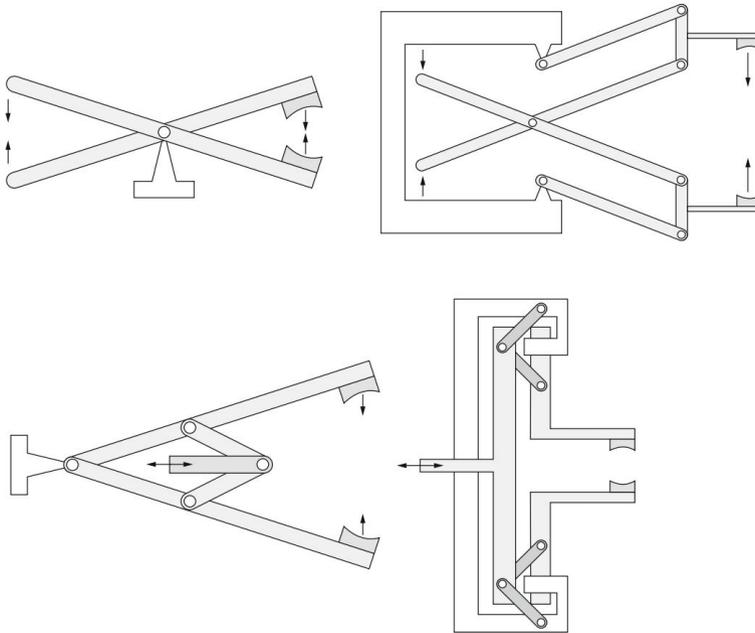


Abb. 11.17 Kinematische Darstellungen von Zweifingergreifern

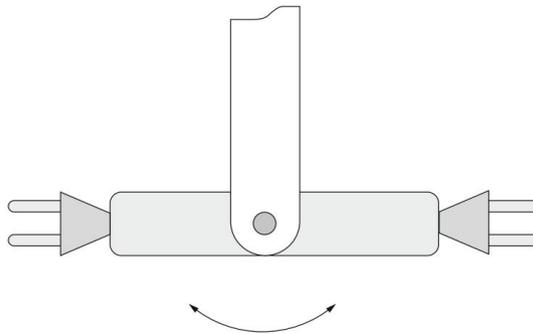


Abb. 11.18 Zweifach-Robotergrifer

Oberflächen oder nur auf den Innenflächen eines Werkstücks. Auch ein Zwischengriff ist möglich, bei dem der Gegenstand an Innen- und Außenflächen gegriffen wird (Abb. 11.21).

Unter den Robotergreifern ohne Finger sind Vakuumgreifer mit Abstand am häufigsten im Einsatz. Vakuumgreifer oder Greifer mit Unterdruck werden dort erfolgreich eingesetzt, wo die Oberfläche des gegriffenen Gegenstandes flach oder gleichmäßig gewölbt, glatt, trocken und relativ sauber ist. Die Vorteile dieser Greifer sind Zuverlässigkeit, niedrige Kosten und geringes Gewicht. Saugköpfe in verschiedenen Formen sind im Handel erhältlich.

Oftmals werden mehrere Saugköpfe zusammen verwendet und in einem passenden Muster angeordnet

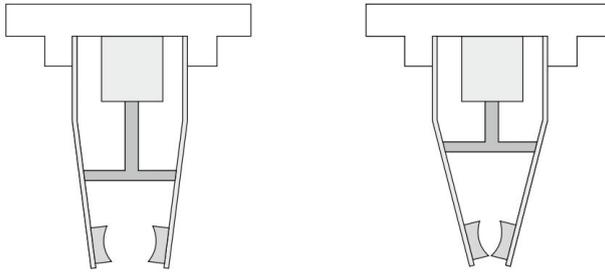


Abb. 11.19 Greifer mit Federfingern

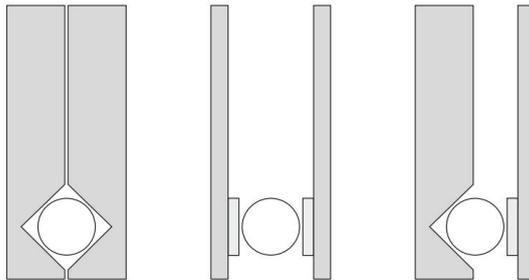


Abb. 11.20 Formschluss, Kraftschluss und kombinierter Griff



Abb. 11.21 Außen-, Innen- und Zwischengriff

Form des zu greifenden Gegenstandes. Abbildung 11.22 zeigt die Form zweier häufig verwendeter Saugköpfe. Der Kopf links eignet sich für Fälle, in denen die Oberfläche nicht ganz glatt ist. Das weiche Material des Kopfes passt sich der Form des Objekts an.

Die kleinen Nippel am Kopf, die auf der rechten Seite von Abb. 11.22 dargestellt sind, verhindern eine Beschädigung der Oberfläche des Objekts. Vakuum wird entweder mit Venturi- oder Vakuumpumpen erzeugt. Die Venturi-Pumpe benötigt mehr Leistung und erzeugt nur 70 % Vakuum. Aufgrund seiner Einfachheit und geringen Kosten wird es jedoch häufig in industriellen Prozessen eingesetzt. Vakuumpumpen liefern 90 % Vakuum und erzeugen deutlich weniger Lärm. Bei allen Greifern ist ein schnelles Greifen und Loslassen der Gegenstände erforderlich. Das Lösen sehr leichter und klebriger Objekte kann mit Vakuumgreifern kritisch sein. In diesem Fall lösen wir die Objekte mit Hilfe von Überdruck, wie in Abb. 11.23 dargestellt.

Ein weiteres Beispiel für Greifer ohne Finger sind Magnetgreifer: Sie nutzen entweder Permanentmagnete oder Elektromagnete. Die Elektromagnete werden in größerem Umfang genutzt.

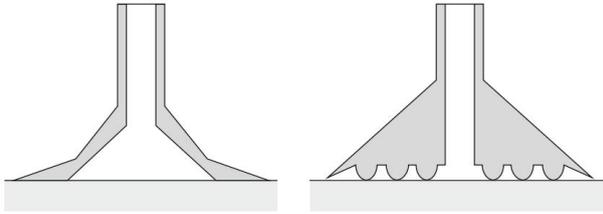


Abb. 11.22 Saugköpfe von Vakuumgreifern

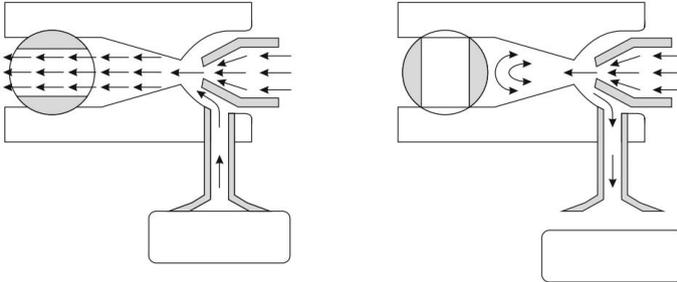


Abb. 11.23 Greifen und Loslassen eines Gegenstandes mit Hilfe von Unter- und Überdruck

Bei Permanentmagneten stellt das Lösen des Objekts eine Schwierigkeit dar. Das Problem wird durch eine speziell geplante Flugbahn des Endeffektors gelöst, bei der das Objekt durch einen Zaun im Arbeitsbereich des Roboters zurückgehalten wird. Bei Magnetgreifern werden mehrere Magnete zusammen verwendet und entsprechend der Form des Objekts in verschiedenen Mustern angeordnet. Bereits kleine Luftrisse zwischen Magnet und Objekt verringern die Magnetkraft erheblich. Die Oberflächen der zu greifenden Gegenstände müssen daher eben und sauber sein.

Perforationsgreifer gelten als spezielle Robotergreifer. Dabei werden die Gegenstände einfach vom Greifer durchstochen. Normalerweise werden diese für die Handhabung von Materialien wie Textilien oder Schaumgummi verwendet. Solche Greifer können nur in Fällen verwendet werden, in denen die Perforation keine Beschädigung des Objekts verursacht. Textilbahnen können mit großen Bürsten aus steifen Nylonhaaren oder einfach aus Klettbändern gegriffen werden.

Beim Greifen sehr leichter Teile können Klebgreifer eingesetzt werden. Die Freigabe der Teile muss durch spezielle Endpunktrajektorien des Roboters gelöst werden, bei denen das Teil mit dem Zaun im Roboterarbeitsraum kollidiert und so vom Klebgreifer entfernt wird. Für eine ausreichende Haftkraft wird durch Klebeband gesorgt, das sich während der Operation bewegen muss.

Neben Greifern können am Ende des Roboters auch andere Werkzeuge angebracht werden. Die Form und Funktion des Werkzeugs richtet sich nach der Aufgabe der Roboterzelle. Der häufigste Vorgang, den Roboter ausführen, ist das Schweißen. Für Schweißzwecke können verschiedene Ansätze verwendet werden. Darunter das am häufigsten am Roboter befestigte Werkzeug

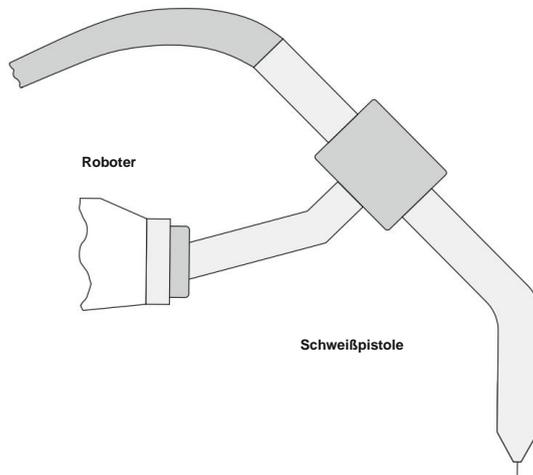


Abb. 11.24 Roboter mit am Ende befestigter Schweißpistole

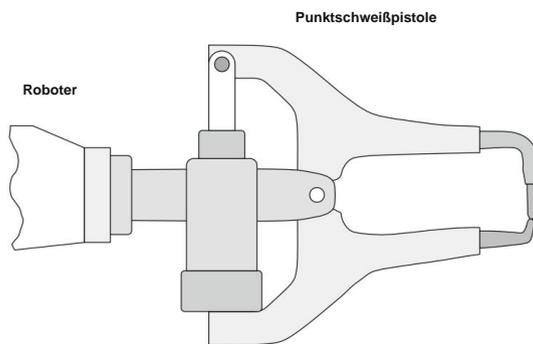


Abb. 11.25 Roboter mit am Ende befestigter Punktschweißpistole

Am Ende befindet sich eine Lichtbogenschweißpistole oder ein Brenner (Abb. 11.24), um den Schweißstrom von einem Kabel zur Elektrode zu übertragen. Die Aufgabe wird in vielen verschiedenen Bereichen der Fertigung wahrgenommen. Neben dem Lichtbogenschweißen sind auch Punktschweißzangen (Abb. 11.25) häufig in Fertigungsprozessen zu finden, vor allem in der Automobilindustrie.

Kapitel 12

Kollaborative Roboter



1942 veröffentlichte Isaac Asimov den Science-Fiction-Roman „I, Robot“, in dem die drei Gesetze der Robotik vorgestellt wurden. Die erste Regel besagte: „Ein Roboter darf keinen Menschen verletzen oder durch Untätigkeit zulassen, dass einem Menschen Schaden zugefügt wird.“

Bisher waren Industrieroboter immer schnelle und robuste Geräte, die speziell für sie konzipierte Aufgaben übernehmen. Um die oben genannte Regel einzuhalten, arbeiteten sie hinter festen und verriegelten Schutzvorrichtungen und empfindlicher Schutzausrüstung, um das Eindringen von Menschen in ihren Arbeitsbereich zu verhindern. Mit der Einführung kollaborativer Roboter entfallen die Käfige, da diese Roboter für die Zusammenarbeit mit Menschen konzipiert sind. Sie sind mit unterschiedlichen Sicherheitsfunktionen ausgestattet, um Kollisionen zu verhindern. Sollte es jedoch zu einer Kollision kommen, bewegt sich der Mechanismus in die entgegengesetzte Richtung oder stoppt ganz, um Verletzungen zu vermeiden.

Die technische Spezifikation ISO/TS 15066:2016: Roboter und Robotergeräte – Kollaborative Roboter ergänzt die Anforderungen und Leitlinien zum kollaborativen Industrieroboterbetrieb gemäß ISO 10218-1:2011 und ISO 10218-2:2011 (ANSI/RIA R15.06). :2012). Es legt Sicherheitsanforderungen für kollaborative Industrierobotersysteme und die Arbeitsumgebung fest. Insbesondere bietet ISO/TS 15066:2016 umfassende Leitlinien für die Risikobewertung bei kollaborativen Roboteranwendungen.

12.1 Kollaboratives Industrierobotersystem

Ein kollaborativer Roboter ist ein Roboter, der in einem kollaborativen Betrieb eingesetzt werden kann, bei dem ein speziell entwickeltes Robotersystem und ein menschlicher Bediener in einem definierten Arbeitsbereich direkt zusammenarbeiten. Der Begriff „Roboter“ definiert den Roboterarm und die Robotersteuerung und umfasst nicht den Endeffektor oder Teil des Roboters. Mit dem Begriff Robotersystem beschreiben wir Roboter, Endeffektor und Werkstück.

Für das kollaborative Robotersystem können wir verschiedene Arbeitsbereiche definieren (Abb. 12.1):

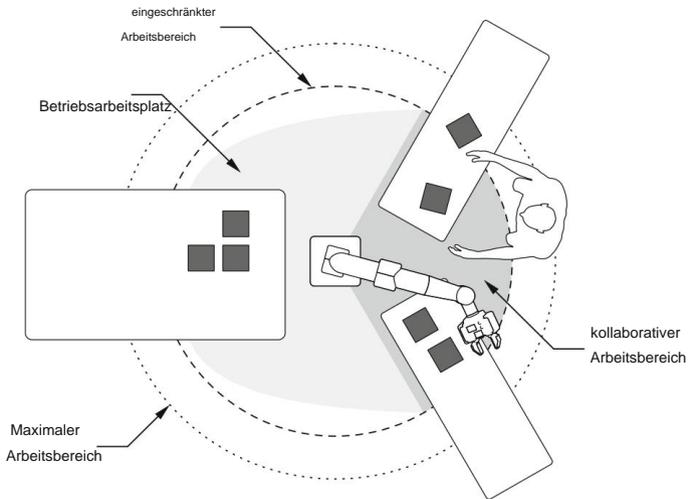


Abb. 12.1 Maximaler Arbeitsbereich (begrenzt durch gepunktete Linie), eingeschränkter Arbeitsbereich (begrenzt durch gestrichelte Linie), Betriebsarbeitsbereich (graue Bereiche) und kollaborativer Arbeitsbereich (dunkelgrauer Bereich)

- **maximaler Arbeitsbereich:** Raum, der von den beweglichen Segmenten des Roboters überstrichen werden kann, wie vom Hersteller definiert, plus der Raum, der vom Endeffektor und dem Werkstück überstrichen werden kann;
- **eingeschränkter Arbeitsbereich:** Teil des maximalen Speicherplatzes, der durch die Gerätebeschränkung eingeschränkt wird die Grenzwerte festlegen, die nicht überschritten werden dürfen;
- **Betriebsarbeitsplatz:** Teil des eingeschränkten Raums, der tatsächlich genutzt wird Ausführen aller vom Aufgabenprogramm befohlenen Bewegungen;
- **Kollaborativer Arbeitsbereich:** Teil des Betriebsraums, in dem das Robotersystem und ein Mensch während des Produktionsbetriebs gleichzeitig Aufgaben ausführen können.

Der kollaborative Arbeitsbereich muss so gestaltet sein, dass der Bediener alle vorgesehenen Aufgaben ausführen kann. Der Standort von Maschinen und Geräten sollte keine zusätzlichen Sicherheitsrisiken mit sich bringen. Im kollaborativen Arbeitsbereich gelten strenge Beschränkungen hinsichtlich Geschwindigkeit, Platzbeschränkungen und Drehmomenterfassung, um die Sicherheit des Bedieners zu gewährleisten. Außerhalb des kollaborativen Arbeitsbereichs kann der Roboter ohne besondere Einschränkungen, mit Ausnahme derjenigen, die aufgabenbezogen sind, als herkömmlicher Industrieroboter agieren.

Der Begriff Bediener umfasst alle Personen, die mit der Roboteranlage in Kontakt stehen, nicht nur Produktionsbediener. Es umfasst Wartungs-, Fehlerbehebungs-, Einrichtungs-, Reinigungs- und Produktionspersonal.

Die Betriebseigenschaften kollaborativer Robotersysteme unterscheiden sich erheblich von denen herkömmlicher Industrierobotersysteme, die in ISO 10218-1:2011 und ISO 10218-2:2011 vorgestellt werden. Bei kollaborativen Roboteroperationen können Bediener in unmittelbarer Nähe des Robotersystems arbeiten, während das System aktiv ist, und es kann innerhalb des kollaborativen Arbeitsbereichs zu physischem Kontakt zwischen einem Bediener und dem Robotersystem kommen. Daher müssen angemessene Schutzmaßnahmen für die Zusammenarbeit eingeführt werden.

rationelle Robotersysteme, um die Sicherheit des Bedieners während der Zusammenarbeit jederzeit zu gewährleisten Roboterbetrieb.

12.2 Kollaborativer Roboter

Beim Design kollaborativer Roboter geht es weg von schweren, steifen und starren Industrierobotern hin zu leichten Geräten mit aktiver und/oder passiver Nachgiebigkeit.

Die Verwendung von leichten, hochfesten Metallen oder Verbundwerkstoffen für Roboterverbindungen trägt zu einer geringen Bewegungsträgheit bei, die sich weiter auf den Stromverbrauch auswirkt die Motoren. Serielle Manipulatoren können mit leistungsstarken/drehmomentstarken Motoren ausgestattet werden Zahnräder mit hohem Übersetzungsverhältnis in jedem Gelenk oder Motoren, die an der Basis positioniert sind Die Kraftübertragung erfolgt über Sehnen. Wenn das Übersetzungsverhältnis klein ist, ist das System klein von Natur aus rückwärts antreibbar.

Der Einsatz von intrinsisch flexiblen Aktoren ermöglicht die Gestaltung biologisch inspirierter Roboter, da die Aktoren die Leistung menschlicher/tierischer Muskeln nachahmen. Der Aktoren können eine feste mechanische Impedanz haben, die über eine aktive Steuerung gesteuert wird, z als seriell elastischer Aktuator (SEA), oder die Impedanz kann durch Veränderung angepasst werden Parameter einer mechanischen Verbindung, wie bei einem Stellantrieb mit variabler Steifigkeit (VSA). SEA ist ein Kombination aus Motor, Getriebe und einer Feder, bei der die Drehung der Feder gemessen wird um die Kraftabgabe zu steuern, während die Messung der Drehung der Feder verwendet wird als Kraftsensor. VSA kann verwendet werden, um den Roboter im Falle einer Kollision sicherer zu machen da die Gelenksteifigkeit und die Stoßträgheit reduziert werden. Konzeptionelle Entwürfe von SEA und VSA sind in Abb. 12.2 dargestellt .

Kollaborative Roboter verfügen außerdem über spezielle Geometrien, die die Kontaktenergie minimieren Übertragung durch Maximierung der Aufprallfläche. Roboter haben runde Formen und sind integriert Merkmale, die das Risiko von Quetschstellen und die Schwere eines Aufpralls verringern. Hauptsächlich Merkmale des kollaborativen Roboters sind in Abb. 12.3 dargestellt

Um ein hohes Maß an Sicherheit zu gewährleisten, muss das Robotersystem über verschiedene Sensoren verfügen zur Überwachung des Zustands des Roboters und seines Arbeitsbereichs, wie in Kap. 7.

Roboter können mit Gelenkdrehmomentsensoren, Kraft-/Drehmomentsensoren am Endeffektor und verschiedenen taktilen Sensoren ausgestattet werden, die als weiche Haut oder harte Schale für den Roboter dienen. Alle diese Sensoren ermöglichen es dem Roboter, den Kontakt mit der Umgebung (Bediener) zu erkennen. oder vermeiden Sie eine Kollision, indem Sie sie antizipieren und entsprechend reagieren. Einige Roboter verwenden

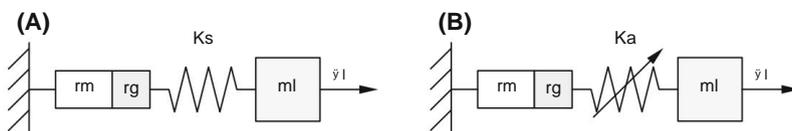


Abb. 12.2 a Serieller elastischer Aktuator (SEA), b variabler Steifigkeitsaktor (VSA); rm und rg repräsentieren Motor und Getriebe, K_s - nachgiebiges Element mit fester Steifigkeit, K_a einstellbares nachgiebiges Element, ml Masse des beweglichen Glieds und \bar{y}_l Gelenkdrehmoment, was zur Bewegung des Glieds führt

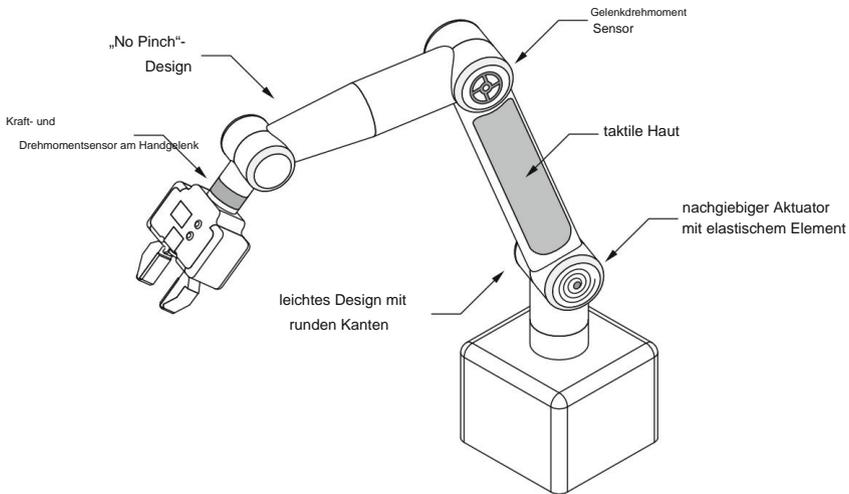


Abb. 12.3 Konstruktionsmerkmale eines kollaborativen Roboters

reduzante Encoder in jedem Gelenk als Ersatz für teure Gelenkdrehmomente; Die Kraft kann aus dem bekannten Motorstrom und der bekannten Gelenkposition abgeleitet werden. Robotersysteme können andere sicherheitsrelevante Sensoren umfassen, z. B. Sicherheitskameras, Laserscanner, Laservorhänge, Sicherheitsmatten und andere elektrosensible Schutzausrüstungen, um die Anwesenheit des Bedieners in der Roboterumgebung zu erkennen. Diese Informationen können dann für eine ordnungsgemäße Reaktion des Roboters verwendet werden, um ein Einklemmen, Zusammendrücken o

Die eingebauten Sensoren können zur sicheren Steuerung des Roboters genutzt werden. Das Hauptparadigma ist der Umgang mit dem physischen Kontakt zwischen dem Mechanismus und der Umgebung. Eines der beliebtesten Regelungsschemata ist die Impedanzregelung, die auf dem dynamischen Robotermodell (5.56) basiert. Das dynamische Modell wird verwendet, um die notwendigen Gelenkdrehmomente für eine ordnungsgemäße Bewegung des Roboters zu ermitteln. Weichen die gemessenen Gelenkmomente von den ermittelten ab, so wird die Differenz als Kollision erkannt. Wenn eine Kollision erkannt wurde, sollte die richtige Reaktionsstrategie aktiviert werden, um eine potenzielle Gefahr für den Bediener zu verhindern. Der Roboter kann den Kontakt ignorieren und der Referenztrajektorie folgen, oder der Roboter kann angehalten werden. Weitere Möglichkeiten sind der Wechsel von der Positionssteuerung zur Schwerelosigkeits-Drehmomentsteuerung (sehr hohe Nachgiebigkeit des Roboters), der Wechsel zur Drehmomentsteuerung unter Verwendung von Signalen von Gelenkdrehmomenten zur Minimierung der Verbindungs- und Motorträgheit (noch „leichterer“ Roboter) oder zu Verwenden Sie extern gemessene Drehmomente und wechseln Sie zur Admittanzsteuerung, bei der Roboter und kollidiertes Objekt wie zwei Magnete wirken, die mit den gleichen Polen ein

Das Ziel kollaborativer Roboter besteht darin, das Beste von Robotern und dem des menschlichen Bedieners zu kombinieren: die Präzision, Kraft und Ausdauer des Roboters gepaart mit der hervorragenden Fähigkeit des menschlichen Bedieners, ungenaue Probleme zu lösen. Da der Roboter und die Bediener im selben Arbeitsbereich zusammenarbeiten, ist der Kontakt zwischen Robotern und Menschen möglich. Wenn es zu einem zufälligen Kontakt kommt, sollte dieser Kontakt nicht stattfinden

kann zu Schmerzen oder Verletzungen führen. So können kollaborative Roboter neben den Bedienern eingesetzt werden und die Produktivität der Arbeiter steigern. Roboter sind leicht und haben eine geringe Stellfläche, sodass sie problemlos in der Werkstatt bewegt werden können, was ihre Vielseitigkeit erhöht. Die Programmierung kollaborativer Roboter ist einfach und erfolgt meist per Handführung, sodass der Einsatz des Roboters sehr flexibel ist; Der Roboter kann in kürzester Zeit an einem neuen Arbeitsplatz einsatzbereit sein.

12.3 Kollaborativer Betrieb

Der kollaborative Betrieb wird nicht allein durch den Einsatz des Roboters definiert, sondern ist durch die Aufgabe, die Tätigkeit des Robotersystems und den Raum, in dem die Aufgabe ausgeführt wird, bedingt. Vier Haupttechniken (eine oder eine Kombination mehrerer) können in den kollaborativen Betrieb einbezogen werden:

- sicherheitsbewerteter überwachter Stopp;
- Handführung;
- Geschwindigkeits- und Abstandsüberwachung;
- Kraft- und Kraftbegrenzung.

Bei allen vier Techniken arbeitet der Roboter im Automatikmodus. Die wichtigsten Details aller vier Methoden sind in Tabelle 12.1 dargestellt. Ausführlichere Beschreibungen finden Sie weiter unten.

Tabelle 12.1 Arten von kollaborativen Operationen

	Geschwindigkeitsdrehmomente	Bedienelemente Technik	
Sicherheitsbewerteter überwachter Stopp	Null, während sich der Bediener im kollaborativen Arbeitsbereich befindet	Nur Schwerkraft- und Lastausgleich	Keine, während sich der Bediener im kollaborativen Arbeitsbereich befindet
Handführung	Sicherheitsbewertete überwachte Geschwindigkeit	B. durch direkte Bedieneringabe	Not-Aus, Freigabevorrichtung, Bewegungseingang
Geschwindigkeits- und Abstandsüberwachung	Sicherheitsbewertete überwachte Geschwindigkeit	Je nach Bedarf zur Einhaltung von min. Sicherheitsabstand einzuhalten und die Anwendung auszuführen	Keine, während sich der Bediener im kollaborativen Arbeitsbereich befindet
Kraft- und Kraftbegrenzung	Max. bestimmte Geschwindigkeit, um die Aufprallkräfte zu begrenzen	Max. ermitteltes Drehmoment zur Begrenzung statischer Kräfte	Je nach Antrag
			Keine Bewegung in Anwesenheit des Bedieners
			Bewegung nur durch direkte Bedieneringabe
			Verhindert den Kontakt zwischen dem Robotersystem und dem Bediener
			Der Roboter kann keine übermäßige Kraft ausüben (konstruktionsbedingt oder gesteuert).

12.3.1 Sicherheitsbewerteter überwachter Stopp

Bei dieser Methode muss das Robotersystem mit sicherheitsbewerteten Geräten ausgestattet sein, die die Anwesenheit des Bedieners im kollaborativen Arbeitsbereich erkennen (z. B. Lichtvorhänge oder Laserscanner). Der Bediener darf nur dann mit dem Robotersystem im kollaborativen Arbeitsbereich interagieren, wenn die sicherheitsbewertete überwachte Stoppfunktion des Roboters aktiv ist und die Roboterbewegung gestoppt wird, bevor der Bediener den gemeinsamen Arbeitsbereich betritt. Während der kollaborativen Aufgabe steht der Roboter still und die Motoren werden mit Strom versorgt. Die Bewegung des Robotersystems kann erst wieder aufgenommen werden, wenn der Bediener den kollaborativen Arbeitsbereich verlassen hat. Wenn sich im kollaborativen Arbeitsbereich kein Bediener befindet, kann der Roboter als klassischer Industriearbeiter arbeiten.

Die Vorgänge des sicherheitsbewerteten überwachten Stopps sind in Tabelle 12.2 dargestellt. Wenn sich der Bediener außerhalb des kollaborativen Arbeitsbereichs befindet, kann der Roboter ohne Einschränkungen arbeiten. Für den Fall, dass sich der Roboter jedoch gleichzeitig mit dem Bediener im Arbeitsbereich befindet, sollte der sicherheitsbewertete überwachte Stopp des Roboters aktiv sein. Andernfalls muss der Roboter im Fehlerfall einen Schutzstopp der Kategorie 0 (unkontrollierter Stopp des Roboters durch sofortiges Trennen der Stromzufuhr zu den Aktoren) durchführen (IEC 60204-1).

Diese Methode kann auf Anwendungen des manuellen Be- oder Entladens von Endeffektoren, Inspektionen laufender Arbeiten und Anwendungen angewendet werden, bei denen sich nur eine Person im kollaborativen Arbeitsbereich bewegt (z. B. Roboter oder Bediener). Sicherheitsüberwachte Stopps können auch in andere kollaborative Techniken integriert werden.

12.3.2 Handführung

Für die Handführung muss der Roboter mit einer speziellen Führungsvorrichtung ausgestattet sein, die sich am oder in der Nähe des Roboter-Endeffektors befindet und der Übertragung von Bewegungsbefehlen an das Robotersystem dient. Das Gerät muss über einen Not-Aus-Schalter und eine Freigabevorrichtung verfügen, es sei denn, das Robotersystem erfüllt inhärent sichere Konstruktionsmaßnahmen oder Sicherheitsbegrenzungsfunktionen. Die Position der Führungsvorrichtung sollte o

Tabelle 12.2 Roboteraktionen für sicherheitsbewerteten überwachten Stopp

		Nähe des Bedieners zum kollaborativen Arbeitsbereich	
		Draußen	Innen
Die Nähe des Roboters zur Zusammenarbeit Arbeitsplatz	Draußen	Weitermachen	Weitermachen
	Drinnen und in Bewegung	Weitermachen	Schutzstopp
	Im Inneren sicherheitstechnisch überwachter Anschlag	Weitermachen	Weitermachen

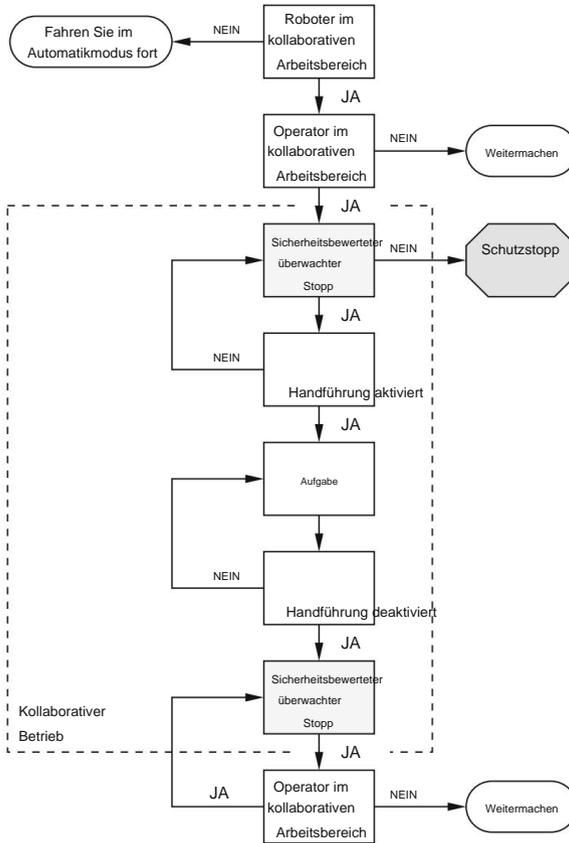


Abb. 12.4 Der Bedienablauf bei der Handführung

Beobachten Sie die Bewegung des Roboters direkt und vermeiden Sie gefährliche Situationen (z. B. wenn der Bediener unter schwerer Last steht). Die Steuerung von Roboter und Endeffektor soll intuitiv verständlich und kontrollierbar sein.

Das Robotersystem ist für die Handführung bereit, wenn es den kollaborativen Arbeitsbereich betritt und einen sicherheitsbewerteten überwachten Stopp ausgibt. An diesem Punkt kann der Bediener den kollaborativen Arbeitsbereich betreten und mit dem Handführungsgerät die Kontrolle über das Robotersystem übernehmen. Wenn der Bediener den kollaborativen Arbeitsbereich betritt, bevor das System zur Handführung bereit ist, muss ein Schutzstopp ausgesprochen werden. Nachdem der sicherheitsüberwachte Stopp aufgehoben wurde, kann der Bediener die Handführung aktivieren. Wenn der Bediener die Führungsvorrichtung loslässt, wird der sicherheitsbewertete überwachte Stopp ausgegeben. Der nicht kollaborative Betrieb wird wieder aufgenommen, wenn der Bediener den kollaborativen Arbeitsbereich verlässt. Der Bedienablauf für die Handführung ist in Abb. 12.4 dargestellt.

Diese Kollaborationstechnik eignet sich für die Implementierung in Anwendungen, in denen das Robotersystem als Leistungsverstärker fungiert, in hochvariablen Anwendungen,

wo Robotersysteme als Werkzeug eingesetzt werden und in Anwendungen, bei denen die Koordination manueller und teilweise automatisierter Schritte erforderlich ist. Die handgeführte Zusammenarbeit kann erfolgreich in limitierte oder kleine Serienproduktionen implementiert werden.

12.3.3 Geschwindigkeits- und Abstandsüberwachung

Bei dieser Methode können sich Bediener und Robotersystem gleichzeitig im kollaborativen Arbeitsbereich bewegen. Bei gemeinsamen Einsätzen wird der Mindestschutzabstand zwischen Bediener und Robotersystem stets eingehalten. Der Schutzabstand ist der kürzeste zulässige Abstand zwischen allen beweglichen gefährlichen Teilen des Robotersystems und dem Bediener im kollaborativen Arbeitsbereich.

Der Schutzabstand S_p zum Zeitpunkt t_0 kann durch (12.1) beschrieben werden:

$$S_p(t_0) = S_h + S_r + S_s + C + Z_d + Z_r, \quad (12.1)$$

Dabei ist S_h der Beitrag zum Schutzabstand, der auf den Standortwechsel des Bedieners zurückzuführen ist. Die Formel berücksichtigt den Bremsweg S_r , der der Weg aufgrund der Reaktionszeit des Roboters ist, und S_s , der den Weg beschreibt, der durch den Bremsweg des Robotersystems bedingt ist. C stellt die Eindringdistanz dar, also die Distanz, die ein Körperteil in das Erfassungsfeld eindringen kann, bevor es erkannt wird. Der Schutzabstand S_p umfasst auch die Positionsunsicherheit des Bedieners Z_d , die sich aus der Abtastmesstoleranz ergibt, und die Positionsunsicherheit des Robotersystems Z_r , die sich aus der Genauigkeit des Roboterpositionsmesssystems ergibt. Die maximal zulässigen Geschwindigkeiten und die minimalen Schutzabstände in einer Anwendung können entweder variabel oder konstant sein. Die verschiedenen Beiträge zum Schutzabstand sind in Abb. 12.5 dargestellt.

Der Roboter muss mit einer sicherheitsbewerteten, überwachten Geschwindigkeitsfunktion und einem sicherheitsbewerteten, überwachten Stopp ausgestattet sein. Das Robotersystem umfasst auch zusätzliche sicherheitsbewertete Peripheriegeräte zur Überwachung durch Menschen (z. B. sicherheitsbewertete Kamerasysteme). Das Robotersystem kann den Mindestschutzabstand durch eine Geschwindigkeitsreduzierung einhalten, der ein sicherheitsbewerteter überwachter Stopp folgen könnte, oder die Ausführung eines alternativen Pfads, der den Schutzabstand nicht verletzt, wie in Abb. 12.6 dargestellt. Wenn der tatsächliche Abstand zwischen dem Robotersystem und dem Bediener unter den Schutzabstand fällt, sollte das Robotersystem einen Schutzstopp einleiten und mit dem Robotersystem verbundene sicherheitsrelevante Funktionen einleiten (z. B. alle gefährlichen Werkzeuge ausschalten). Wenn sich der Bediener vom Roboter entfernt, entspricht der tatsächliche Abstand dem Schutzabstand und überschreitet diesen; An diesem Punkt kann der Roboter die Bewegung automatisch fortsetzen.

Geschwindigkeits- und Abstandsüberwachung ist in Anwendungen nützlich, in denen Robotersysteme zum Einsatz kommen und die Aufgaben des Bedieners laufen gleichzeitig.

12.3 Kollaborativer Betrieb

181

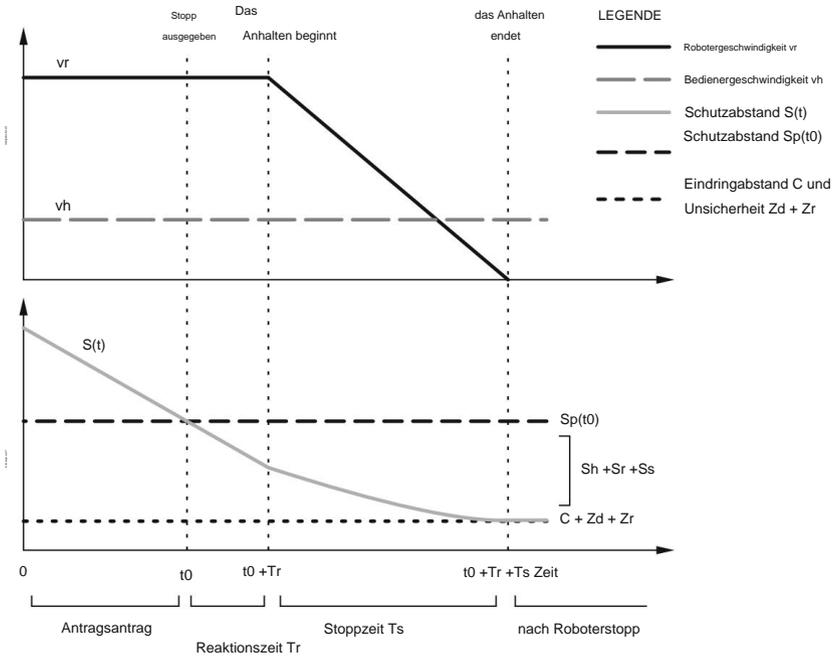


Abb. 12.5 Grafische Darstellung der Beiträge zum Schutzabstand zwischen einem Bediener und einem Roboter

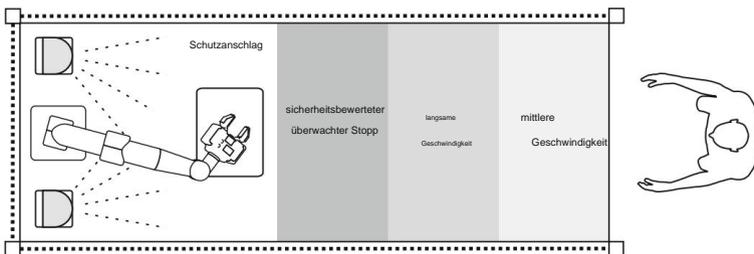


Abb. 12.6 Sicherheitsbewertete Ebenen zur Einhaltung des Mindestschutzabstands

12.3.4 Leistungs- und Kraftbegrenzung

Die Methode der Leistungs- und Kraftbegrenzung ermöglicht einen physischen Kontakt zwischen dem Robotersystem und dem Bediener, der entweder absichtlich oder unabsichtlich erfolgen kann. Die Methode erfordert eine spezifische Konstruktion der Roboter durch geringe Trägheit, geeignete Geometrie (abgerundete Kanten und Ecken, glatte und nachgiebige Oberflächen), Materialien (Polsterung, Dämpfung, verformbare Komponenten) und Steuerungsfunktionen. Ersteres umfasst aktive Sicherheitsdesignmethoden wie die

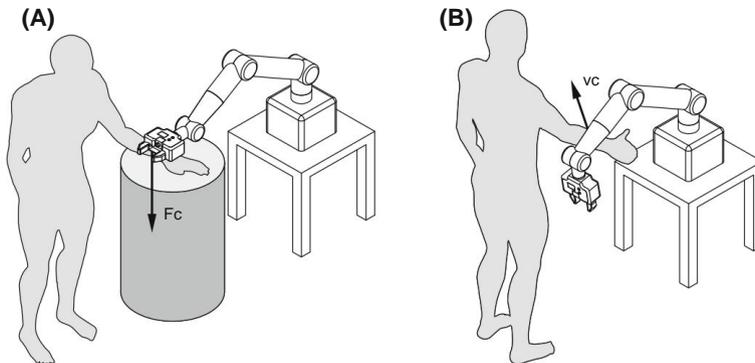


Abb. 12.7 a Quasistatischer und b transientser Kontakt

Begrenzung der Geschwindigkeiten bewegter Teile, Begrenzung des Impulses durch Begrenzung bewegter Massen und Begrenzung mechanischer Leistung oder Energie als Funktion von Massen und Geschwindigkeiten. Das Design des Roboters kann auch die Verwendung von sicherheitsbewerten Soft-Achsen, raumbegrenzenden Funktionen und sicherheitsbewerten überwachten Stoppfunktionen umfassen. Einige Roboter verfügen auch über Sensoren, um einen Kontakt zu antizipieren oder zu erkennen.

Der Kontakt zwischen dem kollaborativen Roboter und den Körperteilen des Bediener könnte sein:

- als Teil der Bewegungssequenz gedacht; • Zufällig aufgrund der Nichteinhaltung des Arbeitsverfahrens, jedoch ohne technisches Versagen;
- ein Fehlermodus, der zu Kontaktsituationen führt.

Es gibt zwei mögliche Arten des Kontakts zwischen beweglichen Teilen des Robotersystems und Bereichen am Körper des Bediener. Der quasistatische Kontakt (Abb. 12.7a) umfasst eine Klemm- oder Quetschsituation, bei der der Körperteil des Bediener zwischen einem beweglichen Teil des Robotersystems und einem anderen festen oder beweglichen Teil der Arbeitszelle eingeklemmt wird. In dieser Situation wird der Druck oder die Kraft F_c des Robotersystems über einen längeren Zeitraum ausgeübt, bis sich die Bedingungen bessern. Der vorübergehende Kontakt (d. h. dynamischer Stoß, Abb. 12.7b) beschreibt den Kontakt zwischen dem beweglichen Teil des Robotersystems und dem Körperteil des Bediener, ohne dass dieser Teil eingeklemmt oder eingeklemmt wird.

Der tatsächliche Kontakt ist kürzer als der oben erwähnte quasistatische Kontakt (<50 ms) und hängt von der Trägheit des Roboters, der Trägheit des Körperteils des Bediener und der Relativgeschwindigkeit v_c der beiden ab.

Das Robotersystem muss ausreichend ausgelegt sein, um das Risiko für einen Bediener zu verringern, indem die geltenden Grenzwerte für Kraft und Druck für quasistatischen und vorübergehenden Kontakt nicht überschritten werden. Die Grenzen können für Kräfte, Drehmomente, Geschwindigkeiten, Impulse, mechanische Leistung, Gelenkbewegungsbereiche oder Raumbereiche gelten. Für beide Kontaktarten werden Schwellenwerte für das jeweilige Kontaktereignis an der exponierten Körperregion für ein Worst-Case-Szenario ermittelt.

Die in ISO/TS 15066:2016 dargestellten Grenzwerte basieren auf einer konservativen Schätzung und wissenschaftlicher Forschung zu Schmerzempfindungen. Einige informative Werte für

Tabelle 12.3 Biomechanische Grenzwerte für quasistatischen Kontakt

Körperbereich	Maximal zulässig Druck pQS/N/cm ²	Maximal zulässig FQS/N erzwingen
Siebter Nackenmuskel	210	150
Schultergelenk	160	210
Sternum	120	140
Abdomen	140	110
Becken	210	180
Humerus	220	150
Unterarm	180	160
Palme	260	140
Zeigefingerauflage	300	140
Endgelenk des Zeigefingers	280	140
Handrücken	200	140
Oberschenkel	250	220
Kniescheibe	220	220
Schienbein	220	130
Kalb	210	130

maximal zulässiger Druck und maximal zulässige Kraft zwischen den Robotern

Teil und Körperregion des Bedieners in quasistatischem Kontakt sind in Tabelle 12.3 dargestellt.

Druck- und Kraftwerte für transienten Kontakt (p_T , F_T) können mindestens das Zweifache betragen

Werte für quasistatischen Kontakt (p_{QS} , F_{QS}).

$$p_T = 2 \cdot p_{QS} \quad F_T = \quad (12.2)$$

$$2 \cdot F_{QS} \quad (12.3)$$

Der Kontakt mit Gesicht, Schädel oder Stirn ist nicht zulässig und muss verhindert werden.

Für ordnungsgemäße Reaktionen des Robotersystems müssen sowohl Druck- als auch Kraftgrenzen eingehalten werden je nach Situation in Betracht gezogen werden. Im Falle einer Klemmung des Körpers des Bedieners

B. der Hand des Bedieners), kann die resultierende Kraft deutlich unter dem Grenzwert liegen

Daher ist die Druckgrenze der begrenzende Faktor. Andererseits, wenn der Kontakt

liegt zwischen zwei relativ großen und weichen Bereichen (z. B. gepolstertem Roboterteil und Bedienerbereich).

Bauch) wird der resultierende Druck unterhalb der Grenzwelle und des Grenzwertes liegen

Faktor ist dann die Kraftgrenze.

Im Falle einer Berührung muss das Robotersystem so reagieren, dass die Wirkung des

Der identifizierte Kontakt bleibt unter den angegebenen Grenzwerten

in Abb. 12.8. Beim Einklemmen oder Fixieren eines Körperteils zwischen einem Robotersegment

und einem anderen Objekt muss der Roboter die Geschwindigkeit begrenzen, um den Schutzbestimmungen zu entsprechen

Grenzen. Der Roboter sollte außerdem über eine integrierte Option zur manuellen Bedienung durch den Bediener verfügen

Befreien Sie den betroffenen Körperbereich.

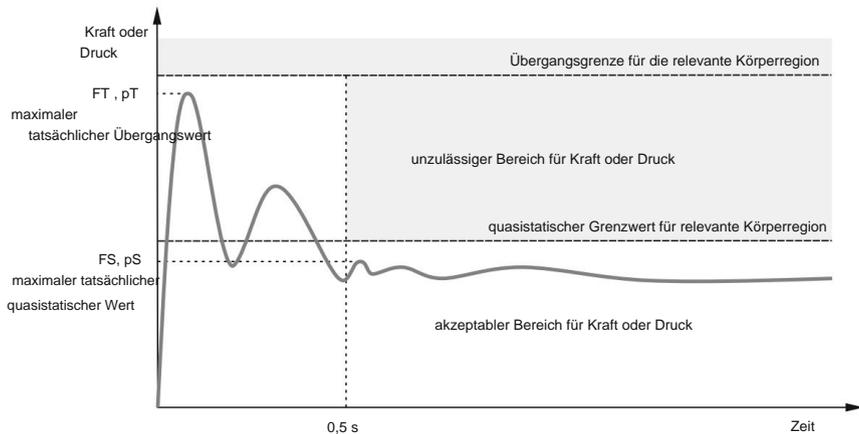


Abb. 12.8 Grafische Darstellung der zulässigen und unzulässigen Kräfte oder Drücke bei quasistatischem oder transientem Kontakt

Die Kraft- und Kraftbegrenzungsmethode kann in kollaborativen Anwendungen eingesetzt werden, bei denen die Anwesenheit des Bedieners häufig erforderlich ist, bei zeitabhängigen Vorgängen (bei denen Verzögerungen aufgrund sicherheitsbewerteter Stopps unerwünscht sind, es jedoch zu physischem Kontakt zwischen dem Robotersystem und dem Bediener kommen kann.) und Anwendungen mit kleinen Teilen und hoher Montagevariabilität.

12.4 Kollaborative Robotergreifer

Das Design und die Steuerung eines kollaborativen Roboters ermöglichen eine sichere Zusammenarbeit des Roboters mit dem Bediener. Aber der Roboter selbst ist nur ein Teil des Robotersystems. Greifer stellen einen wichtigen Bestandteil des Robotersystems dar, da sie zur Objektmanipulation in unmittelbarer Nähe des Bedieners eingesetzt werden. Daher müssen Greifer ein hohes Maß an Sicherheit erreichen.

Die Greifer sind in der Regel starr am bereits sicheren Roboter befestigt und verfügen über eingebaute Geschwindigkeits- und Kraftbegrenzungen. Form und Materialien des Greifers müssen mit der Sicherheitskonstruktion übereinstimmen, um eine Überschreitung der Druckgrenzen im Kontaktbereich des Körpers des Bedieners zu verhindern. Darüber hinaus sollten die Greifer an der Spitze des Roboters möglichst wenig Trägheit erzeugen, um die Sicherheitsfunktionen des Roboters möglichst wenig zu beeinträchtigen.

Durch die Konstruktion der Greifer soll verhindert werden, dass der Bediener mit den Fingern in den Greifer oder in die Anschlusskabel gerät. Die Greifer müssen über einen sicheren Not-Aus-Modus verfügen, dessen Funktion von der Anwendung abhängt.

Wenn ein gegriffenes Teil vorhanden ist, möchte der Bediener normalerweise, dass das Teil sicher gegriffen bleibt. Beim Teachen und Schließen des Greifers möchte der Bediener, dass der Greifer keine Kraft mehr ausübt.

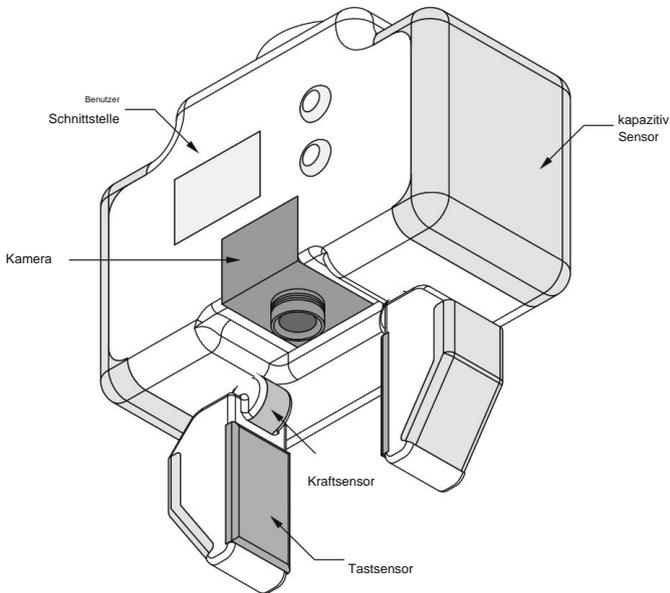


Abb. 12.9 Konzeptioneller Entwurf eines Greifers zum kollaborativen Greifen

Wenn der Greifer mit dem Teil interagiert, möchte der Bediener einen guten, festen Halt haben. Der Griff muss auch bei einem Notstopp oder einem Stromausfall sicher sein, da ein herunterfallendes Teil eine Gefahr für den Bediener, den Roboter oder die Umgebung darstellen könnte. Wenn sich der Roboter schnell bewegt, könnte das heruntergefallene Teil zu einem Projektil werden.

Um die Sicherheit des Bedieners zu erhöhen, können Greifer mit unterschiedlichen Sensoren ausgestattet werden (Abb. 12.9). Kapazitive Sensoren dienen der frühzeitigen Bedienererkennung und damit der Vermeidung unerwünschter Kontakte. Kamerasysteme können die Umgebung des Roboters erfassen und bei der Objektsuche helfen. Zur Unterscheidung zwischen Werkstück und Bediener werden taktile Sensoren eingesetzt. Um eine ausreichende Greifkraft einzustellen, können verschiedene Kraftsensoren verwendet werden. Das Greiferdesign kann auch verschiedene Benutzerschnittstellen umfassen, wie z. B. einen LCD-Bildschirm, Signalleuchten und Steuertasten.

Greifer, die in den kollaborativen Robotersystemen eingesetzt werden, sollten einfach zu installieren und zu programmieren sein. Das zukünftige Design der Greifer tendiert weg von der Benutzerprogrammierung hin zu Greifern, die sich automatisch an die Teile und Anwendungen anpassen können.

12.5 Anwendungen kollaborativer Robotersysteme

Das Dokument ISO 10218-2:2011 sieht die Einteilung kollaborativer Anwendungen in fünf Kategorien vor, die in Abb. 12.10 dargestellt sind.

Die Anwendung „Übergabefenster“ (siehe Abb. 12.10a) umfasst Be-/Entlade-, Prüf-, Prüf-, Reinigungs- und Serviceaufgaben. Der Roboter wird hinter feststehenden oder positioniert

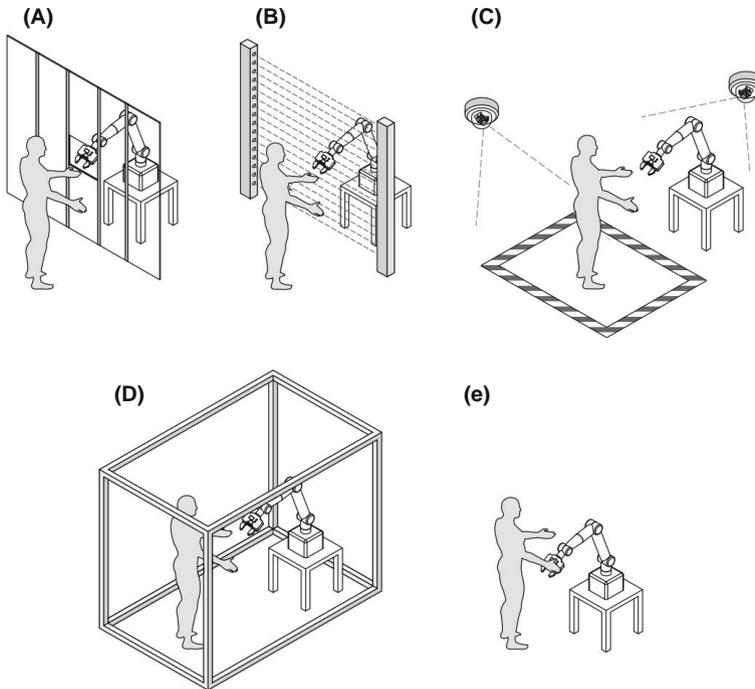


Abb. 12.10 Konzeptionelle Anwendungen kollaborativer Roboter: a Übergabefenster, b Schnittstellenfenster, c kollaborativer Arbeitsbereich, d Inspektion und e handgeführter Roboter (ISO 10218-2:2011)

empfindliche Schutzvorrichtungen rund um den Arbeitsbereich, in dem die Anwendung im automatischen Modus ohne Einschränkungen ausgeführt wird. Die Interaktion mit dem Bediener erfolgt über a Fenster. In der Nähe des Fensters reduziert der Roboter seine Geschwindigkeit. Das Fenster auch fungiert als Grenze für den Roboterarbeitsbereich.

Das Schnittstellenfenster (Abb. 12.10b) fungiert als Barriere für das Robotersystem. Auf der Roboterseite kann der Roboter autonome automatische Vorgänge ausführen. Das Robotersystem wird außerdem durch feste oder empfindliche Schutzvorrichtungen rund um den Arbeitsbereich bewacht. Der Roboter stoppt im Schnittstellenfenster angezeigt und kann dann manuell außerhalb der Schnittstelle verschoben werden. Für für geführte Bewegungen muss der Roboter mit einer Handführungseinrichtung ausgestattet sein. Diese Methode wird zum automatischen Stapeln, zur geführten Montage, zum geführten Befüllen, zum Testen, zum Richten von Prüfständen usw. verwendet. und Reinigung.

Anwendungen mit einfacher Montage und Handhabung können von den Vorteilen profitieren kollaborativer Arbeitsbereich (Abb. 12.10c). Innerhalb des gemeinsamen Arbeitsbereichs kann der Roboter automatische Vorgänge durchführen. Wenn der Bediener den kollaborativen Arbeitsbereich betritt, Der Roboter reduziert die Geschwindigkeit und/oder stoppt. Bei solchen Anwendungen sind zusätzliche Personenerkennungssysteme mit einem oder mehreren Sensoren erforderlich.

Anwendungen einschließlich Inspektion und Parametereinstellung (z. B. Schweißanwendungen, siehe Abb. 12.10d) erfordern geschützte Arbeitsbereiche und Personenerkennungssysteme.

12.5 Anwendungen kollaborativer Robotersysteme

187

Wenn der Bediener den gemeinsamen Arbeitsbereich betritt, setzt der Roboter den Betrieb fort reduzierte Geschwindigkeit. Die Anwendung muss über zusätzliche Maßnahmen zur Verhinderung von Missbrauch verfügen.

Handgeführte Roboter (Abb. 12.10e) werden für handgeführte Anwendungen eingesetzt (z. B. zusammenbauen oder lackieren). Der Roboter ist mit einer Handführungsvorrichtung ausgestattet. Der Bediener führt den Roboter von Hand entlang einer Bahn in einem aufgabenspezifischen Arbeitsbereich mit reduzierter Geschwindigkeit. Der Bereich des kollaborativen Arbeitsbereichs hängt hauptsächlich von den Gefahren ab die erforderliche Anwendung.

Kapitel 13

Mobile Roboter



Ein mobiler Roboter ist ein Gerät, das sich fortbewegen kann. Es hat die Fähigkeit, sich mithilfe von Rädern, Schienen, Beinen oder einer Kombination davon in seiner Umgebung fortzubewegen. Es kann auch fliegen, schwimmen, krabbeln oder rollen. Mobile Roboter werden für verschiedene Anwendungen in Fabriken (fahrerlose Transportfahrzeuge), Privathaushalten (Bodenreinigungsgeräte), Krankenhäusern (Transport von Lebensmitteln und Medikamenten), in der Landwirtschaft (Obst- und Gemüseernte, Düngung, Pflanzung), beim Militär sowie bei der Suche eingesetzt und Rettungseinsätze. Sie adressieren die Nachfrage nach flexibler Materialhandhabung, den Wunsch, dass Roboter große Strukturen bearbeiten können, und die Notwendigkeit einer schnellen Neukonfiguration von Arbeitsbereichen.

Obwohl sich mobile Roboter auf unterschiedliche Weise fortbewegen, liegt der Schwerpunkt in diesem Kapitel auf Geräten, die Räder zur Fortbewegung nutzen (Laufroboter werden in Kapitel 14 vorgestellt). Bei industriellen Anwendungen sind fahrerlose Transportfahrzeuge (FTS) von besonderem Interesse für den Materialtransport in einer Produktionsanlage oder einem Lager. Schlepper ziehen typischerweise Karren (Abb. 13.1a), Lader nutzen eine flache Plattform, um eine auf der Plattform gestapelte Ladeeinheit zu transportieren (Abb. 13.1b) und mobile Gabelstapler werden zum automatischen Aufnehmen und Absetzen von Lasten aus verschiedenen Höhen eingesetzt (Abb. 13.1c). AGVs folgen in der Regel Markierungen oder Drähten im Boden oder nutzen Sichtgeräte, Magnete oder Laser, um sich in der Anlage fortzubewegen. Diese organisierte Bewegung wird Navigation genannt; ein Prozess oder eine Aktivität zur Planung und Steuerung eines Roboters entlang einer Route oder eines Pfads, um sich sicher von einem Ort zum anderen zu bewegen, ohne sich zu verlaufen oder mit anderen Objekten zu kollidieren.

Navigation ist typischerweise eine komplexe Aufgabe, die aus Lokalisierung, Pfadplanung und Bewegungssteuerung besteht. Unter Lokalisierung versteht man die Fähigkeit des Roboters, seine eigene Position und Ausrichtung innerhalb des globalen Koordinatensystems festzulegen. Bei der autonomen Pfadplanung handelt es sich um die Bestimmung eines kollisionsfreien Pfads für einen Roboter zwischen Start- und Zielpositionen zwischen überfüllten Hindernissen in einem Arbeitsbereich. Dazu gehören auch Interaktionen zwischen mobilen Robotern und Menschen sowie zwischen Gruppen mobiler Roboter. Die Bewegungssteuerung muss die Ausführung der Bewegung entlang der geplanten Bahn bei gleichzeitiger Vermeidung von Hindernissen gewährleisten.

In kollaborativen Umgebungen teilen sich Menschen und Roboter einen Arbeitsplatz, was zu einem Bedarf an einer verbesserten Mensch-Roboter-Kommunikation und einer Sensibilisierung des Roboters für die Menschen in seiner Umgebung führt. Der Roboter muss typischerweise einen Sicherheitsabstand zu Menschen einhalten.

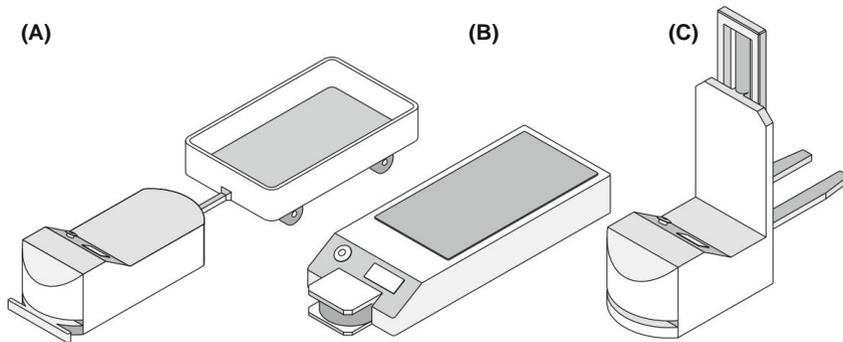


Abb. 13.1 Fahrerlose Transportfahrzeuge: a Schlepper, b Ladeeinheit und c mobiler Gabelstapler

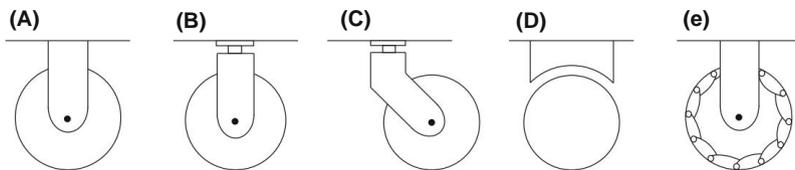


Abb. 13.2 Radausführungen: a Standard-Bockrad, b Standard-Lenkrad, c Lenkrolle, d sphärisches Rad und e schwedisches Rad

Körperpflegeroboter erfordern eine unmittelbare Nähe zwischen Mensch und Roboter
Diese Maschinen sind Beispiele für fortschrittliche interaktive Mensch-Roboter-Systeme.

13.1 Mobile Roboterkinematik

Aufgrund seines einfachen mechanischen Aufbaus ist das Rad der beliebteste Fortbewegungsmechanismus in der mobilen Robotik. Räder sorgen für Traktion und drei Räder garantieren Stabilität Roboterbalance. Räder können in verschiedenen Formen gestaltet sein, wie in Abb. 13.2 dargestellt.

Das Bockrad, das Standard-Lenkrad und das Lenkrad verfügen über ein Primärrad Drehachse und sind gerichtet. Eine Bewegung in eine andere Richtung ist nicht möglich ohne vorher das Rad um die Hochachse zu lenken. Das Kugelrad ist omnidirektional, da es eine Bewegung in alle Richtungen ermöglicht, ohne vorher zu lenken. Der Das schwedische Rad versucht mit angebrachten passiven Rollen ein omnidirektionales Verhalten zu erreichen um den Umfang des Rades herum. Dadurch kann sich das Rad unterschiedlich entlang bewegen Flugbahnen sowie vorwärts und rückwärts.

Auswahl des Radtyps, der Anzahl der Räder sowie deren Befestigung am Roboter Fahrgestelle beeinflussen die Kinematik mobiler Roboter erheblich. Beispiele für kinematische Designs sind in Abb. 13.3 dargestellt . Sie reichen von Zweirad- bis Vierradkonfigurationen. Die beiden Plattformen in der rechten Spalte sind omnidirektional.

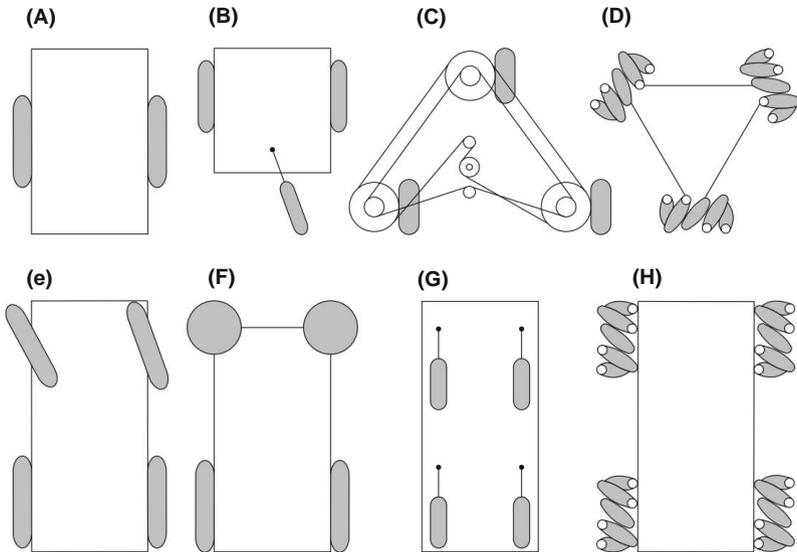


Abb. 13.3 Konfigurationsbeispiele für mobile Roboter: a Zweirad-Differentialantrieb, b Differentialantrieb mit Lenkrolle, c drei synchron motorisch gelenkten Rädern, d drei omnidirektionalen Rädern im Dreieck, e vier Räder mit autoähnlicher Lenkung, f zwei Differential-Antriebsräder und zwei omnidirektionale Räder, g vier motorisierte und gelenkte Lenkrollen und h vier omnidirektionale Räder in rechteckiger Konfiguration

Zu Analyse Zwecken wird ein mobiler Roboter als starrer Körper dargestellt
 Räder, die sich nur in einer horizontalen Ebene bewegen können. Mit diesen Annahmen die Pose
 Die Position des Roboters kann mit drei Koordinaten definiert werden, von denen zwei die Position im Roboter darstellen
 horizontale Ebene und eine, die die Ausrichtung um die vertikale Achse beschreibt. Beziehungen
 sind in Abb. 13.4 für einen einfachen Differentialantriebsmechanismus dargestellt. Achsen x_G und y_G
 Definieren Sie den globalen Koordinatenrahmen. Der lokale Koordinatenrahmen des Roboters wird mit definiert
 Achsen x_m und y_m . Die x_m - Achse zeigt in Vorwärtsrichtung des Roboters.

Position und Ausrichtung des Roboters werden mit dem folgenden Vektor definiert

$$\mathbf{x} = \begin{matrix} x \\ \dot{y}_j \\ \dot{\gamma} \end{matrix} \quad (13.1)$$

Dabei definieren die x - und y -Koordinaten die Roboterposition relativ zur globalen Koordinate
 Rahmen und Winkel $\dot{\gamma}$ bestimmen seine Ausrichtung (Rotation um die vertikale Achse). Roboter
 Die Orientierung kann auch in Form einer Rotationsmatrix beschrieben werden

$$\mathbf{R} = \begin{matrix} \cos \dot{\gamma} & \sin \dot{\gamma} & 0 \\ \sin \dot{\gamma} & \cos \dot{\gamma} & 0 \\ \dot{\gamma} & 0 & 1 \end{matrix} \quad (13.2)$$

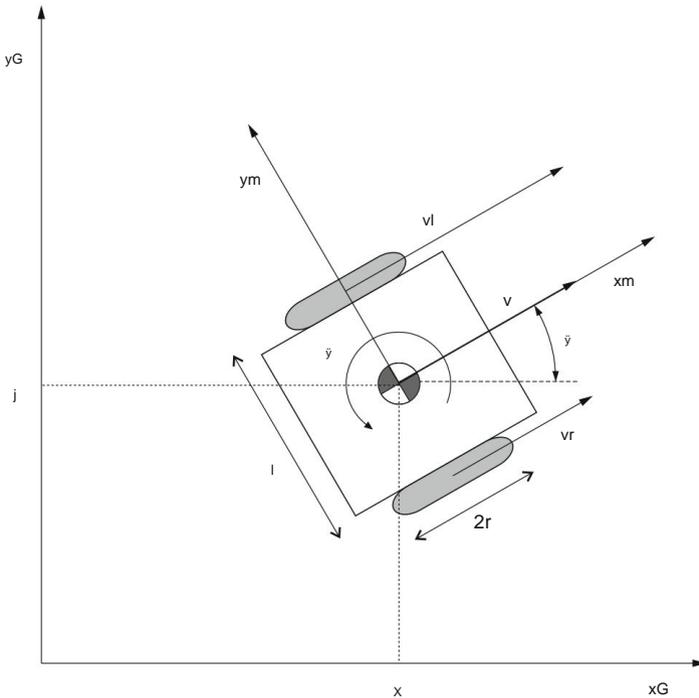


Abb. 13.4 Position und Ausrichtung eines mobilen Roboters – Beispiel eines Roboters mit Differentialantrieb

Es handelt sich dann um eine homogene Transformationsmatrix, die die Pose des mobilen Roboters beschreibt

$$T = \begin{pmatrix} \cos \bar{\gamma} & \sin \bar{\gamma} & 0 & x \\ -\sin \bar{\gamma} & \cos \bar{\gamma} & 0 & y \\ 0 & 0 & 1 & 0 \\ \dot{\bar{\gamma}} & 0 & 0 & 1 \end{pmatrix} \quad (13.3)$$

Der in Abb. 13.4 dargestellte Differentialantriebsroboter hat einen einfachen mechanischen Aufbau. Seine Bewegung basiert auf zwei separat angetriebenen Rädern, die auf beiden Seiten des Roboterkörpers angebracht sind. Der Roboter ändert seine Richtung, indem er die relative Rotationsgeschwindigkeit seiner Räder variiert. Somit ist keine zusätzliche Lenkbewegung erforderlich. Werden die Räder in die gleiche Richtung und mit gleicher Geschwindigkeit angetrieben, folgt der Roboter einer geraden Linie. Wenn die Räder mit gleicher Geschwindigkeit in entgegengesetzte Richtungen gedreht werden, dreht sich der Roboter um den Mittelpunkt zwischen den Rädern. Im Allgemeinen kann das Rotationszentrum des Roboters irgendwo auf der Linie durch die Radachsen liegen und hängt von der Rotationsgeschwindigkeit jedes Rads und seiner Richtung ab.

Mit seiner einfachen Kinematik ist es ein ideales Modell zur Untersuchung der Roboterbewegung. Durch die Darstellung der Roboterbreite (Abstand zwischen den Reifenkontaktpunkten mit dem Boden) mit l und dem Radradius mit r kann die Roboterbewegung analysiert werden.

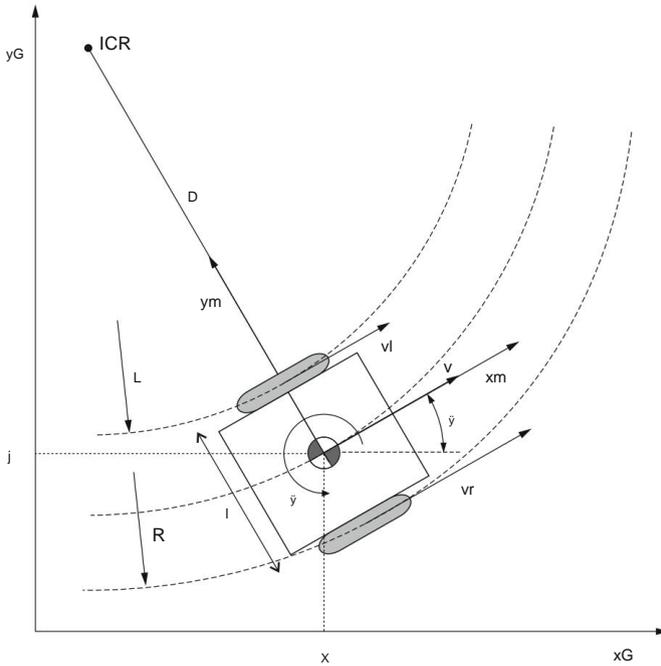


Abb. 13.5 Roboterkinematik mit Differentialantrieb

Winkelgeschwindigkeiten $\dot{\gamma}_r$ (rechtes Rad) und $\dot{\gamma}_l$ (linkes Rad), woraus sich die Radgeschwindigkeiten v_r bzw. v_l des rechten bzw. linken Rades ergeben

$$\begin{aligned} v_r &= \dot{\gamma}_r r, \\ v_l &= \dot{\gamma}_l r. \end{aligned} \quad (13.4)$$

Die beiden Raddrehungen führen zu einer Translationsgeschwindigkeit des Roboters entlang der x_m - Achse des Roboters und einer Winkelgeschwindigkeit um seine vertikale Achse. Unter Bezugnahme auf Abb. 13.5 kann die Winkelgeschwindigkeit definiert werden als:

$$\dot{\gamma} = \frac{v_l}{D - \frac{l}{2}} = \frac{v_r}{D + \frac{l}{2}}, \quad (13.5)$$

Dabei ist D der Abstand zwischen dem Mittelpunkt des Roboters (in diesem Fall dem Ursprung des Rahmens x_m - y_m) und dem Punkt, der das momentane Rotationszentrum (ICR) definiert. Der ICR ist der Punkt in der horizontalen Ebene, um den sich der Roboter zu einem bestimmten Zeitpunkt dreht. Aus der Gleichheit in (13.5) lässt sich folgende Beziehung ableiten

$$\dot{\gamma} = \frac{v_r \dot{\gamma} v_l}{l} = \frac{R}{-(\dot{\gamma}_r \dot{\gamma} \dot{\gamma}_l) \cdot l} \quad (13.6)$$

Die Translationsgeschwindigkeit entlang der x_m - Achse kann dann bestimmt werden als

$$v = \ddot{y}D = 2 \frac{v_r + v_l}{2r} = -\frac{R}{r}(\ddot{y}_r + \ddot{y}_l). \quad (13.7)$$

Die Gleichungen (13.6) und (13.7) definieren Beziehungen zwischen den Winkelgeschwindigkeiten der Räder und der Geschwindigkeit des mobilen Roboters. Aus steuerungsstechnischer Sicht ist es jedoch die relevantere umgekehrte Beziehung, die die Winkelgeschwindigkeiten der Räder gegenüber der gewünschten Robotergeschwindigkeit. Durch Kombination von (13.6) und (13.7) erhält man die folgenden Beziehungen

$$\begin{aligned} \ddot{y}_r &= \frac{2v + \ddot{y}l}{2r}, \\ \ddot{y}_l &= \frac{2v - \ddot{y}l}{2r}. \end{aligned} \quad (13.8)$$

Die als Paar $[v, \ddot{y}]$ bestimmte Robotergeschwindigkeit wird relativ zum lokalen Koordinatensystem des mobilen Roboters x_m-y_m definiert. Die Robotergeschwindigkeit im globalen Koordinatensystem x_G-y_G , definiert als zeitliche Ableitung des Roboterhaltungsvektors x (13.1), kann durch Drehen der lokal ausgedrückten Geschwindigkeit mithilfe der Rotationsmatrix R (13.2) bere

$$\begin{pmatrix} \cos \ddot{y} & \ddot{y} \sin \ddot{y} & 0 \\ \sin \ddot{y} & \cos \ddot{y} & 0 \\ \ddot{y} & 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \ddot{y} \\ 0 \end{pmatrix} = \begin{pmatrix} v \cos \ddot{y} \\ v \sin \ddot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \cos \ddot{y} & \ddot{y} \sin \ddot{y} & 0 \\ \sin \ddot{y} & \cos \ddot{y} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ \ddot{y} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \ddot{y} \\ 0 \end{pmatrix}. \quad (13.9)$$

Durch die Kombination von Translations- und Rotationsteilen der obigen Gleichungen und das Weglassen von Elementen, die Null sind, kann die Geschwindigkeit des mobilen Roboters im globalen Koordinatensystem wie folgt geschrieben werden:

$$\dot{x} = \begin{pmatrix} X \\ y \end{pmatrix} = \begin{pmatrix} v \cos \ddot{y} \\ v \sin \ddot{y} \\ \ddot{y} \end{pmatrix}. \quad (13.10)$$

Aus Gl. (13.10) ist klar, dass relevante Größen zur Beschreibung der Bewegung mobiler Roboter die Translationsgeschwindigkeit entlang der x_m - Achse v des Roboters, die Rotationsgeschwindigkeit um die vertikale Achse \ddot{y} und die Roboterorientierung in Bezug auf den globalen Koord Vor diesem Hintergrund können wir den Differentialantriebsroboter weiter zu einem Einradmodell vereinfachen (wie in Abb. 13.6 dargestellt). Die oben genannten drei Größen beschreiben nun die Bewegung des Einrads, das in Abb. 13.6 als einzelnes Rad mit markierter Vorwärtsrichtung in der Mitte des Differentialantriebsroboters dargestellt ist. Das Einrad lässt sich anhand von Gl. leicht wieder in einen Roboter mit Differentialantrieb

Die attraktive Eigenschaft des Einradmodells ist seine Einfachheit. Daher wird es in diesem Kapitel zur Analyse verwendet. Generell lässt sich das Modell jedoch auch auf jeden anderen kinematisch komplexeren mobilen Roboter umrüsten. Als Beispiel betrachten wir eine mobile Plattform, die auf dem in Abb. 13.7 gezeigten Autolenkungsprinzip basiert

Die Lenkgeometrie des Autos löst das Problem, dass Räder auf der Innen- und Außenseite einer Kurve Kreise mit unterschiedlichen Radien verfolgen müssen. Daher sind die Lenkwinkel der linken und rechten Vorderräder unterschiedlich. Im Einradmodell wird die Ausrichtung des Einrads durch den Winkel \ddot{y} definiert, genau wie die Ausrichtung des Differentialantriebsroboters.

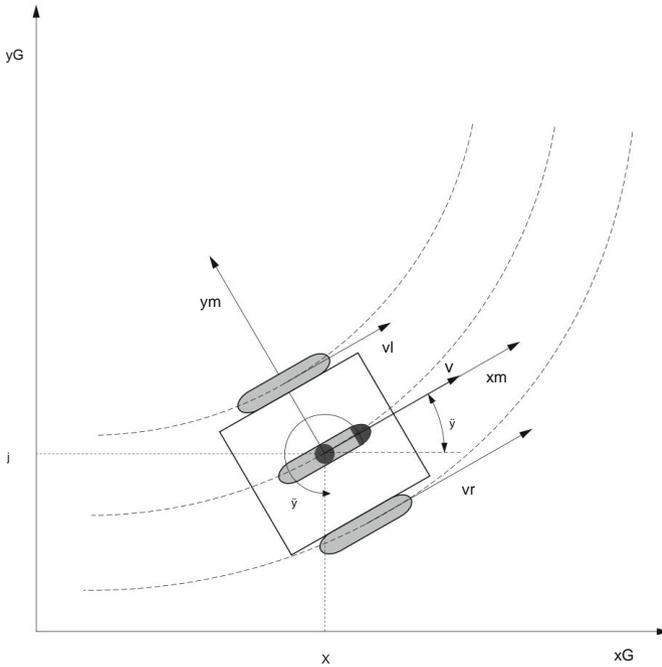


Abb. 13.6 Einradmodell eines mobilen Roboters mit Differenzialantrieb

Beim autoähnlichen Problem wird die Ausrichtung des mobilen Roboters durch den Winkel \tilde{y} definiert. Das Einradmodell wird in der Mitte der Vorderräder positioniert und seine Ausrichtung ist so definiert, dass das gleiche momentane Drehzentrum erreicht wird, das durch die Ausrichtung der linken und rechten Räder des Autos definiert ist. Das Einrad ist nun das dritte Vorderrad und der ICR ist am Schnittpunkt aller drei Linien senkrecht zu den Vorderrädern positioniert. Der Winkel \tilde{y} ist nun als Abweichung der Einradausrichtung von der x_m -Achse des Roboters definiert (wie in Abb. 13.7 dargestellt). Durch die Berechnung des Winkels \tilde{y} wird die Beziehung zwischen dem autoähnlichen Problem

Indem man dem gleichen Prinzip wie in (13.7) folgt, wird die Translationsgeschwindigkeit des Einrads bestimmt kann definiert werden als

$$v = D\dot{\tilde{y}}, \quad (13.11)$$

wobei D der Abstand zwischen dem Einrad und dem ICR ist. Der Abstand D kann dann berechnet werden als:

$$D = \frac{v}{\dot{\tilde{y}}}. \quad (13.12)$$

Die Bahnkrümmung für das Einrad K_u kann als Kehrwert des momentanen Rotationsradius D definiert werden

$$K_u = \frac{1}{D} = \frac{\dot{\tilde{y}}}{v}. \quad (13.13)$$

Schließlich ist der Winkel \tilde{y} gleich

$$\tilde{y} = \arcsin \frac{\tilde{y}l}{v} \quad (13.18)$$

Der Winkel \tilde{y} ist der gewünschte Lenkwinkel für das Auto und kann aus der bekannten Geschwindigkeit v , der Winkelgeschwindigkeit \tilde{y} und der Breite des Autos l berechnet werden.

Mit der definierten Beziehung zwischen dem Einrad und einem mobilen Roboter mit anderer Kinematik kann die Analyse auf einem einfachen Einradmodell basieren und auf den anderen Roboter verallgemeinert werden.

13.2 Navigation

Mobile Roboter agieren oft in unbekanntem und unstrukturierten Umgebungen und müssen sich selbst lokalisieren, einen Weg zu einem Ziel planen, eine Karte der Umgebung erstellen und interpretieren und dann ihre Bewegung durch diese Umgebung steuern.

13.2.1 Lokalisierung

Ein wichtiger Unterschied zwischen einem Manipulator und einem mobilen Roboter besteht in der Positionsschätzung. Ein Manipulator hat eine feste Basis und durch die Messung der Gelenkpositionen des Roboters und die Kenntnis seines kinematischen Modells ist es möglich, die Stellung seines Endeffektors zu bestimmen. Ein mobiler Roboter kann sich als Einheit durch die Umgebung bewegen und es gibt keine direkte Möglichkeit, seine Position und Ausrichtung zu messen. Eine allgemeine Lösung besteht darin, die Position und Ausrichtung des Roboters durch Integration der Bewegung (Geschwindigkeit) über die Zeit abzuschätzen.

Typischerweise sind jedoch genauere und oft auch komplexere Ansätze erforderlich. Wenn die Karte der Umgebung im Voraus bekannt ist, können mobile Roboterwege im Voraus geplant werden. Dies ist besonders nützlich, wenn die Umgebung relativ statisch ist und ein robuster Betrieb erforderlich ist, beispielsweise in industriellen Anwendungen. Komplexere Ansätze basieren auf einer dynamischen Pfadplanung auf Basis von Sensorinformationen und der Erkennung von Merkmalen in der Umgebung. Der Roboter ermittelt zunächst seine eigene Position und plant seine Bewegung durch abfahrbare Bereiche. Wenn sich der Arbeitsbereich oder die Aufgaben häufig ändern, ist es in der Regel besser, dynamisch zu planen. Oft ist ein Kompromiss zwischen Vorplanung und dynamischer Planerstellung erforderlich. Um die Aufgabe zu vereinfachen, können Markierungen in der Umgebung platziert werden. Diese Markierungen können von Sensoren am Roboter leicht erkannt werden und ermöglichen eine genaue Lokalisierung.

Fahrerlose Transportfahrzeuge in Industrieumgebungen nutzen verschiedene Navigations-/Leittechnologien: Magnetband, Draht, Magnetpunkt, Laser und natürliche.

Die Ortung und Wegplanung basiert häufig auf in den Boden eingelassenen stromführenden Leitungen mittels induktiver Führung. Am Fahrzeug ist ein Führungswegsensor montiert.

Der Draht kann durch Magnetband oder eine aufgemalte Linie ersetzt werden (Abb. 13.8a). In Letzterem

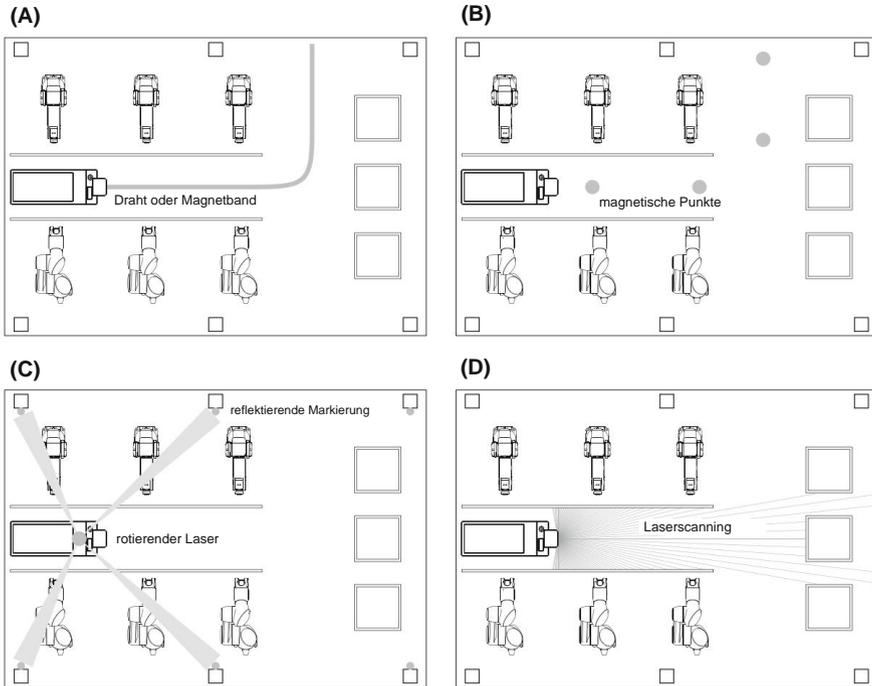


Abb. 13.8 Sensorabstraktionsscheibe aus der Sensoranordnung an Bord des Roboters

In diesem Fall ermittelt der Roboter mithilfe einer Kamera seine relative Position zur Bodenlinie. Die Wege sind fest und durchgehend. Entlang der Linie können eindeutige Markierungen angebracht werden, um bestimmte Positionen anzuzeigen. Anstatt Linien und Markierungen auf dem Boden anzubringen, können Markierungen (zweidimensionale Muster) auch an der Decke angebracht werden, um von einer integrierten Kamera identifiziert zu werden. Die magnetische Punktführung nutzt einen mit magnetischen Punkten versehenen Weg. Die Wege sind offen und wandelbar.

Bodenbasierte Lokalisierungstechniken werden häufig durch laserbasierte Methoden ersetzt. Lasertriangulationsmethoden, bei denen ein rotierender Laser die Entfernung und den Azimut von an der Wand montierten Reflektoren erfasst, liefern genaue Lokalisierungsinformationen, ohne dass bestimmte Linien auf dem Boden verfolgt werden müssen. Bei der Laserführungstechnologie werden mehrere feste Referenzpunkte (Reflexstreifen) im Arbeitsbereich verwendet, die von einem am Fahrzeug montierten Laserkopf erfasst werden können (Abb. 13.8c). Da die Anlage vorab kartiert wird, können Wege leicht geändert und erweitert werden.

Die natürliche Navigation basiert auf Informationen der vorhandenen Umgebung, die mit Hilfe von Laserscannern mithilfe einiger fester Referenzpunkte erfasst werden (Abb. 13.8d). Das Gebiet wird im Voraus kartiert. Natürliche Navigation ist flexibel und entbehrlich. Es eignet sich für Umgebungen, die sich häufig, aber nicht wesentlich ändern. In engen Räumen kann der Roboter ausgehend von der Wand der Wand durch die Umgebung folgen.

Außerdem werden funkbasierte Indoor-Ortungssysteme eingeführt, die eine Roboterlokalisierung ähnlich wie das Outdoor-Global-Positioning-System ermöglichen. Die Lokalisierung basiert auf der Triangulation mit fest in der Anlage montierten Beacons und dem am Roboter montierten Sensor. Entfernungen werden berechnet, indem die Laufzeit der Funkwellen vom Beacon zum Sensor gemessen wird.

13.2.1.1 Odometrie

Ein einfacher und häufig verwendeter Ansatz zur Roboterlokalisierung besteht darin, sich auf die Odometrie zu verlassen, die Informationen von Bewegungssensoren (typischerweise Radencodern) nutzt, um Positionsänderungen im Laufe der Zeit abzuschätzen. Diese Positionsänderungen werden mithilfe von Integrationsprinzipien akkumuliert und liefern die Position des Roboters relativ zu einem Startort. Die Methode ist fehleranfällig, da Geschwindigkeitsmessungen über die Zeit hinweg integriert werden, um Positionsschätzungen zu erhalten.

Die Analyse der Roboterbewegung beginnt mit dem Verständnis des Beitrags jedes Rads zur Geschwindigkeit des Roboters. Für den speziellen Fall eines Roboters mit Differenzialantrieb sind diese Beziehungen in (13.6) und (13.7) definiert. Die Radgeschwindigkeit kann direkt mit einem Drehzahlmesser gemessen werden. Wenn ein solcher Sensor nicht verfügbar ist, kann die Geschwindigkeit durch numerische Differenzierung der von E in einem solchen Fall können die Geschwindigkeiten für das rechte und linke Rad berechnet werden als:

$$\begin{aligned} &= 2\ddot{y}_r \frac{nr(t) \ddot{y} \quad nr(t) \ddot{y} \quad \ddot{y}t) \quad vr}{N\dot{y}t}, \\ &= 2\ddot{y}_r \frac{nl(t) \ddot{y} \quad nl(t) \ddot{y} \quad \ddot{y}t) \quad vl}{N\dot{y}t}, \end{aligned} \quad (13.19)$$

Dabei ist r der Radradius, N die Encoderauflösung in Form von Zählungen pro Umdrehung, nr und nl die Encoderzählungen des rechten bzw. linken Rads zum Zeitpunkt t und $nr(t) \ddot{y} \quad \ddot{y}t)$ und $nl(t) \ddot{y} \quad \ddot{y}t)$ sind die gleichen Größen zum vorherigen Abtastzeitpunkt

Anschließend können die Position und Ausrichtung des Roboters durch numerische Integration geschätzt werden von Gl. (13.10) und Berücksichtigung von (13.6) und (13.7) als

$$\begin{aligned} x(t) &= x(t) \ddot{y} \quad \ddot{y}t) + v \cos \ddot{y} \quad \ddot{y}t) = x(t) \ddot{y} \quad \ddot{y}t) + \cos \ddot{y} \quad \ddot{y}t), \frac{vr + vl}{2} \\ vl \quad y(t) &= y(t) \ddot{y} \quad \ddot{y}t) + v \sin \ddot{y} \quad \ddot{y}t) = x(t) \ddot{y} \quad \ddot{y}t) + \sin \ddot{y} \quad \ddot{y}t), \frac{2 \quad vr +}{2} \\ \ddot{y}(t) &= \ddot{y}(t) \ddot{y} \quad \ddot{y}t) + \ddot{y} \quad \ddot{y}t) = \ddot{y}(t) \ddot{y} \quad \ddot{y}t) + \ddot{y}t). \frac{vr \quad \ddot{y} \quad vl}{1} \end{aligned} \quad (13.20)$$

Verschiedene Faktoren verringern die Wirksamkeit odometriebasierter Methoden zur Roboterpositionsschätzung. Ein sehr wichtiger Faktor ist der Radschlupf, der die Genauigkeit der Positionsschätzung erheblich verringert. Die Leistung kann durch die Verwendung von Modellen der Fehler und des Fahrzeugs verbessert werden. Bodenflecken oder Magnete können verwendet werden, um Odometriefehler zu korrigieren, die sich zwischen diesen Punkten ansammeln

durch sensorbasierte Messungen von Lasern, Kameras, Radiofrequenz-Identifikationssystemen und Beacons ergänzt werden.

13.2.1.2 Gleichzeitige Lokalisierung und Kartierung

Fortgeschrittenere Systeme nutzen Algorithmen, die die Teilaufgaben der Navigation (Lokalisierung, Pfadplanung) gleichzeitig ausführen. Der Ansatz, der sich mit dem Problem befasst, durch einen mobilen Roboter eine Karte einer unbekanntenen Umgebung zu erstellen und gleichzeitig mithilfe der Karte durch die Umgebung zu navigieren, wird als Simultaneous Localization and Mapping (SLAM) bezeichnet. Durch die Beobachtung derselben Merkmale in mehreren Ansichten mithilfe von Sensoren, die sich mit dem Fahrzeug bewegen, sammelt der SLAM-Algorithmus die Sensorinformationen und kombiniert sie. Durch die Kombination der Roboterpositionsschätzung mit den gesammelten Informationen kann eine lokale Karte erstellt werden, indem die verfügbaren Daten zusammengefügt werden. Im Laufe der Zeit kann die komplette Umgebung kartiert werden und die Karten können zur Planung der Roboterpfade verwendet werden.

SLAM besteht aus mehreren Teilen, wie z. B. Landmark-Extraktion, Datenzuordnung, Zustandsschätzung, Zustandsaktualisierung und Landmark-Aktualisierung. Es gibt viele Möglichkeiten, die einzelnen kleineren Teile zu lösen, diese gehen jedoch über den Rahmen dieses Buches hinaus.

13.2.1.3 Sensor-Abstraktions-Disk

Wenn sich der mobile Roboter durch die Umgebung bewegt, muss er auch seine Umgebung beobachten. Sensoren an Bord des Roboters suchen nach Hindernissen oder unerwarteten Objekten auf dem Weg des Fahrzeugs und der Roboter kann möglicherweise einen Weg um sie herum planen, bevor er zur vorgeplanten Route zurückkehrt. Eine typische Suite von Sensoren umfasst Infrarot-Näherungssensoren, Ultraschall-Abstandssensoren, Laserscanner, Bildverarbeitungs-, Tastsensoren und globale Positionierungssensoren. Sensoren sind strategisch an Bord des Roboters und rund um seinen Umfang platziert. Jeder Sensor liefert unterschiedliche Informationen hinsichtlich Quantität, Qualität, Reichweite und Auflösung. Typischerweise werden jedoch die Informationen aller Sensoren kombiniert, um ein genaues Bild der Roboterumgebung zu liefern. Ohne sich speziell mit der Analyse einzelner Sensoren und der Integration sensorischer Informationen zu befassen, kann man davon ausgehen, dass Entfernung und Richtung zu allen Hindernissen aus Sicht des Roboters aus der Sensorsuite ermittelt werden können. Die in Abb. 13.9 dargestellte Sensorabstraktionsscheibe ist ein Beispiel für sensorische Integration, die Informationen über Hindernisse im Radius der Scheibe um d

Aus der bekannten Position d_0 und Orientierung \tilde{y}_0 des Hindernisses und dem bekannten ist es möglich, Pose des Roboters $[x, y, \tilde{y}]^T$, die Hindernisposition (x_0, y_0) zu bestimmen im globalen Koordinatensystem als

$$\begin{aligned} x_0 &= x + d_0 \cos(\tilde{y} + \tilde{y}_0), & y_0 &= \\ y_0 &= y + d_0 \sin(\tilde{y} + \tilde{y}_0). \end{aligned} \tag{13.21}$$

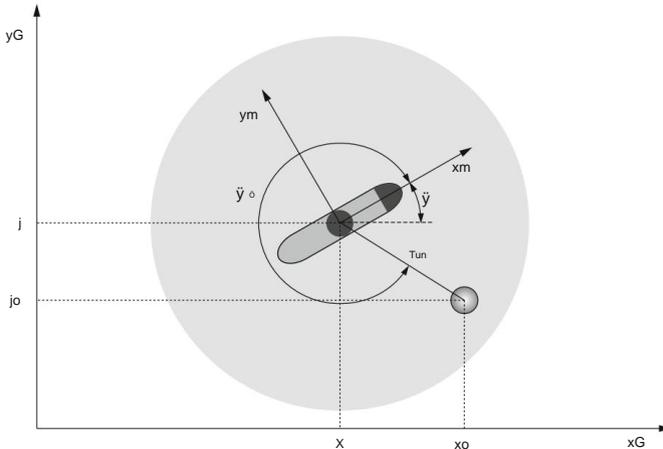


Abb. 13.9 Sensorabstraktionsscheibe aus der Sensoranordnung an Bord des Roboters

Die folgende Analyse basiert auf den Annahmen eines Einradrobotermodells und die von der Sensorabstraktionsscheibe erhaltenen Informationen über Objekte.

13.2.2 Pfadplanung

Durch die Bahnplanung können autonome mobile Roboter einen optimalen kollisionsfreien Weg von der Startposition zum Ziel verfolgen, ohne mit Hindernissen im Arbeitsbereich zu kollidieren. Ein idealer Pfadplaner muss in der Lage sein, mit Unsicherheiten im erfassten Weltmodell umzugehen, die Auswirkungen von Objekten auf den Roboter zu minimieren und den optimalen Pfad in kürzester Zeit zu finden, insbesondere wenn der Pfad regelmäßig befahren werden soll. Im Allgemeinen sollte die Pfadplanung den Pfad mit den geringstmöglichen Kosten ergeben, schnell und robust sowie generisch in Bezug auf verschiedene Karten sein.

Für die (Echtzeit-)Pfadplanung stehen verschiedene Algorithmen zur Verfügung. Eine einfache Methode besteht darin, geradlinige Segmente zu kombinieren, die mit Eckpunkten verbunden sind. Eine weitere Standardsuchmethode zum Finden des optimalen Pfades ist der A*-Algorithmus mit seinen Modifikationen. Der Algorithmus findet einen gerichteten Pfad zwischen mehreren Punkten. Die durch eine Karte dargestellte Roboterumgebung kann in freie und belegte Räume zerlegt werden. Anschließend kann eine A*-Suche durchgeführt werden, um einen stückweise linearen Pfad durch die freien Knoten zu finden.

Zur Hindernisvermeidung kann ein künstlicher Potentialfeldalgorithmus eingesetzt werden. Der Algorithmus nutzt abstoßende Potenzialfelder um die Hindernisse herum, um den Roboter, der diesem Potenzial ausgesetzt ist, wegzudrängen und nutzt ein attraktives Potenzialfeld um das Ziel, um den Roboter dazu zu bewegen, zum Ziel zu gehen. Abstoßende und anziehende Felder verändern die Flugbahn des Roboters. Der Algorithmus ermöglicht den Echtzeitbetrieb eines mobilen Roboters in einer komplexen Umgebung.

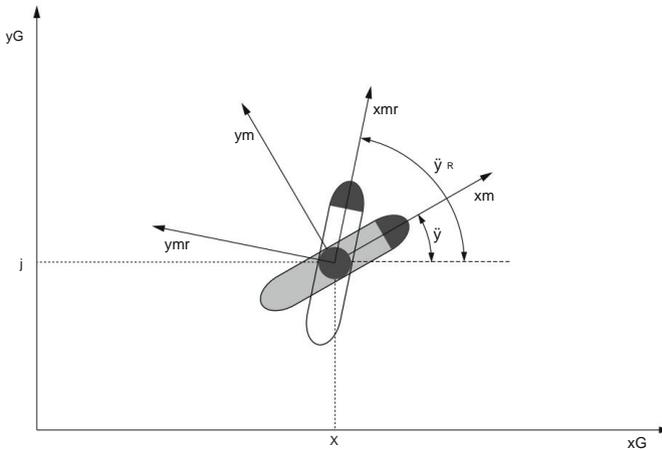


Abb. 13.10 Orientierungskontrolle des Einrads; Das graue Einrad stellt den tatsächlichen Roboter dar und das weiße Einrad stellt die gewünschte Ausrichtung dar

13.2.3 Pfadsteuerung

Um die Aufgabe zu erledigen, muss sich der mobile Roboter von seinem ursprünglichen Standort in die gewünschte Endposition und Ausrichtung bewegen. Zur Steuerung des Fahrzeugs auf seinem Weg ist ein Steuerungssystem erforderlich.

13.2.3.1 Kontrolle der Orientierung

Basierend auf dem in Abb. 13.10 dargestellten Einradmodell wird zunächst die Steuerung der Orientierung betrachtet. Ein ähnlicher Ansatz würde für mobile Roboter gelten, die ihre Ausrichtung ändern können, ohne ihre Position zu ändern (ein Roboter mit Differenzialantrieb ist ein solches Fahrzeug, das Auto jedoch nicht).

Das Kontrollziel besteht darin, den Orientierungsfehler zu minimieren

$$\ddot{\gamma} = \ddot{\gamma}_r - \ddot{\gamma}, \quad (13.22)$$

Dabei ist $\ddot{\gamma}_r$ die gewünschte Ausrichtung und $\ddot{\gamma}$ die tatsächliche Ausrichtung. Wir gehen davon aus, dass die Regelung auf dem Proportional-Integral-Derivativ-Regelungsansatz (PID) basiert

$$\text{PID}(\ddot{\gamma}) = K_p \ddot{\gamma} + K_i \int \ddot{\gamma} dt + K_d \dot{\ddot{\gamma}} \quad (13.23)$$

oder eine seiner Subversionen, wie etwa der Proportional-Differential-Regler. Dann kann die gewünschte Winkelgeschwindigkeit des mobilen Roboters berechnet werden als:

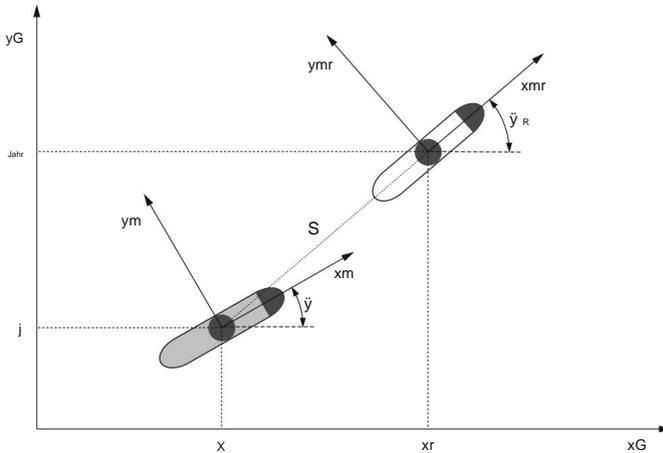


Abb. 13.11 Positions- und Orientierungskontrolle des Einrads; Das graue Einrad repräsentiert den tatsächlichen Roboter und das weiße Einrad den Zielort

$$\ddot{\gamma} = K_p \ddot{\gamma} + K_i \dot{\gamma} + K_d \dot{\gamma} \quad (13.24)$$

Es ist zu beachten, dass Winkel periodische Funktionen sind und wenn wir von einer Konfiguration ausgehen

$$\dot{\gamma}_r = 0 \quad \dot{\gamma} = 2\dot{\gamma} \quad \ddot{\gamma} = \ddot{\gamma}_r \quad (13.25)$$

Der Roboter dreht sich einmal, bevor er die endgültige Ausrichtung erreicht. Dies ist normalerweise kein wünschenswertes Roboterverhalten. Daher muss der Orientierungsfehler so begrenzt werden, dass eine Drehung mit maximalem $\dot{\gamma}$ -Radiant in beide Richtungen erforderlich ist

$$\ddot{\gamma} = \ddot{\gamma} [\ddot{\gamma}_r, \dot{\gamma}_r] \quad (13.26)$$

Eine einfache Lösung besteht darin, eine Vier-Quadranten-Arctan-Funktion als zu verwenden

$$\ddot{\gamma} = \arctan(\sin \ddot{\gamma}_r, \cos \ddot{\gamma}_r) \ddot{\gamma} [\ddot{\gamma}_r, \dot{\gamma}_r] \quad (13.27)$$

Mit der Kombination von (13.27) und (13.24) erreicht der Roboter die gewünschte Ausrichtung, ohne mehr als einen halben Kreis in positiver oder negativer Richtung zu drehen.

13.2.3.2 Kontrolle von Position und Ausrichtung

Der mobile Roboter bewegt sich normalerweise von seinem ursprünglichen Standort zu seinem endgültigen (Ziel-)Standort, was eine Änderung seiner Position und Ausrichtung erfordert. Da sich der Roboter zu seinem Zielort bewegen muss, bezeichnen wir diese Aufgabe als „Go-to-Goal“. Abbildung 13.11 stellt

Bedingungen. Der Koordinatenrahmen $x_m - y_m$ definiert die aktuelle Pose des Roboters und der Rahmen $x_{mr} - y_{mr}$ definiert die Zielpose. Das Liniensegment S stellt den kürzesten Weg zur Erledigung der Aufgabe dar.

Die gewünschte Roboterausrichtung zur Erledigung der Aufgabe kann als Winkel zwischen dem Liniensegment S und der horizontalen Achse des globalen Koordinatenrahmens definiert werden. Mit der bekannten gewünschten Position (x_r, y_r) und der aktuellen Position des Roboters (x, y) kann der Winkel $\dot{\gamma}_r$ zu jedem Zeitpunkt während der Roboterbewegung berechnet werden als

$$\dot{\gamma}_r = \arctan \frac{y_r \dot{y} - y \dot{y}_r}{x_r \dot{x} - x \dot{x}_r}. \quad (13.28)$$

Unter der Annahme, dass sich der Roboter mit konstanter Vorwärtsgeschwindigkeit v_0 bewegt, kann die Roboterbewegung im globalen Koordinatensystem mit dem folgenden Gleichungssatz beschrieben werden

$$\begin{aligned} \dot{x} \dot{y} &= v_0 \cos \gamma_r \\ \ddot{y}, \dot{y} \ddot{y} &= v_0 \dot{\gamma}_r \sin \gamma_r \\ \sin \dot{\gamma}_r, \ddot{\gamma}_r &= \dot{\gamma}_r = \text{PID}(\ddot{\gamma}_r). \end{aligned} \quad (13.29)$$

Bei diesem Ansatz besteht das Steuerungsziel darin, die Geschwindigkeit v_0 konstant zu halten und dem aus (13.28) berechneten gewünschten Winkel $\dot{\gamma}_r$ zu folgen. Wenn wir einen Roboter mit Differenzialantrieb annehmen, können die Radwinkelgeschwindigkeiten aus (13.8) wie folgt berechnet werden

$$\begin{aligned} \dot{\gamma}_r &= \frac{2v_0 + \dot{\gamma}_l}{2r}, \\ \dot{\gamma}_l &= \frac{2v_0 \dot{\gamma}_r \dot{\gamma}_l}{2r}. \end{aligned} \quad (13.30)$$

Bei einer Bewegung mit konstanter Geschwindigkeit v_0 würde der Roboter über seinen Zielort hinauschießen. Daher ist es sinnvoll, die Vorwärtsgeschwindigkeit des Roboters basierend auf der Entfernung zum Ziel zu definieren

$$G = \sqrt{(x_r - x)^2 + (y_r - y)^2}. \quad (13.31)$$

Mit einem Proportionalregler kann die gewünschte Geschwindigkeit definiert werden

$$vG = KvG, \quad (13.32)$$

wobei Kv die Geschwindigkeitsverstärkung ist. Die Gleichungen (13.29) können dann umgeschrieben werden als

$$\begin{aligned} \dot{x} \dot{y} &= vG \cos \gamma_r \\ \ddot{y}, \dot{y} \ddot{y} &= vG \dot{\gamma}_r \sin \gamma_r \\ \sin \dot{\gamma}_r, \ddot{\gamma}_r &= \dot{\gamma}_r = \text{PID}(\ddot{\gamma}_r) \end{aligned} \quad (13.33)$$

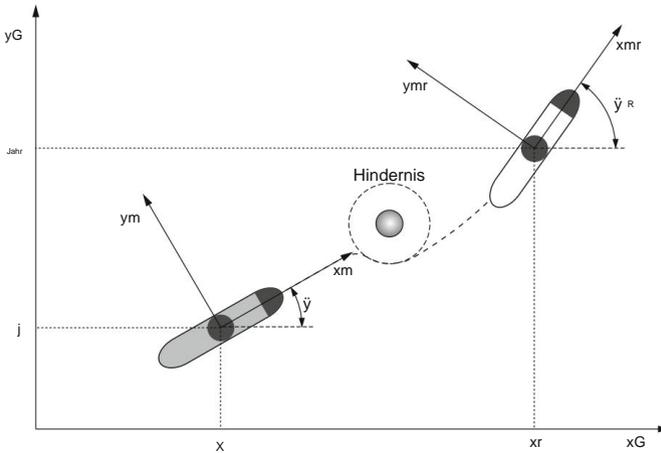


Abb. 13.12 Positions- und Orientierungskontrolle des Einrads mit Hindernisvermeidung; Das graue Einrad stellt den tatsächlichen Roboter dar und das weiße Einrad den Zielort. Der graue Kreis ist das Hindernis und die gestrichelte Kreislinie ist die sichere Zone um das Hindernis herum

und in (13.30) muss v_0 durch v_G ersetzt werden. Bei diesem Ansatz bremsst der Roboter ab, wenn er sich dem Zielort nähert. Da die gewünschte Geschwindigkeit mit der Entfernung zum Ziel zunimmt, kann eine maximale Grenze für $v_G \dot{y} [0, v_{Gmax}]$ festgelegt werden.

13.2.3.3 Hindernisvermeidung

Abbildung 13.12 zeigt Bedingungen mit einem Hindernis auf dem Weg des Roboters zur Zielposition. Der Roboter kann nicht direkt zum Zielort vordringen, ohne zuvor dem Hindernis auszuweichen. Basierend auf dem Konzept der Sensor-Abstraktionsscheibe gehen wir davon aus, dass der Roboter in der Lage ist, das Hindernis aus sicherer Entfernung zu erkennen und zu lokalisieren und anhand dieser Informationen Vermeidungsaktivitäten zu planen. Das Hindernis in Abb. 13.12 wird durch einen grauen Kreis dargestellt und die gestrichelte Kreislinie um das Hindernis stellt eine sichere Zone um das Hindernis herum dar. Der Roboter durfte den gestrichelten Kreis nicht betreten.

Vor diesem Hintergrund haben wir nun zwei Kontrollziele. Die erste ist das Ziel erreichen und die zweite das Vermeiden von Hindernissen. Eine detailliertere Darstellung der beiden Kontrollziele ist in Abb. 13.13 dargestellt, wobei d_0 den Abstand vom Roboter zum Hindernis angibt, u_G die Kontrollvariable ist, die dem Ziel „Ziel erreichen“ zugeordnet ist, und u_0 die Kontrollvariable ist, die der Vermeidung zugeordnet ist -Hindernisziel. Um die Aufgabe erfolgreich abzuschließen, muss der u_G auf das Ziel zeigen, während der u_0 vom Hindernis weg zeigen muss. Die eigentliche Regelgröße u ergibt sich aus der Mischung von u_G und u_0 .

Der Zielsteuerungsteil kann basierend auf der Entfernung zur Zielposition definiert werden als

$$\begin{matrix} u_{Gx} \\ \text{hässlich} \end{matrix} = k_G \begin{matrix} x_r - \dot{y}_x \\ y_r - \dot{y}_y \end{matrix} \quad (13.34)$$

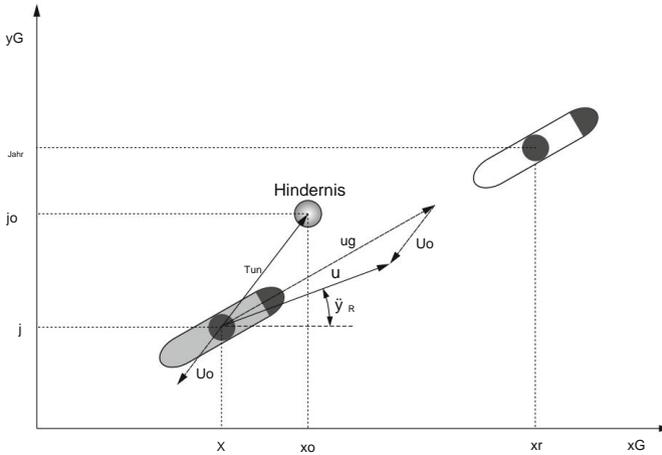


Abb. 13.13 Einrad-Hindernisvermeidung; Das graue Einrad stellt den tatsächlichen Roboter dar, das weiße Einrad den Zielort und der graue Kreis das Hindernis

Ebenso kann die Steuervariable „Hindernis vermeiden“ basierend auf der Entfernung zum Hindernis definiert werden

$$\begin{aligned} u_{ox} &= K_o \frac{x - \tilde{y} x_o}{x - \tilde{y} x_o + y - \tilde{y} y_o} \\ u_{oy} &= K_o \frac{y - \tilde{y} y_o}{x - \tilde{y} x_o + y - \tilde{y} y_o} \end{aligned} \quad (13.35)$$

Es ist zu beachten, dass u_g auf das Ziel zeigt und u_o vom Hindernis weg zeigt, wie aus der Definition der Entfernungen in den beiden obigen Gleichungen hervorgeht. Die Mischung der beiden Steuervariablen muss basierend auf der Entfernung zum Hindernis erfolgen, die als definiert ist

$$d_o = \sqrt{(x_o - \tilde{y} x)^2 + (y_o - \tilde{y} y)^2} \quad (13.36)$$

Wenn der Roboter weit vom Hindernis entfernt ist, muss er sich nur direkt zum Ziel begeben. In der Nähe des Hindernisses besteht die Hauptaufgabe jedoch darin, dem Hindernis auszuweichen. Konsekutiv kann das Mischen implementiert werden als

$$\begin{aligned} u_x &= \tilde{y}(d_o) u_{gx} + (1 - \tilde{y}(d_o)) u_{ox} \\ u_y &= \tilde{y}(d_o) u_{gy} + (1 - \tilde{y}(d_o)) u_{oy} \end{aligned}, \quad \tilde{y}(d_o) \in [0, 1]. \quad (13.37)$$

Parameter \tilde{y} kann beispielsweise als Exponentialfunktion basierend auf der Entfernung $\tilde{y} d_o$ der Funktion $\tilde{y} = 1 - e^{-\tilde{y} d_o}$ definiert werden und Parameter \tilde{y} definiert die Konvergenzrate Hindernis in Richtung 1. Wie aus Abb. 13.13 ersichtlich, definiert die Steuervariable u den gewünschten Roboter Geschwindigkeiten im globalen Koordinatensystem

$$\begin{aligned} \dot{x} &= u_x \\ \dot{y} &= u_y \end{aligned} \quad (13.38)$$

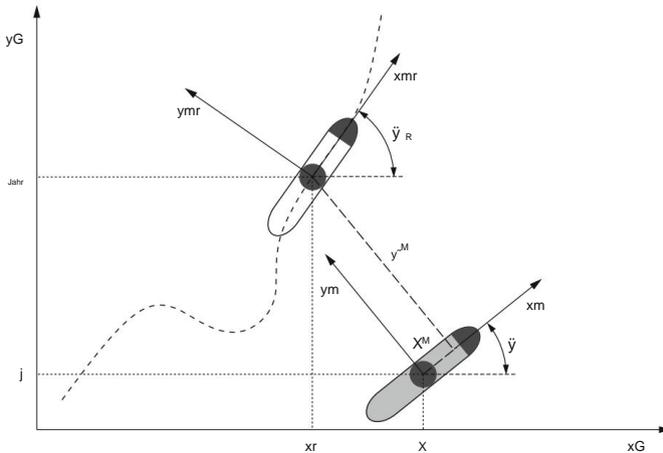


Abb. 13.14 Einrad-Wegverfolgungssteuerung; Das graue Einrad stellt den tatsächlichen Roboter dar und das weiße Einrad stellt das virtuelle Fahrzeug auf dem Weg dar

Die gewünschte Roboterausrichtung kann dann berechnet werden als:

$$\dot{\gamma}_r = \arctan \frac{u_y}{u_x}, \quad (13.39)$$

was zu einer Winkelgeschwindigkeit führt

$$\ddot{\gamma} = \dot{\gamma} = \text{PID}(\ddot{\gamma}). \quad (13.40)$$

Die Vorwärtsgeschwindigkeit des Roboters kann wie folgt berechnet werden:

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} = v \sqrt{\cos^2 \dot{\gamma} + v^2 \sin^2 \dot{\gamma}} = u \sqrt{2 \frac{2 + u}{x_j}}. \quad (13.41)$$

Unter der Annahme eines Differentialantriebsroboters können die Radwinkelgeschwindigkeiten wiederum aus (13.8) berechnet werden.

13.2.3.4 Pfadverfolgung

Oftmals kann der Roboter nicht einfach den kürzesten Weg zum Ziel nehmen und muss einem vorgegebenen Weg folgen. In diesem Fall besteht das Kontrollziel darin, auf dem Pfad zu bleiben. Die Aufgabe lässt sich vereinfachen, indem man ein virtuelles Fahrzeug betrachtet, das sich mit einer vordefinierten Geschwindigkeit entlang der Strecke bewegt. Dann besteht das Kontrollziel darin, das virtuelle Fahrzeug zu verfolgen, wie in Abb. 13.14 dargestellt.

Der Tracking Error kann definiert werden als:

$$\mathbf{x}^- = x_r \ddot{y} \mathbf{x}, \quad (13.42)$$

wobei x_r und \mathbf{x} die Position und Ausrichtung des virtuellen Fahrzeugs bzw. des mobilen Roboters darstellen. Alle Größen werden im globalen Koordinatensystem ausgedrückt und können als in das Roboterkoordinatensystem umgewandelt werden

$$\mathbf{X}^M = \begin{matrix} x^-_m \\ y^-_m \end{matrix} = \mathbf{R}^T \mathbf{X}, \quad (13.43)$$

$$\ddot{y} \ddot{y}^m \ddot{y}$$

wobei \mathbf{R} wie in (13.2) definiert ist. Die Vorwärtsgeschwindigkeit des Roboters kann aus dem Spurfehler entlang der x_m -Achse wie folgt berechnet werden

$$v = K_x x^-_m, \quad (13.44)$$

Dabei ist K_x die Proportionalverstärkung des Reglers. Die Winkelgeschwindigkeit muss den Winkelverfolgungsfehler $\ddot{y}^-_m = \ddot{y}$ sowie den Wegabstand y^-_m berücksichtigen. Wenn sich der Roboter nämlich vom Pfad entfernt, muss er auf den Pfad zusteuern. Somit wird der Steueralgorithmus

$$\ddot{y} = K_y y^-_m + K_{\dot{y}} \ddot{y}^-_m, \quad (13.45)$$

wobei K_y und $K_{\dot{y}}$ reglerproportionale Verstärkungen sind. Da die Geschwindigkeit des virtuellen Fahrzeugs bekannt ist (die Winkelgeschwindigkeit kann als Änderung der tangentialen Richtung entlang des Pfades berechnet werden, wenn sich das virtuelle Fahrzeug vorwärts bewegt), kann sie als Vorwärtsregelungsterm berücksichtigt werden. Wenn v_r die Vorwärtsgeschwindigkeit des virtuellen Fahrzeugs und \dot{y}_r seine Winkelgeschwindigkeit ist, gelten die Gl. (13.44) und (13.45) können mit dem Feedforward-Term als umgeschrieben werden

$$v = v_r \cos \ddot{y}^- + K_x x^-_m \quad (13.46)$$

Und

$$\ddot{y} = \dot{y}_r + K_y y^-_m + K_{\dot{y}} \ddot{y}^-_m. \quad (13.47)$$

Kapitel 14

Humanoide Robotik



Schon bevor sich die moderne Robotik zu entwickeln begann, interessierten sich Philosophen, Ingenieure und Künstler für menschenähnliche Maschinen. Das erste bekannte Beispiel eines humanoiden Mechanismus, dessen Konstruktion erhalten geblieben ist und noch heute nachgebaut werden kann, ist ein mechanischer Ritter, der von Leonardo da Vinci geschaffen und um 1495 dem Mailänder Herrscher Ludovico Sforza geschenkt wurde. Der Mechanismus hatte eine ähnliche kinematische Struktur. Heutige humanoide Roboter könnten sich über ein System aus Drähten und Flaschenzügen fortbewegen.

In jüngerer Zeit dachten Schriftsteller wie Karel Capek und Isaac Asimov an Roboter, die eine ähnliche Form wie Menschen haben. Es gibt mehrere Gründe, warum humanoide Roboter als interessant gelten:

- Menschliche Umgebungen sind für Menschen gebaut, daher sollte ein Allzweckroboter, der für menschliche Umgebungen, z. B. Häuser, Fabriken, Krankenhäuser, Schulen usw., entwickelt wurde, eine ähnliche Form wie Menschen haben, um in solchen Umgebungen erfolgreich agieren zu können. • Für Menschen ist es natürlicher, mit Robotern zu interagieren und zu kommunizieren, die aussehen und sich wie Menschen verhalten.
- Ein humanoider Roboter kann als experimentelles Werkzeug dienen, um die Theorien über menschliches Verhalten zu testen, die von computergestützten Neurowissenschaftlern entwickelt wurden, die sich für die Funktionsweise des menschlichen Gehirns interessieren.

Man kann sagen, dass die moderne humanoide Robotik mit einer Reihe humanoider Roboter begann, die an der Universität Waseda in Tokio, Japan, entwickelt wurden. Der erste dieser Roboter war WABOT-1, der 1973 entwickelt wurde.

Trotz jüngster Fortschritte in verwandten Bereichen wie der Soft-Robotik und der künstlichen Intelligenz sind humanoide Roboter, die in von Menschen besiedelten Umgebungen operieren können, wo sie auf natürliche Weise mit Menschen zusammenarbeiten und kommunizieren können, immer noch nur ein ferner Traum. Derzeit befinden sich humanoide Roboter in einem Stadium, in dem sie eine Vielzahl von Aufgaben ausführen können. Zu den Aufgaben, die beispielsweise bei humanoiden Roboterwettbewerben, z. B. der DARPA Robotics Challenge, eingesetzt werden, gehören:

1. Fahren: Fahren Sie mit einem Nutzfahrzeug eine durch Schranken blockierte Fahrspur entlang.
2. Ausstieg: Verlassen Sie das Fahrzeug und bewegen Sie sich in einen bestimmten Bereich.
3. Tür: Öffnen Sie eine Tür und gehen Sie durch einen Durchgang.
4. Ventil: Drehen Sie ein Ventil, das mit einem Handrad betätigt wird.

5. Wand: Schneiden Sie mit einem Werkzeug (Bohrer oder Säge) eine Betonplatte durch.
6. Überraschungsaufgabe, die bis zum Wettkampftag noch nicht bekannt war: a entfernen
Ziehen Sie den magnetischen Stecker aus einer Steckdose und stecken Sie ihn in eine andere Steckdose.
7. Schutt: Überqueren Sie ein Trümmerfeld oder befahren Sie unebenes Gelände.
8. Treppen: Steigen Sie die Treppe hinauf.

Moderne humanoide Roboter können solche Aufgaben bereits autonom ausführen, sofern der ungefähre Zustand der Umgebung im Voraus bekannt ist. Allerdings ist es für moderne humanoide Roboter immer noch schwierig, solche Aufgaben ohne vorherige Informationen über die Umgebungsbedingungen auszuführen, die ein Programmierer nutzen kann, um den humanoiden Roboter auf die Ausführung mehrerer Aufgaben vorzubereiten. Die Integration und kontinuierliche Abfolge mehrerer Roboteraktionen bleibt ein Problem, und bei der Ausführung längerer Aufgabenabläufe ist immer noch ein gewisses Maß an Teleoperation erforderlich.

Während die meisten Standardmethoden der Robotik in Bezug auf Roboterkinematik, Dynamik, Steuerung, Trajektorienplanung und Sensorik auch bei der Entwicklung humanoider Roboter relevant sind, muss sich die humanoide Robotik mit mehreren spezifischen Problemen befassen.

Im Vordergrund steht das Problem der Fortbewegung und des Gleichgewichts des Zweibeiners. Im Gegensatz zu anderen Robotern müssen humanoide Roboter während ihrer Arbeit gehen und das Gleichgewicht halten. Bei der oben genannten Robotik-Herausforderung erwies sich die Fortbewegung als eines der größten. Der grundlegende Indikator, der das Gleichgewicht eines humanoiden Roboters beschreibt, ist das Konzept des Nullmomentpunktes, üblicherweise als ZMP abgekürzt. Das Konzept des ZMP wurde 1968 von Miomir Vukobratović eingeführt. Es ist bis heute der am weitesten verbreitete Ansatz zur Erzeugung dynamisch stabiler Gehbewegungen, bei denen der oder die Standfüße jederzeit Kontakt zur Bodenoberfläche haben. Dies ist wichtig, um zu verhindern, dass der Roboter herunterfällt. Die grundlegenden Konzepte im Zusammenhang mit ZMP werden in Abschn. beschrieben. 14.1.

Ein weiteres spezifisches Problem bei der Programmierung humanoider Roboter ist die im Vergleich zu Standard-Industrierobotern sehr hohe Anzahl an Freiheitsgraden. Während typische Industrieroboter nur 6 und selten 7 Freiheitsgrade haben, verfügen humanoide Roboter oft über mehr als 30 Freiheitsgrade. Beispielsweise verfügt Honda Asimo, einer der bekanntesten humanoiden Roboter, über 34 Freiheitsgrade: 3 im Kopf, 7 in jedem Arm (3 in der Schulter, 1 im Ellenbogen und 3 im Handgelenk), 1 in der Taille, 6 in jedem Bein und 2 in jeder Hand. Eine so große Anzahl an Freiheitsgraden macht die klassische Roboterprogrammierung mit Programmiergeräten und textuellen Programmiersprachen unpraktikabel. Stattdessen können wir die Ähnlichkeit zwischen humanoiden Robotern und Menschen ausnutzen. Aufgrund dieser Ähnlichkeit können humanoide Roboter Aufgaben auf ähnliche Weise ausführen wie Menschen. Diese Tatsache lässt die Idee aufkommen, dass ein menschlicher Lehrer, anstatt einen humanoiden Roboter zu programmieren, dem Roboter zeigen kann, wie er die gewünschte Aufgabe ausführen soll. Der Roboter kann dann versuchen, die menschliche Ausführung nachzubilden. Diese Art der Roboterprogrammierung wird als Programmierung durch Demonstration oder Nachahmungslernen bezeichnet. Für eine erfolgreiche Anwendung ist es erforderlich, dass ein Roboter die demonstrierte Bewegung auf seine eigene kinematische und dynamische Struktur überträgt. Da natürliche Umgebungen darüber hinaus selten statisch sind, sondern sich häufig verändern, kann der Roboter die beobachteten Bewegungen nicht einfach nachbilden. Stattdessen sollten die beobachteten Bewegungen an die aktuellen Umweltbedingungen angepasst werden. Diese Themen werden in Abschn. 14.2.

14.1 Zweibeinige Fortbewegung

Die Fortbewegung von Zweibeinern ist ein wichtiges Thema in der humanoiden Robotik. Hier konzentrieren wir uns auf das Gehen, das sich von anderen Formen der zweibeinigen Fortbewegung wie dem Laufen dadurch unterscheidet, dass immer mindestens ein Fuß Bodenkontakt hat. Wie in der Einleitung erläutert, nutzen die meisten modernen humanoiden Roboter das Nullmomentpunktprinzip, um stabile Laufmuster zu erzeugen.

14.1.1 Nullmomentpunkt

In diesem Abschnitt gehen wir davon aus, dass der Boden flach und orthogonal zur Schwerkraft ist. Wir beginnen mit der Analyse der Verteilung einer vertikalen Komponente der Bodenreaktionskräfte (dh der Komponente orthogonal zum Boden, wie in Abb. 14.1 dargestellt). Der Nullmomentpunkt ist der Punkt, an dem die Resultierende dieser Kräfte den Boden schneidet. Wir konzentrieren uns zunächst auf die Bewegung in der Sagittalebene (also der Ebene, die den Körper in den linken und rechten Teil teilt). Wie in Abb. 14.1 dargestellt, muss eine zum Boden orthogonale Komponente der Bodenreaktionskraft an allen Kontaktpunkten positiv sein, sonst würde der Fuß den Kontakt zum Boden verlieren, da er nicht fest mit dem Boden verbunden ist. Der Nullmomentpunkt p_x gemäß obiger Definition kann wie folgt berechnet werden

$$p_x = \frac{\int_{x_b}^x x f_z(x) dx}{f_n}, \quad (14.1)$$

$$f_n = \int_{x_b}^x f_z(x) dx, \quad (14.2)$$

Dabei ist $f_z(x)$ die vertikale Komponente der Bodenreaktionskraft am Kontaktpunkt x und f_n die vertikale Netto-Bodenreaktionskraft. Der Grund, warum p_x als Momentennullpunkt bezeichnet wird, wird klar, wenn das Moment bei p_x berechnet wird:

$$\begin{aligned} \ddot{y}(p_x) &= \ddot{y} \frac{\int_{x_b}^x (x - \ddot{y} p_x) f_z(x) dx}{f_n} = \ddot{y} \frac{\int_{x_b}^x x f_z(x) dx}{f_n} - \ddot{y} p_x \frac{\int_{x_b}^x f_z(x) dx}{f_n} \\ &= \ddot{y}(p_x - \ddot{y} p_x) = 0. \end{aligned} \quad (14.3)$$

Hier haben wir das Moment $\ddot{y} = \ddot{y}(x - \ddot{y} p_x) f_z$ über die gesamte Sohlenfläche integriert, d. h. $\ddot{y} \int x f_z$. Somit ist das Nettomoment am Nullmomentpunkt p_x gleich Null. Das $x_b - \ddot{y} p_x$ Nullmomentpunkt wird üblicherweise als ZMP abgekürzt. Es ist der Punkt auf der Bodenoberfläche, an dem der Nettodrehimpuls gleich Null ist. Wenn es vorhanden ist, muss ZMP innerhalb des Stützpolygons liegen.

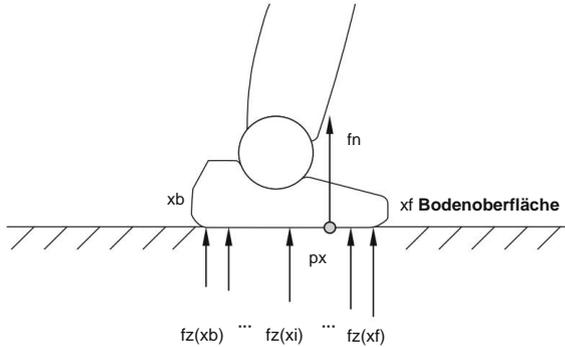


Abb. 14.1 Bodenreaktionskräfte $fz(xi)$ an verschiedenen Kontaktpunkten xi und die Nullmomentpunkt px und die Netto-Bodenreaktionskräfte orthogonal zur Auflagefläche fn werden nach Gl. (14.1) bzw. (14.2).

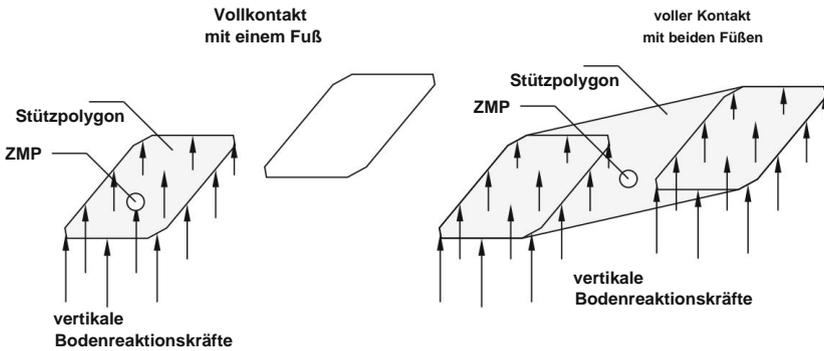


Abb. 14.2 Das Stützpolygon (die von einer grauen Linie umschlossene Fläche) ist definiert als die konvexe Hülle aller Punkte, die mit dem Boden in Kontakt stehen. Links: Das Stützpolygon entspricht der Sohlenfläche, wenn nur ein Fuß vollen Bodenkontakt hat. Rechts: Das Stützpolygon entspricht der konvexen Hülle der Ecken beider Füße, wenn beide Füße vollständigen Bodenkontakt haben

Für allgemeine humanoide Roboter, die in 3D gehen, sollte auch die seitliche Bewegung berücksichtigt werden. Wie in Abb. 14.2 dargestellt, müssen zwei Fälle unterschieden werden: Entweder hat nur ein Fuß vollständigen Bodenkontakt oder beide Füße haben vollständigen Bodenkontakt. Der Boden wird in der Höhe p_z als eben angenommen. Die Ableitung von ZMP basiert auf der Beziehung zwischen dem Moment um den Punkt $p = (p_x, p_y, p_z)$ der vertikalen Bodenreaktionskraft $[0, 0, f_z(\tilde{y})]$ an allen Punkten $\tilde{y} = (\tilde{y}_x, \tilde{y}_y, p_z)$ auf der Kontaktfläche. Der Moment ist gegeben durch

$$\tilde{y}(\mathbf{p}) = (\tilde{y}_y \tilde{y}_x - \tilde{y}_x \tilde{y}_y) \times \begin{pmatrix} 0 \\ 0 \\ f_z(\tilde{y}) \end{pmatrix} = \begin{pmatrix} (\tilde{y}_y \tilde{y}_x - \tilde{y}_x \tilde{y}_y) f_z(\tilde{y}) \\ \tilde{y}_y \tilde{y}_x - \tilde{y}_x \tilde{y}_y \\ 0 \end{pmatrix} \quad (14.4)$$

Um das Moment um den Punkt $\mathbf{p} = (p_x, p_y, p_z)$ aufgrund der orthogonalen Bodenreaktionskräfte $[0, 0, f_z(\tilde{\mathbf{y}})]$ zu erhalten, die an allen Kontaktpunkten $\tilde{\mathbf{y}}$ zwischen Sohle und Boden auftreten, müssen wir integrieren über alle Kontaktpunkte hinweg

$$\tilde{\mathbf{y}} \mathbf{n}(\mathbf{p}) = \begin{pmatrix} \tilde{y}_x \tilde{y}_x p_x \\ \tilde{y}_y \tilde{y}_y p_y \\ \tilde{y}_z \tilde{y}_z p_z \end{pmatrix} \times \int_S \mathbf{dS} = \begin{pmatrix} \int_S (\tilde{y}_y \tilde{y}_y p_y) f_z(\tilde{\mathbf{y}}) dS \\ \int_S (\tilde{y}_x \tilde{y}_x p_x) f_z(\tilde{\mathbf{y}}) dS \\ 0 \end{pmatrix} \quad (14.5)$$

wobei S die Kontaktfläche bezeichnet. Ähnlich wie im 2D-Fall ist der Punkt auf dem Boden, an dem das Moment der Normalen der Bodenreaktionskraft Null wird (dh der Nullmomentpunkt $\tilde{\mathbf{y}} \mathbf{n}(\mathbf{p}) = \mathbf{0}$), gegeben durch

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \frac{\int_S \tilde{y}_x f_z(\tilde{\mathbf{y}}) dS}{f_n}, \frac{\int_S \tilde{y}_y f_z(\tilde{\mathbf{y}}) dS}{f_n}, \mathbf{p}_z \quad (14.6)$$

Wo

$$f_n = \int_S f_z(\tilde{\mathbf{y}}) dS \quad (14.7)$$

ist die Summe der Bodenreaktionskräfte orthogonal zum Boden bei allen Kontakten zwischen Sohle und Boden.

Bei einem echten humanoiden Roboter liegt ZMP (falls vorhanden) garantiert innerhalb des Stützpolygons, denn wenn der Kontakt zwischen der Sohle und der Bodenoberfläche besteht, muss die Komponente der Bodenreaktionskraft orthogonal zum Boden positiv sein. Andernfalls würde der Kontakt zwischen Sohle und Bodenoberfläche verloren gehen, da der Roboter nicht am Boden befestigt ist und daher keine negativen vertikalen Bodenreaktionskräfte erzeugen kann. Der humanoide Roboter kann seine Haltung nur dann mit seinen Füßen steuern, wenn das ZMP innerhalb des Stützpolygons vorhanden ist. Andernfalls verliert der Roboter den Kontakt zum Boden und kann die Haltung mit seinen Füßen nicht mehr

14.1.2 Generierung von Laufmustern

Beim zweibeinigen Gehen wechseln die Füße des Roboters zwischen zwei Phasen:

- Standphase, in der sich die Position des Fußes nicht verändern soll,
- Schwungphase, in der sich der Fuß bewegt.

Abbildung 14.3 zeigt diese beiden unterschiedlichen Phasen im Gangzyklus: Wenn beide Füße Bodenkontakt haben, befindet sich der Roboter in der Phase der doppelten Unterstützung. Die Füße bewegen sich in dieser Phase nicht. Sobald sich einer der Füße zu bewegen beginnt, wechselt der Roboter von der Doppel- in die Einzelunterstützungsphase, in der sich einer der beiden Füße bewegt. Auf die einfache Stützphase folgt eine weitere doppelte Stützphase, sobald der Fuß in der Schwungphase Bodenkontakt herstellt.

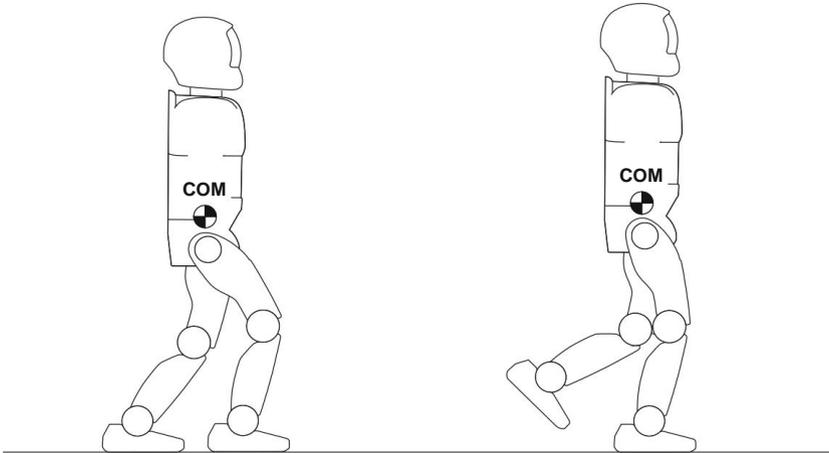


Abb. 14.3 Einfache und doppelte Stützphase. In der Doppelstützphase haben beide Füße Kontakt zum Boden und das Gewicht des Roboters wird von beiden Beinen getragen. In der Einzelstützphase ist ein Fuß in Bewegung, während der andere Fuß, der den Roboter stützt, Bodenkontakt hat.

Beim ZMP-basierten Gehen haben ein oder beide Füße des Roboters immer Kontakt mit dem Boden. Somit existiert ZMP und der Roboter kann das Gleichgewicht halten, indem er sicherstellt, dass das Stützpolygon das ZMP enthält. Allerdings kann der Roboter den ZMP nicht direkt steuern, wie in Gleichungen definiert. (14.1) und (14.6). Wir führen daher das Konzept des Schwerpunkts (COM) ein. ZMP kann durch Ausnutzung seiner Beziehung zu COM kontrolliert werden.

Der Schwerpunkt (COM) ist definiert als die durchschnittliche Position aller Körperteile eines humanoiden Roboters, gewichtet mit der Masse der Körperteile. Für einen Roboter mit starren D-Verbindungen kann COM wie folgt berechnet werden:

$$\mathbf{c} = \frac{\sum_{i=1}^D m_i \mathbf{c}_i}{M}, \quad M = \sum_{i=1}^D m_i, \quad (14.8)$$

Dabei ist m_i die Masse des i -ten Glieds und \mathbf{c}_i seine Position, die durch direkte Kinematik berechnet werden kann, sofern der Massenschwerpunkt jedes Glieds in den lokalen Koordinaten des Glieds bekannt ist. Mit einigen Näherungen kann die Beziehung zwischen ZMP und COM wie folgt angegeben werden

$$p_x = c_x \ddot{y} - \frac{(c_z \ddot{y} - p_z) c''_x}{c''_z + g}, \quad (14.9)$$

$$p_y = c_y \ddot{y} - \frac{g (c_z \ddot{y})}{c''_z + g}, \quad (14.10)$$

Dabei bezeichnet p_z die Höhe des Erdgeschosses, g die Schwerkraftkonstante und $c = (c_x, c_y, c_z)$ und $p = (p_x, p_y, p_z)$ die Koordinaten von COM bzw. ZMP.

Beachten Sie, dass, wenn der Roboter ruht, also $\dot{c} = \dot{c}_x = \dot{c}_y = 0$, ZMP und die Projektion von COM als $p_x = c_x$ und $p_y = c_y$ übereinstimmen. Beachten Sie auch, dass, wenn der Boden flach und orthogonal zur Schwerkraft ist, wie wir in Abschn. 14.1.1 ist p_z eine Konstante.

Generell unterscheidet man zwischen statischem und dynamischem Gehen. Statisches Gehen ist definiert als jede stabile Gehbewegung, bei der die COM-Projektion immer innerhalb des Stützpolygons bleibt. Dies bedeutet, dass, wenn der Roboter zu irgendeinem Zeitpunkt während des Gehens vollständig aufhört, sich zu bewegen, er nicht herunterfällt, da für den ruhenden Roboter die Projektion von COM auf die Bodenoberfläche gleich dem ZMP ist (siehe Gleichungen (14.9) und (14.10)). Beim statischen Gehen muss die Bewegung im Allgemeinen langsam sein, damit die Projektion von COM nahe am ZMP liegt. Diese Art des Gehens erfordert normalerweise große Füße und starke Knöchelgelenke, um ausreichende Kräfte an den Knöcheln zu erzeugen. Wenn die Bewegung des Roboters schneller wird, werden ZMP und die Projektion von COM unterschiedlicher und die Stabilität kann nicht durch die Steuerung der Projektion von COM allein gewährleistet werden.

Effektiveres Gehverhalten wird durch dynamische Gehmuster erzeugt, bei denen die Projektion von COM nicht gleich ZMP ist und während einer bestimmten Bewegungsperiode außerhalb des Stützpolygons liegen kann. Ein ZMP-basiertes dynamisches Gehmuster ist in Abb. 14.4 dargestellt. Solche Muster sind so geplant, dass der ZMP in allen Phasen des Gehens innerhalb der Grenzen des Stützpolygons bleibt. Dies kann wie folgt bewerkstelligt werden:

- Geben Sie die kartesische Bewegung der Roboterfüße an. Dabei wird die Schrittlänge des Roboters und der Zeitpunkt der Fußbewegung vorgegeben.

Geben Sie die Referenz-ZMP-Trajektorie an, damit ZMP innerhalb der Stütze bleibt Polygon jederzeit.

- Bestimmen Sie die Oberkörperbewegung des humanoiden Roboters, um die Referenz-ZMP-Bewegung zu realisieren. Dies kann mithilfe der Gleichungen erreicht werden. (14.9) und (14.10).
- Die Beinbewegung des humanoiden Roboters wird schließlich aus dem Körper und den Füßen berechnet Bewegung mittels inverser Kinematik.

Die Bewegung von COM wird durch die Gleichungen nicht vollständig spezifiziert. (14.9) und (14.10), da es nur zwei Gleichungen und drei unbekannte Parameter gibt. Um die Bewegung von COM und folglich die Bewegung des Oberkörpers des humanoiden Roboters vollständig zu spezifizieren, muss eine zusätzliche Einschränkung auferlegt werden. Es gibt mehrere mögliche Ansätze. Die einfachste davon besteht darin, die Höhe von COM auf einen konstanten Wert zu setzen ($\dot{c}_z = \text{const}$, $c_z = 0$).

Unter dieser Annahme wird die Bewegung von COM vollständig durch die Gleichungen spezifiziert. (14.9) und (14.10). Eine anpassungsfähigere und aktivere Bewegung kann erreicht werden, wenn c_z variiert werden kann.

Beachten Sie, dass der obige Ansatz die Bewegung von COM ohne Berücksichtigung der Beine bestimmt. Da jedoch der Großteil der Masse üblicherweise im Oberkörper eines humanoiden Roboters konzentriert ist und es nicht notwendig ist, der vorgeschriebenen ZMP-Trajektorie genau zu folgen, reicht der obige Ansatz aus, um dynamisch stabile Laufmuster zu erzeugen.

Wenn ein genaues Modell des Roboters verfügbar ist, kann das zweibeinige Gehen durch einfaches Befolgen eines vorgegebenen Laufmusters realisiert werden. Aufgrund von Lärm und Modellungenauigkeiten führt ein solcher Ansatz in der Praxis meist nicht zu einem stabilen Gehverhalten

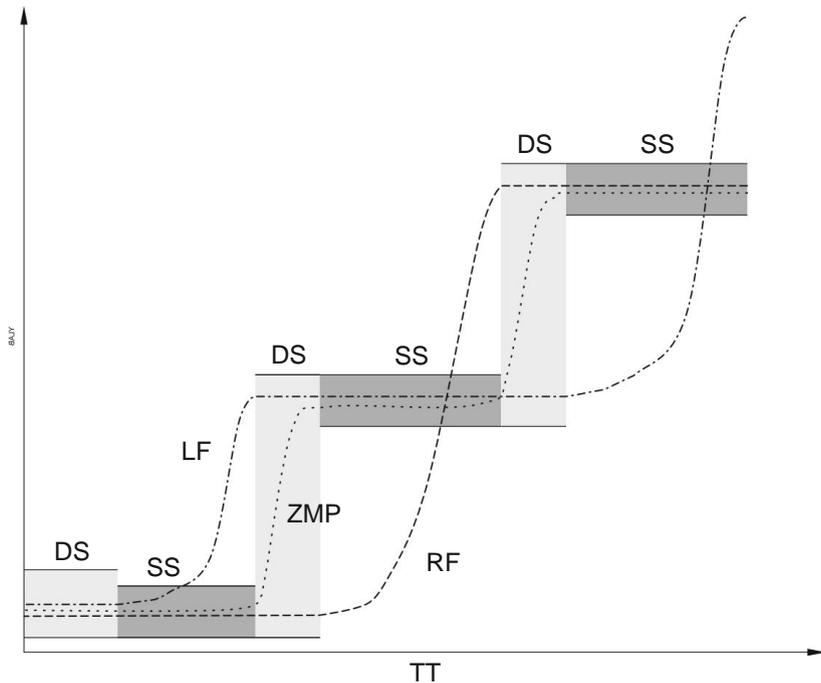


Abb. 14.4 Ein Beispiel für ein ZMP-basiertes Laufmuster in der Sagittalebene. Der Roboter stellt zunächst beide Füße ungefähr parallel auf den Boden und erzeugt dann drei Schritte, beginnend mit dem linken Fuß. Die schattierten Bereiche zeigen das Ausmaß des Stützpolygons während der Einzelstützphase (dunkel schattierter Bereich) und der Doppelstützphase (hell schattierter Bereich). Die ZMP-Trajektorie (gestrichelt gepunktet) ist so geplant, dass sie während der gesamten Gehdauer innerhalb des Stützpolygons bleibt. Außerdem sind die Flugbahnen beider Füße (links: gestrichelt gepunktet, rechts: gestrichelt)

ohne das vorberechnete Laufmuster durch einen Stabilisator zu ergänzen, der das Muster entsprechend dem sensorischen Input von Kreiseln, Beschleunigungsmessern, Kraftsensoren, Kameras usw. modifiziert.

Es sollte darauf hingewiesen werden, dass ZMP nicht das einzige Prinzip ist, mit dem sich stabile Laufmuster erzeugen lassen. Es ist möglich, ein Laufmuster zu erzeugen, bei dem ein Roboter für einen bestimmten Bewegungszeitraum instabil ist. Solche Laufmuster müssen so geplant werden, dass sich der Roboter von Instabilitäten erholen kann, bevor er zu Boden fällt.

14.2 Nachahmungslernen

Um ihr Potenzial voll auszuschöpfen, sollten humanoide Roboter in der Lage sein, eine Vielzahl von Aufgaben in unstrukturierten Umgebungen auszuführen (z. B. zu Hause, in Krankenhäusern, Geschäften, Büros und sogar im Freien). Die oben erwähnte Robotik-Herausforderung war darauf ausgerichtet

hin zu humanoiden Robotern an Katastrophenstandorten. Im Gegensatz zu vielen industriellen Umgebungen, in denen Roboter heutzutage weit verbreitet sind, können solche Umgebungen nicht im Voraus vorbereitet werden, um die Bedienung eines humanoiden Roboters zu erleichtern. Die Programmierung humanoider Roboter wird durch die große Anzahl an Freiheitsgraden, die mit der Bewegung humanoider Roboter verbunden sind, noch komplizierter. Daher reichen klassische Roboterprogrammieretechniken, die auf Programmierhandgeräten, sorgfältig vorbereiteten Offline-Simulationssystemen und Programmiersprachen basieren, für Humanoide nicht aus. Stattdessen ist es notwendig, humanoide Roboter mit Lern- und Anpassungsfähigkeiten auszustatten. Dadurch lassen sie sich leichter programmieren und können sich sogar selbstständig zusätzliches V

Das Erlernen des Verhaltens humanoider Roboter ist ein schwieriges Problem, da der Raum aller zu erforschenden Bewegungen humanoider Roboter sehr groß ist und exponentiell mit der Anzahl der Freiheitsgrade zunimmt. Eine Lösung für dieses Problem besteht darin, das Lernen auf die Teile des Roboterbewegungsraums zu konzentrieren, die für die gewünschte Aufgabe tatsächlich relevant sind. Dies kann durch Nachahmungslernen erreicht werden, das auch als „Programmieren durch Demonstration“ bezeichnet wird. Bei diesem Ansatz demonstriert ein menschlicher Lehrer einem Roboter, wie er die gewünschte Aufgabe ausführen soll. Damit es funktioniert, muss ein Roboter in der Lage sein, die wichtigen Informationen aus menschlichen Demonstrationen zu extrahieren und die wesentlichen Teile der Aufgabenausführung nachzubilden. Während es in den meisten Fällen nicht notwendig ist, die demonstrierten Bewegungen genau nachzubilden, um die gewünschte Aufgabe erfolgreich auszuführen, ist es von Vorteil, wenn der Roboter die demonstrierten Bewegungen so weit wie möglich nachahmen kann. Da der Körper eines humanoiden Roboters einem menschlichen Körper ähnelt, ist Nachahmungslernen oft ein guter Ansatz, um das Lernen auf die relevanten Teile des Bewegungsraums eines humanoiden Roboters zu konze

14.2.1 Beobachtung menschlicher Bewegung und ihrer Übertragung zur humanoiden Roboterbewegung

Es gibt viele mögliche Messsysteme und Technologien, mit denen menschliche Bewegungen beobachtet und gemessen werden können. Sie beinhalten

- optische Bewegungserfassungssysteme,
- Ensembles von Inertialmesseinheiten (IMU), • Computer-Vision-Methoden zur Schätzung menschlicher Bewegung, • passive Exoskelette, • Handführung.

Im Folgenden erläutern wir die wesentlichen Vor- und Nachteile dieser Systeme.

14.2.1.1 Optische Trackinggeräte zur Erfassung menschlicher Bewegungen

Optische Tracker basieren auf einer Reihe von Markierungen, die am menschlichen Körper angebracht werden. Marker können entweder passiv oder aktiv sein. Passive Markierungen bestehen aus retroreflektierenden Materialien, die das Licht in die Richtung reflektieren, aus der es kommt. In Syst

Markierungen, Kameras sind mit einem Band aus Infrarot-Leuchtdioden (LEDs) ausgestattet. Das emittierte Licht wird von der Markierung zurück in Richtung der Kamera reflektiert, wodurch die Markierung viel heller ist als jeder andere Punkt im Bild. Durch diese Eigenschaft sind retroreflektierende Markierungen in Kamerabildern leicht zu erkennen. Mittels Triangulation kann ein 3D-Markerstandort berechnet werden, wenn der Marker in mindestens zwei gleichzeitig erfassten Kamerabildern erkannt wird. Die vorhergesagte Bewegung sichtbarer Marker wird verwendet, um die sichtbaren Marker abzugleichen, die zu zwei aufeinanderfolgenden Messzeitpunkten extrahiert wurden.

Im Gegensatz zu passiven Markern, die Licht reflektieren, sind aktive Marker mit LEDs ausgestattet und strahlen so ihr eigenes Licht ab. Daher müssen sie mit Strom versorgt werden. Optische Tracker mit aktiven Markern beleuchten in der Regel für eine sehr kurze Zeit jeweils nur einen Marker. Dadurch weiß das System immer, welcher Marker gerade sichtbar ist, und stellt so die Identität des Markers bereit. Aus diesem Grund können optische Trackingsysteme mit aktiven Markern temporäre Verdeckungen besser bewältigen als Systeme mit passiven Markern, da ein verdeckter aktiver Marker identifiziert werden kann, sobald er wieder sichtbar wird. Dies ist bei passiven Markern nicht der Fall. Da aktive Marker andererseits Strom benötigen, müssen sie über Kabel an eine Stromquelle angeschlossen werden. Dies macht ihre Verwendung umständlicher als passive Markierer, die keine Kabel erfordern.

Um menschliche Bewegungen zu messen, müssen sowohl passive als auch aktive Marker an geeigneten Stellen an den menschlichen Körpersegmenten angebracht werden. In der Regel sind an jedem Körpersegment mindestens drei Markierungen angebracht, da sonst die Lage starrer Körpersegmente nicht abgeschätzt werden kann. Um das Anbringen von Markern an den relevanten Körpersegmenten zu erleichtern, wurden in der Vergangenheit verschiedene spezielle Motion-Capture-Anzüge entwickelt.

Optische Trackingsysteme mit aktiven oder passiven Markern liefern 3D-Positionen von am menschlichen Körper angebrachten Markern, die gerade sichtbar sind. Die dreidimensionale Position und Ausrichtung eines Körpersegments kann geschätzt werden, wenn mindestens drei am Segment angebrachte Markierungen sichtbar sind. Um die beobachtete Bewegung mit einem Roboter zu reproduzieren, müssen diese Informationen mit der Roboterbewegung in Beziehung gesetzt werden. Mit einem gewissen Grad an Genauigkeit kann die menschliche Bewegung als Gelenkbewegung starrer Körperteile modelliert werden. Wenn die Kinematik eines humanoiden Roboters der Kinematik des menschlichen Körpers nahe genug ist, können wir ihn wie in [Abb. 14.5 in einen menschlichen Körper einbetten](#). Eine solche Einbettung kann später verwendet werden, um die Gelenkwinkel aus den Orientierungen zu berechnen. Nehmen wir an, dass die Orientierung zweier aufeinanderfolgender Körpersegmente durch die Orientierungsmatrizen R_1 und R_2 gegeben ist und dass das Gelenk, das die beiden Segmente verbindet, aus drei aufeinanderfolgenden Gelenkachsen j_1 , j_2 und j_3 mit Drehwinkeln besteht, die mit \dot{y}_1 , \dot{y}_2 und \dot{y}_3 bezeichnet werden. Wir gehen weiterhin davon aus, dass zwei aufeinanderfolgende Gelenkachsen orthogonal sind und dass sich alle drei Achsen in einem gemeinsamen Punkt schneiden. In einer solchen Anordnung entsprechen die drei Gelenkwinkel den in Kap. eingeführten Euler-Winkeln. 4. Insgesamt gibt es 12 verschiedene Gelenkachsenkombinationen, die alle möglichen Achsenanordnungen in Gelenken mit drei Freiheitsgraden abdecken. In [Abb. 14.5](#) können Rumpf-, Nacken-, Schulter-, Handgelenk- und Sprunggelenke durch eine geeignete Kombination von Euler-Winkeln beschrieben werden. Die Beziehung zwischen diesen Werten ist gegeben durch

$$R_1 = R(j_1, \dot{y}_1)R(j_2, \dot{y}_2)R(j_3, \dot{y}_3)R_2 = R(\dot{y}_1, \dot{y}_2, \dot{y}_3)R_2. \quad (14.11)$$

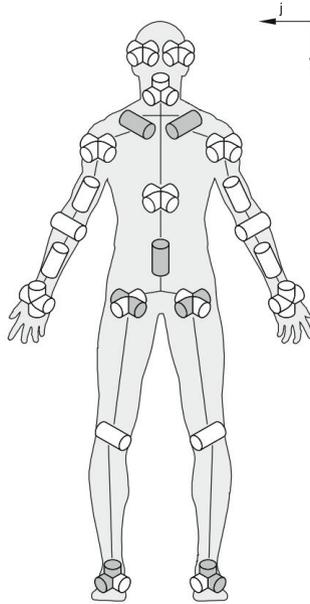


Abb. 14.5 Kinematische Struktur eines humanoiden Roboters. In der aufrechten Position mit ausgestreckten Armen und Beinen liegen alle Gelenkachsen parallel zu einer der drei Hauptachsen des Körpers (vorwärts/rückwärts: x-Achse, links/rechts: y-Achse, oben/unten: z-Achse)

Die Gelenkwinkel $\dot{\gamma}$, $\ddot{\gamma}$ und $\ddot{\gamma}$ können dann durch Lösen der Gleichung berechnet werden

$$R(\dot{\gamma}, \ddot{\gamma}, \ddot{\gamma}) = \mathbf{R} \mathbf{1} \mathbf{R} \quad \mathbf{I} \quad \mathbf{2} . \quad (14.12)$$

Diese Gleichung hängt von der Wahl der Gelenkachsen j_1 , j_2 und j_3 ab. Die beobachtete Bewegung kann von einem Roboter nachgebildet werden, sobald alle relevanten Gelenkwinkel aus dem eingebetteten Modell geschätzt wurden.

Optische Trackingsysteme können auch die absolute Position und Ausrichtung des menschlichen Körpers in einem Weltkoordinatensystem genau schätzen. Da die Wurzel der Kinematik eines humanoiden Roboters normalerweise im lokalen Koordinatenrahmen liegt, der am Rumpf befestigt ist, entspricht die geschätzte Position und Ausrichtung des Rumpfes der absoluten Position und Ausrichtung des menschlichen Körpers in Weltkoordinaten.

14.2.1.2 Inertiale Messeinheiten (IMUs)

Inertiale Messeinheiten (IMUs) enthalten verschiedene Sensoren, darunter Beschleunigungsmesser zur Messung der 3D-Linearbeschleunigung und Gyroskope zur Messung der Änderungsrate der 3D-Ausrichtung (dh der Winkelgeschwindigkeit). IMUs umfassen häufig auch Magnetometer, um redundante Messungen bereitzustellen, die Genauigkeit zu verbessern und die zu reduzieren

Drift. Aus diesen Daten kann die Position und Ausrichtung einer IMU abgeschätzt werden, wie in Abschn. 7.2.6.

Im Zusammenhang mit der Übertragung menschlicher Bewegungen auf Bewegungen humanoider Roboter können IMU-Daten verwendet werden, um die Position und Ausrichtung jedes Körpersegments abzuschätzen, an dem eine IMU angebracht ist. Genau wie bei markerbasierten Trackern können die Gelenkwinkel aus den Ausrichtungen aufeinanderfolgender Körpersegmente mithilfe von Gl. geschätzt werden. (14.12)

Im Gegensatz zu optischen Trackingsystemen kommt es bei IMUs nicht zu Verdeckungen, da keine externen Kameras zur Messung der IMU-Bewegung erforderlich sind. Andererseits sind IMUs nicht so genau wie optische Trackingsysteme, da sie die Integration von Linearbeschleunigung und Winkelgeschwindigkeit erfordern. Die Integration kann auch zu Drift führen, insbesondere bei der Schätzung der absoluten Körperposition und -orientierung im Raum. Die Drift kann durch die Entwicklung geeigneter Filter reduziert werden, die die Redundanz der von Beschleunigungsmessern, Gyroskopen und Magnetometern erhaltenen Messungen ausnutzen.

14.2.1.3 Passive Exoskelette und Handführung

Ein entscheidendes Problem, mit dem sich alle oben genannten Systeme befassen müssen, besteht darin, dass sie menschliche Bewegungen messen, ohne die Unterschiede zwischen der Kinematik und Dynamik von Mensch und Roboter zu berücksichtigen. Solche Messungen müssen häufig an die Randbedingungen des Roboters angepasst werden, da der Roboter sonst die demonstrierten Bewegungen nicht ausführen kann. Alternativ kann ein nichtlineares Optimierungsproblem formuliert werden, um die demonstrierte Bewegung an die Fähigkeiten eines Zielroboters anzupassen.

Das Problem der Übertragung menschlicher Bewegungen auf Roboterbewegungen kann durch den Einsatz verschiedener Messsysteme vermieden werden. Eine Möglichkeit besteht darin, ein spezielles passives Gerät zu entwerfen, das wie ein Exoskelett getragen wird und dessen Freiheitsgrade den Freiheitsgraden des Roboters entsprechen. Das passive Exoskelett muss so konzipiert sein, dass es die Bewegung bei den meisten Bewegungen nicht einschränkt. Es verfügt über keine Motoren, sollte aber mit Goniometern zur Messung der Gelenkwinkel ausgestattet sein. Die vom Exoskelett gemessenen Gelenkwinkel können zur direkten Steuerung des Roboters genutzt werden, wenn die Kinematik des Zielroboters mit der Kinematik des Exoskeletts übereinstimmt. Ein Nachteil passiver Exoskelette besteht darin, dass sie wie Kleidung auf die spezifische Größe eines menschlichen Demonstranten zugeschnitten sein müssen.

Wie in Abschn. erläutert. 12.3.2 können einige Roboter physisch durch die gewünschten Bewegungen geführt werden (siehe auch Abb. 14.6). Beim Handführen wird die Bewegung durch robotereigene Gelenkwinkelsensoren erfasst und ist somit standardmäßig kinematisch durchführbar. Dieser Ansatz ist effektiv, wenn der Roboter nachgiebig ist und die Schwerkraft kompensieren kann, sodass ein menschlicher Demonstrator ihn problemlos in die gewünschte Richtung bewegen kann.

Der Hauptnachteil der Handführung besteht darin, dass die Demonstration der gewünschten Bewegung für einen menschlichen Demonstrator weniger natürlich ist als beispielsweise bei der Verwendung markerbasierter Trackingsysteme. Daher ist es mit solchen Systemen manchmal nicht so einfach, komplexe Bewegungen darzustellen. Beispielsweise ist Handführung nicht effektiv, um komplexe Tanzbewegungen zu demonstrieren. Andererseits kann das Tanzen leicht von einem Menschen direkt vorgeführt und mit einem optischen Tracker, IMUs oder einem passiven Exoskelett gemessen werden.

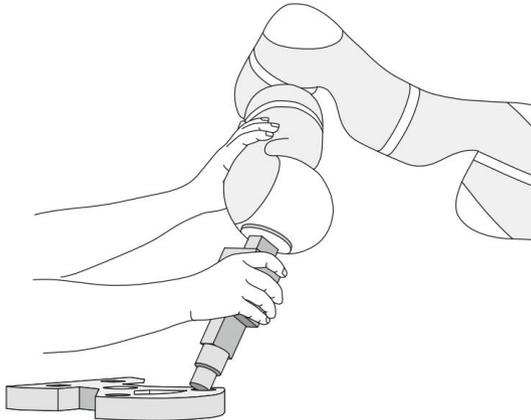


Abb. 14.6 Demonstration der „Plock-in-Loch“-Aufgabe durch kinästhetischen Unterricht. Der menschliche Demonstrator führt den anthropomorphen Arm mit seinen eigenen Händen durch die Aufgabenausführung

14.2.2 Dynamische Bewegungsprimitive

In Abschn. [In Abschnitt 14.2.1](#) haben wir besprochen, wie man menschliche Demonstrationen misst und wie Transformieren Sie die gemessenen Bewegungen in die Gelenkwinkelbahnen des Roboters. In manchen Fällen ist es auch notwendig, die gemessene Bewegung an die kinematischen und dynamischen Fähigkeiten des Zielroboters anzupassen. Normalerweise erhalten wir am Ende eine Messsequenz

$$\{y_d(t_j), t_j\}_{j=1}^T, \quad (14.13)$$

Dabei sind $y_d(t_j) \in \mathbb{R}^D$ die gemessenen Gelenkwinkel zum Zeitpunkt t_j , D die Anzahl der Freiheitsgrade und T die Anzahl der Messungen. Diese Sequenz definiert die Referenztrajektorie. Für eine effektive Steuerung müssen wir jedoch Motorbefehle mit der Servogeschwindigkeit des Zielroboters generieren. Die Servorate des Roboters ist häufig höher als die Erfassungsrate des Messsystems. Daher müssen wir aus den Messdaten (14.13) eine kontinuierliche Referenztrajektorie generieren, um Motorbefehle zur Steuerung des Roboters mit der entsprechenden Geschwindigkeit zu generieren.

In diesem Abschnitt stellen wir Dynamic Movement Primitives (DMPs) vor, die einen umfassenden Rahmen für das effektive Nachahmungslernen und die Steuerung von Roboterbewegungen bieten. DMPs basieren auf einer Reihe nichtlinearer Differentialgleichungen mit wohldefinierter Attraktordynamik. Für einen einzelnen Roboterfreiheitsgrad, hier mit y bezeichnet und als einer der D aufgezeichneten Gelenkwinkel angenommen, wird das folgende System linearer Differentialgleichungen mit konstanten Koeffizienten analysiert, um einen DMP abzuleiten

$$\ddot{y} + \alpha(\dot{y} - \dot{y}_z) + \beta(y - y_z) = \ddot{y}_z, \quad \dot{y}(0) = \dot{y}_z, \quad y(0) = y_z. \quad (14.14)$$

$$(14.15)$$

Beachten Sie, dass die Hilfsvariable z nur eine skalierte Geschwindigkeit der Steuervariablen y ist. Die Konstanten $\ddot{y}z$ und $\dot{y}z$ haben eine Interpretation im Hinblick auf Federsteifigkeit und Dämpfung. Für die entsprechend gewählten Konstanten $\ddot{y}z, \dot{y}z, \ddot{y} > 0$ bilden diese Gleichungen ein global stabiles lineares dynamisches System mit g als eindeutigem Punktattraktor. Wir bezeichnen g oft als das Ziel der Bewegung. Das bedeutet, dass für jede Startkonfiguration $y(0) = y_0$ die Variable y nach einer bestimmten Zeit die Zielkonfiguration g erreicht, so wie eine gedehnte Feder beim Entspannen in ihre Ruheposition zurückkehrt. \dot{y} wird als Zeitkonstante bezeichnet. Es beeinflusst die Konvergenz

14.2.3 Konvergenzeigenschaften linearer dynamischer Systeme

Lassen Sie uns analysieren, warum das obige System nützlich ist. Wir beginnen damit, eine allgemeine Lösung des inhomogenen linearen Differentialgleichungssystems (14.14) und (14.15) aufzuschreiben. Es ist bekannt, dass die allgemeine Lösung eines solchen Systems als Summe der besonderen und homogenen Lösung geschrieben werden kann

$$\begin{aligned} z(t) &= z_p(t) + z_h(t) \\ y(t) &= y_p(t) + y_h(t) \end{aligned} \quad (14.16)$$

Hier $[z_p(t), y_p(t)]^T$ bezeichnet jede Funktion, die das lineare System (14.14) löst – (14.15), während $[z_h(t), y_h(t)]^T$ ist die allgemeine Lösung des homogenen Teils der Gleichungen. (14.14)–(14.15), d. h.

$$\begin{aligned} \dot{z} &= \frac{1}{\dot{y}} \ddot{y}z (\ddot{y}z y + z) = A z, \quad A = \frac{1}{\dot{y}} \begin{bmatrix} \ddot{y}z & \ddot{y}z \dot{y}z \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Es ist leicht zu überprüfen, dass die konstante Funktion $[z_p(0), y_p(0)]^T$ das Gleichungssystem (14.14) und (14.15) löst. Darüber hinaus ist bekannt, dass die allgemeine Lösung von Das homogene System (14.17) ist gegeben durch $[z_h(t), y_h(t)]^T = \exp(At) c$, wobei $c \in \mathbb{R}^2$ eine beliebige Konstante ist. Somit ist die allgemeine Lösung der Gl. (14.14) und (14.15) können geschrieben werden als

$$\begin{aligned} z(t) &= 0 \\ y(t) &= g + \exp(At) c. \end{aligned} \quad (14.17)$$

Die Konstante c sollte aus den Anfangsbedingungen berechnet werden, $[z(0), y(0)]^T = [z_0, y_0]^T$. Die Eigenwerte von A sind gegeben durch $\lambda_{1,2} = \ddot{y}z \pm \dot{y} \sqrt{\frac{2}{z} \ddot{y}z \dot{y}z / (2\dot{y})}$. Lösung, (14.17) konvergiert gegen $[0, g]$, wenn der Realteil der Eigenwerte $\lambda_{1,2}$ kleiner als 0 ist, was für alle $\ddot{y}z, \dot{y}z, \dot{y} > 0$ gilt. Das System ist kritisch gedämpft, was bedeutet, dass y ohne Schwingung und schneller als bei jeder anderen Wahl gegen g konvergiert von A , wenn A zwei gleiche negative Eigenwerte hat. Dies geschieht bei $\dot{y}z = 4\ddot{y}z$ mit $\lambda_{1,2} = \ddot{y}z / (2\dot{y})$.

14.2.4 Dynamische Bewegungselemente für Punkt-zu-Punkt-Bewegungen

Das Differentialgleichungssystem (14.14)–(14.15) stellt sicher, dass y von jedem Startpunkt y_0 gegen g konvergiert. Damit lassen sich einfache Punkt-zu-Punkt-Bewegungen realisieren.

Um einen eher begrenzten Satz von Trajektorien, die durch (14.14) und (14.15) erzeugt werden können, zu vergrößern und so die Erzeugung allgemeiner Punkt-zu-Punkt-Bewegungen zu ermöglichen, können wir Gleichung (1) um eine nichtlineare Komponente hinzufügen. (14.14). Diese nichtlineare Funktion wird oft als Forcierungsterm bezeichnet. Eine Standardauswahl besteht darin, eine lineare Kombination radialer Basisfunktionen \tilde{y}_i hinzuzufügen

$$f(x) = \frac{\sum_{i=1}^N w_i \tilde{y}_i(x)}{\sum_{i=1}^N \tilde{y}_i(x)} x(g - y_0), \quad (14.18)$$

$$\tilde{y}_i(x) = \exp(-h_i(x - c_i)^2), \quad (14.19)$$

Dabei sind c_i die Zentren radialer Basisfunktionen, die entlang der Phase der Trajektorie verteilt sind, und $h_i > 0$. Der Term $g - y_0$, $y_0 = y(t_1)$ wird zur Skalierung der Trajektorie verwendet, wenn sich die Anfangs- und/oder Endkonfiguration ändert. Solange Bewegungsanfang und -ende konstant gehalten werden, hat dieser Skalierungsfaktor keine Auswirkung und kann weggelassen werden. Die Phasenvariable x wird im Forcierungsterm (14.18) anstelle der Zeit verwendet, um die Abhängigkeit der resultierenden Steuerpolitik von der Zeit impliziter zu machen. Seine Dynamik wird definiert durch

$$\dot{x} = \tilde{y} x - \tilde{y} x^2, \quad (14.20)$$

mit dem Anfangswert $x(0) = 1$. Eine Lösung zu (14.20) ist gegeben durch

$$x(t) = \exp(-\int_0^t \tilde{y} dt). \quad (14.21)$$

Die attraktive Eigenschaft der Verwendung der Phasenvariablen x anstelle der expliziten Zeit besteht darin, dass durch entsprechende Modifikation von Gl. (14.20) kann die Entwicklung der Zeit gestoppt werden, um Störungen während der Bewegung zu berücksichtigen. Es besteht keine Notwendigkeit, die interne Uhr des Systems zu verwalten. Wir erhalten das folgende System nichtlinearer Differentialgleichungen

$$\dot{z} = \tilde{y} z - \tilde{y} z^2 + f(x), \quad \dot{y} = z. \quad (14.22)$$

$$(14.23)$$

Die Phasenvariable x und folglich $f(x)$ tendieren mit zunehmender Zeit gegen 0. Daher nimmt der Einfluss des nichtlinearen Termes $f(x)$ mit der Zeit ab. Folglich konvergieren durch die Integration des Systems (14.22)–(14.23) die Systemvariablen z sind garantiert, $[z, y]$ gegen $[0, g]$, T genau wie das lineare System (14.14)–(14.15). Die Kontrollpolitik das durch die Variable y und ihre Ableitungen erster und zweiter Ordnung angegeben wird, was wir ein dynamisches Bewegungsprimitiv nennen (DMP). Für ein System mit vielen Graden von

Freiheit wird jeder Freiheitsgrad durch sein eigenes Differentialgleichungssystem (14.22)–(14.23) dargestellt, wobei die Phase x allen Freiheitsgraden gemeinsam ist. Dies ist möglich, weil Phase Gl. (14.20) enthält keine Variablen y und z .

Normalerweise reicht es aus, die Parameter c_i und h_i von Gl. zu bestimmen. (14.19) durch Festlegen eines vordefinierten Verteilungsmusters und Erhöhen der Anzahl der Basisfunktionen N , bis die gewünschte Rekonstruktionsgenauigkeit erreicht werden kann. Zum Beispiel können wir für ein gegebenes N definieren

$$c_i = \exp(\ddot{y}x) \frac{\text{ich } \ddot{y} 1}{N \ddot{y} 1}, \quad i = 1, \dots, N, \quad (14.24)$$

$$h_i = \frac{2}{(c_{i+1} \ddot{y} c_i)^2}, \quad i = 1, \dots, N \ddot{y} 1, \quad h_N = h_N \ddot{y} 1. \quad (14.25)$$

Beachten Sie, dass $c_1 = 1 = x(0)$ und $c_N = \exp(\ddot{y}x) = x(tT)$.

In den obigen Gleichungen werden $\ddot{y}x$, $\ddot{y}z$ und $\ddot{y}z$ auf konstante Werte gesetzt. Die Werte müssen so gewählt werden, dass die Konvergenz des zugrunde liegenden dynamischen Systems gewährleistet ist, wie in Abschn. 14.2.3. Dies ist der Fall, wenn wir $\ddot{y}x = 2$, $\ddot{y}z = 3$, $\ddot{y}z = 4\ddot{y}z = 12$ setzen.

DMPs wurden entwickelt, um eine Darstellung bereitzustellen, die eine genaue Kodierung der gewünschten Punkt-zu-Punkt-Bewegungen ermöglicht und gleichzeitig die Modulation verschiedener Eigenschaften der kodierten Flugbahn ermöglicht. Dabei werden durch Integration des Gleichungssystems (14.20), (14.22) und (14.23) die Formparameter w_i ermittelt, damit der Roboter der gewünschten Trajektorie genau folgen kann. Die anderen Parameter dienen der Modulation und der Berücksichtigung von Störungen.

Für eine Bewegung mit zwei Freiheitsgraden zeigt Abb. 14.7 eine grafische Darstellung der vom dynamischen Bewegungsgrundelement erzeugten Attraktorfelder. Das Attraktorfeld ändert sich mit der Entwicklung der Phase x . Solange der Roboter der demonstrierten Trajektorie folgt, weist das Attraktorfeld den Roboter an, sich entlang der demonstrierten Trajektorie zu bewegen. Wenn der Roboter jedoch gestört ist und von der gezeigten Flugbahn abweicht, lenken die entlang der Phase x erzeugten Attraktorfelder den Roboter so, dass er die gewünschte Endkonfiguration (Ziel) erreicht, wenn auch entlang einer veränderten Flugbahn.

Eine Flugbahn kann aus einem vollständig spezifizierten DMP durch Integration der Gleichungen reproduziert werden. (14.22), (14.23) und (14.20) unter Verwendung der Euler-Integrationsmethode:

$$z_{k+1} = z_k + \frac{\ddot{y}}{\ddot{y}} (\ddot{y}z(\ddot{y}g \ddot{y} y_k) \ddot{y} z_k) + f(x_k) \ddot{y} t, \quad (14.26)$$

$$y_{k+1} = y_k + \frac{\ddot{y}}{\ddot{y}} z \ddot{y} t, \quad (14.27)$$

$$x_{k+1} = x_k \ddot{y} \frac{\ddot{y}-1}{\ddot{y}} \ddot{y} x \quad x_k \ddot{y} t, \quad (14.28)$$

Dabei ist $\ddot{y}t > 0$ die Integrationskonstante, die normalerweise auf die Servogeschwindigkeit des Roboters eingestellt ist. Die Anfangsparameter für die Integration müssen auf den aktuellen Zustand des Roboters eingestellt werden

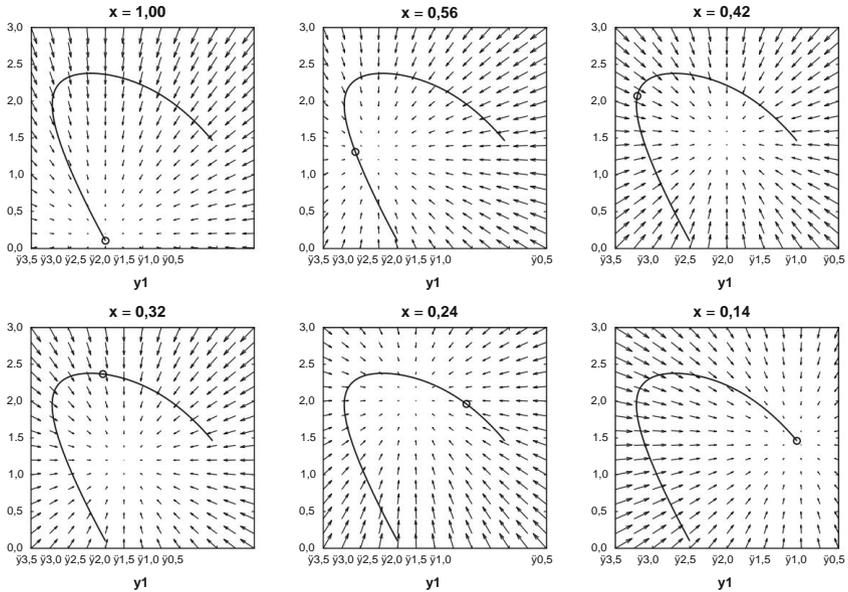


Abb. 14.7 Diagramme der von einem DMP erzeugten Attraktorfelder, die die Bewegung eines Roboters mit zwei angeben Freiheitsgrade y_1, y_2 , da es entlang der Phase x integriert wird. Die Pfeile in jedem Diagramm zeigen \dot{y}_1, \dot{y}_2 an unterschiedliche Werte von y_1, y_2 in der gegebenen Phase x , unter der Annahme, dass sich nur y_1 und y_2 im Vergleich geändert haben zur ungestörten Flugbahn. Die Kreise zeigen die gewünschten Konfigurationen y_1, y_2 bei der gegebenen Phase x

Zu Beginn der Bewegung wird davon ausgegangen, dass sie sich in der angegebenen Ausgangsposition befindet und mit Nullgeschwindigkeit. Daraus ergeben sich folgende Initialisierungsformeln: $y_0 = y_0, \dot{y}_0 = 0, x = 1$.

14.2.5 Schätzung der DMP-Parameter aus einer einzelnen Demonstration

Um den DMP abzuschätzen, der die Messsequenz (14.13) darstellt, berechnen wir zunächst die Ableitungen \ddot{y}^j und \dot{y}^j durch numerische Differentiation. Für alle D-Abschlüsse der Freiheit y erhalten wir die folgende Messsequenz

$$\{y^d(t_j), \dot{y}^d(t_j), \ddot{y}^d(t_j)\}_{j=1}^T, \quad (14.29)$$

Dabei sind $y^d(t_j), \dot{y}^d(t_j), \ddot{y}^d(t_j) \in \mathbb{R}$ die gemessenen Positionen, Geschwindigkeiten und Beschleunigungen auf der Trainingstrajektorie und T die Anzahl der Abtastpunkte. Benutzen Die DMP-Bewegungsdarstellung kann die Flugbahn jeder glatten Bewegung darstellen durch Schätzung der Parameter w_i von Gl. angenähert werden. (14.18). Zu diesem Zweck haben wir

Schreiben Sie das System zweier linearer Gleichungen erster Ordnung um. (14.22) und (14.23) als eine Gleichung zweiter Ordnung. Dies geschieht durch Ersetzen von \dot{z} durch \ddot{y} in Gl. (14.22). Wir erhalten

$$\ddot{y} y'' + \ddot{y} \ddot{y} y \ddot{y} \ddot{y} z (g \ddot{y} y) = f(x), \quad (14.30)$$

mit f definiert wie in Gl. (14.18). Beachten Sie, dass die Zeitkonstante \ddot{y} für alle Freiheitsgrade gleich sein muss. Eine mögliche Wahl ist $\ddot{y} = tT \ddot{y} t1$, wobei $tT \ddot{y} t1$ die Dauer der Trainingsbewegung ist. Andererseits variiert der Attraktorpunkt g über die Freiheitsgrade hinweg. Es kann direkt aus den Daten extrahiert werden: $g = yd(tT)$. Schreiben

$$Fd(tj) = \ddot{y}^2 y''d(tj) + \ddot{y} \ddot{y} y \ddot{y} d(tj) \ddot{y} \ddot{y} z (g \ddot{y} yd(tj)), \quad (14.31)$$

$$f = \begin{matrix} Fd(t1) \\ \dots \\ \ddot{y} Fd(tT) \end{matrix} \ddot{y} \ddot{y}, \quad W = \begin{matrix} w1 \\ \dots \\ \ddot{y} \ddot{y} N \end{matrix} \ddot{y} \ddot{y},$$

wir erhalten das folgende lineare Gleichungssystem

$$Xw = f, \quad (14.32)$$

Dies muss gelöst werden, um die Gewichte eines DMP abzuschätzen, das die gewünschte Bewegung kodiert. Die Systemmatrix X ist gegeben durch

$$X = (g \ddot{y} y0) \begin{matrix} \ddot{y} & \frac{\ddot{y}1(x1)}{N \ddot{y}i(x1)} x1 & \dots & \frac{\ddot{y}N(x1)}{N \ddot{y}i(x1)} x1 \\ & \dots & \dots & \dots \\ \ddot{y} & \frac{\ddot{y}1(xT)}{N \ddot{y}i(xT)} xT & \dots & \frac{\ddot{y}N(xT)}{N \ddot{y}i(xT)} xT \end{matrix} \ddot{y} \ddot{y}. \quad (14.33)$$

Die Phasenabstastpunkte x_j werden durch Einsetzen der Messzeiten t_j in Gleichung erhalten. (14.21). Die Parameter w können durch Lösen des obigen Systems linearer Gleichungen im Sinne der kleinsten Quadrate berechnet werden. Eine beispielhafte DMP-Schätzung ist in Abb. 14.8 dargestellt. Der berechnete DMP stellt sicher, dass der Roboter zum Zeitpunkt tT den Attraktorpunkt g erreicht. Da DMPs so konzipiert sind, dass sie Punkt-zu-Punkt-Bewegungen darstellen, muss die demonstrierte Bewegung am Ende der Demonstration vollständig zum Stillstand kommen, wenn der Roboter nach tT weiterbewegt werden soll. Bewegung durch einen DMP angenähert wird, schießt der Roboter über den Attraktorpunkt hinaus und kehrt zu diesem zurück, nachdem die Dynamik des linearen Differentialgleichungssystems zweiter Ordnung beginnt, die Bewegung zu dominieren. Zumindest theoretisch muss die Geschwindigkeit zu Beginn der Bewegung nicht Null sein, aber es ist schwer, sich ein echtes Demonstrationssystem vorzustellen, mit dem eine solche Flugbahn erf

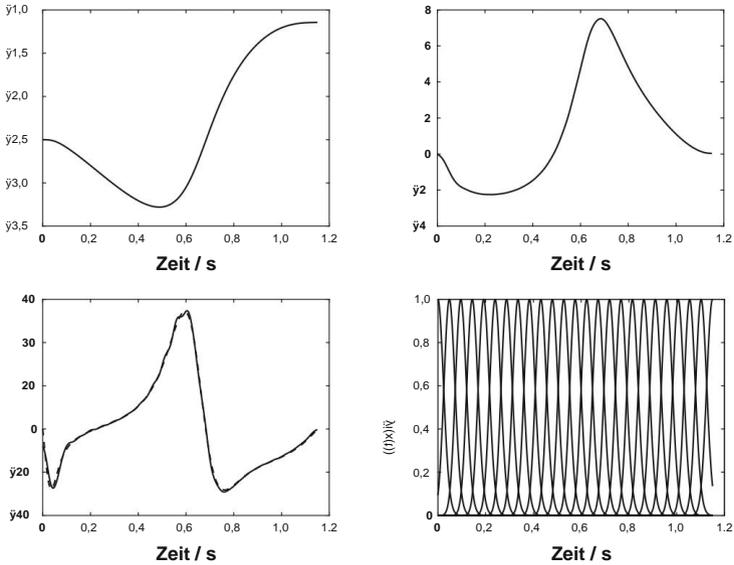


Abb. 14.8 Zeitliche Entwicklung eines beispielhaften dynamischen Bewegungselements: Kontrollvariable y und ihre Ableitungen, Phase x und radiale Basisfunktionen \tilde{y}_i sind alle mit durchgezogenen Linien dargestellt. Gestrichelte Linien zeigen die nachgewiesenen Werte von y , \dot{y} und \ddot{y}

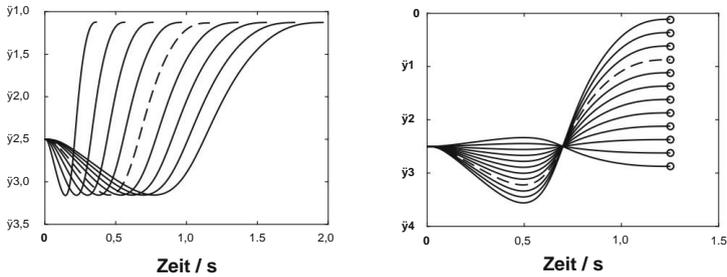


Abb. 14.9 DMP-Modulationen. Die gestrichelten Trajektorien zeigen das ursprüngliche DMP ohne Anwendung Modulation. Links: Zeitmodulation. Durchgezogene Trajektorien zeigen DMPs mit verändertem \tilde{y} . Rechts: Ziel Modulation. Durchgezogene Trajektorien zeigen DMPs mit verändertem Ziel g . Kreise zeigen die Zielposition

14.2.6 Modulation von DMPs

Ein wichtiger Vorteil von DMPs besteht darin, dass sie eine einfache Modulation des Gelernten ermöglichen Bewegung. Abbildung 14.9 links zeigt, dass durch Änderung des Parameters \tilde{y} die Bewegung möglich ist beschleunigt oder verlangsamt werden. Die gleiche Abbildung zeigt dies auch durch die Änderung des Ziels Parameter g , die endgültige Konfiguration auf der Flugbahn kann so geändert werden, dass der Roboter bewegt sich zu einem neuen Ziel. Der Term $y_0 \tilde{y} g$ im forzierenden Term (14.18) sorgt dafür, dass die Die Bewegung wird entsprechend skaliert, wenn sich das Ziel oder die anfängliche Konfiguration ändert.

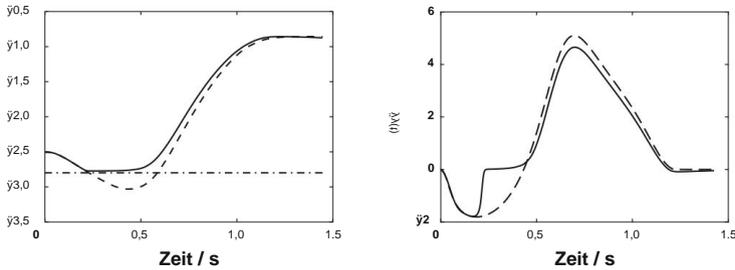


Abb. 14.10 DMP-Modulation mit Joint-Limit-Vermeidung bei $y = \bar{y}2,8$. Die durchgezogenen Trajektorien zeigen die DMP-Flugbahn und ihre Geschwindigkeit, die durch Integration von (14.34) anstelle von (14.23) erhalten wurden, während die gestrichelten Trajektorien das ursprüngliche DMP und seine Geschwindigkeit ohne Anwendung einer Modulation zeigen

Komplexere Modulationen erfordern eine Änderung der zugrunde liegenden Differentialgleichungen. (14.22), (14.23) und/oder (14.20). Zum Beispiel Gl. (14.23) kann geändert werden in

$$\ddot{y} = z \ddot{y} \frac{\dot{y}}{(y_L - \dot{y})^3} \quad (14.34)$$

die Vermeidung einer unteren Gelenkgrenze umzusetzen. Dies geschieht, weil, sobald y beginnt, sich y_L zu nähern, der Nenner in Gl. (14.34) wird klein und es gibt einen signifikanten Unterschied zwischen der Integration von Gl. (14.23) oder (14.34). Abbildung 14.10 rechts zeigt, dass der zweite Term in Gl. (14.34) wirkt als abstoßende Kraft und verhindert, dass y sich y_L zu sehr annähert. Andererseits ist der Nenner in Gl. (14.34) bleibt groß, solange der Gelenkwinkel y weit von der Gelenkgrenze y_L entfernt ist. Daher

gibt es in diesem Fall kaum einen Unterschied zwischen der Integration von Gl. (14.23) oder (14.34) und die vom DMP erzeugte Flugbahn folgt der demonstrierten Bewegung. Beachten Sie, dass es aufgrund der Modulation nicht erforderlich ist, neue Parameter w , Ziel g oder Zeitkonstante \dot{y} zu lernen. Sie können so bleiben, wie sie ursprünglich gelernt wurden. Nur Gl. (14.23) muss in (14.34) geändert werden, um eine gemeinsame Grenzvermeidung während der Online-Steuerung

Die attraktive Eigenschaft der Anwendung der Phasenvariablen anstelle der Zeit besteht darin, dass wir die zeitliche Entwicklung der Phase leicht modulieren können, indem wir beispielsweise eine Bewegung mithilfe von Kopplungstermen je nach Bedarf beschleunigen oder verlangsamen. Anstatt die Gleichungen zu integrieren. (14.20) und (14.23) zum Zeitpunkt der Ausführung, die modifizierten Gleichungen. (14.20) und (14.36) könnten integriert werden

$$\dot{y} \dot{x} = \dot{y} \frac{\dot{y} x}{\dot{y} x + \dot{y} p x (y - \tilde{y}) \dot{y} \dot{y}^2}, \quad (14.35)$$

$$= z + \dot{y} p y (y - \tilde{y}), \quad (14.36)$$

wobei y und \tilde{y} jeweils die gewünschte bzw. tatsächliche Winkelposition des Robotergelenks bezeichnen. Kann der Roboter der gewünschten Bewegung nicht folgen, wird $\dot{y} p x (y - \tilde{y})$ groß, was wiederum die Phasenänderung $\dot{x} \dot{y}$ klein macht. Somit wird die Phasenentwicklung gestoppt, bis der Roboter die gewünschte Konfiguration y eingeholt hat. Dieser Wille

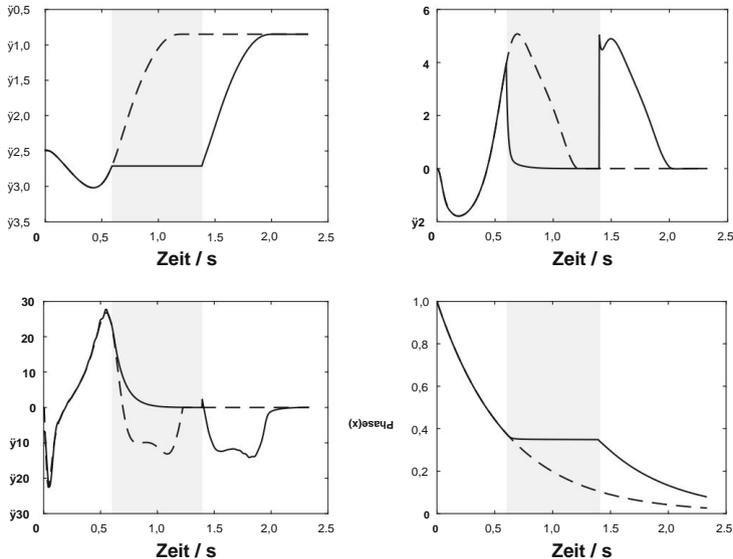


Abb. 14.11 Der Effekt des Phasenstopps, der durch die Blockierung der Entwicklung der Gelenkposition \tilde{y} im Zeitintervall $[0,6, 1,4]$ (grauer Bereich) verursacht wird. Die gestrichelten Trajektorien zeigen die ursprüngliche DMP-, Geschwindigkeits-, Beschleunigungs- und Phasenentwicklung, während die durchgezogenen Trajektorien ihre Gegenstücke aus der gestörten Bewegung mit aktiviertem Phasenstopp zeigen. Beachten Sie, dass der Roboter außerhalb des Zeitintervalls $[0,6, 1,4]$, in dem die Gelenkbewegung blockiert ist, genau der gewünschten Bewegung folgt.

geschieht schließlich aufgrund des hinzugefügten Terms in Gl. (14.36). Folgt der Roboter hingegen der gewünschten Bewegung präzise, dann ist $\ddot{y} - \ddot{y} = 0$ und Gl. (14.35) und (14.36) unterscheiden sich nicht von den Gleichungen (14.20) bzw. (14.23). Somit wird in diesem Fall die durch DMP erzeugte Bewegung nicht verändert. Abbildung 14.11 veranschaulicht die Auswirkung des Phasenstopps, wenn die Bewegung des Roboters vorübergehend blockiert ist.

Zusammenfassend stellen DMPs eine effektive Darstellung zum Erlernen der Flugbahnen humanoider Roboter und zur Steuerung humanoider Roboter dar. Sie basieren auf autonomen, nichtlinearen Differentialgleichungen, die eine reibungslose kinematische Steuerung gewährleisten. Eine wichtige Eigenschaft von DMPs besteht darin, dass sie aus einer einzigen Demonstration der gewünschten Aufgabe erlernt werden können. Sie haben im Vergleich zu anderen Motordarstellungen mehrere Vorteile, darunter:

- Sie verfügen über freie Parameter, die leicht zu erlernen sind, um jedes gewünschte zu reproduzieren Bewegung,
- Sie sind nicht explizit von der Zeit abhängig und ermöglichen eine Zeitmodulation.
- Sie sind robust gegenüber Störungen.
- Sie sind durch Anpassung verschiedener Parameter und Gleichungen leicht zu modulieren.

Aufgrund ihrer Flexibilität und Robustheit gelten DMPs als Methode der Wahl, wenn es darum geht, Robotertrajektorien aus Einzeldemonstrationen zu lernen.

Kapitel 15

Genauigkeit und Wiederholbarkeit industrieller Manipulatoren



In diesem Kapitel werden wir kurz die Leistungskriterien und die Methoden zum Testen von Industrieroboter-Manipulatoren betrachten, wie sie in der Norm ISO 9283 beschrieben sind. Bevor wir uns mit der Genauigkeit und Wiederholbarkeit industrieller Manipulatoren befassen, fassen wir grundlegende Informationen über Roboter-Manipulatoren zusammen.

Zu den grundlegenden Roboterdaten gehört typischerweise eine schematische Zeichnung der mechanischen Struktur des Roboters:

- kartesischer Roboter (Abb. 15.1 links),
- zylindrischer Roboter (Abb. 15.1 rechts),
- polarer (sphärischer) Roboter (Abb. 15.2 links),
- anthropomorpher Roboter (Abb. 15.2 rechts),
- SCARA-Roboter (Abb. 15.3).

In allen Zeichnungen müssen die Freiheitsgrade des Robotermechanismus gekennzeichnet sein. Die Zeichnung muss auch den Basiskoordinatenrahmen und den mechanischen Schnittstellenrahmen enthalten, die vom Hersteller festgelegt werden.

Von besonderer Bedeutung ist das Diagramm, das die Grenzen des Arbeitsbereichs zeigt (Abb. 15.4). Die maximale Reichweite des Roboterarms muss in mindestens zwei Ebenen deutlich dargestellt werden. Für jede Roboterachse muss der Bewegungsbereich (Freiheitsgrad) angegeben werden. Der Hersteller muss auch die Mitte des Arbeitsbereichs c_w angeben, in der die meisten Roboteraktivitäten stattfinden.

Den Roboterdaten müssen die charakteristischen Belastungsparameter wie Masse (kg), Drehmoment (Nm), Trägheitsmoment (kgm^2) und Schubkraft (N) beigefügt sein. Die maximale Geschwindigkeit muss mit einer konstanten Geschwindigkeit angegeben werden, wenn keine Beschleunigung oder Verzögerung erfolgt. Die maximalen Geschwindigkeiten für bestimmte Roboterachsen müssen mit der auf den Endeffektor ausgeübten Last angegeben werden. Außerdem müssen die Auflösung jeder Achsbewegung ($\text{mm } \ddot{y}$), die ^{oder} Beschreibung des Steuerungssystems und der Programmiermethoden dargestellt werden.

Die drei wichtigsten Roboterkoordinatensysteme (rechtshändig) sind in Abb. 15.5 dargestellt. Erstens ist der Weltkoordinatenrahmen $x_0-y_0-z_0$. Der Ursprung des Rahmens wird vom Benutzer definiert. Die z_0 -Achse verläuft parallel zum Schwerkraftvektor, jedoch in entgegengesetzter Richtung. Zweitens ist der Basiskoordinatenrahmen $x_1-y_1-z_1$, dessen Ursprung vom Hersteller definiert wird. Seine Achsen sind auf das Basissegment des Roboters ausgerichtet.

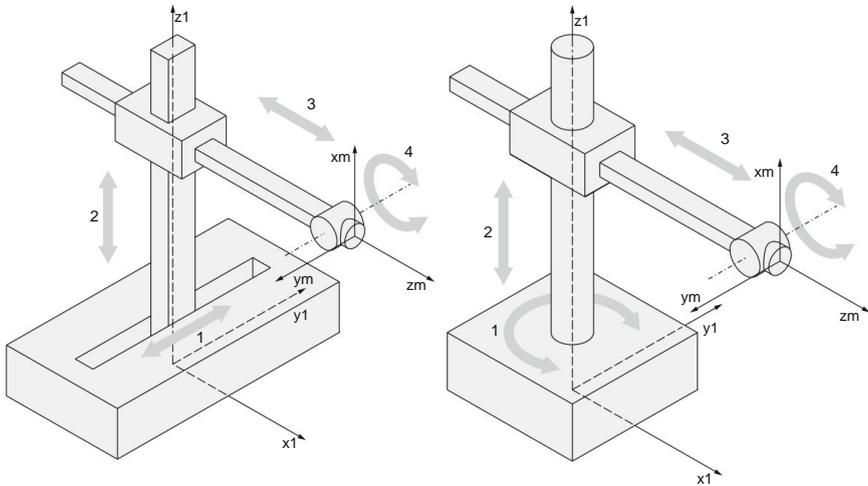


Abb. 15.1 Mechanische Strukturen des kartesischen Roboters (links) und des zylindrischen Roboters (rechts)

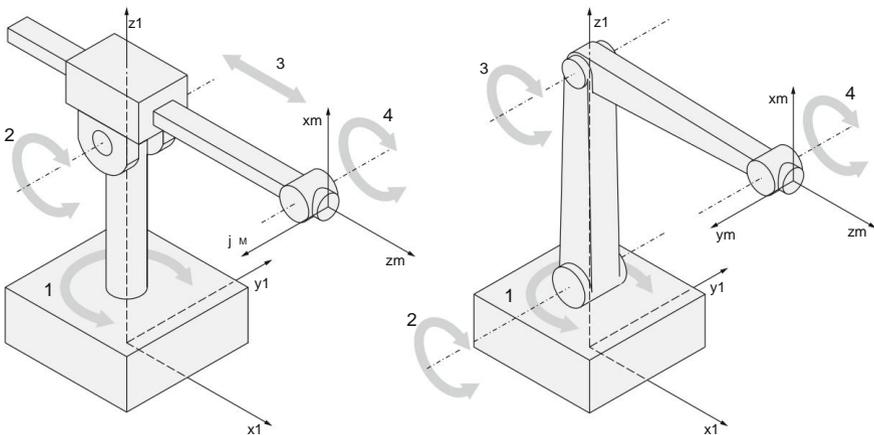


Abb. 15.2 Mechanischer Aufbau des Polarroboters (links) und des anthropomorphen Roboters (rechts)

Die positive z_1 -Achse zeigt senkrecht von der Basismontagefläche weg.

Die x_1 -Achse verläuft im Uhrzeigersinn durch die Projektion der Mitte des Roboterarbeitsraums.

Der Rahmen x_m - y_m - z_m wird als Koordinatenrahmen der mechanischen Schnittstelle

bezeichnet. Sein Ursprung liegt in der Mitte der mechanischen Schnittstelle

(Roboterhandfläche), die den Roboterarm mit dem Greifer verbindet. Die positive z_m -Achse zeigt von der mechanischen Schnittstelle weg in Richtung Endeffektor. Die x_m -Achse liegt in der durch die Grenzfläche definierten Ebene, die senkrecht zur z_m -Achse steht.

Die positiven Richtungen der Roboterbewegungen, angegeben als translatorische und rotatorische Verschiebungen, sind in Abb. 15.6 dargestellt.

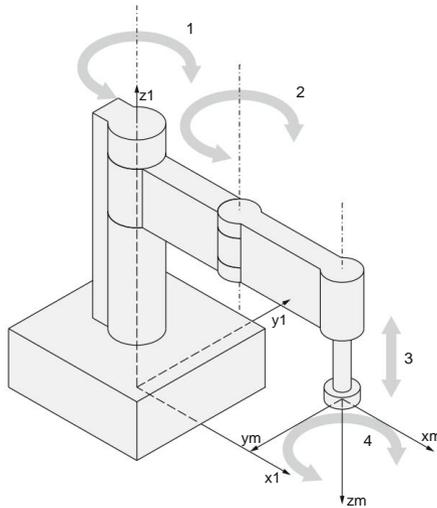


Abb. 15.3 Mechanischer Aufbau des SCARA-Roboters

Die Norm ISO 9283 befasst sich mit Kriterien und Methoden zum Testen von Industrieroboter-Manipulatoren. Dies ist der wichtigste Standard, da er den Dialog zwischen Herstellern und Anwendern der Robotersysteme erleichtert. Es definiert die Art und Weise, wie bestimmte Leistungsmerkmale eines Roboter-Manipulators getestet werden sollen. Die Tests können während der Roboterabnahmephase oder in verschiedenen Einsatzphasen des Roboters durchgeführt werden, um die Genauigkeit und Wiederholbarkeit der Roboterbewegungen zu überprüfen. Die Roboter-Eigenschaften, die die Leistung einer Roboter-Aufgabe maßgeblich beeinflussen, sind folgende:

- Posengenauigkeit und Wiederholbarkeit (Pose ist definiert als Position und Ausrichtung eines bestimmtes Robotersegment, normalerweise Endeffektor),
- Abstandsgenauigkeit und Wiederholbarkeit,

• Posenstabilisierungszeit, • Posenüberschwinger, • Drift der Posengenauigkeit und Wiederholbarkeit.

Diese Leistungsparameter sind bei Punkt-zu-Punkt-Roboter-Aufgaben wichtig. Ähnliche Parameter werden für Fälle definiert, in denen sich der Roboter-Endeffektor entlang einer kontinuierlichen Bahn bewegt. Diese Parameter werden in diesem Buch nicht berücksichtigt und sind den Originaldokumenten zu entnehmen.

Beim Testen der Genauigkeit und Wiederholbarkeit eines Robotermechanismus sind zwei Begriffe wichtig, nämlich der Cluster und das Cluster-Schwerpunkt. Der Cluster ist als eine Reihe erreichter Endeffektor-Posen definiert, die derselben Befehlspose entsprechen. Der Schwerpunkt ist ein Punkt, dessen Koordinaten die Mittelwerte der x -, y - und z -Koordinaten aller Punkte im Cluster sind. Die gemessenen Positions- und Orientierungsdaten müssen in einem Koordinatensystem parallel zum Basissystem ausgedrückt werden.

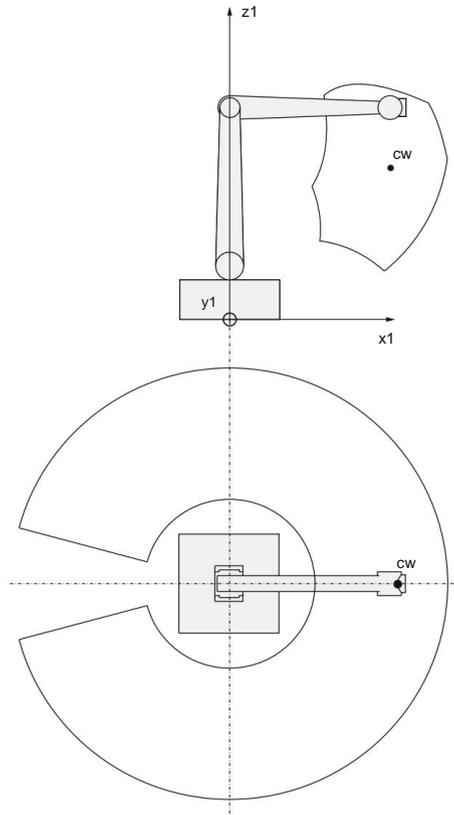


Abb. 15.4 Roboterarbeitsbereich

Der Messpunkt sollte möglichst nahe am Ursprung des mechanischen Schnittstellenrahmens liegen. Empfehlenswert sind berührungslose optische Messverfahren. Die Messgeräte müssen ausreichend kalibriert sein. Die Genauigkeits- und Wiederholbarkeitstests des Roboters müssen mit maximaler Last am Endeffektor und maximaler Geschwindigkeit zwischen den angegebenen Punkten durchgeführt werden.

Der Standard definiert die Posen, die getestet werden sollen. Die Messungen müssen an fünf Punkten durchgeführt werden, die sich in einer Ebene befinden, die diagonal innerhalb eines Würfels liegt (Abb. 15.7). Außerdem wird die Pose des Würfels im Arbeitsbereich des Roboters angegeben. Es sollte sich in dem Teil des Arbeitsbereichs befinden, in dem die meisten Roboteraktivitäten erwartet werden. Der Würfel muss das maximal zulässige Volumen im Arbeitsbereich des Roboters haben und seine Kanten sollten parallel zum Basiskoordinatenrahmen sein. Der Punkt P1 liegt im Schnittpunkt der Diagonalen in der Würfelmitte. Die Punkte P2 – P5 liegen in einem Abstand von $10\% \pm 2\%$ der Länge der Diagonale L von den Ecken des Würfels. Die Norm legt auch die Mindestanzahl der Zyklen fest, die bei der Prüfung jedes charakteristischen P

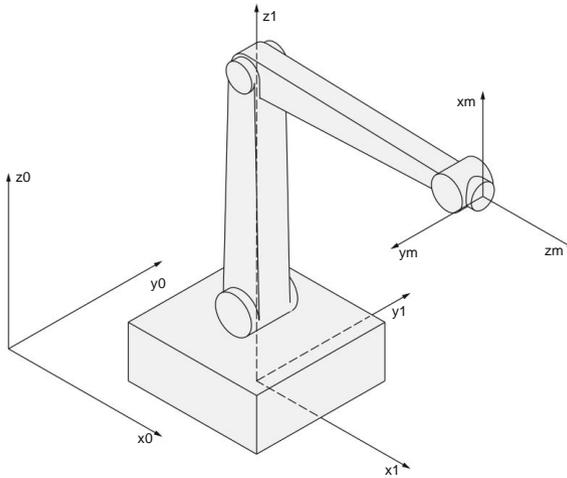


Abb. 15.5 Die Koordinatensysteme des Roboterarmes

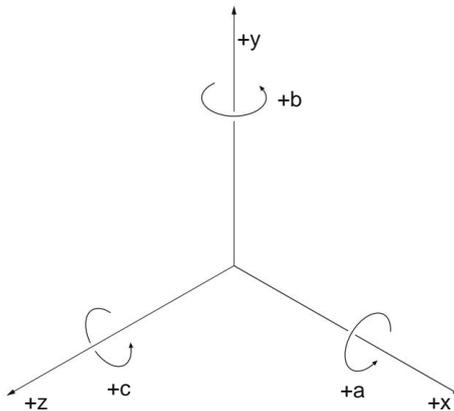


Abb. 15.6 Positive Richtungen translatorischer und rotatorischer Verschiebungen

• Posengenauigkeit und Wiederholbarkeit: 30

Zyklen, • Distanzgenauigkeit und Wiederholbarkeit: 30

Zyklen, • Posenstabilisierungszeit: 3

Zyklen, • Posenüberschreitung:

3 Zyklen, • Drift der Posengenauigkeit und Wiederholbarkeit: kontinuierlicher Zyklus während 8 Stunden.

Beim Testen der Genauigkeit und Wiederholbarkeit der Endeffektorposen müssen wir zwischen der sogenannten Kommandopose und der erreichten Pose unterscheiden (Abb. 15.8).

Die Befehlspose ist die gewünschte Pose, die durch Roboterprogrammierung oder manuelle Eingabe der gewünschten Koordinaten über ein Programmierhandgerät festgelegt wird. Die erreichte Pose ist die tatsächlich erreichte Pose des Roboter-Endeffektors als Reaktion auf den Befehl.

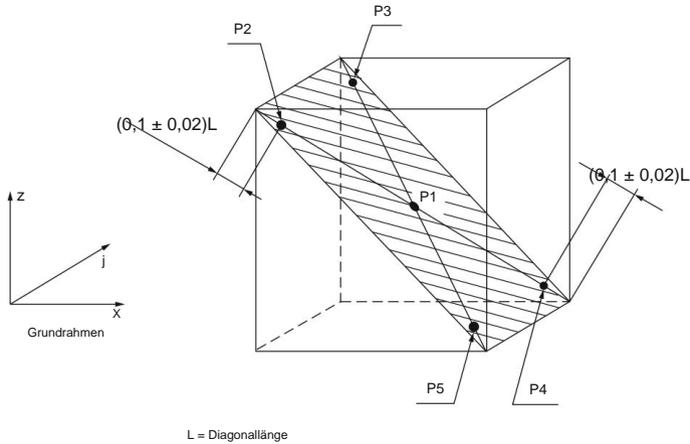


Abb. 15.7 Der Würfel mit den zu prüfenden Punkten

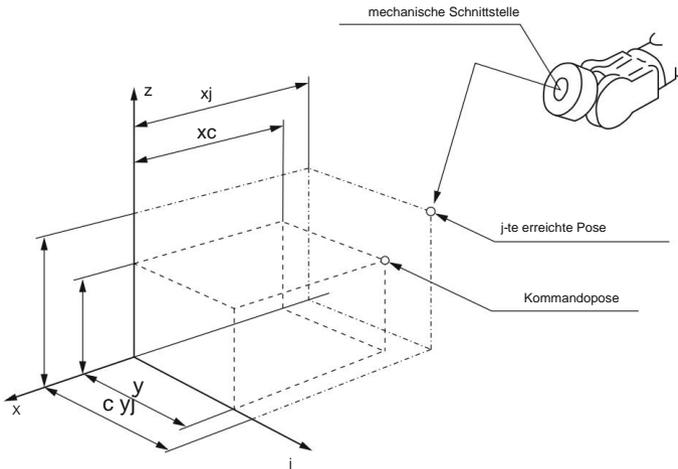


Abb. 15.8 Die Kommandopose und die erreichte Endeffektorpose

Pose. Die Posengenauigkeit bewertet die Abweichungen, die zwischen dem Befehl und der erreichten Pose auftreten. Die Posenwiederholbarkeit schätzt die Schwankungen der erreichten Posen für eine Reihe wiederholter Besuche derselben Befehlspose. Die Posengenauigkeit und Wiederholbarkeit sind daher der Genauigkeit und Wiederholbarkeit beim wiederholten Schießen auf ein Ziel sehr ähnlich. Die Gründe für die Abweichungen sind: durch den Steuerungsalgorithmus verursachte Fehler, Koordinatentransformationsfehler, Unterschiede zwischen den Abmessungen der mechanischen Struktur des Roboters und dem Robotersteuerungsmodell, mechanische Fehler wie Hysterese oder Reibung sowie äußere Einflüsse.

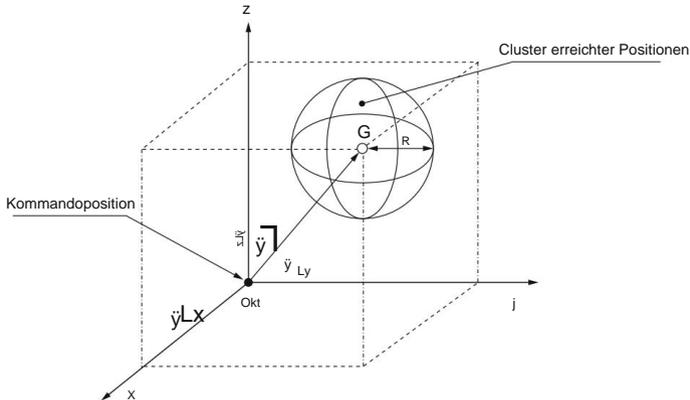


Abb. 15.9 Die Positionsgenauigkeit und Wiederholbarkeit

Die Posengenaugkeit ist definiert als die Abweichung zwischen der Kommandopose und dem Mittelwert der erreichten Posen, wenn sich der Endeffektor der Kommandopose aus derselben Richtung näherte. Die Positions- und Orientierungsgenauigkeit werden separat behandelt. Die Positionsgenauigkeit wird durch den Abstand zwischen der Kommandopose und dem Schwerpunkt des Clusters erreichter Posen bestimmt (Abb. 15.9). Die Positionsgenauigkeit $L = [L_x \ L_y \ L_z]^T$ wird durch die folgende Gleichung ausgedrückt

$$L = \sqrt{(x^- - \bar{x})^2 + (y^- - \bar{y})^2 + (z^- - \bar{z})^2}, \quad (15.1)$$

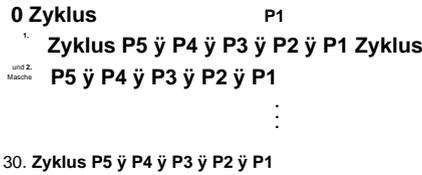
wobei (x^-, y^-, z^-) die Koordinaten des Schwerpunkts des Clusters sind, die durch Mittelung der 30 Messpunkte ermittelt werden, die durch Wiederholen der Bewegungen in die gleiche Befehlspose O_c mit den Koordinaten (x_c, y_c, z_c) ermittelt werden.

Die Ausrichtungsgenauigkeit ist die Differenz zwischen der befohlenen Winkelausrichtung und dem Durchschnitt der erreichten Winkelausrichtungen. Sie wird für jede Achse des Basiskoordinatensystems separat ausgedrückt. Die Orientierungsgenauigkeit um die z-Achse hat die folgende Form

$$L_c = C^- \bar{y} C_c, \quad (15.2)$$

Dabei ist C^- der Mittelwert der Orientierungswinkel um die z-Achse, der in 30 Messungen beim Versuch, denselben Sollwinkel C_c zu erreichen, ermittelt wurde. Ähnliche Gleichungen werden für die Orientierungsgenauigkeit um die x- und y-Achse geschrieben.

Die Norm legt auch den Ablauf der Messungen genau fest. Der Roboter startet am Punkt P1 und bewegt sich zu den Punkten P5, P4, P3, P2, P1. Jeder Punkt wird immer aus der gleichen Richtung erreicht



Für jeden Punkt werden die Positionsgenauigkeit L und die Orientierungsgenauigkeiten L_a , L_b und L_c berechnet.

Für die gleiche Messreihe ist auch die Posenwiederholbarkeit zu ermitteln. Die Posenwiederholbarkeit drückt die Nähe der Positionen und Ausrichtungen der 30 erreichten Posen aus, wenn die Roboterbewegungen in derselben Befehlspose wiederholt werden. Die Positionswiederholgenauigkeit (Abb. 15.9) wird durch den Radius der Kugel r bestimmt, deren Mittelpunkt das Schwerpunktzentrum des Clusters ist. Der Radius

$$r = D^- + 3SD. \quad (15.3)$$

Die Berechnung des Radius r nach Gl. (15.3) wird durch die folgenden Gleichungen weiter erläutert

$$D^- = \frac{1}{N} \sum_{j=1}^N D_j$$

$$D_j = \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2} \quad (15.4)$$

$$SD = \frac{\sum_{j=1}^N (D_j - D^-)^2}{n - 1}$$

In den obigen Gleichungen wählen wir wiederum $n = 30$, wobei (x_j, y_j, z_j) die Koordinaten der j -ten erreichten Position sind.

Die Wiederholgenauigkeit der Orientierung für den Winkel um die z -Achse ist in Abb. 15.10 dargestellt. Die Orientierungswiederholbarkeit drückt aus, wie stark die 30 erreichten Winkel um ihren Durchschnitt für denselben Befehlswinkel gestreut sind. Sie wird durch die dreifachen Standardabweichungen beschrieben. Für den Winkel um die z -Achse

$$rc = \pm 3Sc = \pm 3 \sqrt{\frac{\sum_{j=1}^N (C_j - \bar{C})^2}{n - 1}} \quad (15.5)$$

In Gl. (15.5) C_j stellt den Winkel dar, der bei der j -ten erreichten Pose gemessen wird. Der Ablauf der Messungen ist der gleiche wie bei der Prüfung der Genauigkeit. Der Radius r und die Winkelabweichungen ra , rb und rc werden für jede Pose separat berechnet.

Die Distanzgenauigkeit und Wiederholgenauigkeit werden auf ähnliche Weise getestet. Die Distanzgenauigkeit quantifiziert die Abweichungen, die in der Distanz zwischen zwei Sollpositionen und zwei Sätzen der durchschnittlich erreichten Positionen auftreten. Die Distanzwiederhol-

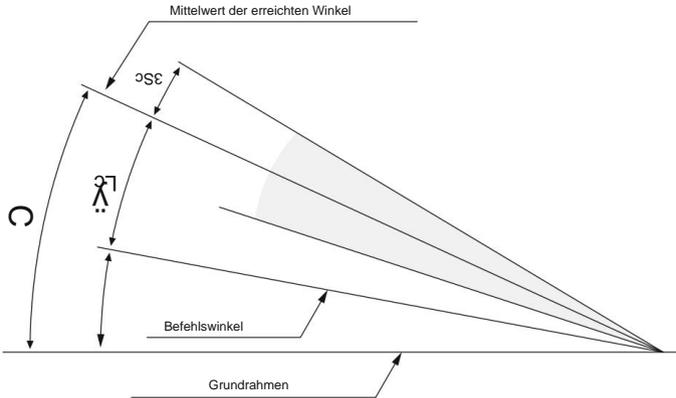


Abb. 15.10 Die Orientierungsgenauigkeit und Wiederholbarkeit

Bestimmt die Abstandsschwankungen für eine Reihe wiederholter Roboterbewegungen zwischen zwei ausgewählten Punkten. Die Distanzgenauigkeit ist definiert als die Abweichung zwischen der Soll-Distanz und dem Mittelwert der erreichten Distanzen (Abb. 15.11). Unter der Annahme, dass P_{c1} und P_{c2} das befohlene Positionspaar sind und P_{1j} und P_{2j} das j -te Paar der 30 Paare der erreichten Positionen sind, ist die Distanzgenauigkeit als

$$B = D_c \cdot \bar{D} \quad (15.6)$$

Wo

$$D_c = P_{c1} \cdot P_{c2} = (x_{c1} \cdot x_{c2})^2 + (y_{c1} \cdot y_{c2})^2 + (z_{c1} \cdot z_{c2})^2$$

$$\bar{D} = \frac{1}{N} \sum_{j=1}^N D_j$$

$$D_j = P_{1j} \cdot P_{2j} = (x_{1j} \cdot x_{2j})^2 + (y_{1j} \cdot y_{2j})^2 + (z_{1j} \cdot z_{2j})^2$$

In den obigen Gleichungen, die die Distanzgenauigkeit beschreiben, stellen $P_{c1} = (x_{c1}, y_{c1}, z_{c1})$ und $P_{c2} = (x_{c2}, y_{c2}, z_{c2})$ das Paar gewünschter Positionen dar, während $P_{1j} = (x_{1j}, y_{1j}, z_{1j})$ und $P_{2j} = (x_{2j}, y_{2j}, z_{2j})$ sind das Paar der erreichten Positionen. Der Abstandsgenauigkeitstest wird bei maximaler Belastung des Roboter-Endeffektors durchgeführt, der 30-mal zwischen den Punkten P2 und P4 des Messwürfels verschoben werden muss. Die Distanzwiederholbarkeit RB ist definiert als

$$RB = \pm 3 \cdot \frac{\sum_{j=1}^N (D_j \cdot \bar{D})^2}{n \cdot \bar{D}} \quad (15.7)$$

Betrachten wir weitere vier charakteristische Parameter, die bei Industrierobotern, die sich von Punkt zu Punkt bewegen, getestet werden sollten. Das erste ist die Posenstabilisierung

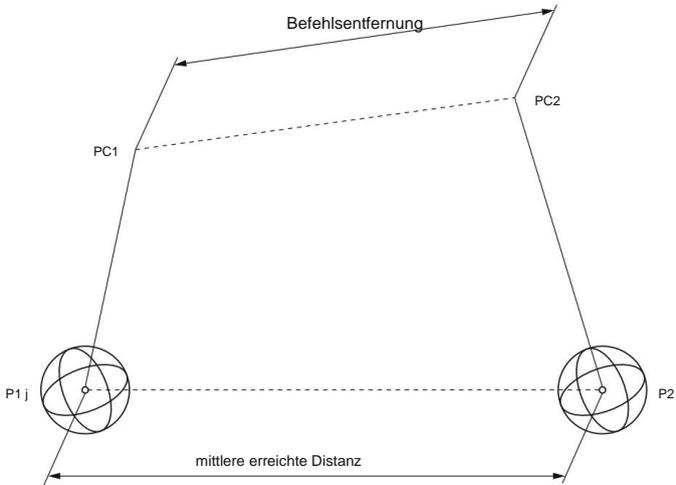


Abb. 15.11 Distanzgenauigkeit

Zeit. Die Stabilisierungszeit ist das Zeitintervall zwischen dem Zeitpunkt, an dem der Roboter das Signal „erreichte Pose“ und den Zeitpunkt, zu dem entweder oszilliert oder gedämpft wird. Die Bewegung des Endeffektors des Roboters liegt innerhalb eines vom Hersteller angegebenen Grenzwerts. Die Definition der Posenstabilisierungszeit ist aus Abb. 15.12 ersichtlich. Der Test ist mit maximaler Belastung und Geschwindigkeit durchgeführt. Alle fünf Messstellen werden besucht. Reihenfolge P1 → P2 → P3 → P4 → P5. Für jede Pose der Mittelwert von Es werden drei Zyklen berechnet.

Ein ähnlicher Parameter ist das Pose Overshoot, ebenfalls in Abb. 15.12 dargestellt. Die Pose Überschwingen ist die maximale Abweichung zwischen der sich nähernden Endeffektor-Flugbahn und die erreichte Pose, nachdem der Roboter das Signal „erreichte Pose“ gegeben hat. In Abb. 15.12 Im ersten Beispiel ist ein negatives Überschwingen und im folgenden ein positives Überschwingen dargestellt zweites Beispiel. Der Zeitpunkt $t = 0$ ist der Zeitpunkt, zu dem das Signal „erreichte Pose“ erfolgte geliefert. Die Messbedingungen sind die gleichen wie bei der Prüfung der Stabilisierung Zeit.

Die letzten beiden Parameter werden beim Bewegen des Industrieroboter-Manipulators getestet von Punkt zu Punkt sind die Drift der Posengenauigkeit und die Drift der Posenwiederholbarkeit. Die Drift der Positionsgenauigkeit L_{DR} ist definiert als

$$L_{DR} = |L_{t=0} - L_{t=T}|, \quad (15.8)$$

wobei $L_{t=0}$ und $L_{t=T}$ die Positionsgenauigkeitswerte zum Zeitpunkt $t = 0$ und zur Zeit sind $t=T$. Die Drift der Orientierungsgenauigkeit L_{OR} ist gleich

$$L_{OR} = |L_{c,t=0} - L_{c,t=T}|, \quad (15.9)$$

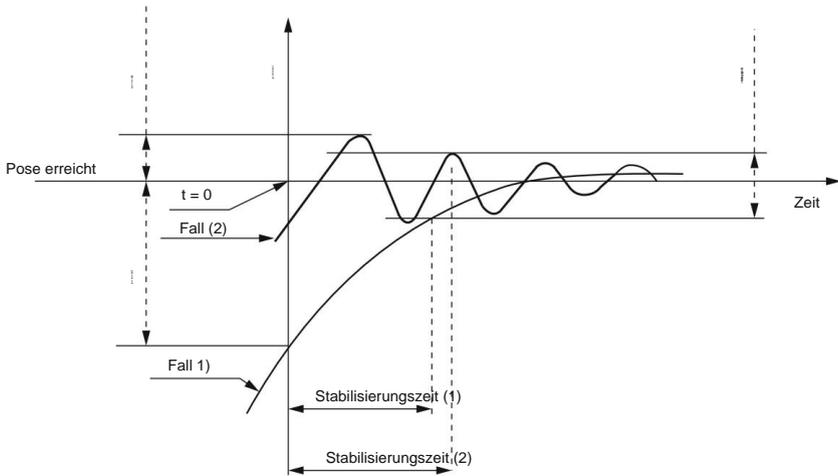


Abb. 15.12 Stabilisierungszeit und Überschwingen der Pose

Dabei sind $L_{c,t=0}$ und $L_{c,t=T}$ die Orientierungsgenauigkeitswerte zum Zeitpunkt $t = 0$ bzw. zum Zeitpunkt $t=T$. Die Drift der Positionswiederholgenauigkeit wird durch die folgende Gleichung definiert

$$r_{DR} = r_{t=0} \dot{y} r_{t=T}, \quad (15.10)$$

Dabei sind $r_{t=0}$ und $r_{t=T}$ die Positionswiederholbarkeitswerte zum Zeitpunkt $t = 0$ bzw. zum Zeitpunkt $t=T$. Die Drift der Orientierungswiederholbarkeit ist für die Drehung um die z- Achse definiert als

$$r_{DRC} = r_{c,t=0} \dot{y} r_{c,t=T}, \quad (15.11)$$

Dabei sind $r_{c,t=0}$ und $r_{c,t=T}$ die Orientierungswiederholbarkeitswerte zum Zeitpunkt $t = 0$ bzw. zum Zeitpunkt $t=T$. Die Messungen werden bei maximaler Roboterbelastung und -geschwindigkeit durchgeführt. Der Roboter wird zyklisch zwischen den Punkten P4 und P2 verschoben. Die zyklischen Bewegungen dauern acht Stunden. Messungen werden nur im Punkt P4 durchgeführt.

Anhang A

Ableitung der Beschleunigung im Kreis

Bewegung

Erinnern wir uns zunächst an die Definitionen von Ortsvektor, Geschwindigkeit und Beschleunigung eines Teilchens. In einem gegebenen Bezugssystem (dh einem Koordinatensystem) wird die Position eines Partikels durch einen Vektor angegeben, der sich vom Ursprung des Koordinatensystems zur momentanen Position des Partikels erstreckt. Dieser Vektor könnte eine Funktion der Zeit sein und somit die Flugbahn des Teilchens angeben

$$\mathbf{r}(t) = (x(t), y(t), z(t)). \quad (\text{A.1})$$

Die Geschwindigkeit des Teilchens ist definiert als die Ortsänderung pro Zeiteinheit

$$\mathbf{v} = \lim_{\Delta t \rightarrow 0} \frac{\Delta \mathbf{r}}{\Delta t} = \frac{d\mathbf{r}}{dt}. \quad (\text{A.2})$$

Die Beschleunigung ist definiert als die Geschwindigkeitsänderung pro Zeiteinheit,

$$\mathbf{a} = \lim_{\Delta t \rightarrow 0} \frac{\Delta \mathbf{v}}{\Delta t} = \frac{d\mathbf{v}}{dt}. \quad (\text{A.3})$$

Wir stellen fest, dass es sich hierbei um eine Vektorgleichung handelt, sodass sich die Geschwindigkeitsänderung sowohl auf eine Änderung der Größe der Geschwindigkeit als auch auf eine Änderung der Geschwindigkeitsrichtung bezieht.

Die Kreisbewegung wird durch einen rotierenden Vektor $\mathbf{r}(t)$ fester Länge $|\mathbf{r}|$ beschrieben = konstant. Der Ortsvektor wird somit durch den Radius des Kreises r und durch den Winkel $\dot{\varphi}(t)$ von \mathbf{r} bezüglich der x -Achse bestimmt (Abb. A.1).

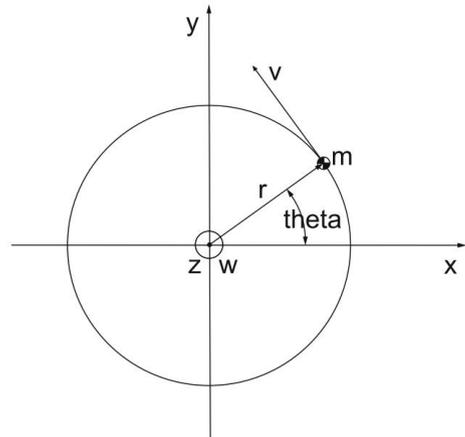
Lassen Sie uns nun einen Satz von drei orthogonalen Einheitsvektoren einführen: \mathbf{e}_r in Richtung von \mathbf{r} , \mathbf{e}_t in Richtung der Tangente an den Kreis und \mathbf{e}_z in Richtung der z -Achse. Die Beziehung zwischen den drei Einheitsvektoren ist gegeben durch $\mathbf{e}_t = \mathbf{e}_z \times \mathbf{e}_r$.

Wir definieren den Winkelgeschwindigkeitsvektor als senkrecht zur Ebene des kreisförmigen Flugbahn mit einer Größe gleich der zeitlichen Ableitung des Winkels $\dot{\varphi}$

$$\dot{\varphi} = \dot{\varphi} \mathbf{e}_z. \quad (\text{A.4})$$

Fahren wir mit der Berechnung der Geschwindigkeit fort

Abb. A.1 Parameter und Variablen in Kreisbewegung



$$\mathbf{v} = \frac{D\mathbf{R}}{dt} \quad (\text{A.5})$$

Die Geschwindigkeitsrichtung ist durch die Tangente an den Kreis gegeben: $\mathbf{e}_t = \mathbf{e}_z \times \mathbf{e}_r$. Der Betrag der Geschwindigkeit ergibt sich aus der Länge des infinitesimalen Kreisbogens $ds = r d\theta$ dividiert durch die infinitesimale Zeit dt , die das Teilchen benötigt, um diesen Weg zurückzulegen

$$\frac{ds}{dt} = r \frac{d\theta}{dt} = r \dot{\theta} \quad (\text{A.6})$$

Einschließlich der tangentialen Geschwindigkeitsrichtung ergibt sich

$$\mathbf{v} = \dot{\theta} \mathbf{e}_t = \dot{\theta} \mathbf{e}_z \times \mathbf{e}_r = \dot{\theta} \times \mathbf{r} \quad (\text{A.7})$$

Um die Beschleunigung zu erhalten, berechnen wir die zeitliche Ableitung der Geschwindigkeit

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d}{dt}(\dot{\theta} \times \mathbf{r}) \quad (\text{A.8})$$

Wir differenzieren das Vektorprodukt so, wie man ein normales Produkt zweier Funktionen differenzieren würde

$$\mathbf{a} = \frac{d\dot{\theta}}{dt} \times \mathbf{r} + \dot{\theta} \times \frac{d\mathbf{r}}{dt} \quad (\text{A.9})$$

Definieren der Winkelbeschleunigung $\ddot{\theta}$ als zeitliche Ableitung der Winkelgeschwindigkeit $\dot{\theta} = \frac{d\theta}{dt}$, Wir sehen, dass der erste Term der Tangentialbeschleunigung entspricht

$$\text{bei} = \ddot{\theta} \times \mathbf{r} \quad (\text{A.10})$$

Im zweiten Term fügen wir den Ausdruck für die Geschwindigkeit ein

$$\frac{d\mathbf{r}}{dt} = \mathbf{v} = \dot{\mathbf{y}} \times \mathbf{r}, \quad (\text{A.11})$$

und wir erhalten ein Doppelvektorprodukt $\dot{\mathbf{y}} \times (\dot{\mathbf{y}} \times \mathbf{r})$. Unter Verwendung der Identität $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$ aus der Vektoralgebra und der Tatsache, dass $\dot{\mathbf{y}}$ und \mathbf{r} orthogonal sind, erhalten wir für den zweiten Term in der Beschleunigungsgleichung

$$\dot{\mathbf{y}} \times \frac{d\mathbf{r}}{dt} = \dot{\mathbf{y}} \times (\dot{\mathbf{y}} \times \mathbf{r}) = \dot{\mathbf{y}}(\dot{\mathbf{y}} \cdot \mathbf{r}) - \mathbf{r}(\dot{\mathbf{y}} \cdot \dot{\mathbf{y}}) = -\dot{\mathbf{y}}\dot{\mathbf{y}} \quad \text{2 r}, \quad (\text{A.12})$$

Dies ist die radiale (oder zentripetale) Komponente der Beschleunigung. Endlich haben wir es geschafft

$$\mathbf{a} = \mathbf{a}_t + \mathbf{a}_r = \dot{\mathbf{y}} \times \mathbf{r} - \dot{\mathbf{y}}\dot{\mathbf{y}} \quad \text{2 r}. \quad (\text{A.13})$$

Weiterführende Literatur

1. Bajd T, Mihelj M, Munih M (2013) Introduction to Robotics, Springer 2. Craig JJ (2005) Introduction to Robotics – Mechanics and Control, Pearson Prentice Hall
3. Kajita S, Hirukawa H, Harada K, Yokoi K (2014) Introduction to Humanoid Robotics, Springer 4. Klanýcar G, Zdešar A, Blažič S, Škrjanc I (2017) Wheeled Mobile Robotics – Von den Grundlagen zu autonomen Systemen, Elsevier 5. Lenarčič J, Bajd T, Stanišič MM (2013) Robot Mechanisms, Springer 6. Merlet JP (2006) Parallel Robots (Zweite Auflage), Springer 7. Mihelj M, Podobnik J (2012) Haptik für Virtual Reality und Teleoperation, Springer
8. Mihelj M, Novak D, Beguš S (2014) Virtual Reality Technology and Applications, Springer
9. Natale C (2003) Interaction Control of Robot Manipulators, Springer 10. Nof SY (1999) Handbook of Industrial Robotics, John Wiley & Sons 11. Paul RP (1981) Robot Manipulators: Mathematics, Programming, and Control, Die MIT-Press
12. Sciavicco L, Siciliano B (2002) Modellierung und Steuerung von Roboter-Manipulatoren, Springer
13. Spong MW, Hutchinson S, Vidyasagar M (2006) Robot Modeling and Control, John Wiley & Sons
14. Tsai LW (1999) Roboteranalyse: Die Mechanik der seriellen und parallelen Manipulatoren, John Wiley & Sons
15. Xie M (2003) Grundlagen der Robotik – Verknüpfung von Wahrnehmung mit Aktion, Welt Wissenschaftlich

Index

A

Winkelbeschleunigung,
 62, 244 zentripetal, 62,
 245 radial, 62,
 245 tangential, 62, 244

Beschleunigungsmesser, 94

Genauigkeit
 Entfernung, 239
 Drift, 233, 240
 Ausrichtung, 237
 Pose, 233, 235
 Position, 237

Aktuator
 flexibel, 175
 Serie elastisch, 175
 variable Steifigkeit, 175

Anthropomorpher Roboter, 6, 231

Bewerbung
 handgeführt, 187
 Übergabefenster, 185

Versammlung, 147

Attraktorpunkt, 222, 226

Fahrerloses Transportfahrzeug, 189

B

Rückprojektion, 110, 113, 114 modellbasiert,
 115

Saldo, 210

Stoßstange, 98

C

Kamera, 185
 Kalibrierung, 114, 118

Kalibrierungsmuster, 118
 Kalibrierungsspitze,
 118 extrinsische Parameter,
 114 intrinsische Parameter, 112

Kartesischer Roboter, 231

Schwerpunkt (COM), 214, 215

Geschlossene kinematische Kette, 69

Cobot, 9

Kollaborative
 Anwendung, 185
 Greifer, 184
 Bedienung, 173, 177
 Roboter, 157, 173, 175, 184
 Arbeitsbereich, 174, 186

Kollision, 175

Compliance, 175

Einschränkungen, 71, 77

Kontakt
 quasistatisch, 182
 transient, 182

Kontaktsensor, 96

Kontrolle, 133
 Admittanz, 176
 Kraft, 133, 147, 149, 151
 Schwerkraftkompensation, 136, 137, 144
 Impedanz, 176
 inverse Dynamik, 137, 139, 144, 148,
 152
 inverse Jacobi-Funktion,
 143 Orientierung,
 202 parallele Zusammensetzung,
 151 Position,
 133 Position und Orientierung, 203
 Proportional-Ableitung, 135, 136, 144
 Drehmoment,
 176 transponierte Jacobi-Funktion, 142

Förderband, 164

- Riemenantrieb,
 - 165 Kette,
 - 164 Metallkette,
 - 164 Rolle, 165
- Förderband, 158
- Kooperation, 173

- Koordinatenrahmenbasis, 231 Kamera,
 - 109 global,
 - 189, 194,
 - 200, 204,
 - 206 Bild, 109 Index, 111 lokal,
 - 191 mechanische
- Schnittstelle,
 - 232 Referenz, 23, 31 Roboter, 208 Welt, 133, 231
- Koordinaten
 - extern, 49, 80, 133, 141, 145, 151, 152 Bild, 111
 - Index, 110
 - intern, 49,
 - 133 Gelenk, 80
- Coriolis, 68
- Zykluszeit, 159
- Zylinder
 - pneumatisch, 158
- Zylindrischer Roboter, 231

- D
- Gefahr, 153
- Freiheitsgrade: 5, 6, 210, 217
- Delta-Roboter, 69, 75
- Differentialantrieb, 192
- Verschiebung, 16

- Distanzbremsung,
 - 180 Einbruch,
 - 180 Schutztrennung, 180
 - Anhalten, 180
- Dynamische Bewegungsprimitive (DMP), 221, 223
 - Schätzung, 225
 - Modulation, 227
- Dynamisches
 - System linear, 222

- E
- Elektrischer Greifer, 167
- Encoder
 - magnetisch,
 - 92 optisch, 89

- Endeffektor, 133, 142, 149, 160
- Euler-Winkel, 39, 218
- Exoskelett, 2, 3
 - passiv, 220

- F
- Fehlervermeidung, 154
- Feeder, 160
 - Magazin, 163
 - Teil, 163
 - Vibration, 163
 - Vibrationsschüssel,
 - 163 Vibrations-Inline, 164
- Zuführgerät, 160

- Fingerfeder, 167
- Vorrichtung, 160

- Kraftkontakt, 148, 149, 152
 - Bodenreaktion, 211
 - begrenzend, 177,
 - 181 magnetisch,
 - 170 zulässig, 183
- Kraft- und Drehmomentsensor, 98
- Kraftsensor, 98
- Zwangs begriff, 223
- Vorwärtsprojektion, 108, 114

- G
- Mit
 - Gewalt greifen,
 - 167 mit der
 - Form, 167 zuverlässig, 167
- Schwerkraft, 68
- Greifer, 39, 41, 46, 165
 - elektrisch, 167
 - hydraulisch, 167
 - magnetisch, 169
 - mehrfingerig, 166
 - pneumatisch, 167
 - zweifingerig, 166
 - Vakuum, 168
- Führungsgerät, 178

- Punktschweißen mit Pistole,
 - 171 Schweißen, 171
- Gyroskop, 94

- H
- Handführung, 177, 178, 220
- Haptischer Roboter, 2

Erfassung menschlicher Bewegungen, 217
 Humanoider Roboter, 3, 209, 210
 Hydraulischer Greifer, 167

OH

Bildkoordinaten, 111
 Bildebene, 109
 Bildbearbeitung, 118
 Nachahmungslernen, 210, 216, 217
 Neigungsmesser, 94
 Indexkoordinaten, 110
 Trägheit, 68
 Trägheitsmesseinheit, 94, 96, 219
 Momentaner Drehpunkt, 193
 Schnittstellenfenster, 186

J

Gelenkdrehmomentsensor, 101

K

Kinematisches Paar, 27
 Kinematik, 49
 direkt, 49, 51, 73, 80, 214
 invers, 49, 51, 73, 80

L

Laser-Entfernungsmesser, 103
 Laserscanner, 104
 Bein, 70
 LiDAR, 105
 Endschalter, 98
 Lokalisierung, 189, 197
 induktive Führung, 197
 Laser, 197
 magnetischer Punkt,
 197 Magnetband, 197
 natürlich, 197
 Odometrie, 199
 funkbasiert, 199
 Draht, 197
 Fortbewegung
 auf zwei Beinen,
 210, 211 Doppelstützphase,
 213 Einzelstützphase, 213
 Standphase, 213
 Schwungphase, 213
 Gangmuster, 213

M

Maschine

Indizierung, 159

Magnet-
 Elektromagnet, 169
 Permanent, 170
 Magnetischer Encoder, 92
 Magnetgreifer, 169
 Magnetometer, 94

Homogene Matrixtransformation, 11, 13, 16,
 20, 27

Jacobian, 51, 55, 142–144, 148
 Rotation, 13, 39, 42

Mobiler Roboter, 1, 189, 191
 mobiler Gabelstapler,
 189 Schlepper,
 189 Lader, 189

Modell

dynamisch, 136, 138, 148, 176
 geometrisch, 19, 24
 kinematisch, 142

Überwachung von Geschwindigkeit und Abstand, 177

N

Nanoroboter, 1
 Navigation, 189, 197

Ö

Hindernisvermeidung, 205
 Odometrie, 199
 Omnidirektional, 190
 Operation
 kollaborativ, 173
 Betreiber, 174
 Optischer Encoder, 89
 absolut, 89
 inkremental, 91
 Orientierung, 6, 16, 39
 Überschwinger,
 240 Pose, 233

P

Palette, 158–160
 spritzgegossen, 161
 Metall, 161
 vakuumgeformt, 161
 Parallelroboter, 69, 73, 76, 80
 Teilhaber, 159

Pfadkontrolle, 202

Pfadverfolgung, 207

Wegplanung, 189, 201

- Perspektivische Matrixgleichung, 110
 Perspektivische Projektion, 108, 110
 Phase
 Evolution, 228
 variabel, 223, 228
 Plattform, 70
 Pneumatikzylinder, 158
 Pneumatischer Greifer, 167
 Punkt
 final, 126, 129
 initial, 126, 129
 via, 126, 128, 129
 Punkt-zu-Punkt, 123, 223, 226
 Pose, 6, 16
 Initiale, 23, 29, 31, 35
 Platz, 6, 16
 Potentiometer, 87
 Druck
 zulässig, 183
 Produktionslinie, 158
 asynchron, 159
 Hybrid, 158
 in Reihe, 158
 rotierend,
 158 synchron, 159
 Programmierung durch Demonstration, 210, 217
 Projektion
 rückwärts, 110, 113–115
 vorwärts, 108, 114
 Perspektive, 108, 110
 Näherungssensor, 102

 Pumpenvakuum, 169
 Venturi, 169

 Q
 Quadrocopter, 2
 Quaternion, 39, 44, 45

 R
 Entfernungssensor, 102
 Reduzierstück, 86

 Wiederholbarkeitsdistanz,
 239 Drift, 233, 240
 Orientierung, 238 Pose, 233, 235, 238 Position, 238
 Robo-Ethik, 3

 Anthropomorpher Roboter, 6, 231

 kartesisch, 231
 kollaborativ, 9, 157, 173, 175, 184
 zylindrisch, 231
 Delta, 69, 75
 Endeffektor, 160
 Umgebung, 153
 Haptik, 2
 Humanoid, 3, 209, 210
 Industrie, 173, 174, 178
 Mobil, 1, 189, 191
 Parallel, 1, 69, 73, 76, 80
 Rehabilitation, 2
 SCARA, 6, 22, 35, 231
 Segment, 27
 weich,
 9 sphärisch,
 231 System,
 173 Sichtsystem, 160
 Roboterarm, 4
 Roboterzelle, 6, 8, 153
 Roboterdynamik, 60, 67
 Robotergreifer, 4
 Roboterhand, 9
 Robotik, 1
 Humanoid, 209
 Roboter gelenk, 1,
 4 prismatisch,
 4 rotierend,
 4 rotatorisch,
 4 translatorisch, 4
 Roboter manipulator, 1, 4, 6, 7
 Robotersegment, 1, 4
 Roboterfahrzeug, 1
 Robotervision, 107
 Roboterhandgelenk, 4
 Rotation, 6, 12, 16
 Rotationsmatrix, 13, 39, 42
 RPY-Winkel, 39, 42, 134

 S
 Sicherheit, 153–155
 Sicherheitsrisiko, 174
 SCARA-Roboter, 6, 231
 Sensation
 Schmerz, 182
 Sensor, 2, 4, 85
 Stoßfänger,
 98 kapazitiv, 185
 Kontakt, 96
 elektrisch,
 85 elektromagnetisch,
 85 exterozeptiv, 85

- Kraft, [98](#), [185](#)
- Kraft und Drehmoment, [98](#),
[175](#) Neigungsmesser,
[94](#) Trägheitsmesseinheit, [94](#)
- Gelenkdrehmoment, [101](#),
[175](#) Laser-
Entfernungsmesser, [103](#) Laserscanner, [104](#)
- LiDAR, [105](#)
- Endschalter, [98](#)
- magnetischer Encoder,
[92](#) optisch,
[86](#) optischer Encoder,
[89](#) Potentiometer,
[87](#) propriozeptiv, [85](#)
- Näherungsschalter ,
[102](#)
- Entfernungsmessung ,
[102](#)
- sicherheitsbewertet,
[176](#) Tachometer, [93](#) taktil, [96](#), [175](#), [185](#) Ultraschall-Entfernungsmesser, [102](#)
- Sensorabstraktionsscheibe, [200](#)
- Gleichzeitige Lokalisierung und Kartierung
(SLAM), [200](#)
- Kugelroboter, [231](#)
- Federfinger, [167](#)
- Stabilisierungszeit, [233](#), [240](#)
- Normen, [153](#)
- Statik, [54](#)
- Stereovision, [115](#)
- Stewart-Gough-Plattform, [69](#), [74](#), [78](#), [82](#)

- Stopp-Notfall, [178](#), [184](#)
 schützend, [178–180](#)
 sicherheitsbewertet überwacht, [177–180](#), [182](#)
- Stützpolygon, [211](#), [214](#)

- T

- Tischzifferblatt,
 [159](#) drehbar, [161](#)
- Drehzahlmesser, [93](#)
- Taktiler Sensor, [96](#)
 kapazitiv, [97](#)
 verformungsbasiert, [97](#)
 magnetisch,
 [98](#) mechanisch,
 [98](#) optisch,
 [97](#) piezoelektrisch, [97](#)
- Telemanipulator, [2](#), [3](#), [9](#)

- Brennerschweißen, [171](#)
- Optische
 Verfolgung, [217](#)
- Flugbahn, [123](#), [132](#)
 Interpolation, [126](#), [129](#)
 Planung, [123](#)
- Übersetzung, [5](#), [12](#), [16](#)
- Trapezförmiges Geschwindigkeitsprofil, [123](#), [126](#)

- U
- Ultraschall-Entfernungsmesser, [102](#)
- Einradmodell, [194](#)

- V
- Variable
 rotatorisch, [29](#)
 translatorisch, [29](#)
- Virtuelle Umgebung, [2](#)
- Vision-
 Roboter, [107](#)
- Vision-System, [158](#)

- W

- Gehdynamik, [215](#)
 statisch, [215](#)
- Rad, [190](#)
 Lenkrolle,
 [190](#)
 Bockrad , [190](#)
 Kugelgelenk , [190](#) Standardlenkung, [190](#)
 Schwedisch, [190](#)
- Arbeitsbereich, [58](#)
- Arbeitsbereich, [6](#), [55](#), [153](#), [231](#)
 kollaborativ, [174](#), [186](#)
 geschickt, [59](#)
 maximal, [174](#) in
 Betrieb, [174](#)
 erreichbar, [59](#)
 eingeschränkt, [174](#)
- Handgelenksensor, [98](#)

- Z
- Zero-Moment-Point (ZMP), [210](#), [211](#), [214](#)