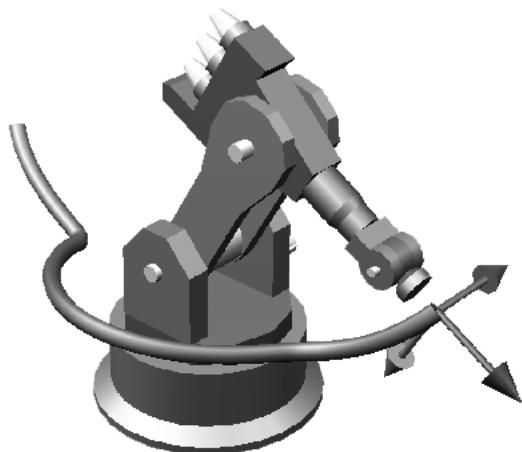


---

**Supplementary Lecture Notes  
for Chapters 1 - 5 and 6.3 of the Course**

---

# **Foundations of Robotics**



**Univ.-Prof. Dr. Oskar von Stryk  
Simulation, Systems Optimization and Robotics Group  
Technical University of Darmstadt**

---

**Winter Semester 2025/26 Edition**

---

**This is a work in progress version of the translation of the original German scriptum in English with first translations of Chapter 1-4. More updates will follow during the semester.**

Last updated on October 20, 2025

This version of the lecture notes has space for notes on the right margin.

*Robotics is the intelligent connection of perception to action.*  
(Michael Brady, 1985)

*You end up with a tremendous respect for a human being  
if you're a roboticist.*  
(Joseph Engelberger, 1985)

© 2025 Simulation, Systems Optimization and Robotics  
Group, Technical University of Darmstadt

All rights reserved, including the right to reproduce in whole or in part (photocopy, microcopy), to store in data processing systems, and to translate.

Please send typos and constructive suggestions of any kind directly to stryk@sim.tu-darmstadt.de.  
Thank you very much!

Further accompanying materials for the lecture are available online for registered students  
in the Moodle system used for the course.

The present version of the supplementary lecture notes emerged from the continuous re-  
vision and expansion of the manuscript for introductory lectures on robotics which are  
regularly taught by the author at Technical University of Darmstadt.

# Contents

<b>Note on Usage</b>	<b>vii</b>
<b>1 Introduction and Overview</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Foundational Courses in Robotics . . . . .	2
1.3 Literature Accompanying the Lecture on Foundations of Robotics . . . . .	3
1.4 Overview of Topics Covered in Foundations of Robotics . . . . .	5
1.4.1 Modeling of Position and Orientation (Chapter 2) . . . . .	5
1.4.2 Forward Kinematics (Chapter 3) . . . . .	6
1.4.3 Inverse Kinematics (Chapter 3) . . . . .	8
1.4.4 Velocities, Jacobian Matrix (and Static Forces) (Chapter 4) . . . . .	8
1.4.5 Robot Dynamics (Chapter 5) . . . . .	8
1.4.6 Robot Control (Chapter 6) . . . . .	9
1.4.7 Path and Trajectory Planning (Chapter 7) . . . . .	10
1.4.8 External Sensor Systems (Chapter 8) . . . . .	10
1.4.9 Localization and Navigation of Mobile Robots (Chapter 9) . . . . .	10
1.4.10 Middleware and Simulation (Chapter 10) . . . . .	10
1.4.11 Control of Autonomous Robots (Chapter 11) . . . . .	10
<b>2 Spatial Representations and Transformations</b>	<b>11</b>
2.1 Mathematical Quantities, Properties and Notations . . . . .	11
2.2 Coordinate Systems, Position and Orientation . . . . .	13
2.3 Classical Transformation Relationship . . . . .	17
2.4 Rotation of a Coordinate System $S_a \rightarrow S_b$ . . . . .	18
2.4.1 Elementary Rotation Matrices . . . . .	18
2.4.2 General Rotation Matrix ${}^aR_b$ . . . . .	19
2.4.3 Concatenated Elementary Rotations . . . . .	19
2.4.4 Orientation Representation with Three Angle Parameters . . . . .	22
2.5 Homogeneous Transformations . . . . .	24
2.6 Computational Efficiency . . . . .	26
<b>3 Robot Kinematics</b>	<b>28</b>
3.1 Forward Kinematics Model . . . . .	28
3.1.1 Kinematic Chains . . . . .	28
3.1.2 Kinematic Modeling with Frame Concept . . . . .	30
3.1.3 Denavit-Hartenberg (DH) Conventions . . . . .	31
1. Beispiel: SCARA-Manipulator . . . . .	36
2. Beispiel: 2-DOF-Schub-Drehgelenk-Arm . . . . .	38

3.2	Inverse (or Reverse) Kinematics Model . . . . .	40
3.2.1	General Properties of the IK . . . . .	40
3.2.2	Iterative, Numerical Calculation of IK . . . . .	43
3.2.2.1	General Procedure . . . . .	43
3.2.2.2	Selection of Some Iterative Calculation Methods for Inverse Kinematics . . . . .	44
	KDL . . . . .	44
	Drake . . . . .	44
	Corke's Robotics Toolbox . . . . .	44
3.2.3	Determination of an Explicit, Analytical Solution Formula for IK . . . . .	45
3.2.3.1	Algebraic Determination of IK . . . . .	46
	Beispiel: Ebener SCARA-Manipulator . . . . .	46
3.2.3.2	Geometric Determination of IK . . . . .	48
	Beispiel: Ebener SCARA-Manipulator . . . . .	48
3.2.3.3	Algorithmic Determination of the Analytical IK . . . . .	49
	Matlab Symbolic Math Toolbox . . . . .	49
	ikfast . . . . .	49
	Weitere . . . . .	49
3.3	Accuracy of the Kinematic Model . . . . .	50
3.4	Kinematic Modeling of Robot Legs and Arms . . . . .	51
3.4.1	Kinematics of a Three-Jointed Leg for a Quadruped Robot . . . . .	51
3.4.2	Kinematics of a Six-Jointed Leg for a Humanoid Robot . . . . .	52
3.4.3	Kinematics of an Arm for a Humanoid Robot . . . . .	54
<b>4</b>	<b>Velocity, Jacobian Matrix (and Static Forces)</b> . . . . .	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Skew-Symmetric Matrices, Vectors of Angular Velocities and Accelerations . . . . .	60
4.3	Jacobian Matrix of a Manipulator . . . . .	64
4.3.1	Addition of Angular Velocities . . . . .	64
4.3.2	Derivation of the Jacobian Matrix of a Manipulator . . . . .	66
	(i) Calculation of the Angular Velocity ${}^0\omega_n(t)$ . . . . .	67
	(ii) Calculation of the Linear Velocity ${}^0\mathbf{v}_n(t)$ . . . . .	69
	Summary of (i) and (ii) . . . . .	72
	Example: Planar 2-DOF SCARA Manipulator . . . . .	73
4.4	Inverse Jacobian Model . . . . .	75
4.4.1	Application for Velocity Control . . . . .	75
	Example: Motion of the 2-DOF SCARA Manipulator . . . . .	77
4.4.2	Kinematic Singularities . . . . .	79
	Example: Kinematic Singularities of the 2-DOF SCARA Manipulator . . . . .	80
4.5	Non-holonomic Kinematics of Multi-wheeled Vehicles . . . . .	85
4.5.1	Preliminary Remarks . . . . .	85
	Three Pose Degrees of Freedom of a (Point-like) Vehicle on a Plane . . . . .	86
4.5.2	Differential Drive Multi-wheeled Vehicles . . . . .	87
4.5.3	General Forward Kinematics Problem for Vehicles . . . . .	89
	Forward Kinematics of a Point-like Vehicle . . . . .	90
4.5.4	Inverse Kinematics Problem . . . . .	91
4.5.5	Omnidirectional Three-wheeled Kinematics . . . . .	92

4.5.6	Other Drive Types of Wheeled Vehicles . . . . .	93
4.5.7	Tracked Vehicles . . . . .	93
4.6	Static Forces in Manipulators . . . . .	95
<b>5</b>	<b>Roboterdynamik</b>	<b>96</b>
5.1	Einleitung . . . . .	96
5.2	Massenverteilung eines Starrkörpers . . . . .	99
5.3	Iterative Newton-Euler Formulierung der Manipulatordynamik . . . . .	103
5.3.1	Einleitung, Impuls- und Drallsatz . . . . .	103
5.3.2	Iterative Berechnung von INV DYN . . . . .	105
	Zusammenfassung der Newton-Euler Rekursionen . . . . .	113
5.3.3	Beispiel: Newton-Euler Formulierung der Dynamik des ebenen 2-DOF-SCARA-Manipulators . . . . .	115
	Weitere Bemerkungen zur rekursiven Newton-Euler Formulierung	123
5.4	Lagrangesche Formulierung der Roboterdynamik . . . . .	125
5.4.1	Kinetische und potentielle Energie, Lagrangefunktion . . . . .	125
5.4.2	Beispiel: Lagrangesche Formulierung der Dynamik des ebenen 2-DOF-SCARA-Manipulators . . . . .	128
5.5	Numerische Aspekte . . . . .	136
5.5.1	Berechnungseffizienz . . . . .	136
5.5.2	Modularität der Dynamikmodellierung . . . . .	136
5.5.3	Simulation dynamischer Roboterbewegungen . . . . .	137
5.6	Rekursive Verfahren zur Berechnung der Vorwärtsdynamik DIR DYN . .	138
5.6.1	Verfahren mit expliziter Berechnung der Massenmatrix $M$ . . . . .	138
5.6.1.1	Verfahren 1: Berechnung von $M$ durch wiederholte Auswertung des Newton-Euler-Verfahrens . . . . .	138
5.6.1.2	Verfahren 2: Ausnutzung der Symmetrie von $M$ . . . . .	139
5.6.1.3	Verfahren 3: Aggregation von Teilmanipulatoren (CRBA)	139
5.6.1.4	Vergleich der Verfahren bzgl. Berechnungsaufwand .	141
5.6.2	Berechnen der Vorwärtsdynamik ohne explizites Aufstellen der Massenmatrix . . . . .	141
5.6.3	Quelloffene Roboterdynamik-Bibliothek MBSlib . . . . .	142
5.7	Geschlossene kinematische Ketten . . . . .	142
5.8	Berücksichtigung von Nichtstarrkörpereffekten . . . . .	143
5.8.1	Reibung . . . . .	143
5.8.2	Elastizität . . . . .	145
5.8.2.1	Grundlagen zu elastischen Gelenken und Gliedern . .	146
5.8.2.2	Elastizitäten in der Robotik . . . . .	147
5.8.2.3	Berechnung des inversen Dynamikmodells bei (hohen) Drehgelenkelastizitäten . . . . .	151
5.8.2.4	Elastizitäten in der Biologie – Bewegungsdynamik des Menschen . . . . .	152
5.8.2.5	Bezug zur Steuerung und Regelung von Bewegungen bei Mensch und Tier (siehe 6.12) . . . . .	152
5.9	Spezielle Dynamikmodelle für zweibeinige Roboter und deren Stabilitätsregelung . . . . .	153
5.9.1	Zero-Moment-Point (ZMP) . . . . .	154
5.9.2	Invertiertes Pendel . . . . .	156

5.9.3	Erweiterte Modelle des menschlichen Gehens und Rennens . . . . .	159
5.9.3.1	Feder-Masse-Modell für menschliches Rennen . . . . .	159
5.9.3.2	Feder-Masse-Modell für menschliches Gehen . . . . .	160
5.9.4	Globale Stabilitätsbegriffe . . . . .	162
5.9.5	Unterschiedliche Ansätze zu zweibeinigem (Roboter-)Gehen . . . . .	163
5.9.6	Einige Referenzen zu Kapitel 5.9 . . . . .	163
<b>6</b>	<b>Roboterregelungen</b>	<b>164</b>
6.1	Antriebssysteme . . . . .	164
6.2	Interne Sensoren . . . . .	164
6.3	Lineare (Positions-)Regelung . . . . .	164
6.3.1	Einführung und Vorbemerkungen . . . . .	164
6.3.2	Lineare Systemdynamik . . . . .	167
6.3.3	PD-Regelung linearer Systeme zweiter Ordnung . . . . .	169
6.3.4	Partitionierung des Regelgesetzes durch Feedback-Linearisierung . . . . .	171
6.3.5	Sollwerttrajektorien-Folgeregelung . . . . .	172
6.3.6	PID-Regelung linearer Systeme (mit Anti-Windup) . . . . .	173
—	<i>Ab hier zur Information die weitere Gliederung der Vorlesung und ab hier stellen die ausgegebenen Folien das Skriptum dar —</i>	174
6.3.7	Kaskadenregelung . . . . .	174
6.3.8	Stabilität als Sprungantwortverhalten und Wahl der PID-Reglerparameter	174
6.4	Nichtlineare Regelung . . . . .	175
6.4.1	Systemlinearisierung . . . . .	175
6.4.2	Modellbasierte Manipulatorregelung (Feedback Linearization) . .	175
6.4.3	Adaptive Manipulatorregelung . . . . .	175
6.4.4	Bahnregelung in Weltkoordination (Kartesische Bahnregelung) . .	175
6.5	Kraft-/Momenten-Regelung . . . . .	175
6.6	Hybride Bahn-/Kraft-Regelung . . . . .	175
6.7	Nachgiebigkeitsregelungen (Compliant Control) . . . . .	175
6.7.1	Impedanzregelung . . . . .	175
6.7.2	Admittanzregelung . . . . .	175
6.7.3	Aktiv-passive Konzepte für Impedanz-/Admittanzregelungen . .	175
6.8	Bildgeführte Regelung . . . . .	175
6.9	Multimodale Regelung physikalischer Interaktion . . . . .	175
6.10	Roboterhände und Greifen . . . . .	175
6.11	Elementare Roboterbewegungen . . . . .	175
6.12	Regelung und Steuerung von Bewegungen bei Mensch und Tier . .	175
—	<i>From here on, the further structure of the lecture for information and from here on, the printed slides represent the script —</i>	164
<b>7</b>	<b>Bahnplanung</b>	<b>176</b>
7.1	Einleitung und Übersicht . . . . .	176
7.2	Topologische Wegeplanung . . . . .	176
7.3	Konfigurationsraum . . . . .	176
7.4	Geometrische Bahnplanung . . . . .	176
7.4.1	Roadmap-Verfahren . . . . .	176
7.4.2	Exakte Zellzerlegung . . . . .	176

7.4.3	Approximative Zellzerlegung . . . . .	176
7.4.4	Potentialfeldmethoden . . . . .	176
7.4.5	Neuere Verfahren: Stichprobenverfahren, RRTs . . . . .	176
7.4.6	Komplexität der geometrischen Bahnplanung . . . . .	176
7.4.7	Anwendungsbeispiele . . . . .	176
7.5	Kinematische und dynamische Trajektorienplanung . . . . .	176
7.6	Greifplanung . . . . .	176
<b>8</b>	<b>Externe Sensorsysteme</b>	<b>177</b>
8.1	Nichtvisuelle externe Sensoren . . . . .	177
8.2	Visuelle Sensoren und Algorithmen . . . . .	177
8.2.1	Einleitung . . . . .	177
8.2.2	Bilderzeugung . . . . .	177
8.2.3	Grundlagen der Bildverarbeitung . . . . .	177
8.2.4	Bildvorverarbeitung . . . . .	177
8.2.5	Bildverarbeitung . . . . .	177
8.2.6	Merkalsextraktion . . . . .	177
8.2.7	Objektklassifikation . . . . .	177
8.3	3D-Sensoren, -Perzeption und Umgebungsrepräsentation . . . . .	177
<b>9</b>	<b>Navigation mobiler Roboter</b>	<b>178</b>
9.1	Lokalisierung und Positionierung . . . . .	178
9.1.1	Relative Positionsbestimmung . . . . .	178
9.1.2	Absolute Positionsbestimmung . . . . .	178
9.1.3	Stochastische Positionsbestimmung . . . . .	178
9.2	Selbstlokalisierung und Navigation . . . . .	178
9.2.1	Einleitung und Übersicht . . . . .	178
9.2.2	Lokalisierung mit einer Hypothese (Kalman-Filter) . . . . .	178
9.2.3	Lokalisierung mit mehreren Hypothesen . . . . .	178
9.2.3.1	Verwendung mehrerer Kalman-Filter . . . . .	178
9.2.3.2	Diskretisierte Wahrscheinlichkeitsverteilung . . . . .	178
9.2.3.3	Monte-Carlo Lokalisierung . . . . .	178
9.2.4	SLAM (Simultaneous Localization and Mapping) . . . . .	178
<b>10</b>	<b>Middleware und Simulation</b>	<b>179</b>
10.1	Middleware für Robotersysteme . . . . .	179
10.2	Simulation autonomer Roboter . . . . .	179
10.3	Robot Operating System (ROS) . . . . .	179
<b>11</b>	<b>Steuerung autonomer Roboter</b>	<b>180</b>
11.1	Steuerungsarchitekturen . . . . .	180
11.2	Programmierung von autonomem Verhalten . . . . .	180
11.3	Roboterverhalten durch generative KI . . . . .	180
<b>12</b>	<b>Herausforderung und Verantwortung der Robotik</b>	<b>181</b>

---

<b>Anhang</b>	<b>182</b>
A.1 Orientierungsrepräsentation mit Quaternionen . . . . .	183
A.2 Zusammenhang zwischen Rotationsmatrix und Drehvektor und Drehwinkel	186
A.2.1 Berechnung der Rotationsmatrix aus Drehvektor und Drehwinkel	186
A.2.2 Berechnung von Drehvektor und Drehwinkel aus der Rotationsmatrix	186
A.3 Vergleich einiger Notationen im Skript mit anderen Lehrbüchern . . . . .	189
A.4 Typische Notationen und Symbole in Robotik 1 . . . . .	190
A.4.1 Griechisches Alphabet . . . . .	190
A.4.2 Typische Symbole und ihre Bedeutung . . . . .	190
<b>Index</b>	<b>192</b>

## Note on Usage

This supplementary lecture notes for the course Foundations of Robotics contains

- the detailed table of contents of all planned main chapters 1 to 12 of the lecture,
- a supplementary elaboration for the presentations and notes made in the lecture in chapters 1 to 6 (beginning)
- as well as an index of key terms.

The lecture notes for chapters 7-12 are represented by the accompanying lecture slides.

Digital copies of all notes made in the lecture as well as all slides used are made available to all enrolled students in the corresponding Moodle course.



# Chapter 1

## Introduction and Overview

### 1.1 Introduction

**What is a robot?** The term robot is typically associated in the general public with either an industrial robot or a humanoid robot (e.g., the Honda Asimo<sup>1</sup> or the Terminator from Hollywood films).

However, modern robotic systems are also being investigated for a growing variety of new applications, for example

- as robotic assistance systems in production (“third hand”)<sup>2</sup>,
- as autonomous or semi-autonomous search and rescue robotic systems for use in disasters<sup>3</sup>,
- as household robots, autonomous vacuum cleaners<sup>4</sup> or autonomous window cleaners<sup>5</sup> or
- as robotic systems in modern automobiles, which can monitor or automate parts of human control tasks<sup>6</sup> or
- as intelligent leg prostheses<sup>7</sup> that can autonomously adapt to patients, gaits, and terrain,

and many other possibilities. For the latter two, less obvious applications of robotics, one also speaks of **hidden robotics** or **embedded robotics**.

Modern robotic systems enable (semi-)autonomous interactions with the environment, which is perceived through **sensors**. Based on the sensor data interpretation, decisions and actions are determined algorithmically and implemented via **actuators** (drives).

---

<sup>1</sup>[world.honda.com/ASIMO/](http://world.honda.com/ASIMO/)

<sup>2</sup>e.g., Franka Emika [www.franka.de](http://www.franka.de) and Baxter [www.rethinkrobotics.com](http://www.rethinkrobotics.com)

<sup>3</sup>e.g., Team Hector Darmstadt [www.teamhector.de](http://www.teamhector.de)

<sup>4</sup>[www.irobot.com/de](http://www.irobot.com/de)

<sup>5</sup>[www.euron.org/resources/robotgallery.html](http://www.euron.org/resources/robotgallery.html)

<sup>6</sup>e.g., the vehicles in the DARPA Urban Challenge 2007: [en.wikipedia.org/wiki/DARPA\\_Grand\\_Challenge](http://en.wikipedia.org/wiki/DARPA_Grand_Challenge) or their autonomous predecessor vehicles, e.g., VaMP: [de.wikipedia.org/wiki/VaMP](http://de.wikipedia.org/wiki/VaMP), [www.idsia.ch/~juergen/robotcars.html](http://www.idsia.ch/~juergen/robotcars.html)

<sup>7</sup>e.g., Spring Active [www.springactive.com](http://www.springactive.com) and Research Training Group LokoAssist at TU Darmstadt [www.tu-darmstadt.de/lokoassist](http://www.tu-darmstadt.de/lokoassist)

From the perspective of electrical engineering, mechanics, and mechanical engineering, mechatronic systems are becoming increasingly “intelligent” through growing mobile computing power and efficient algorithms. From the computer science perspective (figuratively speaking), “the PC of the future will have eyes and ears (or external sensors) to perceive the environment, hands (or drives) for physical interaction, and legs for locomotion (mobility)”. By connecting a **mobile PC** (consisting of a fast processor, large memory capacity, and powerful algorithms) with sensors for perceiving the environment and actuators for the physical execution of actions, autonomous (so-called **autonomous**) actions with a certain “intelligence” are possible. The term “intelligence” is to be understood here phenomenologically in the sense of observing or demonstrating properties of natural, intelligent systems in robots, such as

- the adaptability to a new or changing environment,
- the ability to independently solve complex problems,
- learning ability,
- but also social and emotional intelligence.

Only quite late in the evolutionary development of natural intelligence does the development of the ability of logical reasoning emerge. Therefore, this will not be at the beginning in the development of intelligent robots either.

Many augurs of technical progress predict that, after the last 50 years were determined by the advancement of “computers” into all areas, the next 50 years will be decisively shaped by the advancement of robots and robotic systems into all areas of public and private life.

Many subfields of computer science play a crucial role when a mere assembly of motors, sensors, cables, and mobile computers is to become an “intelligent”, (semi-)autonomously acting robotic system.

## 1.2 Foundational Courses in Robotics

**What is robotics?** Robotics is a highly multidisciplinary field. For the design and operation of a high-performance robotic system, questions of hardware and software must be solved equally, which, from a technical perspective, concern areas of mechanics, electrical engineering, and computer science. If one additionally wants to develop completely new robotic systems with fundamentally new capabilities, many basic research questions need to be solved (e.g., why can humans and animals move so efficiently and so capably, and acquire new skills at many different levels (from learning efficient movements to learning complex relationships) so well?). McKerrow gives a very good definition of robotics in his textbook from 1986:

„Robotics is the discipline which involves:

- (a) the design, manufacture, control, and programming of robots;
- (b) the use of robots to solve problems;
- (c) the study of the control processes, sensors, and algorithms used in humans, animals, and machines; and

(d) the application of these control processes and algorithms to the design of robots.“

The Simulation, Systems Optimization and Robotics (SIM) Group at the Department of Computer Science at the Technical University of Darmstadt offers **Foundations of Robotics** with 6 semester hours per week (SWS) as well as **Robotics Labs** and **Robotics Seminars**, which are typically integrated into a two-semester **Robotics Project**.

The **goal** of the foundational course in robotics is to familiarize participants with the scientific foundations of robotics. The tasks to be solved in modern robotics are highly complex, multidisciplinary, difficult, and challenging. Only those who have a good “general education” in robotics will be able to develop good solutions that are competitive, and implement completely new ideas and approaches into high-performance robots!

The contents of the individual robotics courses are:

#### **Lecture Foundations of Robotics (6 SWS:V4+Ü2) :**

The course teaches the mathematical foundational knowledge and basic skills necessary for serious engineering investigations in robotics, including: Spatial representation and transformations, modeling of position and orientation, manipulator kinematics, vehicle kinematics, kinematic velocity, Jacobian matrix, robot dynamics, robot controls, localization of mobile robots, path planning. Theoretical and practical exercises as well as programming tasks serve to deepen the course content.

**Integrated Robotics Project (8 SWS, 2 semesters):** The integrated project combines elements of seminar, project lab, and team project work, and offers, among other things, a deeper foundation compared to a lab. It deals, for example, with the development and implementation of methods and algorithms for (semi-)autonomous mobile search and rescue robots (Team Hector), and their evaluation at national and international competitions, wearable robotic systems (as exoskeletons, orthoses, or prostheses), as well as biomechanically inspired robots.

Successful participation in the foundational courses in robotics is a necessary prerequisite for a subsequent bachelor’s or master’s thesis in the field of robotics.

## **1.3 Literature Accompanying the Lecture on Foundations of Robotics**

The best and currently most comprehensive overview of robotics is given by

**[Siciliano/Khatib, 2008]** B. Siciliano, O. Khatib: Springer Handbook of Robotics (Springer Verlag, 2008).

However, this is not a textbook but rather a reference and overview work with more than 1500 pages.

Accompanying the lecture *Foundations of Robotics*, the following textbooks are recommended, on which parts of the lecture are based.

For **Chapters 1 to 6 of the lecture**, the following are recommended:

**[Craig, 2004]** J.J. Craig: Introduction to Robotics – Mechanics and Control (3rd edition, Pearson Education Inc., 2005).

This “classic” book is one of the most widely used textbooks on robotics.

**[Spong/Vidyasagar, 1989]** M.W. Spong, M. Vidyasagar: Robot Dynamics and Control (John Wiley & Sons, 1989).

The revised and expanded new edition was published as:

**[Spong/Hutchinson/Vidyasagar, 2005]** M.W. Spong, S. Hutchinson, M. Vidyasagar: Robot Modeling and Control (J. Wiley & Sons, 2005).

This book has a somewhat more current and comprehensive presentation of the foundations than Craig’s book. Additionally, it contains foundations of image processing in robotics and image-based robot control.

As supplementary or advanced literature, the following are recommended:

**[Kajita, 2007]** S. Kajita (Ed.): Humanoid Roboter - Theorie und Technik des künstlichen Menschen (Akademische Verlagsgesellschaft Aka GmbH, Berlin, 2007).

**[Niku, 2001]** S.B. Niku: Introduction to Robotics, Analysis, Systems, Applications (Prentice Hall, 2001).

**[Dudek/Jenkin, 2000]** G. Dudek, M. Jenkin, M.: Computational Principles of Mobile Robotics (Cambridge University Press, 2000).

**[Fu/Gonzalez/Lee, 1987]** K.-S. Fu, R.C. Gonzalez, C.S.G. Lee: Robotics: Control, Sensing, Vision and Intelligence (New York: McGraw-Hill, 1987).

**[McKerrow, 1991]** P.J. McKerrow: Introduction to Robotics (Addison-Wesley, 1991).

**[Niku, 2001]** S.B. Niku: Introduction to Robotics - Analysis, Systems, Applications (Prentice Hall, 2001).

**[Schmidt, 1999]** G. Schmidt: Grundlagen intelligenter Roboter (Skriptum, Lehrstuhl für Steuerungs- und Regelungstechnik, TU München, 1999).

**[Yoshikawa, 1990]** T. Yoshikawa: Foundations of Robotics: Analysis and Control (MIT Press, 1990).

**[Wong, 2001]** J. Y. Wong: Theory of Ground Vehicles (Wiley, 3rd edition, 2001).

As supplementary or advanced literature for **Chapter 7 Path Planning**, the following are recommended:

**[Choset et al., 2005]** H. Choset, K.M. Lynch, S. Hutchinson, G.A. Kantor, W. Burgard, L.E. Kavraki, S. Thrun: Principles of Robot Motion: Theory, Algorithms, and Implementations (A Bradford Book, 2005).

**[Latombe, 1991]** J.-C. Latombe: Robot Motion Planning (Kluwer Academic, 1991).

As supplementary or advanced literature for **Chapter 8 External Sensors** and **Chapter 9 Navigation and Localization**, the following are recommended:

**[Siegwart, Nourbakhsh, Scaramuzza, 2011]** R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza: Introduction to Autonomous Mobile Robots (MIT Press, 2nd revised edition, 2011).

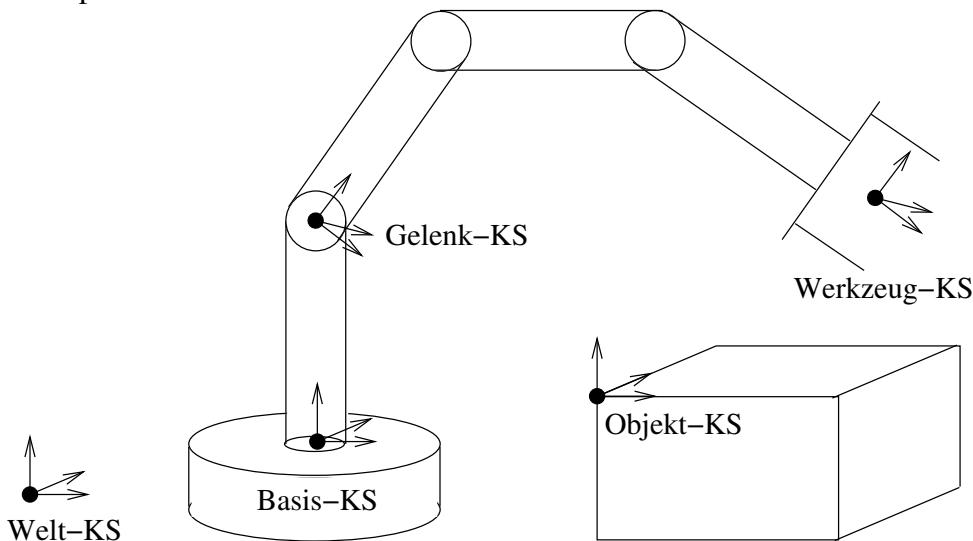
**[Thrun, Burgard, Fox, 2005]** S. Thrun, W. Burgard, D. Fox: Probabilistic Robotics (MIT Press, 2005).

## 1.4 Overview of Topics Covered in Foundations of Robotics

This concluding section of the first chapter briefly introduces the individual topics covered in Foundations of Robotics.

### 1.4.1 Modeling of Position and Orientation (Chapter 2)

- A constantly occurring fundamental problem is the localization (position determination) of objects in 3D space: e.g., the joints of the robot, the workpieces, or other objects (obstacles in space), etc.
- The description is done through two attributes: position and orientation.
- Consequently, methods are needed for representing and manipulating the position and orientation of objects.
- Solution approach: A fixed coordinate system (abbr.: CS, Engl.: frame) is associated with each object. The description of position and orientation of this local CS is done relative to a reference CS.
- Example:



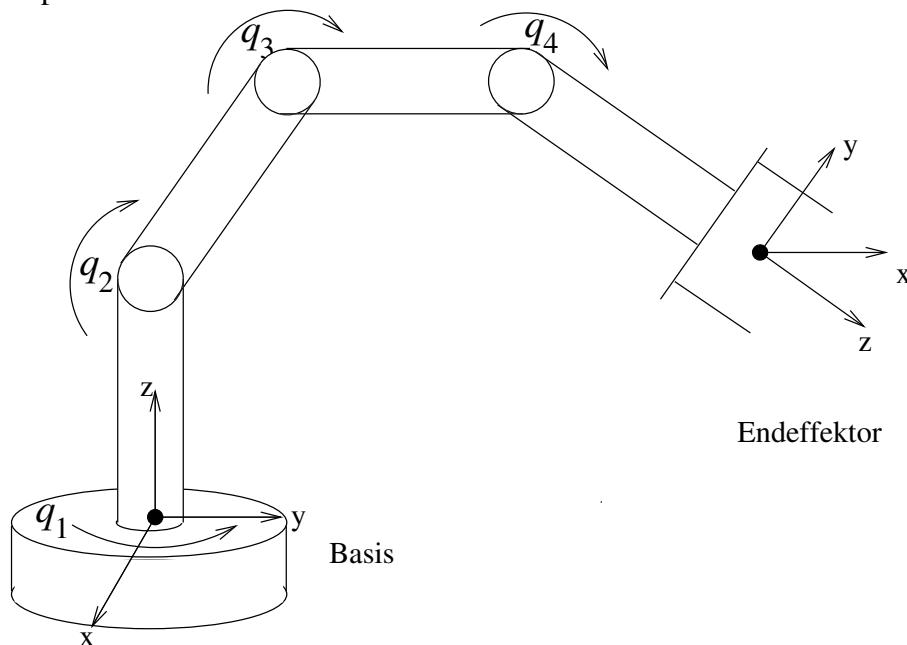
- In principle, any CS can serve as a reference CS: This leads to a transformation of the description of position and orientation when switching from one CS to another.

### 1.4.2 Forward Kinematics (Chapter 3)

- **Kinematics** is the study of motion (i.e., position, velocity, acceleration) *without* consideration of the causing forces.
- For this purpose, a robot manipulator is idealized and considered as a **kinematic chain**, i.e., a concatenation of adjacent rigid<sup>8</sup> links (bars) by means of rigid (revolute or prismatic) joints.
- In the joints, there are generally position sensors that measure the relative position of adjacent joints:
  - for revolute joints, the relative joint angle,
  - for prismatic joints, the relative displacement

between adjacent joints.

- Example:



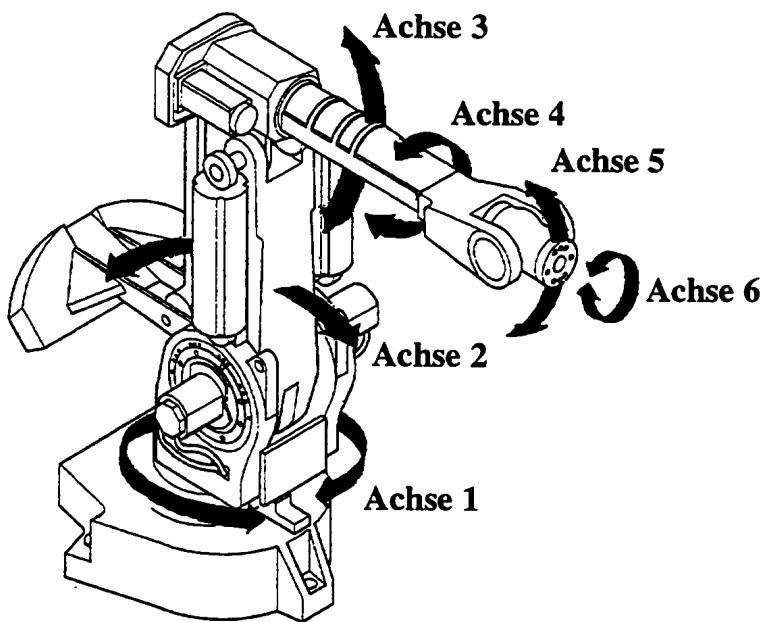
- The **forward kinematics problem** consists of calculating the position and orientation of the end effector relative to the base CS from given/measured values for  $q_1, q_2, \dots, q_n$ .
- Notation:  
 $x, z, y$ : Cartesian coordinates,     $q_1, q_2, \dots, q_n$ : joint coordinates.
- **Remark:** If there are joints that can be neither directly nor indirectly moved in a controlled manner by one of the robot drives, e.g., a pure hinge joint, then one speaks of an **underactuated** robot.
- The **configuration** (configuration) of a robot manipulator determines a complete specification of the position of every point on the manipulator. A robot manipulator

---

<sup>8</sup>no elasticity, perfect positioning

has  $n$  **degrees of freedom** (Engl.: degrees of freedom, abbr.: DOF)<sup>9</sup>, if its configuration space can be described by  $n$  minimal parameters. For a robot, the number of independent joints determines the number of DOF.

- Example of an industrial robot with 6 revolute joints (150 kg payload at 3 m reach and 2 t own weight):



<sup>9</sup>For a general explanation of degrees of freedom in kinematics, see e.g., [www.learnengineering.org/2013/04/degrees-of-freedom-mechanics.html](http://www.learnengineering.org/2013/04/degrees-of-freedom-mechanics.html).

### 1.4.3 Inverse Kinematics (Chapter 3)

- Given: Position and orientation of the end effector.  
Sought: All possible joint angle configurations that lead to this.
- The *inverse kinematics problem* is more difficult than the forward kinematics problem:
  - nonlinear equations, whose solution is often only possible numerically,
  - there can be *multiple* solutions,
  - there can be *no* solutions (e.g., if not in the workspace of the robot).

### 1.4.4 Velocities, Jacobian Matrix (and Static Forces) (Chapter 4)

- Motion is characteristic of robots!
- The **Jacobian matrix of a manipulator**  $J(\mathbf{q})$ 
  - is **not** to be confused with the Jacobian matrix of a vector-valued function from the basic mathematics course!
  - is an important quantity for investigating the velocity of robots and mechanisms;
  - describes a mapping (abbr.: map) from velocities in joint coordinates to velocities in Cartesian coordinates at the end effector;
  - kinematic singularity:** For certain values of the joint angles  $\mathbf{q}$ ,  $J(\mathbf{q})$  is *singular*, i.e., the mapping is not invertible.  
⇒ A local loss of degrees of freedom of motion is the consequence.  
Kinematic singularities must therefore be avoided in motion planning.
- $J(\mathbf{q})$  is also important in the calculation of contact forces, e.g., between robot hand and object or robot foot and ground.
- The velocities considered here are still part of the field of kinematics.

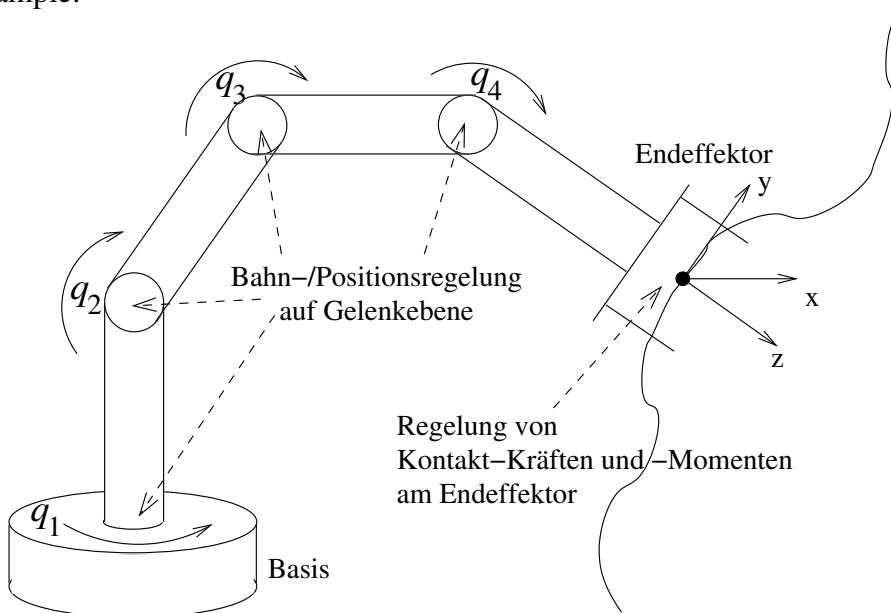
### 1.4.5 Robot Dynamics (Chapter 5)

- Kinetics** is the extension of kinematics by investigating the *forces* and *moments* that are needed for a motion or that occur during it.
- For a specific motion profile (accelerations from rest, motion with prescribed velocity of the end effector, braking to a halt), appropriate drive torques in the joints must be generated by the drives.
- The exact values of the required drive torques depend on
  - the spatial and temporal properties of the motion path,
  - the masses and inertias of the joints, the payload,
  - the friction forces in the joints, etc.

- The modeling and simulation of robot dynamics is needed, among other things,
  - for *precise control* of the robot along a desired motion path,
  - for *realistic simulation* of the motion behavior of the robot.

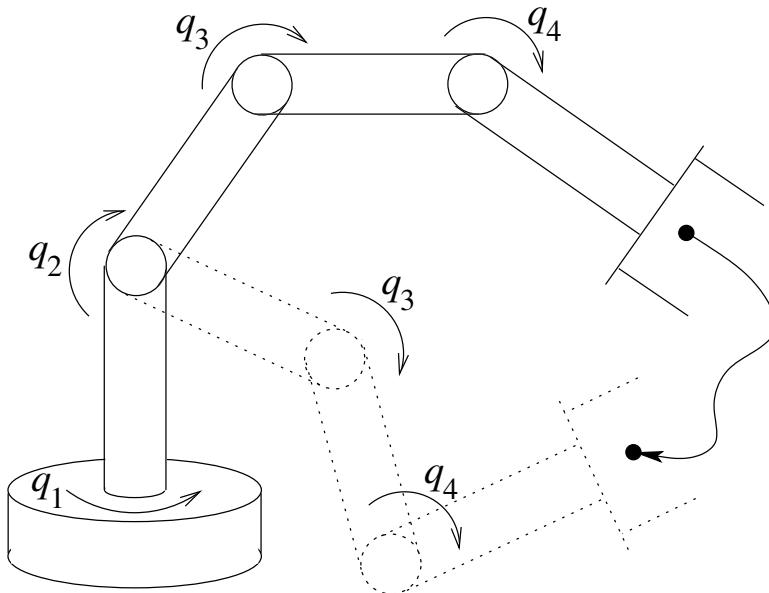
### 1.4.6 Robot Control (Chapter 6)

- Joint drives can only apply a force or a torque in the joint. An algorithm is needed to calculate the torques in the joints that result in a desired motion and that can also compensate for disturbances and model uncertainties.
- The ***path or position control*** at the joint level automatically compensates for (unavoidable) deviations from a desired path using position and velocity sensors in the joints.
- The ***force/torque control*** controls the (three-dimensional) contact forces and torques, e.g., when grasping objects or during ground contact of mobile, walking robots.
- Example:



### 1.4.7 Path and Trajectory Planning (Chapter 7)

- Generation of a “smooth”<sup>10</sup> (Engl. smooth) motion from an initial to a final position for a manipulator (or a vehicle) through “soft”, simultaneous and coordinated motion of the individual joints.
- Example:



- For example, description of the path in Cartesian world coordinates by waypoints (so-called *via points*), which must be traversed either exactly or within a certain tolerance, and an interpolation rule (e.g., straight lines, circular arcs, splines, etc.). The Cartesian path must then be converted into corresponding motions of the individual joints.

### 1.4.8 External Sensor Systems (Chapter 8)

Overview of various internal and external sensors of robots

### 1.4.9 Localization and Navigation of Mobile Robots (Chapter 9)

The autonomous determination and implementation of a collision-free path from a start to a goal point is a basic requirement for mobile robots. Here, important techniques for sensor data fusion and localization, such as Kalman filters and stochastic localization methods, are discussed.

### 1.4.10 Middleware and Simulation (Chapter 10)

### 1.4.11 Control of Autonomous Robots (Chapter 11)

Control architectures are responsible for ensuring that a complex task is accomplished by the robot autonomously or semi-autonomously. There are different approaches.

---

<sup>10</sup>“uniform, non-jerky”

# Chapter 2

## Spatial Representations and Transformations

### 2.1 Mathematical Quantities, Properties and Notations

In this section, some elementary notations and relationships from linear algebra are reviewed, insofar as they are needed for the lecture Foundations of Robotics.

#### Vectors

Vector  $\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \in \mathbb{R}^3 = \mathbb{R}^{3 \times 1}$  (column vector),  $\mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  is the zero vector

- Transposed vector  $\mathbf{p}^T = (p_1 \ p_2 \ p_3) \in \mathbb{R}^{1 \times 3}$  (row vector)
- (Euclidean) norm of a vector:  $\|\mathbf{p}\| = \sqrt{p_1^2 + p_2^2 + p_3^2}$
- Scalar product of two vectors  $\mathbf{p}, \mathbf{r} \in \mathbb{R}^3$ :  $\langle \mathbf{p}, \mathbf{r} \rangle = \mathbf{p}^T \mathbf{r} = p_1 r_1 + p_2 r_2 + p_3 r_3$
- Orthogonal vectors  $\mathbf{p} \perp \mathbf{r}$ :  $\langle \mathbf{p}, \mathbf{r} \rangle = \mathbf{p}^T \mathbf{r} = 0$
- Angle  $\phi$  between two vectors  $\mathbf{p}, \mathbf{r}$ :  $\cos \phi = (\mathbf{p}^T \mathbf{r}) / (\|\mathbf{p}\| \|\mathbf{r}\|)$
- The cross product (or vector product) of two vectors  $\mathbf{p}, \mathbf{r}$ , which are not parallel,

$$\mathbf{p} \times \mathbf{r} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \times \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} p_y r_z - p_z r_y \\ p_z r_x - p_x r_z \\ p_x r_y - p_y r_x \end{pmatrix} =: \mathbf{s}$$

yields a vector  $\mathbf{s}$  that is orthogonal to both  $\mathbf{p}$  and  $\mathbf{r}$ , such that  $\mathbf{p}, \mathbf{r}, \mathbf{s}$  in this order form a right-handed system.

Further properties of the cross product are (with scalar  $\lambda \in \mathbb{R}$ ):

$$\mathbf{p} \times \mathbf{r} = -\mathbf{r} \times \mathbf{p}, \quad \mathbf{p} \times (\mathbf{r} + \mathbf{s}) = \mathbf{p} \times \mathbf{r} + \mathbf{p} \times \mathbf{s}, \quad \lambda \cdot (\mathbf{p} \times \mathbf{r}) = (\lambda \cdot \mathbf{p}) \times \mathbf{r} = \mathbf{p} \times (\lambda \cdot \mathbf{r})$$

## Matrices

$$(3 \times 3)\text{-matrix } R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \in \mathbb{R}^{3 \times 3}$$

- **Transpose matrix**  $R^T = \begin{pmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{pmatrix} \in \mathbb{R}^{3 \times 3}$  (interchange of rows and columns)
- **Determinant** of a  $(3 \times 3)$ -matrix:

$$\begin{aligned} \det(R) &= |R| = \begin{vmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{vmatrix} \\ &= \text{e.g., by formal expansion along the 1st row} \\ &= r_{11} \begin{vmatrix} r_{22} & r_{23} \\ r_{32} & r_{33} \end{vmatrix} - r_{12} \begin{vmatrix} r_{21} & r_{23} \\ r_{31} & r_{33} \end{vmatrix} + r_{13} \begin{vmatrix} r_{21} & r_{22} \\ r_{31} & r_{32} \end{vmatrix} \\ &= \text{continuing with determinant rule for } (2 \times 2)\text{-matrices} \\ &= r_{11}(r_{22}r_{33} - r_{32}r_{23}) - r_{12}(r_{21}r_{33} - r_{31}r_{23}) + r_{13}(r_{21}r_{32} - r_{31}r_{22}) \\ &= r_{11}r_{22}r_{33} + r_{12}r_{23}r_{31} + r_{13}r_{21}r_{32} \\ &\quad - r_{13}r_{22}r_{31} - r_{11}r_{23}r_{32} - r_{12}r_{21}r_{33} \end{aligned}$$

The latter formula can also be used directly for calculation using Sarrus's rule. (Caution: Sarrus's rule only applies to  $(3 \times 3)$ -matrices!)

- Identity matrix  $E \in \mathbb{R}^{3 \times 3}$  (often  $I$  in English)  $E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Occasionally one also uses  $E_n$  for the identity matrix in  $\mathbb{R}^{n \times n}$ .

- Zero matrix  $O \in \mathbb{R}^{3 \times 3}$ :  $O = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

- **Inverse** matrix  $R^{-1}$ :  $RR^{-1} = E = R^{-1}R$

- For general  $(n \times n)$ -matrices, one normally computes  $R^{-1}$  numerically for “large”  $n$ .
- For  $(3 \times 3)$ -matrices, there is an explicit formula.
- In practice, one actually never needs  $R^{-1}$  explicitly. In context, an explicit calculation is almost never necessary and would even be inefficient.

Example: Solution of a general linear system of equations:

given:  $A \in \mathbb{R}^{n \times n}$ ,  $\mathbf{b} \in \mathbb{R}^n$

sought:  $\mathbf{x} \in \mathbb{R}^n$  with  $A\mathbf{x} = \mathbf{b}$

numerical solution approach:

- 1.) Numerical triangular decomposition of  $A$  (e.g., with Gaussian elimination):

$A = LR$  with  $L, R \in \mathbb{R}^{n \times n}$

$$\text{and } L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ * & 1 & 0 & \dots & 0 \\ * & * & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & \dots & 1 \\ * & * & * & \dots & * \\ 0 & * & * & \dots & * \\ 0 & 0 & * & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & * \end{pmatrix} \text{ (left, lower triangular matrix)}$$

$$\text{and } R = \begin{pmatrix} 0 & 0 & * & \dots & * \\ 0 & 0 & * & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & * \end{pmatrix} \text{ (right, upper triangular matrix).}$$

Thus  $Ax = L \cdot \underbrace{Rx}_{=y \in \mathbb{R}^n} = b$

2.) Solution of  $Ly = b$  with forward substitution

3.) Solution of  $Rx = y$  with backward substitution

- For **orthonormal**<sup>1</sup> (i.e., orthogonal and normalized) matrices, it holds that  $RR^T = E$  (i.e.,  $R^{-1} = R^T$ , the transposed matrix is the inverse matrix)

furthermore, for orthonormal matrices, it holds that:

$$1 = \det(E) = \det(RR^T) = \det(R)\det(R^T) = (\det(R))^2$$

$$\Rightarrow \det(R) = \begin{cases} +1, & \text{right-handed system} \\ -1, & \text{left-handed system} \end{cases} \text{ of column vectors}$$

and the resulting mapping is  $\begin{cases} \text{rotation} \\ \text{reflection.} \end{cases}$

Simplest example:

Identity matrix  $E$  is orthonormal:  $EE^T = E \Rightarrow E^T = E^{-1}$ .

**$R$  regular** ( $n \times n$ )-matrix

- $\Leftrightarrow \det(R) \neq 0$
- $\Leftrightarrow$  Inverse matrix  $R^{-1}$  exists.
- $\Leftrightarrow$  The linear system of equations  $Rx = 0$  with unknown  $x \in \mathbb{R}^n$  has exactly one solution:  $x = \mathbf{0} \in \mathbb{R}^n$ .
- $\Leftrightarrow$  The column vectors of  $R$  are linearly independent.
- $\Leftrightarrow$  The row vectors of  $R$  are linearly independent.

**$R$  singular** ( $n \times n$ )-matrix

- $\Leftrightarrow \det(R) = 0$
- $\Leftrightarrow$  Inverse matrix  $R^{-1}$  does not exist.
- $\Leftrightarrow$  The linear system of equations  $Rx = 0$  with unknown  $x \in \mathbb{R}^n$  has multiple, different solutions  $x \in \mathbb{R}^n$ .
- $\Leftrightarrow$  The column vectors of  $R$  are linearly dependent.
- $\Leftrightarrow$  The row vectors of  $R$  are linearly dependent.

## 2.2 Coordinate Systems, Position and Orientation

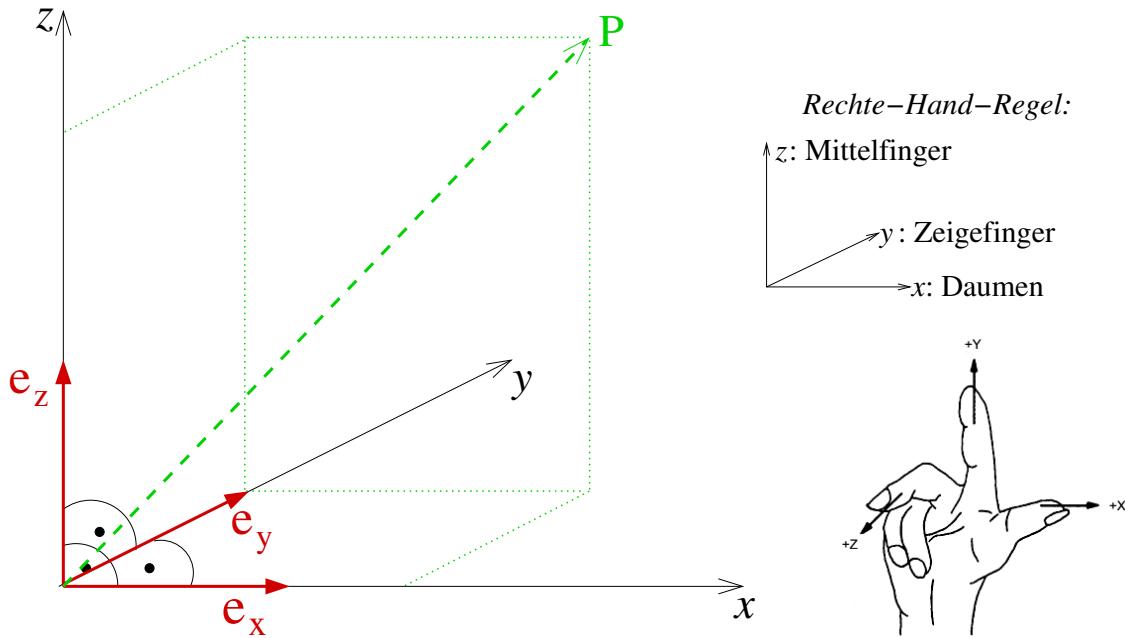
The use of robots means that the robot moves itself and/or other objects through some mechanism in space. To describe the position and orientation of the individual joints of

---

<sup>1</sup>In some contexts, one only speaks of orthogonal matrices, although orthonormal matrices are meant.

the robot, the manipulated objects, and the objects of the surrounding world, *coordinate systems* and conventions for formal description are needed. Often all descriptions are made with reference to a base or world coordinate system.

In robotics, *right-angled, Cartesian coordinate systems* are used:



The vectors  $e_x, e_y, e_z \in \mathbb{R}^3$  are mutually orthogonal unit vectors, i.e., of length  $\|e_x\| = \|e_y\| = \|e_z\| = 1$  and with  $e_x^T e_y = e_x^T e_z = e_y^T e_z = 0$ , and form a right-handed system,

$$\text{i.e., } e_z = e_x \times e_y. \text{ Usually, } e_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, e_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, e_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

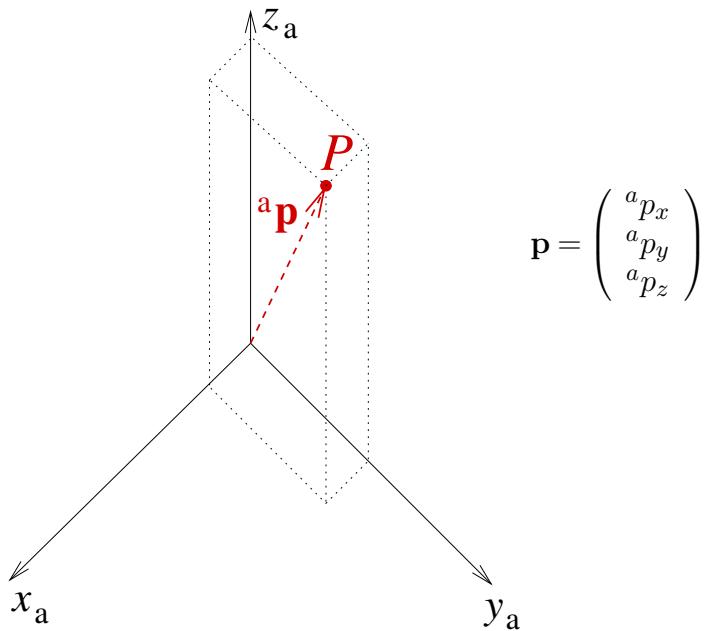
The position vector  $\mathbf{p} \in \mathbb{R}^3$  of a point  $P$  in three-dimensional, Euclidean space is

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}.$$

For example:  $\mathbf{p} = \begin{pmatrix} 1.5 \\ 1 \\ 2 \end{pmatrix}$ , or in coordinate notation:  $\mathbf{p} = 1.5\mathbf{e}_x + 1\mathbf{e}_y + 2\mathbf{e}_z$  or as a matrix-vector product:

$$\mathbf{p} = \underbrace{[\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z]}_{(3 \times 3)-\text{Matrix}} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1.5 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 1 \\ 2 \end{pmatrix}$$

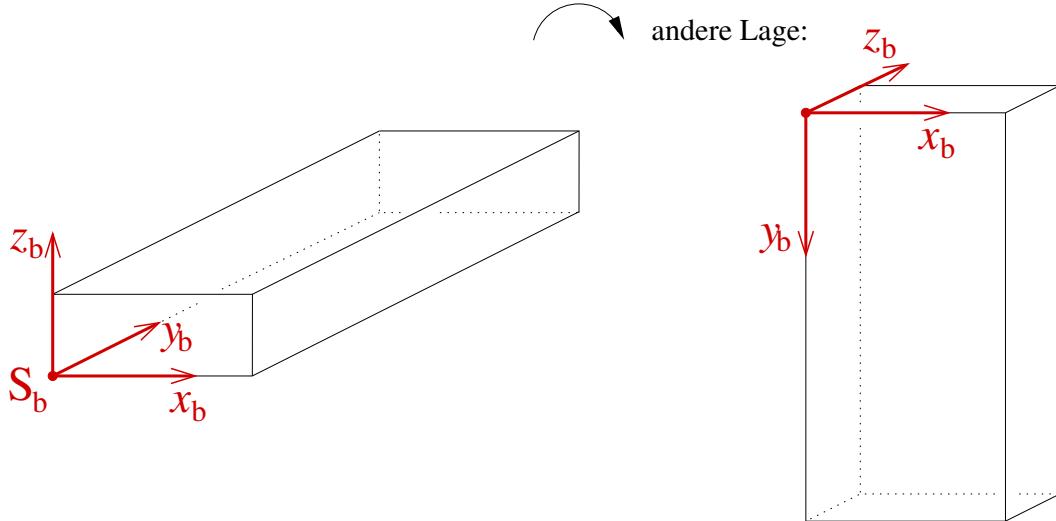
In general, the position vector  ${}^a\mathbf{p}$  of a point  $P$  is specified *relative* to a coordinate system  $S_a$  with coordinate axes  $x_a, y_a, z_a$ :



## Spatial Arrangement of an Object

The pose of an object in space is described by its *position* and *orientation*. To describe the orientation, a coordinate system, e.g.,  $S_b$ , is *fixedly* attached to the object.

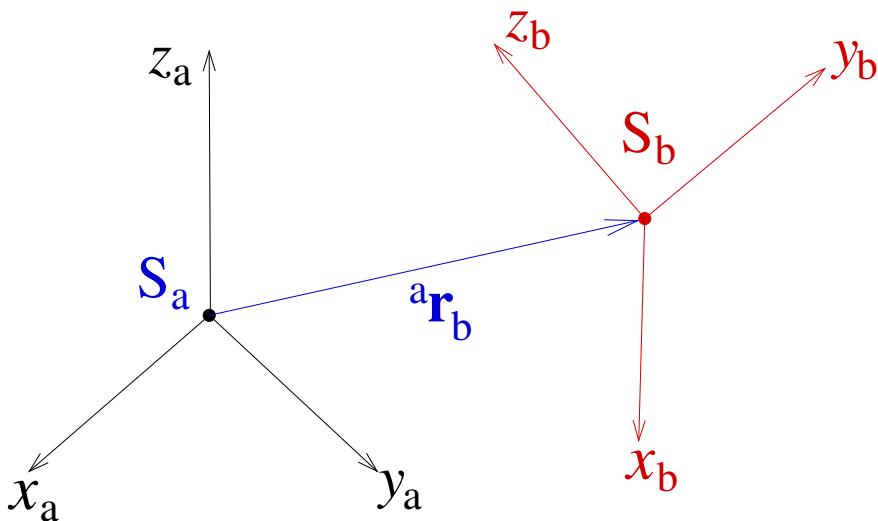
Example:



**Definition 2.1** The spatial arrangement (RAN, cf. VDI Guideline 2860) of an object comprises its position (three translational degrees of freedom, i.e., three real numbers) and orientation (three rotational degrees of freedom, e.g., three rotation angles).

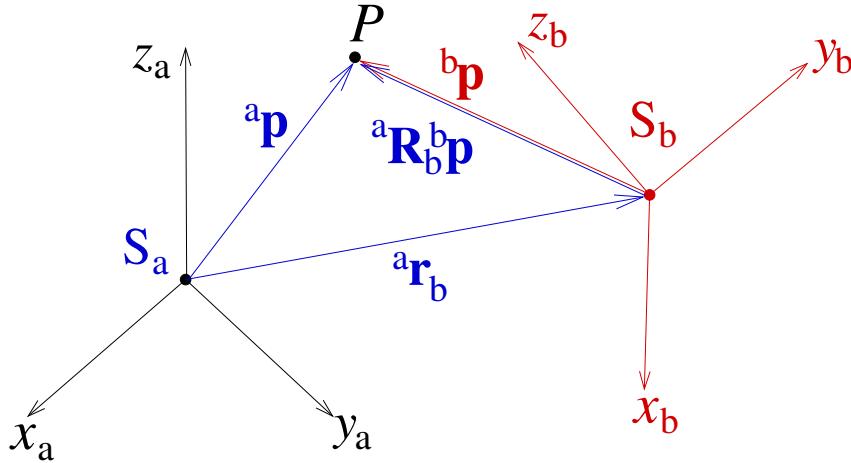
Alternatively, the pose of the object coordinate system  $S_b$  with respect to the world coordinate system ( $S_a$  or  $S_0$ ) can be specified (so-called frame concept).

Example:



The spatial arrangement of the object coordinate system  $S_b$  results from the reference coordinate system  $S_a$  through displacement (translation) and rotation. The vector  ${}^a\mathbf{r}_b \in \mathbb{R}^3$  is the translation vector and represents the coordinates of the origin of  $S_b$  with respect to  $S_a$ .

## 2.3 Classical Transformation Relationship



The classical transformation relationship for point coordinates when changing a coordinate system from \$S\_b\$ to \$S\_a\$ is:

$${}^a \mathbf{p} = {}^a \mathbf{r}_b + {}^a \mathbf{R}_b {}^b \mathbf{p}$$

${}^a \mathbf{p} = \begin{pmatrix} * \\ * \\ * \end{pmatrix} \in \mathbb{R}^3$ : Coordinates of point \$P\$ with respect to \$S\_a\$,

${}^b \mathbf{p} = \begin{pmatrix} * \\ * \\ * \end{pmatrix} \in \mathbb{R}^3$ : Coordinates of point \$P\$ with respect to \$S\_b\$,

${}^a \mathbf{r}_b = \begin{pmatrix} * \\ * \\ * \end{pmatrix} \in \mathbb{R}^3$ : Translation vector,

${}^a \mathbf{R}_b = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} \in \mathbb{R}^{3 \times 3}$ : Rotation (or rotational) matrix between \$S\_a\$ and \$S\_b\$.

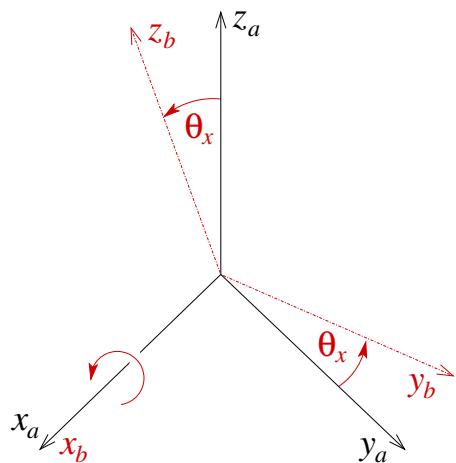
## 2.4 Rotation of a Coordinate System $S_a \rightarrow S_b$

I.e., transformations with  ${}^a\mathbf{r}_b = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  and  ${}^a\mathbf{p} = {}^aR_b {}^b\mathbf{p}$  are considered.

### 2.4.1 Elementary Rotation Matrices

describe rotations about a *single* axis of  $S_a$ .

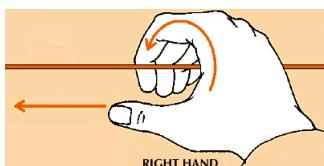
**Example:** Rotation about the  $x_a$ -axis by angle  $\theta_x$



$${}^aR_b := R(x_a; \theta_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{pmatrix}$$

Notes:

- Rotation by a (mathematically) *positive* rotation angle proceeds *counter-clockwise*.
- *Right Hand Rule (2):*



Picture of a right hand: thumb extended and remaining fingers curved. If the thumb corresponds to the direction of the rotation axis, then the direction of the curved remaining fingers corresponds to the positive rotation direction.

Analogous formulas hold for the rotation matrices of the elementary rotations about the  $y$ - and  $z$ -axes:

$$R(y_a; \theta_y) = \begin{pmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{pmatrix}$$

$$R(z_a; \theta_z) = \begin{pmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## 2.4.2 General Rotation Matrix ${}^aR_b$

A general rotation matrix  ${}^aR_b \in \mathbb{R}^{3 \times 3}$  has the following properties:

- The coordinates of the unit vectors  $\mathbf{e}_{x_b}, \mathbf{e}_{y_b}, \mathbf{e}_{z_b}$  of  $S_b$  represented in  $S_a$  are the columns of  ${}^aR_b \in \mathbb{R}^{3 \times 3}$ :

$${}^aR_b = ({}^a\mathbf{e}_{x_b} | {}^a\mathbf{e}_{y_b} | {}^a\mathbf{e}_{z_b})$$

- Orthonormality:  ${}^aR_b \cdot {}^aR_b^T = E$

From orthonormality follows  $\det({}^aR_b) = +1$  (for right-handed coordinate system) or  $-1$  (for left-handed coordinate system), here:  $+1$ , since right-handed coordinate system and rotation.

An orthonormal matrix with determinant equal to 1 is also called a „direction cosine matrix“.

- For the columns, the orthonormality conditions (ONB) hold:

$$\|{}^a\mathbf{e}_{x_b}\| = 1 = \|{}^a\mathbf{e}_{y_b}\| = \|{}^a\mathbf{e}_{z_b}\| \quad (\text{unit vectors})$$

$${}^a\mathbf{e}_{x_b}^T {}^a\mathbf{e}_{y_b} = 0 = {}^a\mathbf{e}_{x_b}^T {}^a\mathbf{e}_{z_b} = {}^a\mathbf{e}_{y_b}^T {}^a\mathbf{e}_{z_b} \quad (\text{pairwise orthogonal})$$

- A consequence of the above properties is:

The  $(3 \times 3)$ -matrix  ${}^aR_b$  has 9 elements, which must satisfy 6 equations (ONB)  
 $\Rightarrow 3$  free parameters remain for describing general rotation matrices.

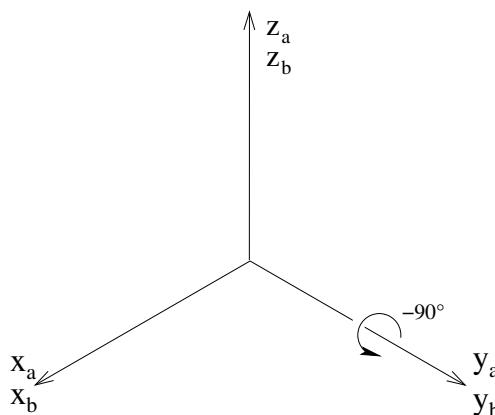
There are different possibilities for choosing these 3 parameters. We will get to know the most frequently used ones in Section 2.4.4.

## 2.4.3 Concatenated Elementary Rotations

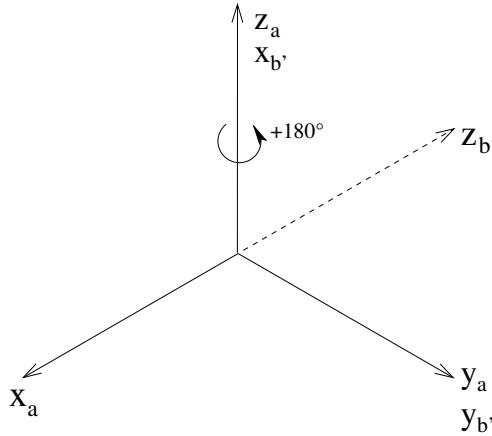
General rotations can be generated by concatenation (i.e., sequential execution) of elementary rotations. The order of sequentially executed elementary (as well as general) rotations is, however, not *commutative*: Changing the order results in a different overall rotation.

**Example:**

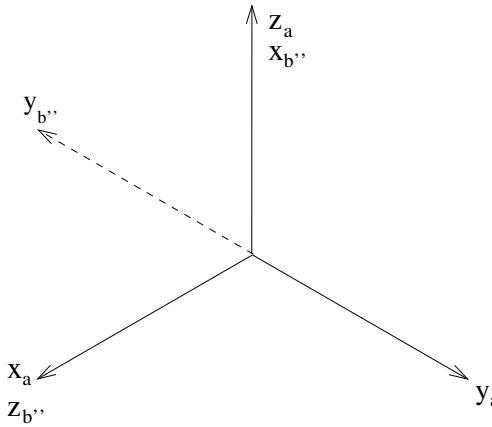
- Coordinate system  $S_b$  is initially coincident with  $S_a$  (i.e., has the same spatial arrangement):



(ii) Rotation of  $S_b$  about the  $y_a$ -axis by  $-90^\circ$  (or  $-\frac{\pi}{2}$ ) yields:



(iii) Rotation of  $S_{b'}$  about the  $z_a$ -axis by  $180^\circ$  (or  $\pi$ ) yields furthermore:



A vector  $\mathbf{u} \in \mathbb{R}^3$  fixed to the coordinate system  $S_b$  is transformed by (ii) into  $\mathbf{v}$  and by (iii) into  $\mathbf{w}$ :

$$\begin{aligned}\mathbf{v} := R(y_a; -90^\circ) \cdot \mathbf{u} &= \begin{pmatrix} \cos \frac{-\pi}{2} & 0 & \sin \frac{-\pi}{2} \\ 0 & 1 & 0 \\ -\sin \frac{-\pi}{2} & 0 & \cos \frac{-\pi}{2} \end{pmatrix} \mathbf{u} \\ &= \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} -u_3 \\ u_2 \\ u_1 \end{pmatrix} \\ \mathbf{w} := R(z_a; 180^\circ) \cdot \mathbf{v} &= \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} -v_1 \\ -v_2 \\ v_3 \end{pmatrix}\end{aligned}$$

$\Rightarrow$  The compound rotation results from  $\mathbf{w} = R(z_a; 180^\circ) \cdot \mathbf{v} = R(z_a; 180^\circ)R(y_a; -90^\circ) \cdot \mathbf{u}$  and thus the rotation matrix is obtained as

$${}^aR_b = R(z_a; 180^\circ) \cdot R(y_a; -90^\circ) = \left( \begin{array}{c|cc} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{array} \right) = \left( {}^a\mathbf{e}_{x_b''} \mid {}^a\mathbf{e}_{y_b''} \mid {}^a\mathbf{e}_{z_b''} \right).$$

The vectors  ${}^a\mathbf{e}_{x_b''}, {}^a\mathbf{e}_{y_b''}, {}^a\mathbf{e}_{z_b''}$  are the coordinates of the three unit vectors of  $S_b$  represented in  $S_a$ . The rotation first about the  $y_a$ -axis and then about the  $z_a$ -axis of the fixed reference system  $S_a$  yields an overall rotation in which the individual rotation matrices are successively multiplied from the left to the preceding rotation matrix (pre-multiplication).

**Bemerkung:**

Es gibt zwei Interpretationen von verketteten Rotationen, ohne dass sich die Gesamtrotationsmatrix  ${}^aR_b$  ändert:

$${}^aR_b = \overbrace{R_n \cdot R_{n-1} \cdot \dots \cdot R_2 \cdot R_1}^{\text{a)} \text{b)}} \quad \text{b)}$$

- a) *Vormultiplikation* (wie im vorangegangenen Beispiel, siehe Rotation bezüglich *festem* Bezugssystem im Abschnitt 2.4.4):

$${}^aR_b = (R_n \cdot (R_{n-1} \cdot \dots \cdot (R_2 \cdot R_1) \dots))$$

Interpretation: Rotation in der Reihenfolge 1, 2, ..., n bedeutet Drehung des (momentanen)  $S_b$  um die *festen*  $S_a$ -Achsen. Bezugssystem ist stets  $S_a$ .

- b) *Nachmultiplikation* (siehe Euler-Winkel im Abschnitt 2.4.4):

$${}^aR_b = ((\dots(R_n \cdot R_{n-1}) \cdot \dots \cdot R_2) \cdot R_1)$$

Interpretation: Eine Rotation in der Reihenfolge n, n - 1, ..., 2, 1 bedeutet eine Drehung des (momentanen)  $S_b$  um die *momentanen*  $S_b$ -Achsen. Bezugssystem ist hier immer das momentane  $S_b$ .

**Wichtig:** Eine Vertauschung der Reihenfolge der einzelnen R-Multiplikationen verändert die Gesamtabbildung! Matrixmultiplikation ist nicht kommutativ!

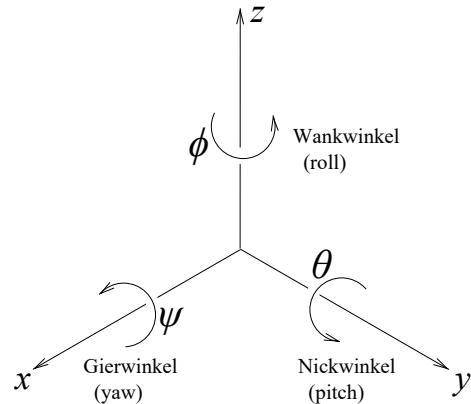
## 2.4.4 Orientation Representation with Three Angle Parameters

The three parameters for representing general rotations can be chosen with respect to a *fixed* or the *current* reference system (see also Section 2.5 of [Spong/Hutchinson/Vidyasagar, 2006] and Section 1.2 of [Siciliano/Khatib, 2016]). In both cases, the concatenated rotations can occur in six different orders about three or two different axes (e.g., X-Y-Z or Z-Y-X), thus in twelve different ways.

Common angle conventions:

### a) Rotation with respect to *fixed* reference system

A common rotation sequence is X-Y-Z about a *fixed* reference system. This is referred to as **RPY angles** ([Spong/Hutchinson/Vidyasagar, 2006], [Siciliano/Sciavicco/Villani/Oriolo, 2009]) or **X-Y-Z Fixed Angles** ([Craig, 1989], [Siciliano/Khatib, 2016]).



In the case of the X-Y-Z sequence, three concatenated rotations are performed with angle  $\psi$  (yaw) about the  $x_a$ -axis, angle  $\theta$  (pitch) about the  $y_a$ -axis, and angle  $\phi$  (roll) about the  $z_a$ -axis of the *fixed* reference system  $S_a$ .

*Pre-multiplication* yields:

$$\begin{aligned} {}^aR_b(\psi, \theta, \phi) &= \left( R(z; \phi) \cdot (R(y; \theta) \cdot R(x; \psi)) \right) \\ &= \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{pmatrix} \\ &= \begin{pmatrix} \cos \phi \cos \theta & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \sin \phi \cos \theta & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi \\ -\sin \theta & \cos \theta \sin \psi & \cos \theta \cos \psi \end{pmatrix} \quad (2.1) \end{aligned}$$

The *direct relationship* between  $\begin{pmatrix} \psi \\ \theta \\ \phi \end{pmatrix}$  and  ${}^aR_b$  consists merely of trigonometry and matrix multiplication and is therefore relatively easy to solve:

$$\underbrace{\begin{pmatrix} \psi \\ \theta \\ \phi \end{pmatrix}}_{\in \mathbb{R}^3} \rightarrow \boxed{\text{DIR RPY: } R(z; \phi) \cdot R(y; \theta) \cdot R(x; \psi)} \rightarrow \underbrace{{}^aR_b(\psi, \theta, \phi)}_{\in \mathbb{R}^{3 \times 3}} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

The *inverse relationship*, on the other hand, leads to a nonlinear system of equations and is considerably more difficult to solve:

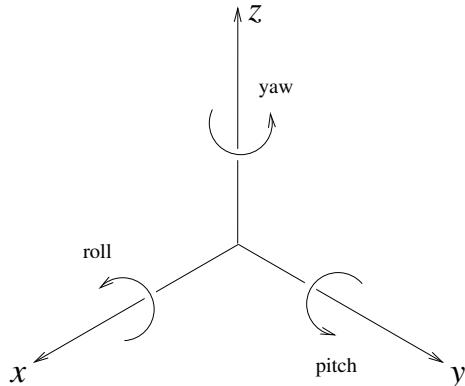
$$\begin{array}{ccc} {}^a\hat{R}_b & \longrightarrow & \boxed{\text{INV RPY:} \\ \text{nonlinear system of equations} \\ {}^aR_b(\psi, \theta, \phi) = {}^a\hat{R}_b} \\ = \text{const} \in \mathbb{R}^{3 \times 3} & & \longrightarrow \underbrace{\begin{pmatrix} \psi \\ \theta \\ \phi \end{pmatrix}}_{\in \mathbb{R}^3} \end{array}$$

The solution of the linear system of equations  ${}^aR_b(\psi, \theta, \phi) = {}^a\hat{R}_b$  can be done step by step using equation (2.1):

1. First,  $\theta$  is determined from matrix element  $R_{3,1}$ ,
2. then  $\psi$  from element  $R_{3,2}$  or  $R_{3,3}$ , and
3. finally  $\phi$  from one of the other matrix elements, taking into account trigonometric relationships.

**Note:** In vehicle engineering (and in the book by Craig (1989)), a different convention for RPY angles is usually used (see figure on the right).

**Caution:** Angle conventions are often context-dependent!



b) **Rotation with respect to *current* reference system (Euler angles):**

An alternative orientation representation is achieved through three concatenated rotations by the angles  $\alpha, \beta, \gamma$  (in this order) about the axes of the *current* reference system  $S_b$ .

With the convention of the most commonly used Z-Y-Z Euler angles (rotation by  $\alpha$  about the original  $z$ -axis, rotation by  $\beta$  about the rotated  $y$ -axis, and rotation by  $\gamma$  about the rotated  $z$ -axis), using *post-multiplication*:

$${}^aR_b(\alpha, \beta, \gamma) = ((R(z; \alpha) \cdot R(y; \beta)) \cdot R(z; \gamma))$$

or with the Z-Y-X Euler angles accordingly:

$${}^aR_b(\alpha, \beta, \gamma) = ((R(z; \alpha) \cdot R(y; \beta)) \cdot R(x; \gamma)).$$

**Remarks:**

- Due to the two interpretations of concatenated rotations, rotation about three axes of the *fixed* reference system yields the identical overall rotation as rotation in reverse order about the axes of the *current* reference system, i.e., the X-Y-Z RPY angles (X-Y-Z fixed angles) correspond to the Z-Y-X Euler angles.
- Regardless of the rotation order, a singularity occurs in the orientation representation with three angle parameters when the axes of the first and last rotations coincide.

- If the rotations occur about three different axes of the *fixed* reference system, these representations are also often called **Cardan angles** (X-Y-Z, X-Z-Y, Y-X-Z, Y-Z-X, Z-X-Y, Z-Y-X).
- The solution to the inverse problem  ${}^aR_b \rightarrow \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$  or  ${}^aR_b \rightarrow \begin{pmatrix} \psi \\ \theta \\ \phi \end{pmatrix}$  is not always unique.
- There exists *no* set of three minimal rotation angles that can represent all possible orientations in angle space *uniquely*.
- Besides rotation matrices or three angle parameters, an orientation representation can also be done using *quaternions* (see Appendix A.1).
- A general rotation can also be specified with respect to a single rotation angle, which is measured with respect to a general rotation axis that does not coincide with any of the axes of the fixed or current reference system (see Appendix A.2).

**Literature reference:** In [Craig, 1989], Appendix B, one can find 12 representations each (and a total of 24) of general rotations through variations of the conventions for Euler and Cardan angles.

## 2.5 Homogeneous Transformations

The classical transformation relationship was (see Section 2.3):

$${}^a\mathbf{p} = {}^a\mathbf{r}_b + {}^aR_b \cdot {}^b\mathbf{p}$$

A simpler representation is obtained with **homogeneous transformation matrices**  ${}^aT_b$ :

$$\underbrace{{}^a\hat{\mathbf{p}}}_{\in \mathbb{R}^4} = \underbrace{\begin{pmatrix} {}^a\mathbf{p} \\ 1 \end{pmatrix}}_{=: {}^aT_b} = \underbrace{\left( \begin{array}{c|c} {}^aR_b & {}^a\mathbf{r}_b \\ \hline \mathbf{0}^T & 1 \end{array} \right)}_{\in \mathbb{R}^{4 \times 4}} \cdot \underbrace{\begin{pmatrix} {}^b\mathbf{p} \\ 1 \end{pmatrix}}_{\in \mathbb{R}^4} = \underbrace{{}^aT_b}_{\in \mathbb{R}^{4 \times 4}} \cdot \underbrace{{}^b\hat{\mathbf{p}}}_{\in \mathbb{R}^4}$$

The **advantage** lies in the unified treatment of rotations **and** translations of an object through a single matrix/vector multiplication instead of a matrix/vector multiplication and a vector addition.

The **inverse of the homogeneous transformation matrix** is given by:

$$({}^aT_b)^{-1} = \left( \begin{array}{c|c} ({}^aR_b)^T & -({}^aR_b)^T \cdot {}^a\mathbf{r}_b \\ \hline \mathbf{0}^T & 1 \end{array} \right)$$

Proof: For the multiplication of block matrices, the same rules apply as for normal matrix multiplication

$$\underbrace{\begin{pmatrix} {}^aR_b & | & {}^a\mathbf{r}_b \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix}}_{({}^aT_b)} \underbrace{\begin{pmatrix} ({}^aR_b)^T & | & -({}^aR_b)^T \cdot {}^a\mathbf{r}_b \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix}}_{({}^aT_b)^{-1}} = \\ = \begin{pmatrix} {}^aR_b \cdot ({}^aR_b)^T + {}^a\mathbf{r}_b \cdot \mathbf{0}^T & | & {}^aR_b \cdot (-{}^aR_b)^T \cdot {}^a\mathbf{r}_b + {}^a\mathbf{r}_b \cdot 1 \\ \hline \mathbf{0}^T \cdot ({}^aR_b)^T + 1 \cdot \mathbf{0}^T & | & \mathbf{0}^T \cdot (-{}^aR_b)^T \cdot {}^a\mathbf{r}_b + 1 \cdot 1 \end{pmatrix} = \begin{pmatrix} E_3 & | & \mathbf{0} \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix} = E_4$$

### Note:

While it holds for rotation matrices that the transposed matrix equals the inverse matrix:

$$({}^aR_b)^{-1} = ({}^aR_b)^T,$$

for homogeneous transformation matrices this does **not** hold:

$$({}^aT_b)^{-1} \neq ({}^aT_b)^T!$$

### Special cases:

a) pure translation (=: Trans)

$${}^aT_b = \text{Trans}(r_x, r_y, r_z) := \begin{pmatrix} E_3 & | & \mathbf{r} \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix}$$

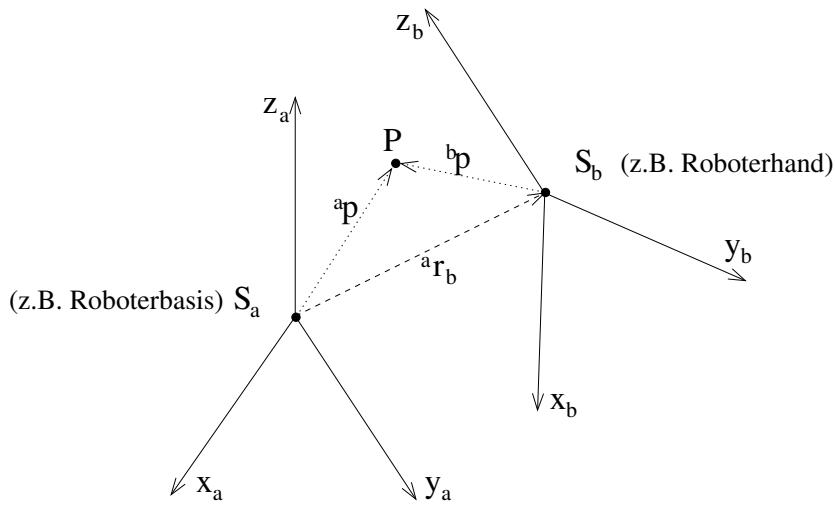
b) pure elementary rotations (=: Rot)

$$\begin{aligned} {}^aT_b &= \text{Rot}(x; \theta_x) := \begin{pmatrix} R(x; \theta_x) & | & \mathbf{0} \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix} \\ {}^aT_b &= \text{Rot}(y; \theta_y) := \begin{pmatrix} R(y; \theta_y) & | & \mathbf{0} \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix} \end{aligned}$$

$\text{Rot}(z, \theta_z)$  analogously.

**Product of two homogeneous matrices** (concatenation of two homogeneous transformations):

$$\begin{aligned} {}^aT_b &= {}^aT_1 \cdot {}^1T_b = \begin{pmatrix} {}^aR_1 & | & {}^a\mathbf{r}_1 \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^1R_b & | & {}^1\mathbf{r}_b \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix} \\ &= \begin{pmatrix} {}^aR_1 {}^1R_b & | & {}^aR_1 \cdot {}^1\mathbf{r}_b + {}^a\mathbf{r}_1 \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix} = \begin{pmatrix} {}^aR_b & | & {}^a\mathbf{r}_b \\ \hline \mathbf{0}^T & | & 1 \end{pmatrix} \end{aligned}$$

**Example:**

Coordinate vector of  $P$  with respect to  $S_a$  is  ${}^a \mathbf{p}$ ,  
 Coordinate vector of  $P$  with respect to  $S_b$  is  ${}^b \mathbf{p}$ :  $\Rightarrow \begin{pmatrix} {}^a \mathbf{p} \\ 1 \end{pmatrix} = {}^a T_b \begin{pmatrix} {}^b \mathbf{p} \\ 1 \end{pmatrix}$

**Application and interpretation of homogeneous transformation matrices  ${}^a T_b$ :**

- a)  ${}^a T_b$  describes the spatial arrangement of a frame (or coordinate system, CS)  $S_b$  relative to CS  $S_a$ .  
 Here,  ${}^a \mathbf{r}_b$  is the coordinate vector of the origin of  $S_b$  with respect to  $S_a$ .  
 The column vectors of the rotation matrix  ${}^a R_b$  are the coordinate vectors of the unit vectors of  $S_b$  with respect to  $S_a$  (with coinciding origins of  $S_a$  and  $S_b$ ).
- b) The associated transformation mapping  ${}^a T_b : {}^b \mathbf{p} \mapsto {}^a \mathbf{p}$  is given by  ${}^a \hat{\mathbf{p}} = {}^a T_b \cdot {}^b \hat{\mathbf{p}}$ .

**Generalized homogeneous transformations:**

$${}^a T_b = \left( \begin{array}{c|c} {}^a R_b & {}^a \mathbf{r}_b \\ \hline \mathbf{f}^T & w \end{array} \right)$$

${}^a R_b$ : rotation,  $\mathbf{r}$ : translation,  $w > 0$ : scaling factor (in Robotics 1:  $w = 1$ ),

$\mathbf{f}^T$ : perspective transformation (in Robotics 1:  $\mathbf{f}^T = (0, 0, 0)$ ).

Generalized homogeneous transformations are used, e.g., for camera modeling (see chapter on external sensor systems) or in computer graphics.

## 2.6 Computational Efficiency

- (i) Practical aspects of computing with homogeneous transformation matrices:
  - Advantage: compact, symbolic description of general coordinate transformations,
  - Disadvantage: inefficient in terms of computation time, many unnecessary multiplications with zeros or ones.

In computation-time-intensive applications, the classical transformation relationship  ${}^a\mathbf{p} = {}^a\mathbf{r}_b + {}^aR_b \cdot {}^b\mathbf{p}$  is therefore often used.

(ii) The most expensive sub-operations are concatenated rotations:

$$\text{a) } {}^a\mathbf{p} = {}^aR_b \cdot {}^b\mathbf{p} = \underbrace{{}^aR_n \cdot {}^nR_{n-1}}_{1)} \cdot \underbrace{{}^{n-1}R_{n-2} \cdot \dots \cdot {}^2R_1}_{2)} \cdot \underbrace{{}^1R_b \cdot {}^b\mathbf{p}}_{n)}$$

$\Rightarrow n$  matrix-matrix multiplications, one matrix-vector multiplication

$$\text{b) } {}^a\mathbf{p} = {}^aR_b \cdot {}^b\mathbf{p} = {}^aR_n \cdot {}^nR_{n-1} \cdot {}^{n-1}R_{n-2} \cdot \dots \cdot {}^2R_1 \cdot \underbrace{{}^1R_b \cdot {}^b\mathbf{p}}_{1)} \underbrace{{}^2\mathbf{p}}_{2)}$$

$\Rightarrow n+1$  matrix-vector multiplications

$(3 \times 3)$ -matrix-matrix multiplication: 27 multiplications + 18 additions

$(3 \times 3)$ -matrix-vector multiplication: 9 multiplications + 6 additions

$\Rightarrow$  The calculation according to b) is (for a single evaluation of the coordinate transformation) much more efficient than according to a) (by approximately a factor of 3)!

# Chapter 3

## Robot Kinematics

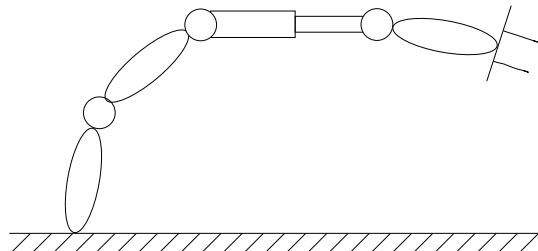
In this chapter, mainly manipulators, i.e. multi-jointed robot arms and legs, are examined. The kinematics of mobile (wheeled and legged) robots will be considered in connection with the topics velocity and Jacobian matrix in Chapter 4.

### 3.1 Forward Kinematics Model

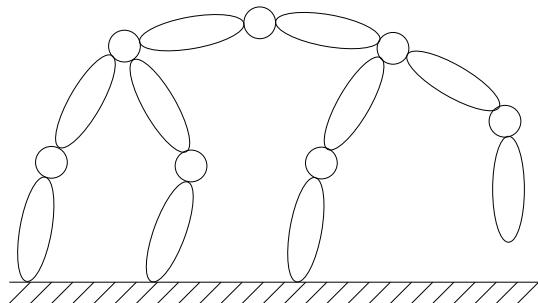
#### 3.1.1 Kinematic Chains

We consider the idealized modeling of a manipulator (or a robot arm or leg) as ***kinematic chain***, i.e. as a set of concatenated, adjacent **rigid** (not elastic!) ***links*** (rigid links) connected by rigid ***joints*** (axes, rigid joints).

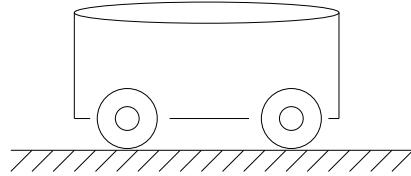
Industrial robots are generally modeled as ***open*** kinematic chains. That is, there is no force closure via the environment with the manipulator base point:



In contrast, for example, to legged robots

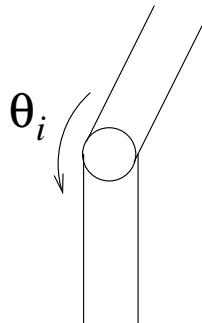


or wheeled robots.

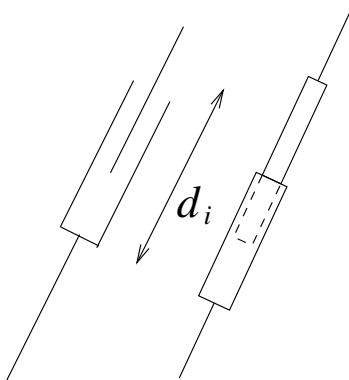


Examples of elementary joint structures, each with one degree of freedom (DOF), are

- **Revolute joints** (revolute joints, rotary joints)



- **Prismatic joints** (prismatic joints, linear joints)



**More complex joint structures** (e.g., ball joints with two DOF) can be modeled by concatenated single-DOF revolute joints with zero distance.

## Procedure

A robot (manipulator) consists of a set of bodies, denoted as links, in a chain arrangement, which are connected by joints:

- Each joint has one degree of freedom of motion, either as a revolute or prismatic joint.
- For a manipulator with  $n$  joints, which are numbered from 1 to  $n$ , there are  $n + 1$  links, which are numbered from 0 to  $n$ . Link 0 is the base of the manipulator, which is generally fixed, and link  $n$  carries the end effector.
- Joint  $i$  connects links  $i - 1$  and  $i$ .

To describe the positions of the individual links, a local coordinate system  $S_i$  is rigidly attached to each link  $i$ . The most widespread method for the systematic assignment of coordinate systems to the individual links goes back to Denavit and Hartenberg (1955).

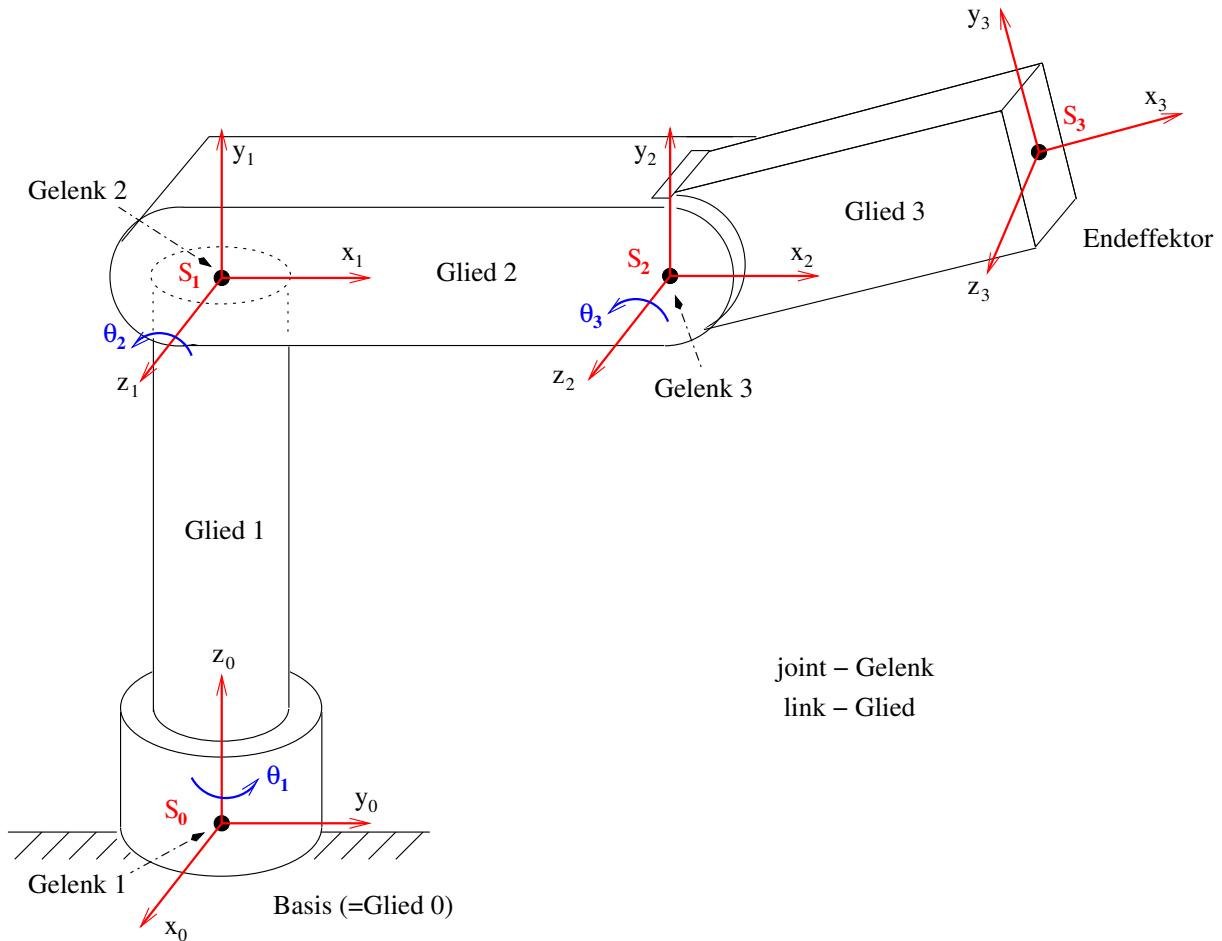
### 3.1.2 Kinematic Modeling with Frame Concept

A link-fixed coordinate system  $S_i$  is attached to each robot link  $i$  at the respective joint  $i+1$ :

- at the base point  $F$ : the base coordinate system (base frame)  $S_0 = S_F$  (also denoted as inertial coordinate system, inertial system, inertial frame) and
- at the end effector  $E$ : the end effector coordinate system  $S_n = S_E$ .

That is, a robot (manipulator) with  $n$  joints is defined by  $n+1$  coordinate systems (short: CSs, frames)  $S_i$ .

**Example:** Three-link manipulator



The **goal** of the considerations is to determine the kinematic model structure for a manipulator with frames  $S_i$ ,  $i = 0, \dots, n$ :

$${}^F T_E = {}^0 T_n = {}^0 T_1 \cdot {}^1 T_2 \cdot \dots \cdot {}^{n-2} T_{n-1} \cdot {}^{n-1} T_n = \prod_{i=1}^n {}^{i-1} T_i.$$

The homogeneous transformation matrix  ${}^F T_E = {}^0 T_n(\mathbf{q}, \mathbf{f})$  is a function

- of the generalized joint variables  $\mathbf{q}$ :

$$\mathbf{q}^T = (\theta_1, \theta_2, \dots, d_i, \dots, \theta_n) \in \mathbb{R}^n \quad (\theta_i: \text{revolute joint}, d_i: \text{prismatic joint})$$

- and the geometric parameters  $\mathbf{f}$  of the manipulator links.

Often only local dependencies occur, i.e.  ${}^{i-1}T_i = {}^{i-1}T_i(q_i, f_i)$  ( $q_i \in \mathbb{R}^1$ , the link-specific parameters  $f_i$  can be multidimensional), which simplifies the modeling.

The frames  $S_i$  can in principle be defined almost arbitrarily. For complex manipulators ( $n \geq 4$ ) a formal definition of the coordinate systems  $S_i$  and the transformations  ${}^{i-1}T_i$  is necessary for computer-aided modeling (→ Section 3.1.3 on Denavit-Hartenberg).

### 3.1.3 Denavit-Hartenberg (DH) Conventions

**Preliminary remarks:**

- The DH conventions go back to Denavit and Hartenberg (1955)<sup>1</sup>
- and are generally only used for *open* kinematic chains.
- There are several variants of DH conventions that differ in details.
- The DH conventions are an international standard. For example, for industrial robots, a data sheet specifying the DH parameters is available for each model.
- However, in principle there are also many other possibilities for frame definition. Depending on the individual case, an individual definition of the frames can be much more practical than the formal DH convention!

Some basic properties of the DH convention are:

- The coordinate systems are located in the respective motion axes.
- The  $z_{i-1}$ -axis lies along the motion axis of the  $i$ -th joint.
- The  $x_i$ -axis is perpendicular to the  $z_{i-1}$ -axis and points away from it.
- The  $x_i$ -axis and the  $z_{i-1}$ -axis have an intersection point.

The specific steps for implementing the DH convention are not contained in the convention itself. In the following, an attempt is made to outline a constructive algorithm for determining the local coordinate systems according to the DH convention, which is applicable in many cases (but not all).

---

<sup>1</sup>R.S. Hartenberg, J. Denavit: *A kinematic notation for lower pair mechanisms based on matrices*, Journal of Applied Mechanics, Vol. 77 (1955) pp. 215-221.

## A Pseudo-Algorithm for the Formal Definition of Frames $S_i$ :

### Prerequisite:

The manipulator is in the ***zero position*** (zero position) with all  $n$  joints.<sup>2</sup>

**Note:** Should the procedure in the individual steps not appear unique in the concrete application case, the observance of the DH properties must first be guaranteed and in case of ambiguities, the choice of a solution with the least effort for the resulting transformation is to be preferred (while observing the DH properties).

- (1) (a) Numbering of links from 0 to  $n$  (0 is the base,  $n$  is the number of joints) and the joints from 1 to  $n$ .

- (b) Definition of the  $z_i$ -axes as coincident with the motion axis of joint  $i + 1$ :

Prismatic joint:  $z_i$ -axis as sliding axis in the direction away from joint  $i + 1$ , i.e. in the direction of the sliding direction (in zero position the prismatic joint is retracted),

Revolute joint:  $z_i$ -axis as rotation axis in the direction of positive rotation angle (is defined)

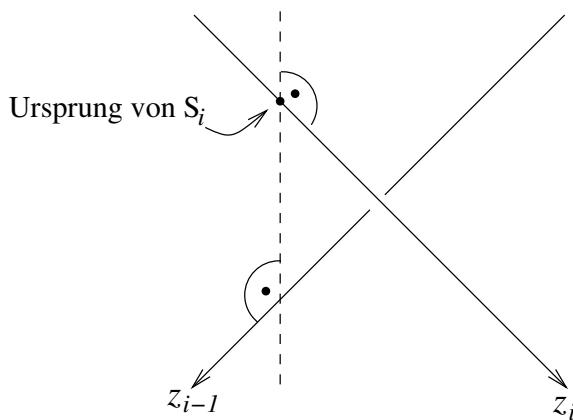
- (2) Definition of the base frame  $S_0$  with origin on the  $z_0$ -axis;

Choice of  $x_0, y_0$  as right-handed coordinate system; often  $x_0, y_0$  parallel to  $x, y$ -axes of the world CS.

For  $i = 1, \dots, n - 1$ : Repetition of points (3) to (5)

- (3) Definition of the origin of  $S_i$ :

- If  $z_{i-1}$  and  $z_i$  intersect: Intersection point becomes origin of  $S_i$ .<sup>3</sup>
- If  $z_{i-1}$  and  $z_i$  are parallel: Definition of the origin on  $z_i$  at joint  $i + 1$ .
- Otherwise (i.e.  $z_{i-1}$  and  $z_i$  are skew): Form common normal to  $z_i$  and  $z_{i-1}$ , origin becomes the intersection point of this with  $z_i$ .



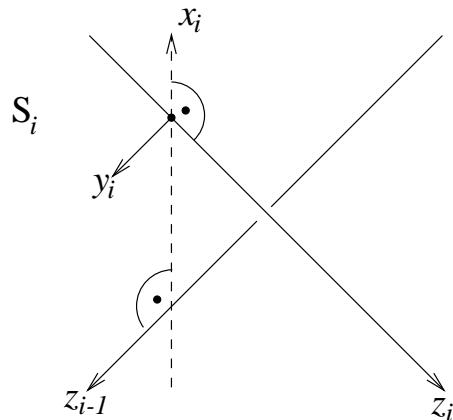
<sup>2</sup>The zero position is usually a matter of definition. For industrial robots with 6 revolute joint axes, the zero position is usually defined such that all arms are extended straight upward. For a prismatic joint, the zero position is usually the state of the fully retracted prismatic joint.

<sup>3</sup>as in the example of the three-link manipulator on page 30 in Section 3.1.2

(4) Definition of the  $x_i$ -axis:

- If  $z_{i-1}$  and  $z_i$  intersect:  $x_i$  in (positive or negative) direction of the normal of the plane spanned by  $z_{i-1}$  and  $z_i$ .
- If  $z_{i-1}$  and  $z_i$  are parallel or skew:  $x_i$ -axis in the direction of the common normal of  $z_{i-1}$  and  $z_i$  through the origin of  $S_i$ , so that the  $x_i$ -axis intersects the  $z_{i-1}$ -axis.
  - If the orientation of the  $x_i$ -axis is not unique, it should, if possible, be directed from the current frame along the link to the next frame.
- If the  $x_i$ -axis is not yet uniquely determined by the above specifications, it should be chosen such that the resulting DH table is as simple as possible.  
Please note: The DH properties must also be observed after defining the  $x_i$ -axis. This can particularly affect the choice of the positive  $x_i$ -direction.

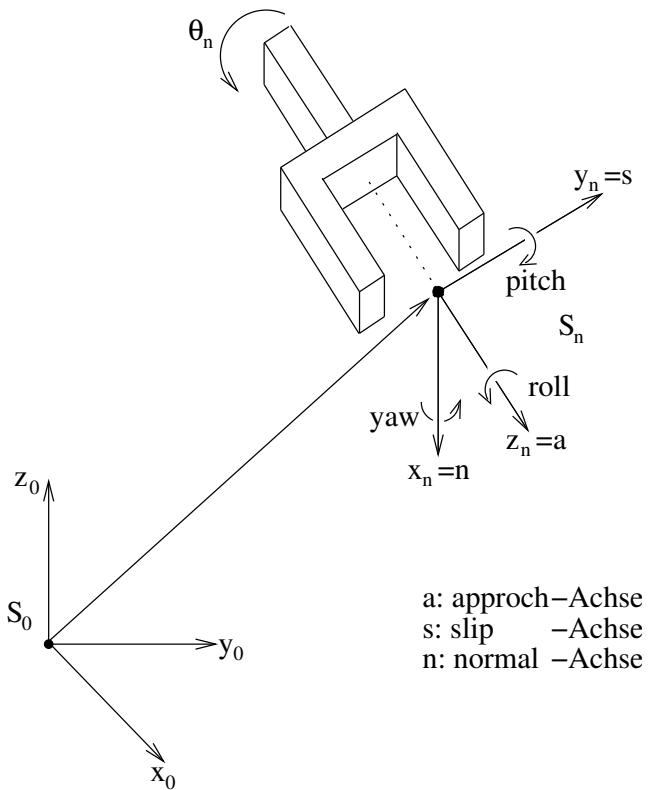
(5) Definition of the  $y_i$ -axis, so that  $x_i, y_i, z_i$  form a right-handed coordinate system.



(6) Definition of the end effector coordinate system  $S_n$ :

The origin of  $S_n$  is usually placed in the so-called tool center point (TCP).

- (a) If there are no special requirements for the orientation of  $S_n$ , the simplest possible transformation can be used for the transformation from  $S_{n-1}$  to  $S_n$ . Often a simple translation is sufficient.
- (b) If the  $n$ -th joint is a revolute joint and the tool (tool) is a simple gripper, the end effector frame  $x_n, y_n, z_n$  (tool frame) is usually defined as follows, i.e. compared to variant (a), an additional rotation may be necessary to obtain this position:

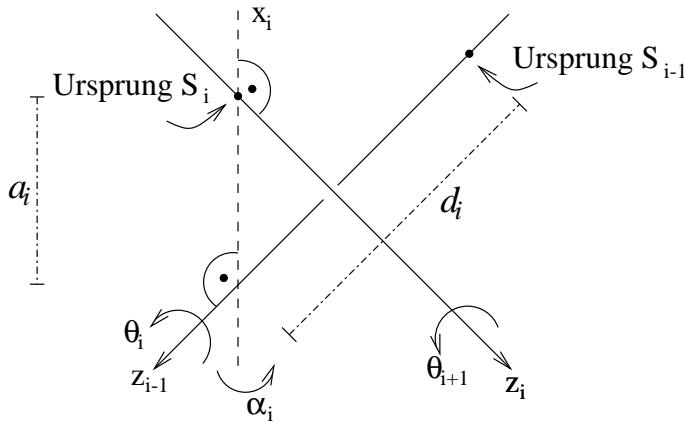


As a result of steps (1) – (5) and the transformations possibly additionally necessary for (6) for the sketched position of the end effector coordinate system  $S_n$ , one then obtains:

$${}^0T_E = \begin{pmatrix} {}^0R_E & {}^0\mathbf{r}_E \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} n_x & s_x & a_x & r_x \\ n_y & s_y & a_y & r_y \\ n_z & s_z & a_z & r_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} & {}^0\mathbf{r}_E \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(7) Creation of a table of link parameters  $\theta_i, d_i, a_i$  and  $\alpha_i, i = 1, \dots, n$ :

- $\theta_i$  = Angle between  $x_{i-1}$  and  $x_i$ , measured about  $z_{i-1}$ ;  
 $\theta_i$  is variable if joint  $i$  is a revolute joint;
- $d_i$  = Distance from the  $S_{i-1}$  origin along the  $z_{i-1}$ -axis to the intersection with the  $x_i$ -axis;  
 $d_i$  is variable if joint  $i$  is a prismatic joint;
- $a_i$  = Distance from the intersection of the  $z_{i-1}$ -axis with the  $x_i$ -axis to the origin  
of  $S_i$  along the  $x_i$ -axis ( $a_i$  can also be negative, depending on the orientation of  $x_i$ );
- $\alpha_i$  = Angle between  $z_{i-1}$  and  $z_i$ , measured about  $x_i$ .



(8) Formation of the homogeneous transformation matrices  ${}^{i-1}T_i$ <sup>4</sup>:

$$\begin{aligned} {}^{i-1}T_i &= \text{Rot}(z; \theta_i) \cdot \text{Trans}(0, 0, d_i) \cdot \text{Trans}(a_i, 0, 0) \cdot \text{Rot}(x; \alpha_i) \\ &= \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

(9) Calculation of  ${}^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot \dots \cdot {}^{n-1}T_n$ ;

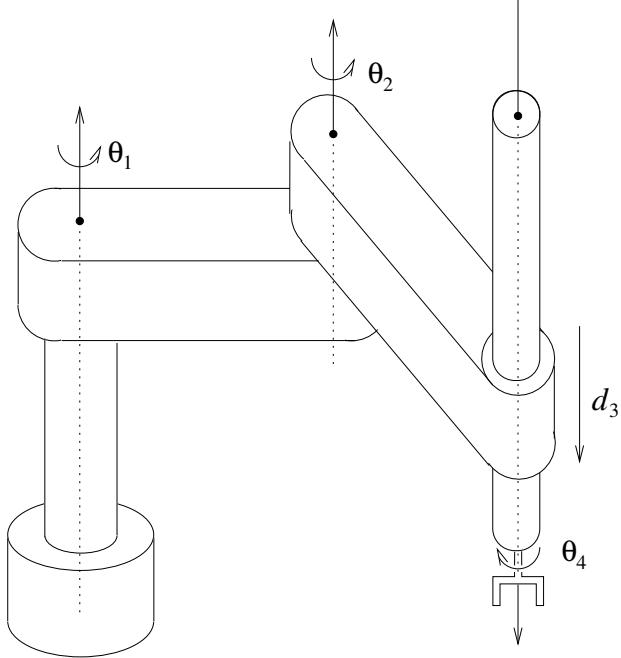
${}^0T_n$  now specifies the position and orientation of the tool frame in base coordinates.

---

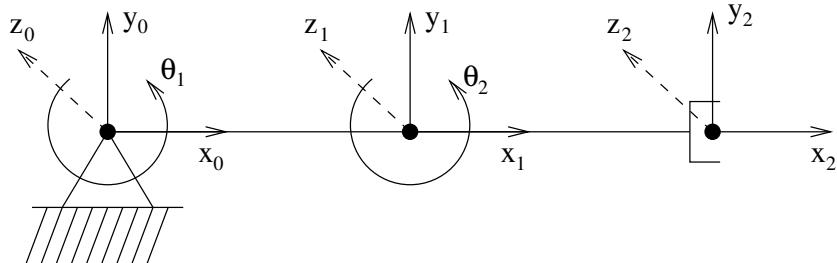
<sup>4</sup>In contrast to the sequential execution of general transformations, rotation about and translation along the same axis are commutative, i.e.  $\text{Trans}(a_i, 0, 0) \cdot \text{Rot}(x; \alpha_i) = \text{Rot}(x; \alpha_i) \cdot \text{Trans}(a_i, 0, 0)$  and  $\text{Rot}(z; \theta_i) \cdot \text{Trans}(0, 0, d_i) = \text{Trans}(0, 0, d_i) \cdot \text{Rot}(z; \theta_i)$ .

## 1. Example: SCARA Manipulator

A SCARA<sup>5</sup> manipulator generally consists of three horizontal revolute joints and one vertical prismatic joint



Here we consider a simplification as a planar 2-DoF SCARA manipulator with the first two revolute joints. The *zero position* is defined as:

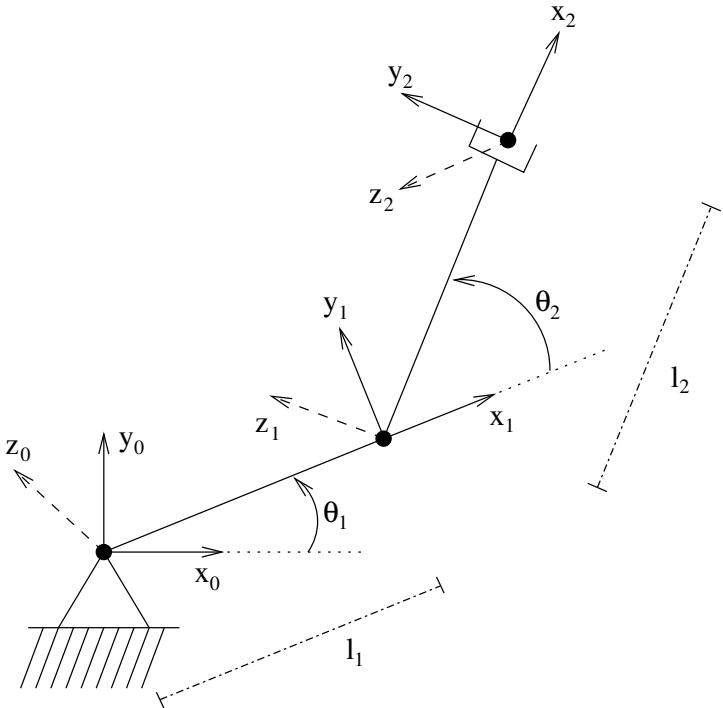


The dashed  $z_i$ -axes point here orthogonally out of the drawing plane in the direction toward the viewer. The  $x_{i-1}$ -axis generally points in the direction toward joint  $i$ . For further simplification, frame  $S_2$  is defined here as in step (6.a) of the DH algorithm and not as in (6.b).

---

<sup>5</sup>SCARA: Selectively Compliant Assembly Robot Arm

General position:



The resulting table for modeling from step (7) with the parameters in the order of the transformations is:

Link $i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$ (variable)	0	$l_1$	0
2	$\theta_2$ (variable)	0	$l_2$	0

Continuing with step (8) of the DH algorithm:

$${}^{i-1}T_i = \underbrace{\text{Rot}(z; \theta_i) \cdot \text{Trans}(0, 0, \underbrace{d_i}_0)}_{E_4} \cdot \underbrace{\text{Trans}(\underbrace{a_i}_{l_i}, 0, 0)}_{E_4} \cdot \underbrace{\text{Rot}(x; \underbrace{\alpha_i}_0)}_{E_4}$$

$$= \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot E_4 \cdot \left( \begin{array}{ccc|c} E_3 & & l_i & \\ & 0 & 0 & \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \cdot E_4$$

$$\implies {}^0T_E(\theta_1, \theta_2) = {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) = \prod_{i=1}^2 \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & \cos \theta_i l_i \\ \sin \theta_i & \cos \theta_i & 0 & \sin \theta_i l_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 & -\cos \theta_1 \sin \theta_2 - \sin \theta_1 \cos \theta_2 & 0 & (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)l_2 + \cos \theta_1 l_1 \\ \sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2 & -\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 & 0 & (\sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2)l_2 + \sin \theta_1 l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

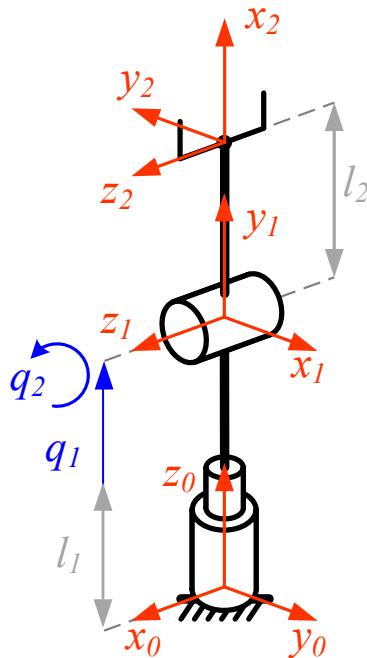
$$\text{with } \sin(\theta_1 + \theta_2) = \sin \theta_1 \cdot \cos \theta_2 + \cos \theta_1 \cdot \sin \theta_2 \\ \cos(\theta_1 + \theta_2) = \cos \theta_1 \cdot \cos \theta_2 - \sin \theta_1 \cdot \sin \theta_2$$

follows:  ${}^0T_E(\theta_1, \theta_2) = \left( \begin{array}{c|c} {}^0R_E(\theta_1, \theta_2) & {}^0\mathbf{r}_E(\theta_1, \theta_2) \\ \hline \mathbf{0}^\top & 1 \end{array} \right)$

$$= \left( \begin{array}{cccc} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right)$$

## 2. Example: 2-DOF Prismatic-Revolute Joint Arm<sup>6</sup>

Now a two-joint robot consisting of a prismatic joint followed by a revolute joint is considered. The following sketch shows the robot in zero position with drawn coordinate systems  $S_0$  to  $S_2$  according to steps (1) to (6) of the DH algorithm.



The resulting table of DH parameters from step (7) is:

Link $i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\frac{\pi}{2}$	$l_1 + q_1$ (variable)	0	$\frac{\pi}{2}$
2	$\frac{\pi}{2} + q_2$ (variable)	0	$l_2$	0

The transformations from the base CS to the end effector CS are obtained according to step (8):

$${}^0T_1 = \text{Rot}(z; \frac{\pi}{2}) \cdot \text{Trans}(0, 0, l_1 + q_1) \cdot \text{Trans}(0, 0, 0) \cdot \text{Rot}(x; \frac{\pi}{2})$$

$$= \left( \begin{array}{cccc} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} & 0 & 0 \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right) \cdot \left( \begin{array}{c|c} E_3 & 0 \\ \hline 0 & l_1 + q_1 \\ 0 & 0 \\ 1 & 1 \end{array} \right) \cdot E_4 \cdot \left( \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} & 0 \\ 0 & \sin \frac{\pi}{2} & \cos \frac{\pi}{2} & 0 \\ 0 & 0 & 0 & 1 \end{array} \right)$$

<sup>6</sup>This example is based on an elaboration by Jérôme Kirchhoff.

$$= \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & l_1 + q_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_1 + q_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Analog zum 2-DoF-SCARA Beispiel erhält man für  ${}^1T_2$ :

$${}^1T_2 = \begin{pmatrix} \cos(\frac{\pi}{2} + q_2) & -\sin(\frac{\pi}{2} + q_2) & 0 & \cos(\frac{\pi}{2} + q_2) \cdot l_2 \\ \sin(\frac{\pi}{2} + q_2) & \cos(\frac{\pi}{2} + q_2) & 0 & \sin(\frac{\pi}{2} + q_2) \cdot l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Rightarrow {}^0T_E(q_1, q_2) = {}^0T_1(q_1) \cdot {}^1T_2(q_2) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \cos(\frac{\pi}{2} + q_2) & -\sin(\frac{\pi}{2} + q_2) & 0 & \cos(\frac{\pi}{2} + q_2) \cdot l_2 \\ \sin(\frac{\pi}{2} + q_2) & \cos(\frac{\pi}{2} + q_2) & 0 & \sin(\frac{\pi}{2} + q_2) \cdot l_2 + l_1 + q_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**Intermediate conclusion:** If

- the kinematic robot model with frames  $S_i, i = 0, \dots, n$ , (e.g. according to DH convention),
- the constant parameters  $\mathbf{f}$  of the robot links (i.e. the geometric dimensions) and
- the current values of the variable joint coordinates  $\mathbf{q} \in \mathbb{R}^n$  (with  $q_i = \theta_i$ , if joint  $i$  is a revolute joint, or  $q_i = d_i$ , if joint  $i$  is a prismatic joint)

are given, then the **forward kinematics model** (or the forward solution, FK) is given as:

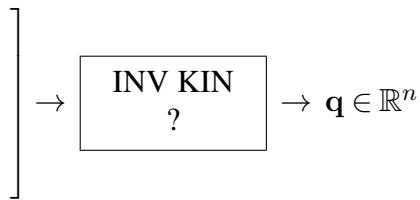
given joint positions  $\mathbf{q} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix} \rightarrow \boxed{{}^0T_E = {}^0T_1 \cdot \dots \cdot {}^{n-1}T_E}$

$\rightarrow {}^0T_E = \underbrace{\begin{pmatrix} {}^0R_E & {}^0\mathbf{r}_E \\ * & * \\ * & * \\ * & * \\ \hline 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{pose of the tool frame}}$

Much more difficult than the FK is the determination of the kinematic inverse solution (IK) examined in the following.

## 3.2 Inverse (or Reverse) Kinematics Model

Given: The position  ${}^0\tilde{\mathbf{r}}_E \in \mathbb{R}^3$  and  
either a) the orientation  ${}^0\tilde{R}_E \in \mathbb{R}^{3 \times 3}$  of the tool frame  
in base coordinates, i.e.  ${}^0\tilde{T}_E \in \mathbb{R}^{4 \times 4}$   
or b) three orientation angles of the tool frame  
(e.g. the Cardan or the Euler angles)



### 3.2.1 General Properties of the IK

Sought is  $\mathbf{q} = ? \in \mathbb{R}^n$  such that:

$${}^0T_E(\mathbf{q}) \stackrel{VWL}{=} \left( \begin{array}{c|c} {}^0R_E(\mathbf{q}) & {}^0\mathbf{r}_E(\mathbf{q}) \\ \hline \mathbf{0}^T & 1 \end{array} \right) \stackrel{!}{=} \left( \begin{array}{c|c} {}^0\tilde{R}_E & {}^0\tilde{\mathbf{r}}_E \\ \hline \mathbf{0}^T & 1 \end{array} \right) = {}^0\tilde{T}_E$$

This formally represents a system of  $3 \cdot 3 + 3 = 12$  nonlinear, mostly trigonometric equations for  $n$  unknowns  $q_i$  ( $n$  is the number of independent elementary joints, DOF).

The calculation of inverse kinematics is a fundamental problem in robotics as well as computer animation, for which there exists a variety of different solution approaches. Before some of them are examined more closely, fundamental properties of inverse kinematics should first be considered.

#### General considerations:

${}^0R_E(\mathbf{q}) - {}^0\tilde{R}_E = O$  provides not 9 but only 3 independent, not redundant, nonlinear equations, since  ${}^0R_E$  is a rotation matrix.

With  ${}^0R_E(\mathbf{q}) \xrightarrow[\text{DIR Euler}]{\text{INV Euler}} \begin{pmatrix} \alpha(\mathbf{q}) \\ \beta(\mathbf{q}) \\ \gamma(\mathbf{q}) \end{pmatrix}$  or  ${}^0R_E(\mathbf{q}) \xrightarrow[\text{DIR RPY}]{\text{INV RPY}} \begin{pmatrix} \psi(\mathbf{q}) \\ \theta(\mathbf{q}) \\ \phi(\mathbf{q}) \end{pmatrix}$

the nonlinear matrix equation system  ${}^0T_E(\mathbf{q}) - {}^0\tilde{T}_E = O \in \mathbb{R}^{4 \times 4}$  can formally be transformed to a system of 6 independent, nonlinear equations:

$$\begin{aligned} F_1(\mathbf{q}) &= {}^0r_{E,i}(\mathbf{q}) - {}^0\tilde{r}_{E,i} = 0, \quad i = 1, 2, 3 \\ F_4(\mathbf{q}) &= \alpha(\mathbf{q}) - \tilde{\alpha} = 0 \\ F_5(\mathbf{q}) &= \beta(\mathbf{q}) - \tilde{\beta} = 0 \\ F_6(\mathbf{q}) &= \gamma(\mathbf{q}) - \tilde{\gamma} = 0 \end{aligned}$$

in short:  $\mathbf{F}(\mathbf{q}) = \mathbf{0}$ ,  $\mathbf{q} \in \mathbb{R}^n$ ,  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^6$ .

**Consequence:** If the pose  ${}^0\tilde{T}_E$  is given arbitrarily:

- For  $n < 6$  joint variables  $q_i$  there are fewer parameters than equations to be satisfied, i.e. “in general” no solution (or only in special cases of  $n$ -dimensional subsets of  $\mathbb{R}^6$ ).
- For  $n = 6$  joint variables  $q_i$  there are exactly as many parameters as equations to be satisfied, i.e. “in general” one solution for “independently” arranged joint variables.
- For  $n > 6$  there are more parameters than these determining equations, i.e. “in general” infinitely many solutions (infinitely redundant manipulator).

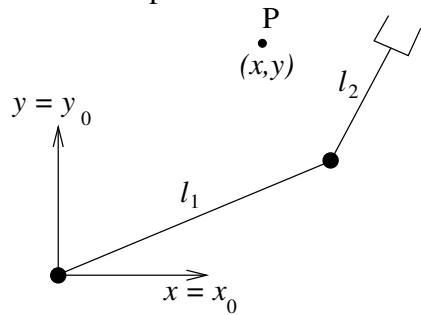
For further considerations, the following are needed:

**Definition 3.1** *Reachable workspace  $\mathcal{A}$ : The subset of 3D space<sup>7</sup> that can be reached by the manipulator with at least one orientation.*

**Definition 3.2** *Dexterous workspace : The subset of  $\mathcal{A}$  that can be reached by the manipulator with all orientations.*

Only if the point  $P$  to be reached by the end effector (= origin of the end effector frame) lies within the reachable workspace, can there be an IK solution!

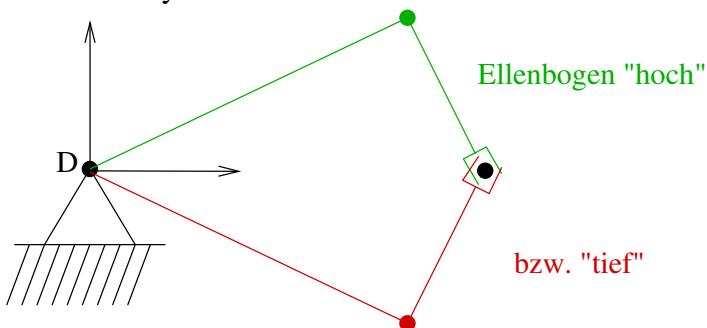
**Example:** Planar 2-DOF SCARA manipulator:



$l_1 \geq l_2$ : The reachable workspace  $\mathcal{A} \subset \mathbb{R}^2$  is the annulus:  $|l_1 - l_2| \leq \sqrt{x^2 + y^2} \leq |l_1 + l_2|$  (and  $z = 0$  in  $\mathbb{R}^3$ ).

Possible cases ( $l_1 \geq l_2$ ):

- $P$  outside of  $\mathcal{A} \Rightarrow$  no IK solution;
- $P$  on outer boundary of  $\mathcal{A} \Rightarrow$  exactly one solution;
- $P$  on inner boundary of  $\mathcal{A}$  (and  $l_1 > l_2$ )  $\Rightarrow$  exactly one solution;
- $P$  inside of  $\mathcal{A} \Rightarrow$  exactly two solutions:



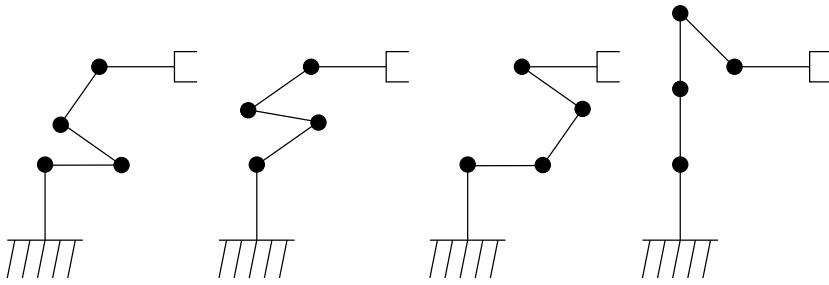
- $P$  on pivot point  $D$  (for  $l_1 = l_2$ )  $\Rightarrow$  infinitely many solutions.

Ambiguity of the IK means that *different* manipulator configurations can lead to *same* workspace specifications.

---

<sup>7</sup>or 2D space in the case of a planar manipulator

**Example:** planar manipulator with four DOF

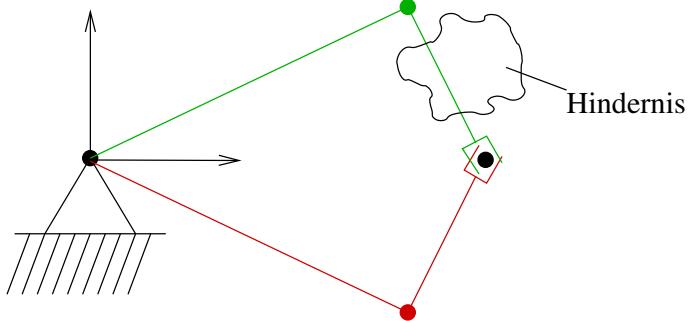


The theoretically existing ambiguity is, however, frequently reduced by:

- technical limitations of the adjustable joint variables  $\mathbf{q}$  (e.g. no  $360^\circ$  rotations)

$$q_{i,\min} \leq q_i \leq q_{i,\max},$$

- geometric constraints to avoid collisions, e.g.



In this case, it is necessary to calculate **all** solutions of the inverse kinematics problem.  
How?

→ In 3.2.2–3.2.3.2, various approaches for calculating INV KIN are presented.

### 3.2.2 Iterative, Numerical Calculation of IK

#### 3.2.2.1 General Procedure

Iterative calculation methods for inverse kinematics are based on the approximate numerical solution of the nonlinear equation system  $\mathbf{F}(\mathbf{q}) = \mathbf{0}$ ,  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^6$ ,  $n = \# \text{DOF}$ , with a numerical calculation method. Often, either methods for solving nonlinear equation systems or optimization methods for minimizing the residual of the squared errors  $\|\mathbf{F}\|^2$  are used.

**Example:** Classical Newton method for numerical solution of nonlinear equation systems in case  $n = 6$ :

- Initial value:  $\mathbf{q}^{(0)} \in \mathbb{R}^n$  (estimated approximation for the sought joint variable values, here  $n = 6$  assumed)
- Execution of iteration steps for  $k = 0, 1, 2, \dots$ :

$$(1) \text{ calculate Jacobian matrix } J_F(\mathbf{q}^{(k)}) = \begin{pmatrix} \frac{\partial F_1}{\partial q_1} & \dots & \frac{\partial F_1}{\partial q_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_6}{\partial q_1} & \dots & \frac{\partial F_6}{\partial q_6} \end{pmatrix} \text{ of } \mathbf{F}(\mathbf{q}^{(k)});$$

$$(2) \text{ solve linear equation system } J_F(\mathbf{q}^{(k)}) \cdot \delta^{(k)} = -\mathbf{F}(\mathbf{q}^{(k)}) \text{ for } \delta^{(k)} \in \mathbb{R}^6;$$

$$(3) \text{ set } \mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \delta^{(k)}.$$

- Termination when either

- $\|\mathbf{F}(\mathbf{q}^{(k)})\| \leq \varepsilon_1$  or
- $\|\delta^{(k)}\| \leq \varepsilon_2$  or
- $k = k_{\max}$ .

**Advantage:** Since very good algorithms are easily accessible in program libraries, an iterative calculation can be implemented with relatively little development effort.

**Disadvantages:**

- There is **no** iterative method that always **guarantees** to calculate **all** solutions of the inverse kinematics problem! Iterative methods for solving nonlinear problems are generally only locally convergent. That is, they usually calculate at most one solution. The basin of attraction of the calculated solution in the case of multiple theoretical solutions is unclear and depends among other things on the initial value  $\mathbf{q}^{(0)}$ : However, for many applications of inverse kinematics, all solutions of the IK and not just individual ones are needed.
- Iterative calculation methods are significantly more complex and much slower IK algorithms compared to explicit, closed-form solution formulas (see following sections): Even if the computation time of the method on a powerful computer would be far below one second, this would be of little use, because the inverse kinematic model must be implemented for many applications on a microcontroller or a similarly limited computing platform and deliver the required result within guaranteed computation time (real-time requirement).

- The convergence behavior of iterative calculation methods for nonlinear problems is often problematic, i.e. it cannot be predicted and possibly no solution can be calculated at all.
  - For example, the convergence behavior of the classical Newton method is problematic when  $J_F(\mathbf{q}^{(k)})$  is singular, i.e.  $\det(J_F) = 0$ : That is, an iterative Newton method terminates (without suitable modification) the calculation without providing a solution, although (at least) one may exist for an inverse kinematics problem.

Therefore, one tries to determine an explicit, analytical model for the IK of a manipulator *in closed form* from the outset if possible (see following sections 3.2.3.1 and 3.2.3.2), either for the entire IK or at least for as large a subsystem as possible. Only for the remainder is a numerical method then used.

### 3.2.2.2 Selection of Some Iterative Calculation Methods for Inverse Kinematics

**KDL** is the C++-written “Kinematics and Dynamics Library” of the Orococos project<sup>8</sup> (“Open Robot Control Software”). A connection to the Robot Operating System ROS is available<sup>9</sup>. For the numerical solution of inverse kinematics, an iterative Newton-Raphson algorithm is used. This is very easy to use for kinematic chains with 6 or more degrees of freedom in the widespread URDF<sup>10</sup> format, since one only needs to specify base and end link. The computation times are relatively high and solutions of inverse kinematics are not calculated very reliably.

**Drake** Within the framework of Drake<sup>11</sup>, an iterative algorithm for inverse kinematics is also available for many degrees of freedom, such as the whole-body posture of a humanoid robot. This is based on the formulation of the inverse kinematics problem as a nonlinear constrained optimization problem, which is iteratively solved using a sequential quadratic programming method. Due to this general approach, additional constraints, e.g. for maintaining the balance of a humanoid robot, can also be considered. In the considered applications of humanoid robots, sufficiently good approximations for the sought solutions can often be determined, so that the method can deliver meaningful solutions.

**Corke’s Robotics Toolbox** The Robotics and Vision Toolbox<sup>12</sup> for Matlab by P. Corke offers several iterative algorithms for numerical solution of the inverse kinematics problem (ikine6s, ikine, ikunc, ikcon).

---

<sup>8</sup>[www.orocos.org/kdl](http://www.orocos.org/kdl)

<sup>9</sup>[wiki.ros.org/kdl](http://wiki.ros.org/kdl)

<sup>10</sup>Unified Robot Description Format (URDF), [wiki.ros.org/urdf](http://wiki.ros.org/urdf)

<sup>11</sup>R. Tedrake: *Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems*, 2014, [drake.mit.edu](http://drake.mit.edu)

<sup>12</sup>[petercorke.com/Robotics\\_Toolbox.html](http://petercorke.com/Robotics_Toolbox.html)

### 3.2.3 Determination of an Explicit, Analytical Solution Formula for IK

**Advantage:** If an explicit, analytical solution formula for the inverse kinematics problem is available, all solutions can be determined in guaranteed computation time.

**Disadvantage:** Such explicit solution formulas do not always exist. If this is the case, they can often only be determined with great effort.

**Recommendation:** Wherever possible, explicit solution formulas should be determined, at least for the largest possible part of the solution.

Whether a closed-form IK exists for a 6-DOF manipulator depends on the (geometric) design! The kinematic structure of most industrial robots is also chosen so that an explicit formula for the IK can be given. In general, there are several typical arrangements of 6 revolute joints in an arm or robot leg with known or determinable solution formulas for the inverse kinematics problem.

Sufficient for the existence of a closed-form IK for a manipulator with six joints is that there are three adjacent joint axes that have a common intersection point (Pieper (1968), see [Craig, 1989], Section 4.6)<sup>13</sup>. This is the case for many industrial robots for axes 4, 5, and 6 (see e.g. the robot shown on page 7 in Section 1.4.2). This allows the IK for position  $(q_1, q_2, q_3)$  and orientation  $(q_4, q_5, q_6)$  to be decoupled and determined separately from each other (see also Chapter 4 in [Spong/Vidyasagar, 1989]).

---

<sup>13</sup>This also includes the case that three consecutive axes are parallel, i.e. intersect at a point at infinity.

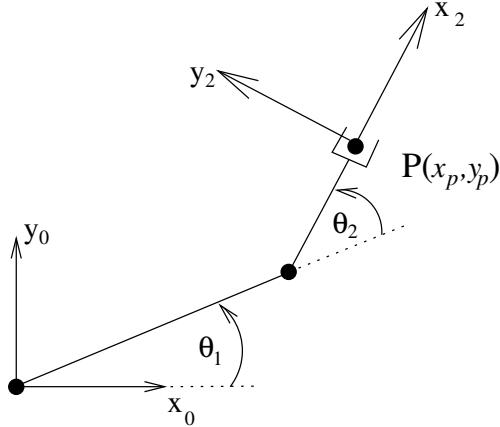
### 3.2.3.1 Algebraic Determination of IK

#### Procedure:

Use of the FK  ${}^0T_E(\mathbf{q})$  and “clever” transformations of the equations  ${}^0T_E(\mathbf{q}) = {}^0\tilde{T}_E$ , so that solving for  $\mathbf{q}$  is possible.

#### Example: Planar SCARA Manipulator

The forward kinematics model DIRKIN for the planar SCARA manipulator was described in Section 3.1.3, page 37 ff.



$$\underbrace{{}^0T_E(\mathbf{q})}_{=?} = \begin{pmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_1 \cdot \cos \theta_1 + l_2 \cdot \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_1 \cdot \sin \theta_1 + l_2 \cdot \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\underbrace{\begin{pmatrix} {}^0\mathbf{e}_{x_2} & {}^0\mathbf{e}_{y_2} & 0 & \begin{pmatrix} x_p \\ y_p \end{pmatrix} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{given}} = {}^0\tilde{T}_E$$

$\Rightarrow$  6 nonlinear equations for 2 unknowns  $\theta_1, \theta_2$ . Only solvable if 4 equations are redundant. (Clear, because with 2 DOF a point  $\in \mathcal{A}$  can just be reached, but not with arbitrary orientation.)

Sought:  $\mathbf{q} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$  as a function of  $\begin{pmatrix} x_p \\ y_p \end{pmatrix} \in \mathcal{A}$ :

One uses the (nonlinear) equations of the forward kinematics model and tries to solve them by hand for the joint variables  $\mathbf{q}$ :

$$\begin{aligned} x_p &= l_1 \cdot \cos \theta_1 + l_2 \cdot \cos(\theta_1 + \theta_2) \\ y_p &= l_1 \cdot \sin \theta_1 + l_2 \cdot \sin(\theta_1 + \theta_2) \end{aligned}$$

A transformation with trigonometric addition theorems yields in a few steps (see [Craig, 1989], Section 4.4), where the sign determines the elbow position (+  $\hat{=}$  down or -  $\hat{=}$  up):

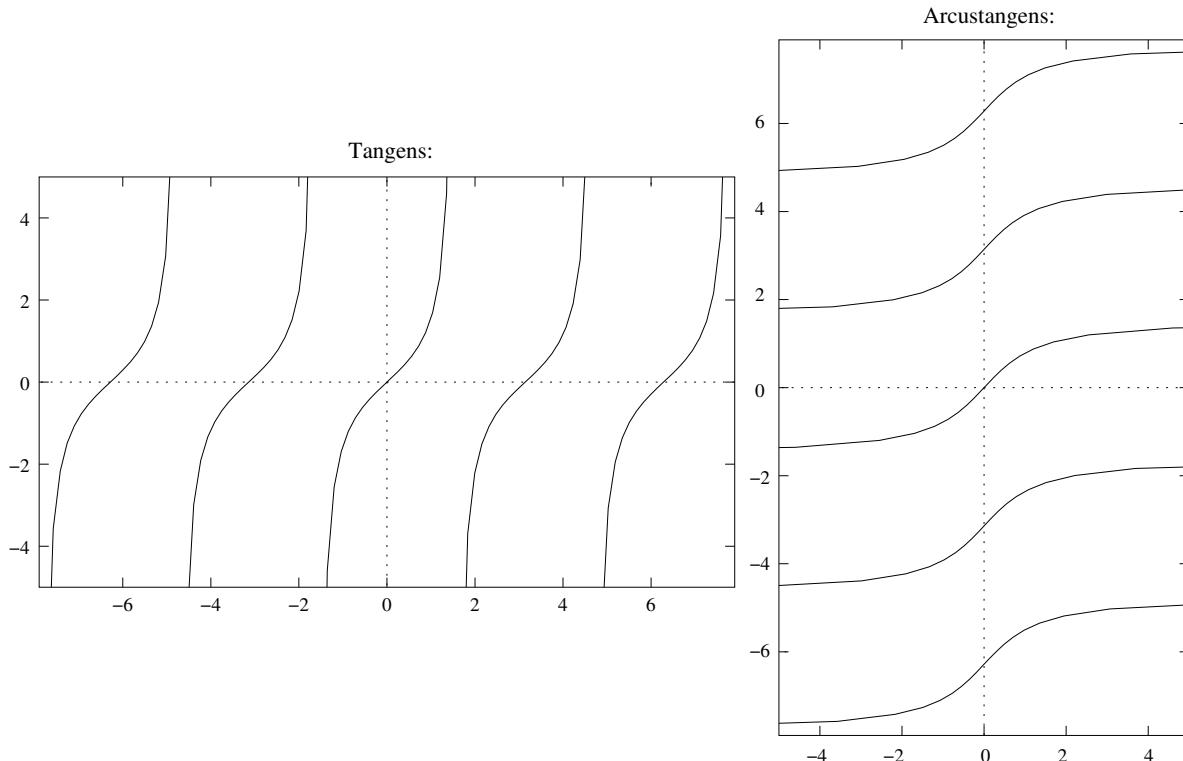
$$\begin{aligned} \cos \theta_2 &= (x_p^2 + y_p^2 - l_1^2 - l_2^2) / (2 \cdot l_1 \cdot l_2) \\ \sin \theta_2 &= \pm \sqrt{1 - (\cos \theta_2)^2} \end{aligned}$$

$\Rightarrow$  2 solutions for  $\theta_2$ :

$$\theta_2^{\pm} = \text{atan2}(\sin \theta_2, \cos \theta_2)$$

Where

- the just determined formulas are to be inserted for  $\sin \theta_2$  and  $\cos \theta_2$  and
- $\text{atan2}(y, x)$  denotes the arctangent function  $\arctan(\frac{y}{x}) = \tan^{-1}(\frac{y}{x})$  with quadrant information (cf. [Craig, 1989]: p.47 and [McKerrow, 1991]: p.199)



With the help of further transformations, one also obtains  $\theta_1$  (see [Craig, 1989], Section 4.4):

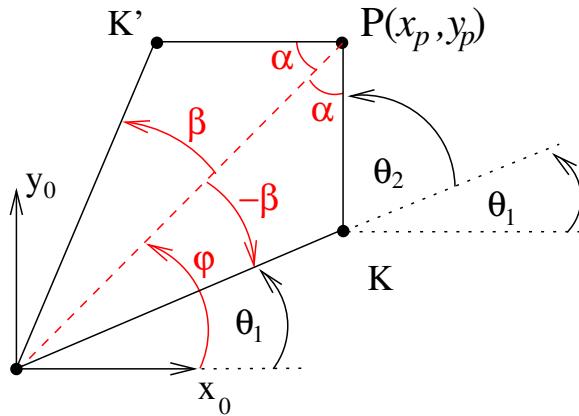
$$\theta_1^{\pm} = \text{atan2}(y_p, x_p) - \text{atan2}(l_2 \cdot \sin \theta_2, l_1 + l_2 \cdot \cos \theta_2).$$

### 3.2.3.2 Geometric Determination of IK

#### Procedure:

Determination of the IK *without* using the FK, solely from geometric considerations (i.e. individually for each robot), e.g. by decomposing the spatial geometry of the manipulator into several planar trigonometric problems.

#### Example: Planar SCARA Manipulator



After introducing the auxiliary angles  $\varphi$ ,  $\alpha$ , and  $\beta$  shown in the figure, the sought joint angles  $(\theta_1, \theta_2)$  and  $(\theta'_1, \theta'_2)$  for the two manipulator configurations with elbow “up” or “down” can be expressed by

$$\begin{array}{ll} \text{“Elbow down”} & \text{“Elbow up”} \\ \theta_1 = \varphi - \beta & \theta'_1 = \varphi + \beta \\ \theta_1 + \theta_2 = \varphi + \alpha & \theta'_1 + \theta'_2 = \varphi - \alpha \end{array}$$

with

$$\varphi = \text{atan2}(y_p, x_p)$$

After some transformations using theorems of planar trigonometry, the auxiliary angles are given by

$$\alpha = 2 \cdot \arctan \left( \frac{b}{a - l_1} \right)$$

$$\beta = 2 \cdot \arctan \left( \frac{b}{a - l_2} \right)$$

with  $a - l_{1,2} > 0$ ,  $b > 0$  and  $0 \leq \alpha, \beta \leq \pi$  and the abbreviations

$$a = \frac{1}{2} \left( l_1 + l_2 + \sqrt{x_p^2 + y_p^2} \right) \quad (\text{half perimeter of triangles } 0KP \text{ and } 0K'P)$$

$$b = \sqrt{\frac{(a - l_1)(a - l_2)(a - \sqrt{x_p^2 + y_p^2})}{a}} \quad (\text{incircle radius of triangles } 0KP \text{ and } 0K'P).$$

### 3.2.3.3 Algorithmic Determination of the Analytical IK

Algorithmic solution methods for determining a closed-form solution formula, e.g. in the form of program code, for the inverse kinematics problem have made great progress in recent years. It is certainly worthwhile to investigate their application for a specific problem.

**Advantage:** If a solution formula can be determined, it enables much faster and more reliable computation than iterative calculation methods and in a guaranteed computation time important for real-time applications.

**Disadvantage:** It cannot be guaranteed that an algorithmic determination can always determine an explicit solution when one exists.

**Matlab Symbolic Math Toolbox** can help to determine an explicit analytical solution formula.

**ikfast** Far faster in computation than iterative methods like kdl is ikfast, which is part of the OpenRAVE project<sup>14</sup>. A ROS interface is also available<sup>15</sup>. Based on a chain in a URDF robot model, it attempts to automatically generate the inverse kinematics in closed form as a C++ header. If this is successful, the resulting numerical evaluation is by far faster than with iterative methods. However, not only a complex robot kinematics can lead to generation failure or missing solutions at runtime. ikfast also had difficulties in practical use even with 4- and 5-jointed robot arms.

**Others** Several methods for the symbolic solution of the inverse kinematics problem are described in the technical literature. However, these are often not available as directly usable code (either as open or closed source).

---

<sup>14</sup>[openrave.org/docs/0.8.0/openravepy/ikfast/](http://openrave.org/docs/0.8.0/openravepy/ikfast/)

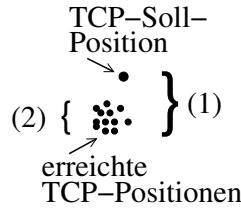
<sup>15</sup>[wiki.ros.org/Industrial/Tutorials/Create\\_a\\_Fast\\_IK\\_Solution/moveit\\_plugin](http://wiki.ros.org/Industrial/Tutorials/Create_a_Fast_IK_Solution/moveit_plugin)

### 3.3 Accuracy of the Kinematic Model

Three criteria for evaluating the accuracy of the positionability of a real manipulator with respect to the tool center point (usually TCP = origin of  $S_n$ ).

(1) *Position accuracy* (accuracy):

Difference between actual and specified position of the TCP.



(2) *Repeatability* (repeatability):

Variations in position accuracy with multiple repetitions of TCP movements to the same point.

(3) *Resolution* (resolution):

is the smallest possible distance over which the TCP can be moved with certainty.

For most robots, (2) repeatability and (3) resolution are much better than (1) position accuracy. Why? There are two types of sources of inaccuracy in the kinematic model: a) geometric and b) non-geometric parameters. Both can vary due to manufacturing but also due to operation. For example, there are noticeable parameter fluctuations due to thermal effects in the transient warm-up phase between robot standstill and continuous operation.

a) *Geometric parameters* are the link parameters of the kinematic model. Due to manufacturing tolerances, inaccuracies can occur in the dimensions of the robot links and the orientations of the axes (up to  $0.5^\circ$ ) as well as internal sensors.

⇒ Inaccuracies in forward and inverse kinematics solutions.

An improvement in the accuracy of the geometric parameters specified by the manufacturer can be achieved through calibration based on measurements with parameter estimation methods. The robot does not need to be disassembled<sup>16</sup>:

- With the TCP, a set of test poses  ${}^0\tilde{T}_n^i \in \mathbb{R}^{4 \times 4}, i = 1, \dots, m$  with joint positions  $\tilde{\mathbf{q}}^i \in \mathbb{R}^n, i = 1, \dots, m$  are approached,
- Using a suitable optimization method, the geometric parameters of the manipulator links  $\mathbf{f}$  are calculated such that the sum of squared deviations  $\Phi$

$$\Phi(\mathbf{f}) = \sum_{i=1}^m \underbrace{\| {}^0\tilde{T}_n^i - {}^0T_n^i(\tilde{\mathbf{q}}^i, \mathbf{f}) \|^2}_{(*)} \quad (*) \text{ sum of squares of elements of the difference matrix}$$

is minimized.

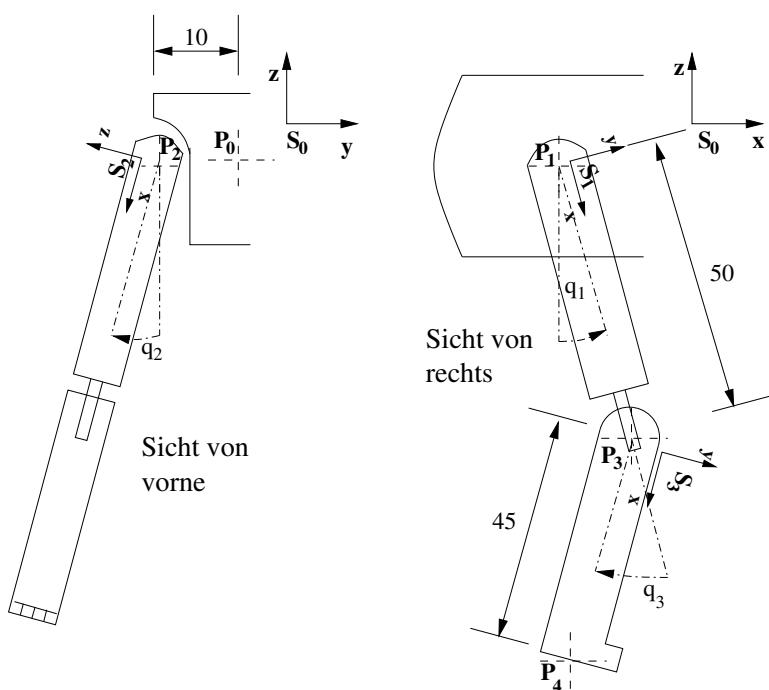
b) *Non-geometric* parameters are not considered in the kinematic model. These include, for example:

- backlash, friction, elasticities and eccentricities occurring in the gearboxes,
- temperature-dependent expansions,
- the measurement accuracy of the joint-internal position sensors.

<sup>16</sup>See R. Bernhardt, S.L. Albright (eds.): *Robot Calibration* (London: Chapman & Hall, 1994).

## 3.4 Kinematic Modeling of Robot Legs and Arms

### 3.4.1 Kinematics of a Three-Jointed Leg for a Quadruped Robot



Possibly see exercise.

### 3.4.2 Kinematics of a Six-Jointed Leg for a Humanoid Robot

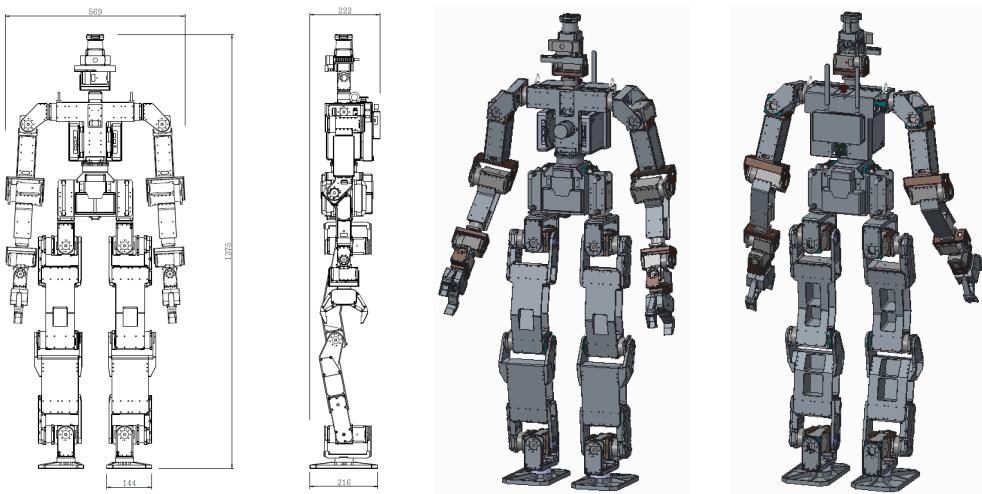


Figure 3.1: Model of the 142 cm tall humanoid robot THORMANG3.

In recent decades, significant progress has been made in humanoid robotics, particularly in the area of humanoid walking. In recent years, the availability of human-sized robots such as the THORMANG3 (Figure 3.1) has also increased, some of which can even walk stably on difficult uneven surfaces.

Most walking humanoid robots have a relatively similar kinematic leg structure consisting of six actuated revolute joints per leg: three mutually orthogonal revolute joints in the hip (hip joints), one revolute joint in the knee (knee joint) and two mutually orthogonal ankle joints (ankle joints). Two examples are shown in Figures 3.2 and 3.3.

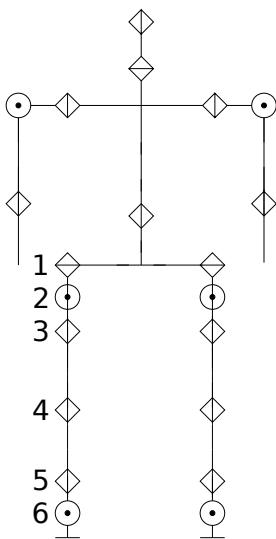


Figure 3.2: Kinematic structure of the 21 revolute joints of the approx. 57 cm tall, autonomous humanoid robot Bruno of the Darmstadt Dribblers.

(Source: <http://www.dribblers.de>; M. Friedmann, J. Kiener, S. Petters, H. Sakamoto, D. Thomas, O. von Stryk: Versatile, high-quality motions and behavior control of a humanoid soccer robot In: International Journal of Humanoid Robotics, Vol. 5, No. 3, pp. 417-436, September, 2008.

The **inverse kinematics problem** for the six-jointed robot leg can be solved explicitly due to the special arrangement of the joints. Also Pieper's criterion, page 45, for the existence

of an IK can be applied, since the successive joints 3, 4 and 5 (see Figure 3.2 right) have a common point of intersection (at infinity).

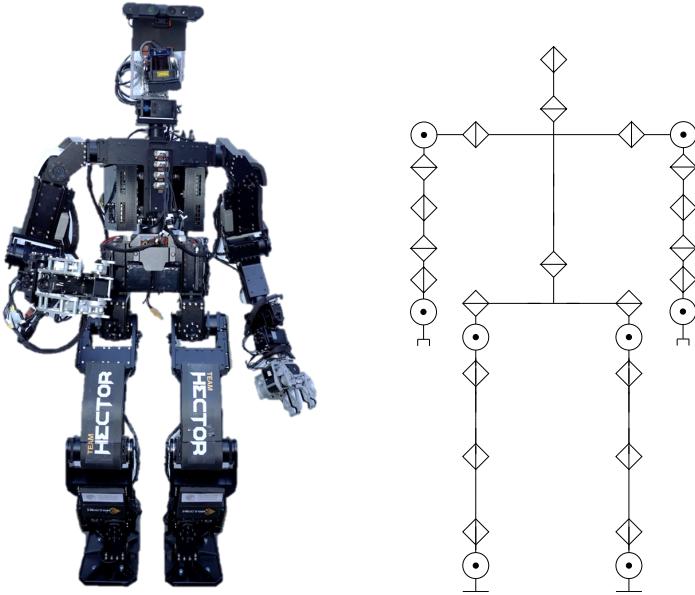


Figure 3.3: Team Hector's Johnny #5 ([www.teamhector.de](http://www.teamhector.de)) is a modified version of the THORMANG3. Its kinematic structure consists of a total of 29 joints without finger links.

The determination of the solution of the inverse kinematics problem for the robot leg (Figure 3.2) is done in two steps:

1. Solution of the inverse kinematics problem for the three-jointed, planar arm consisting of joints 3, 4 and 5 (i.e. 3rd hip joint, knee joint and 1st ankle joint),
2. Determination of the first, second and sixth joint, so that the middle three joints lie in a plane.

Behind both procedural steps lies a certain computational effort. The approach can be transferred not only to the joint arrangement shown in Figure 3.2 but also to other arrangements, as long as three successive, middle revolute joints lie in a plane.

### 3.4.3 Kinematics of an Arm for a Humanoid Robot

The ability to interact with the environment requires a complex arm kinematics. For arbitrary positioning and orientation of the end-effector in the workspace, a kinematics with at least 6 degrees of freedom is needed. When considering the human arm, a total of 7 degrees of freedom can even be identified: 3 in the shoulder, 1 in the elbow and 3 in the wrist. Therefore, anthropomorphic robot arms, such as Johnny #5 (see Figure 3.3), are typically also equipped with at least 7 joints, allowing for redundant joint configurations.

This redundancy is extremely helpful in practical applications, as manipulation tasks frequently induce additional kinematic constraints. A typical example is closing or opening a valve, as illustrated in Figure 3.4. Due to the fixed object, the end-effector loses degrees of freedom during physical contact with the valve wheel. If additionally the arm enters a singular configuration (see Chapter 4.4.2) or joint angle limitation, further losses of degrees of freedom can occur, so that the end-effector may no longer be able to move in the desired directions.

Therefore, for manipulation tasks it can be helpful to be able to involve as many degrees of freedom as possible. In the example from Figure 3.4, a total of 8 joints (7 in the arm, 1 in the torso) are used.

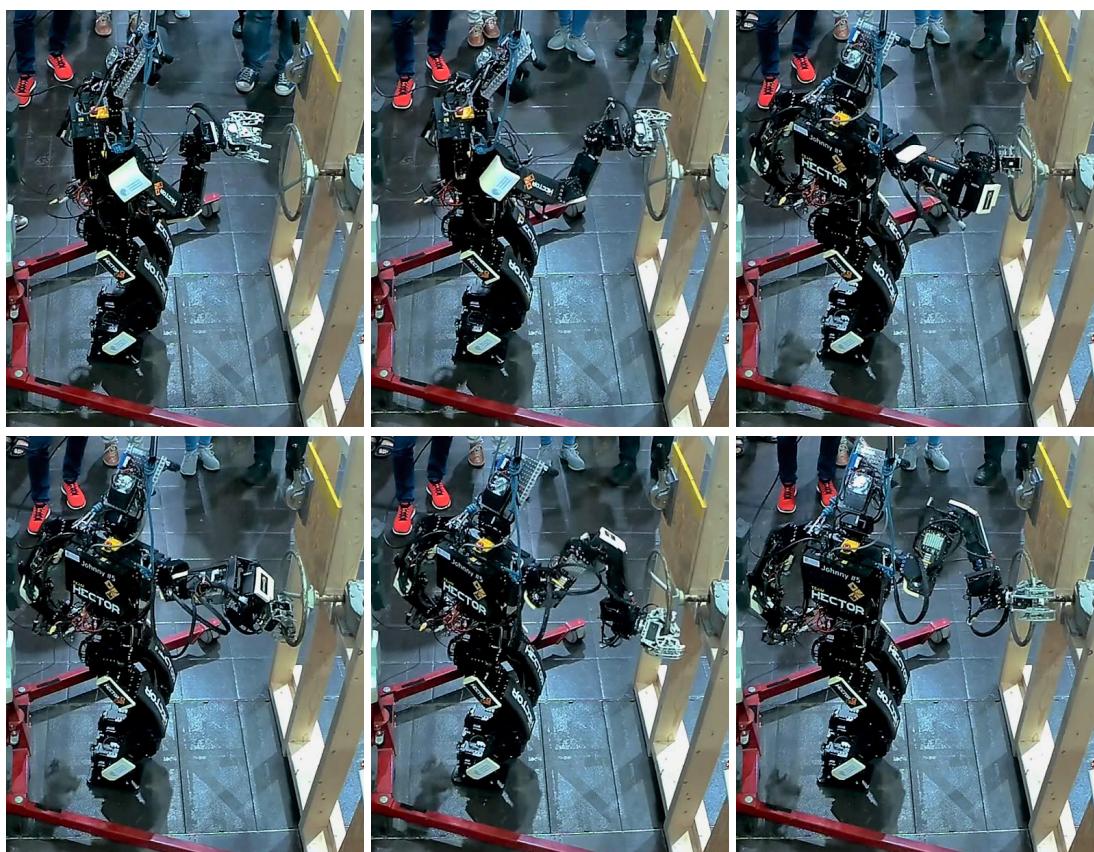


Figure 3.4: Johnny #5 executing a complex arm trajectory for turning a valve wheel. A total of 8 joints are used here: 7 from the arm and 1 from the torso.

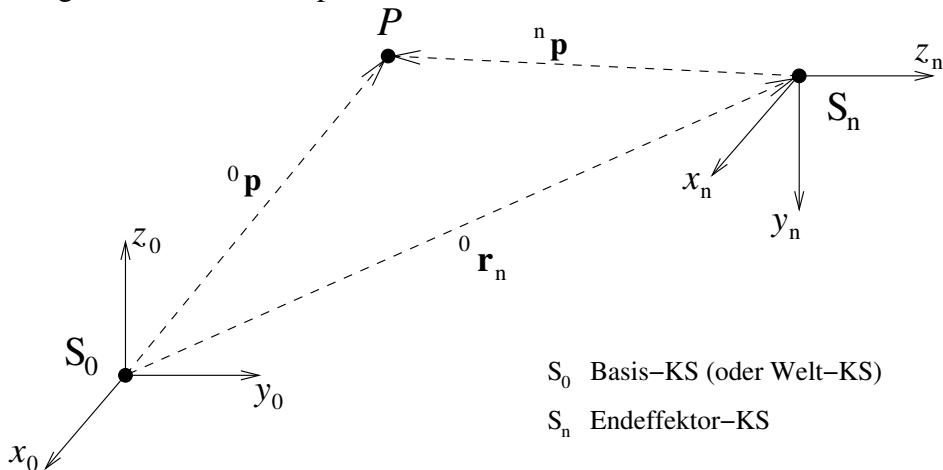
# Chapter 4

## Velocity, Jacobian Matrix (and Static Forces)

The velocities examined in Chapter 4 are still part of the "kinematics" of robots. Only in the case of open kinematic chains can (as discussed in Chapter 3) position and orientation be calculated independently of linear and angular velocities.

### 4.1 Introduction

**So far:** Investigation of individual positions:



The FK and IK correspond to equations that relate the position of the joint variables  $\mathbf{q} \in \mathbb{R}^n$  with the position and orientation of the end-effector  ${}^0T_n$ :

$$\mathbf{q} \xrightleftharpoons[\text{IK}]{\text{FK}} {}^0T_n = \left( \begin{array}{c|c} {}^0R_n & {}^0\mathbf{r}_n \\ \hline \mathbf{0}^T & 1 \end{array} \right)$$

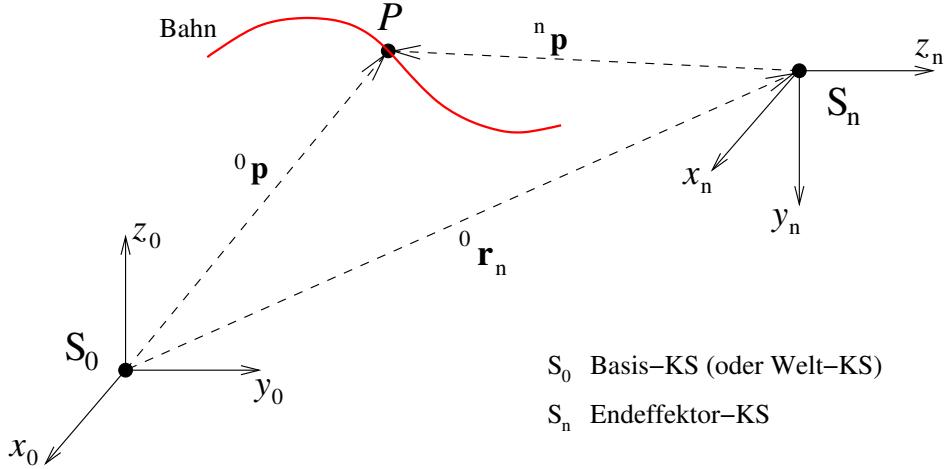
The coordinates  ${}^n\mathbf{p} \in \mathbb{R}^3$  of a point  $P$  w.r.t.  $S_n$  are transformed into coordinates w.r.t.  $S_0$  by

$$\left( \begin{array}{c} {}^0\mathbf{p} \\ 1 \end{array} \right) = {}^0T_n(\mathbf{q}) \cdot \left( \begin{array}{c} {}^n\mathbf{p} \\ 1 \end{array} \right), \quad \mathbf{p} = \left( \begin{array}{c} p_x \\ p_y \\ p_z \end{array} \right)$$

Conversely, it holds

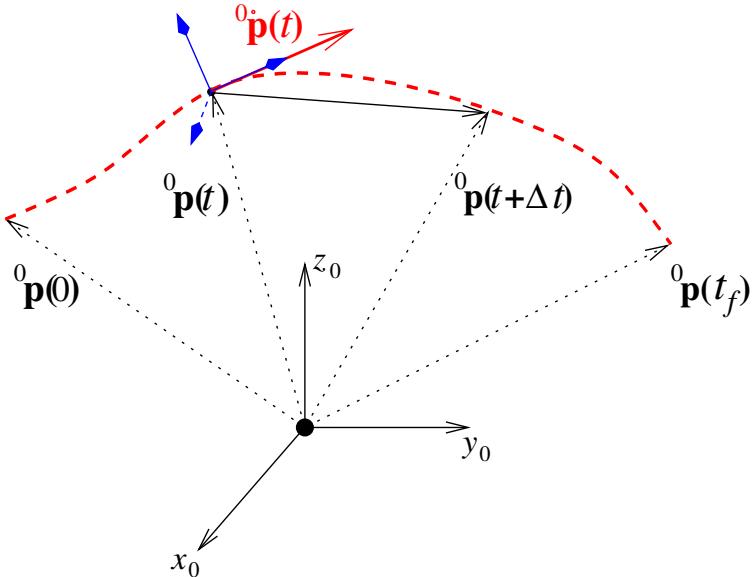
$$\begin{pmatrix} {}^n\mathbf{p} \\ 1 \end{pmatrix} = ({}^0T_n(\mathbf{q}))^{-1} \cdot \begin{pmatrix} {}^0\mathbf{p} \\ 1 \end{pmatrix} = {}^nT_0(\mathbf{q}) \cdot \begin{pmatrix} {}^0\mathbf{p} \\ 1 \end{pmatrix}$$

**Now:** also consideration of the temporal progression of a motion:



**Definition 4.1** A path (or path or way, engl. path) is a continuous, spatial sequence of points, which is traversed, e.g., by the TCP during a motion.

**Definition 4.2** A trajectory (engl. trajectory) is a path with temporal progression, i.e. position, velocity and acceleration are given at each point of the path in space.



$$\lim_{\substack{\Delta t \rightarrow 0 \\ \Delta t > 0}} \frac{1}{\Delta t} ({}^0\mathbf{p}(t + \Delta t) - {}^0\mathbf{p}(t)) = \frac{d}{dt} {}^0\mathbf{p}(t) = {}^0\dot{\mathbf{p}}(t) = \begin{pmatrix} {}^0\dot{p}_x(t) \\ {}^0\dot{p}_y(t) \\ {}^0\dot{p}_z(t) \end{pmatrix}$$

The quantity  ${}^0\dot{\mathbf{p}} \in \mathbb{R}^3$  is the **velocity vector** of a point  $P$  with coordinate vector  $({}^0\mathbf{p}(t))$  at time  $t$ , i.e. the tangent vector of the trajectory  $\{{}^0\mathbf{p}(\tilde{t}) \in \mathbb{R}^3 : 0 \leq \tilde{t} \leq t_f\}$  at  $P(t)$  ( $t$ :

time,  $t_f$ : final time). Thereby  ${}^0\dot{\mathbf{p}}$  depends on the reference coordinate system (here  $S_0$ ). The length  $\|{}^0\dot{\mathbf{p}}\|$  of the velocity vector is the absolute velocity and does not change when changing the coordinate system (e.g. from  $S_0$  to  $S_n$ ) (as long as  $S_n$  is fixed w.r.t.  $S_0$ ).

⇒ The velocity vector  ${}^0\dot{\mathbf{p}}$  of a point on a trajectory is always **relative** to the respective reference coordinate system (here  $S_0$ )!

Analogously, the **acceleration vector**  ${}^0\ddot{\mathbf{p}} \in \mathbb{R}^3$  of a point mass  $P$  at time  $t$  can also be calculated from the velocity change:

$$\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} ({}^0\dot{\mathbf{p}}(t + \Delta t) - {}^0\dot{\mathbf{p}}(t)) = \frac{d}{dt} {}^0\dot{\mathbf{p}}(t) = {}^0\ddot{\mathbf{p}}(t)$$

In many applications, the HTM of the end-effector of a manipulator should follow a prescribed trajectory  $\{{}^0\tilde{T}_E(t) : 0 \leq t \leq t_f\}$  with prescribed velocity  $\{{}^0\dot{\tilde{T}}_E(t) : 0 \leq t \leq t_f\}$  in space.

I.e. not only the corresponding path in joint variables  $\mathbf{q}(t) \in \mathbb{R}^n$ ,  $0 \leq t \leq t_f$ , but also the corresponding velocity  $\frac{d}{dt}\mathbf{q}(t) = \dot{\mathbf{q}}(t) \in \mathbb{R}^n$  (and possibly acceleration  $\frac{d^2}{dt^2}\mathbf{q}(t) = \ddot{\mathbf{q}}(t) \in \mathbb{R}^n$ ) must be calculated.

⇒ Kinematic models for velocity (and acceleration) are also needed!

The relations between the velocities of the joint variables  $\dot{\mathbf{q}}(t) \in \mathbb{R}^n$  and the velocities of the Cartesian coordinates of the HTM  ${}^0\dot{T}_n(\mathbf{q}(t))$  (or  ${}^0\dot{\mathbf{r}}_n(\mathbf{q}(t))$  and  ${}^0\dot{R}_n(\mathbf{q}(t))$ ):

$$\dot{\mathbf{q}} \xrightleftharpoons[?]{} {}^0\dot{T}_n = \begin{pmatrix} {}^0\dot{R}_n & {}^0\dot{\mathbf{r}}_n \\ \mathbf{0}^T & 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix}, {}^0\dot{\mathbf{r}}_n$$

are determined by the so-called **Jacobian matrix of the manipulator** (engl. Manipulator Jacobian)  ${}^0J_n(\mathbf{q})$  determined<sup>1</sup>, as will be shown soon.

The Jacobian matrix of a manipulator is needed for many tasks, e.g.

- in the planning (see Chapter 8) and execution (see Chapter 6) of "uniform, smooth" (smooth) trajectories,
- in the determination of so-called singular robot configurations (see Chapter 4.4.2),
- in the transformation of forces and moments that occur when grasping the end-effector to the actuated joints,
- in the derivation of the dynamic equations of motion for robots (see Chapter 5).

---

<sup>1</sup>Please do not confuse the "Jacobian matrix of a manipulator" with the "normal" Jacobian matrix of a (vector-valued) function of several independent variables from analysis!

Since the velocity vector depends on the reference coordinate system, we first investigate how the velocities of position  ${}^n\dot{\mathbf{p}}(t)$  and orientation  ${}^0R_n(\mathbf{q}(t))$  transform when changing the reference coordinate system.

**1st attempt** to determine the transformation relationship for the velocities of position and orientation (with respect to different CSs).

It is obvious to use the transformation relationship for point coordinates developed in Chapters 2 and 3 when changing from coordinate system  $S_n$  to  $S_0$

$$\boxed{T} \quad \underbrace{{}^0\mathbf{p}(t)}_{\in \mathbb{R}^3} = \underbrace{{}^0\mathbf{r}_n(\mathbf{q}(t))}_{\in \mathbb{R}^3} + \underbrace{{}^0R_n(\mathbf{q}(t))}_{\in \mathbb{R}^{3 \times 3}} \cdot \underbrace{{}^n\mathbf{p}(t)}_{\in \mathbb{R}^3}$$

to consider it as time-dependent via the time-dependent joint variables  $\mathbf{q}(t)$  and to differentiate it totally with respect to time  $t$ . Thereby  $(\dots)^\cdot = \frac{d}{dt}(\dots)$  denotes the total derivative with respect to time  $t$  and the product rule is applied:

$${}^0\dot{\mathbf{p}}(t) = \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)^\cdot + \left({}^0R_n(\mathbf{q}(t))\right)^\cdot \cdot {}^n\mathbf{p}(t) + {}^0R_n(\mathbf{q}(t)) \cdot {}^n\dot{\mathbf{p}}(t)$$

Thereby denote:

- ${}^0\dot{\mathbf{p}}(t)$  the velocity vector of point P w.r.t.  $S_0$  (exactly:  ${}^0\dot{\mathbf{p}}(t) := {}^0\dot{\mathbf{p}}(\mathbf{q}(t))$ , since  ${}^0\mathbf{p}$  depends on  $t$  via  $\mathbf{q}$ ) and
- ${}^n\dot{\mathbf{p}}(t)$  the velocity vector w.r.t.  $S_n$ .

Normally the coordinates of point P in the end-effector coordinate system describe a reference point P of a workpiece or object that is **rigidly connected** to the end-effector. Then it holds  ${}^n\mathbf{p}(t) = \text{const}$  (e.g. = 0 for TCP, if it is the origin of  $S_n$ )  $\Rightarrow$   ${}^n\dot{\mathbf{p}}(t) = \mathbf{0}$  as a rule.

Applying the chain rule to the above formula yields:

$${}^0\dot{\mathbf{p}}(t) = \left[ \underbrace{\left( \underbrace{\frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial \mathbf{q}}}_{(3 \times n)\text{-matrix}, \text{(normal) Jacobian matrix of } {}^0\mathbf{r}_n(\mathbf{q})} \right) \cdot \underbrace{\dot{\mathbf{q}}(t)}_{\in \mathbb{R}^n}}_{(*1)} + \underbrace{\left( \underbrace{\frac{\partial}{\partial \mathbf{q}} \underbrace{{}^0R_n(\mathbf{q})}_{(3 \times 3)\text{-matrix}}}_{(3 \times 3 \times n)\text{-"three-dimensional matrix"} \text{ (3rd order tensor)}} \right) \cdot \underbrace{\dot{\mathbf{q}}(t)}_{\in \mathbb{R}^n}}_{(*2)} \cdot {}^n\mathbf{p}(t) + \underbrace{{}^0R_n(\mathbf{q}(t)) \cdot \underbrace{{}^n\dot{\mathbf{p}}(t)}_{\in \mathbb{R}^3}}_{(3 \times 3)\text{-matrix}}$$

Thereby the individual terms describe

- (\*) the relationship of the (linear) velocity of the *position* of the origin of  $S_n$  (w.r.t.  $S_0$ ) and the velocities of the joint variables  $\mathbf{q}$  and
- (\*\*) the relationship of the (angular) velocity of the *orientation* of  $S_n$  (w.r.t.  $S_0$ ) and the velocities of the joint variables  $\mathbf{q}$ .

⇒ The result of this first consideration is a very complicated and unwieldy representation!

Isn't there another way?

One can try to take special properties into account. For example, the orthogonality of  ${}^0R_n(\mathbf{q})$  has not yet been used.

**2nd attempt:** The orthogonality relation

$$R(\mathbf{q}(t)) \cdot R(\mathbf{q}(t))^T = E \quad 0 \leq t \leq t_f$$

is differentiated totally with respect to time  $t$ , with the product rule it follows ( $\dot{(.)} = d(./)dt$ ):

$$\underbrace{(R(\mathbf{q}(t)))\dot{\cdot} R^T(\mathbf{q}(t))}_{=:B} + \underbrace{R(\mathbf{q}(t))\cdot(R(\mathbf{q}(t)))\dot{\cdot}^T}_{=:B^T} = O \quad (\in \mathbb{R}^{3 \times 3})$$

Setting

$$B := (R(\mathbf{q}(t)))\dot{\cdot} R^T(\mathbf{q}(t)) \iff \boxed{R(\mathbf{q}(t))\dot{\cdot} = B \cdot R(\mathbf{q}(t))}$$

then one obtains the derivative of a rotation matrix by multiplication with a suitable, skew-symmetric matrix  $B$

$$B + B^T = O.$$

Due to the definition of  $B$  as  $B = (R(\mathbf{q}(t)))\dot{\cdot} R^T(\mathbf{q}(t))$ ,  $B$  must implicitly depend on  $\mathbf{q}(t)$  and  $\dot{\mathbf{q}}(t)$ , i.e.  $B = B(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ .

## 4.2 Skew-Symmetric Matrices, Vectors of Angular Velocities and Accelerations

**Definition 4.3** A  $(3 \times 3)$ -matrix  $B$  is called **skew-symmetric** (engl. skew symmetric)

$$\iff B^T + B = O = B + B^T.$$

**Consequence:**

$$\begin{aligned} b_{ij} + b_{ji} &= 0 \quad i, j = 1, 2, 3 \\ \Rightarrow b_{ij} &= -b_{ji} \text{ and } b_{ii} = 0 \end{aligned}$$

$$\Rightarrow B = \begin{pmatrix} 0 & -b_{2,1} & -b_{3,1} \\ b_{2,1} & 0 & -b_{3,2} \\ b_{3,1} & b_{3,2} & 0 \end{pmatrix}$$

A skew-symmetric matrix  $B$  has only three independent entries, which can be called  $a_x, a_y, a_z$  instead of  $b_{2,1}, b_{3,1}, b_{3,2}$ .

With  $\mathbf{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} b_{3,2} \\ -b_{3,1} \\ b_{2,1} \end{pmatrix}$  one defines the matrix  $B(\mathbf{a})$  as

$$B(\mathbf{a}) := \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}$$

**Further properties:**

(i) For  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$  and  $\alpha, \beta \in \mathbb{R}$  it holds:

$$B(\alpha \cdot \mathbf{a} + \beta \cdot \mathbf{b}) = \alpha \cdot B(\mathbf{a}) + \beta \cdot B(\mathbf{b})$$

(ii) For all  $\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \in \mathbb{R}^3$  it holds:

$$B(\mathbf{a}) \cdot \mathbf{p} = \mathbf{a} \times \mathbf{p} \quad \left[ = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} a_y p_z - a_z p_y \\ a_z p_x - a_x p_z \\ a_x p_y - a_y p_x \end{pmatrix} \right]$$

(iii) First, for rotation matrices  $R \in \mathbb{R}^{3 \times 3}$  it holds:

$$R \cdot (\mathbf{a} \times \mathbf{b}) = (R \cdot \mathbf{a}) \times (R \cdot \mathbf{b}) \quad (*)$$

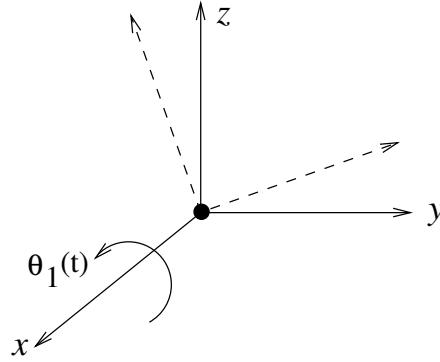
Furthermore, for arbitrary  $\mathbf{b} \in \mathbb{R}^3$  it holds:

$$\begin{aligned} R \cdot B(\mathbf{a}) \cdot R^T \cdot \mathbf{b} &= R \cdot (\mathbf{a} \times R^T \mathbf{b}) && \text{(ii)} \\ &= (R\mathbf{a}) \times (RR^T \mathbf{b}) && \text{(*)} \\ &= (R\mathbf{a}) \times \mathbf{b} && RR^T = E \\ &= B(R\mathbf{a}) \cdot \mathbf{b} && \text{(ii)} \end{aligned}$$

$$\Rightarrow \text{(iii)} \quad \boxed{R \cdot B(\mathbf{a}) \cdot R^T = B(R\mathbf{a})}$$

**Example:** Elementary rotations about the  $x$ -axis, i.e.  $n = 1$  and  $\mathbf{q} = (q_1) = (\theta_1(t))$

$$R(\mathbf{q}) = R(x; \theta_1(t)) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1(t) & -\sin \theta_1(t) \\ 0 & \sin \theta_1(t) & \cos \theta_1(t) \end{pmatrix}$$



$$\begin{aligned} \Rightarrow B(\mathbf{q}(t), \dot{\mathbf{q}}(t)) &= (R(\mathbf{q}(t))) \cdot R^T(\mathbf{q}(t)) \\ &= \left( \frac{\partial}{\partial \mathbf{q}} (R(\mathbf{q}(t))) \cdot \dot{\mathbf{q}}(t) \right) \cdot R^T(\mathbf{q}(t)) \\ &= \left( \frac{\partial}{\partial \theta_1} (R(x; \theta_1(t))) \cdot \dot{\theta}_1(t) \right) \cdot R^T(x; \theta_1(t)), \quad \text{since here } n = 1 \text{ and } \mathbf{q} = (q_1) = (\theta_1) \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin \theta_1 & -\cos \theta_1 \\ 0 & \cos \theta_1 & -\sin \theta_1 \end{pmatrix} \cdot \dot{\theta}_1 \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & \sin \theta_1 \\ 0 & -\sin \theta_1 & \cos \theta_1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \dot{\theta}_1(t) \\ &= B(\omega) \quad \text{with} \quad \omega = \omega(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} \dot{\theta}_1 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

Thereby  $\dot{\theta}_1 = \dot{q}_1$  is the angular velocity of the rotation about the  $x$ -axis.

Analogously, one obtains for

$$R(y; \theta_2) \Rightarrow B(\omega) = B \begin{pmatrix} 0 \\ \dot{\theta}_2 \\ 0 \end{pmatrix}$$

$$R(z; \theta_3) \Rightarrow B(\omega) = B \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{pmatrix}$$

**Generalization:** The rotation of a coordinate system about the rotation vector  $\mathbf{a}(t) \in \mathbb{R}^3$  by the rotation angle  $\phi(t)$  is described by the general rotation matrix  $R(\mathbf{q}(t)) = R(\mathbf{a}(t); \phi(t))$  (see A.2).

The angular velocity  $\dot{\phi}(t)$  at which the coordinate system rotates about  $\mathbf{a}(t)$  determines the rate of orientation change  $\dot{R}(t)$  according to the angular velocity vector  $\omega(t) = \mathbf{a}(t) \cdot \dot{\phi}(t)$  at time  $t$ .

Thus one obtains:

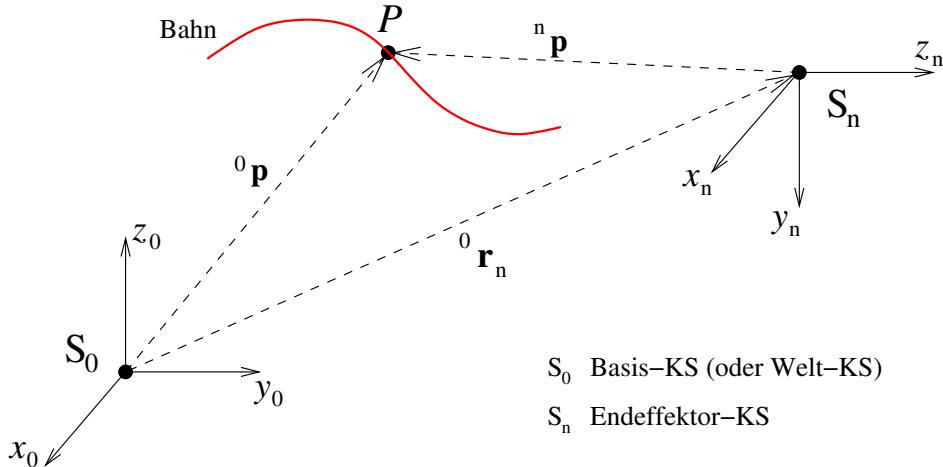
$$\frac{d}{dt}(R(\mathbf{q}(t))) = (R(\mathbf{q}(t)))\dot{\cdot} = B(\omega(t)) \cdot R(\mathbf{q}(t))$$

whereby the matrix  $B$  is skew-symmetric and can be represented as  $B(\omega(t))$  with unique

$$\omega(t) = \omega((\mathbf{q}(t)), \dot{\mathbf{q}}(t)) = \begin{pmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{pmatrix}$$

The vector  $\omega \in \mathbb{R}^3$  is the **angular velocity vector** of the rotating coordinate system with respect to the fixed reference coordinate system at time  $t$ . How  $\omega$  depends on  $\mathbf{q}(t)$  and  $\dot{\mathbf{q}}(t)$  in general will be investigated in the further course of Chapter 4.

**Consequence:**



$$[\text{T}] \quad {}^0\mathbf{p}(t) = {}^0\mathbf{r}_n(\mathbf{q}(t)) + {}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)$$

$$[\dot{\text{T}}] \quad {}^0\dot{\mathbf{p}}(t) = \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)\dot{\cdot} + \left({}^0R_n(\mathbf{q}(t))\right)\cdot {}^n\mathbf{p}(t) + {}^0R_n(\mathbf{q}(t)) \cdot {}^n\dot{\mathbf{p}}(t)$$

$$\text{where} \quad \left({}^0R_n(\mathbf{q}(t))\right)\cdot {}^n\mathbf{p}(t) = B({}^0\omega_n(t)) \cdot \underbrace{{}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)}_{= {}^0\mathbf{p}(t) - {}^0\mathbf{r}_n(\mathbf{q}(t))}$$

$$\text{(according to Chap. 4.2 (ii), p. 60)} = {}^0\omega_n(t) \times \underbrace{({}^0\mathbf{p}(t) - {}^0\mathbf{r}_n(\mathbf{q}(t)))}_{= {}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)}$$

$$\Rightarrow [\dot{\text{T}}] \quad {}^0\dot{\mathbf{p}}(t) = \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)\dot{\cdot} + {}^0\omega_n(t) \times \underbrace{\left({}^0\mathbf{p}(t) - {}^0\mathbf{r}_n(\mathbf{q}(t))\right)}_{= {}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)} + {}^0R_n(\mathbf{q}(t)) \cdot {}^n\dot{\mathbf{p}}(t)$$

## Relative Acceleration between $S_n$ and $S_0$

The considerations for relative acceleration are only relevant for Chapter 5.

**Investigation of the second derivative of  $\boxed{T}$  for the case  ${}^n\mathbf{p}(t) = \mathbf{const}$  ( $\Rightarrow {}^n\dot{\mathbf{p}}(t) = \mathbf{0}$ ):**

$$\begin{aligned} {}^0\ddot{\mathbf{p}}(t) &= \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)\ddot{\ } + \left({}^0\omega_n(t) \times \left({}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)\right)\right)\cdot \\ &\quad (\text{cross product satisfies product rule for derivatives}) \\ &= \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)\ddot{\ } + {}^0\dot{\omega}_n(t) \times \left({}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)\right) + {}^0\omega_n(t) \times \left({}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)\right)\cdot \\ \Rightarrow \boxed{\ddot{T}_0} \quad {}^0\ddot{\mathbf{p}}(t) &= \underbrace{\left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)\ddot{\ }}_{(*1)} + \underbrace{{}^0\dot{\omega}_n(t) \times \left({}^0\mathbf{p}(t) - {}^0\mathbf{r}_n(\mathbf{q}(t))\right)}_{(*2)} + \underbrace{{}^0\omega_n(t) \times \left({}^0\omega_n(t) \times \left({}^0\mathbf{p}(t) - {}^0\mathbf{r}_n(\mathbf{q}(t))\right)\right)}_{(*2)} \end{aligned}$$

thereby are  
 (\*1) the linear acceleration and  
 (\*2) the centrifugal acceleration (points orthogonal to the rotation axis)  
 of a point mass  $P$ .

### Consequence:

For rotation with constant angular velocity, i.e.  ${}^0\omega_n(t) = \mathbf{const.} \in \mathbb{R}^3$ , it holds that  ${}^0\dot{\omega}_n(t) = \mathbf{0}$ , but from  $\boxed{\ddot{T}_0}$  it follows:

$${}^0\ddot{\mathbf{p}}(t) = \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)\ddot{\ } + {}^0\omega_n(t) \times \left({}^0\omega_n(t) \times \left({}^0R_n(\mathbf{q}(t)) \cdot {}^n\mathbf{p}(t)\right)\right).$$

$\Rightarrow$  Due to the centrifugal acceleration, consequently the different points  ${}^n\mathbf{p}$  of a rigid body have different accelerations.

### Relative acceleration in the general case ${}^n\dot{\mathbf{p}}(t) \neq \mathbf{0}$ :

On the right-hand side of equation  $\boxed{\ddot{T}_0}$  — statement without proof — the term

$$+ 2{}^0\omega_n(t) \times \left({}^0R_n(\mathbf{q}(t)) \cdot {}^n\dot{\mathbf{p}}(t)\right) + {}^0R_n(\mathbf{q}(t)) \cdot {}^n\ddot{\mathbf{p}}(t)$$

is added, where  $2{}^0\omega_n(t) \times \left({}^0R_n(\mathbf{q}(t)) \cdot {}^n\dot{\mathbf{p}}(t)\right)$  is called the *Coriolis acceleration* of a point mass  $P$ .

From this follows the general equation

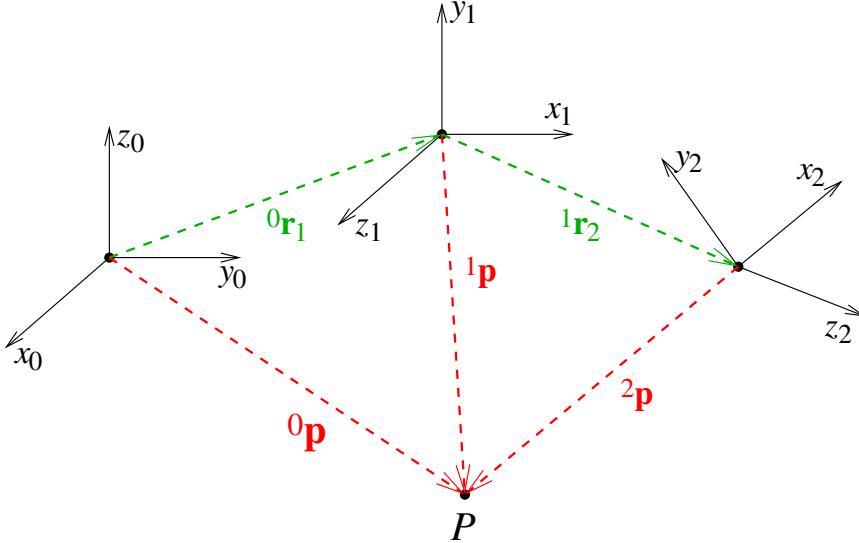
$$\begin{aligned} \boxed{\ddot{T}} \quad {}^0\ddot{\mathbf{p}}(t) &= \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)\ddot{\ } + {}^0\dot{\omega}_n(t) \times \left({}^0\mathbf{p}(t) - {}^0\mathbf{r}_n(\mathbf{q}(t))\right) + {}^0\omega_n(t) \times \left({}^0\omega_n(t) \times \left({}^0\mathbf{p}(t) - {}^0\mathbf{r}_n(\mathbf{q}(t))\right)\right) \\ &\quad + 2{}^0\omega_n(t) \times \left({}^0R_n(\mathbf{q}(t)) \cdot {}^n\dot{\mathbf{p}}(t)\right) + {}^0R_n(\mathbf{q}(t)) \cdot {}^n\ddot{\mathbf{p}}(t). \end{aligned}$$

## 4.3 Jacobian Matrix of a Manipulator

**Caution:** Do not confuse the Jacobian matrix of a manipulator with the Jacobian matrix of a function of several variables!

### 4.3.1 Addition of Angular Velocities

**Example:** Consider a fixed reference coordinate system  $S_0$  and two movable coordinate systems  $S_1, S_2$ :



It holds that

$$\begin{aligned} {}^0\mathbf{p} &= {}^0\mathbf{r}_1 + {}^0R_1 \cdot {}^1\mathbf{p} && (\text{coordinate transformation from } S_1 \rightarrow S_0) \\ {}^1\mathbf{p} &= {}^1\mathbf{r}_2 + {}^1R_2 \cdot {}^2\mathbf{p} && (\text{coordinate transformation from } S_2 \rightarrow S_1) \end{aligned}$$

Concatenation of both transformations yields the coordinate transformation from  $S_2 \rightarrow S_0$ :

$$\begin{aligned} {}^0\mathbf{p} &= {}^0\mathbf{r}_1 + {}^0R_1 \cdot {}^1\mathbf{r}_2 + {}^0R_1 \cdot {}^1R_2 \cdot {}^2\mathbf{p} \\ {}^0\mathbf{p}(t) &= \underbrace{{}^0\mathbf{r}_2(\mathbf{q}(t))}_{+} + \underbrace{{}^0R_2(\mathbf{q}(t))}_{+} \cdot {}^2\mathbf{p}(t) \end{aligned}$$

For the derivative of the rotation matrix  ${}^0R_2 = {}^0R_1 \cdot {}^1R_2$  resulting from the concatenation of both transformations, it follows that

$$\Rightarrow \underbrace{\frac{d}{dt}({}^0R_2(\mathbf{q}(t)))}_{(*1)} = \underbrace{({}^0R_1(\mathbf{q}(t))) \cdot {}^1R_2(\mathbf{q}(t))}_{(*2)} + \underbrace{{}^0R_1(\mathbf{q}(t)) \cdot ({}^1R_2(\mathbf{q}(t)))}_{(*3)}$$

Because of  $\dot{R} = B(\omega) \cdot R$ , on the one hand it holds

$$(*1) \quad ({}^0R_2(\mathbf{q}(t))) \cdot = B({}^0\omega_2(t)) \cdot {}^0R_2(\mathbf{q}(t))$$

and on the other hand

$$(*2) \quad ({}^0R_1(\mathbf{q}(t))) \cdot {}^1R_2(\mathbf{q}(t)) = B({}^0\omega_1(t)) \cdot \underbrace{{}^0R_1(\mathbf{q}(t)) \cdot {}^1R_2(\mathbf{q}(t))}_{={}^0R_2(\mathbf{q}(t))}$$

and

$$\begin{aligned}
 (*3) \quad {}^0R_1(\mathbf{q}(t)) \cdot ({}^1R_2(\mathbf{q}(t)))^\cdot &= {}^0R_1(\mathbf{q}(t)) \cdot B({}^1\omega_2(t)) \cdot {}^1R_2(\mathbf{q}(t)) \\
 &= {}^0R_1(\mathbf{q}(t)) \cdot B({}^1\omega_2(t)) \cdot \overbrace{{}^0R_1^T(\mathbf{q}(t)) \cdot {}^0R_1(\mathbf{q}(t))}^{=E} \cdot {}^1R_2(\mathbf{q}(t)) \\
 &= \underbrace{{}^0R_1(\mathbf{q}(t)) \cdot B({}^1\omega_2(t)) \cdot {}^0R_1^T(\mathbf{q}(t))}_{\text{Chap. 4.2 (iii), p. 60}} \cdot \underbrace{{}^0R_1(\mathbf{q}(t)) \cdot {}^1R_2(\mathbf{q}(t))}_{{}^0R_2} \\
 &= B({}^0R_1(\mathbf{q}(t)) \cdot {}^1\omega_2(t)) \cdot {}^0R_2(\mathbf{q}(t))
 \end{aligned}$$

All together yields with  $(*)1 = (*)2 + (*)3$ :

$$\begin{aligned}
 \underbrace{B({}^0\omega_2(t)) \cdot {}^0R_2(\mathbf{q}(t))}_{(*)4} &= B({}^0\omega_1(t)) \cdot {}^0R_2(\mathbf{q}(t)) + B({}^0R_1(\mathbf{q}(t)) \cdot {}^1\omega_2(t)) \cdot {}^0R_2(\mathbf{q}(t)) \\
 &= \underbrace{\left( B({}^0\omega_1(t)) + B({}^0R_1(\mathbf{q}(t)) \cdot {}^1\omega_2(t)) \right)}_{\text{Chap. 4.2 (i), p. 60: } B(a+b) = B(a) + B(b)} \cdot {}^0R_2(\mathbf{q}(t)) \\
 &= \underbrace{B({}^0\omega_1(t) + {}^0R_1(\mathbf{q}(t)) \cdot {}^1\omega_2(t)) \cdot {}^0R_2(\mathbf{q}(t))}_{(*)5}
 \end{aligned}$$

$$(*)4 = (*)5 \Rightarrow B({}^0\omega_2(t)) = B({}^0\omega_1(t) + {}^0R_1(\mathbf{q}(t)) \cdot {}^1\omega_2(t))$$

$$(\text{Chap. 4.2 (i), p. 60}) \Rightarrow {}^0\omega_2(t) = {}^0\omega_1(t) + {}^0R_1(\mathbf{q}(t)) \cdot {}^1\omega_2(t)$$

### Result:

For the relative rotations of several concatenated coordinate systems, one obtains the angular velocity of the total motion  ${}^0\omega_n$  by **addition** of the relative local angular velocities  ${}^{i-1}\omega_i$  transformed to the fixed reference coordinate system  $S_0$ , i.e. by addition of  ${}^0R_{i-1} \cdot {}^{i-1}\omega_i$ .

**Generalization:** For  $n$  concatenated rotations:

$${}^0R_n(\mathbf{q}(t)) = {}^0R_1(\mathbf{q}(t)) \cdot {}^1R_2(\mathbf{q}(t)) \cdot \dots \cdot {}^{n-1}R_n(\mathbf{q}(t))$$

it holds:

$$({}^0R_n(\mathbf{q}(t)))^\cdot = B({}^0\omega_n(t)) \cdot {}^0R_n(\mathbf{q}(t))$$

with the angular velocity vector

$${}^0\omega_n(t) = {}^0\omega_1(t) + {}^0R_1(\mathbf{q}(t)) \cdot {}^1\omega_2(t) + \dots + {}^0R_{n-1}(\mathbf{q}(t)) \cdot {}^{n-1}\omega_n(t)$$

### 4.3.2 Derivation of the Jacobian Matrix of a Manipulator

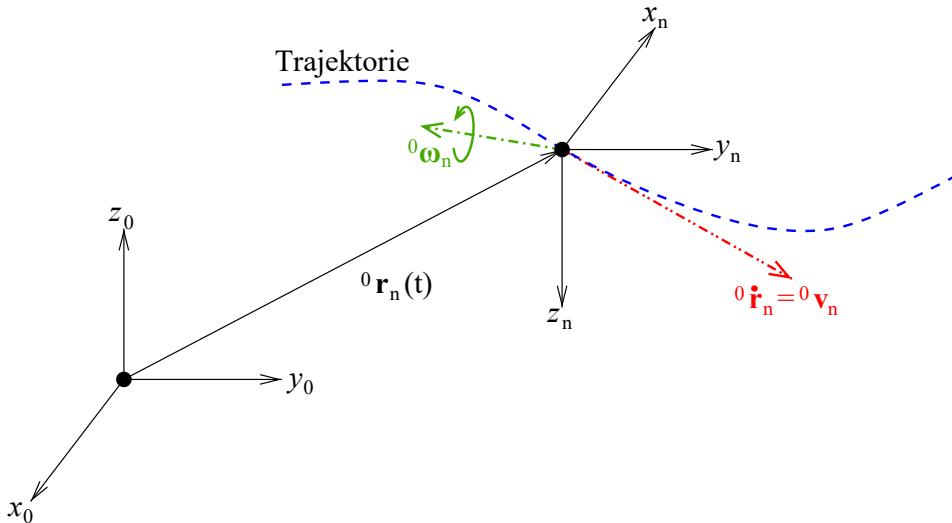
For an  $n$ -jointed manipulator (robot arm or leg) with the joint variables  $q_1, q_2, \dots, q_n$ , the transformation from the end-effector coordinate system  $S_n$  to the fixed base coordinate system  $S_0$  is described by

$${}^0T_n(\mathbf{q}) = \begin{pmatrix} {}^0R_n(\mathbf{q}) & {}^0\mathbf{r}_n(\mathbf{q}) \\ \mathbf{0}^\text{T} & 1 \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix} \in \mathbb{R}^n$$

During a motion,  $\mathbf{q} = \mathbf{q}(t)$  changes. This also causes  ${}^0R_n(\mathbf{q}(t))$  and  ${}^0\mathbf{r}_n(\mathbf{q}(t))$  to change over time  $t$ .

Relevant for the velocity of position and orientation of the end-effector are:

- a)  ${}^0\mathbf{v}_n(t) := {}^0\dot{\mathbf{r}}_n(\mathbf{q}(t))$ : the (linear) velocity vector of the origin of the end-effector frame.
- b)  ${}^0\omega_n(t)$ : the (rotational) angular velocity vector of the end-effector frame with respect to the fixed reference coordinate system  $S_0$ , and



**Sought:** is the relationship between  ${}^0\mathbf{v}_n(t)$  and  ${}^0\omega_n(t)$  on the one hand and  $\dot{\mathbf{q}}(t)$  on the other hand:

$\underbrace{{}^0\mathbf{v}_n(t)}_{\text{vector each } \in \mathbb{R}^3}$	$= \underbrace{{}^0J_{n,v}(\mathbf{q}(t))}_{(3 \times n)\text{-matrices, sought!}} \cdot \underbrace{\dot{\mathbf{q}}(t)}_{\text{vector } \in \mathbb{R}^n}$
$\underbrace{{}^0\omega_n(t)}_{\text{vector each } \in \mathbb{R}^3}$	$= \underbrace{{}^0J_{n,\omega}(\mathbf{q}(t))}_{(3 \times n)\text{-matrices, sought!}} \cdot \underbrace{\dot{\mathbf{q}}(t)}_{\text{vector } \in \mathbb{R}^n}$

or in one equation:

$$\underbrace{\begin{pmatrix} {}^0\mathbf{v}_n(t) \\ {}^0\omega_n(t) \end{pmatrix}}_{\in \mathbb{R}^6} = \underbrace{{}^0J_n(\mathbf{q}(t))}_{(6 \times n)\text{-matrix}} \cdot \underbrace{\dot{\mathbf{q}}(t)}_{\in \mathbb{R}^n} \quad \text{with} \quad {}^0J_n = \begin{pmatrix} {}^0J_{n,v} \\ {}^0J_{n,\omega} \end{pmatrix}.$$

**Definition 4.4** The  $(6 \times n)$ -matrix  ${}^0J_n(\mathbf{q}(t))$  is called the **Jacobian matrix of an  $n$ -jointed manipulator**.

### (i) Calculation of the Angular Velocity ${}^0\omega_n(t)$

According to 4.3.1, to determine the angular velocity vector  ${}^0\omega_n(t)$  of the end-effector frame with respect to the base frame  $S_0$ , the (relative) angular velocities  ${}^{i-1}\omega_i(t)$  must be expressed with respect to  $S_0$  (i.e. as  ${}^0\omega_i(t)$ ) and summed up.

#### 1st Case: Joint $i$ , $i \in \{1, 2, \dots, n\}$ is a revolute joint

Then (according to DH convention)  $q_i = \theta_i$  (rotation angle) holds and the  $z_{i-1}$ -axis is the rotation axis.

$\Rightarrow$  The angular velocity  ${}^{i-1}\omega_i(t)$  of the  $i$ -th joint with respect to frame  $S_{i-1}$  is given by

$${}^{i-1}\omega_i(t) = \dot{q}_i(t) \cdot {}^{i-1}\mathbf{e}_{z_{i-1}} = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_i(t) \end{pmatrix}, \quad \left[ \begin{array}{l} {}^i\mathbf{e}_{x_i} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ \text{with } {}^i\mathbf{e}_{y_i} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ {}^i\mathbf{e}_{z_i} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{array} \right].$$

For the following considerations, the relative angular velocity  ${}^{i-1}\omega_i(t)$  must be expressed in coordinates with respect to frame  $S_0$  by multiplying it with  ${}^0R_{i-1}$ :

$${}^0R_{i-1}(\mathbf{q}(t)) \cdot {}^{i-1}\omega_i(t) = \dot{q}_i \cdot \underbrace{{}^0R_{i-1}(\mathbf{q}(t)) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}_{{}^0\mathbf{e}_{z_{i-1}}} = \dot{q}_i \cdot {}^0\mathbf{e}_{z_{i-1}}$$

**Remark:** Caution,  ${}^0R_{i-1}(\mathbf{q}(t)) \cdot {}^{i-1}\omega_i(t)$  is **not** equal to the absolute angular velocity vector  ${}^0\omega_i(t)$ . For this the formula at the end of Section 4.3.1 on page 65 holds.

#### 2nd Case: Joint $i$ is a prismatic joint

Then (according to DH convention)  $q_i = d_i$  (length of displacement) holds and the motion between frame  $S_{i-1}$  and  $S_i$  is a pure translation, i.e. there is no angular velocity

$$\Rightarrow {}^{i-1}\omega_i(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

## Overall

one obtains with

$$\rho_i = \begin{cases} 1, & \text{if } q_i = \theta_i \text{ (revolute joint)} \\ 0, & \text{if } q_i = d_i \text{ (prismatic joint)} \end{cases}$$

for the angular velocity vector of the end-effector orientation with respect to  $S_0$ :

$$\begin{aligned} {}^0\omega_n(t) &= \rho_1 \cdot \dot{q}_1(t) \cdot \mathbf{e}_z + \rho_2 \cdot \dot{q}_2(t) \cdot {}^0R_1(\mathbf{q}(t)) \cdot \mathbf{e}_z + \rho_3 \cdot \dot{q}_3(t) \cdot {}^0R_2(\mathbf{q}(t)) \cdot \mathbf{e}_z + \\ &\dots + \rho_n \cdot \dot{q}_n(t) \cdot \underbrace{{}^0R_{n-1}(\mathbf{q}(t)) \cdot \mathbf{e}_z}_{= {}^0\mathbf{e}_{z_{n-1}}(\mathbf{q}(t))}, \quad \text{with } \mathbf{e}_z = {}^i\mathbf{e}_{z_i} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ &= \sum_{i=1}^n \underbrace{\rho_i \dot{q}_i(t)}_{\in \mathbb{R}} \underbrace{{}^0\mathbf{e}_{z_{i-1}}(\mathbf{q}(t))}_{\in \mathbb{R}^3 \text{ (*1)}} \end{aligned}$$

(\*1)  ${}^0\mathbf{e}_{z_{i-1}}(\mathbf{q}(t))$  is the third unit vector  $\mathbf{e}_z$  of coordinate system  $S_{i-1}$  expressed in coordinates of  $S_0$  (with coincident origin)

Overall it follows from case 1 and case 2:

$$\Rightarrow \underbrace{{}^0J_{n,\omega}(\mathbf{q}(t))}_{\in \mathbb{R}^{3 \times n}} = \left[ \begin{array}{c|c|c|c} \rho_1 \cdot {}^0\mathbf{e}_{z_0} & \dots & \rho_i \cdot {}^0\mathbf{e}_{z_{i-1}} & \dots & \rho_n \cdot {}^0\mathbf{e}_{z_{n-1}} \\ \hline \uparrow & & \uparrow & & \uparrow \\ \text{1st column} & & i\text{-th column} & & n\text{-th column} \end{array} \right] \left. \right\} 3 \text{ rows}$$

Thereby the vector  ${}^0\mathbf{e}_{z_{i-1}}$  is nothing other than the third column of the matrix  ${}^0R_{i-1}(\mathbf{q}(t))$ .

## (ii) Calculation of the Linear Velocity ${}^0\mathbf{v}_n(t)$

$$\begin{aligned}
 {}^0\mathbf{v}_n(t) &= \left({}^0\mathbf{r}_n(\mathbf{q}(t))\right)' = \frac{d}{dt} \begin{pmatrix} {}^0r_{n,x}(q_1(t), \dots, q_n(t)) \\ {}^0r_{n,y}(q_1(t), \dots, q_n(t)) \\ {}^0r_{n,z}(q_1(t), \dots, q_n(t)) \end{pmatrix} \\
 &= \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_1} \cdot \dot{q}_1(t) + \dots + \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_n} \cdot \dot{q}_n(t) \quad \left(\text{with } \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_i} \in \mathbb{R}^{3 \times 1}\right) \\
 &= \sum_{i=1}^n \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_i} \cdot \dot{q}_i(t) = \sum_{i=1}^n \begin{pmatrix} \frac{\partial {}^0r_{n,x}(\mathbf{q}(t))}{\partial q_i} \\ \frac{\partial {}^0r_{n,y}(\mathbf{q}(t))}{\partial q_i} \\ \frac{\partial {}^0r_{n,z}(\mathbf{q}(t))}{\partial q_i} \end{pmatrix} \cdot \dot{q}_i(t) \\
 &= \left( \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial \mathbf{q}} \right) \cdot \dot{\mathbf{q}}(t)
 \end{aligned}$$

$$\Rightarrow \underbrace{{}^0J_{n,v}(\mathbf{q}(t))}_{\in \mathbb{R}^{3 \times n}} = \left[ \begin{array}{c|c|c|c}
 \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_1} & \dots & \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_i} & \dots & \frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_n} \\
 \hline
 \uparrow & & \uparrow & & \uparrow \\
 \text{1st column} & & i\text{-th column} & & n\text{-th column}
 \end{array} \right] \left. \right\} \text{3 rows}$$

I.e. the part of the Jacobian matrix of the manipulator that concerns the linear velocity of the origin of  $S_n$  coincides with the normal Jacobian matrix of  ${}^0\mathbf{r}_n(\mathbf{q})$ . This does not hold, however, for the angular velocity.

**Remark:** In general, the first  $i - 1$  rotation matrices  ${}^0R_j$ ,  $j = 1, \dots, i - 1$ , are by construction always independent of  $q_i$ , as are the vectors  ${}^0\mathbf{r}_j$ ,  $j = 1, \dots, i - 1$ . The last  $n - i + 1$  rotation matrices  ${}^0R_j$ ,  $j = i, \dots, n$ , however, generally depend on  $q_i$ .

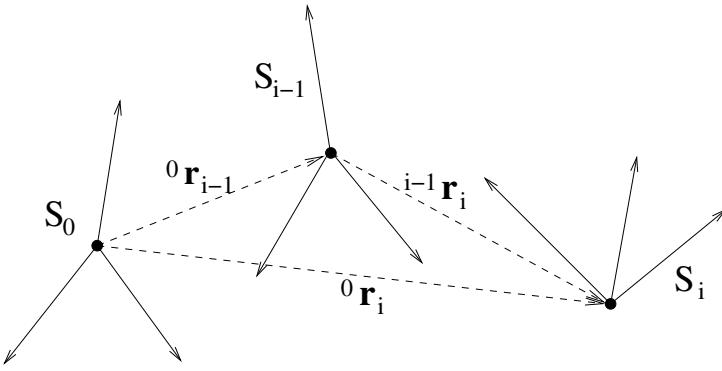
### 1st Case: Joint $i$ is a prismatic joint

Then (according to DH convention)  $q_i = d_i$  (length of displacement) holds and **all** rotation matrices  ${}^0R_j$ ,  $j = 1, \dots, n$ , are independent of  $q_i$ .

$$\Rightarrow \frac{\partial {}^0R_j(\mathbf{q})}{\partial q_i} = O \in \mathbb{R}^{3 \times 3}, j = 1, \dots, n.$$

According to the DH conventions (or an analogous procedure for an alternative kinematic modeling):

$${}^{i-1}\mathbf{r}_i(\mathbf{q}(t)) = d_i(t) \cdot \mathbf{e}_z + {}^{i-1}R_i(\mathbf{q}(t)) \cdot (a_i \cdot \mathbf{e}_x) = \begin{pmatrix} 0 \\ 0 \\ d_i(t) \end{pmatrix} + {}^{i-1}R_i(\mathbf{q}(t)) \begin{pmatrix} a_i \\ 0 \\ 0 \end{pmatrix}$$



With the relation (in which all quantities depend on  $\mathbf{q}(t)$ )

$${}^0\mathbf{r}_i = {}^0\mathbf{r}_{i-1} + {}^0R_{i-1} \cdot {}^{i-1}\mathbf{r}_i$$

by total differentiation with respect to  $t$  for the case that only the  $i$ -th joint moves and all others are held constant (i.e.  $q_j(t) = \text{const}$ , for all  $j \in \{1, \dots, n\}$  with  $j \neq i$ ), it follows:

$$\begin{aligned} ({}^0\mathbf{r}_i(\mathbf{q}(t)))^\cdot &= ({}^0\mathbf{r}_{i-1}(\mathbf{q}(t)))^\cdot + ({}^0R_{i-1}(\mathbf{q}(t)))^\cdot \cdot {}^{i-1}\mathbf{r}_i(\mathbf{q}(t)) + {}^0R_{i-1}(\mathbf{q}(t)) \cdot ({}^{i-1}\mathbf{r}_i(\mathbf{q}(t)))^\cdot \\ &= 0 + 0 + \underbrace{{}^0R_{i-1}}_{\text{constant}} \cdot \underbrace{{}^{i-1}\dot{\mathbf{r}}_i(\mathbf{q}(t))}_{\dot{d}_i(t) \cdot \mathbf{e}_z} \\ &= {}^0\mathbf{e}_{z_{i-1}} \cdot \dot{d}_i(t) \\ &= \frac{\partial {}^0\mathbf{r}_i(\mathbf{q})}{\partial q_i} \cdot \dot{q}_i(t) \end{aligned}$$

Because  ${}^0\mathbf{r}_n = {}^0\mathbf{r}_i + {}^0R_i \cdot {}^i\mathbf{r}_n$  in this case it holds that  $\frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_i} = \frac{\partial {}^0\mathbf{r}_i(\mathbf{q})}{\partial q_i}$ . Overall it thus follows

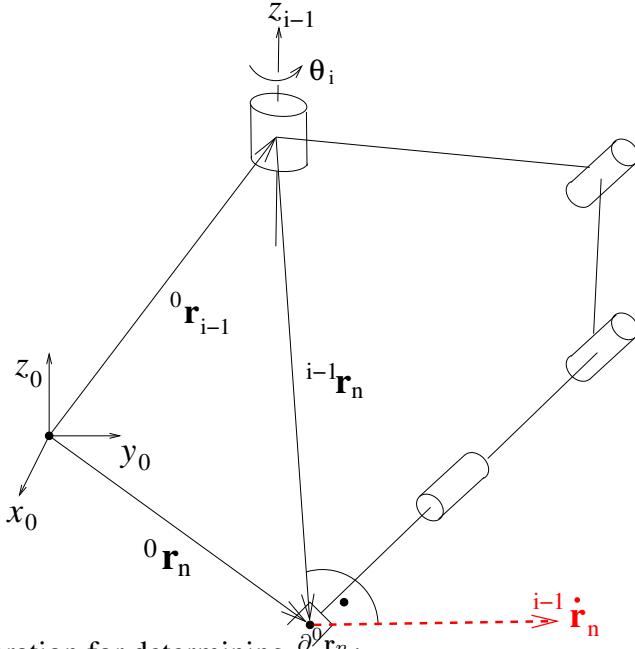
$$\frac{\partial {}^0\mathbf{r}_n(\mathbf{q})}{\partial q_i} = {}^0\mathbf{e}_{z_{i-1}}, \quad \text{if } q_i = d_i \text{ (prismatic joint)}$$

**Remark:** The vector  ${}^0\mathbf{e}_{z_i}$  can be determined directly from the forward kinematics model, namely from the third column of the homogeneous transformation matrix between  $S_i$  and  $S_0$

$${}^0T_i = \begin{array}{|c|c|c|c|} \hline & {}^0\mathbf{e}_{x_i} & {}^0\mathbf{e}_{y_i} & {}^0\mathbf{e}_{z_i} & {}^0\mathbf{r}_i \\ \hline & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

## 2nd Case: Joint $i$ is a revolute joint

In this case the linear velocity vector is tangential to the rotational motion:



Preliminary consideration for determining  $\frac{\partial \dot{r}_n}{\partial q_i}$ :

$$\begin{aligned} {}^0 \mathbf{r}_n &= {}^0 \mathbf{r}_{i-1} + {}^0 R_{i-1} {}^{i-1} \mathbf{r}_n \\ \Rightarrow {}^0 \dot{\mathbf{r}}_n - {}^0 \dot{\mathbf{r}}_{i-1} &= {}^0 R_{i-1} {}^{i-1} \dot{\mathbf{r}}_n \end{aligned}$$

If **only** the  $i$ -th joint is moved,  ${}^0 \mathbf{r}_{i-1}$  and  ${}^0 R_{i-1}$  are constant:

$$\frac{d}{dt} \Rightarrow {}^0 \dot{\mathbf{r}}_n(t) = {}^0 R_{i-1} {}^{i-1} \dot{\mathbf{r}}_n(t)$$

Because the motion is a rotation about the  $z_{i-1}$ -axis with angle  $q_i = \theta_i$ , it holds:

$${}^{i-1} \dot{\mathbf{r}}_n = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_i \end{pmatrix} \times {}^{i-1} \mathbf{r}_n$$

i.e. the velocity vector  ${}^{i-1} \dot{\mathbf{r}}_n$  is orthogonal to the rotation axis  $z_{i-1}$  and to the vector  ${}^{i-1} \mathbf{r}_n$  and its length depends on the rotational velocity  $\dot{q}_i$  of the rotation axis and its distance  ${}^0 R_{i-1} {}^{i-1} \mathbf{r}_n$  from the rotation axis. Thus it follows further

$$\begin{aligned} {}^0 \dot{\mathbf{r}}_n &= {}^0 R_{i-1} \cdot {}^{i-1} \dot{\mathbf{r}}_n \\ &= {}^0 R_{i-1} \cdot (\dot{q}_i \mathbf{e}_z \times {}^{i-1} \mathbf{r}_n) \\ &= (\dot{q}_i {}^0 R_{i-1} \mathbf{e}_z) \times ({}^0 R_{i-1} {}^{i-1} \mathbf{r}_n) \\ &= (\dot{q}_i {}^0 \mathbf{e}_{z_{i-1}}) \times ({}^0 \mathbf{r}_n - {}^0 \mathbf{r}_{i-1}) \\ &= ({}^0 \mathbf{e}_{z_{i-1}} \times ({}^0 \mathbf{r}_n - {}^0 \mathbf{r}_{i-1})) \dot{q}_i \end{aligned}$$

Because  ${}^0 \dot{\mathbf{r}}_n = \frac{d}{dt} {}^0 \mathbf{r}_n(\mathbf{q}(t)) = \frac{\partial {}^0 \mathbf{r}_n}{\partial q_i} \dot{q}_i(t)$  it follows:

$$\frac{\partial {}^0 \mathbf{r}_n}{\partial q_i} = {}^0 \mathbf{e}_{z_{i-1}} \times ({}^0 \mathbf{r}_n - {}^0 \mathbf{r}_{i-1})$$

if  $q_i = \theta_i$  (revolute joint)

## Summary of (i) and (ii)

It holds that

$$\underbrace{\begin{pmatrix} {}^0\mathbf{v}_n(t) \\ {}^0\omega_n(t) \end{pmatrix}}_{\in \mathbb{R}^6} = \underbrace{{}^0J_n(\mathbf{q}(t))}_{(6 \times n)-\text{Matrix}} \cdot \underbrace{\dot{\mathbf{q}}(t)}_{\in \mathbb{R}^n}$$

mit der Manipulator-Jacobi-Matrix (engl. Manipulator Jacobian)

$${}^0J_n = \left[ \begin{array}{c|c|c|c|c} & & & & \\ \hline {}^0J_{n,1} & \dots & {}^0J_{n,i} & \dots & {}^0J_{n,n} \\ \hline \end{array} \right] \quad \left. \right\} 6 \text{ Zeilen}$$

↑                    ↑                    ↑  
1. Spalte      i. Spalte      n. Spalte

The  $i$ -th column  ${}^0J_{n,i}$  of the manipulator Jacobian matrix is, if joint  $i$  is

- a) a revolute joint ( $q_i = \theta_i$ ):

$$\underbrace{{}^0J_{n,i}}_{\in \mathbb{R}^6} = \left( \begin{array}{c} {}^0\mathbf{e}_{z_{i-1}} \times \left( {}^0\mathbf{r}_n - {}^0\mathbf{r}_{i-1} \right) \\ {}^0\mathbf{e}_{z_{i-1}} \end{array} \right) \quad \left. \right\} \in \mathbb{R}^3$$

- b) a prismatic joint ( $q_i = d_i$ ):

$$\underbrace{{}^0J_{n,i}}_{\in \mathbb{R}^6} = \left( \begin{array}{c} {}^0\mathbf{e}_{z_{i-1}} \\ 0 \end{array} \right) \quad \left. \right\} \in \mathbb{R}^3$$

$\Rightarrow$  The Jacobian matrix of a manipulator can be easily calculated from the forward kinematics model, namely using the third and fourth columns of the transformation matrices:

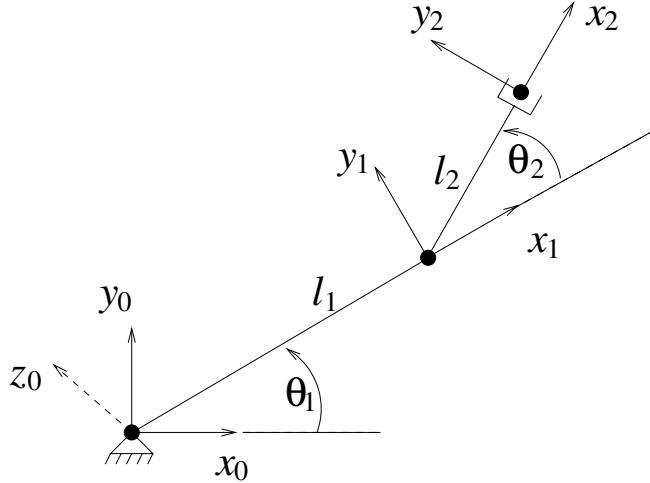
$${}^0T_i(\mathbf{q}) = \left[ \begin{array}{c|c|c|c} {}^0\mathbf{e}_{x_i} & {}^0\mathbf{e}_{y_i} & {}^0\mathbf{e}_{z_i} & {}^0\mathbf{r}_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

This „recipe“ applies of course only if (as in this chapter) the kinematic modeling was performed according to the DH conventions. With a different specification of the local coordinate systems and transformations, however, the Jacobian matrix can be determined in an analogous manner.

**Remark:** With this procedure, not only the velocity vectors of position and orientation for the end-effector (TCP), but also for every point on the robot can be calculated (e.g. for the center of mass of a robot link as in Chapter 5).

## Example: Planar 2-DOF SCARA Manipulator

As an example we examine the Jacobian matrix of the two-jointed robot arm whose forward kinematics model was investigated in Section 3.1.3 (page 36 ff.).



(Caution: Position of  $S_n = S_2$  is here simplified compared to the special convention in step (6) of the DH algorithm, which would require a further rotation of  $S_2$ .)

$$n = 2, \quad \mathbf{q} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$$

$${}^0T_1(\theta_1) = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & l_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & l_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0T_2(\theta_1, \theta_2) = \begin{pmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0J_n(\mathbf{q}) = {}^0J_2(\theta_1, \theta_2) \quad \text{is } (6 \times 2)\text{-matrix with}$$

$${}^0J_2(\theta_1, \theta_2) = \left[ \begin{array}{c|c} {}^0\mathbf{e}_{z_0} \times ({}^0\mathbf{r}_2 - {}^0\mathbf{r}_0) & {}^0\mathbf{e}_{z_1} \times ({}^0\mathbf{r}_2 - {}^0\mathbf{r}_1) \\ \hline {}^0\mathbf{e}_{z_0} & {}^0\mathbf{e}_{z_1} \end{array} \right]$$

With  ${}^0\mathbf{r}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  and  ${}^0\mathbf{e}_{z_0} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$  it follows:

$${}^0J_2(\theta_1, \theta_2) = \left[ \begin{array}{cc} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{array} \right] \quad \begin{array}{l} {}^0J_{2,v} \\ {}^0J_{2,\omega} \end{array}$$

**Result:**

- ${}^0\mathbf{v}_2(t) = {}^0\dot{\mathbf{r}}_2(t) = {}^0J_{2,v}(\theta_1, \theta_2) \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$  is the linear velocity vector of the origin of  $S_2$  relative to  $S_0$ .

**Remark:** Precisely formulated,  ${}^0\mathbf{v}_2$  depends not only on  $t$  but on  $\mathbf{q}(t)$  and  $\dot{\mathbf{q}}(t)$ . That is, with  ${}^0\mathbf{v}_2(t)$  is actually meant in this equation  ${}^0\mathbf{v}_2(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ .

- ${}^0\omega_2(t) = {}^0J_{2,\omega}(\theta_1, \theta_2) \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{pmatrix}$  is the angular velocity vector of frame  $S_2$  with respect to  $S_0$  (in this case a rotation about the  $z_0$ -axis with angular velocity  $\dot{\theta}_1 + \dot{\theta}_2$ ).

**Remark:** Precisely formulated,  ${}^0\omega_2$  depends not only on  $t$  but on  $\mathbf{q}(t)$  and  $\dot{\mathbf{q}}(t)$ . That is, with  ${}^0\omega_2(t)$  is actually meant in this equation  ${}^0\omega_2(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ .

## 4.4 Inverse Jacobian Model

**So far:** Investigation of forward and inverse kinematics

$$\begin{array}{c}
 \text{Joint variables } \mathbf{q} \in \mathbb{R}^n \xrightleftharpoons[\text{INV KIN (IK)}]{\text{DIR KIN (FK)}} \\
 \left\{ \begin{array}{l}
 \text{(a) Cartesian pose:} \\
 {}^0T_n(\mathbf{q}) = \left( \begin{array}{c|c} {}^0R_n & {}^0\mathbf{r}_n \\ \hline 0^T & 1 \end{array} \right) \in \mathbb{R}^{4 \times 4} \\
 \text{or} \\
 \text{(b) } \begin{pmatrix} {}^0\mathbf{r}_n \\ \text{three orientation angles} \end{pmatrix} \in \mathbb{R}^6
 \end{array} \right.
 \end{array}$$

**INV KIN:** Calculation of the joint variables needed to reach a desired pose.

**Now:** Investigation of the forward and backward relationship for velocities:

$$\text{Joint velocities } \dot{\mathbf{q}} \in \mathbb{R}^n \xrightleftharpoons[\text{INV JAC[q]}]{\text{DIR JAC[q]}} \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} \in \mathbb{R}^6$$

$$\text{DIR JAC[q]: } \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} = {}^0J_n(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

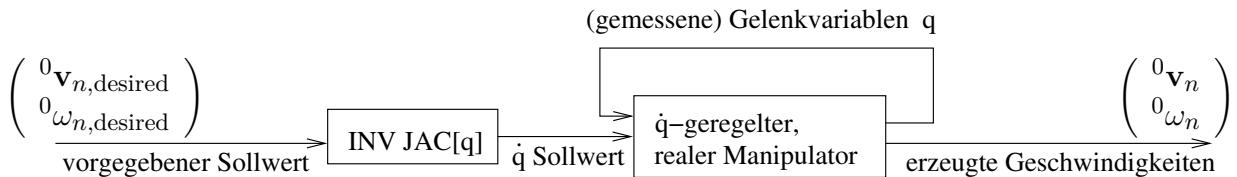
**INV JAC[q]:** Calculation of the joint variable velocities necessary for given linear (i.e.  ${}^0\mathbf{v}_n$ ) and angular velocity vectors (i.e.  ${}^0\omega_n$ )

$$\dot{\mathbf{q}} = \text{INV JAC[q]} \cdot \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix}.$$

Questions to be clarified: When does INV JAC exist? And how can it be calculated in these cases?

### 4.4.1 Application for Velocity Control

**Simple velocity control of a manipulator (resolved rate control)**



**Remark:**

This type of velocity control is only suitable for **slow** motions due to the small influence there of the kinetic effects not yet considered (forces, mass distributions, etc., see Chapter 5).

### Investigation of INV JAC[q]

That is, investigation of solving the linear system of equations for  $\dot{\mathbf{q}}$ .

From the theory of linear systems of equations, it follows that a solution for INV JAC can only exist if  $\text{rank}({}^0J_n(\mathbf{q})) = \text{rank}\left({}^0J_n(\mathbf{q}), \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix}\right)$ .

$n = 6$  : Then  ${}^0J_n(\mathbf{q})$  is a square  $(6 \times 6)$ -matrix. For 6-DOF manipulators with **regular** Jacobian matrix,  $\text{INV JAC}[\mathbf{q}] = {}^0J_n(\mathbf{q})^{-1}$  (i.e. the inverse of the Jacobian matrix of the manipulator).

This case applies to many 6-DOF industrial robots with suitably arranged joints.

$n < 6$  : Then  ${}^0J_n(\mathbf{q})$  is not square and

$$\begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} = {}^0J_n(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

⇒ Is an **overdetermined** linear system of equations for  $\dot{\mathbf{q}}$ !

Only  $m$  of the 6 values  $\begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix}$  can be specified. Here  $m \leq n < 6$  is the dimension of the manipulation space of reachable poses (consisting of positions and orientations).

(→ see following example with 2-DOF SCARA manipulator)

$n > 6$  : Then  ${}^0J_n(\mathbf{q})$  is not square and

$$\begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} = {}^0J_n(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

⇒ is an **underdetermined** linear system of equations for  $\dot{\mathbf{q}}$ !

For given  $\begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix}$  of the manipulator space there are then generally infinitely many solutions for  $\dot{\mathbf{q}}$ , provided that  $\text{rank}({}^0J_n) = \text{rank}\left({}^0J_n, \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix}\right)$  holds.

For such **kinematically redundant** manipulators, additional conditions must therefore be imposed so that INV JAC yields a unique solution. The specification of the „redundant“ degrees of freedom can be done e.g.

- by adding **kinematic constraints** (i.e. prescribing individual components  $q_i$  or  $\dot{q}_i$ ) or
- by maximizing a quality criterion, e.g. the manipulability measure (manipulability measure) according to Yoshikawa

$$\mu(\mathbf{q}) = \sqrt{\det \left( \underbrace{{}^0 J_n(\mathbf{q}) \cdot {}^0 J_n^T(\mathbf{q})}_{(6 \times 6\text{-matrix})} \right)},$$

under the linear side conditions  $\begin{pmatrix} {}^0 \mathbf{v}_n \\ {}^0 \omega_n \end{pmatrix} - {}^0 J_n(\mathbf{q}) \cdot \dot{\mathbf{q}} = 0 \in \mathbb{R}^6$ .

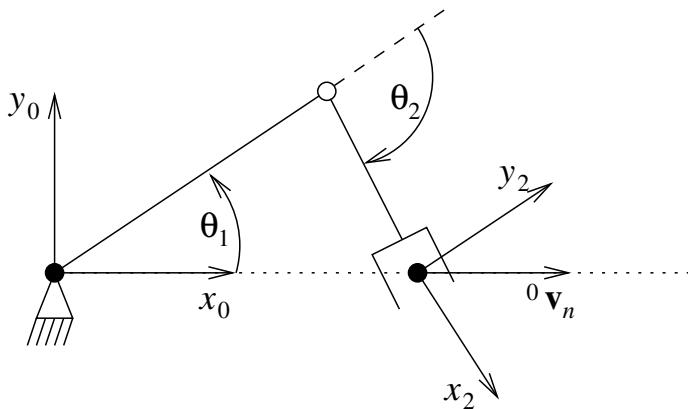
The value of the manipulability measure  $\mu(\mathbf{q})$  corresponds to the magnitude of the product of the six singular values of the Jacobian matrix  ${}^0 J_n(\mathbf{q})$ . The maximization of  $\mu(\mathbf{q})$  therefore means that essentially a solution is sought „as far away as possible“ from singular values that are small in magnitude, close to zero, and with singular values that are „as large as possible“ in magnitude.

Since for the above minimization problem with a redundant manipulator the sought solution  $\dot{\mathbf{q}}$  is generally not yet uniquely determined, an extended quality criterion can be considered, e.g.  $\mu(\mathbf{q}) - \|\dot{\mathbf{q}}\|$ .

Kinematically redundant manipulators are e.g.:

- a 6 DOF manipulator on a 2-DOF mobile base,
- the 7-DOF robot arm from Franka Emika<sup>2</sup>,
- the human arm: 7 DOF,
- the human hand: approx. 22 DOF.

### Example: Motion of the 2-DOF SCARA Manipulator



A motion of the end-effector of the 2-DOF SCARA manipulator is examined along the  $x_0$ -axis with constant velocity 1 m/s. That is, the desired linear velocity vector of the origin of the end-effector coordinate system throughout the motion is:

$${}^0 \mathbf{v}_n(t) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \left[ \frac{\text{m}}{\text{s}} \right], \quad 0 \leq t \leq t_f.$$

---

<sup>2</sup><https://www.franka.de/>

The relationship

$$\begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} = {}^0J_n(\mathbf{q}) \cdot \dot{\mathbf{q}}$$

reads here explicitly

$$\begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$$

The robot has  $n = 2$  joints and the dimension of the manipulation space is  $m = 2$ . Therefore three variants for the inverse Jacobian model are possible:

- (i) Specification of  $\dot{x}_0, \dot{y}_0$  or
- (ii) Specification of  $\dot{x}_0, \omega_z$  or
- (iii) Specification of  $\dot{y}_0, \omega_z$ .

INV JAC exists for the linear  $(2 \times 2)$ -system of equations that can be formed from any two of the three rows 1, 2, or 6. (Rows 3, 4, and 5 with  $\dot{z}_0 = 0 = \omega_x = \omega_y$  are trivially always satisfied and provide no information.)

In the following investigation of case (i):

$$\begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \end{pmatrix} = \underbrace{{}^0\hat{J}_n(\theta_1, \theta_2)}_{(2 \times 2)\text{-matrix}} \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$$

with  ${}^0\hat{J}_n(\theta_1, \theta_2) = \begin{pmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{pmatrix}$

Because of

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

and

$$\det({}^0\hat{J}_n(\theta_1, \theta_2)) = ad - bc = \dots \text{ inserting and calculating } \dots = l_1 l_2 \sin(\theta_2)$$

it follows

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} = \frac{1}{l_1 l_2 \sin(\theta_2)} \begin{pmatrix} l_2 \cos(\theta_1 + \theta_2) & l_2 \sin(\theta_1 + \theta_2) \\ -l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) & -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) \end{pmatrix} \cdot \begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \end{pmatrix}$$

$$=: \text{INV JAC}[(\theta_1, \theta_2)] \cdot \begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \end{pmatrix}$$

By inserting the desired velocity vector

$$\begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} [\text{m/s}]$$

the necessary desired velocities of the joint variables result as

$$\begin{aligned}\dot{\theta}_1 &= \frac{\cos(\theta_1 + \theta_2)}{l_1 \cdot \sin(\theta_2)} \\ \dot{\theta}_2 &= \frac{-l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2)}{l_1 l_2 \sin(\theta_2)}\end{aligned}$$

INV JAC has now been determined for the SCARA manipulator for the case that  $\dot{x}_0$  and  $\dot{y}_0$  are given. The motion and the resulting relationships for the velocities are examined more closely:

As the manipulator arm is extended,  $|\dot{\theta}_2|$  becomes smaller and smaller and approaches 0. Because

$$\theta_2 \leq 0 \quad \text{and} \quad \theta_2 \rightarrow 0 \quad \Rightarrow \quad \frac{1}{\sin(\theta_2)} \rightarrow -\infty$$

$\dot{\theta}_1$  and  $\dot{\theta}_2$  must according to the determined formulas grow more and more to maintain the actually small, constant desired velocity of the end-effector of 1 [m/s]. In the vicinity of  $\theta_2 = 0$  the values for  $\dot{\theta}_1$  and  $\dot{\theta}_2$  literally „explode“!

And for  $\theta_2 = 0$  the joint velocities  $\dot{\theta}_1$  and  $\dot{\theta}_2$  must even become infinitely large!

What happened?

For  $\theta_2 = 0 \Rightarrow \det(^0\hat{J}_n(\theta_1, \theta_2)) = 0$  i.e.  ${}^0\hat{J}_n$  is singular!  
 $\theta_2 = 0$  is a so-called **kinematic singularity**.

#### 4.4.2 Kinematic Singularities

**Repetition:**

- Rank of a matrix  $J$  = maximum number of linearly independent row (or column) vectors of  $J$ .
- $\text{Rank}(J) = m \Leftrightarrow J$  has (after deleting suitable rows and/or columns) an  $(m \times m)$ -submatrix  $\hat{J}$  that is regular, i.e.  $\det(\hat{J}) \neq 0$ , and there exists no  $(m+1) \times (m+1)$ -submatrix of  $J$  that is regular.

**Definition 4.5**  $m$  shall be the maximum rank of  ${}^0J_n(\mathbf{q})$  for all  $\mathbf{q} \in \mathcal{A}$  ( $m \leq \min\{n, 6\}$ ). A joint configuration  $\mathbf{q}_s \in \mathbb{R}^n$  with

$$\text{rank}({}^0J_n(\mathbf{q}_s)) < m$$

is called **kinematic singularity or singular configuration** (engl. *singularity, singular configuration*).

#### One possible approach for determining kinematic singularities

- a) In general, one determines all configurations  $\mathbf{q}_s$  that lead to a rank loss in the Jacobian matrix  ${}^0J_n(\mathbf{q})$ . That is, one considers e.g. all possible combinations of a maximum number of linearly independent columns (or rows) and investigates for which configurations  $\mathbf{q}_s$  these become linearly dependent.

b) The determination of candidates for singularities can be performed by investigating suitable square submatrices, e.g.

1.) Restrict  $(6 \times n)$ -matrix  ${}^0J_n(\mathbf{q})$  to a suitable  $(m \times m)$ -submatrix  ${}^0\hat{J}_n(\mathbf{q})$  with (cf. INV JAC for the planar SCARA robot page 78):

- $m = \max_{\mathbf{q}} \text{rank}({}^0J_n(\mathbf{q}))$
- $m \leq \min\{n, 6\}$
- $m = \text{dimension of the manipulation space, i.e. dimension of the reachable, independent positions and orientations.}$

**Example: 2-DOF SCARA**

$m = 2$ , only arbitrary  $x_0, y_0 \in \mathcal{A}$  reachable,  $z_0$  and arbitrary orientation not.

- Interpretation: Only  $m$  rows of  ${}^0J_n$  are used, which correspond to degrees of freedom of the manipulator space.

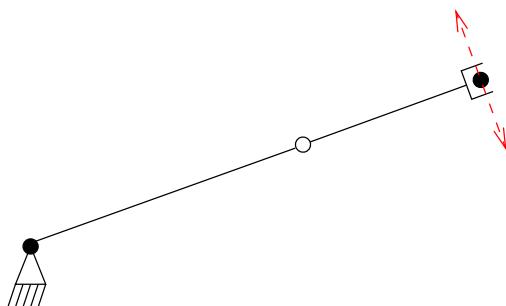
2.)  $\mathbf{q}_s$  is kinematic singularity  $\iff \text{rank}({}^0\hat{J}_n(\mathbf{q}_s)) < m \iff \det({}^0\hat{J}_n(\mathbf{q}_s)) = 0$ .

**Example: Kinematic Singularities of the 2-DOF SCARA Manipulator**

$$\begin{aligned} m &= \max_{\mathbf{q} \in \mathcal{A}} \text{rank}({}^0J_n(\mathbf{q})) = 2 \\ \text{rank}({}^0J_n(\mathbf{q}_s)) < 2 &\Leftrightarrow \det({}^0\hat{J}_2(\theta_1, \theta_2)) = 0 \\ &\Leftrightarrow l_1 l_2 \sin(\theta_2) = 0 \\ &\Leftrightarrow \theta_2 = 0 \quad \text{or} \quad \pm 180^\circ \quad (\text{or } \pm \pi) \\ &\quad \text{kinematic singularities} \end{aligned}$$

**Physical interpretations:**

$\theta_2 = 0^\circ$ : Arm is fully extended:

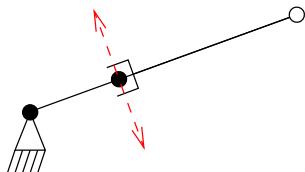


$\Rightarrow$  Motion is now only possible in one Cartesian direction, namely orthogonal to the arm.

$\Rightarrow$  Local loss of one degree of freedom of motion!

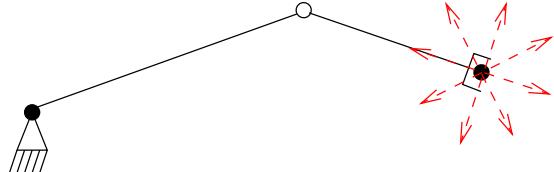
$\theta_2 = 180^\circ$ :

Arm is fully folded:



$\Rightarrow$  Motion is again locally only possible in one Cartesian direction.

$\theta_2 \neq 0, \pm 180^\circ$  : Arm in general configuration



$\Rightarrow$  Motion is locally possible in every Cartesian  $(x, y)$ -direction.

Two categories of kinematic singularities are distinguished:

- (i) **at the boundaries of the (three-dimensional Cartesian) workspace** (workspace boundary singularities): These inevitably occur (mostly with revolute joints), when the robot is fully extended or retracted. However, they can simply be avoided by moving the robot only strictly inside and not on the boundaries of the (three-dimensional Cartesian) workspace.
- (ii) **in the interior of the (three-dimensional) workspace** (workspace interior singularities): These are frequently caused by „identical alignments“ of two or more joints. They are more difficult to avoid, since they can occur somewhere within the reachable workspace.

### Example: Typical Industrial Robot with 6 Revolute Joints

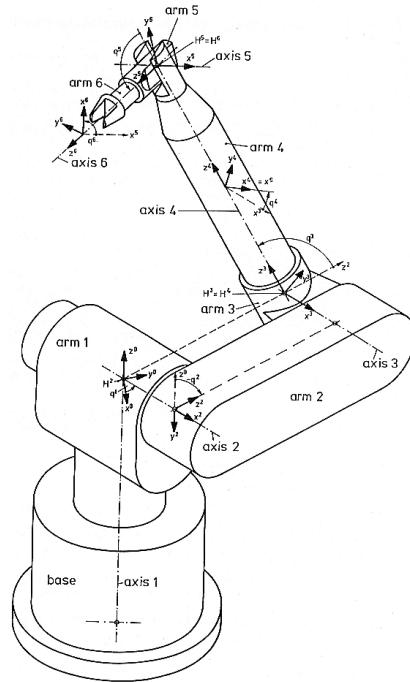
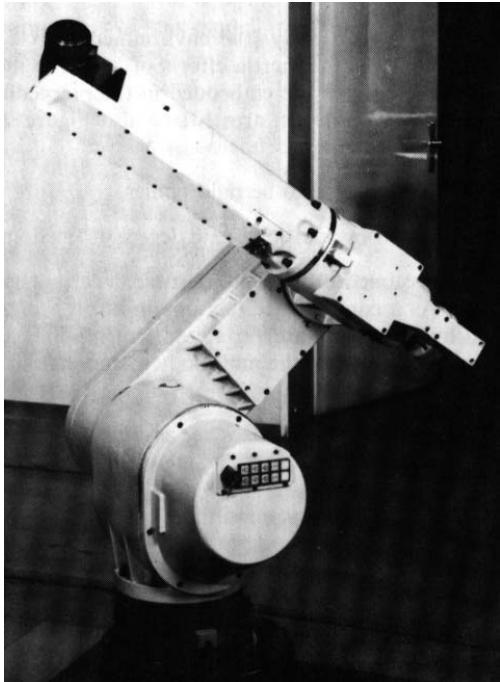
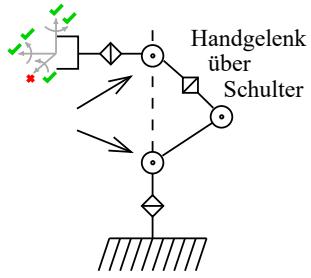


Figure: Manutec r3 Robot<sup>3</sup>

<sup>3</sup>Source: M. Otter, S. Türk: *The DFVLR Models 1 and 2 of the Manutec r3 Robot*. Report DFVLR-Mitteilung 88-13, Institut für Robotik und Systemdynamik, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Oberpfaffenhofen (1988).

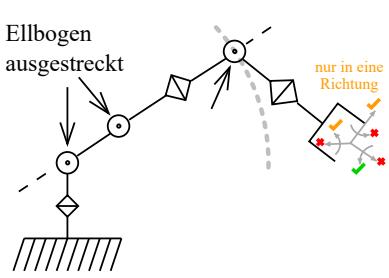
a) Shoulder singularity:



Bewegung orthogonal zur Oberarm-Unterarm-Ebene nicht mehr möglich.

∅ Drehgelenk in Bildebene

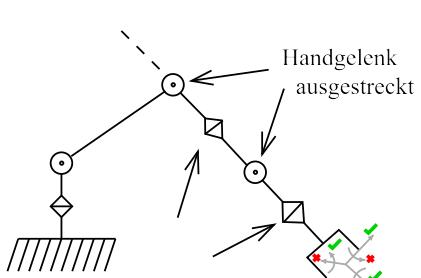
b) Elbow singularity:



Handgelenk am Rand seines Arbeitsbereichs. Daher eingeschränkte Bewegungsmöglichkeit.

◊ Drehgelenk in Ebene orthogonal zur Bildebene

c) Wrist singularity:

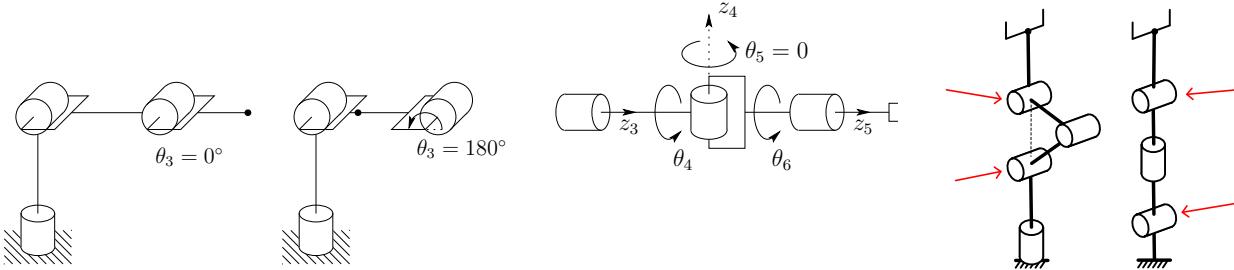


Erste und dritte Handgelenksachse ( $z_4, z_6$ ) fallen zusammen, daher Rotation um  $z_5 \times z_6$  und Bewegung entlang  $z_5$  nicht möglich.

**Remark:** a) and c) can also occur in the interior of the 3D manipulation space, b) only at the boundary, since the third joint must be fully extended (cf. [Yoshikawa, 1990], Section 2.5.6).

**Remark:** In some manipulators (e.g. typical 6-DOF revolute-joint industrial robots) arm singularities (loss of positioning degrees of freedom) can be investigated separately from wrist singularities (loss of orientation degrees of freedom) due to design (see next example).

### Further Examples of Kinematic Singularities



Left: Elbow singularities of the 6-DOF robot from page 81 (Source: Fig. 5.5 from [Spong/Hutchinson/Vidyasagar, 2006])

Center: Spherical wrist singularity with linearly dependent revolute joint axes  $z_3, z_4, z_5$  (Source: Fig. 5.3 from [Spong/Hutchinson/Vidyasagar, 2006])

Right: Further examples of singularities (Source: S. Kurowski).

Already the proximity of singular configurations must be avoided, since for real robots only limited joint velocities are possible, i.e. technical constraints of the form apply

$$\dot{q}_{i,\min} \leq \dot{q}_i(t) \leq \dot{q}_{i,\max}, \quad i = 1, \dots, n,$$

with real constants  $\dot{q}_{i,\min}$ ,  $\dot{q}_{i,\max}$ , whose values depend on the respective robot hardware.

### Possibilities for Avoiding Kinematic Singularities

- 1.) Careful trajectory planning (i.e., „circumnavigating“ the kinematic singularities)

## 2.) Modification of INV JAC[q] by relaxation

**Example:**  $n = 6$ :

$$\begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} = {}^0J_n(\mathbf{q}) \cdot \dot{\mathbf{q}} \quad \Rightarrow \quad \dot{\mathbf{q}} = \left( {}^0J_n(\mathbf{q}) \right)^{-1} \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} \quad (\text{if } {}^0J_n \text{ regular})$$

The hard requirement

$$\dot{\mathbf{q}} = \left( {}^0J_n(\mathbf{q}) \right)^{-1} \cdot \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix}$$

is relaxed to

$$\min_{\dot{\mathbf{q}}} \left( \underbrace{\left\| \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix} - {}^0J_n(\mathbf{q}) \cdot \dot{\mathbf{q}} \right\|^2}_{\text{equation error}} + \underbrace{\lambda^2 \cdot \|\dot{\mathbf{q}}\|^2}_{\substack{\text{penalty term,} \\ \lambda = \text{const} \geq 0}} \right)$$

The solution to the quadratic minimization problem is

$$\dot{\mathbf{q}}_\lambda = \left( {}^0J_n^T(\mathbf{q}) \cdot {}^0J_n(\mathbf{q}) + \lambda^2 E \right)^{-1} \cdot {}^0J_n^T(\mathbf{q}) \cdot \begin{pmatrix} {}^0\mathbf{v}_n \\ {}^0\omega_n \end{pmatrix}$$

and depends on  $\lambda$ .

$\lambda = 0$ : yields the original solution.

$\lambda > 0$ : an increase in  $\lambda$  causes a significant circumnavigation of kinematic singularities, but also a larger trajectory error.

## 3.) Consideration of kinematic singularities already in the mechanical design:

- „Relocation“ of kinematic singularities to uninteresting regions of the workspace,
- Adding one or more degrees of freedom of motion to avoid kinematic singularities by exploiting redundancy.

## Possibility for Dealing with „Unavoidable“ Kinematic Singularities

1.) Identification of the affected degrees of freedom and joints, i.e. determination of all  $\mathbf{q}_s$

2.) Removal of those rows from  ${}^0J_n(\mathbf{q})$  that correspond to the „lost“ Cartesian degrees of freedom, and those columns that correspond to the respective joints.

$\Rightarrow$  The Jacobian matrix  ${}^0\tilde{J}_n(\mathbf{q})$  reduced in this way has „full“ rank.

3.) When traversing a kinematic singularity:

- Keep the singular joints or their velocities constant and
- Calculate the velocities of the remaining joints with  ${}^0\tilde{J}_n(\mathbf{q})$ .

## Summary of Difficulties and Properties of Singular Configurations

- Kinematic singularities mean locally the loss of degrees of freedom of motion.
- Limited velocities at the end-effector can correspond to unlimited joint velocities  $\dot{q}$ .
- Limited forces and moments at the end-effector can correspond to unlimited joint forces and moments.
- Kinematic singularities are usually at the boundary of the 3D workspace, sometimes also in the interior.
- Kinematic singularities correspond to points in the workspace that with small changes in the geometric parameters of the robot links may no longer be reachable.
- It can therefore happen that for spatial arrangements in the vicinity of a kinematic singularity, no solution to the inverse kinematics problem exists at all.

**Discussion of kinematic singularities in textbooks:** [Spong/Vidyasagar, 1989], Section 5.3, [Yoshikawa, 1990], Section 2.5.6, and [McKerrow, 1991], Section 5.6.

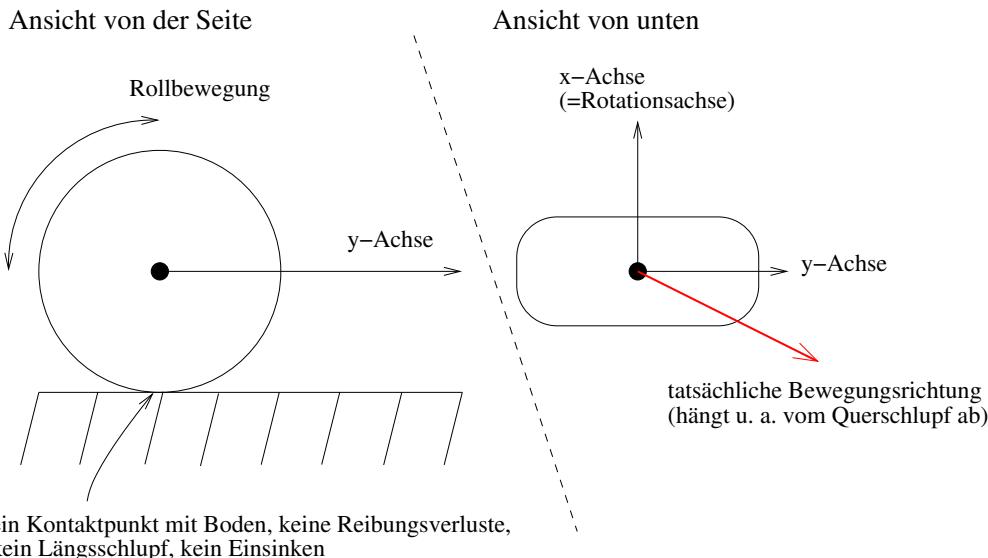
## 4.5 Non-holonomic Kinematics of Multi-wheeled Vehicles

### 4.5.1 Preliminary Remarks

#### Odometry (engl. odometry, dead reckoning):

Odometry refers to the calculation of position and orientation (pose) of a vehicle solely from measurements of wheel or leg movements with internal sensors (and by means of the forward kinematic solution).

#### Idealized Wheel (Pure Rolling):



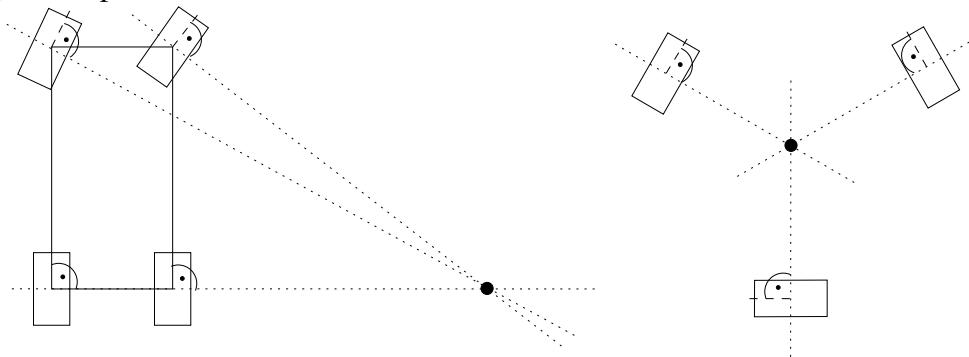
#### Odometry Example: Single-wheeled Vehicle<sup>4</sup>

The distance traveled by a single, ideal, freely movable wheel is:

$$2 \cdot \pi \cdot \text{wheel radius} \cdot \text{number of revolutions} .$$

#### Multi-wheeled Vehicles:

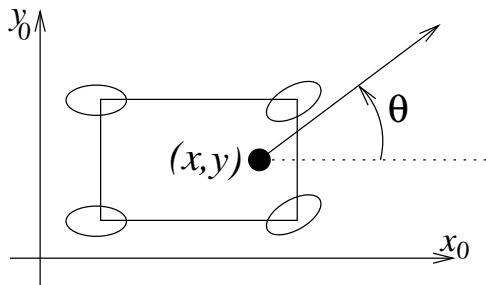
For locomotion, **with pure rolling** the wheel normals (= axes of rolling motions) must have a common intersection point, the **instantaneous center of rotation** (instantaneous center of curvature/rotation, ICC/ICR), which in the case of parallel wheel normals can also be at infinity. Examples:



**Consequence:** The angles of the wheel steering (orientations) must be coordinated for locomotion on wheels. In addition, the velocity of each individual wheel must be consistent with the rotational velocity of the entire vehicle around the ICC.

<sup>4</sup>The odometry for a multi-wheeled vehicle is (of course) more complicated than for a single wheel.

### Three Pose Degrees of Freedom of a (Point-like) Vehicle on a Plane



Position (coordinates):  $(x, y)$   
 Orientation (angle):  $\theta$

} vehicle pose  
 } (engl. pose)

The pose of each individual wheel can be described by these three degrees of freedom as well as the three pose degrees of freedom of the vehicle with respect to a suitable, fixed vehicle reference point (see figure above).

In contrast to many manipulators, for vehicles with different numbers and arrangements of wheels, the three pose parameters per wheel **cannot** be independently controlled<sup>5</sup>. An example from everyday life is parallel parking a four-wheeled car. For example, there are driven or freely rolling (caster wheels) and actively steerable or passively trailing wheels. In contrast to most manipulators, therefore, not all degrees of freedom are actuated in vehicles.

Furthermore, the pose degrees of freedom of the wheels during locomotion generally must satisfy constraints such as the common intersection point of the wheel normals or the non-lifting of individual wheels. Therefore, in vehicles there generally exist so-called **non-holonomic constraints** between the individual pose degrees of freedom  $\mathbf{q}$  in the form

$$F(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \dots) = 0,$$

i.e. restrictions of the admissible position-velocity range. This can mean, for example, that the vehicle can move in some, but not in all directions (cf. parking).

⇒ Motion planning (trajectory planning) for mobile robots becomes complicated by this!

In trajectory planning, a point-like vehicle is often used for simplification, and the vehicle properties are only approximately considered as constraints (e.g. minimum turning circle). Optimal trajectory planning with a vehicle kinematics problem as a constraint leads mathematically to an infinite-dimensional optimization problem (a so-called optimal control problem (optimal control problem) with the differential equations of vehicle kinematics as constraints).

**Remark:** Constraints of the form  $F(\mathbf{q}) = 0$  (or  $> 0$ ) are called **holonomic**, e.g. static obstacles in manipulator motions.

**Examples:** The holonomic condition

$$F(\mathbf{q}) := \sqrt{({}^0r_{n,x}(\mathbf{q}))^2 + ({}^0r_{n,y}(\mathbf{q}))^2} - 1.5 > 0$$

describes that the  $x$ - and  $y$ -coordinates of the end-effector of a manipulator (or the origin of  $S_n$ ) around the  $z$ -axis of the base coordinate system  $S_0$  should maintain a cylindrical distance of at least radius 1.5.

---

<sup>5</sup>An exception are omnidirectional wheels (omni wheels).

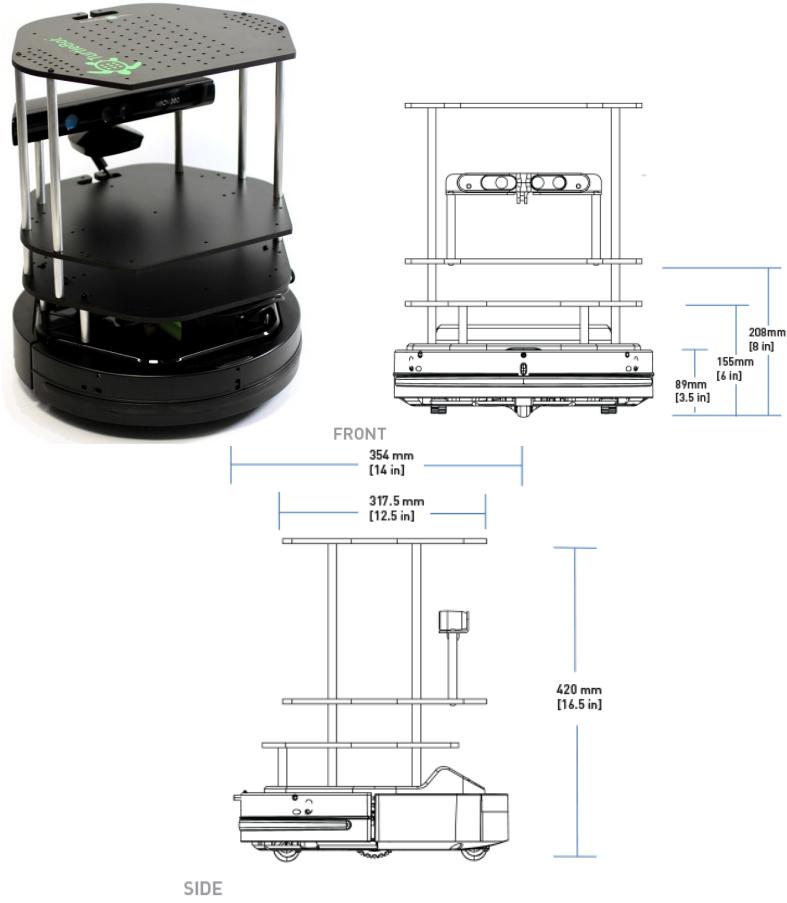
A non-holonomic constraint for a manipulator contains a constraint on the velocities. For example, the non-holonomic condition

$$F(\mathbf{q}, \dot{\mathbf{q}}) := \|{}^0\mathbf{v}_n(\mathbf{q}, \dot{\mathbf{q}})\| - 3 = \|{}^0J_{n,v}(\mathbf{q}) \cdot \dot{\mathbf{q}}\| - 3 \leq 0$$

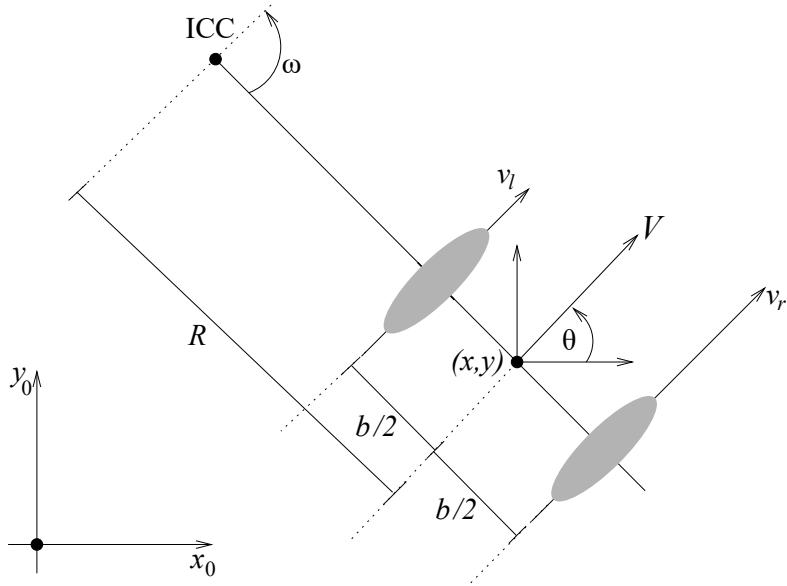
describes that the magnitude of the linear velocity of the end-effector (or the origin of  $S_n$ ) must not be greater than 3.

**Remark:** Analogous to the DH conventions for manipulators, detailed kinematic chains can be modeled for vehicles that consider the 3D geometry (see McKerrow, Ch. 8.1, pp. 402-421). This chapter is limited to the usual simplified representation: the projection onto the  $(x, y)$ -plane (the ground) and the consideration of three pose degrees of freedom of point-like vehicles  $(x, y, \theta)$ .

## 4.5.2 Differential Drive Multi-wheeled Vehicles



Differential drive multi-wheeled vehicles have two separately driven and non-steerable wheels connected by a rigid axle and a support mechanism. Common are one (e.g. Pioneer P3-DX, Roomba) or two (e.g. the TurtleBot shown above) free-rolling passive caster wheels (engl. passive caster wheels) or no caster wheel and instead an active balance control (e.g. Segway).



**Principle:** By varying the relative velocities  $v_l, v_r$  of the two wheels, rotational motions are generated around the respective instantaneous center of rotation (ICC) and thereby different driving trajectories. During motion, the vehicle (and thus also both wheels) must rotate with **consistent angular velocity**  $\omega(t)^6$  at distance  $R(t)$  around the ICC.

The general relationship

$$\omega(t) R(t) = V(t)$$

applied to the left and right wheel yields

$$\begin{aligned} \omega(t)(R(t) + b/2) &= v_r(t) \\ \omega(t)(R(t) - b/2) &= v_l(t) \\ \Rightarrow v_l + v_r &= 2\omega R \end{aligned}$$

How do  $R$  and  $\omega$  depend on the control variables  $v_l$  and  $v_r$  of the vehicle?

Solving for  $R$  and  $\omega$  yields:

$$R(t) = \frac{b}{2} \cdot \frac{v_r(t) + v_l(t)}{v_r(t) - v_l(t)}, \quad \omega(t) = \frac{v_r(t) - v_l(t)}{b}.$$

Special cases:

- $v_r = v_l \Rightarrow R = \infty$ , i.e. motion is straight driving.
- $v_r = -v_l \Rightarrow R = 0$ , i.e. motion is rotation in place.

**Advantages** of this drive:

- inexpensive production thanks to simple construction,
- suitable for tight environments in buildings, since maneuverable in the smallest space.

---

<sup>6</sup>In this section 4.5,  $\omega$  denotes the scalar angular velocity of rotation around the instantaneous center of rotation (and not a three-dimensional vector as in the previous sections).

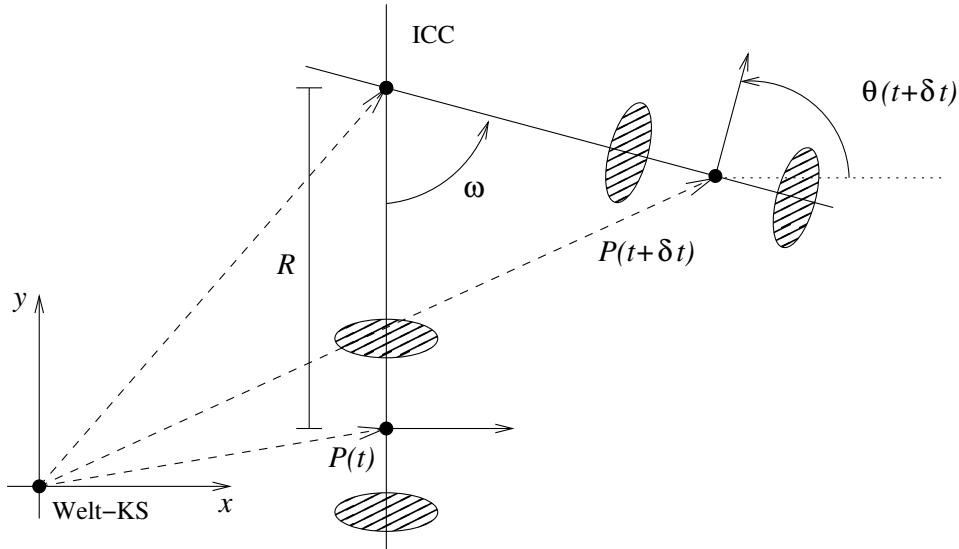
**Disadvantages** of this drive:

- Sensitivity to disturbances and measurement errors in  $v_l, v_r$ :  
Already small changes cause not only a different vehicle velocity, but also a different trajectory shape, i.e. a different trajectory.
- Due to the caster wheel construction, the vehicle reacts sensitively to ground unevenness.
- Thus, for this drive, pure odometry methods deliver inaccurate values already after very short running time due to design (and indeed worse than for other vehicles)!

### 4.5.3 General Forward Kinematics Problem for Vehicles

The forward kinematics problem for (robot) vehicles refers to the determination of the vehicle's own pose  $(x(t), y(t), \theta(t))$  for  $0 \leq t \leq t_f$  from the controllable variables (e.g.  $v_l(t)$  and  $v_r(t)$ ) or, equivalently, from  $R(t)$  and  $\omega(t)$ .

In general, for a vehicle that rotates around its ICC at distance  $R$  with constant angular velocity  $\omega$ :



$$P(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \quad \text{ICC}(t) = \begin{pmatrix} x(t) - R(t) \sin \theta(t) \\ y(t) + R(t) \cos \theta(t) \end{pmatrix}$$

The vector  $(P(t) - \text{ICC}(t)) \in \mathbb{R}^2$  is rotated around  $\text{ICC}(t)$  in the time interval  $\delta t$  with the angle  $\theta(t + \delta t) - \theta(t) = \omega \cdot \delta t$  at constant angular velocity  $\omega$ . From this and with  $\text{ICC}(t + \delta t) = \text{ICC}(t)$  follows (for  $\omega = \text{const}$ ):

$$P(t + \delta t) - \text{ICC}(t) = \begin{pmatrix} \cos(\omega \cdot \delta t) & -\sin(\omega \cdot \delta t) \\ \sin(\omega \cdot \delta t) & \cos(\omega \cdot \delta t) \end{pmatrix} \cdot (P(t) - \text{ICC}(t))$$

For different types of vehicles, different expressions for  $\omega$  and  $R$  enter here, describing their respective dependence on the controllable variables!

From this general relationship, the equations of forward kinematics for a general, point-like vehicle can be obtained with the following considerations for the limit case  $\delta t \rightarrow 0$  ( $\theta = \theta(t)$ ):

$$\begin{aligned}
P(t + \delta t) - \text{ICC}(t) &= \begin{pmatrix} \cos(\omega \cdot \delta t) & -\sin(\omega \cdot \delta t) \\ \sin(\omega \cdot \delta t) & \cos(\omega \cdot \delta t) \end{pmatrix} \cdot (P(t) - \text{ICC}(t)) \\
\Leftrightarrow \begin{pmatrix} x(t + \delta t) - x(t) + R \sin \theta \\ y(t + \delta t) - y(t) - R \cos \theta \end{pmatrix} &= \begin{pmatrix} \cos(\omega \cdot \delta t) & -\sin(\omega \cdot \delta t) \\ \sin(\omega \cdot \delta t) & \cos(\omega \cdot \delta t) \end{pmatrix} \cdot \begin{pmatrix} R \sin \theta \\ -R \cos \theta \end{pmatrix} \\
&= \begin{pmatrix} \cos(\omega \cdot \delta t) R \sin \theta + \sin(\omega \cdot \delta t) R \cos \theta \\ \sin(\omega \cdot \delta t) R \sin \theta - \cos(\omega \cdot \delta t) R \cos \theta \end{pmatrix} \\
\Leftrightarrow \frac{1}{\delta t} \begin{pmatrix} x(t + \delta t) - x(t) \\ y(t + \delta t) - y(t) \end{pmatrix} &= \frac{1}{\delta t} \begin{pmatrix} \sin(\omega \cdot \delta t) R \cos \theta \\ \sin(\omega \cdot \delta t) R \sin \theta \end{pmatrix} + \frac{1}{\delta t} \begin{pmatrix} -\cos(\omega \cdot \delta t) R \sin \theta - R \sin \theta \\ -\cos(\omega \cdot \delta t) R \cos \theta + R \cos \theta \end{pmatrix} \\
&= \frac{\sin(\omega \cdot \delta t)}{\delta t} \begin{pmatrix} R \cos \theta \\ R \sin \theta \end{pmatrix} + \frac{1 - \cos(\omega \cdot \delta t)}{\delta t} \begin{pmatrix} -R \sin \theta \\ R \cos \theta \end{pmatrix}
\end{aligned}$$

Because of

$$\begin{aligned}
\frac{\sin(\omega \cdot \delta t)}{\delta t} &= \frac{1}{\delta t} \left( \omega \cdot \delta t - \frac{(\omega \cdot \delta t)^3}{3!} + \frac{(\omega \cdot \delta t)^5}{5!} - \dots \right) = \omega - \frac{\omega^3 \cdot \delta t^2}{3!} + \frac{\omega^5 \cdot \delta t^4}{5!} - \dots \\
\frac{1 - \cos(\omega \cdot \delta t)}{\delta t} &= \frac{1}{\delta t} \left( 1 - 1 + \frac{(\omega \cdot \delta t)^2}{2!} - \frac{(\omega \cdot \delta t)^4}{4!} - \dots \right) = \frac{\omega^2 \cdot \delta t}{2!} - \frac{\omega^4 \cdot \delta t^3}{4!} - \dots
\end{aligned}$$

it follows

$$\lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \begin{pmatrix} x(t + \delta t) - x(t) \\ y(t + \delta t) - y(t) \end{pmatrix} = \omega \begin{pmatrix} R \cos \theta \\ R \sin \theta \end{pmatrix} + 0.$$

### Forward Kinematics of a Point-like Vehicle

Thus, in general, for the reference point of a point-like vehicle that moves in instantaneous direction  $\theta(t)$  with instantaneous velocity  $V(t) = \omega(t) \cdot R(t) \in \mathbb{R}$ :

$$\begin{aligned}
\dot{x}(t) &= V(t) \cdot \cos \theta(t) \\
\dot{y}(t) &= V(t) \cdot \sin \theta(t) \\
\dot{\theta}(t) &= \omega(t).
\end{aligned}$$

The linear velocity  $V(t) \in \mathbb{R}$  (or because of  $V(t) = \omega(t) \cdot R(t)$  the distance  $R(t) \in \mathbb{R}$  of the vehicle reference point to the center of rotation) and the angular velocity  $\omega(t) \in \mathbb{R}$  depend on the kinematics and the controllable variables of the drive type of the respective vehicle type. For example, for a differential drive robot  $R(t) = \frac{l}{2} \cdot \frac{v_r(t) + v_l(t)}{v_r(t) - v_l(t)}$  and  $V(t) = \frac{1}{2}(v_r(t) + v_l(t))$ .

For a given initial pose

$$\begin{pmatrix} x(0) \\ y(0) \\ \theta(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ \theta_0 \end{pmatrix}$$

and given functions

$$V(t), \omega(t), \quad 0 \leq t \leq t_f$$

one obtains the resulting trajectory

$$\begin{pmatrix} x(t) \\ y(t) \\ \theta(t) \end{pmatrix}, \quad 0 \leq t \leq t_f$$

(with pure rolling) by (numerical) solution of the initial value problem:

$$(*1) \left\{ \begin{array}{l} \dot{x}(t) = V(t) \cdot \cos \theta(t), \quad x(0) = x_0, \\ \dot{y}(t) = V(t) \cdot \sin \theta(t), \quad y(0) = y_0, \\ \dot{\theta}(t) = \omega(t), \quad \theta(0) = \theta_0 \end{array} \right.$$

or equivalently formulated as

$$(*2) \left\{ \begin{array}{l} x(t) = x_0 + \int_0^t V(\tau) \cdot \cos \theta(\tau) d\tau, \\ y(t) = y_0 + \int_0^t V(\tau) \cdot \sin \theta(\tau) d\tau, \\ \theta(t) = \theta_0 + \int_0^t \omega(\tau) d\tau. \end{array} \right.$$

For the differential drive three-wheeled vehicle:

$$\begin{aligned} V(t) &= \frac{1}{2}(v_l(t) + v_r(t)), \\ \omega(t) &= \frac{1}{b}(v_r(t) - v_l(t)). \end{aligned}$$

### Conclusion:

Equations (\*1) or (\*2) describe the forward kinematics problem for the vehicle reference point, how in general from given linear velocity  $V(t)$  and given rotational velocity  $\omega(t)$  (or in the special case of the differential drive three-wheeled vehicle from  $v_l(t)$  and  $v_r(t)$ ) the trajectory of the vehicle  $(x(t), y(t), \theta(t))^T$ ,  $0 \leq t \leq t_f$ , can be determined.

The reverse question, how the control variables  $v_l(t)$  and  $v_r(t)$  (or  $V(t)$  and  $\omega(t)$ ) are determined so that the vehicle follows a given trajectory  $(\tilde{x}(t), \tilde{y}(t), \tilde{\theta}(t))^T$ ,  $0 \leq t \leq t_f$ , leads to the inverse kinematics problem.

#### 4.5.4 Inverse Kinematics Problem

The constraints (\*1) or (\*2) are non-holonomic and are therefore generally difficult and complex (i.e. solvable only with suitable numerical calculation methods).

Given is the target pose of the vehicle  $(x(t_f), y(t_f), \theta(t_f))$  after a driving time  $t_f$  and the initial pose  $(x(0), y(0), \theta(0))$ :

$$\begin{aligned} x(t_f) &= x_0 + \int_0^{t_f} V(\tau) \cdot \cos \theta(\tau) d\tau, \\ y(t_f) &= y_0 + \int_0^{t_f} V(\tau) \cdot \sin \theta(\tau) d\tau, \\ \theta(t_f) &= \theta_0 + \int_0^{t_f} \omega(\tau) d\tau. \end{aligned}$$

For a general, point-like vehicle, it must be determined with which  $V(t)$  and  $\omega(t)$ ,  $0 \leq t \leq t_f$ , the target pose can be reached. It must be noted that  $V(t)$  and  $\omega(t)$  cannot be chosen arbitrarily, but depend on the specific kinematics of the respective vehicle.

For the differential drive three-wheeled vehicle, the equations to be solved are:

$$\begin{aligned}x(t_f) &= x_0 + \int_0^{t_f} \frac{1}{2}(v_l(\tau) + v_r(\tau)) \cdot \cos \theta(\tau) d\tau, \\y(t_f) &= y_0 + \int_0^{t_f} \frac{1}{2}(v_l(\tau) + v_r(\tau)) \cdot \sin \theta(\tau) d\tau, \\ \theta(t_f) &= \theta_0 + \int_0^{t_f} \frac{1}{b}(v_r(\tau) - v_l(\tau)) d\tau.\end{aligned}$$

These must be solved for the wheel velocities  $v_l(t)$  and  $v_r(t)$ ,  $0 \leq t \leq t_f$ .

For the differential drive three-wheeled vehicle, forward and inverse kinematics problems can only be solved explicitly in special cases, i.e. without using numerical methods (see exercise).

**Remark:** Vehicle paths or vehicle trajectories are usually planned as (smooth) connections of simple segments such as straight line sections and circular arcs, for which the forward or inverse kinematics problem can often be solved explicitly as special cases.

**Remark:** A possible mathematical formulation of the inverse kinematics problem for vehicles as an infinite-dimensional, constrained optimization problem<sup>7</sup> is:

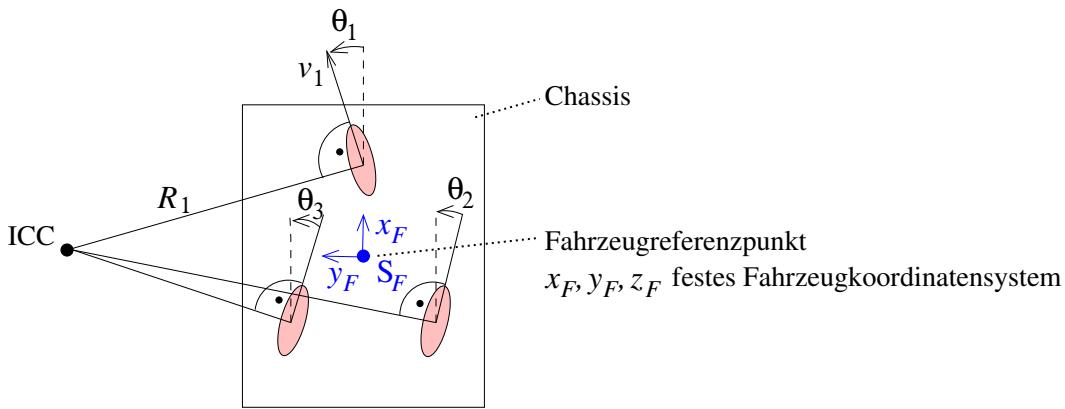
$$\min_{V, \omega} \int_0^{t_f} \left\| \begin{pmatrix} \tilde{x}(t) - x(t) \\ \tilde{y}(t) - y(t) \\ \tilde{\theta}(t) - \theta(t) \end{pmatrix} \right\|_2^2 dt$$

under the constraints (\*1) or (\*2). Here  $\tilde{x}(t), \tilde{y}(t), \tilde{\theta}(t)$ ,  $0 \leq t \leq t_f$ , denote the specified or desired trajectory of the vehicle.

#### 4.5.5 Omnidirectional Three-wheeled Kinematics

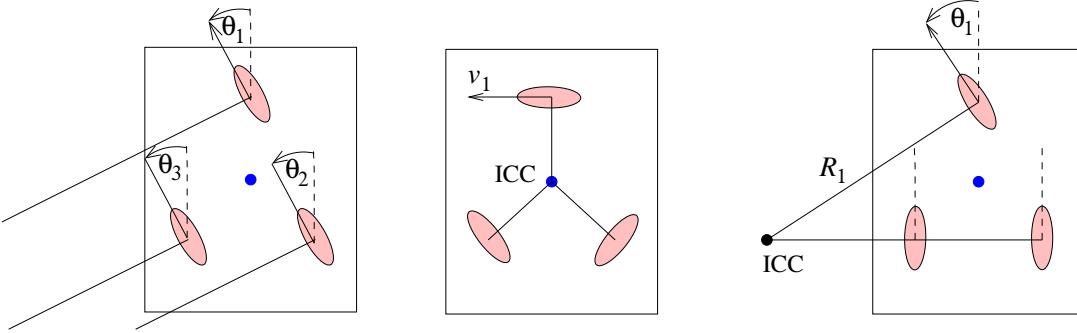
A non-holonomic but omnidirectional vehicle with three steerable wheels is considered. Only the front wheel is driven.

For pure rolling, an instantaneous center of rotation ICC is again necessary, where the three wheel normals intersect.



<sup>7</sup>see lecture „Optimization of Static and Dynamic Systems“

Some possible motion patterns are:



a) Straight-sideways motion with parallel steering:  
 $\theta_1 = \theta_2 = \theta_3$ ,  
wheel normals parallel,  
ICC at infinity

b) Rotation in place around vertical axis  $z_F$ :  
ICC = origin of  $S_F$ ,  
 $\theta_1 = \pm 90^\circ (\pm \pi)$

c) Left turn with fixed, parallel rear wheels,  
 $\theta_2 = \theta_3 = 0^\circ$

A possible steering coordination strategy through restriction of steering freedom is the choice of

- $\theta_1, R_1, v_1$  as control variables and
- $\theta_2$  and  $\theta_3$  as dependent variables, which should be tracked as functions of  $\theta_1, R_1$  and other parameters to the control variables as accurately as possible.

#### 4.5.6 Other Drive Types of Wheeled Vehicles

Other drive types of vehicles with wheels are e.g.

- Synchronous drive: three steerable and driven wheels (e.g. IS Robotics B21),
- Three-wheeled kinematics: one driven and steerable front wheel, two passive, parallel rear wheels,
- four-wheeled „automobile“ kinematics (Ackermann steering): two steerable front wheels and two parallel rear wheels, either front wheels or rear wheels or all four wheels driven.

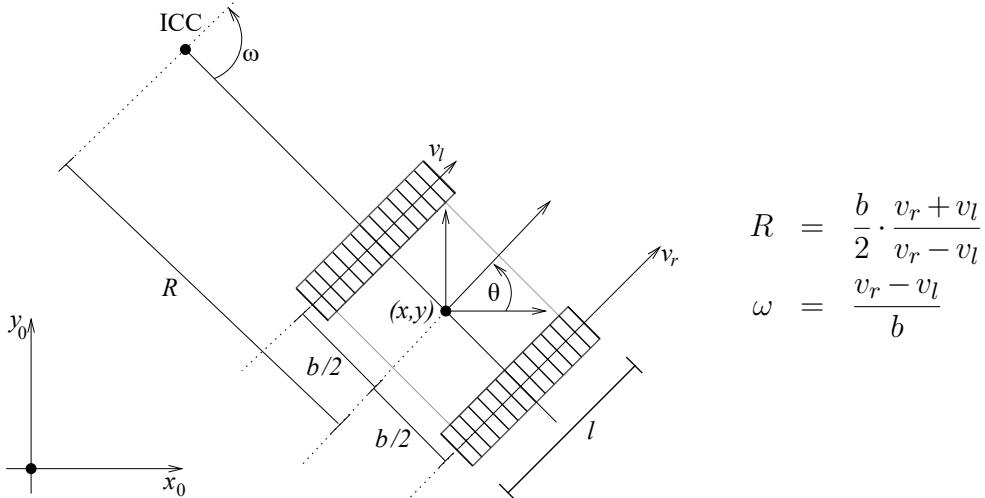
Literature: see exercises as well as [Dudek/Jenkin, 2000] (Chapter 2), [McKerrow, 1991] (Chapter 8) and [Siegwart, Nourbakhsh, Scaramuzza, 2011] (Chapter 3).

#### 4.5.7 Tracked Vehicles<sup>8</sup>

Compared to multi-wheeled vehicles, a track drive offers a significantly larger contact area with the ground, whereby higher traction can be achieved. Thus, tracked vehicles are particularly suitable for difficult terrain and are used, among other things, in the field of rescue robotics or remote inspection of industrial facilities. The common construction variant considered here consists of two closed tracks on the sides of the chassis.

<sup>8</sup>This section is based on work by Jérôme Kirchhoff and Martin Oehler.

The increased contact area changes the steering capability of the vehicle. Thus, a rotational motion is only possible with simultaneous slipping (engl. *slip*) or skidding (engl. *skid*) of the track drive. This control method is also called *skid steering*. To determine the necessary control variables, the two drive tracks are simplified as wheels. This makes it possible to use the known relationships of differential drive vehicles for control.

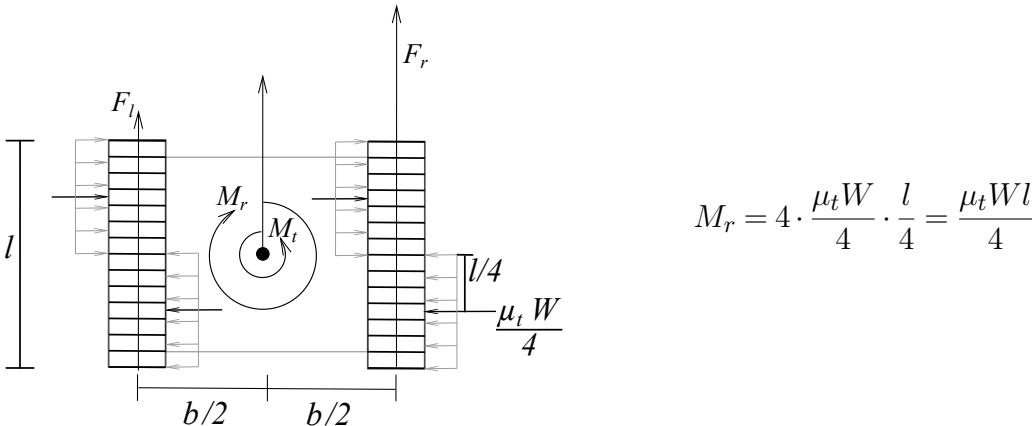


$$R = \frac{b}{2} \cdot \frac{v_r + v_l}{v_r - v_l}$$

$$\omega = \frac{v_r - v_l}{b}$$

Through the tracks, drive forces  $F_l$ ,  $F_r$  are transmitted to the ground. For a rotation, the torque  $M_t = \frac{b}{2} \cdot (F_r - F_l)$  generated by this must overcome the opposing rotational resistance  $M_r$  of the ground, otherwise no rotation is possible.

The rotational resistance is calculated from the lateral resistance factor  $\mu_t$  (caused by ground- and track-dependent friction), the weight force of the vehicle  $W$ , and the length of the (here assumed symmetric) contact  $l$  of the two tracks with the ground.



$$M_r = 4 \cdot \frac{\mu_t W}{4} \cdot \frac{l}{4} = \frac{\mu_t W l}{4}$$

To perform a rotation,  $M_t \geq M_r$  must hold. However,  $M_t$  cannot become arbitrarily large, since the force that can be transmitted to the ground through the tracks is limited by the nature of the ground and properties of the vehicle. If the drive force is too high, slipping can occur. The ratio  $l/b$  between the length of the contact surfaces of the tracks  $l$  and the distance of the track centers  $b$  is decisive for this. If this ratio is too large, the tracks slip before a rotation can be generated ( $M_t < M_r$ ). A lower ratio leads to better mobility, since less force must be applied to perform the rotation, whereby the remaining force that can be generated remains available for forward motion. Longer tracks, on the other hand, are better suited for climbing stairs.

The slipping during rotations, which is particularly difficult to predict accurately in uneven terrain, means that an accurate determination of the trajectory traveled solely from the

control signals is poorly possible (poor odometry), so that better inertial sensors should be used for this purpose. Another disadvantage compared to wheel drive is the greater ground friction, which leads to increased current consumption and wear (track abrasion) compared to wheeled vehicles.

More complex approaches enable more accurate modeling of kinematics, odometry, and dynamics. Further information is available in [Wong, 2001].

## 4.6 Static Forces in Manipulators

The structure of a manipulator linked with rigid joints and links naturally leads to considerations of how forces and moments are transmitted from one robot link to the next. Typically, a manipulator moves something in the environment with the free end of the kinematic chain, the end-effector, or carries a load with the robot hand. The solution to the question of which torques in revolute joints or linear forces in prismatic joints of the manipulator must be applied so that the entire system is in static equilibrium leads to the investigation of static forces. Literature: see e.g. [Craig, 1989], Section 5.9 – 5.11, [Yoshikawa, 1990], Section 2.6, and [McKerrow, 1991], Chapter 6.

# Chapter 5

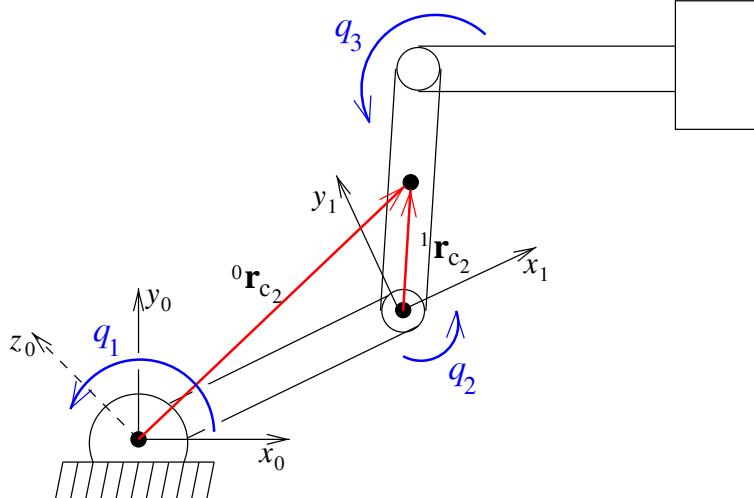
## Roboterdynamik

### 5.1 Einleitung

**Kinetik** ist die Erweiterung der Kinematik durch Untersuchung der bei einer Bewegung wirkenden bzw. der für eine Bewegung erforderlichen Kräfte und Momente.

#### Beispiel: Dreigelenkiger, ebener Manipulator

Zur Veranschaulichung von Unterschieden zwischen Kinematik und Kinetik betrachten wir einen dreigelenkigen, ebenen Manipulator.



Dabei bezeichnen  ${}^0\mathbf{r}_{c_2}$ ,  ${}^1\mathbf{r}_{c_2}$  den Koordinatenvektor des Schwerpunktes des zweiten Robotergliedes bezüglich des Koordinatensystems  $S_0$  bzw.  $S_1$ . Der Index „c“ steht für den Schwerpunkt („center of mass“).

**Kinematisches Robotmodell:** Die Berechnung der (kinematischen) Geschwindigkeiten  $({}^0\mathbf{v}_{c_2}, {}^0\omega_{c_2})$  im Schwerpunkt des zweiten Gliedes hängt **allein** von  $\dot{q}_1$  und  $\dot{q}_2$  und **nicht** von  $\dot{q}_3$  ab. Bei Manipulatoren (mit lediglich holonomen Zwangsbedingungen) erhält man ein **lineares** Differentialgleichungssystem für den Zusammenhang zwischen  $\dot{\mathbf{q}}$  und  $({}^0\mathbf{v}_{c_2}, {}^0\omega_{c_2})$  (siehe Kapitel 4.3 Jacobi-Matrix eines Manipulators).

**Kinetisches Robotmodell:** Durch die Bewegung des dritten Gelenks treten Reaktionsskräfte im zweiten Gelenk auf, die dessen Bewegung mit beeinflussen! Diese sind im kinematischen Modell nicht berücksichtigt. Bisher (d.h. im kinematischen Modell) sind die Roboterglieder als masselos betrachtet worden. Doch unterschiedliche Massen der

Roboterglieder oder unterschiedliche Lasten am Endeffektor führen im Allgemeinen zu unterschiedlichen Bewegungseffekten. Auch die Leistungsfähigkeit eines Gelenkantriebmotors ist bisher nicht berücksichtigt worden. Alle diese Effekte gehen in das kinetische Modell ein.

## Allgemeine Überlegungen

Zur Beschreibung der zeitlichen Entwicklung eines mechanischen Systems unter Berücksichtigung der auftretenden Kräfte und Momente müssen **kinetische** und **potentielle Energie** berücksichtigt werden. In diesem Kapitel wird gezeigt, wie man daraus ein **nicht-lineares** Differentialgleichungssystem zweiter Ordnung erhält, das den Zusammenhang zwischen  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}(t) = \frac{d^2\mathbf{q}}{dt^2}(t)$  und den wirkenden Kräften bzw. Drehmomenten beschreibt. Position  $q_i$ , Geschwindigkeit  $\dot{q}_i$  und Beschleunigung  $\ddot{q}_i$  des  $i$ -ten Gliedes hängen ab vom Drehmoment  $n_i$  bei einem Drehgelenkantrieb bzw. der Kraft  $f_i$  bei einem Schubgelenkantrieb ( $\tau_i$  als Notation für  $n_i$  oder  $f_i$ ) **und** den über die Verkopplung mit den anderen Gelenken und Gliedern wirkenden Kräften und Momenten!

$$\begin{array}{ccc} \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n & \xrightleftharpoons[\text{DIR DYN}]{\text{INV DYN}} & \begin{array}{l} \text{Vektor } \tau \in \mathbb{R}^n \text{ der} \\ \text{Drehmomente (bei Drehgelenken)} \\ \text{bzw. linearen Kräfte (bei Schubgelenken)} \end{array} \end{array}$$

DIR DYN: Vorwärtsdynamikmodell, INV DYN: inverses Dynamikmodell.

Die vollständige kinetische Modellierung eines Roboters ist **wichtig** für

- Steuerung und Regelung mit hohen Güteanforderungen,
- realistische Simulation und Bahnplanung

und ist **schwierig** weil

- ein Roboter aus vielen, über Gelenke gekoppelten, massebehafteten Teilsystemen (Gliedern) besteht,
- weil die verallgemeinerten Kräfte (Trägheits-, Gravitations-, Coriolis-, Zentrifugalkräfte und -Momente) nichtlinear und verkoppelt und außerdem
- abhängig von Manipulatorstellung und manipulierter Last sind,
- und weil darüber hinaus Reibung sowie mögliche (ungewollte oder gewollte) Elastizitäten in Gelenken und Gliedern auftreten können.

Es existieren verschiedene, allgemeine CAE-Systeme zur rechnergestützten Herleitung und Simulation dynamischer Bewegungsgleichungen, z.B. ADAMS, AutoLev, SIMPACK<sup>1</sup>. Die Dynamik von Mechanismen ist ein großes Teilgebiet der Mechanik (Dynamik von Mehrkörpersystemen, engl.: dynamics of multibody systems, MBS dynamics). Wir betrachten hier nur einen Ausschnitt. Viele Robotertypen (Manipulatoren ebenso wie humanoide oder vierbeinige Roboter) sind Mehrkörpersysteme mit Baumstruktur, für die spezielle Methoden der Mehrkörpersystemdynamik angewandt werden können. Für Echtzeit-Anwendungen der kinetischen Robotermodelle zur Online-Regelung und -Bahnplanung

<sup>1</sup>[www.adams.com](http://www.adams.com), [www.autolev.com](http://www.autolev.com), [www.simpack.com](http://www.simpack.com)

liefern allgemeine CAE-Systeme zu langsame Auswertungen, so dass das dynamische RobotermodeLL meist noch „von Hand“ bzgl. der Anzahl notweniger Rechenoperationen optimiert werden muss.

Am Fachgebiet Simulation, Systemoptimierung und Robotik wurde die modulare und quelloffene C++-Bibliothek MBSLIB<sup>2</sup> zur Modellierung und Simulation von Mehrkörpersystemen mit Baumstruktur entwickelt und veröffentlicht. Die Bibliothek enthält effiziente Methoden zur numerischen Berechnung der Vorwärtskinematik sowie Vorwärts- und Inversdynamik. Durch ihre flexible Architektur und die für die Systemstruktur optimierten Berechnungsverfahren eignet sich die Bibliothek auch für die Planung, Steuerung und Regelung von Robotersystemen in Echtzeit. In den folgenden Kapiteln werden bei der Beschreibung der jeweiligen Berechnungsverfahren auch die Bezüge zu den in der MBSLIB implementierten Methoden jeweils kurz erläutert.

Allgemein existieren mehrere, **verschiedene** Methoden zur Herleitung **derselben** Bewegungs(differential)gleichungen (engl. equations of motion). Am gebräuchlichsten und bekanntesten sind die Formulierungen nach Euler-Lagrange (EL) und Newton-Euler (NE).

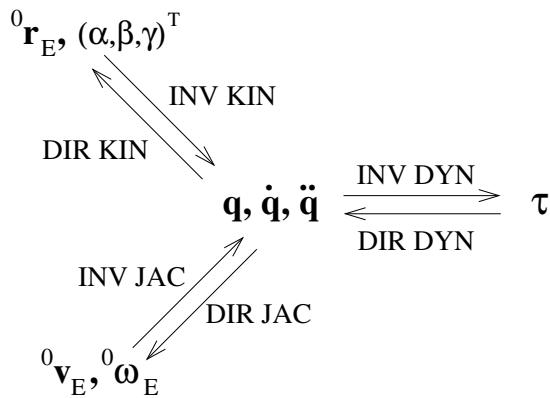


Figure 5.1: Vollständige kinematische und dynamische Modellierung von Manipulatoren:  $q \in \mathbb{R}^n$  Gelenkvariablen,  $\dot{q} \in \mathbb{R}^n$  Gelenkgeschwindigkeiten,  $\ddot{q} \in \mathbb{R}^n$  Gelenkbeschleunigungen; Position  ${}^0 r_E \in \mathbb{R}^3$  und Orientierung  $(\alpha, \beta, \gamma)^T$  des Endeffektors (bzw. Ursprungs von  $S_E$ ); lineare Geschwindigkeit  ${}^0 v_E \in \mathbb{R}^3$  und Drehwinkelgeschwindigkeit  ${}^0 \omega_E \in \mathbb{R}^3$  des Endeffektors (bzw. Ursprungs von  $S_E$ );  $\tau \in \mathbb{R}^n$  Kräfte bzw. Momente der Schub- bzw. Drehgelenksantriebe.

<sup>2</sup><https://github.com/SIM-TU-Darmstadt/mbslib>

## 5.2 Massenverteilung eines Starrkörpers

Gegenstände mit *gleichem* Gewicht aber *unterschiedlicher* Massenverteilung (z.B. eine Kugel mit Durchmesser 6 cm und ein 1.20 m langer, zylinderförmiger Stab mit Durchmesser 2 cm, die beide 2 kg wiegen) verhalten sich ganz unterschiedlich beim Schleudern aus der Hand, wobei die Beschleunigung aus den Drehgelenken des Arms, vor allem dem Schultergelenk, heraus erzeugt wird. Die Ursache für unterschiedliches, bewegungsdynamisches Verhalten der beiden gleichschweren Körper liegt in der unterschiedlichen Massenverteilung.

Die Massenverteilung eines Starrkörpers wird durch den Trägheitstensor  ${}^a I \in \mathbb{R}^{3 \times 3}$  ( $I$ : für „inertia tensor“, Trägheitstensor) bzgl. eines körperfesten Koordinatensystems  $S_a$  beschrieben als:

$${}^a I = \begin{pmatrix} {}^a I_{xx} & -{}^a I_{xy} & -{}^a I_{xz} \\ -{}^a I_{xy} & {}^a I_{yy} & -{}^a I_{yz} \\ -{}^a I_{xz} & -{}^a I_{yz} & {}^a I_{zz} \end{pmatrix}$$

mit

$V$ : dreidimensionale, geometrische Ausdehnung des Körpers („Volumen“)

$dv$ : differentielles Volumenelement (z.B.  $dv = dx dy dz$  bei kartesischen Koordinaten)

$\rho = \rho(x, y, z) \geq 0$ : Materialdichte,  $\rho : \mathbb{R}^3 \rightarrow \mathbb{R}_0^+$

gilt:

$$\underbrace{{}^a I_{xx} = \int \int \int_V (y^2 + z^2) \rho(x, y, z) dv}_{\text{Massenträgheitsmomente}}$$

$$\underbrace{{}^a I_{yy} = \int \int \int_V (x^2 + z^2) \rho(x, y, z) dv}_{\text{Massenträgheitsmomente}}$$

$$\underbrace{{}^a I_{zz} = \int \int \int_V (x^2 + y^2) \rho(x, y, z) dv}_{\text{Massenträgheitsmomente}}$$

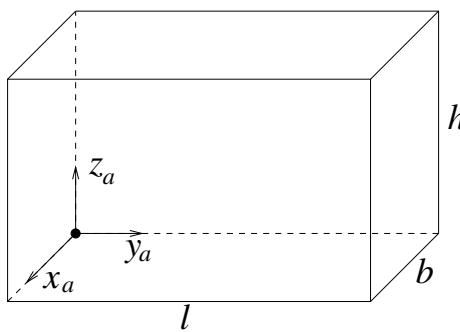
$$\underbrace{{}^a I_{xy} = \int \int \int_V xy \rho(x, y, z) dv = {}^a I_{yx}}_{\text{Massenträgheitsprodukte}}$$

$$\underbrace{{}^a I_{xz} = \int \int \int_V xz \rho(x, y, z) dv = {}^a I_{zx}}_{\text{Massenträgheitsprodukte}}$$

$$\underbrace{{}^a I_{yz} = \int \int \int_V yz \rho(x, y, z) dv = {}^a I_{zy}}_{\text{Massenträgheitsprodukte}}$$

### Einfaches Beispiel: Quader mit gleichförmiger homogener Massenverteilung

Bei gleichförmiger Massenverteilung gilt  $\rho(x, y, z) = \rho = \text{const.} > 0$  für  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \text{Quader}$ .



$$\Rightarrow \text{Quadervolumen} = l b h$$

$$\text{Quadermasse } m = l b h \rho = \text{Volumen} \cdot \rho$$

Anwendung der Formeln für die Massenträgheitsmomente und -produkte ergibt für das

erste Massenträgheitsmoment:

$$\begin{aligned}
 {}^a I_{xx} &= \int_V \int \int (y^2 + z^2) \underbrace{\rho(x, y, z)}_{\text{const.}} dv = \rho \int_0^h \int_0^l \int_0^b (y^2 + z^2) dx dy dz \\
 &= \rho \int_0^h \int_0^l [(y^2 + z^2)x]_{x=0}^{x=b} dy dz = \rho \int_0^h \int_0^l (y^2 + z^2)b dy dz \\
 &= \rho \int_0^h \left( \frac{l^3}{3} + z^2 l \right) b dz = \rho \left( \frac{l^3}{3} h + \frac{h^3}{3} l \right) b \\
 &= \frac{1}{3} l b h \rho (l^2 + h^2) = \frac{m}{3} (l^2 + h^2)
 \end{aligned}$$

Analog erhält man:

$$\begin{aligned}
 {}^a I_{yy} &= \frac{m}{3} (b^2 + h^2) \\
 {}^a I_{zz} &= \frac{m}{3} (l^2 + b^2) \\
 {}^a I_{xy} &= \int_0^h \int_0^l \int_0^b xy \rho dx dy dz = \dots = \frac{m}{4} b l \\
 {}^a I_{xz} &= \frac{m}{4} b h \\
 {}^a I_{yz} &= \frac{m}{4} l h
 \end{aligned}$$

Wird die Lage des körperfesten Bezugskoordinatensystems  $S_a$  verändert, so ändern sich im Allgemeinen auch die Elemente von  ${}^a I$ .

### Allgemeine Eigenschaften von Trägheitstensoren

- 1.) Im Allgemeinen sind die Eigenwerte von  ${}^a I$  die Hauptträgheitsmomente des Körpers und die zugehörigen Eigenvektoren sind die Hauptträgheitsachsen.
- 2.) Wenn zwei Achsen von  $S_a$  (z.B.  $x_a, y_a$ ) eine Symmetrieebene für die Massenverteilung des Körpers bilden, dann sind die Trägheitsprodukte mit der dritten Achse als Index gleich Null (z. B.  ${}^a I_{xz} = {}^a I_{yz} = 0$ ).
- 3.) Die sechs unabhängigen Größen  ${}^a I_{xx}, {}^a I_{yy}, {}^a I_{zz}, {}^a I_{xy}, {}^a I_{xz}, {}^a I_{yz}$  hängen von der Lage des körperfesten Koordinatensystems  $S_a$  ab. Falls durch geeignete Wahl der Lage von  $S_a$ <sup>3</sup> erreicht werden kann, dass

$${}^a I_{xy} = {}^a I_{xz} = {}^a I_{yz} = 0$$

gilt, heißen die Koordinatenachsen  $\{x_a, y_a, z_a\}$  die **Hauptträgheitsachsen** und  ${}^a I_{xx}, {}^a I_{yy}, {}^a I_{zz}$  die **Hauptträgheitsmomente** (engl. principal moments of inertia). Die typische Einheit der Massenträgheitsmomente ist **kg m<sup>2</sup>**.

---

<sup>3</sup>Dafür ist notwendig, dass der Ursprung von  $S_a$  im Körperschwerpunkt liegt, und zusätzlich müssen die Koordinatenachsen paarweise Symmetriebenen für die Massenverteilung bilden.

- 4.) Die Trägheitsmomente  ${}^aI_{xx}$ ,  ${}^aI_{yy}$ ,  ${}^aI_{zz}$  sind **immer positiv** (d. h.  $> 0$ ), egal wie  $S_a$  gewählt wird. Dies gilt **nicht** für die Trägheitsprodukte  ${}^aI_{xy}$ ,  ${}^aI_{xz}$ ,  ${}^aI_{yz}$ .
- 5.)  $\text{Spur}({}^aI) = {}^aI_{xx} + {}^aI_{yy} + {}^aI_{zz}$  ändert sich bei Rotationen des Bezugskoordinatensystems  $S_a$  **nicht**.

### Transformation von Trägheitstensoren

Der Trägheitstensor eines Starrkörpers ist abhängig vom Koordinatensystem bzgl. dessen der Trägheitstensor bestimmt wurde. Ist der Trägheitstensor bzgl. eines Koordinatensystems bekannt, kann er auf andere Systeme transformiert werden.

- Rotation des Bezugssystems: Ist der Trägheitstensor  ${}^aI$  bzgl. eines Koordinatensystems  $S_a$  gegeben, kann er in einem System  $S_b$ , dass dazu um  ${}^aR_b$  verdreht, aber nicht verschoben, ist durch

$${}^bI = {}^bR_a \cdot {}^aI \cdot {}^aR_b$$

dargestellt werden.

- Translation des Bezugssystems: Ist der Trägheitstensor  ${}^cI$  bzgl. eines Koordinatensystems  $S_c$  gegeben, dessen Ursprung im Schwerpunkt des Körpers liegt, kann er in einem System  $S_b$ , das dazu um  ${}^c r_b$  verschoben, aber nicht verdreht, ist durch

$${}^bI = {}^cI + m \cdot \left( {}^c r_b^T \cdot {}^c r_b \cdot E - {}^c r_b \cdot {}^c r_b^T \right)$$

dargestellt werden, wobei  $m$  die Masse des Körpers ist. Dieser Zusammenhang ist als „Satz von Steiner“ bekannt.

- Zusammenfügen von Körpern: Falls zwei Starrkörper mit den Trägheitstensoren  $I_1$  und  $I_2$  zusammengefügt werden, so ergibt sich der Trägheitstensor des entstehenden Körpers durch Addition der Trägheitstensoren, sofern diese im gleichen Koordinatensystem dargestellt werden.

### Bemerkungen:

- Bei rein linearen Bewegungen, bei denen die beschleunigende Kraft nur am Schwerpunkt eines Körpers angreift, spielt der Trägheitstensor keine Rolle. Dieser hat erst dann einen Einfluss auf die Bewegung, wenn entweder eine beschleunigende Kraft nicht am Schwerpunkt des Körpers oder ein Drehmoment am Körper angreift.
- Die Berechnung von  ${}^aI$  nach den Formeln  $\int \int \int \dots$  ist schwierig, wenn Geometrien und Zusammensetzungen von Robotergliedern komplex sind. Allerdings gibt es CAE-Systeme, die die Berechnungen aus einem CAD-Datensatz des Roboters heraus unterstützen (z.B. Autodesk Inventor<sup>4</sup> oder SolidWorks<sup>5</sup>). Dies ist in der Praxis die gebräuchlichste Methode zur Bestimmung der Trägheitstensoren und Schwerpunkte der Roboterglieder.
- Falls der Roboter zerlegt werden kann, so kann  ${}^aI$  für jedes Glied individuell experimentell vermessen werden („auspendeln“).

<sup>4</sup>[www.autodesk.de/inventor](http://www.autodesk.de/inventor)

<sup>5</sup>[www.solidworks.de](http://www.solidworks.de)

- In den letzten 15-20 Jahren wurden mehrere numerische Verfahren entwickelt, um bei geeigneten Voraussetzungen die Elemente der Trägheitstensoren  ${}^aI$  der Roboterglieder durch Vergleich von Simulationen für und Messungen von  $q$  und  $\dot{q}$  bei gegebenen Kräften bzw. Momenten der Gelenkmotoren  $\tau$  für geeignete Testbewegungen des Roboters zu berechnen ohne den Roboter physikalisch zerlegen und die Glieder einzeln ausmessen zu müssen.

## 5.3 Iterative Newton-Euler Formulierung der Manipulatordynamik

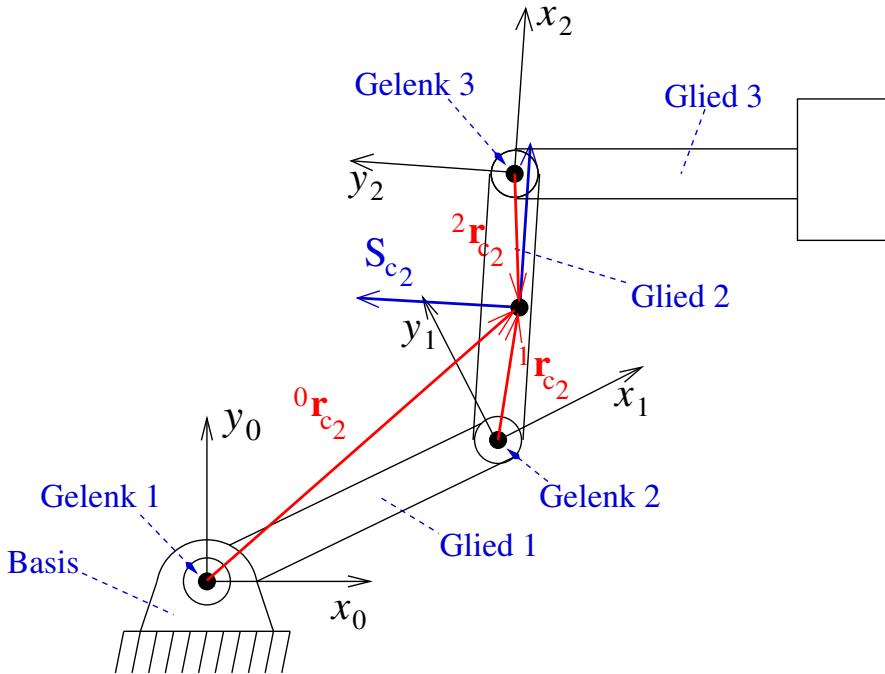
### 5.3.1 Einleitung, Impuls- und Drallsatz

Das  $i$ -te Roboterglied wird als Starrkörper betrachtet, für den folgende Daten gegeben sind

- der Koordinatenvektor des Schwerpunkts von Glied  $i$  bzgl.  $S_i$  als (konstanter Vektor)  ${}^i\mathbf{r}_{c_i}$  und bzgl.  $S_0$  bzw.  $S_{i-1}$  als (i. allg. nicht konstanter Vektor)  ${}^0\mathbf{r}_{c_i}$  bzw.  ${}^{i-1}\mathbf{r}_{c_i}$  sowie
- die Gesamtmasse  $m_i > 0$  von Glied  $i$  (einschließlich Gelenk  $i+1$ ) und
- der Trägheitstensor  ${}^c_i I_i \in \mathbb{R}^{3 \times 3}$  bzgl.  $S_{c_i}$ .

Dabei sei  $S_{c_i}$  das Koordinatensystem mit Ursprung im Schwerpunkt  ${}^i\mathbf{r}_{c_i}$  und derselben Orientierung wie  $S_i$ . Wie  $S_i$  ist damit folglich auch  $S_{c_i}$  fest mit Glied  $i$  verbunden. Da der Ursprung von  $S_{c_i}$  im Schwerpunkt von Glied  $i$  liegt, sind nach Kapitel 5.2 für den Trägheitstensor  ${}^c_i I_i$  die Massenträgheitsprodukte  ${}^c_i I_{i,xy} = {}^c_i I_{i,xz} = {}^c_i I_{i,yz} = 0$ , falls die Koordinatenachsen von  $S_{c_i}$  paarweise Symmetriebenen für die Massenverteilung bilden, und die Diagonalelemente  ${}^c_i I_{i,xx}$ ,  ${}^c_i I_{i,yy}$ ,  ${}^c_i I_{i,zz}$  die Hauptträgheitsmomente.

**Beispiel:**



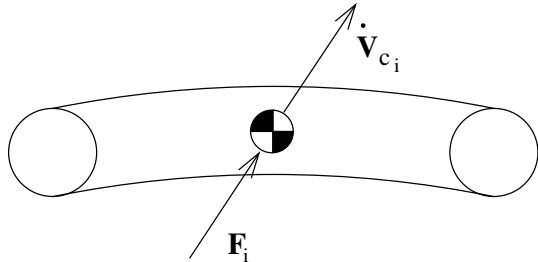
Um das  $i$ -te Glied zu bewegen, muss es beschleunigt bzw. abgebremst werden. Die dafür benötigten Kräfte hängen von der gewünschten Beschleunigung  $\ddot{q}_i$  sowie der Massenverteilung aller Roboterglieder und den über die Gelenkkopplungen wirkenden Wechselwirkungen ab.

Wir nehmen nun vorübergehend an, dass das  $i$ -te Roboterglied (z.B. das im Bild skizzierte, zweite Glied) an den Gelenken  $i$  und  $i+1$  „freigeschnitten“ wird. Die Beziehungen zwischen den Kräften (forces) bzw. den Drehmomenten (torques), den Massen, den Trägheiten

(inertias) und den Beschleunigungen (accelerations), die an einem Starrkörper wirken, werden durch zwei grundlegende Gleichungen beschrieben: den Impulssatz und den Drallsatz.

### (1) Newtonsche Gleichung (Impulssatz)

$$\mathbf{F}_i = m_i \cdot \dot{\mathbf{v}}_{c_i}$$



Dabei bezeichnen

$\bullet$  : die Lage des Schwerpunktes des  $i$ -ten Roboterglieds

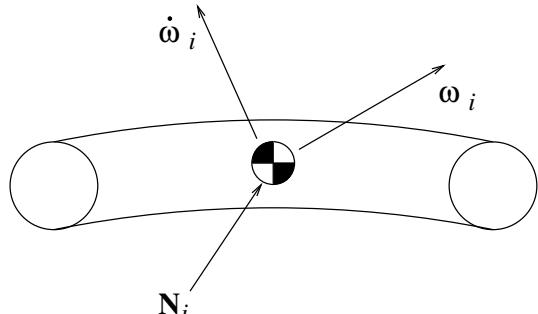
$\dot{\mathbf{v}}_{c_i} \in \mathbb{R}^3$  : die Beschleunigung des Schwerpunktes des  $i$ -ten Roboterglieds, gemessen bezüglich  $S_0$  und dargestellt in  $S_i$ , d.h.  $\dot{\mathbf{v}}_{c_i} = {}^i R_0 \cdot {}^0 \dot{\mathbf{v}}_{c_i}$

$\mathbf{F}_i \in \mathbb{R}^3$  : die am Schwerpunkt des  $i$ -ten Roboterglieds angreifende Kraft  ${}^0 \mathbf{F}_i$ , die die Beschleunigung hervorruft, gemessen bezüglich  $S_0$  und dargestellt in  $S_i$ , d.h.  $\mathbf{F}_i = {}^i R_0 \cdot {}^0 \mathbf{F}_i$

$m_i \in \mathbb{R}^+$  : die Gesamtmasse des  $i$ -ten Körpers (d.h. Glied  $i$  mit zugehörigem Gelenk).

### (2) Eulersche Gleichung (Drallsatz)

$$\mathbf{N}_i = {}^{c_i} I_i \cdot \dot{\omega}_i + \omega_i \times ({}^{c_i} I_i \cdot \omega_i)$$



Dabei bezeichnen

$\omega_i \in \mathbb{R}^3$  : die Winkelgeschwindigkeit des  $i$ -ten Roboterglieds,

$\dot{\omega}_i \in \mathbb{R}^3$  : die Winkelbeschleunigung des  $i$ -ten Roboterglieds,

$\mathbf{N}_i \in \mathbb{R}^3$  : das im Schwerpunkt angreifende Drehmoment, welches die Drehgeschwindigkeit und -beschleunigung verursacht,

${}^{c_i} I_i \in \mathbb{R}^{3 \times 3}$  : den Trägheitstensor des  $i$ -ten Roboterglieds bzgl. des Koordinatensystems  $S_{c_i}$  (mit gleicher Orientierung wie  $S_i$ ).

**Bemerkung:** Die Zahlenwerte der jeweiligen Größen sind meist abhängig vom jeweiligen Bezugskoordinatensystem. Die (relativen) Positions-, Geschwindigkeits- und Beschleunigungsvektoren von Glied  $i$  können beispielsweise bezüglich der Koordinatensysteme  $S_{i-1}$  oder  $S_i$  dargestellt werden. Je nachdem können sich die nachfolgenden Formeln ändern.

Im folgenden Abschnitt verkoppeln wir das freigeschnittene  $i$ -te Roboterglied wieder mit den anderen Gliedern und Gelenken der kinematischen Kette. Dann hängen die linearen Beschleunigungen  $\dot{\mathbf{v}}_{c_i}$ , die Winkelbeschleunigungen  $\dot{\omega}_i$ , die Kräfte  $\mathbf{F}_i$  und die Momente

$\mathbf{N}_i$  an den Gliederschwerpunkten aber auch an den Gelenken (bzw. den Ursprüngen der lokalen Koordinatensysteme  $S_k$ ) von den entlang der kinematischen Kette wirkenden Wechselwirkungen mit den entsprechenden Werten der benachbarten Glieder rekursiv ab.

### 5.3.2 Iterative Berechnung von INV DYN

In diesem Abschnitt wird die iterative Berechnung des inversen Dynamikmodells INV DYN nach der Methode von Newton-Euler beschrieben.

**Gegeben** ist eine Trajektorie in Gelenkvariablen:  $\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t) \in \mathbb{R}^n, 0 \leq t \leq t_f$ .

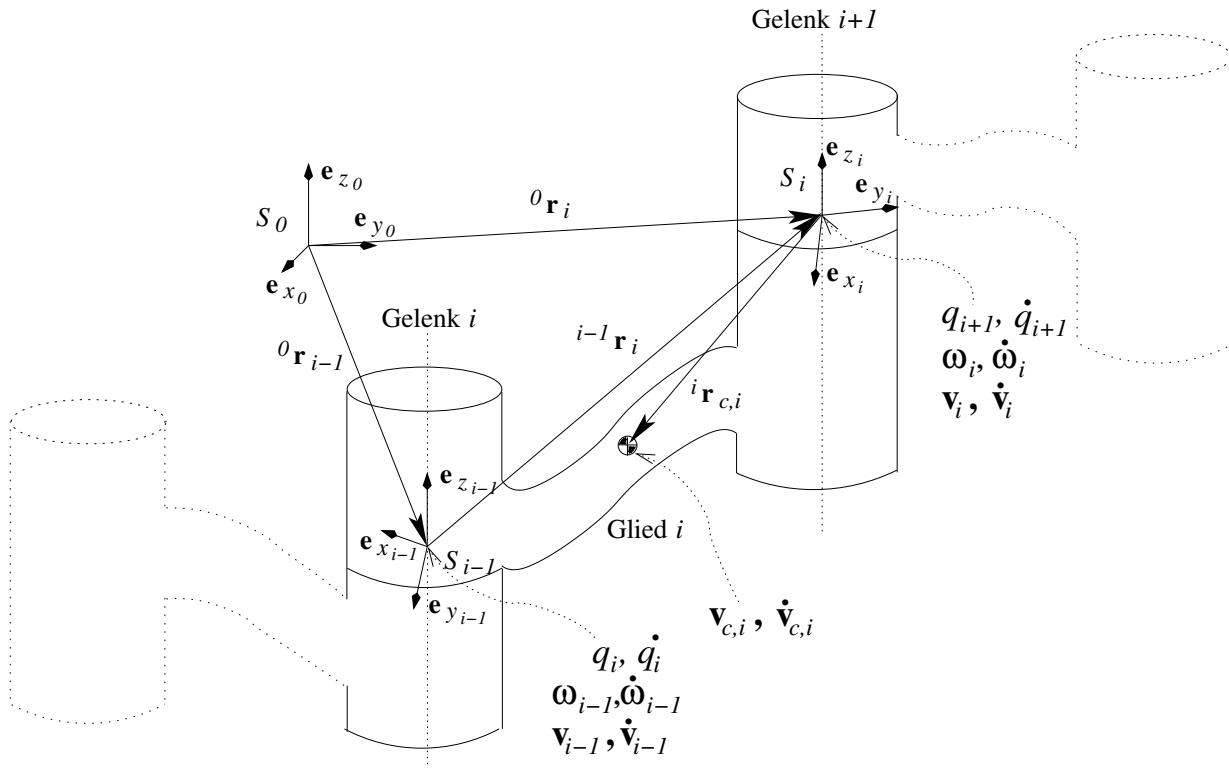
**Gesucht** sind die für diese Trajektorie benötigten Gelenkdrehmomente bzw. -kräfte  $\tau(t) \in \mathbb{R}^n, 0 \leq t \leq t_f$ .

Diese werden unter Verwendung der Roboterkinematik, der Schwerpunktsslagen  ${}^i\mathbf{r}_{c,i}$ , der Trägheitstensoren  ${}^c_i I_i$  und Massen  $m_i$  der Roboterglieder in drei Schritten (NE.1), (NE.2) und (NE.3) berechnet:

$$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n \longrightarrow \boxed{\text{INV DYN}} \longrightarrow \tau \in \mathbb{R}^n.$$

**(NE.1) Iteration „von innen nach außen“, d.h. von der Basis zum Endeffektor, zur Berechnung der linearen und Winkel-Beschleunigungen der Schwerpunkte jedes Roboterglieds  $\dot{\mathbf{v}}_{c,i}$  und  $\ddot{\mathbf{v}}_{c,i}$**

**Prinzip:** Man startet die Berechnungen beim ersten Roboterglied (die Roboterbasis ist das 0-te Glied) und führt die Berechnung sukzessive „nach außen“ zum  $n$ -ten Glied fort.



**Bezeichnungen:**

- $\omega_i := {}^i R_0 \cdot {}^0 \omega_i \in \mathbb{R}^3$ : **Winkelgeschwindigkeitsvektor** des  $i$ -ten Glieds bzgl. Basiskoordinatensystem  $S_0$  und dargestellt im Koordinatensystem  $S_i$ , das mit Glied  $i$  im Gelenk  $i+1$  fest verbunden ist,
- $\dot{\omega}_i := {}^i R_0 \cdot {}^0 \dot{\omega}_i \in \mathbb{R}^3$ : **Winkelbeschleunigungsvektor** des  $i$ -ten Glieds bzgl. Basiskoordinatensystem  $S_0$  und dargestellt im Koordinatensystem  $S_i$ ,
- $\mathbf{v}_i := {}^i R_0 \cdot {}^0 \mathbf{v}_i \in \mathbb{R}^3$ : **(linearer/Translations-)Geschwindigkeitsvektor** des  $i$ -ten Glieds am Gelenk  $i+1$  bzgl. Basiskoordinatensystem  $S_0$ , dargestellt im Koordinatensystem  $S_i$ ,
- $\dot{\mathbf{v}}_i := {}^i R_0 \cdot {}^0 \dot{\mathbf{v}}_i \in \mathbb{R}^3$ : **(linearer/Translations-)Beschleunigungsvektor** des  $i$ -ten Glieds am Gelenk  $i+1$  bzgl. Basiskoordinatensystem  $S_0$ , dargestellt im Koordinatensystem  $S_i$ ,

Die linearen Geschwindigkeiten und Beschleunigungen  $\mathbf{v}_i, \dot{\mathbf{v}}_i$  beziehen sich auf den Ursprung von  $S_i$ . Die Winkelgeschwindigkeiten und -beschleunigungen  $\omega_i, \dot{\omega}_i$  sind für alle Punkte des Glieds  $i$  identisch.

$\dot{\mathbf{v}}_{c_i} := {}^i R_0 \cdot {}^0 \dot{\mathbf{v}}_{c_i} \in \mathbb{R}^3$ : **(linearer/Translations-)Beschleunigungsvektor** des Schwerpunkts des  $i$ -ten Glieds bzgl.  $S_0$ , dargestellt in  $S_i$ ,

${}^i \mathbf{e}_{z_{i-1}} = ({}^{i-1} R_i)^T \cdot {}^{i-1} \mathbf{e}_{z_{i-1}} \in \mathbb{R}^3$ : Einheitsvektor der Achse, um die  $q_i$  bewegt wird (d.h. die Rotations- oder Translationsachse), dargestellt im Koordinatensystem  $S_i$ . Der Vektor  ${}^i \mathbf{e}_{z_{i-1}}$  entspricht der dritten Zeile von  ${}^{i-1} R_i$  bzw. der dritten Spalte von  ${}^i R_{i-1}$ , da  ${}^{i-1} \mathbf{e}_{z_{i-1}} = (0, 0, 1)^T$ ,  
[Zur Erinnerung: Es gilt  $({}^{i-1} R_i)^T = ({}^{i-1} R_i)^{-1} = {}^i R_{i-1}$ .]

$\rho_i \in \{0, 1\}$ : binäre Variable zur Unterscheidung des Gelenktyps:  
 $\rho_i = 1$  für ein Drehgelenk,  $\rho_i = 0$  für ein Schubgelenk.

**Vorgehensweise:** Ausgehend von der Basis ( $i = 0$ ) sollen zunächst sukzessiv, d.h. für  $i = 1, 2, \dots, n$ , die Größen  $\omega_i, \dot{\omega}_i, \mathbf{v}_i, \dot{\mathbf{v}}_i$  und  $\dot{\mathbf{v}}_{c_i}$  rekursiv mit Hilfe der für  $i-1$  berechneten Größen bestimmt werden.

Die Winkelgeschwindigkeit  $\omega_i$  von Glied  $i$  setzt sich zusammen aus der Winkelgeschwindigkeit von Glied  $i-1$  und der durch das Gelenk  $i$  zusätzlich hervorgerufenen Winkelgeschwindigkeit, falls Gelenk  $i$  ein Drehgelenk ist, wobei das Ergebnis bezüglich Koordinatensystem  $S_i$  darzustellen ist:

$$\omega_i = {}^i R_{i-1} \left( \omega_{i-1} + \rho_i {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i \right). \quad (5.1)$$

Zur Berechnung von  $\dot{\omega}_i$  kann Gleichung (5.1) unter Berücksichtigung der Produktregel für

$$(\omega_i)' = ({}^i R_0 \cdot {}^0 \omega_i)' = ({}^i R_0)' \cdot {}^0 \omega_i + {}^i R_0 \cdot ({}^0 \omega_i)' = ({}^i R_0)' \cdot {}^0 \omega_i + \dot{\omega}_i$$

und von

$$\begin{aligned} \dot{\omega}_i &= {}^i R_0 {}^0 \dot{\omega}_i \\ &= (\omega_i)' - ({}^i R_0)' \cdot {}^0 \omega_i = (\omega_i)' - ({}^0 \dot{R}_i)^T \cdot {}^0 \omega_i = (\omega_i)' - \left( B({}^0 \omega_i)^T R_i \right)^T \cdot {}^0 \omega_i \\ &= (\omega_i)' - ({}^0 R_i)^T (B({}^0 \omega_i))^T \cdot {}^0 \omega_i = (\omega_i)' - {}^i R_0 (-B({}^0 \omega_i)) \cdot {}^0 \omega_i = (\omega_i)' + {}^i R_0 ({}^0 \omega_i \times {}^0 \omega_i) \\ &= (\omega_i)' \end{aligned}$$

total nach der Zeit  $t$  differenziert werden, wobei auf die rechte Seite der Gleichung die Produktregel anzuwenden ist:

$$\dot{\omega}_i = \underbrace{\left( {}^i R_{i-1} \right) \cdot \left( \underbrace{\omega_{i-1} + \rho_i {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i}_{:=\mathbf{b}} \right)}_{:=\mathbf{a}} + {}^i R_{i-1} \left( \omega_{i-1} + \rho_i {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i \right).$$

Falls  $q_i$  Schubgelenk ist, gilt  $({}^i R_{i-1}) \cdot = 0$  und damit  $\mathbf{a} = \mathbf{0}$ .

Falls  $q_i$  Drehgelenk ist, gilt  $\rho_i = 1$  und nach Kapitel 4.2, Seite 62, gilt für die Ableitung einer Rotationsmatrix

$$({}^i R_{i-1}) \cdot = B(\omega) {}^i R_{i-1} \quad \text{wobei hier } \omega = {}^i R_{i-1} {}^{i-1} \mathbf{e}_{z_{i-1}} (-\dot{q}_i).$$

Nach Kapitel 4.2 gilt außerdem  $\dot{R} \cdot \mathbf{b} = B(\omega) \cdot R \cdot \mathbf{b} = \omega \times (R \cdot \mathbf{b})$ , damit folgt für  $\mathbf{a}$  falls  $q_i$  Drehgelenk:

$$\begin{aligned} \mathbf{a} &= ({}^i R_{i-1}) \cdot \mathbf{b} = \omega \times ({}^i R_{i-1} \cdot \mathbf{b}) \\ &= ({}^i R_{i-1} {}^{i-1} \mathbf{e}_{z_{i-1}} (-\dot{q}_i)) \times \left( {}^i R_{i-1} \cdot \left( \omega_{i-1} + \underbrace{\rho_i {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i}_{=1} \right) \right) \\ &= {}^i R_{i-1} \cdot \left[ ({}^{i-1} \mathbf{e}_{z_{i-1}} (-\dot{q}_i)) \times \left( \omega_{i-1} + {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i \right) \right] \\ &= {}^i R_{i-1} \cdot \left[ ({}^{i-1} \mathbf{e}_{z_{i-1}} (-\dot{q}_i)) \times \omega_{i-1} + \underbrace{({}^{i-1} \mathbf{e}_{z_{i-1}} (-\dot{q}_i)) \times {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i}_{=0} \right] \\ &= {}^i R_{i-1} \cdot \left[ \omega_{i-1} \times ({}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i) \right] \end{aligned}$$

Insgesamt folgt damit für den Winkelbeschleunigungsvektor  $\dot{\omega}_i \in \mathbb{R}^3$  des  $i$ -ten Glieds:

$$\begin{aligned} \dot{\omega}_i &= \rho_i {}^i R_{i-1} \cdot \left[ \omega_{i-1} \times ({}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i) \right] + {}^i R_{i-1} \left( \dot{\omega}_{i-1} + \rho_i {}^{i-1} \mathbf{e}_{z_{i-1}} \ddot{q}_i \right) \\ \text{bzw. } \dot{\omega}_i &= {}^i R_{i-1} \left( \dot{\omega}_{i-1} + \rho_i \left[ {}^{i-1} \mathbf{e}_{z_{i-1}} \ddot{q}_i + \omega_{i-1} \times ({}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i) \right] \right) \end{aligned} \quad (5.2)$$

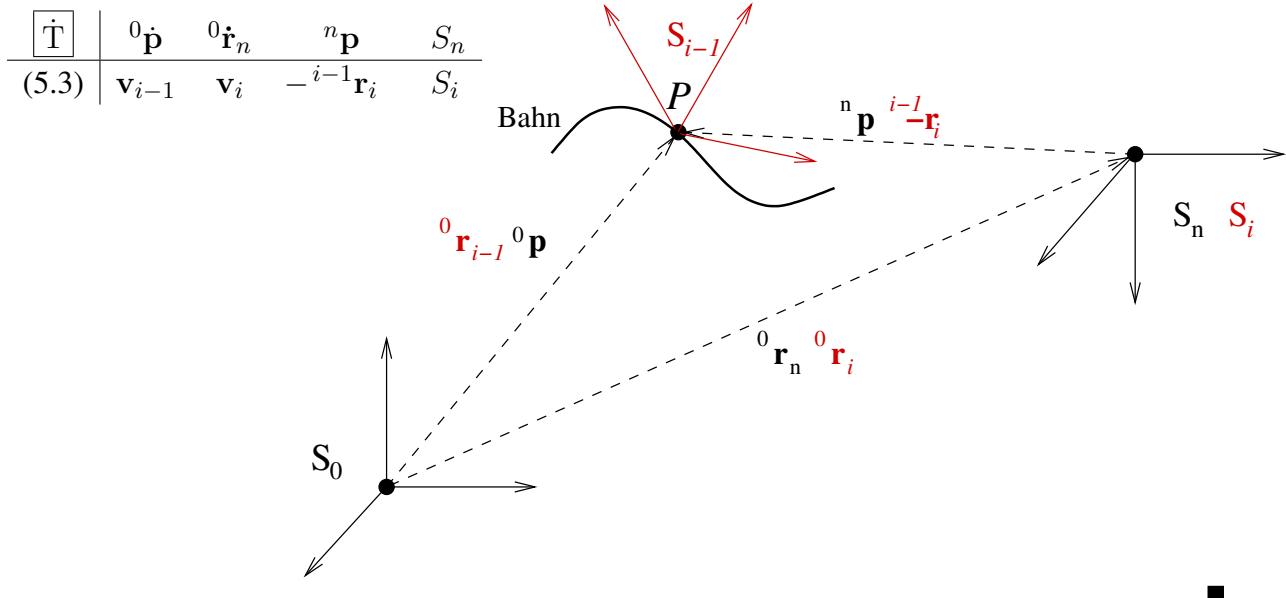
Der lineare Geschwindigkeitsvektor  $\mathbf{v}_i$  des  $i$ -ten Glieds ergibt sich durch Addition

- aus dem linearen Geschwindigkeitsvektor  $\mathbf{v}_{i-1}$  des  $(i-1)$ -ten Glieds, der sich auf den Ursprung von  $S_{i-1}$  bezieht,
- dem von der Winkelgeschwindigkeit  $\omega_i$  des  $i$ -ten Glieds im Abstand  ${}^{i-1} \mathbf{r}_i$  vom Ursprung von  $S_{i-1}$  zu  $S_i$  hervorgerufenen linearen Geschwindigkeitsvektors  $\omega_i \times {}^{i-1} \mathbf{r}_i$  und
- der durch das  $i$ -te Gelenk hervorgerufenen linearen Geschwindigkeit, falls  $q_i$  ein Schubgelenk ist.

Der resultierende lineare Geschwindigkeitsvektor und die auftretenden Größen sind bezüglich  $S_i$  darzustellen:

$$\begin{aligned} \mathbf{v}_i &= {}^i R_{i-1} \left( \mathbf{v}_{i-1} + \left( {}^{i-1} R_i \omega_i \right) \times {}^{i-1} \mathbf{r}_i + (1 - \rho_i) {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i \right) \\ &= {}^i R_{i-1} \mathbf{v}_{i-1} + \omega_i \times ({}^i R_{i-1} {}^{i-1} \mathbf{r}_i) + (1 - \rho_i) {}^i \mathbf{e}_{z_{i-1}} \dot{q}_i. \end{aligned} \quad (5.3)$$

**Bemerkung:** Gleichung (5.3) lässt sich direkt aus Gleichung  $\dot{\mathbf{T}}$  (Seite 62) ableiten, wenn folgende Entsprechungen — jeweils bis auf Multiplikation mit entsprechenden R-Matrizen — berücksichtigt werden. Die abgebildete Skizze zeigt die Notationen aus Kapitel 4.2 sowie zum Vergleich die entsprechende Notation in Gleichung (5.3) in rot bzw. grau:



Zur Berechnung von  $\dot{\mathbf{v}}_i$  könnte Gleichung (5.3) unter Berücksichtigung von  $\dot{\mathbf{v}}_i = {}^iR_0 \cdot {}^0\dot{\mathbf{v}}_i = (\mathbf{v}_i) \cdot ({}^iR_0) \cdot {}^0\mathbf{v}_i$  total nach der Zeit  $t$  abgeleitet werden. Andererseits können die Überlegungen und Ergebnisse zur relativen Beschleunigung zwischen zwei Koordinatensystemen in der kinematischen Kette aus Kapitel 4.2 (Seite 63) zur Berechnung von  $\dot{\mathbf{v}}_i$  angewendet werden:

- Falls das Gelenk  $i$  ein Drehgelenk ist, so ist der Vektor  ${}^iR_{i-1} {}^{i-1}\mathbf{r}_i(t) = \text{const.}$  und  $({}^iR_{i-1} {}^{i-1}\mathbf{r}_i(t)) \cdot = \mathbf{0}$ . Dies entspricht Gleichung  $\ddot{\mathbf{T}}_0$  (Seite 63) mit  ${}^n\mathbf{p}(t) = \text{const.}$  und  ${}^n\dot{\mathbf{p}}(t) = \mathbf{0}$ :

$$\ddot{\mathbf{T}}_0 \quad \underbrace{{}^0\ddot{\mathbf{p}}}_{{}^iR_{i-1}\dot{\mathbf{v}}_{i-1}} = \underbrace{({}^0\mathbf{r}_n) \cdot}_{\dot{\mathbf{v}}_i} + \underbrace{\dot{\omega}_i \times \underbrace{({}^0R_n {}^n\mathbf{p})}_{-{}^iR_{i-1} {}^{i-1}\mathbf{r}_i}}_{\omega_i} + \underbrace{\omega_i \times \left( \underbrace{\dot{\omega}_i \times \underbrace{({}^0R_n {}^n\mathbf{p})}_{-{}^iR_{i-1} {}^{i-1}\mathbf{r}_i} \right)}_{\omega_i} \right)$$

Durch Umsortierung und Anwendung der Rechenregeln für das Vektorprodukt erhält man für  $\dot{\mathbf{v}}_i$ :

$$\begin{aligned} \dot{\mathbf{v}}_i &= {}^iR_{i-1} \left( \dot{\mathbf{v}}_{i-1} + ({}^{i-1}R_i \dot{\omega}_i) \times {}^{i-1}\mathbf{r}_i + ({}^{i-1}R_i \omega_i) \times (({}^{i-1}R_i \omega_i) \times {}^{i-1}\mathbf{r}_i) \right) \\ &= {}^iR_{i-1} \dot{\mathbf{v}}_{i-1} + \dot{\omega}_i \times ({}^iR_{i-1} {}^{i-1}\mathbf{r}_i) + \omega_i \times ({}^iR_{i-1} {}^{i-1}\mathbf{r}_i) . \end{aligned}$$

- Falls das Gelenk  $i$  ein Schubgelenk ist, so gilt  ${}^{i-1}\dot{\mathbf{r}}_i(t) \neq \mathbf{0}$ . Dies entspricht dem allgemeinen Fall von Gleichung  $\ddot{\mathbf{T}}$  (Seite 63). Daraus folgt für  $\dot{\mathbf{v}}_i$ :

$$\begin{aligned} \dot{\mathbf{v}}_i &= {}^iR_{i-1} \dot{\mathbf{v}}_{i-1} + \dot{\omega}_i \times ({}^iR_{i-1} {}^{i-1}\mathbf{r}_i) + \omega_i \times ({}^iR_{i-1} {}^{i-1}\mathbf{r}_i) \\ &\quad + 2\omega_i \times {}^i\mathbf{e}_{z_{i-1}} \dot{q}_i + {}^i\mathbf{e}_{z_{i-1}} \ddot{q}_i \end{aligned}$$

Insgesamt erhält man also

$$\begin{aligned}\dot{\mathbf{v}}_i &= {}^i R_{i-1} \dot{\mathbf{v}}_{i-1} + \dot{\omega}_i \times ({}^i R_{i-1} {}^{i-1} \mathbf{r}_i) + \omega_i \times (\omega_i \times ({}^i R_{i-1} {}^{i-1} \mathbf{r}_i)) \\ &\quad + (1 - \rho_i) \cdot [2\omega_i \times {}^i \mathbf{e}_{z_{i-1}} \dot{q}_i + {}^i \mathbf{e}_{z_{i-1}} \ddot{q}_i]\end{aligned}\quad (5.4)$$

Durch entsprechendes Vorgehen kann man die lineare Beschleunigung am Schwerpunkt von Glied  $i$  bestimmen zu

$$\dot{\mathbf{v}}_{c_i} = \dot{\mathbf{v}}_i + \dot{\omega}_i \times {}^i \mathbf{r}_{c_i} + \omega_i \times (\omega_i \times {}^i \mathbf{r}_{c_i}) \quad (5.5)$$

**Zwischenfazit:** Die Gleichungen (5.1) bis (5.5) stellen für  $i = 1, \dots, n$  ein iteratives Berechnungsschema für die Größen  $\omega_i$ ,  $\dot{\omega}_i$ ,  $\mathbf{v}_i$ ,  $\dot{\mathbf{v}}_i$  und  $\dot{\mathbf{v}}_{c_i}$  dar.

Als Startwerte für diese Iteration werden

$$\begin{aligned}\omega_0 &= \mathbf{0}, \quad \dot{\omega}_0 = \mathbf{0} \quad \left[ = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^3 \right], \\ \mathbf{v}_0 &= \mathbf{0}, \quad \dot{\mathbf{v}}_0 = \mathbf{0}\end{aligned}$$

verwendet.

**Bemerkung:** Der Einfluss des Gravitationsvektors kann statt im Schritt (NE.2) auch im Anfangswert

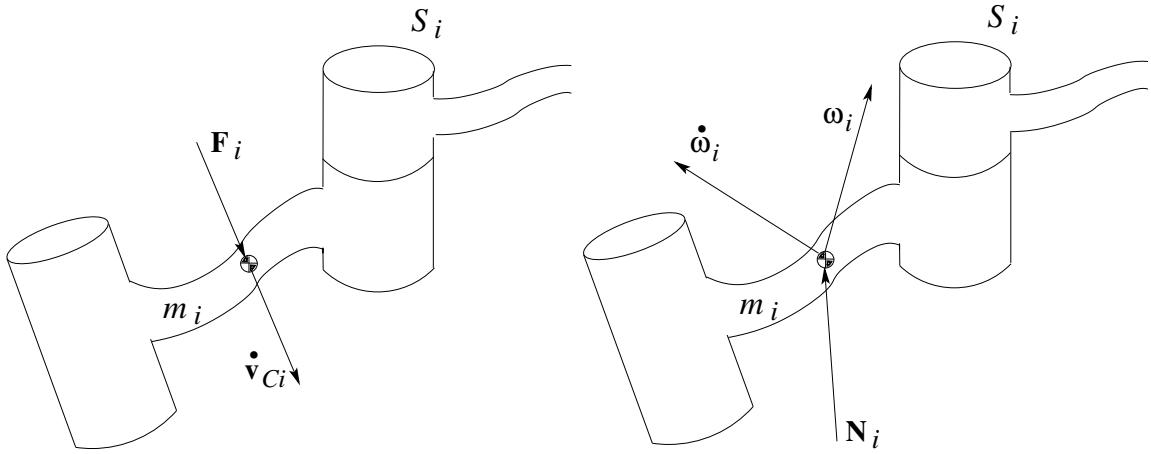
$$\dot{\mathbf{v}}_0 = -\mathbf{g} = -\begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix}$$

berücksichtigt werden. Dabei bezeichnet  $\mathbf{g} \in \mathbb{R}^3$  den Gravitationsvektor mit  $\|\mathbf{g}\| = 9.8062 \frac{\text{m}}{\text{s}^2}$  (auf Meereshöhe). Dies hat den Vorteil einer insgesamt geringeren Anzahl benötigter arithmetischer Operationen.

**Bemerkung:** Die Startwerte  $\omega_0 = \mathbf{0} = \dot{\omega}_0$  und  $\mathbf{v}_0 = \mathbf{0} = \dot{\mathbf{v}}_0$  gelten nur bei fester Roboterbasis. Bei bewegter Basis treten hier entsprechende, von Null verschiedene Winkelgeschwindigkeits- und Winkelbeschleunigungs-Vektoren auf.

**(NE.2) Berechnung der am Schwerpunkt  ${}^i r_{c_i}$  jedes Roboterglieds wirkenden Kräfte  $\mathbf{F}_i \in \mathbb{R}^3$  und Drehmomente  $\mathbf{N}_i \in \mathbb{R}^3$**

Die Berechnung erfolgt mit Impuls- und Drallsatz (Seite 104) und mit den im Schritt (NE.1) berechneten Werten:



$$\mathbf{F}_i = m_i \cdot \dot{\mathbf{v}}_{c_i} - m_i {}^i \mathbf{g} \quad \in \mathbb{R}^3 \quad (5.6)$$

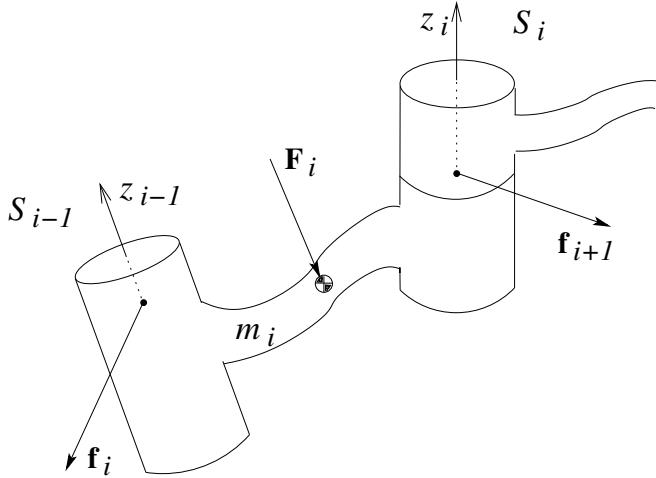
$$\mathbf{N}_i = {}^{c_i} I_i \cdot \dot{\omega}_i + \omega_i \times ({}^{c_i} I_i \cdot \omega_i) \quad \in \mathbb{R}^3 \quad (5.7)$$

Dabei bezeichnet  ${}^i \mathbf{g} := {}^i R_0 \mathbf{g}$  den in  $S_i$  dargestellten Gravitationsvektor (siehe z.B. [Spong/Vidyasagar, 1989], Section 6.5, Seite 157).

**Bemerkung:** Die Drehwinkelbeschleunigung  $\dot{\omega}$  ist an jedem Punkt eines starren Körpers gleich im Unterschied zur linearen Beschleunigung  $\dot{\mathbf{v}}$ .

**(NE.3) Iteration „von außen nach innen“ zur Berechnung der Kräfte und Drehmomente  $\tau_i$  am Gelenk jedes Roboterglieds**

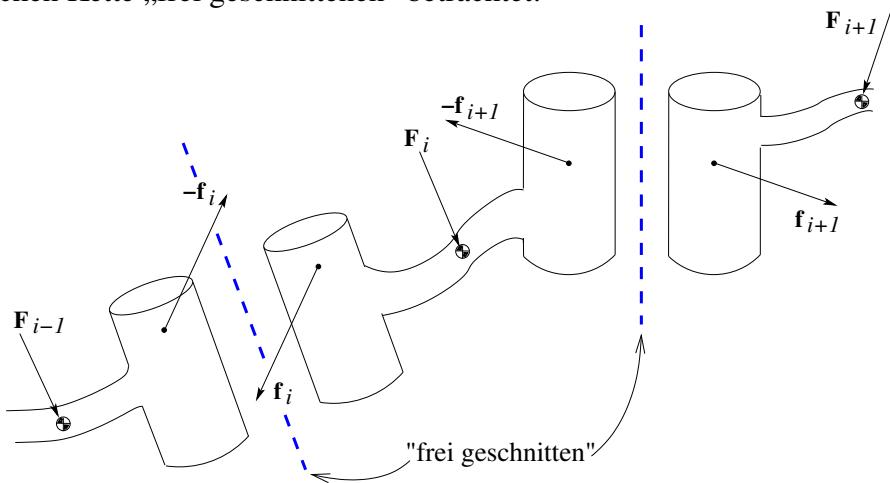
**Prinzip:** Die Berechnungen werden nun sukzessive vom „Äußersten“, d.h. vom  $n$ -ten (Endeffektor), bis zum ersten Glied „nach innen“ unter Berücksichtigung des Gleichgewichts der an jedem Glied wirkenden Kräfte und Momente abgeschlossen.

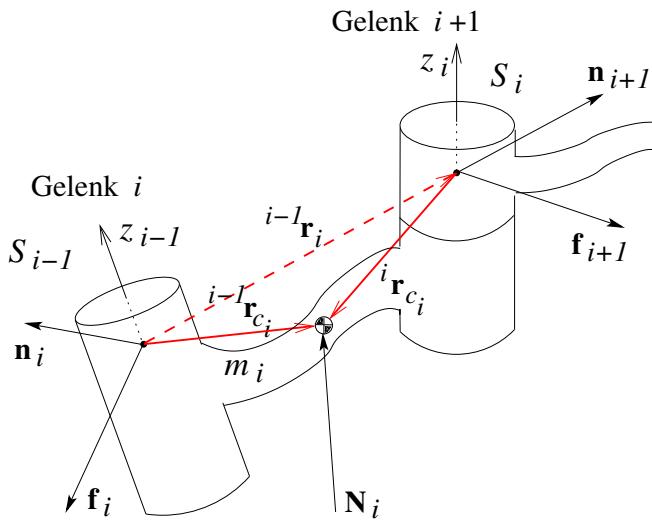


Aus dem Gleichgewicht der am  $i$ -ten Roboterglied wirkenden Kräfte folgt, dass die am Schwerpunkt wirkende Kraft  $F_i \in \mathbb{R}^3$  aus Addition der am Gelenk  $i + 1$  (d.h. am Ursprung von  $S_i$ ) und am Gelenk  $i$  (d.h. am Ursprung von  $S_{i-1}$ ) wirkenden Kräfte hervorgeht:

$$F_i = f_i + {}^i R_{i+1} \cdot (-f_{i+1}).$$

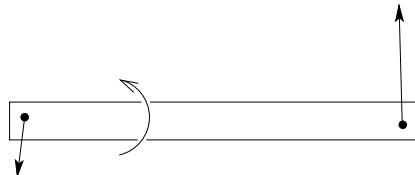
Dabei ist zu beachten, dass zur Durchführung der Vektoraddition die in  $S_{i+1}$  dargestellte Kraft  $f_{i+1}$  im selben Koordinatensystem  $S_i$  wie  $F_i$  und  $f_i$  darzustellen ist. Die Vorzeichen der zu addierenden Kräfte sind offensichtlich, wenn man das  $i$ -te Roboterglied aus der kinematischen Kette „frei geschnittenen“ betrachtet:





Aus dem Gleichgewicht der am  $i$ -ten Roboterglied wirkenden Drehmomente folgt, dass das am Schwerpunkt wirkende Drehmoment  $\mathbf{N}_i \in \mathbb{R}^3$  sich (analog zum eben betrachteten Kräftegleichgewicht) ergibt

- aus den bei den Gelenken  $i$  und  $i+1$  jeweils angreifenden Drehmomenten  $\mathbf{n}_i$  und  $\mathbf{n}_{i+1}$
- sowie aus Drehmomenten, die durch die an den Gelenken  $i$  bzw.  $i+1$  wirkenden Kräfte  $\mathbf{f}_i$  bzw.  $\mathbf{f}_{i+1}$  im Abstand  $i^{-1}\mathbf{r}_{c_i}$  bzw.  $i\mathbf{r}_{c_i}$  erzeugt werden. Denn zwei an unterschiedlichen Punkten eines Starrkörpers angreifende Kräfte erzeugen ein Drehmoment:



Die solchermaßen erzeugten Drehmomente zeigen in unterschiedliche Richtungen als das jeweilige  $\mathbf{n}_i$  bzw.  $\mathbf{n}_{i+1}$ .

Damit folgt

$$\mathbf{N}_i = \underbrace{\mathbf{n}_i + (-1) \underbrace{\left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i + {}^i \mathbf{r}_{c_i} \right)}_{= {}^i R_{i-1} {}^{i-1} \mathbf{r}_{c_i}} \times \mathbf{f}_i - \underbrace{\left( {}^i R_{i+1} \cdot \mathbf{n}_{i+1} + (-1) \left( {}^i \mathbf{r}_{c_i} \right) \times \left( {}^i R_{i+1} \mathbf{f}_{i+1} \right) \right)}_{\text{Drehmomente am Gelenk } i+1}}$$

Verwendet man die zuvor ermittelte Beziehung für die Kräfte

$$\mathbf{f}_i = {}^i R_{i+1} \mathbf{f}_{i+1} + \mathbf{F}_i$$

so erhält man

$$\begin{aligned} \mathbf{N}_i &= \mathbf{n}_i + \left( -{}^i R_{i-1} {}^{i-1} \mathbf{r}_i - {}^i \mathbf{r}_{c_i} \right) \times \left( {}^i R_{i+1} \mathbf{f}_{i+1} + \mathbf{F}_i \right) - {}^i R_{i+1} \cdot \mathbf{n}_{i+1} + {}^i \mathbf{r}_{c_i} \times \left( {}^i R_{i+1} \mathbf{f}_{i+1} \right) \\ &= \mathbf{n}_i - {}^i R_{i+1} \cdot \mathbf{n}_{i+1} - \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i \right) \times \left( {}^i R_{i+1} \mathbf{f}_{i+1} \right) - \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i + {}^i \mathbf{r}_{c_i} \right) \times \mathbf{F}_i. \end{aligned}$$

Löst man diese Beziehung nach  $\mathbf{n}_i$  auf, so kann eine Iteration „von außen nach innen“ zur Berechnung der Kräfte  $\mathbf{f}_i \in \mathbb{R}^3$  und Drehmomente  $\mathbf{n}_i \in \mathbb{R}^3$ , die in den einzelnen Gelenken wirken müssen, hergeleitet werden:

Für  $i = n, \dots, 1$ :

$$\begin{aligned}\mathbf{f}_i &= {}^i R_{i+1} \mathbf{f}_{i+1} + \mathbf{F}_i \\ \mathbf{n}_i &= {}^i R_{i+1} \mathbf{n}_{i+1} + \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i + {}^i \mathbf{r}_{c_i} \right) \times \mathbf{F}_i + \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i \right) \times \left( {}^i R_{i+1} \mathbf{f}_{i+1} \right) + (\text{N9})\end{aligned}\quad (5.8)$$

Startwerte sind  $\mathbf{f}_{n+1} = \mathbf{0} = \mathbf{n}_{n+1}$  bzw. die am Ursprung des Endeffektorkoordinatensystems  $S_n$  bei Kontakt situationen wirkenden Kräfte und Momente ( $S_{n+1} := S_n$ ,  ${}^n R_{n+1} = E_3$ ).

Der Schub- oder Drehgelenksantrieb im Gelenk  $i$  erzeugt eine Kraft oder ein Drehmoment bezüglich der  $z_{i-1}$ -Achse von  $S_{i-1}$ . Aus den bezüglich  $S_i$  dargestellten Vektoren  $\mathbf{f}_i$  und  $\mathbf{n}_i$  können diese aus der dritten Zeile von  ${}^{i-1} R_i \cdot \mathbf{f}_i$  bzw.  ${}^{i-1} R_i \cdot \mathbf{n}_i$  ermittelt werden:

$$\tau_i = \begin{cases} \left( {}^i \mathbf{e}_{z_{i-1}} \right)^T \mathbf{n}_i, & \text{falls } \rho_i = 1 \text{ (Drehgelenk)} \\ \left( {}^i \mathbf{e}_{z_{i-1}} \right)^T \mathbf{f}_i, & \text{falls } \rho_i = 0 \text{ (Schubgelenk).} \end{cases} \quad (5.10)$$

### Zusammenfassung der Newton-Euler Rekursionen

Die Formeln (5.1) bis (5.10) in (NE.1) bis (NE.3):

- Für die Basis (Glied 0) werden die Werte für  $\omega_0, \dot{\omega}_0, \mathbf{v}_0, \dot{\mathbf{v}}_0$  vorgegeben.
- Für  $i = 1, \dots, n$ :

$$\begin{aligned}\omega_i &= {}^i R_{i-1} \left( \omega_{i-1} + \rho_i {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i \right) \\ \dot{\omega}_i &= {}^i R_{i-1} \left( \dot{\omega}_{i-1} + \rho_i \left[ {}^{i-1} \mathbf{e}_{z_{i-1}} \ddot{q}_i + \omega_{i-1} \times \left( {}^{i-1} \mathbf{e}_{z_{i-1}} \dot{q}_i \right) \right] \right) \\ \mathbf{v}_i &= {}^i R_{i-1} \mathbf{v}_{i-1} + \omega_i \times \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i \right) + (1 - \rho_i) {}^i \mathbf{e}_{z_{i-1}} \dot{q}_i \\ \dot{\mathbf{v}}_i &= {}^i R_{i-1} \dot{\mathbf{v}}_{i-1} + \dot{\omega}_i \times \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i \right) + \omega_i \times \left( \omega_i \times \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i \right) \right) \\ &\quad + (1 - \rho_i) \cdot \left[ 2\omega_i \times {}^i \mathbf{e}_{z_{i-1}} \dot{q}_i + {}^i \mathbf{e}_{z_{i-1}} \ddot{q}_i \right] \\ \dot{\mathbf{v}}_{c_i} &= \dot{\mathbf{v}}_i + \dot{\omega}_i \times {}^i \mathbf{r}_{c_i} + \omega_i \times \left( \omega_i \times {}^i \mathbf{r}_{c_i} \right) \\ \mathbf{F}_i &= \begin{cases} m_i \cdot \dot{\mathbf{v}}_{c_i} - m_i \mathbf{g}_i, & \text{falls Einfluss der Gravitation im Schritt (NE.2) berücksichtigt wird} \\ m_i \cdot \dot{\mathbf{v}}_{c_i}, & \text{falls Einfluss der Gravitation im Anfangswert } \dot{\mathbf{v}}_0 = -\mathbf{g} \text{ berücksichtigt wird} \end{cases} \\ \mathbf{N}_i &= {}^{c_i} I_i \cdot \dot{\omega}_i + \omega_i \times ({}^{c_i} I_i \cdot \omega_i)\end{aligned}$$

- Für den Endeffektor (Glied  $n$ ) werden die Werte für  $\mathbf{f}_{n+1}$  und  $\mathbf{n}_{n+1}$  vorgegeben.

- Für  $i = n, \dots, 1$ :

$$\begin{aligned}\mathbf{f}_i &= {}^i R_{i+1} \mathbf{f}_{i+1} + \mathbf{F}_i \\ \mathbf{n}_i &= {}^i R_{i+1} \mathbf{n}_{i+1} + \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i + {}^i \mathbf{r}_{c_i} \right) \times \mathbf{F}_i + \left( {}^i R_{i-1} {}^{i-1} \mathbf{r}_i \right) \times \left( {}^i R_{i+1} \mathbf{f}_{i+1} \right) + \mathbf{N}_i \\ \tau_i &= \begin{cases} \left( {}^i \mathbf{e}_{z_{i-1}} \right)^T \mathbf{n}_i, & \text{falls } \rho_i = 1 \text{ (Drehgelenk)} \\ \left( {}^i \mathbf{e}_{z_{i-1}} \right)^T \mathbf{f}_i, & \text{falls } \rho_i = 0 \text{ (Schubgelenk)} \end{cases}\end{aligned}$$

stellen ein iteratives und rekursives Berechnungsschema dar, um aus  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$  die dafür erforderlichen Drehmomente in den Drehgelenken bzw. Kräfte in den Schubgelenken  $\tau \in \mathbb{R}^n$  numerisch zu berechnen (inverse Dynamik, INV DYN).

Die benötigten „Zutaten“ sind der Trägheitstensor  ${}^{c_i}I_i \in \mathbb{R}^{3 \times 3}$ , die Masse  $m_i \in \mathbb{R}^+$  und die Schwerpunktslage  ${}^i\mathbf{r}_{c_i} \in \mathbb{R}^3$  für jedes Roboterglied sowie die lokale Rotationsmatrix  ${}^{i-1}R_i \in \mathbb{R}^{3 \times 3}$  und der lokale Verschiebungsvektor  ${}^{i-1}\mathbf{r}_i \in \mathbb{R}^3$ , die den Koordinatenwechsel zwischen den lokalen Koordinatensystemen  $S_i$  und  $S_{i-1}$  beschreiben.

## Bemerkungen

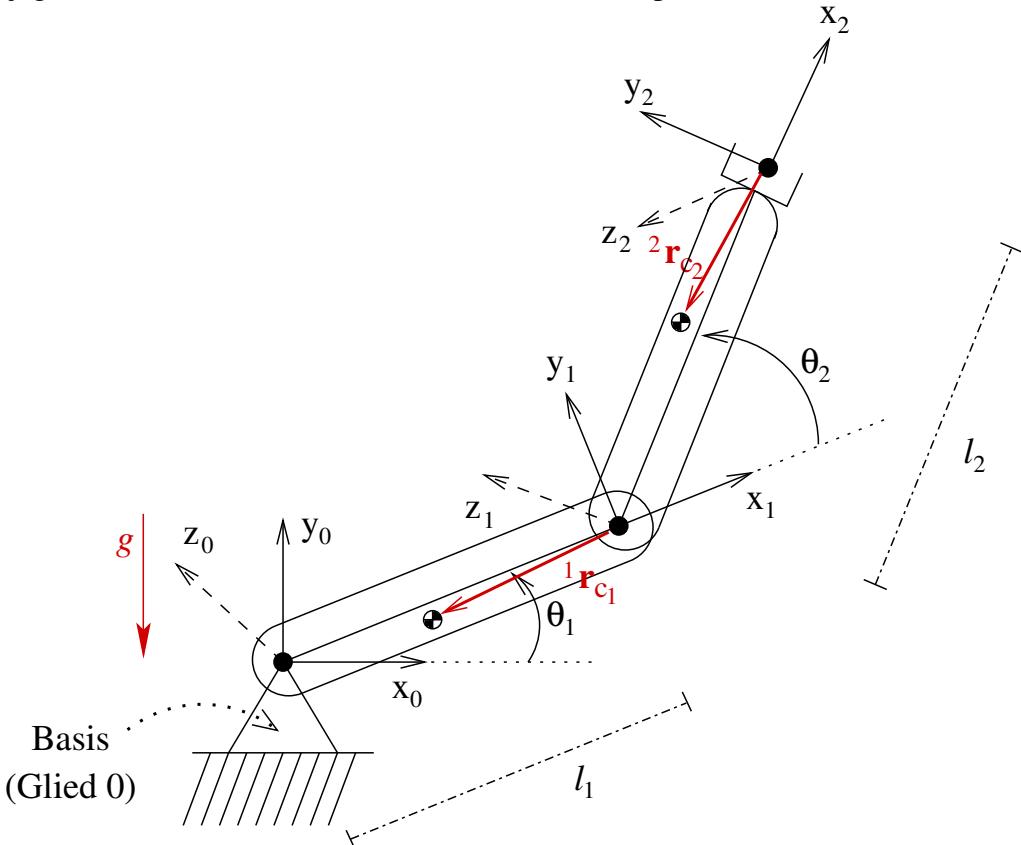
- Die Anbringung einer Traglast am Endeffektor ändert die Daten des letzten Roboterglieds: Masse  $m_n$ , Trägheitstensor  ${}^{c_n}I_n$  und Schwerpunkt  ${}^n\mathbf{r}_{c_n}$ .
- Zur Berechnung von INV DYN wird in der Newton-Euler Rekursion die Berechnung von  $\mathbf{v}_i$  in Gleichung (5.3) bzw. in der Zusammenfassung auf Seite 113 eigentlich nicht benötigt. D.h. in der Rekursion kann die entsprechende Zeile mit Berechnungen eingespart werden.

Die Darstellung der Berechnung von  $\mathbf{v}_i$  dient dem besseren Gesamtverständnis. Die Berechnung von  $\mathbf{v}_{c_i}$  wird jedoch in Kapitel 5.4 für die Berechnung der kinetischen Energie in der Lagrangeschen Formulierung der Roboterdynamik benötigt.

- Die Newton-Euler Rekursion ist in der in Kapitel 5.1, Seite 98, beschriebenen Mehrkörpersystem-Bibliothek MBSLIB zur effizienten numerischen Lösung der Inversdynamik von Mehrkörpersystemen mit Baumstruktur implementiert. Die Methode `dORne()` führt das Verfahren für ein zuvor definiertes Mehrkörpersystem mit angegebenen Gelenkpositionen, -geschwindigkeiten und -beschleunigungen durch und bestimmt so die jeweiligen Gelenkdrehmomente und -kräfte.
- **Literatur zu Abschnitt 5.3.2:**  
Section 6.5 bei [Craig, 1989], Section 6.5 bei [Spong/Vidyasagar, 1989], Section 3.4 bei [Yoshikawa, 1990], Section 7.2 bei [McKerrow, 1991].

### 5.3.3 Beispiel: Newton-Euler Formulierung der Dynamik des ebenen 2-DOF-SCARA-Manipulators

In nahezu allen Lehrbüchern und in den Übungen werden zur Untersuchung der Roboterdynamik nur sehr vereinfachte Modelle betrachtet, die aus Punktmassen mit masselosen Verbindungen bestehen (d.h.  $c_i I_i = O$ ) und deren Schwerpunkte meist als in den Gelenken liegend angenommen werden. Das hier untersuchte Beispiel wird dagegen fast ohne jegliche, vereinfachende Annahmen in voller Komplexität untersucht!



Das Vorwärtsskinematikmodell und die Transformationen  ${}^0T_1(\theta_1)$ ,  ${}^1T_2(\theta_2)$  und  ${}^0T_2(\theta_1, \theta_2)$  wurden bereits im Abschnitt 3.1.3 (Seite 36 folgende) entwickelt. Die Jacobi-Matrix  ${}^0J_2(\theta_1, \theta_2)$  und die Geschwindigkeiten  ${}^0v_2(t)$  und  ${}^0\omega_2(t)$  wurden im Abschnitt 4.3.2 (Seite 73) ermittelt.

Zur Anwendung der Newton-Euler Rekursion stellen wir zunächst die „Zutaten“ bei gegebenem Vorwärtsskinematikmodell nach DH-Konvention zusammen:

- Die lokalen Rotationsmatrizen  ${}^0R_1$  und  ${}^1R_2$  sowie die lokalen Verschiebungsvektoren  ${}^0r_1$  und  ${}^1r_2$  definieren die lokalen, homogenen Transformationsmatrizen  ${}^0T_1$  und  ${}^1T_2$ :

$${}^{i-1}R_i(\theta_i) = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad {}^{i-1}r_i(\theta_i) = \begin{pmatrix} l_i \cos \theta_i \\ l_i \sin \theta_i \\ 0 \end{pmatrix}, \quad i = 1, 2.$$

- Für die beiden Glieder seien die Massen  $m_1$  und  $m_2$  (gemessen in kg), die bezüglich

$S_i$  jeweils konstanten Schwerpunktslagen

$${}^i \mathbf{r}_{c_i} = \begin{pmatrix} {}^i r_{c_i,x} \\ {}^i r_{c_i,y} \\ {}^i r_{c_i,z} \end{pmatrix}, \quad i = 1, 2,$$

sowie die Trägheitstensoren bezüglich  $S_{c_i}$  als Diagonalmatrizen

$${}^{c_i} I_i = \begin{pmatrix} {}^{c_i} I_{i,xx} & 0 & 0 \\ 0 & {}^{c_i} I_{i,yy} & 0 \\ 0 & 0 & {}^{c_i} I_{i,zz} \end{pmatrix}, \quad i = 1, 2,$$

(bei entsprechender symmetrischer Massenverteilung) gegeben.

Beide Gelenke sind Drehgelenke ( $q_1 = \theta_1$ ,  $q_2 = \theta_2$ ), folglich gilt

$$\rho_1 = 1, \quad \rho_2 = 1.$$

### Durchführung von (NE.1) und (NE.2)

Die Basis sei fixiert und die Erdgravitation wirke in Richtung der negativen  $y_0$ -Achse. Damit lauten die linearen und Winkel-Geschwindigkeiten und -Beschleunigungen der Basis:

$$\omega_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \dot{\omega}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{v}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \dot{\mathbf{v}}_0 = -\mathbf{g} = \begin{pmatrix} 0 \\ g \\ 0 \end{pmatrix}, \quad g = 9.81[\text{m/s}].$$

$i = 1:$

$$\begin{aligned} \omega_1 &= {}^1R_0(\omega_0 + \rho_1 {}^0\mathbf{e}_{z_0}\dot{\theta}_1) = ({}^0R_1)^T \left( \mathbf{0} + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{pmatrix} \right) = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{pmatrix} \\ \dot{\omega}_1 &= {}^1R_0 \left( \underbrace{\dot{\omega}_0}_{=\mathbf{0}} + \rho_1 \left[ {}^0\mathbf{e}_{z_0}\ddot{\theta}_1 + \underbrace{\omega_0 \times ({}^0\mathbf{e}_{z_0}\dot{\theta}_1)}_{=\mathbf{0}} \right] \right) \\ &= ({}^0R_1)^T \left( \mathbf{0} + \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{pmatrix} \right) = \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{pmatrix} \\ \mathbf{v}_1 &= {}^1R_0 \underbrace{\mathbf{v}_0}_{=\mathbf{0}} + \omega_1 \times ({}^1R_0 {}^0\mathbf{r}_1) + \underbrace{(1-\rho_1)}_{=0} {}^1\mathbf{e}_{z_0}\dot{\theta}_1 \\ \text{mit } ({}^0R_1)^T {}^0\mathbf{r}_1 &= \begin{pmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta_1 \\ \sin\theta_1 \\ 0 \end{pmatrix} l_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} l_1 \quad \text{folgt} \\ \mathbf{v}_1 &= \omega_1 \times ({}^1R_0 {}^0\mathbf{r}_1) = \dot{\theta}_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} l_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_1 \dot{\theta}_1 \\ \dot{\mathbf{v}}_1 &= {}^1R_0 \dot{\mathbf{v}}_0 + \dot{\omega}_1 \times ({}^1R_0 {}^0\mathbf{r}_1) + \omega_1 \times (\omega_1 \times ({}^1R_0 {}^0\mathbf{r}_1)) \\ &\quad + \underbrace{(1-\rho_1)}_{=0} \cdot [2\omega_1 \times {}^1\mathbf{e}_{z_0}\dot{\theta}_1 + {}^1\mathbf{e}_{z_0}\ddot{\theta}_1] \\ &= \begin{pmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ g \\ 0 \end{pmatrix} + \ddot{\theta}_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} l_1 + \dot{\theta}_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_1 \dot{\theta}_1 \\ &= \begin{pmatrix} \sin\theta_1 \\ \cos\theta_1 \\ 0 \end{pmatrix} g + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_1 \ddot{\theta}_1 + \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} l_1 (\dot{\theta}_1)^2 = \begin{pmatrix} \dot{v}_{1,x} \\ \dot{v}_{1,y} \\ 0 \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
\dot{\mathbf{v}}_{c_1} &= \dot{\mathbf{v}}_1 + \dot{\omega}_1 \times {}^1\mathbf{r}_{c_1} + \omega_1 \times (\omega_1 \times {}^1\mathbf{r}_{c_1}) \\
\text{mit } \dot{\omega}_1 \times {}^1\mathbf{r}_{c_1} &= \ddot{\theta}_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} {}^1r_{c_1,x} \\ {}^1r_{c_1,y} \\ {}^1r_{c_1,z} \end{pmatrix} = \begin{pmatrix} -{}^1r_{c_1,y} \\ {}^1r_{c_1,x} \\ 0 \end{pmatrix} \ddot{\theta}_1 \\
\omega_1 \times {}^1\mathbf{r}_{c_1} &= \begin{pmatrix} -{}^1r_{c_1,y} \\ {}^1r_{c_1,x} \\ 0 \end{pmatrix} \dot{\theta}_1 \\
\omega_1 \times (\omega_1 \times {}^1\mathbf{r}_{c_1}) &= \dot{\theta}_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} -{}^1r_{c_1,y} \\ {}^1r_{c_1,x} \\ 0 \end{pmatrix} \dot{\theta}_1 = \begin{pmatrix} -{}^1r_{c_1,x} \\ -{}^1r_{c_1,y} \\ 0 \end{pmatrix} (\dot{\theta}_1)^2 \quad \text{folgt} \\
\dot{\mathbf{v}}_{c_1} &= \dot{\mathbf{v}}_1 + \begin{pmatrix} -{}^1r_{c_1,y} \\ {}^1r_{c_1,x} \\ 0 \end{pmatrix} \ddot{\theta}_1 + \begin{pmatrix} -{}^1r_{c_1,x} \\ -{}^1r_{c_1,y} \\ 0 \end{pmatrix} (\dot{\theta}_1)^2 = \begin{pmatrix} \dot{v}_{c_1,x} \\ \dot{v}_{c_1,y} \\ 0 \end{pmatrix} \\
\mathbf{F}_1 &= m_1 \cdot \dot{\mathbf{v}}_{c_1} = m_1 \left[ \dot{\mathbf{v}}_1 + \begin{pmatrix} -{}^1r_{c_1,y} \\ {}^1r_{c_1,x} \\ 0 \end{pmatrix} \ddot{\theta}_1 + \begin{pmatrix} -{}^1r_{c_1,x} \\ -{}^1r_{c_1,y} \\ 0 \end{pmatrix} (\dot{\theta}_1)^2 \right] \\
\mathbf{N}_1 &= {}^{c_1}I_1 \cdot \dot{\omega}_1 + \omega_1 \times ({}^{c_1}I_1 \cdot \omega_1) \\
\text{mit } {}^{c_1}I_1 \cdot \dot{\omega}_1 &= \begin{pmatrix} {}^{c_1}I_{1,xx} & 0 & 0 \\ 0 & {}^{c_1}I_{1,yy} & 0 \\ 0 & 0 & {}^{c_1}I_{1,zz} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ {}^{c_1}I_{1,zz} \end{pmatrix} \ddot{\theta}_1 \\
{}^{c_1}I_1 \cdot \omega_1 &= \begin{pmatrix} 0 \\ 0 \\ {}^{c_1}I_{1,zz} \end{pmatrix} \dot{\theta}_1 \\
\omega_1 \times ({}^{c_1}I_1 \cdot \omega_1) &= \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ {}^{c_1}I_{1,zz} \end{pmatrix} \dot{\theta}_1 = \mathbf{0} \quad \text{folgt} \\
\mathbf{N}_1 &= \begin{pmatrix} 0 \\ 0 \\ {}^{c_1}I_{1,zz} \end{pmatrix} \ddot{\theta}_1
\end{aligned}$$

$i = 2:$

$$\begin{aligned}
\omega_2 &= {}^2R_1 \left( \omega_1 + \rho_2 {}^1\mathbf{e}_{z_1} \dot{\theta}_2 \right) = ({}^1R_2)^T \left( \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{pmatrix} \right) = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{pmatrix} \\
\dot{\omega}_2 &= {}^2R_1 \left( \dot{\omega}_1 + \rho_2 \left[ {}^1\mathbf{e}_{z_1} \ddot{\theta}_2 + \underbrace{\omega_1 \times ({}^1\mathbf{e}_{z_1} \dot{\theta}_2)}_{=\mathbf{0}} \right] \right) \\
&= ({}^1R_2)^T \left( \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_2 \end{pmatrix} \right) = \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
\mathbf{v}_2 &= {}^2 R_1 \mathbf{v}_1 + \omega_2 \times \left( {}^2 R_1^{-1} \mathbf{r}_2 \right) + \underbrace{(1-\rho_2)}_{=0} {}^2 \mathbf{e}_{z_1} \dot{\theta}_2 \\
\text{mit } {}^1 R_2^T \cdot \mathbf{v}_1 &= \begin{pmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_1 \dot{\theta}_1 = \begin{pmatrix} \sin \theta_2 \\ \cos \theta_2 \\ 0 \end{pmatrix} l_1 \dot{\theta}_1 \\
\omega_2 \times \left( {}^2 R_1^{-1} \mathbf{r}_2 \right) &= (\dot{\theta}_1 + \dot{\theta}_2) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} l_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_2 (\dot{\theta}_1 + \dot{\theta}_2) \quad \text{folgt} \\
\mathbf{v}_2 &= \begin{pmatrix} \sin \theta_2 \\ \cos \theta_2 \\ 0 \end{pmatrix} l_1 \dot{\theta}_1 + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_2 (\dot{\theta}_1 + \dot{\theta}_2) \\
\dot{\mathbf{v}}_2 &= {}^2 R_1 \dot{\mathbf{v}}_1 + \dot{\omega}_2 \times \left( {}^2 R_1^{-1} \mathbf{r}_2 \right) + \omega_2 \times \left( \omega_2 \times \left( {}^2 R_1^{-1} \mathbf{r}_2 \right) \right) + \underbrace{(1-\rho_2)}_{=0} \cdot [...] \\
\text{mit } {}^1 R_2^T \cdot \dot{\mathbf{v}}_1 &= \begin{pmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left( \begin{pmatrix} \sin \theta_1 \\ \cos \theta_1 \\ 0 \end{pmatrix} g + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_1 \ddot{\theta}_1 + \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} l_1 (\dot{\theta}_1)^2 \right) \\
&= \begin{pmatrix} \sin(\theta_1 + \theta_2) \\ \cos(\theta_1 + \theta_2) \\ 0 \end{pmatrix} g + \begin{pmatrix} \sin \theta_2 \\ \cos \theta_2 \\ 0 \end{pmatrix} l_1 \ddot{\theta}_1 + \begin{pmatrix} -\cos \theta_2 \\ \sin \theta_2 \\ 0 \end{pmatrix} l_1 (\dot{\theta}_1)^2 \\
\dot{\omega}_2 \times \left( {}^2 R_1^{-1} \mathbf{r}_2 \right) &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\
\omega_2 \times \left( \omega_2 \times \left( {}^2 R_1^{-1} \mathbf{r}_2 \right) \right) &= (\dot{\theta}_1 + \dot{\theta}_2) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_2 (\dot{\theta}_1 + \dot{\theta}_2) = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\
\text{folgt} \\
\dot{\mathbf{v}}_2 &= \begin{pmatrix} \sin(\theta_1 + \theta_2) \\ \cos(\theta_1 + \theta_2) \\ 0 \end{pmatrix} g + \begin{pmatrix} \sin \theta_2 \\ \cos \theta_2 \\ 0 \end{pmatrix} l_1 \ddot{\theta}_1 + \begin{pmatrix} -\cos \theta_2 \\ \sin \theta_2 \\ 0 \end{pmatrix} l_1 (\dot{\theta}_1)^2 \\
&\quad + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 = \begin{pmatrix} \dot{v}_{2,x} \\ \dot{v}_{2,y} \\ 0 \end{pmatrix} \\
\dot{\mathbf{v}}_{c_2} &= \dot{\mathbf{v}}_2 + \dot{\omega}_2 \times {}^2 \mathbf{r}_{c_2} + \omega_2 \times \left( \omega_2 \times {}^2 \mathbf{r}_{c_2} \right) \\
\text{mit } \dot{\omega}_2 \times {}^2 \mathbf{r}_{c_2} &= (\ddot{\theta}_1 + \ddot{\theta}_2) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} {}^2 r_{c_2,x} \\ {}^2 r_{c_2,y} \\ {}^2 r_{c_2,z} \end{pmatrix} = \begin{pmatrix} -{}^2 r_{c_2,y} \\ {}^2 r_{c_2,x} \\ 0 \end{pmatrix} (\ddot{\theta}_1 + \ddot{\theta}_2) \\
\omega_2 \times \left( \omega_2 \times {}^2 \mathbf{r}_{c_2} \right) &= (\dot{\theta}_1 + \dot{\theta}_2) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} -{}^2 r_{c_2,y} \\ {}^2 r_{c_2,x} \\ 0 \end{pmatrix} (\dot{\theta}_1 + \dot{\theta}_2) = \begin{pmatrix} -{}^2 r_{c_2,x} \\ -{}^2 r_{c_2,y} \\ 0 \end{pmatrix} (\dot{\theta}_1 + \dot{\theta}_2)^2 \\
\text{folgt} \\
\dot{\mathbf{v}}_{c_2} &= \dot{\mathbf{v}}_2 + \begin{pmatrix} -{}^2 r_{c_2,y} \\ {}^2 r_{c_2,x} \\ 0 \end{pmatrix} (\ddot{\theta}_1 + \ddot{\theta}_2) + \begin{pmatrix} -{}^2 r_{c_2,x} \\ -{}^2 r_{c_2,y} \\ 0 \end{pmatrix} (\dot{\theta}_1 + \dot{\theta}_2)^2 = \begin{pmatrix} \dot{v}_{c_2,x} \\ \dot{v}_{c_2,y} \\ 0 \end{pmatrix}
\end{aligned}$$

$$\mathbf{F}_2 = m_2 \cdot \dot{\mathbf{v}}_{c_2}$$

$$\mathbf{N}_2 = {}^c I_2 \cdot \dot{\omega}_2 + \omega_2 \times ({}^c I_2 \cdot \omega_2)$$

mit  ${}^c I_2 \cdot \dot{\omega}_2 = \begin{pmatrix} {}^c I_{2,xx} & 0 & 0 \\ 0 & {}^c I_{2,yy} & 0 \\ 0 & 0 & {}^c I_{2,zz} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ {}^c I_{2,zz} \end{pmatrix} (\ddot{\theta}_1 + \ddot{\theta}_2)$

$$\omega_2 \times ({}^c I_2 \cdot \omega_2) = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ {}^c I_{2,zz} \end{pmatrix} (\dot{\theta}_1 + \dot{\theta}_2) = \mathbf{0} \text{ folgt}$$

$$\mathbf{N}_2 = \begin{pmatrix} 0 \\ 0 \\ {}^c I_{2,zz} \end{pmatrix} (\ddot{\theta}_1 + \ddot{\theta}_2)$$

### Durchführung von (NE.3)

Rekursion vom Endeffektor zur Basis zur Berechnung der in den Gelenken wirkenden Kräfte und Momente: Der Endeffektor bewege sich frei im Raum. Dann wirken am Endeffektor keine Kräfte und Momente. Die Startwerte der Rekursion ergeben sich daher zu:

$$\mathbf{f}_3 = \mathbf{0}, \quad \mathbf{n}_3 = \mathbf{0}.$$

$$[i=2:]$$

$$\mathbf{f}_2 = {}^2 R_3 \underbrace{\mathbf{f}_3}_{=\mathbf{0}} + \mathbf{F}_2 = \mathbf{F}_2 = m_2 \cdot \dot{\mathbf{v}}_{c_2}$$

$$\begin{aligned} \mathbf{n}_2 &= {}^2 R_3 \underbrace{\mathbf{n}_3}_{=\mathbf{0}} + ({}^2 R_1^{-1} \mathbf{r}_2 + {}^2 \mathbf{r}_{c_2}) \times \mathbf{F}_2 + ({}^2 R_1^{-1} \mathbf{r}_2) \times \left( {}^2 R_3 \underbrace{\mathbf{f}_3}_{=\mathbf{0}} \right) + \mathbf{N}_2 \\ &= m_2 \left( \begin{pmatrix} l_2 \\ 0 \\ 0 \end{pmatrix} + {}^2 \mathbf{r}_{c_2} \right) \times \dot{\mathbf{v}}_{c_2} + \mathbf{N}_2 = m_2 \begin{pmatrix} l_2 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} \dot{v}_{c_2,x} \\ \dot{v}_{c_2,y} \\ 0 \end{pmatrix} + m_2 {}^2 \mathbf{r}_{c_2} \times \dot{\mathbf{v}}_{c_2} + \mathbf{N}_2 \\ &= m_2 l_2 \begin{pmatrix} 0 \\ 0 \\ \dot{v}_{c_2,y} \end{pmatrix} + m_2 \begin{pmatrix} {}^2 r_{c_2,x} \\ {}^2 r_{c_2,y} \\ {}^2 r_{c_2,z} \end{pmatrix} \times \left[ \begin{pmatrix} \sin(\theta_1 + \theta_2) \\ \cos(\theta_1 + \theta_2) \\ 0 \end{pmatrix} g + \begin{pmatrix} \sin \theta_2 \\ \cos \theta_2 \\ 0 \end{pmatrix} l_1 \ddot{\theta}_1 \right. \\ &\quad \left. + \begin{pmatrix} -\cos \theta_2 \\ \sin \theta_2 \\ 0 \end{pmatrix} l_1 (\dot{\theta}_1)^2 + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \right. \\ &\quad \left. + \begin{pmatrix} -{}^2 r_{c_2,y} \\ {}^2 r_{c_2,x} \\ 0 \end{pmatrix} (\ddot{\theta}_1 + \ddot{\theta}_2) + \begin{pmatrix} -{}^2 r_{c_2,x} \\ -{}^2 r_{c_2,y} \\ 0 \end{pmatrix} (\dot{\theta}_1 + \dot{\theta}_2)^2 \right] + \begin{pmatrix} 0 \\ 0 \\ {}^c I_{2,zz} \end{pmatrix} (\ddot{\theta}_1 + \ddot{\theta}_2) \end{aligned}$$

Bevor  $\mathbf{n}_2$  nun vollständig ausgerechnet wird, lohnt sich ein Blick voraus auf das, was eigentlich gesucht ist, nämlich das Drehmoment  $\tau_2$  des Antriebs des zweiten Gelenks:

$$\tau_2 = ({}^2 \mathbf{e}_{z_1})^T \mathbf{n}_2 = ({}^2 R_1^{-1} \mathbf{e}_{z_1})^T \mathbf{n}_2 = (0 \ 0 \ 1) \cdot \mathbf{n}_2 = n_{2,z}.$$

D.h. von  $\mathbf{n}_2$  wird nur die dritte Komponente benötigt!

$$\begin{aligned}
 n_{2,z} &= m_2 l_2 \left\{ \cos(\theta_1 + \theta_2) g + \cos \theta_2 l_1 \ddot{\theta}_1 + \sin \theta_2 l_1 (\dot{\theta}_1)^2 + l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \right. \\
 &\quad \left. + {}^2 r_{c_2,x} (\ddot{\theta}_1 + \ddot{\theta}_2) - {}^2 r_{c_2,y} (\dot{\theta}_1 + \dot{\theta}_2)^2 \right\} \\
 &+ m_2 \left[ \left( {}^2 r_{c_2,x} \cos(\theta_1 + \theta_2) - {}^2 r_{c_2,y} \sin(\theta_1 + \theta_2) \right) g \right. \\
 &\quad + \left( {}^2 r_{c_2,x} \cos \theta_2 - {}^2 r_{c_2,y} \sin \theta_2 \right) l_1 \ddot{\theta}_1 + \left( {}^2 r_{c_2,x} \sin \theta_2 + {}^2 r_{c_2,y} \cos \theta_2 \right) l_1 (\dot{\theta}_1)^2 \\
 &\quad \left. + {}^2 r_{c_2,x} l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + {}^2 r_{c_2,y} l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + \left( ({}^2 r_{c_2,x})^2 + ({}^2 r_{c_2,y})^2 \right) (\ddot{\theta}_1 + \ddot{\theta}_2) + 0 \right] \\
 &+ {}^{c_2} I_{2,zz} (\ddot{\theta}_1 + \ddot{\theta}_2)
 \end{aligned}$$

Die Terme auf der rechten Seite von  $\tau_2 = n_{2,z}$  können in drei Gruppen sortiert werden:

a) Terme mit Faktor  $\ddot{\theta}_i$ , b) Terme mit Faktor  $\dot{\theta}_i$  und c) Terme mit Faktor  $g$ :

$$\begin{aligned}
 \tau_2 &= m_2 l_2 \cos \theta_2 l_1 \ddot{\theta}_1 + m_2 l_2 l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 \left( {}^2 r_{c_2,x} \cos \theta_2 - {}^2 r_{c_2,y} \sin \theta_2 \right) l_1 \ddot{\theta}_1 \quad \left. \right\} a) \\
 &\quad + 2 m_2 {}^2 r_{c_2,x} l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 \left( ({}^2 r_{c_2,x})^2 + ({}^2 r_{c_2,y})^2 \right) (\ddot{\theta}_1 + \ddot{\theta}_2) + {}^{c_2} I_{2,zz} (\ddot{\theta}_1 + \ddot{\theta}_2) \quad \left. \right\} a) \\
 &\quad + m_2 l_2 \sin \theta_2 l_1 (\dot{\theta}_1)^2 + m_2 \left( {}^2 r_{c_2,x} \sin \theta_2 + {}^2 r_{c_2,y} \cos \theta_2 \right) l_1 (\dot{\theta}_1)^2 \quad \left. \right\} b) \\
 &\quad + m_2 l_2 \cos(\theta_1 + \theta_2) g + m_2 \left( {}^2 r_{c_2,x} \cos(\theta_1 + \theta_2) - {}^2 r_{c_2,y} \sin(\theta_1 + \theta_2) \right) g \quad \left. \right\} c) \\
 &=: (M_{2,1}(\theta) \ M_{2,2}(\theta)) \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + C_2(\theta, \dot{\theta}) + G_2(\theta)
 \end{aligned}$$

Dabei werden neben  $\theta = (\theta_1, \theta_2)^T$  folgende Abkürzungen verwendet:

$$\begin{aligned}
 M_{2,1}(\theta) &= m_2 \left[ ({}^2 r_{c_2,x} + l_2)^2 + ({}^2 r_{c_2,y})^2 + l_1 l_2 \cos \theta_2 + l_1 \left( {}^2 r_{c_2,x} \cos \theta_2 - {}^2 r_{c_2,y} \sin \theta_2 \right) \right] + {}^{c_2} I_{2,zz} \\
 M_{2,2}(\theta) &= m_2 \left[ ({}^2 r_{c_2,x} + l_2)^2 + ({}^2 r_{c_2,y})^2 \right] + {}^{c_2} I_{2,zz} \\
 C_2(\theta, \dot{\theta}) &= m_2 l_1 \left[ ({}^2 r_{c_2,x} + l_2) \sin \theta_2 + {}^2 r_{c_2,y} \cos \theta_2 \right] (\dot{\theta}_1)^2 \\
 G_2(\theta) &= m_2 g \left( (l_2 + {}^2 r_{c_2,x}) \cos(\theta_1 + \theta_2) - {}^2 r_{c_2,y} \sin(\theta_1 + \theta_2) \right)
 \end{aligned}$$

$i = 1:$

$$\begin{aligned}\mathbf{f}_1 &= {}^1R_2 \mathbf{f}_2 + \mathbf{F}_1 \\ \mathbf{n}_1 &= {}^1R_2 \mathbf{n}_2 + \left({}^1R_0 {}^0 \mathbf{r}_1 + {}^1\mathbf{r}_{c_1}\right) \times \mathbf{F}_1 + \left({}^1R_0 {}^0 \mathbf{r}_1\right) \times \left({}^1R_2 \mathbf{f}_2\right) + \mathbf{N}_1 \\ \tau_1 &= \left({}^1\mathbf{e}_{z_0}\right)^T \mathbf{n}_1 = \left({}^1R_0 {}^0 \mathbf{e}_{z_0}\right)^T \mathbf{n}_1 = (0 \ 0 \ 1) \cdot \mathbf{n}_1 = n_{1,z}.\end{aligned}$$

D.h. auch von  $\mathbf{n}_1$  wird nur die dritte Komponente benötigt!

$$\begin{aligned}\mathbf{n}_1 &= {}^1R_2 \mathbf{n}_2 + \left({}^1R_0 {}^0 \mathbf{r}_1\right) \times \mathbf{F}_1 + {}^1\mathbf{r}_{c_1} \times \mathbf{F}_1 + \left({}^1R_0 {}^0 \mathbf{r}_1\right) \times \left({}^1R_2 \mathbf{f}_2\right) + \mathbf{N}_1 \\ \text{mit } {}^1R_2 \mathbf{n}_2 &= \begin{pmatrix} * \\ * \\ n_{2,z} \end{pmatrix} \\ \left({}^1R_0 {}^0 \mathbf{r}_1\right) \times \mathbf{F}_1 &= m_1 \begin{pmatrix} l_1 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} \dot{v}_{c_1,x} \\ \dot{v}_{c_1,y} \\ 0 \end{pmatrix} = m_1 l_1 \begin{pmatrix} 0 \\ 0 \\ \dot{v}_{c_1,y} \end{pmatrix} \\ {}^1\mathbf{r}_{c_1} \times \mathbf{F}_1 &= m_1 \begin{pmatrix} {}^1r_{c_1,x} \\ {}^1r_{c_1,y} \\ {}^1r_{c_1,z} \end{pmatrix} \times \begin{pmatrix} \dot{v}_{c_1,x} \\ \dot{v}_{c_1,y} \\ 0 \end{pmatrix} = m_1 \begin{pmatrix} * \\ * \\ {}^1r_{c_1,x} \dot{v}_{c_1,y} - {}^1r_{c_1,y} \dot{v}_{c_1,x} \end{pmatrix} \\ {}^1R_2 \mathbf{f}_2 &= m_2 \begin{pmatrix} \cos \theta_2 & -\sin \theta_2 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{v}_{c_2,x} \\ \dot{v}_{c_2,y} \\ 0 \end{pmatrix} = \begin{pmatrix} \cos \theta_2 \dot{v}_{c_2,x} - \sin \theta_2 \dot{v}_{c_2,y} \\ \sin \theta_2 \dot{v}_{c_2,x} + \cos \theta_2 \dot{v}_{c_2,y} \\ 0 \end{pmatrix} =: \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \\ \left({}^1R_0 {}^0 \mathbf{r}_1\right) \times \left({}^1R_2 \mathbf{f}_2\right) &= m_2 \begin{pmatrix} l_1 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} = m_2 \begin{pmatrix} 0 \\ 0 \\ l_1 b \end{pmatrix} = m_2 l_1 \begin{pmatrix} 0 \\ 0 \\ \sin \theta_2 \dot{v}_{c_2,x} + \cos \theta_2 \dot{v}_{c_2,y} \end{pmatrix}\end{aligned}$$

folgt

$$\begin{aligned}n_{1,z} &= n_{2,z} + m_1 l_1 \dot{v}_{c_1,y} + m_1 \left( {}^1r_{c_1,x} \dot{v}_{c_1,y} - {}^1r_{c_1,y} \dot{v}_{c_1,x} \right) + m_2 l_1 (\sin \theta_2 \dot{v}_{c_2,x} + \cos \theta_2 \dot{v}_{c_2,y}) + {}^{c_1}I_{1,zz} \ddot{\theta}_1 \\ &= n_{2,z} - m_1 {}^1r_{c_1,y} \dot{v}_{c_1,x} + m_1 \left( l_1 + {}^1r_{c_1,x} \right) \dot{v}_{c_1,y} + m_2 l_1 (\sin \theta_2 \dot{v}_{c_2,x} + \cos \theta_2 \dot{v}_{c_2,y}) + {}^{c_1}I_{1,zz} \ddot{\theta}_1\end{aligned}$$

Nach Einsetzen von  $n_{2,z}$ ,  $\dot{v}_{c_1,x}$ ,  $\dot{v}_{c_2,x}$ ,  $\dot{v}_{c_1,y}$  und  $\dot{v}_{c_2,y}$ , entsprechenden Umformulierungen und Umsortierungen lässt sich auch  $\tau_1 = n_{z,1}$  analog zu  $\tau_2$  in folgender Form darstellen:

$$\tau_1 =: (M_{1,1}(\theta) \ M_{1,2}(\theta)) \begin{pmatrix} \dot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + C_1(\theta, \dot{\theta}) + G_1(\theta)$$

Somit hält man die explizite Darstellung des inversen Dynamikmodells für den untersuchten, zweigelenkigen Manipulator in der Form:

$$\text{INV DYN: } \tau = M(\theta) \cdot \ddot{\theta} + \mathbf{C}(\theta, \dot{\theta}) + \mathbf{G}(\theta)$$

mit den Abkürzungen

$$\begin{aligned}\tau &= \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}, \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \quad \dot{\theta} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \quad \ddot{\theta} = \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} \\ M(\theta) &= \begin{pmatrix} M_{1,1}(\theta_1, \theta_2) & M_{1,2}(\theta_1, \theta_2) \\ M_{2,1}(\theta_1, \theta_2) & M_{2,2}(\theta_1, \theta_2) \end{pmatrix}, \quad \mathbf{C}(\theta, \dot{\theta}) = \begin{pmatrix} C_1(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2) \\ C_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2) \end{pmatrix}, \quad \mathbf{G}(\theta) = \begin{pmatrix} G_1(\theta_1, \theta_2) \\ G_2(\theta_1, \theta_2) \end{pmatrix}.\end{aligned}$$

### Weitere Bemerkungen zur rekursiven Newton-Euler Formulierung

- Durch sukzessives, symbolisches Einsetzen und Ausrechnen der Rekursionsformeln (NE.1) bis (NE.3) können die Bewegungsdifferentialgleichungen prinzipiell auch in geschlossener Form dargestellt werden und zwar in der Form — Angabe ohne Beweis —:

$$\text{INV DYN: } \tau = M(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \quad (5.11)$$

mit

$$\begin{aligned} M(\mathbf{q}) &\in \mathbb{R}^{n \times n}: & \text{symmetrische, positiv definite Massenmatrix,} \\ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) &\in \mathbb{R}^n: & \text{Vektor der Zentrifugal- und Coriolisanteile,} \\ \mathbf{G}(\mathbf{q}) &\in \mathbb{R}^n: & \text{Vektor der Gravitationsanteile.} \end{aligned}$$

- Die formale Auflösung des Gleichungssystems nach  $\ddot{\mathbf{q}}$  ergibt das direkte Dynamikmodell als ein System von Differentialgleichungen zweiter Ordnung für die unbekannten, zeitabhängigen Gelenkvariablen  $\mathbf{q}(t)$  in Abhängigkeit von den zeitabhängigen Antriebsmomenten und -kräften  $\tau(t)$ :

$$\text{DIR DYN: } \ddot{\mathbf{q}} = M^{-1}(\mathbf{q}) \cdot [\tau - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q})]$$

Daraus kann  $\mathbf{q}(t)$ ,  $0 \leq t \leq t_f$ , durch numerische Integration bei gegebener Anfangsposition  $\mathbf{q}(0)$  und Anfangsgeschwindigkeit  $\dot{\mathbf{q}}(0)$  in Gelenkvariablen bestimmt werden, falls der zeitliche Verlauf der Antriebskräfte bzw. -momente  $\tau(t)$ ,  $0 \leq t \leq t_f$ , bekannt ist oder vorgegeben wird (siehe Kap. 5.5.3).

Die in DIR DYN auftretende Matrix  $M^{-1}(\mathbf{q})$  wird in der Praxis niemals explizit berechnet. Aufgrund der speziellen Struktur kann DIR DYN mit geeigneten Verfahren sogar mit  $\mathcal{O}(n)$  Operationen berechnet werden (siehe z.B. Kapitel 5.6.2).

- Die Newton-Euler Rekursionen (NE.1) bis (NE.3) werden üblicherweise nur mit Zahlenwerten durchlaufen, so dass die Methode von Newton-Euler im Allgemeinen nur eine implizite, numerische Darstellung von INV DYN liefert. Man gibt aktuelle Zahlenwerte für  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$  hinein und bekommt die zugehörigen Werte  $\tau$  heraus. Das Newton-Euler Verfahren wird im Allgemeinen *nicht* zur Berechnung einer expliziten Formeldarstellung der inversen Dynamik verwendet.
- Falls eine explizite Formel-Darstellung von INV DYN gewünscht ist, ist ein symbolisches Vorgehen wie die Lagrangesche Formulierung der Dynamik in Kapitel 5.4 sinnvoller als die symbolische Auflösung der Newton-Euler Rekursion. Im hier besprochenen Beispiel wurde die explizite Form der Manipulatordynamik durch Auflösen der Newton-Euler-Rekursion nur zur Demonstration und zur Erläuterung ermittelt.
- Im untersuchten Beispiel wurden nicht alle Trägheiten der beiden Manipulatorglieder benötigt, sondern nur  ${}^{c_1}I_{1,zz}$  und  ${}^{c_2}I_{2,zz}$ ! Im Allgemeinen werden nur diejenigen Hauptträgheitsmomente benötigt, die denjenigen Hauptträgheitsachsen entsprechen, um die tatsächlich eine Bewegung des Roboterglieds stattfinden kann.
- Durch vereinfachende Annahmen, die allerdings selten der Realität entsprechen, kann die explizite Formulierung der Dynamik teilweise erheblich vereinfacht werden:

- Betrachtet man das  $i$ -te Roboterglied idealisiert als reine Punktmasse ohne räumliche Ausdehnung und mit masselosen Verbindungen zwischen den jeweiligen Gelenken, so ist der zugehörige Trägheitstensor  ${}^c_i I_i$  die Nullmatrix.
- Betrachtet man den Schwerpunkt des  $i$ -ten Körpers als im Ursprung von  $S_i$  liegend, so ist der zugehörige Koordinatenvektor des Schwerpunkts  ${}^i \mathbf{r}_{c_i} = \mathbf{0}$ . Falls der Schwerpunkt im Ursprung von  $S_{i-1}$  liegt, so ist  ${}^i \mathbf{r}_{c_i} = -{}^i R_{i-1} {}^{i-1} \mathbf{r}_i$  (vgl. Abbildung auf Seite 112). In beiden Fällen kann die Rekursionsformel (5.5) zur Berechnung von  $\dot{\mathbf{v}}_{c_i}$  aus  $\dot{\mathbf{v}}_i$  vereinfacht werden.
- Wie am Beispiel ersichtlich ist, gilt in der Newton-Euler Rekursion die Beziehung

$$\frac{d(\mathbf{v}_i)}{dt} \neq \dot{\mathbf{v}}_i$$

und **nicht** „ $\frac{d\mathbf{v}_i}{dt} = \dot{\mathbf{v}}_i$ “, obwohl es die Notation suggerieren könnte! Die Größen  $\mathbf{v}_i$  und  $\dot{\mathbf{v}}_i$  beschreiben den linearen Geschwindigkeits- bzw. Beschleunigungsvektor des  $i$ -ten Glieds bzgl. des festen Basiskoordinatensystems, die im lokalen Koordinatensystem  $S_i$  dargestellt werden (vgl. Seite 106), d.h.

$$\mathbf{v}_i = {}^i R_0 \cdot {}^0 \mathbf{v}_i, \quad \dot{\mathbf{v}}_i = {}^i R_0 \cdot {}^0 \dot{\mathbf{v}}_i$$

und damit  $\frac{d}{dt}(\mathbf{v}_i) = \frac{d}{dt}({}^i R_0 \cdot {}^0 \mathbf{v}_i) = ({}^i R_0) \cdot {}^0 \mathbf{v}_i + {}^i R_0 {}^0 \dot{\mathbf{v}}_i = ({}^i R_0) \cdot {}^0 \mathbf{v}_i + \dot{\mathbf{v}}_i$ .

Für Winkelgeschwindigkeit  $\omega_i = {}^i R_0 {}^0 \omega_i$  und Winkelbeschleunigung  $\dot{\omega}_i = {}^i R_0 {}^0 \dot{\omega}_i$  gilt wegen

$$\begin{aligned} ({}^i R_0) \cdot &= (({}^0 R_i) \cdot)^T = ({}^0 R_i)^T B ({}^0 \omega_i)^T = {}^i R_0 (-B({}^0 \omega_i)), \\ \text{jedoch dass } (\omega_i) \cdot &= \frac{d}{dt}(\omega_i) = \frac{d}{dt}({}^i R_0 \cdot {}^0 \omega_i) \\ &= ({}^i R_0) \cdot {}^0 \omega_i + {}^i R_0 {}^0 \dot{\omega}_i = -{}^i R_0 ({}^0 \omega_i \times {}^0 \omega_i) + \dot{\omega}_i \\ &= \dot{\omega}_i. \end{aligned}$$

## 5.4 Lagrangesche Formulierung der Roboterdynamik

Die Formulierung der Roboterdynamik nach Newton-Euler beruht auf Impuls- und Drallsatz und der **Kräfte- und Momentenbilanz** zwischen den Robotergliedern. Die Lagrangesche Formulierung ist dagegen **energiebasiert**. Obwohl beide Formalismen ganz unterschiedlich arbeiten, liefern sie dennoch prinzipiell (und bei exakter Arithmetik) **dieselben Bewegungsgleichungen** INV DYN, nur eben auf unterschiedliche Berechnungsweisen.

### 5.4.1 Kinetische und potentielle Energie, Lagrangefunktion

Die Energie eines mechanischen Systems setzt sich aus *kinetischer* und *potentieller* Energie zusammen.

#### Kinetische Energie

Zunächst betrachten wir die kinetische Energie (engl. *kinetic energy*)  $K_i \in \mathbb{R}$  des  $i$ -ten Roboterglieds in der kinematischen Kette:

$$K_i = K_i(\mathbf{q}, \dot{\mathbf{q}}) = \underbrace{\frac{1}{2} m_i \left( {}^0 \mathbf{v}_{c_i} \right)^T {}^0 \mathbf{v}_{c_i}}_{(*)1} + \underbrace{\frac{1}{2} {}^0 \omega_i^T c_i I_i {}^0 \omega_i}_{(*)2} \quad (5.12)$$

Diese setzt sich aus zwei Teilen zusammen:

- (\*)1) der kinetischen Energie der linearen Geschwindigkeit des Schwerpunkts des  $i$ -ten Glieds und
- (\*)2) der kinetischen Energie der Winkelgeschwindigkeit des Schwerpunkts des  $i$ -ten Glieds.

Analog zur in Kapitel 4.3 (Seiten 66 und 72) ermittelten Darstellung von  ${}^0 \mathbf{v}_n$  und  ${}^0 \omega_n$  mit Jacobi-Matrizen können lineare und Winkelgeschwindigkeit des  $i$ -ten Glied dargestellt werden als

$${}^0 \mathbf{v}_{c_i} = {}^0 J_{c_i, v}(\mathbf{q}) \dot{\mathbf{q}}, \quad {}^0 \omega_i = {}^0 J_{i, \omega}(\mathbf{q}) \dot{\mathbf{q}}.$$

Damit kann die Darstellung der kinetischen Energie  $K_i$  in Gleichung (5.12) umgeformt werden zu

$$\begin{aligned} K_i(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{1}{2} m_i \left( {}^0 \mathbf{v}_{c_i} \right)^T {}^0 \mathbf{v}_{c_i} + \frac{1}{2} {}^0 \omega_i^T c_i I_i {}^0 \omega_i \\ &= \frac{1}{2} m_i \left( {}^0 J_{c_i, v} \dot{\mathbf{q}} \right)^T \left( {}^0 J_{c_i, v} \dot{\mathbf{q}} \right) + \frac{1}{2} \left( {}^0 J_{i, \omega} \dot{\mathbf{q}} \right)^T c_i I_i \left( {}^0 J_{i, \omega} \dot{\mathbf{q}} \right) \\ &= \frac{1}{2} \dot{\mathbf{q}}^T \left( m_i \left( {}^0 J_{c_i, v} \right)^T {}^0 J_{c_i, v} + \left( {}^0 J_{i, \omega} \right)^T c_i I_i {}^0 J_{i, \omega} \right) \dot{\mathbf{q}}. \end{aligned} \quad (5.13)$$

Dabei hängen die  $(3 \times n)$ -Matrizen  ${}^0 J_{c_i, v}$  und  ${}^0 J_{i, \omega}$  von  $\mathbf{q}$  ab.

Die **kinetische Energie  $K$  des Manipulators** erhält man als Summe der kinetischen Energien aller Glieder:

$$\begin{aligned}
 K = K(\mathbf{q}, \dot{\mathbf{q}}) &= \sum_{i=1}^n K_i(\mathbf{q}, \dot{\mathbf{q}}) \\
 &= \frac{1}{2} \dot{\mathbf{q}}^T \underbrace{\left( \sum_{i=1}^n \left( m_i \left( {}^0 J_{c_i, v} \right)^T {}^0 J_{c_i, v} + \left( {}^0 J_{i, \omega} \right)^T c_i I_i {}^0 J_{i, \omega} \right) \right)}_{:= M(\mathbf{q})} \dot{\mathbf{q}} \\
 &= \frac{1}{2} \dot{\mathbf{q}}^T \cdot M(\mathbf{q}) \cdot \dot{\mathbf{q}}. \tag{5.14}
 \end{aligned}$$

Dabei bezeichnet  $M(\mathbf{q})$  die symmetrische, positiv definite  $(n \times n)$ -Massenmatrix (engl. mass matrix). Gleichung (5.14) ist die Verallgemeinerung der aus der Schule bekannten kinetischen Energie einer Punktmasse:  $K = \frac{1}{2}mv^2 = \frac{1}{2}v \cdot m \cdot v$ .

## Potentielle Energie

Die **potentielle Energie** (engl. potential energy)  $P_i \in \mathbb{R}$  des  $i$ -ten Roboterglieds ist

$$P_i = -m_i \mathbf{g}^T {}^0 \mathbf{r}_{c_i} + P_{\text{Ref}, i} \tag{5.15}$$

mit  $\mathbf{g} \in \mathbb{R}^3$ : Vektor der Beschleunigung durch Gravitation dargestellt in  $S_0$ ,

$$\text{z.B. } \mathbf{g} = \begin{pmatrix} 0 \\ -9.81 \\ 0 \end{pmatrix} \text{ m/s}^2 \text{ wie im Beispiel 5.4.2.}$$

$P_{\text{Ref}, i} \in \mathbb{R}$ : Konstante, die der Definition bzgl. einer (festen, aber beliebig wählbaren) Referenzhöhe entspricht (z.B. so dass der minimal mögliche Wert von  $P_i$  gleich Null ist). Diese Konstante fällt bei der Lagrangeschen Formulierung der Dynamik später wieder heraus.

Die **potentielle Energie  $P$  des Manipulators** erhält man dann als

$$P = \sum_{i=1}^n P_i = P(\mathbf{q}), \tag{5.16}$$

d.h.  $P$  ist eine Funktion von  $\mathbf{q}$ , da  ${}^0 \mathbf{r}_{c_i}$  von  $\mathbf{q}$  abhängt (genau genommen nicht von ganz  $\mathbf{q}$  sondern nur von  $q_1, \dots, q_i$ ).

## Lagrangefunktion

Die **Lagrangefunktion** (engl. Lagrangian)  $\mathcal{L}$  eines Manipulators ist definiert als die Differenz von kinetischer und potentieller Energie:

$$\begin{aligned}
 \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) &= K(\mathbf{q}, \dot{\mathbf{q}}) - P(\mathbf{q}) \\
 &= \frac{1}{2} \dot{\mathbf{q}}^T \left( \sum_{i=1}^n \left( m_i \left( {}^0 J_{c_i, v} \right)^T {}^0 J_{c_i, v} + \left( {}^0 J_{i, \omega} \right)^T c_i I_i {}^0 J_{i, \omega} \right) \right) \dot{\mathbf{q}} - \sum_{i=1}^n \left( -m_i \mathbf{g}^T {}^0 \mathbf{r}_{c_i} + P_{\text{Ref}, i} \right). \tag{5.17}
 \end{aligned}$$

Wenn **keine** Antriebskräfte und -momente  $\tau$  wirken (d.h.  $\tau = \mathbf{0} \in \mathbb{R}^n$ ), verläuft die Bewegung  $\mathbf{q}(t)$ ,  $0 \leq t \leq t_f$ , des Manipulators so, dass

$$\int_0^{t_f} \mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) dt \quad \text{minimal wird!}$$

Diese Optimierungsaufgabe ist ein unendlich dimensionales Optimierungsproblem und gehört zur Klasse der sogenannten Variationsprobleme. In der Physik wird diese auch als „Prinzip der kleinsten Wirkung“ (principle of least action)<sup>6</sup> oder Hamiltonsches Prinzip bezeichnet. In Physik und Mechanik wird gezeigt, dass dann die resultierende Bewegung die **Euler-Lagrangischen Differentialgleichungen**

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = 0, \quad i = 1, \dots, n$$

während der Dauer der Bewegung, d.h. in  $0 \leq t \leq t_f$ , erfüllen muss.

Bei gegebenen Kräften bzw. Momenten  $\tau(t)$ ,  $0 \leq t \leq t_f$ , verläuft die Bewegung  $\mathbf{q}(t)$ ,  $0 \leq t \leq t_f$ , so dass

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i, \quad 0 \leq t \leq t_f, \quad i = 1, \dots, n \quad (5.18)$$

bzw. nach Einsetzen von (5.17)

$$\frac{d}{dt} \frac{\partial K(\mathbf{q}(t), \dot{\mathbf{q}}(t))}{\partial \dot{q}_i} - \frac{\partial K(\mathbf{q}(t), \dot{\mathbf{q}}(t))}{\partial q_i} + \frac{\partial P(\mathbf{q}(t))}{\partial q_i} = \tau_i(t), \quad 0 \leq t \leq t_f, \quad i = 1, \dots, n$$

erfüllt sein müssen.

Symbolisches „Ausrechnen“ ergibt das bereits bekannte System von Differentialgleichungen zweiter Ordnung (siehe Gleichung (5.11) auf Seite 123) für das inverse Dynamikmodell von der Form

$$M(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \tau. \quad (5.19)$$

### Bemerkungen:

- Mit der Bezeichnung

$$\Gamma_{i,jk} = \frac{1}{2} \left( \frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right) \in \mathbb{R}$$

gilt für die Elemente des Vektors der Zentrifugal- und Coriolisanteile (Angabe ohne Beweis)

$$C_i(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{j=1}^n \left( \sum_{k=1}^n \Gamma_{i,jk} \dot{q}_k \right) \dot{q}_j, \quad i = 1, \dots, n.$$

Die Größen  $\Gamma_{i,jk}$  heißen **Christoffel-Symbole**.

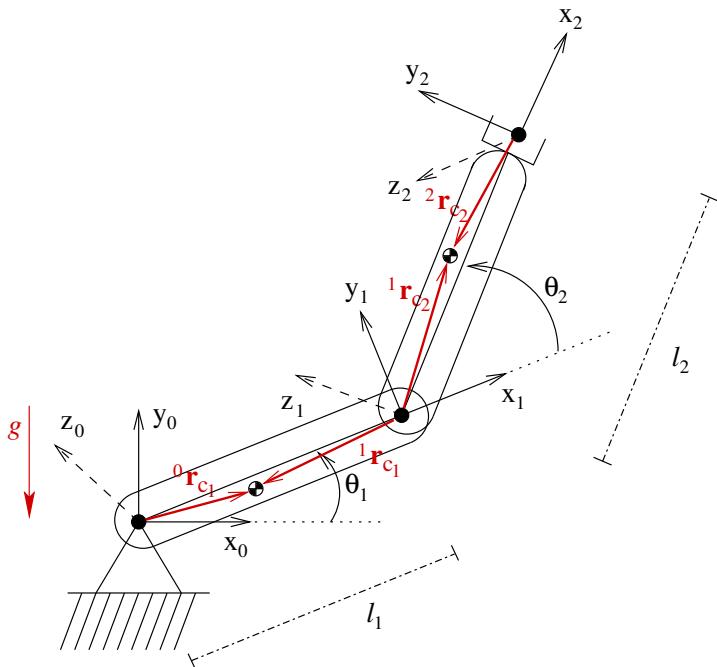
- Für kleine Geschwindigkeiten, d.h.  $\|\dot{\mathbf{q}}\| \approx 0$ , ist  $\|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\| \approx 0$ .
- Die einzelnen Elemente  $M_{ij}(\mathbf{q})$  von  $M(\mathbf{q})$  können je nach aktueller Konfiguration  $\mathbf{q}(t)$  sehr stark schwanken.
- Für Manipulatoren im Weltraum oder unter Wasser gilt:  $\mathbf{G}(\mathbf{q}) \approx 0$ .

<sup>6</sup>Siehe z.B. L.D. Landau, E.M. Lifschitz: Lehrbuch der Theoretischen Physik, Band I: Mechanik (Verlag Harri Deutsch, 14. korrigierte Auflage, 1997).

- Wenn man mit Hilfe der Lagrangeschen Formulierung die Bewegungsgleichungen für einen Manipulator „von Hand“ aufstellen möchte, dann bildet man

- (1) im ersten Schritt die Lagrangefunktion  $\mathcal{L}$  (5.17) durch Formulierung der kinetischen und potentiellen Energie.
- (2) Im zweiten Schritt wertet man die Gleichungen (5.18) aus (z.B. entweder von Hand oder mit Hilfe eines Formelmanipulationsprogramms wie Maple oder Mathematica).
- (3) Abschließend vereinfacht man die resultierenden Bewegungsgleichungen in Bezug auf mehrfach auftretende, gleiche arithmetische Ausdrücke und stellt die Bewegungsgleichungen in der Form (5.19) dar.

### 5.4.2 Beispiel: Lagrangesche Formulierung der Dynamik des ebenen 2-DOF-SCARA-Manipulators



Zunächst werden lineare und Winkel-Geschwindigkeit des ersten und zweiten Glieds bezüglich der jeweiligen Schwerpunkte benötigt. Gemäß der allgemeinen Darstellung der Jacobi-Matrix (Kapitel 4.3, Seite 72) bzw. aus der bereits ermittelten Jacobi-Matrix für das

$(n = 2)$ -te Glied (Kapitel 4.3.2, Seite 73) folgt:

$${}^0J_{c_1}(\theta_1, \theta_2) = \left[ \begin{array}{c|c} {}^0\mathbf{e}_{z_0} \times ({}^0\mathbf{r}_{c_1} - {}^0\mathbf{r}_0) & \mathbf{0} \\ \hline {}^0\mathbf{e}_{z_0} & \mathbf{0} \end{array} \right]$$

mit  ${}^0\mathbf{e}_{z_0} \times {}^0\mathbf{r}_{c_1} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} {}^0r_{c_1,x} \\ {}^0r_{c_1,y} \\ {}^0r_{c_1,z} \end{pmatrix} = \begin{pmatrix} -{}^0r_{c_1,y} \\ {}^0r_{c_1,x} \\ 0 \end{pmatrix}$  folgt

$${}^0J_{c_1}(\theta_1, \theta_2) = \left( \begin{array}{cc} -{}^0r_{c_1,y} & 0 \\ {}^0r_{c_1,x} & 0 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{array} \right), \quad \text{wobei } {}^0\mathbf{r}_{c_1} = {}^0\mathbf{r}_1 + {}^0R_1^{-1}\mathbf{r}_{c_1}.$$

$${}^0J_{c_2}(\theta_1, \theta_2) = \left[ \begin{array}{c|c} {}^0\mathbf{e}_{z_0} \times ({}^0\mathbf{r}_{c_2} - {}^0\mathbf{r}_0) & {}^0\mathbf{e}_{z_1} \times ({}^0\mathbf{r}_{c_2} - {}^0\mathbf{r}_1) \\ \hline {}^0\mathbf{e}_{z_0} & {}^0\mathbf{e}_{z_1} \end{array} \right]$$

mit  ${}^0\mathbf{e}_{z_0} \times {}^0\mathbf{r}_{c_2} = \begin{pmatrix} -{}^0r_{c_2,y} \\ {}^0r_{c_2,x} \\ 0 \end{pmatrix}$

und  ${}^0\mathbf{e}_{z_1} \times ({}^0\mathbf{r}_{c_2} - {}^0\mathbf{r}_1) = {}^0\mathbf{e}_{z_1} \times {}^0\mathbf{r}_{c_2} - {}^0\mathbf{e}_{z_1} \times {}^0\mathbf{r}_1 = \begin{pmatrix} -{}^0r_{c_2,y} \\ {}^0r_{c_2,x} \\ 0 \end{pmatrix} - \begin{pmatrix} -l_1 \sin \theta_1 \\ l_1 \cos \theta_1 \\ 0 \end{pmatrix}$

folgt

$${}^0J_{c_2}(\theta_1, \theta_2) = \left( \begin{array}{cc} -{}^0r_{c_2,y} & -{}^0r_{c_2,y} + l_1 \sin \theta_1 \\ {}^0r_{c_2,x} & {}^0r_{c_2,x} - l_1 \cos \theta_1 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{array} \right), \quad \text{wobei } {}^0\mathbf{r}_{c_2} = {}^0\mathbf{r}_2 + {}^0R_2^{-1}\mathbf{r}_{c_2}.$$

Damit lassen sich die gesuchten Geschwindigkeitsvektoren darstellen als

$${}^0\mathbf{v}_{c_1} = {}^0J_{c_1,v}(\mathbf{q}) \dot{\mathbf{q}} = \begin{pmatrix} -{}^0r_{c_1,y} \\ {}^0r_{c_1,x} \\ 0 \end{pmatrix} \dot{\theta}_1$$

$${}^0\omega_1 = {}^0J_{1,\omega}(\mathbf{q}) \dot{\mathbf{q}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{\theta}_1$$

$${}^0\mathbf{v}_{c_2} = {}^0J_{c_2,v}(\mathbf{q}) \dot{\mathbf{q}} = \begin{pmatrix} -{}^0r_{c_2,y} \\ {}^0r_{c_2,x} \\ 0 \end{pmatrix} \dot{\theta}_1 + \begin{pmatrix} -{}^0r_{c_2,y} + l_1 \sin \theta_1 \\ {}^0r_{c_2,x} - l_1 \cos \theta_1 \\ 0 \end{pmatrix} \dot{\theta}_2$$

$${}^0\omega_2 = {}^0J_{2,\omega}(\mathbf{q}) \dot{\mathbf{q}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} (\dot{\theta}_1 + \dot{\theta}_2)$$

Die Lagen der Schwerpunkte der beiden Glieder  ${}^1\mathbf{r}_{c_1}$  und  ${}^2\mathbf{r}_{c_2}$  sind bezüglich der jeweiligen gliedfesten Koordinatensysteme  $S_1$  bzw.  $S_2$  konstant, nicht jedoch im Basiskoordinatensystem  $S_0$ . Es gelten:

$$\begin{aligned} \begin{pmatrix} {}^0\mathbf{r}_{c_1} \\ 1 \end{pmatrix} &= {}^0T_1 \begin{pmatrix} {}^1\mathbf{r}_{c_1} \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & l_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & l_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^1r_{c_1,x} \\ {}^1r_{c_1,y} \\ {}^1r_{c_1,z} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} {}^1r_{c_1,x} \cos(\theta_1) - {}^1r_{c_1,y} \sin(\theta_1) + l_1 \cos(\theta_1) \\ {}^1r_{c_1,x} \sin(\theta_1) + {}^1r_{c_1,y} \cos(\theta_1) + l_1 \sin(\theta_1) \\ {}^1r_{c_1,z} \\ 1 \end{pmatrix} \\ \Rightarrow \frac{d}{dt} {}^0\mathbf{r}_{c_1} &= \begin{pmatrix} -{}^1r_{c_1,x} \sin(\theta_1) - {}^1r_{c_1,y} \cos(\theta_1) - l_1 \sin(\theta_1) \\ {}^1r_{c_1,x} \cos(\theta_1) - {}^1r_{c_1,y} \sin(\theta_1) + l_1 \cos(\theta_1) \\ 0 \end{pmatrix} \dot{\theta}_1 = \begin{pmatrix} {}^0\dot{r}_{c_1,x} \\ {}^0\dot{r}_{c_1,y} \\ 0 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \begin{pmatrix} {}^0\mathbf{r}_{c_2} \\ 1 \end{pmatrix} &= {}^0T_2 \begin{pmatrix} {}^2\mathbf{r}_{c_2} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^2r_{c_2,x} \\ {}^2r_{c_2,y} \\ {}^2r_{c_2,z} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} ({}^2r_{c_2,x} + l_2) \cos(\theta_1 + \theta_2) - {}^2r_{c_2,y} \sin(\theta_1 + \theta_2) + l_1 \cos(\theta_1) \\ ({}^2r_{c_2,x} + l_2) \sin(\theta_1 + \theta_2) + {}^2r_{c_2,y} \cos(\theta_1 + \theta_2) + l_1 \sin(\theta_1) \\ {}^2r_{c_2,z} \\ 1 \end{pmatrix} = \begin{pmatrix} {}^0r_{c_2,x} \\ {}^0r_{c_2,y} \\ {}^0r_{c_2,z} \\ 1 \end{pmatrix} \\ \Rightarrow \frac{d}{dt} {}^0\mathbf{r}_{c_2} &= \begin{pmatrix} -({}^2r_{c_2,x} + l_2) \sin(\theta_1 + \theta_2) - {}^2r_{c_2,y} \cos(\theta_1 + \theta_2) \\ ({}^2r_{c_2,x} + l_2) \cos(\theta_1 + \theta_2) - {}^2r_{c_2,y} \sin(\theta_1 + \theta_2) \\ 0 \end{pmatrix} (\dot{\theta}_1 + \dot{\theta}_2) + \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \\ 0 \end{pmatrix} l_1 \dot{\theta}_1 \\ &= \begin{pmatrix} {}^0\dot{r}_{c_2,x} \\ {}^0\dot{r}_{c_2,y} \\ 0 \end{pmatrix} \end{aligned}$$

Die kinetische Energie des ersten bzw. des zweiten Glieds ergibt sich gemäß Gleichung

(5.12) zu:

$$\begin{aligned}
 K_1 &= \frac{1}{2} m_1 \left( {}^0 \mathbf{v}_{c_1} \right)^T {}^0 \mathbf{v}_{c_1} + \frac{1}{2} {}^0 \omega_1^T {}^{c_1} I_1 {}^0 \omega_1 \\
 &= \frac{1}{2} m_1 \left( \left( {}^0 r_{c_1,x} \right)^2 + \left( {}^0 r_{c_1,y} \right)^2 \right) (\dot{\theta}_1)^2 + \frac{1}{2} {}^{c_1} I_{1,zz} (\dot{\theta}_1)^2 \\
 &= \frac{1}{2} \left[ m_1 \left( \left( {}^0 r_{c_1,x} \right)^2 + \left( {}^0 r_{c_1,y} \right)^2 \right) + {}^{c_1} I_{1,zz} \right] (\dot{\theta}_1)^2 \\
 K_2 &= \frac{1}{2} m_2 \left( {}^0 \mathbf{v}_{c_2} \right)^T {}^0 \mathbf{v}_{c_2} + \frac{1}{2} {}^0 \omega_2^T {}^{c_2} I_2 {}^0 \omega_2 \\
 &= \frac{1}{2} m_2 \left( \left[ -{}^0 r_{c_2,y} \dot{\theta}_1 + \left( -{}^0 r_{c_2,y} + l_1 \sin \theta_1 \right) \dot{\theta}_2 \right]^2 + \left[ {}^0 r_{c_2,x} \dot{\theta}_1 + \left( {}^0 r_{c_2,x} - l_1 \cos \theta_1 \right) \dot{\theta}_2 \right]^2 \right) \\
 &\quad + \frac{1}{2} {}^{c_2} I_{2,zz} (\dot{\theta}_1 + \dot{\theta}_2)^2
 \end{aligned}$$

Die potentielle Energie des ersten bzw. des zweiten Glieds ergibt sich gemäß Gleichung (5.15) zu:

$$\begin{aligned}
 P_1 &= -m_1 \mathbf{g}^T {}^0 \mathbf{r}_{c_1} + P_{\text{Ref},1}. \quad \text{Mit } \mathbf{g} = \begin{pmatrix} 0 \\ -g \\ 0 \end{pmatrix} \text{ folgt} \\
 P_1 &= m_1 g {}^0 r_{c_1,y} + P_{\text{Ref},1}.
 \end{aligned}$$

$$\text{Analog gilt } P_2 = -m_2 \mathbf{g}^T {}^0 \mathbf{r}_{c_2} + P_{\text{Ref},2} = m_2 g {}^0 r_{c_2,y} + P_{\text{Ref},2}.$$

Dabei bezeichnet  $g$  die (höhenabhängige) Gravitationskonstante (z.B.  $g = 9.81 \text{ m/s}^2$ ).

Aus den kinetischen und potentiellen Energien der beiden Roboterglieder erhält man die Lagrangefunktion als

$$\mathcal{L} = K_1 + K_2 - P_1 - P_2.$$

Damit können aus Gleichung (5.18) die Bewegungsgleichungen gemäß

$$\tau_i = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \frac{d}{dt} \frac{\partial (K_1 + K_2)}{\partial \dot{\theta}_i} - \frac{\partial (K_1 + K_2)}{\partial \theta_i} + \frac{\partial (P_1 + P_2)}{\partial \theta_i}$$

ermittelt werden.

Die einzelnen Berechnungsschritte sollen im Folgenden exemplarisch für  $\tau_2$  durchgeführt werden. Dabei ist zu erwarten, dass dieselbe analytische Formulierung der inversen Dynamik herauskommt, wie sie durch Anwenden und symbolisches Ausrechnen der Newton-Euler-Rekursionen für denselben zweigelenkigen Manipulator in Kapitel 5.3.3 (Seite 121) ermittelt wurde.

$$\begin{aligned}
 \frac{\partial P_1}{\partial \theta_2} &= 0 \\
 \frac{\partial P_2}{\partial \theta_2} &= m_2 g \frac{\partial {}^0 r_{c_2,y}}{\partial \theta_2} \\
 &= m_2 g \left( {}^2 r_{c_2,x} \cos(\theta_1 + \theta_2) - {}^2 r_{c_2,y} \sin(\theta_1 + \theta_2) + l_2 \cos(\theta_1 + \theta_2) \right) \\
 &= m_2 g \left( (l_2 + {}^2 r_{c_2,x}) \cos(\theta_1 + \theta_2) - {}^2 r_{c_2,y} \sin(\theta_1 + \theta_2) \right) = G_2(\theta)
 \end{aligned}$$

Damit stimmt der hier ermittelte Gravitationsanteil  $G_2(\theta)$  mit dem mittels der Newton-Euler-Rekursionen im Kapitel 5.3.3 (Seite 121) ermittelten Gravitationsanteil überein.

$$\begin{aligned}
 \frac{\partial K_1}{\partial \theta_2} &= 0 \\
 \frac{\partial K_2}{\partial \theta_2} &= m_2 \left( \left[ -{}^0r_{c_2,y}\dot{\theta}_1 + (-{}^0r_{c_2,y} + l_1 \sin \theta_1)\dot{\theta}_2 \right] \left( -\frac{\partial {}^0r_{c_2,y}}{\partial \theta_2} \right) (\dot{\theta}_1 + \dot{\theta}_2) \right. \\
 &\quad \left. + \left[ {}^0r_{c_2,x}\dot{\theta}_1 + ({}^0r_{c_2,x} - l_1 \cos \theta_1)\dot{\theta}_2 \right] \left( \frac{\partial {}^0r_{c_2,x}}{\partial \theta_2} \right) (\dot{\theta}_1 + \dot{\theta}_2) \right) \\
 \frac{\partial K_1}{\partial \dot{\theta}_2} &= 0 \quad \Rightarrow \quad \frac{d}{dt} \frac{\partial K_1}{\partial \dot{\theta}_2} = 0 \\
 \frac{\partial K_2}{\partial \dot{\theta}_2} &= m_2 \left[ -{}^0r_{c_2,y}\dot{\theta}_1 + (-{}^0r_{c_2,y} + l_1 \sin \theta_1)\dot{\theta}_2 \right] (-{}^0r_{c_2,y} + l_1 \sin \theta_1) \\
 &\quad + m_2 \left[ {}^0r_{c_2,x}\dot{\theta}_1 + ({}^0r_{c_2,x} - l_1 \cos \theta_1)\dot{\theta}_2 \right] ({}^0r_{c_2,x} - l_1 \cos \theta_1) \\
 &\quad + {}^{c_2}I_{2,zz}(\dot{\theta}_1 + \dot{\theta}_2) \\
 \Rightarrow \frac{d}{dt} \frac{\partial K_2}{\partial \dot{\theta}_2} &= m_2 \left[ -{}^0\dot{r}_{c_2,y}\dot{\theta}_1 - {}^0r_{c_2,y}\ddot{\theta}_1 + (-{}^0\dot{r}_{c_2,y} + l_1 \cos \theta_1 \dot{\theta}_1)\dot{\theta}_2 + (-{}^0r_{c_2,y} + l_1 \sin \theta_1)\ddot{\theta}_2 \right] \\
 &\quad \cdot (-{}^0r_{c_2,y} + l_1 \sin \theta_1) \\
 &\quad + m_2 \left[ -{}^0r_{c_2,y}\dot{\theta}_1 + (-{}^0r_{c_2,y} + l_1 \sin \theta_1)\dot{\theta}_2 \right] (-{}^0\dot{r}_{c_2,y} + l_1 \cos \theta_1 \dot{\theta}_1) \\
 &\quad + m_2 \left[ {}^0\dot{r}_{c_2,x}\dot{\theta}_1 + {}^0r_{c_2,x}\dot{\theta}_1 + ({}^0\dot{r}_{c_2,x} + l_1 \sin \theta_1 \dot{\theta}_1)\dot{\theta}_2 + ({}^0r_{c_2,x} - l_1 \cos \theta_1)\ddot{\theta}_2 \right] \\
 &\quad \cdot ({}^0r_{c_2,x} - l_1 \cos \theta_1) \\
 &\quad + m_2 \left[ {}^0r_{c_2,x}\dot{\theta}_1 + ({}^0r_{c_2,x} - l_1 \cos \theta_1)\dot{\theta}_2 \right] ({}^0\dot{r}_{c_2,x} + l_1 \sin \theta_1 \dot{\theta}_1) \\
 &\quad + {}^{c_2}I_{2,zz}(\ddot{\theta}_1 + \ddot{\theta}_2)
 \end{aligned}$$

Ausmultiplizieren und Umsortieren liefert

$$\begin{aligned} \frac{d}{dt} \frac{\partial K_2}{\partial \dot{\theta}_2} &= m_2 \left[ \left( -{}^0\dot{r}_{c_2,y}\dot{\theta}_1 - {}^0r_{c_2,y}\ddot{\theta}_1 \right) \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) \right. \\ &\quad + \left( -{}^0\dot{r}_{c_2,y} + l_1 \cos \theta_1 \dot{\theta}_1 \right) \dot{\theta}_2 \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) \\ &\quad + \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) \ddot{\theta}_2 \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) + \\ &\quad - {}^0r_{c_2,y}\dot{\theta}_1 \left( -{}^0\dot{r}_{c_2,y} + l_1 \cos \theta_1 \dot{\theta}_1 \right) + \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) \dot{\theta}_2 \left( -{}^0\dot{r}_{c_2,y} + l_1 \cos \theta_1 \dot{\theta}_1 \right) \\ &\quad + \left( {}^0\dot{r}_{c_2,x}\dot{\theta}_1 + {}^0r_{c_2,x}\ddot{\theta}_1 \right) \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) + \left( {}^0\dot{r}_{c_2,x} + l_1 \sin \theta_1 \dot{\theta}_1 \right) \dot{\theta}_2 \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) \\ &\quad + \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) \ddot{\theta}_2 \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) \\ &\quad \left. + {}^0r_{c_2,x}\dot{\theta}_1 \left( {}^0\dot{r}_{c_2,x} + l_1 \sin \theta_1 \dot{\theta}_1 \right) + \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) \dot{\theta}_2 \left( {}^0\dot{r}_{c_2,x} + l_1 \sin \theta_1 \dot{\theta}_1 \right) \right] \\ &\quad + {}^{c_2}I_{2,zz}(\ddot{\theta}_1 + \ddot{\theta}_2) \end{aligned}$$

Zur besseren Übersichtlichkeit betrachten wir die Terme mit Faktoren von  $\ddot{\theta}_j$  getrennt von denen mit  $\dot{\theta}_k$ :

$$\begin{aligned} \frac{d}{dt} \frac{\partial K_2}{\partial \dot{\theta}_2} &=: A(\theta, \ddot{\theta}) + B(\theta, \dot{\theta}) \\ \text{mit } A(\theta, \ddot{\theta}) &= m_2 \left[ -{}^0r_{c_2,y}\ddot{\theta}_1 \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) + \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) \ddot{\theta}_2 \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right) \right. \\ &\quad \left. + {}^0r_{c_2,x}\ddot{\theta}_1 \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) + \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) \ddot{\theta}_2 \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) \right] \\ &\quad + {}^{c_2}I_{2,zz}(\ddot{\theta}_1 + \ddot{\theta}_2) \end{aligned}$$

Ordnen nach  $\ddot{\theta}_1$  und  $\ddot{\theta}_2$  liefert:

$$\begin{aligned} A(\theta, \ddot{\theta}) &= \underbrace{\left\{ m_2 \left[ {}^0r_{c_2,y} \left( {}^0r_{c_2,y} - l_1 \sin \theta_1 \right) + {}^0r_{c_2,x} \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right) \right] + {}^{c_2}I_{2,zz} \right\}}_{:= A_1(\theta)} \ddot{\theta}_1 \\ &\quad + \underbrace{\left\{ m_2 \left[ \left( -{}^0r_{c_2,y} + l_1 \sin \theta_1 \right)^2 + \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right)^2 \right] + {}^{c_2}I_{2,zz} \right\}}_{:= A_2(\theta)} \ddot{\theta}_2 \end{aligned}$$

$$\begin{aligned} \text{mit } \left( {}^0r_{c_2,y} - l_1 \sin \theta_1 \right)^2 + \left( {}^0r_{c_2,x} - l_1 \cos \theta_1 \right)^2 &= \\ &= \left[ \left( {}^2r_{c_2,x} + l_2 \right) \sin(\theta_1 + \theta_2) + {}^2r_{c_2,y} \cos(\theta_1 + \theta_2) + l_1 \sin(\theta_1) - l_1 \sin(\theta_1) \right]^2 \\ &\quad + \left[ \left( {}^2r_{c_2,x} + l_2 \right) \cos(\theta_1 + \theta_2) - {}^2r_{c_2,y} \sin(\theta_1 + \theta_2) + l_1 \cos(\theta_1) - l_1 \cos(\theta_1) \right]^2 \\ &= \left( {}^2r_{c_2,x} + l_2 \right)^2 \sin^2(\theta_1 + \theta_2) + {}^2r_{c_2,y}^2 \cos^2(\theta_1 + \theta_2) \\ &\quad + 2 \left( {}^2r_{c_2,x} + l_2 \right) \sin(\theta_1 + \theta_2) {}^2r_{c_2,y} \cos(\theta_1 + \theta_2) \\ &\quad + \left( {}^2r_{c_2,x} + l_2 \right)^2 \cos^2(\theta_1 + \theta_2) + {}^2r_{c_2,y}^2 \sin^2(\theta_1 + \theta_2) \\ &\quad + 2 \left( {}^2r_{c_2,x} + l_2 \right) \cos(\theta_1 + \theta_2) \left( -{}^2r_{c_2,y} \right) \sin(\theta_1 + \theta_2) \\ &= \left( {}^2r_{c_2,x} + l_2 \right)^2 + {}^2r_{c_2,y}^2 \quad \text{folgt} \end{aligned}$$

$$A_2(\theta) = m_2 \left[ \left( {}^2r_{c2,x} + l_2 \right)^2 + \left( {}^2r_{c2,y} \right)^2 \right] + {}^c I_{2,zz} .$$

Mit

$$\begin{aligned} & {}^0r_{c2,y} \left( {}^0r_{c2,y} - l_1 \sin \theta_1 \right) + {}^0r_{c2,x} \left( {}^0r_{c2,x} - l_1 \cos \theta_1 \right) \\ &= \left( \left( {}^2r_{c2,x} + l_2 \right) \sin(\theta_1 + \theta_2) + {}^2r_{c2,y} \cos(\theta_1 + \theta_2) + l_1 \sin(\theta_1) \right) \\ &\quad \cdot \left( \left( {}^2r_{c2,x} + l_2 \right) \sin(\theta_1 + \theta_2) + {}^2r_{c2,y} \cos(\theta_1 + \theta_2) \right) \\ &+ \left( \left( {}^2r_{c2,x} + l_2 \right) \cos(\theta_1 + \theta_2) - {}^2r_{c2,y} \sin(\theta_1 + \theta_2) + l_1 \cos(\theta_1) \right) \\ &\quad \cdot \left( \left( {}^2r_{c2,x} + l_2 \right) \cos(\theta_1 + \theta_2) - {}^2r_{c2,y} \sin(\theta_1 + \theta_2) \right) \\ &= \left( {}^2r_{c2,x} + l_2 \right)^2 \sin^2(\theta_1 + \theta_2) + {}^2r_{c2,y}^2 \cos^2(\theta_1 + \theta_2) \\ &\quad + 2 \left( {}^2r_{c2,x} + l_2 \right) \sin(\theta_1 + \theta_2) {}^2r_{c2,y} \cos(\theta_1 + \theta_2) \\ &\quad + l_1 \sin(\theta_1) \left( \left( {}^2r_{c2,x} + l_2 \right) \sin(\theta_1 + \theta_2) + {}^2r_{c2,y} \cos(\theta_1 + \theta_2) \right) \\ &+ \left( {}^2r_{c2,x} + l_2 \right)^2 \cos^2(\theta_1 + \theta_2) + {}^2r_{c2,y}^2 \sin^2(\theta_1 + \theta_2) \\ &\quad - 2 \left( {}^2r_{c2,x} + l_2 \right) \cos(\theta_1 + \theta_2) {}^2r_{c2,y} \sin(\theta_1 + \theta_2) \\ &\quad + l_1 \cos(\theta_1) \left( \left( {}^2r_{c2,x} + l_2 \right) \cos(\theta_1 + \theta_2) - {}^2r_{c2,y} \sin(\theta_1 + \theta_2) \right) \\ &= \left( {}^2r_{c2,x} + l_2 \right)^2 + \left( {}^2r_{c2,y} \right)^2 \\ &\quad + l_1 \left( {}^2r_{c2,x} + l_2 \right) \underbrace{\left( \sin(\theta_1) \sin(\theta_1 + \theta_2) + \cos(\theta_1) \cos(\theta_1 + \theta_2) \right)}_{= \cos \theta_2} \\ &\quad + l_1 {}^2r_{c2,y} \underbrace{\left( \sin(\theta_1) \cos(\theta_1 + \theta_2) - \cos(\theta_1) \sin(\theta_1 + \theta_2) \right)}_{= -\sin \theta_2} \quad \text{folgt weiter} \\ A_1(\theta) &= m_2 \left[ \left( {}^2r_{c2,x} + l_2 \right)^2 + \left( {}^2r_{c2,y} \right)^2 + l_1 l_2 \cos \theta_2 + l_1 \left( {}^2r_{c2,x} \cos \theta_2 - {}^2r_{c2,y} \sin \theta_2 \right) \right] + {}^c I_{2,zz} . \end{aligned}$$

Ein Vergleich mit den in Kapitel 5.3.3 (Seite 121) durch Auswerten der Newton-Euler-Rekursionen erhaltenen Koeffizienten ergibt:

$$A_1(\theta) = M_{2,1}(\theta), \quad A_2(\theta) = M_{2,2}(\theta) .$$

Nach den bisherigen Überlegungen ergibt sich die inverse Dynamik für  $\tau_2$  in der Form:

$$\begin{aligned} \tau_2 &= \frac{d}{dt} \frac{\partial(K_1 + K_2)}{\partial \dot{\theta}_2} - \frac{\partial(K_1 + K_2)}{\partial \theta_2} + \frac{\partial(P_1 + P_2)}{\partial \theta_2} \\ &= \underbrace{\frac{d}{dt} \frac{\partial K_1}{\partial \dot{\theta}_2}}_{= 0} + \underbrace{\frac{d}{dt} \frac{\partial K_2}{\partial \dot{\theta}_2}}_{= 0} - \underbrace{\frac{\partial K_1}{\partial \theta_2}}_{= 0} - \underbrace{\frac{\partial K_2}{\partial \theta_2}}_{= 0} + \underbrace{\frac{\partial P_1}{\partial \theta_2}}_{= 0} + \underbrace{\frac{\partial P_2}{\partial \theta_2}}_{= G_2(\theta)} \end{aligned}$$

Für den expliziten Nachweis, dass für den zweigelenkigen Manipulator sich nach Newton-Euler und Lagrange dieselbe inverse Dynamik ergibt, ist (außer der Herleitung für  $\tau_1$ ) für

$\tau_2$  noch nachzuweisen, dass

$$C_2(\theta, \dot{\theta}) = B(\theta, \dot{\theta}) - \frac{\partial K_2(\theta, \dot{\theta})}{\partial \theta_2}.$$

Da die explizite Durchführung dieses Nachweises an dieser Stelle keine grundlegend neuen Erkenntnisse ergeben würde, wird die Durchführung dem Leser bzw. der Leserin zur Übung empfohlen.

## 5.5 Numerische Aspekte

### 5.5.1 Berechnungseffizienz

Wie die Berechnungen der Dynamik nach Newton-Euler und Lagrange gezeigt haben, sind die Bewegungsgleichungen INV DYN oder DIR DYN für typische Manipulatoren mit  $n = 3$  oder mehr Achsen sehr komplex. Da Berechnungen der Dynamik für Zwecke der Regelung oder Trajektorienplanung online benötigt werden, lohnt sich ein Vergleich der benötigten arithmetischen Operationen (nach [Craig,1989], Section 6.13, Seiten 216-219):

Newton-Euler (Standardversion):	ca. $126n - 99$ Multiplikationen, ca. $106n - 92$ Additionen,
Lagrange (naive, direkte Berech- nung):	ca. $32n^4 + 86n^3 + 171n^2 + 53n - 128$ Multiplika- tionen, ca. $25n^4 + 66n^3 + 129n^2 + 42n - 96$ Additionen.

**Ergebnis:** Für  $n = 6$  ist die Standard-Methode von Newton-Euler in der Berechnung der Manipulatordynamik um etwa den Faktor 100 schneller als die (naive Version der) Methode von Lagrange.

**Aber:** Aber die Methode von Lagrange, die durch die mehrfachen Ableitungen der kinetischen Energie sehr schnell sehr komplexe Formeln erzeugt, kann prinzipiell auch schnelleren Code als die Standard-Methode von Newton-Euler produzieren, wenn die resultierenden Formeln für die Bewegungsgleichungen effizient genug zusammengefasst werden. Das muss aber normalerweise mühsam und zeitaufwendig „von Hand“ optimiert und implementiert werden (wie im Beispiel in Kapitel 5.4.2).

Umgekehrt kann die Standard-Version der Newton-Euler-Rekursionen noch bezüglich arithmetischer Operationen optimiert werden, indem mehrfach auftretende Zwischenergebnisse in den Teilschritten (siehe das Beispiel in Kapitel 5.3.3) nur einmal berechnet und zwischengespeichert werden.

### 5.5.2 Modularität der Dynamikmodellierung

Ein weiteres Bewertungskriterium ist der jeweils notwendige Zusatzaufwand beim Austausch von Gelenktypen oder beim Einfügen oder Herausnehmen von Gelenken oder Gliedern.

- Bei der Methode von Newton-Euler können solche Änderungen in der kinematischen Struktur völlig unproblematisch durch Einfügen eines zusätzlichen oder Herausnehmen eines bestehenden Iterationsschrittes bei (NE.1) – (NE.3) berücksichtigt werden.
- Bei der Methode von Lagrange muss die Herleitung der Roboterdynamik von Grund auf neu durchgeführt werden. Hinzu kommt, dass eine mittels der Methode von Lagrange hergeleitete Roboterdynamik in der Regel „von Hand“ nachbearbeitet wird, um die  $\mathcal{O}(n^4)$  Berechnungskomplexität zu vermeiden und möglichst  $\mathcal{O}(n)$  zu erreichen. Diese Nachbearbeitung muss für die modifizierte Kinematikstruktur ebenfalls komplett neu durchgeführt werden.

**Fazit:** Die Herleitung eines bezüglich arithmetischer Operationen besonders effizienten, expliziten Dynamikmodells erfolgt mit Lagrange und in der Regel zeitaufwendiger Nachbearbeitung. Dies wird nur dann eingesetzt, wenn die kinematische Struktur des Roboters

sich nicht ändert und Rechenzeit zur Dynamikauswertung besonders kritisch ist (z.B. bei der Bahn- und Trajektorienregelung von Industrierobotern mit dynamikmodellbasierten Verfahren).

### 5.5.3 Simulation dynamischer Roboterbewegungen

Für einen gegebenen Verlauf der Steuerungen  $\tau(t)$ ,  $0 \leq t \leq t_f$ , und bei gegebenen Anfangszuständen  $\mathbf{q}(0)$  und  $\dot{\mathbf{q}}(0)$  kann der Verlauf der Gelenkvariablen  $\mathbf{q}(t)$ ,  $0 \leq t \leq t_f$ , aus der inversen Dynamik

$$M(\mathbf{q}(t)) \cdot \ddot{\mathbf{q}}(t) = \tau(t) - \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) - \mathbf{G}(\mathbf{q}(t))$$

mit numerischen Verfahren berechnet werden.

Einfache Beispiele für solche Verfahren zur numerischen Integration von gewöhnlichen Differentialgleichungen sind das Euler-Verfahren oder die Runge-Kutta-Verfahren<sup>7</sup>. Diese in Programmhbibliotheken verfügbaren Verfahren setzen in der Regel die Formulierung der Dynamik in der Standardform eines Systems von Differentialgleichungen erster Ordnung voraus, d.h.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t), \quad \text{mit gegebenem Anfangswert } \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^{n_x}.$$

Das inverse Dynamikmodell kann mit einer einfachen Transformation auf diese Form gebracht werden:

$$\begin{aligned} \mathbf{x} &:= \begin{pmatrix} \mathbf{x}_I \\ \mathbf{x}_{II} \end{pmatrix} := \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix}, \quad \mathbf{x}(0) = \begin{pmatrix} \mathbf{q}(0) \\ \dot{\mathbf{q}}(0) \end{pmatrix} \\ \Rightarrow \dot{\mathbf{x}} &= \begin{pmatrix} \dot{\mathbf{x}}_I \\ \dot{\mathbf{x}}_{II} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ M^{-1}(\mathbf{q}) \cdot [\tau - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q})] \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ M^{-1}(\mathbf{x}_I) \cdot [\tau - \mathbf{C}(\mathbf{x}_I, \mathbf{x}_{II}) - \mathbf{G}(\mathbf{x}_I)] \end{pmatrix}. \end{aligned}$$

Dabei wird die hier formal hingeschriebene Inverse der Massenmatrix nie explizit als inverse Matrix berechnet. Vielmehr kann bei jeder Auswertung von  $\mathbf{f}(\mathbf{x}, t)$  der gesuchte Vektor  $\mathbf{x}_{II}$  als numerische Lösung des linearen Gleichungssystems  $M(\mathbf{x}_I) \cdot \mathbf{x}_{II} = \tau - [...]$  mit einer Matrixzerlegung wie in Kapitel 2.1 (Seite 13) beschrieben berechnet werden, sofern nicht zur Berechnung von DIR DYN auf ein Verfahren zurückgegriffen wird, das gar keine explizite Berechnung der Matrix  $M(\mathbf{q})$  benötigt (siehe Abschnitt 5.6).

**Bemerkung:** Es existieren Mehrkörperdynamikalgorithmen, die die Gleichungen für den Ausdruck  $M^{-1}(\mathbf{q}) \cdot [\tau - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q})]$  explizit (aber ohne explizite Bildung der inversen Massenmatrix) und mit nur  $\mathcal{O}(n)$  arithmetischen Operationen aufstellen können<sup>8</sup>. Weitere Verfahren werden im nächsten Abschnitt näher erläutert.

<sup>7</sup>siehe U. Ascher, L.R. Petzold: *Computer methods for ordinary differential equations and differential-algebraic equations* (SIAM, 1998) oder R.W. Freund, R.H.W. Hoppe: *Stoer/Bulirsch: Numerische Mathematik 2* (Springer-Verlag, 6. Aufl. 2012).

<sup>8</sup>Z.B. H. Brandl, R. Johann, M. Otter: *A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix*, In: Proc. IFAC/IFIP/IMACS International Symposium on Theory of Robots, Wien, 3.-5. Dezember 1986.

## 5.6 Rekursive Verfahren zur Berechnung der Vorwärtsdynamik DIR DYN<sup>9</sup>

Zur Simulation dynamischer Roboterbewegungen (wie im vorigen Kapitel beschrieben) wird die Vorwärtsdynamik des zu simulierenden Roboters benötigt. Unter der Berechnung der Vorwärtsdynamik wird dabei die Berechnung der Gelenkbeschleunigungen  $\ddot{\mathbf{q}}$  in Abhängigkeit von den in den Gelenken wirkenden Kräften bzw. Momenten  $\tau$  sowie den Gelenkstellungen  $\mathbf{q}$  und -geschwindigkeiten  $\dot{\mathbf{q}}$  verstanden. Zu diesem Zweck existieren verschiedene Verfahren, die im Folgenden unterschieden werden nach Verfahren, die die Massenmatrix  $M$  explizit aufstellen und im Anschluss daran das entstehende lineare Gleichungssystem lösen, und nach Verfahren, die das Aufstellen der Massenmatrix vermeiden.

### 5.6.1 Verfahren mit expliziter Berechnung der Massenmatrix $M$

Bereits 1982 wurden von Walker und Orin verschiedene Methoden zum Aufstellen der Massenmatrix vorgestellt<sup>10</sup>. Im Folgenden werden diese drei Verfahren mit der in diesem Skript verwendeten Notation beschrieben.

#### 5.6.1.1 Verfahren 1: Berechnung von $M$ durch wiederholte Auswertung des Newton-Euler-Verfahrens

Wie bereits gezeigt wurde, ist der allgemeine Zusammenhang zwischen den Gelenkkräften bzw. -momenten auf der einen und den Gelenkpositionen, -geschwindigkeiten und -beschleunigungen auf der anderen Seite durch die Gleichung

$$\tau = M(\mathbf{q}(t)) \cdot \ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{G}(\mathbf{q}(t))$$

gegeben. Zur Berechnung der inversen Dynamik kann dabei z. B. das rekursive Newton-Euler-Verfahren (RNE) verwendet werden. Im Folgenden wird nun gezeigt, wie sich durch mehrfache Anwendung des RNE die Massenmatrix berechnen lässt. Für diese Diskussion soll

$$\tau = RNE(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t), \mathbf{g})$$

die Berechnung der Gelenkkkräfte/Momente durch das RNE bezeichnen, wobei  $\mathbf{g}$  der Gravitationsvektor ist. Damit ergibt sich zunächst:

$$M(\mathbf{q}(t)) \cdot \ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{G}(\mathbf{q}(t), \mathbf{g}) = RNE(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t), \mathbf{g}).$$

Die aus Corioliskräften und Gravitation resultierenden Kräfte/Momente  $\mathbf{b}$  in den Gelenken lassen sich nach Null setzen von  $\ddot{\mathbf{q}}$  durch

$$\mathbf{b} = \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{G}(\mathbf{q}(t), \mathbf{g}) = M(\mathbf{q}(t)) \cdot \mathbf{0} + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{G}(\mathbf{q}(t), \mathbf{g}) = RNE(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{0}, \mathbf{g})$$

berechnen. Analog lassen sich die allein aus der Gelenkbeschleunigung  $\ddot{\mathbf{q}}$  (d. h. ohne Corioliskräfte oder Gravitation) resultierenden Kräfte/Momente durch Null setzen der Gelenkgeschwindigkeit  $\dot{\mathbf{q}}$  und der Gravitationskonstanten  $\mathbf{g}$  mit

$$M(\mathbf{q}(t)) \cdot \ddot{\mathbf{q}}(t) = RNE(\mathbf{q}(t), \mathbf{0}, \ddot{\mathbf{q}}(t), \mathbf{0})$$

---

<sup>9</sup>Die erste Version dieses Kapitels beruht auf einer Ausarbeitung von Martin Friedmann.

<sup>10</sup>M. W. Walker, D. E. Orin: *Efficient Dynamics Computer Simulation of Robotics Mechanisms*, Journal of Dynamic Systems, Measurement and Control, 104:205-212, 1982.

berechnen. Damit wird eine Berechnung einzelner Spalten  $\{M\}_i$  von  $M$  durch

$$\{M(\mathbf{q}(t))\}_i = M(\mathbf{q}(t)) \cdot \mathbf{e}_i = RNE(\mathbf{q}(t), \mathbf{0}, \mathbf{e}_i, \mathbf{0})$$

ermöglicht.

### 5.6.1.2 Verfahren 2: Ausnutzung der Symmetrie von $M$

Im Verfahren 1 wird die Massenmatrix spaltenweise aufgebaut, wobei jedes Element der Matrix berechnet wird. Durch Ausnutzung der Symmetrie der Massenmatrix lässt sich der Berechnungsaufwand erheblich verringern. Dazu wird im Verfahren 2 das RNE für die einzelnen Spalten nur teilweise ausgeführt. Die Spalten werden dabei beginnend mit der letzten Spalte durchlaufen, wobei die Anzahl der berechneten Zeilen pro Spalte um eine zunimmt. Nicht-berechnete Elemente werden gespiegelt. Es ergibt sich folgende Rechenvorschrift:

- 1 Initialisierung: Berechne (NE.1) für die aktuellen Werte von  $\mathbf{q}$  und  $\dot{\mathbf{q}}$  sowie für  $\ddot{\mathbf{q}} = \mathbf{0}$ .
- 2 Berechnung der Spalten der unteren linken Dreiecksmatrix von  $M$ : für  $j = n, \dots, 1$ :
  - setze die Gelenkbeschleunigung  $\ddot{\mathbf{q}} = \mathbf{e}_j$
  - berechne (NE.1) und (NE.2) für die Glieder  $i = j, \dots, n$
  - berechne (NE.3) für die Glieder  $i = n, \dots, j$
  - für  $i = j, \dots, n$ :
    - setze die Elemente der  $j$ -ten Spalte  $m_{ij} = \tau_i$
    - spiegeln wegen Symmetrie  $m_{ji} = \tau_i$

### 5.6.1.3 Verfahren 3: Aggregation von Teilmanipulatoren (CRBA)

Das dritte Verfahren berechnet ebenso wie das zweite nur einen Teil der Matrix und ergänzt die verbleibenden Elemente unter Ausnutzung der Symmetrie. Allerdings basiert dieses Verfahren nicht mehr auf mehrfachem Auswerten des RNE. Stattdessen wird eine Folge von aggregierten Teilmanipulatoren konstruiert, indem für  $i = n, \dots, 1$  jeweils die Glieder  $i, \dots, n$  des Manipulators in der aktuellen Konfiguration als starr verbunden angenommen werden. Dabei werden die gemeinsame Masse  $\tilde{m}_i$ , der gemeinsame Schwerpunkt  ${}^i\mathbf{r}_{\tilde{c}_i}$  sowie der gemeinsame Trägheitstensor  ${}^i\tilde{I}_i$  iterativ von außen nach innen berechnet.

Mit Hilfe dieser aggregierten Teilmanipulatoren lassen sich nun die Spalten der Massenmatrix bestimmen. Für Spalte  $i$  wird dabei angenommen, dass genau das Gelenk  $i$  mit  $\ddot{q}_i = 1$  beschleunigt wird, während alle anderen Gelenke ruhen ( $\ddot{q}_{j \neq i} = 0$ ). Weiterhin soll die Basis des Manipulators unbewegt sein und die Geschwindigkeit der Gelenke soll Null sein ( $\dot{\mathbf{q}} = \mathbf{v}_0 = \dot{\mathbf{v}}_0 = \omega_0 = \dot{\omega}_0 = \mathbf{0}$ ). Aufgrund dieser Annahme sind die Geschwindigkeiten und Beschleunigungen der Glieder  $j = 1, \dots, i - 1$  Null ( $\mathbf{v}_j = \dot{\mathbf{v}}_j = \omega_j = \dot{\omega}_j = \mathbf{0}$ ), so dass an den Massenschwerpunkten dieser Glieder keine Kräfte wirken müssen ( $\mathbf{F}_j = \mathbf{N}_j = \mathbf{0}$ ). Hieraus ergeben sich starke Vereinfachungen der nachfolgenden Berechnungen.

Die in Gelenk  $i$  benötigte Kraft (bzw. das benötigte Drehmoment)  $\tau_i$  kann nun durch Anwenden der Gelenkbeschleunigung auf den  $i$ -ten aggregierten Teilmanipulator ermittelt werden. Die Kräfte/Momente in den davor liegenden Gelenken ergeben sich durch die

entstehenden Reaktionskräfte über die kinematische Kette. Die berechneten Kräfte/Momente  $\tau_i, \dots, \tau_1$  sind die Elemente der Spalte  $i$  der Massenmatrix und können aufgrund der Symmetrie auch in Zeile  $i$  eingetragen werden.

Damit ergibt sich folgendes Rechenverfahren für die Massenmatrix:

- Berechnen der Massen, Schwerpunkte und Trägheitstensoren der Teilmanipulatoren
  - letztes Glied des Manipulators

$$\begin{aligned}\tilde{m}_n &= m_n \\ {}^n\mathbf{r}_{\tilde{c}_n} &= {}^n\mathbf{r}_{c_n} \\ {}^{\tilde{c}_n}\tilde{I}_n &= {}^{c_n}I_n\end{aligned}$$

- vorherige Glieder für  $i = n - 1, \dots, 1$

$$\begin{aligned}\tilde{m}_i &= m_i + \tilde{m}_{i+1} \\ {}^i\mathbf{r}_{\tilde{c}_{i+1}} &= {}^i\mathbf{r}_{i+1} + {}^iR_{i+1} \cdot {}^{i+1}\mathbf{r}_{\tilde{c}_{i+1}} \\ {}^i\mathbf{r}_{\tilde{c}_i} &= \frac{1}{\tilde{m}_i} \cdot (m_i \cdot {}^i\mathbf{r}_{c_i} + \tilde{m}_{i+1} \cdot {}^i\mathbf{r}_{\tilde{c}_{i+1}}) \\ {}^{\tilde{c}_i}\tilde{I}_i &= {}^{c_i}I_i + m_i \cdot \left( [({}^i\mathbf{r}_{c_i} - {}^i\mathbf{r}_{\tilde{c}_i})^T \cdot ({}^i\mathbf{r}_{c_i} - {}^i\mathbf{r}_{\tilde{c}_i})] \cdot E - ({}^i\mathbf{r}_{c_i} - {}^i\mathbf{r}_{\tilde{c}_i}) \cdot ({}^i\mathbf{r}_{c_i} - {}^i\mathbf{r}_{\tilde{c}_i})^T \right) \\ {}^{\tilde{c}_i}\tilde{I}_{i+1} &= {}^iR_{i+1} \cdot {}^{\tilde{c}_{i+1}}\tilde{I}_{i+1} \cdot {}^{i+1}R_i \\ &\quad + \tilde{m}_{i+1} \cdot \left( [({}^i\mathbf{r}_{\tilde{c}_{i+1}} - {}^i\mathbf{r}_{\tilde{c}_i})^T \cdot ({}^i\mathbf{r}_{\tilde{c}_{i+1}} - {}^i\mathbf{r}_{\tilde{c}_i})] \cdot E - ({}^i\mathbf{r}_{\tilde{c}_{i+1}} - {}^i\mathbf{r}_{\tilde{c}_i}) \cdot ({}^i\mathbf{r}_{\tilde{c}_{i+1}} - {}^i\mathbf{r}_{\tilde{c}_i})^T \right) \\ {}^{\tilde{c}_i}\tilde{I}_i &= {}^{\tilde{c}_i}I_i + {}^{\tilde{c}_i}\tilde{I}_{i+1}\end{aligned}$$

- Berechnen der Spalten der oberen rechten Dreiecksmatrix für  $i = n \dots 1$ :

- Beschleunigung des gemeinsamen Schwerpunktes und daraus resultierende Kraft und Drehmoment

$$\begin{aligned}\dot{\omega}_i &= {}^iR_{i-1} \left( \rho_i \left[ {}^{i-1}\mathbf{e}_{z_{i-1}} \ddot{q}_i \right] \right) \\ \dot{\mathbf{v}}_{\tilde{c}_i} &= \dot{\omega}_i \times {}^i\mathbf{r}_{\tilde{c}_i} + (1 - \rho_i) {}^{i-1}\mathbf{e}_{z_{i-1}} \ddot{q}_i \\ \mathbf{N}_i &= {}^{\tilde{c}_i}\tilde{I}_i \cdot \dot{\omega}_i \\ \mathbf{F}_i &= \tilde{m}_i \cdot \dot{\mathbf{v}}_{\tilde{c}_i}\end{aligned}$$

- Resultierende Kraft/Moment im Gelenk  $i$

$$\begin{aligned}\mathbf{f}_i &= \mathbf{F}_i \\ \mathbf{n}_i &= \left( {}^iR_{i-1} {}^{i-1}\mathbf{r}_i + {}^i\tilde{\mathbf{r}}_{c_i} \right) \times \mathbf{F}_i + \mathbf{N}_i \\ \tau_i &= \begin{cases} \left( {}^i\mathbf{e}_{z_{i-1}} \right)^T \mathbf{n}_i, & \text{falls } \rho_i = 1 \text{ (Drehgelenk)} \\ \left( {}^i\mathbf{e}_{z_{i-1}} \right)^T \mathbf{f}_i, & \text{falls } \rho_i = 0 \text{ (Schubgelenk)} \end{cases}\end{aligned}$$

- Resultierende Momente in den davor liegenden Gelenken  $j = i - 1, \dots, 1$

$$\begin{aligned}\mathbf{f}_j &= {}^jR_{j+1} \mathbf{f}_{j+1} \\ \mathbf{n}_j &= {}^iR_{j+1} \mathbf{n}_{j+1} + \left( {}^jR_{j-1} {}^{j-1}\mathbf{r}_j \right) \times \left( {}^jR_{j+1} \mathbf{f}_{j+1} \right) \\ \tau_j &= \begin{cases} \left( {}^j\mathbf{e}_{z_{j-1}} \right)^T \mathbf{n}_j, & \text{falls } \rho_j = 1 \text{ (Drehgelenk)} \\ \left( {}^j\mathbf{e}_{z_{j-1}} \right)^T \mathbf{f}_j, & \text{falls } \rho_j = 0 \text{ (Schubgelenk)} \end{cases}\end{aligned}$$

- für  $j = i, \dots, 1$ :
  - setze die Elemente der j-ten Spalte  $m_{ij} = \tau_i$
  - spiegeln wegen Symmetrie  $m_{ji} = \tau_i$

**Bemerkung:** Für dieses Verfahren wurde rückwirkend von Featherstone die Bezeichnung „Composite Rigid Body Algorithm“ (CRBA) eingeführt<sup>11</sup>.

#### 5.6.1.4 Vergleich der Verfahren bzgl. Berechnungsaufwand

Alle drei Verfahren zur Berechnung der Massenmatrix haben quadratischen Aufwand, allerdings unterscheiden sie sich dennoch deutlich im Rechenaufwand. Im Verfahren 1 wird jedes Element der Massenmatrix tatsächlich berechnet, es werden  $n$  vollständige Durchläufe des RNE benötigt. Verfahren 2 und 3 reduzieren den Aufwand durch Ausnutzung der Symmetrie der Massenmatrix, so dass nur  $\frac{n(n+1)}{2}$  Elemente berechnet werden müssen. Dazu verwendet Verfahren 2 immer noch komplettte Schritte des RNE (jedoch pro Spalte nur für einen Teil der Elemente). In Verfahren 3 werden durch geschickte Initialisierung und die Annahme eines (bis auf jeweils ein beschleunigtes Gelenk) vollständig ruhenden Manipulators viele Teilausdrücke des RNE Null, so dass sich die Berechnung weiter vereinfacht.

Soll eines der Verfahren zur Berechnung der Vorwärtsdynamik eines Manipulators eingesetzt werden, besteht immer die Notwendigkeit, das entstehende lineare Gleichungssystem

$$M(\mathbf{q}(t)) \cdot \ddot{\mathbf{q}}(t) = \boldsymbol{\tau}(t) - (\mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{G}(\mathbf{q}(t)))$$

nach  $\ddot{\mathbf{q}}$  zu lösen. Dieser Schritt hat die Berechnungskomplexität  $O(n^3)$ , so dass die Komplexität der gesamten Berechnung der Vorwärtsdynamik mit den hier vorgestellten Verfahren im Allgemeinen die Komplexität  $O(n^3)$  hat.

Beim Lösen des linearen Gleichungssystems lassen sich die speziellen Eigenschaften der Massenmatrix ausnutzen. Da diese symmetrisch und positiv definit ist, lässt sich das Cholesky-Verfahren einsetzen. Dieses Verfahren benötigt im Vergleich zu einer herkömmlichen LR-Zerlegung nur ca. die Hälfte der Rechenoperationen.

#### 5.6.2 Berechnen der Vorwärtsdynamik ohne explizites Aufstellen der Massenmatrix

Alle bisher besprochenen Verfahren für DIR DYN haben im Prinzip eine Berechnungskomplexität von  $O(n^3)$ . Es wurden verschiedene Verfahren mit besseren Laufzeiteigenschaften entwickelt. Eines der bekanntesten dieser Verfahren ist der „Articulated Body Algorithm“ (ABA) von Featherstone<sup>12</sup>. Dieses Verfahren kommt ohne explizites Aufstellen der Massenmatrix und ohne Lösen eines linearen Gleichungssystems aus. Es besteht lediglich aus drei Iterationen über die kinematische Kette des Manipulators und hat somit die Komplexität  $O(n)$ .

Von Featherstone wurden Untersuchungen zu den Laufzeiteigenschaften verschiedener Dynamikalgorithmen publiziert. Zwei interessante Ergebnisse sind<sup>13</sup>:

<sup>11</sup>R. Featherstone: *Robot Dynamics Algorithms* (Kluwer Academic Publishers, 1987).

<sup>12</sup>R. Featherstone: *The Calculation of Robot Dynamics Using Articulated-Body Inertia*, International Journal of Robotics Research, 2(1):13-30, 1983.

<sup>13</sup>R. Featherstone: *Rigid Body Dynamics Algorithms* (Springer, 2008), Kapitel 10.3 Efficiency, S. 201ff.

- Der CRBA hat zwar kubische Komplexität, aber bis zu einer Anzahl von ca.  $n = 45$  Gliedern des Manipulators dominieren die quadratischen Terme.
- Aufgrund seiner linearen Komplexität ist der ABA zwar im Allgemeinen schneller als der CRBA, aber für Manipulatoren mit wenigen Gliedern ( $n \leq 8$ ) ist der CRBA dennoch effizienter.

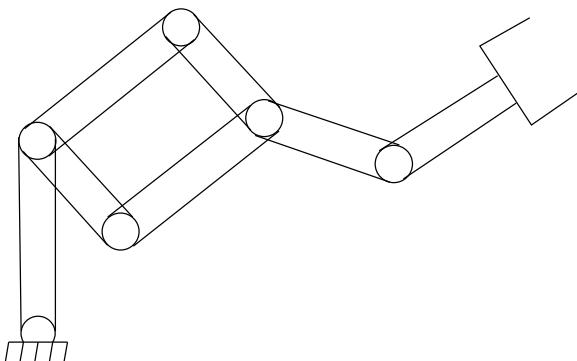
### 5.6.3 Quelloffene Roboterdynamik-Bibliothek MBSlib

Die in Kapitel 5.1, Seite 98, beschriebene Mehrkörpersystem-Bibliothek MBSLIB<sup>14</sup> enthält Implementierungen für CRBA und ABA, um je nach Anzahl der Freiheitsgrade des betrachteten Systems eine effiziente Berechnung der Vorwärtsdynamik zu ermöglichen. Die Methoden `doCRBA` und `doABA` berechnen für ein zuvor definiertes Mehrkörpersystem mit angegebenen Gelenkdrehmomenten und -kräften die jeweiligen Gelenkbeschleunigungen. Aus diesen Gelenkbeschleunigungen können durch geeignete (numerische) Integration über die Zeit die Gelenkgeschwindigkeiten und -positionen bestimmt werden.

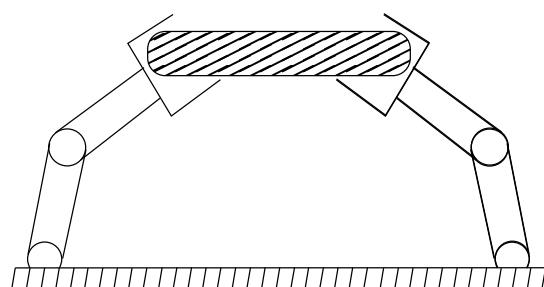
## 5.7 Geschlossene kinematische Ketten

Nicht alle Manipulatoren und Roboter lassen sich als offene kinematische Kette modellieren.

### Beispiele:



Manipulator mit Pantographgelenk



beidhändig manipulierender, zweiarmiger Roboter

Bei Robotern mit **kinematischen Schleifen** (closed kinematic chains) ist die Anzahl der tatsächlichen Freiheitsgrade (DOF) **kleiner** als die Anzahl der Gelenke.

Die effiziente Modellierung und Online-Simulation der Dynamik von Robotern mit kinematischen Schleifen für Zwecke der nichtlinearen, modellgestützten Regelung und Bahnplanung ist seit etwa 15-20 Jahren verstärkt Gegenstand der Forschung. Die für offene kinematische Ketten typische Struktur der inversen Dynamik als System von gewöhnlichen Differentialgleichungen (5.11) geht bei geschlossenen kinematischen Ketten in ein differential-algebraisches Gleichungssystem (differential algebraic equations) über

$$\tau = M(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) - J_c^T(\mathbf{q}) \mathbf{f}_c \quad (5.20)$$

$$0 = \mathbf{g}_c(\mathbf{q}). \quad (5.21)$$

<sup>14</sup>[www.sim.tu-darmstadt.de/mbslib](http://www.sim.tu-darmstadt.de/mbslib)

Dabei beschreibt  $g_c(\mathbf{q}) = 0$  die algebraischen Zwangsbedingungen ( $c$  für „constraint“),  $J_c = \partial g_c / \partial \mathbf{q}$  deren Jacobi-Matrix und  $\mathbf{f}_c$  die in Richtung der Beschränkung wirkende Kraft.

## 5.8 Berücksichtigung von Nichtstarrkörpereffekten

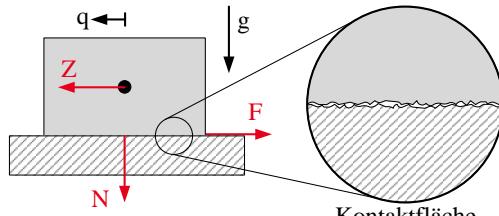
In der Realität treten weitere Effekte auf, welche die Genauigkeit der kinematischen und kinetischen Robotermodellierung und -simulation beeinflussen können und die in der Modellierung mit Starrkörpern bisher noch nicht berücksichtigt wurden. Dazu zählen:

- Reibung (engl. friction) in den Gelenken,
- Elastizität (engl. elasticity) in den Gelenkantrieben oder den Gliedern selbst und
- Lose (Getriebespiel, engl. backlash).

### 5.8.1 Reibung

Reibungseffekte beeinflussen permanent die Qualität von Roboterbewegungen, z.B. als Reibung in den angetriebenen Gelenken oder als Reibung beim Kontakt zwischen einem mobilen Roboter und dem Untergrund, über den dieser sich fortbewegt.

Sich **berührende Oberflächen** sind typischerweise **nicht ideal glatt**, sodass nicht nur **Kräfte** normal zur Berührungsebene, sondern **auch tangential übertragen** werden. Diese werden als **Reibungskräfte** bezeichnet.



Z: Zugkraft N: Normalkraft F: Reibungskraft

**Reibungskräfte und -momente** in den Gelenken, die den Motormomenten und -kräften **entgegen** wirken, können in der Regel **nicht vernachlässigt** werden.

Beispielsweise können bei einem typischen Industrieroboter mit sechs Drehgelenken (wie dem auf Seite 81 abgebildeten Roboter) Reibungskräfte in den ersten drei Gelenken etwa 10 % des Motornennmoments<sup>15</sup> und bei den drei Handachsen (den letzten drei Gelenken) sogar bis zu 50 % des Motornennmoments und mehr ausmachen.

Die Bewegungs(differential)gleichungen INVDYN mit Reibungskräften und -momenten lauten:

$$\tau = M(\mathbf{q}) \cdot \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) + \underbrace{F(\mathbf{q}, \dot{\mathbf{q}})}_{\text{Vektor der Reibungs- und Dämpfungskräfte und -momente}}$$

<sup>15</sup>Das Motornennmoment ist das maximale Motormoment, das über einen längeren Zeitraum permanent aufgebracht werden kann (kann z.B. halb so groß sein wie das Motorspitzenmoment, das nur kurz erzeugt werden kann).

Häufig wird als **einfaches** (aber bereits effektives) **Reibungsmodell** für das  $i$ -te Gelenk verwendet:

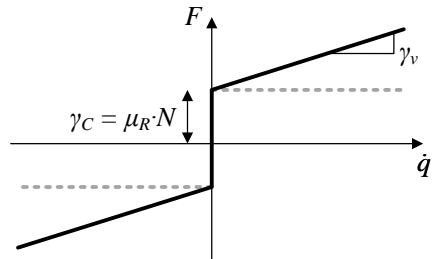
$$F_i(\mathbf{q}, \dot{\mathbf{q}}) = \underbrace{\gamma_{C,i} \cdot \text{sign}(\dot{q}_i)}_{\text{Coulombsche Reibung} \atop (\text{Coulomb friction})} + \underbrace{\gamma_{v,i} \cdot \dot{q}_i}_{\text{viskose Gleitreibung} \atop (\text{viscous friction})}, \quad \text{mit } \gamma_{C,i}, \gamma_{v,i} = \text{const} \in \mathbb{R}.$$

$$\text{sign}(\dot{q}_i) = \begin{cases} +1, & \text{falls } \dot{q}_i > 0, \\ 0, & \text{falls } \dot{q}_i = 0, \\ -1, & \text{falls } \dot{q}_i < 0. \end{cases}$$

Ein **verfeinertes Modell von Reibungseffekten** kann wie nachfolgend beschrieben erhalten werden. Eine physikalisch genaue Modellierung von Reibungseffekten ist in der Regel jedoch wesentlich komplexer und aufwendiger als die hier vorgestellten approximativen Modelle, die allerdings für Echtzeit-Anwendungen in der Regelung und Steuerung von Robotern geeignet sind.

**Coulombsche Reibung:** Sobald die Zugkraft  $Z$  einen Grenzwert überschreitet, wird die Haftung überwunden und die Oberflächen gleiten mit Reibungsverlusten aufeinander. Nach Charles Augustine de Coulomb ist  $\gamma_{C,i}$  näherungsweise proportional zur Normalkraft  $N_i$  (hier als konstant angenommen):

$$\gamma_{C,i} = \underbrace{\mu_{R,i}}_{\text{Reibungskoeffizient}} \cdot N_i.$$



**Viskose Gleitreibung:** Die viskose Gleitreibung wird häufig mit  $\gamma_{v,i} \cdot \dot{q}_i$  als proportional zur Geschwindigkeit modelliert und berücksichtigt zum Beispiel durch Schmiermittel erzeugte hydromechanische Effekte.

Dieses Reibungsmodell kann nun noch um den Effekt der **Haftung** erweitert werden.

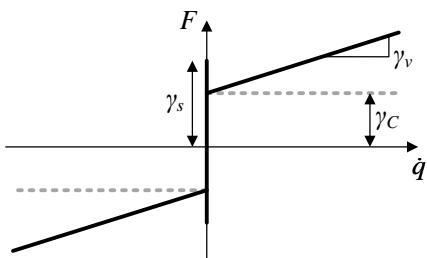
Der Grenzwert der Haftung  $\gamma_{s,i}$  (engl. stiction), auch statische Reibkraft oder Haftriebung genannt, ist proportional zur Normalkraft  $N_i$ :

$$\gamma_{s,i} = \underbrace{\mu_{H,i}}_{\text{Haftungskoeffizient}} \cdot N_i.$$

Der Haftungskoeffizient  $\mu_{H,i}$  ist typischerweise größer als der Reibungskoeffizient  $\mu_{R,i}$ . Hierbei kommt es zu ruckartigem Gleiten (Losbrechen) bzw. Festsetzen der Flächen. Dies wird als Haftgleiteffekt (engl. stick-slip-effect) bezeichnet.

Somit erhält man das klassische Reibungsmodell:

$$F_i(\mathbf{q}, \dot{\mathbf{q}}) = \begin{cases} \gamma_{C,i} \cdot \text{sign}(\dot{q}_i) + \gamma_{v,i} \cdot \dot{q}_i & \text{wenn } \dot{q}_i \neq 0 \\ Z & \text{wenn } \dot{q}_i = 0 \wedge |Z| < \gamma_{s,i} \\ \gamma_{s,i} \cdot \text{sign}(Z) & \text{sonst} \end{cases}$$



In diesem Modell ist der Übergang zwischen Haftung und Gleitreibung sprunghaft, welches dem realen Verhalten nur annähernd entspricht.

Tatsächlich nimmt der Kontakt der Oberflächen und somit die Reibkraft  $F$  mit zunehmender Geschwindigkeit zunächst ab (Schmierspalt entsteht). Dieser Übergang kann mit der **Stribeck-Reibung** beschrieben werden. Die Stribeck-Geschwindigkeit  $\dot{q}_{\text{stribeck}}$  bestimmt den exponentiellen Übergang von Haft- zu Gleitreibung.

Somit erhält man insgesamt das genauere Reibungsmodell<sup>16</sup>:

$$F_i(\mathbf{q}, \dot{\mathbf{q}}) = \left( \gamma_{C,i} + (\gamma_{s,i} - \gamma_{C,i}) e^{-\left| \frac{\dot{q}_i}{\dot{q}_{\text{stribeck}}} \right|} \right) \cdot \text{sign}(\dot{q}_i) + \gamma_{v,i} \cdot \dot{q}_i.$$

### Bemerkungen:

- Modelle für Reibungskräfte sind meist mit erheblichen Unsicherheiten behaftet (z.B. die Werte der Parameter  $\gamma_{C,i}$ ,  $\gamma_{v,i}$ ). Oft hängt die tatsächliche Reibungskraft auch nicht nur von  $\dot{\mathbf{q}}$  sondern auch von der momentanen Gelenkkonfiguration  $\mathbf{q}$ , Temperatur, etc. ab.
- Es sind noch zahlreiche weitere Ansätze zur Modellierung von Reibung in der Literatur zu finden.
- Beim Newton-Euler Formalismus berücksichtigt man Reibungskräfte im letzten Schritt von (NE.3):

$$\tau_{i,\text{neu}} = \tau_i + F_i(\mathbf{q}, \dot{\mathbf{q}}).$$

## 5.8.2 Elastizität<sup>17</sup>

Man versucht üblicherweise bei Industrierobotern Elastizität in Gelenken und Gliedern konstruktiv so weit wie möglich zu vermeiden, um unterschiedlich schwere Massen schnell und positionsgenau bewegen zu können. Allerdings kann ein elastisches Deformationsverhalten nicht völlig eliminiert werden. Üblicherweise verursachen Elastizitäten in den Getrieben von Drehgelenken ca. drei Viertel der noch auftretenden elastischen Effekte bei Industrierobotern und Verformungen der Glieder lediglich etwa ein Viertel. Durch die Verwendung von Robotermodellen mit Elastizitäten in der Robotersteuerung und -regelung versucht man elastische Effekte bei Industrierobotern zu kompensieren und eine höhere Positions- und Bewegungsgenauigkeit zu erreichen.

Andererseits ist hohe, veränderliche Elastizität in Verbindung mit einem „intelligenten“ Steuerungssystem einer der Schlüssel für die überragende Leistungsfähigkeit biologischer Arme und Beine bzgl. hoher Bewegungsgüte bei hohen Geschwindigkeiten und einem hervorragenden Verhältnis von Traglast zu Eigengewicht. Elastizitäten ermöglichen einerseits das Abfangen von Kollisionen und andererseits Speichern und Wiederverwenden von Energie zur Erhöhung der Leistungsfähigkeit. Die in diesem Abschnitt besprochene Erweiterung der Bewegungsdynamik starrer auf gelenkelastische Roboter ist prinzipiell für Industrieroboter und für biologisch inspirierte, elastische Roboter anwendbar.

<sup>16</sup>Es existieren noch weitere Varianten z.B. mit verschiedenen potenzierten Exponentialfunktionen.

<sup>17</sup>Die erste Version dieses Kapitels beruht auf einer Ausarbeitung von Sebastian Klug und Thomas Lens.

### 5.8.2.1 Grundlagen zu elastischen Gelenken und Gliedern

Elastische Körper verformen sich unter Einwirkung einer Kraft  $f$  bzw. eines Drehmoments  $n$  und speichern potentielle Energie, die bei der Rückfederung unter Berücksichtigung von Reibungsverlusten wieder abgegeben wird. Die Verformbarkeit und Dämpfung kann je nach Material sehr unterschiedlich sein.

Bei Belastung durch die Kraft  $f$  bzw. das Moment  $n$  verschiebt sich der Kraftangriffspunkt um den Federweg  $s$  bzw. den Drehwinkel  $\varphi$ . Das Verhältnis aus Federkraft  $f$  und Federweg  $s$  (bzw. Federmoment  $n$  und Verdrehwinkel  $\varphi$ ) wird als Federsteifigkeit  $c$  (oft auch Federrate  $R$ ) bezeichnet

$$c = \frac{\Delta f}{\Delta s} \quad \text{bzw.} \quad c_\varphi = \frac{\Delta n}{\Delta \varphi}.$$

Trägt man die Verformung in Abhängigkeit von der Belastung in einem Kraft-Weg-Diagramm auf, erhält man typische Federkennlinien wie in Abb. 5.2 dargestellt.

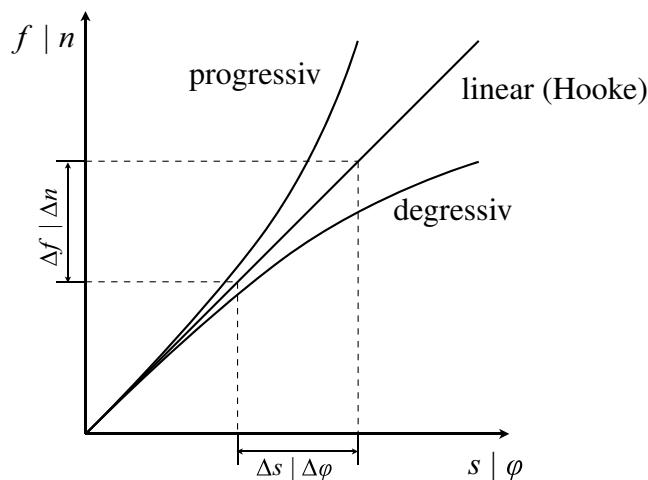


Figure 5.2: Typische Kraft-Weg- bzw. Moment-Winkel-Kennlinien von Federn

Arbeitet eine Feder nach dem Hookeschen Gesetz und reibungsfrei, so ergibt sich eine *lineare* Kennlinie. Das Hookesche Gesetz besagt, dass die Verformung proportional zur angreifenden Belastung ist. Trifft dies zu spricht man auch von einer Federkonstanten. Je steiler die Kennlinie verläuft, umso geringer ist die Verformung bei gleicher Belastung, d.h. desto härter ist die Feder.

Ist die Federsteifigkeit über den Arbeitsbereich veränderlich, so erhält man eine *nichtlineare* Kennlinie. Hierbei unterscheidet man zwischen

- *progressiven Federkennlinien*, die mit steigender Belastung härter werden, wodurch das Durchschlagen der Feder verhindert und ein schnelleres Abklingen von Schwingungen erreicht werden kann, und
- *degressiven Kennlinien*, die nach einer bestimmten Belastung im Verhältnis weniger Kraft brauchen um weiter verformt zu werden, wie es zum Beispiel beim Spiel- und Druckausgleich bei Reglern benötigt wird.

Bei beiden ist  $c$  nicht konstant, sondern von  $s$  bzw.  $\varphi$  abhängig:

$$c(s) = \frac{\Delta f}{\Delta s} \quad \text{bzw.} \quad c_\varphi(\varphi) = \frac{\Delta n}{\Delta \varphi}.$$

Die in einer Feder gespeicherte, potentielle Energie  $P$  berechnet sich, wie aus dem Physikunterricht noch bekannt sein dürfte, mit

$$P = \frac{1}{2}c\Delta s^2 \quad \text{bzw.} \quad P = \frac{1}{2}c_\varphi\Delta\varphi^2.$$

### 5.8.2.2 Elastizitäten in der Robotik

Wie im Kapitel 5.4, beschrieben kann die Bewegungsdynamik eines Roboters, z.B. eines typischen Industrieroboters, durch ein System von gewöhnlichen Differentialgleichungen mit den Gelenkvariablen als verallgemeinerte Koordinaten beschrieben werden.

Bei der Entwicklung von Industrierobotern wird besonderer darauf geachtet, Elastizitäten in den Gliedern und Gelenken konstruktionsbedingt zu vermeiden, so dass einerseits eine hohe Positionsgenauigkeit erreicht und der Roboter mit relativ einfachen Reglerstrukturen beherrschbar ist. Dies bedingt jedoch eine hohe Eigenmasse des Roboters und führt zu Verhältnissen von Traglast zu Eigengewicht von 1:7 bis 1:15 bei typischen Industrierobotern. Bei schnellen, hochfrequenten Bewegungen, so wie bei sehr ausladenden oder Leichtbau-Konstruktionen, ist die Annahme der Starrkörperstruktur nicht mehr so gut zutreffend.

In den folgenden Ausführungen werden ausschließlich Drehgelenke betrachtet. Es lassen sich zwei grundlegende Fälle von bei Drehgelenksrobotern (auch Knickarmroboter genannt) auftretender Elastizität unterscheiden:

- (i) Roboter mit elastischen Gliedern (engl. elastic links) (siehe Abb. 5.3), die von elastischen Verformungen betroffen sind, wie es bei leichten und langen Strukturen der Fall ist, z.B. Leichtbau- oder Kranroboter: Hier hängt die Deformation von der Gelenkwinkelstellung und der Länge des Roboterglieds bzw. der Position  $r$  auf diesem und der Temperatur  $T$  ab.

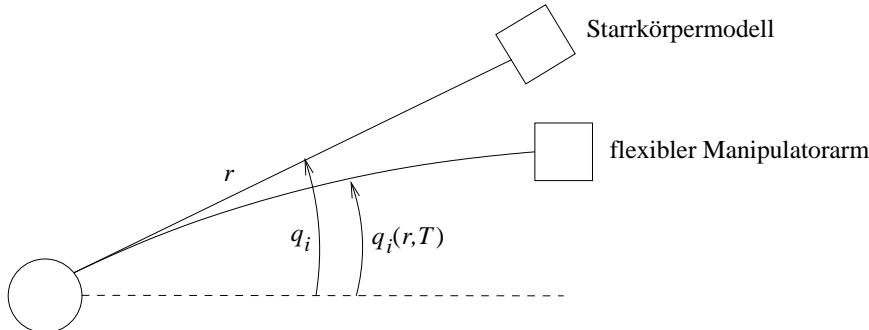


Figure 5.3: Elastisches Roboterglied, wie es z.B. bei spezifischen Typen von Leichtbauroboterarmen auftritt.

- (ii) Roboter mit elastischen (Dreh-)Gelenken (engl. elastic joints), wie sie typischerweise in den unterschiedlichen Getriebearten auftreten. Hier unterscheidet sich die antriebsseitige Gelenkposition  $\theta$  wesentlich von der abtriebsseitigen Gelenkposition  $\theta_{El}$ .

Beide Fälle werden meist separat betrachtet, da jeweils einer der beiden Effekte dominiert. Sie können aber auch in Kombination auftreten und betrachtet werden.

Beide Fälle, insbesondere die Gelenkelastizität, können näherungsweise durch ein vereinfachtes Ersatzmodell beschrieben werden (siehe Abb. 5.5). Jedes Gelenk wird dafür

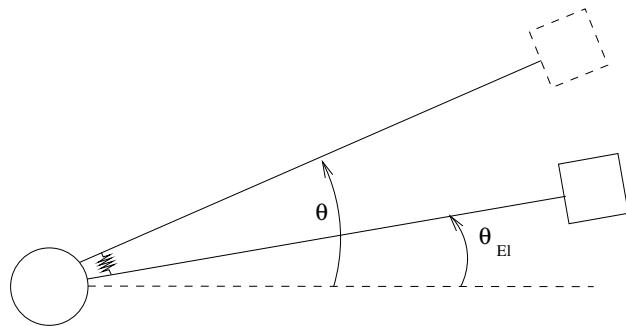


Figure 5.4: Elastisches Robotergelenk

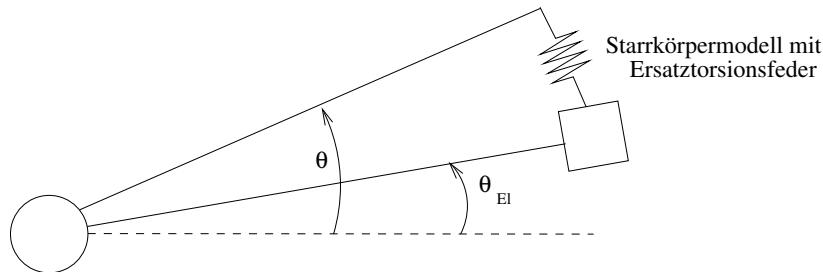


Figure 5.5: Ersatzmodell für ein elastisches Robotergelenk oder -glied.

mit dem darauf folgenden Verbindungssegment als ein Zweimassen-System modelliert, bei dem die skizzierte Gelenkvariable  $\theta$  die Motorposition darstellt und die weitere Gelenkvariable  $\theta_{\text{El}} = \mathbf{q}$  die reale, abtriebsseitige Position des Manipulators. Wenn dies für jedes Gelenk gemacht wird, **verdoppelt** sich dadurch die Anzahl an Bewegungsdifferentialgleichungen in INV DYN bzw. DIR DYN. Für die Beschreibung der Bewegung werden folglich  $2n$  Gelenkvariablen benötigt:

$\theta_i$  = Gelenkvariable der Position des (steifen) Drehgelenksmotors (vor dem elastischen Getriebe),

$q_i = \theta_{\text{El},i}$  Gelenkvariable zur Beschreibung der tatsächlichen Gliedposition, die aufgrund der (Gelenk- oder Glied-)Elastizität von  $\theta_i$  verschieden ist (bei Gelenkelastizitäten ist dies die abtriebsseitige Gelenkposition).

Eine Möglichkeit zur Herleitung der dynamischen Gleichungen eines Roboters mit Elastizität bietet der Lagrange-Formalismus aus Kapitel 5.4 mit seiner allgemeinen Form

$$\mathcal{L} = K_R - P_R$$

als Differenz aus kinetischer und potentieller Energie. Für die Betrachtung von elastischen Robotern müssen die einzelnen Terme allerdings erweitert werden.

Die potentielle Energie des Roboters setzt sich zusammen, aus der potentiellen Energie im Gravitationsfeld  $P_G$  und der in den Gelenkfedern gespeicherten Energie  $P_{\text{El}}$

$$P_R = P_G + P_{\text{El}}.$$

Aufgrund der elastischen Verformung im Gelenk ist die Energie im Gravitationsfeld nur von der abtriebsseitigen Position des Gelenks  $\mathbf{q}$  abhängig

$$P_G = P_G(\mathbf{q}).$$

Die in der Feder gespeicherte Energie ergibt sich aus der Differenz von Motorposition und abtriebsseitiger Armposition mit  $E_S$  als der diagonalen Steifigkeitsmatrix, deren Diagonalelemente die einzelnen Gelenksteifigkeiten enthalten

$$P_{\text{El}} = P_{\text{El}}(\mathbf{q}, \theta) = \frac{1}{2}(\theta - \mathbf{q})^T E_S (\theta - \mathbf{q}).$$

Da sich bei einem elastischen Roboter Gelenkmotoren und Armglieder relativ „unabhängig“ voneinander bewegen können, ergibt sich für die kinetische Energie eine Abhängigkeit sowohl von  $\theta$  als auch von  $\mathbf{q}$  als auch von deren Geschwindigkeiten

$$K_R = K(\mathbf{q}, \theta, \dot{\mathbf{q}}, \dot{\theta}).$$

Durch Zusammenfassung der Variablen für Motor- und Gliedposition zum Vektor  $\mathbf{p}$  als verallgemeinerte Gelenkvariablen

$$\mathbf{p} = \begin{pmatrix} \mathbf{q} \\ \theta \end{pmatrix}$$

ergibt sich die Schreibweise der kinetischen Energie wieder in der bekannten Form

$$K_R(\mathbf{p}, \dot{\mathbf{p}}) = \frac{1}{2} \dot{\mathbf{p}}^T M_R(\mathbf{p}) \dot{\mathbf{p}}.$$

Hierbei ist  $M_R(\mathbf{p})$  jedoch **nicht** die Massenmatrix  $M$  des starren Roboters sondern die erweiterte Trägheitsmatrix mit

$$M_R(\mathbf{p}) = M_R(\mathbf{q}, \theta) = \begin{pmatrix} M(\mathbf{q}) & S(\mathbf{q}, \theta) \\ S^T(\mathbf{q}, \theta) & J_M(\theta) \end{pmatrix}.$$

Die Matrizen  $M$ ,  $S$  und  $J_M$  haben jeweils die Dimension  $n \times n$ .

- Die Matrix  $M(\mathbf{q})$  entspricht der Massenmatrix des starren Roboters unter Abhängigkeit der abtriebsseitigen Gelenkpositionen  $\mathbf{q}$ .
- $J_M(\theta)$  ist die Trägheitsmatrix der Motoren und somit theoretisch abhängig von der Motorposition  $\theta$ . Für die meisten Motoren ist jedoch folgende erste Annahme zutreffend:

**(A1)** *Die Massenverteilung des Rotors innerhalb eines Drehgelenksmotors ist rotationssymmetrisch und sein Schwerpunkt liegt auf der Drehachse des Motors, zu welcher die Massenverteilung rotationssymmetrisch ist.*

Dadurch ergibt sich für  $J_M$  eine konstante, diagonale Matrix, die die Trägheitsmomente der Rotoren als Diagonalelemente enthält.

- Die Matrix  $S(\mathbf{q}, \theta)$  stellt die Trägheitsverkopplung zwischen den Rotoren und den Robotergliedern dar. Sie entsprechen der kinetischen Energie der Rotoren um andere Achsen als die Drehachse des Motors, die durch die Bewegung des Roboters selbst entstehen.

Zur weiteren Modellierung wird eine zweite Annahme über die Motoren gemacht:

**(A2)** *Die kinetische Energie der Rotoren wird hauptsächlich durch ihre eigene Drehgeschwindigkeit um die Motorachse verursacht.*

Da bei den meisten Robotern elektrische Drehgelenksmotoren mit hoher Übersetzung einer schnellen Rotationsbewegung mit niedrigem Drehmoment in eine langsamere Rotationsbewegung mit höherem Drehmoment eingesetzt werden (typischerweise 1:50 bis 1:200), trifft diese Aussage in den meisten Fällen zu. Somit kann der Einfluss der Roboterbewegung auf die Rotoren in erster Näherung vernachlässigt und  $S(\mathbf{q}, \theta) = 0$  gesetzt werden. Daraus folgt

$$M_R(\mathbf{q}, \theta) = \begin{pmatrix} M(\mathbf{q}) & 0 \\ 0 & J_M \end{pmatrix}.$$

Damit lässt sich die kinetische Energie  $K_R = K_A + K_M$  als Summe zweier Terme beschreiben. Der eine beschreibt die kinetische Energie in den Armgliedern des Roboters  $K_A$

$$K_A(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}},$$

und der andere die kinetische Energie der rotierenden Motoren  $K_M$

$$K_M(\dot{\theta}) = \frac{1}{2} \dot{\theta}^T J_M \dot{\theta}.$$

Die Lagrangefunktion lässt sich nun komplett darstellen als

$$\mathcal{L} = (K_A(\mathbf{q}, \dot{\mathbf{q}}) + K_M(\dot{\theta})) - (P_G(\mathbf{q}) + P_{El}(\mathbf{q}, \theta)) = \mathcal{L}(\mathbf{q}, \theta, \dot{\mathbf{q}}, \dot{\theta}) = \mathcal{L}(\mathbf{p}, \dot{\mathbf{p}}).$$

Analog zur Formulierung des **einen** Systems von Differentialgleichungen zweiter Ordnung für Roboter mit starren Gliedern und Gelenken (Gleichung (5.19) auf Seite 127 in Kapitel 5.4 bzw. Gleichung (5.11) auf Seite 123) lassen sich hiermit nun die **zwei** Systeme von Differentialgleichungen zweiter Ordnung für die Beschreibung der motorseitigen bzw. abtriebsseitigen Dynamik des Roboters aufstellen (Angabe ohne Beweis):

Einerseits

$$\tau_{El} = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) \quad (5.22)$$

$$\tau_m = J_M \ddot{\theta} + \tau_{El} \quad (5.23)$$

andererseits

$$\tau_{El} = E_S(\theta - \mathbf{q}),$$

wobei  $\tau_{El}$  die Drehmomente in den Federn beschreibt (mit diagonaler Steifigkeitsmatrix  $E_S$  der Gelenkelastizitäten). Der Ausdruck  $C(\mathbf{q}, \dot{\mathbf{q}})$  fasst die Coriolis- und Zentrifugalterm zusammen und  $G(\mathbf{q})$  ist der Vektor der Beschleunigung durch die Gravitation. Alle auftretenden Größen sind im Übrigen implizit von der Zeit  $t$  als unabhängiger Variablen abhängig.

Gleichung (5.22) entspricht der Dynamik eines starren Roboters, wie aus den Kapiteln 5.3.2 und 5.4 bekannt. Das antreibende Drehmoment kommt jedoch von der Gelenkfeder und nicht vom Motor, während das Motordrehmoment über die Motordynamik (5.23) daran gekoppelt ist. Ein weiterer Vorteil dieses Modells ist, dass es eine direkte und intuitive Verallgemeinerung des Modells für starre Roboterkinematiken darstellt.

### 5.8.2.3 Berechnung des inversen Dynamikmodells bei (hohen) Drehgelenkelastizitäten<sup>18</sup>

Zur Berechnung des inversen Dynamikmodells INV DYN eines Roboterarms mit hochelastischen Drehgelenken

$$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n \rightarrow \boxed{\text{INV DYN}} \rightarrow \tau_m \in \mathbb{R}^n$$

wird ein zusätzlicher Berechnungsschritt benötigt, da die benötigten Motortrajektorien  $\theta(t)$  bzw.  $\ddot{\theta}(t)$  in (5.23) nicht bekannt sind. Diese lassen sich aus den Gleichungen der Drehmomente in den Federn berechnen:

$$\begin{aligned} \tau_{\text{El}} &= E_S \cdot (\theta - \mathbf{q}) \\ \Rightarrow \theta &= E_S^{-1} \cdot \tau_{\text{El}} + \mathbf{q}. \end{aligned} \quad (5.24)$$

Dabei bezeichnen  $\mathbf{q}$  die abtriebsseitigen und  $\theta \in \mathbb{R}^n$  die antriebsseitigen Sollwerte der Gelenkpositionen. Die Inverse  $E_S^{-1}$  der diagonalen Steifigkeitsmatrix ist ebenfalls eine Diagonalmatrix deren Diagonale mit den Kehrwerten der Diagonalelementen der Steifigkeitsmatrix  $E_S$  besetzt ist.

Die Motordrehmomente  $\tau_m$  lassen sich damit durch Einsetzen von Gleichung (5.24) in die Gleichung der Motordynamik (5.23) berechnen:

$$\begin{aligned} \tau_m &= J_M \ddot{\theta} + \tau_{\text{El}} \\ &= J_M \cdot (E_S^{-1} \cdot \ddot{\tau}_{\text{El}} + \ddot{\mathbf{q}}) + \tau_{\text{El}}. \end{aligned}$$

Die Berechnung des inversen Dynamikmodells des Roboterarms mit elastischen Gelenken kann nun folgendermaßen erfolgen:

1. Berechnung der Drehmomente in den Gelenken des starren Modells (Gleichung 5.22) entweder numerisch nach der Methode von Newton-Euler (siehe Kapitel 5.3) oder symbolisch nach der Methode von Lagrange (siehe Kapitel 5.4):

$$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n \rightarrow \boxed{\text{INV DYN des starren Modells}} \rightarrow \tau_{\text{El}} \in \mathbb{R}^n$$

2. Berechnung von  $\ddot{\theta}$  mit Hilfe der zweifachen zeitlichen Ableitung  $\ddot{\tau}_{\text{El}}$  der Gelenkdrehmomente:

Bei Verwendung der Methode nach Lagrange müssen die Gleichungen (5.22) für  $\tau_{\text{El}}$  aus Schritt 1 zweimal total nach der Zeit differenziert werden (symbolisches Ausrechnen).

Bei Verwendung der Methode nach Newton-Euler muss die Ableitung numerisch durch mehrfache Ausführung von Schritt 1 ermittelt werden.

In beiden Fällen müssen die dritte und die vierte totale Zeitableitung von  $\mathbf{q}$  berechnet werden.

3. Berechnung der Motordrehmomente:

$$\tau_m = J_M \cdot (E_S^{-1} \cdot \ddot{\tau}_{\text{El}} + \ddot{\mathbf{q}}) + \tau_{\text{El}}$$

**Bemerkung:** Da in die Berechnung von  $\tau_{\text{El}}$  die Gelenkbeschleunigungen  $\ddot{\mathbf{q}}$  eingehen (siehe Gleichung 5.22), erfordert die Berechnung der Motordrehmomente  $\tau_m$ , die Berechnung von  $\ddot{\theta}$  und damit von  $\ddot{\tau}_{\text{El}}$  und damit von  $d^4\mathbf{q}(t)/dt^4$ . Dies setzt wiederum voraus, dass die Trajektorie  $\mathbf{q}(t)$  vierfach differenzierbar ist.

<sup>18</sup>Die erste Version dieses Kapitel beruht auf einer Ausarbeitung von Thomas Lens.

### 5.8.2.4 Elastizitäten in der Biologie – Bewegungsdynamik des Menschen

Im Gegensatz zu heutigen Industrierobotern ist die Elastizität im biologischen Bewegungsapparat eine für die Leistungsfähigkeit des Bewegungssystems wesentliche Eigenschaft. Auch wenn die einzelnen Glieder (Knochen) starr sind, sind die Antriebe (Muskeln) selbst elastisch.

Man kann die Funktion eines Muskels in einem vereinfachten, mechanischen Modell (Abb. 5.6) schematisch als ein System bestehend aus einem kontraktilen Element (CE), einem Dämpferelement (DE) und in Serie und parallel geschalteten elastischen Elementen (SE, PE) darstellen. Auch wenn der reale Aufbau des Muskels komplexer ist, lassen sich mit diesem mechanischen Ersatzmodell viele der für Bewegungen wesentlichen Eigenschaften eines Muskels beschreiben.

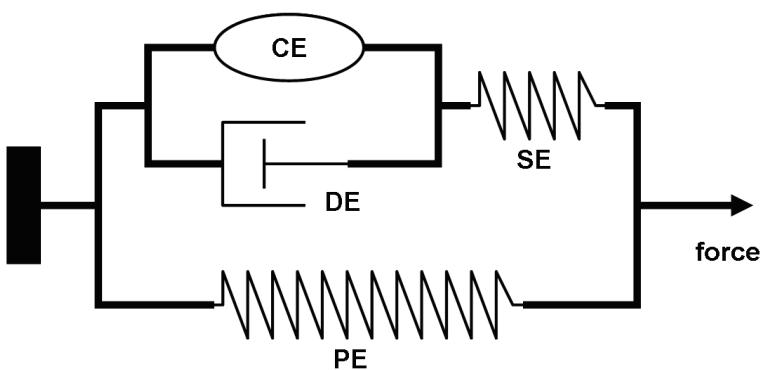


Figure 5.6: Typisches mechanisches Ersatzmodell eines menschlichen Muskels bzw. einer Muskelgruppe.

Im passiven Zustand befindet sich der Muskel in seiner Ruhelage und übt keine Kraft aus. Wird der Muskel durch eine äußere Kraft gedehnt, so werden die Muskelfasern passiv angespannt. Der passive Muskel verhält sich elastisch und zeigt in seinem Kraft-Weg-Diagramm einen stark progressiven Verlauf. Die Federsteifigkeit nimmt also mit der Länge des Muskels zu. Diese Elastizität kommt hauptsächlich durch dehnbare Strukturen zustande, die zu den kontraktilen Strukturen des Muskels parallel geschaltet sind, etwa das Sakrolemm (Zellmembran der Muskelzelle) und Bindegewebe zwischen den Muskelfasern. Aber auch die in Serie geschalteten Sehnen und die Titin-Fasern, die unter anderem eine Überdehnung des Muskels verhindern sollen, tragen zum nichtlinearen elastischen Verhalten des Muskels bei.

Der Muskel ist aber nicht nur ein passives elastisches Element. Seine Hauptfunktion ist die Krafterzeugung um Bewegungen zu ermöglichen. Auch im aktiven (d.h. angespannten) Zustand verliert der Muskel seine elastischen Fähigkeiten nicht. Aber je nach Stärke der Muskelaktivierung verändern sich die elastischen Eigenschaften während einer Bewegung. Die Elastizität und ihre Veränderung scheint mit ein Grund für die eleganten und flüssigen Bewegungen von biologischen Systemen zu sein.

### 5.8.2.5 Bezug zur Steuerung und Regelung von Bewegungen bei Mensch und Tier (siehe 6.12)

## 5.9 Spezielle Dynamikmodelle für zweibeinige Roboter und deren Stabilitätsregelung<sup>19</sup>

Die Bezeichnung *Stabilität* bezieht sich in diesem Abschnitt auf *posturale Stabilität* (postural stability), welche die Stabilität der aufrechten Fortbewegung eines zweibeinigen Roboters beschreibt.

Zur Definition der Stabilität des Laufens betrachtet man die konvexe Hülle der Bodenkontaktpunkte der Füße, über die der Roboter getragen wird.

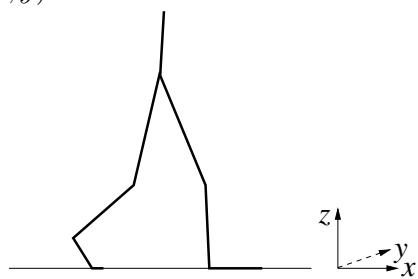


Figure 5.7: Konvexe Hülle der tragenden Bodenkontaktpunkte bei nur einem Fuß in Bodenkontakt (links) und bei Bodenkontakt beider Füße (rechts).

Solange der entlang der Richtung des Gravitationsvektors auf den Boden projizierte Schwerpunkt des zweibeinigen Roboters sich innerhalb der konvexen Hülle der tragenden Bodenkontaktpunkte befindet, so kann der Roboter nicht umfallen. Dies bezeichnet man als statische Stabilität bzw., falls dies während einer Gehbewegung erfüllt ist, bezeichnet man diese als *statisch stabiles Gehen*.

Die *dynamische Stabilität* laufender Roboter, insbesondere zweibeiniger Roboter, sicherzustellen, ist eine der größten Herausforderungen im Bereich der Robotik. Im Folgenden sollen die zwei am meisten verbreiteten und bewährten Ansätze zum Erreichen der Stabilität bei zweibeinigen Robotern vorgestellt werden.

Für die folgenden Überlegungen sind alle Größen soweit sinnvoll und nicht anders angegeben jeweils bezüglich und in einem Inertialkoordinatensystem, in dem der Roboter nach vorne in Richtung der  $x$ -Achse läuft, und die  $z$ -Achse nach oben gerichtet ist, angegeben. Der Bodenkontakt liege in der  $(x, y)$ -Ebene vor:



<sup>19</sup>Dieses Kapitel beruht auf einer Ausarbeitung von Maximilian Stelzer.

### 5.9.1 Zero-Moment-Point (ZMP)

Einfluß auf die Stabilität zweibeiniger Roboter hat u.a. die Kraftverteilung der Bodenreaktionskräfte bezüglich der konvexen Hülle der tragenden Bodenkontaktpunkte. Es treten zwei verschiedene Arten von wechselwirkenden Kräften auf: normale und tangentiale Kräfte.

#### Normale Kräfte

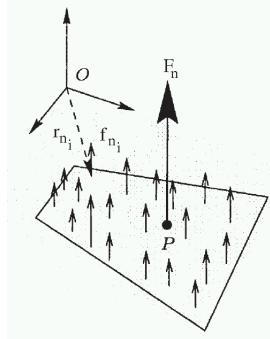
Die normalen Kräfte  $\mathbf{f}_{N_i}$  wirken normal zur Kontaktfläche, d.h.

$$\mathbf{f}_{N_i} = \begin{pmatrix} 0 \\ 0 \\ f_{N_i,z} \end{pmatrix}.$$

Der *Zero-Moment-Point* ist derjenige Punkt  $P$ , in dem die resultierende normale Kraft  $\mathbf{F}_N = \sum \mathbf{f}_{N_i}$  wirkt:

$$\mathbf{r}_{ZMP} := OP := \frac{\sum \mathbf{r}_{N_i} f_{N_i,z}}{\sum f_{N_i,z}}.$$

Dabei ist  $\mathbf{r}_{N_i}$  der Angriffspunkt der Kraft  $\mathbf{f}_{N_i}$ . Da alle  $f_{N_i,z} > 0$  sind, liegt  $P$  in oder am Rand der konvexen Hülle der Punkte, an denen Bodenreaktionskräfte auftreten. Um diesen Punkt  $P$  entsteht durch die normalen Kräfte kein Moment.

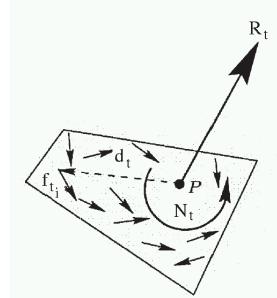


#### Tangentiale Kräfte

Die tangentialen Kräfte  $\mathbf{f}_{t_i}$  wirken in der Kontaktalebene, d.h.

$$\mathbf{f}_{t_i} = \begin{pmatrix} f_{t_i,x} \\ f_{t_i,y} \\ 0 \end{pmatrix}.$$

Im Punkt  $P$  können die angreifenden tangentialen Kräfte durch die resultierende Kraft  $\mathbf{F}_t = \sum \mathbf{f}_{t_i}$  und das resultierende Moment  $\mathbf{N}_t = \sum \mathbf{d}_i \times \mathbf{f}_{t_i}$ , wobei  $\mathbf{d}_i$  der Angriffspunkt der Kraft  $\mathbf{f}_{t_i}$ , dargestellt relativ zum ZMP, ist, beschrieben werden.



Durch die tangentialen Kräfte wirkt zwar ein Moment um  $P$ , das aber nicht zum Kippen führt:

$$\mathbf{N}_t = \sum \mathbf{d}_i \times \mathbf{f}_{t_i} = \begin{pmatrix} * \\ * \\ 0 \end{pmatrix} \times \begin{pmatrix} * \\ * \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ * \end{pmatrix}.$$

### ZMP als Stabilitätsindikator

Der Roboter kann nur dann kippen, wenn der ZMP am Rand der konvexen Hülle der Bodenkontaktepunkte liegt. Ziel einer auf dem ZMP basierenden Stabilitätsregelung muss also sein, den ZMP strikt im Inneren der konvexen Hülle der Bodenkontaktepunkte zu halten.

### Bemerkungen

Die Bezeichnung „Zero-Moment-Point“ ist eigentlich nicht ganz korrekt, da am ZMP immer noch ein Drehmoment auftritt; jedoch verschwinden die Komponenten, die ein Kippen des Roboters verursachen können. In der Literatur findet man mehrere verschiedene Definitionen des Zero-Moment-Point, die aber letztendlich alle das gleiche bedeuten. Neben dem ZMP gibt es mehrere weitere Stabilitätsindikatoren, die zur Stabilitätsregelung eingesetzt werden können.

### Hinweis zur Bestimmung des ZMP

Der ZMP kann über die Messung der  $f_{n_i,z}$  mit Kraftsensoren an den Füßen bestimmt werden.

Der ZMP kann aber auch direkt aus den Zustandsvariablen  $\mathbf{q}$  des Roboters und deren zeitlichen Ableitungen berechnet werden. Das ist besonders dann hilfreich, wenn keine Kontaktkraftsensoren, aber Winkel-, Geschwindigkeits- und Beschleunigungssensoren am Roboter zur Verfügung stehen.

Dazu wird das Moment  $\mathbf{N}_E$ , das bezüglich eines beliebigen mit dem Roboter verbundenen Punktes  $\mathbf{r}_E$  von den Reaktionskräften und Reaktionsmomenten erzeugt wird, mittels Impuls- und Drallsatz wie folgt berechnet (die Summe der äußeren Momente = der zeitlichen Änderung der Impulsmomente)

$$\mathbf{N}_E = \dot{\mathbf{L}}_c + m_R(\mathbf{r}_c - \mathbf{r}_E) \times (\ddot{\mathbf{r}}_c - \mathbf{g}). \quad (5.25)$$

Dabei steht

- $\mathbf{L}_c = {}^c I \omega$  für den Drehimpuls um den Gesamtmassenschwerpunkt, der bei humanoiden Robotern ähnlich wie bei Menschen in der Regel auf Hüfthöhe liegt,

- $\mathbf{r}_c = (r_{c,x}, r_{c,y}, r_{c,z})^T$  für den Positionsvektor des Massenschwerpunkts des gesamten Roboters,
- $\mathbf{g}$  für den Erdbeschleunigungsvektor (wobei im Folgenden die Gravitationskraft in Richtung der  $z$ -Achse wirken soll) und
- $m_R$  für die Gesamtmasse des Roboters.

Der Positionsvektor des ZMP kann aus Gleichung (5.25) mit Hilfe der Forderung  $N_{E,x} = N_{E,y} = 0$  berechnet werden. Zunächst erhält man für (5.25)

$$\begin{pmatrix} 0 \\ 0 \\ N_{E,z} \end{pmatrix} = \begin{pmatrix} \dot{L}_{c,x} \\ \dot{L}_{c,y} \\ \dot{L}_{c,z} \end{pmatrix} + m_R \begin{pmatrix} r_{c,x} - r_{E,x} \\ r_{c,y} - r_{E,y} \\ r_{c,z} - r_{E,z} \end{pmatrix} \times \begin{pmatrix} \ddot{r}_{c,x} \\ \ddot{r}_{c,y} \\ \ddot{r}_{c,z} + g \end{pmatrix}$$

und nach Ausmultiplizieren des Kreuzproduktes

$$\begin{pmatrix} 0 \\ 0 \\ N_{E,z} \end{pmatrix} = \begin{pmatrix} \dot{L}_{c,x} \\ \dot{L}_{c,y} \\ \dot{L}_{c,z} \end{pmatrix} + m_R \begin{pmatrix} (r_{c,y} - r_{E,y})(\ddot{r}_{c,z} + g) - \dot{r}_{c,y}(r_{c,z} - r_{E,z}) \\ (r_{c,z} - r_{E,z})\ddot{r}_{c,x} - (\ddot{r}_{c,z} + g)(r_{c,x} - r_{E,x}) \\ * \end{pmatrix}.$$

Auflösen der ersten Komponente nach  $r_{E,y}$  und der zweiten Komponente nach  $r_{E,x}$  liefert die gesuchten Koordinaten des ZMP

$$\begin{aligned} r_{E,x} &= \frac{-\dot{L}_{c,y} + m_R r_{c,x}(\ddot{r}_{c,z} + g) - m_R(r_{c,z} - r_{E,z})\ddot{r}_{c,x}}{m_R(\ddot{r}_{c,z} + g)} \\ r_{E,y} &= \frac{\dot{L}_{c,x} + m_R r_{c,y}(\ddot{r}_{c,z} + g) - m_R(r_{c,z} - r_{E,z})\ddot{r}_{c,y}}{m_R(\ddot{r}_{c,z} + g)}. \end{aligned}$$

## 5.9.2 Invertiertes Pendel

Um einen Stabilitätsindikator wie den ZMP sinnvoll zu nutzen, müssen die Zustandsvariablen  $\mathbf{q}$  und die Steuervariablen  $\mathbf{u}$  des Roboters entsprechend dem Wert des Stabilitätsindikators angepasst werden. Probleme dabei sind, dass den wenigen Werten des Stabilitätsindikators (zwei im Falle des ZMP) sehr viele Zustands- und Steuergrößen gegen "überstehen und dass jede einzelne dieser Zustände oder Steuerungen den Wert des Stabilitätsindikators beeinflussen kann. Zudem ist ein Gesamtsystemmodell des Roboters sehr groß, was eine erforderliche schnelle Reaktion aufgrund hoher Rechenzeiten auf den für die Regelung typischerweise eingesetzten, eingebetteten Computern (z.B. Microcontroller) zur Behandlung eines vollständig dynamischen Modells schwierig macht.

### Vereinfachtes Modell

Ein möglicher Ansatz, um diese Schwierigkeiten zu lösen, ist, den Roboter nicht im Detail zu modellieren, sondern die Modellierung auf die wesentlichen, die Charakteristik des Gesamtsystems am stärksten beeinflussenden, Teile zu beschränken und das vereinfachte System als Ersatzsystem zu betrachten.

Naherungsweise gelten für laufende Roboter die folgenden vereinfachten Gleichungen:

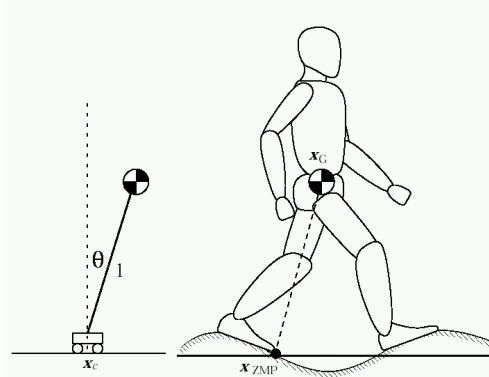
- Bewegungsgleichung:  $m_R \ddot{\mathbf{r}}_c = m_R \mathbf{g} + \mathbf{f}_c$

- Geometrische Beschränkung:  $\mathbf{f}_c = k(\mathbf{r}_c - \mathbf{r}_{ZMP})$

Dabei ist  $m_R$  die Gesamtmasse des Roboters,  $\mathbf{r}_c = (r_{c,x}, r_{c,y}, r_{c,z})^T$  die Lage des Schwerpunkts,  $\mathbf{r}_{ZMP} = (r_{ZMP,x}, r_{ZMP,y}, r_{ZMP,z})^T$  die Position des Zero-Moment-Points,  $\mathbf{g}$  der Erdbeschleunigungsvektor,  $\mathbf{f}_c$  die gesamte externe Kraft, die auf den Roboter am Schwerpunkt wirkt und  $k$  eine positive Konstante, die zu gegebenem  $\mathbf{r}_{ZMP}$  eindeutig bestimmt werden kann. Damit erhält man dann

$$\begin{aligned}\ddot{r}_{c,x} &= \omega_m^2(r_{c,x} - r_{ZMP,x}) \\ \ddot{r}_{c,y} &= \omega_m^2(r_{c,y} - r_{ZMP,y}),\end{aligned}$$

wobei  $\omega_m$  definiert ist als  $\omega_m = \sqrt{\frac{\ddot{r}_{c,z} + g}{r_{c,z} - r_{ZMP,z}}}$



Diese Gleichungen sind sehr ähnlich zu denen, die beim „invertierten Pendel“ auftreten.

### Invertiertes Pendel

Auf ein invertiertes Pendel (engl. inverted pendulum) der Masse  $m_p$  wirkt zum einen die Erdbeschleunigung. Die kinetische und potentielle Energie des Pendels unter der Erdgravitation ergeben sich zu

$$\begin{aligned}K &= 1/2m_p l^2 \dot{\theta}^2 \\ P &= m_p g l \cos \theta + P_{\text{ref},1}\end{aligned}$$

und die Bewegungsgleichung nach Lagrange lautet damit

$$\begin{aligned}0 &= \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{\theta}} \right) - \frac{\partial K}{\partial \theta} + \frac{\partial P}{\partial \theta} \\ &= \frac{d}{dt} (m_p l^2 \dot{\theta}) - 0 - m_p g l \sin \theta.\end{aligned}$$

Damit gilt für die durch die Gravitation bedingte Winkelbeschleunigung  $\ddot{\theta}_g$  des invertierten Pendels

$$\ddot{\theta}_g = (g/l) \sin \theta.$$

Zum anderen wirkt die Beschleunigung des Wagens, auf dem das Pendel montiert ist, in Vorförtsrichtung. Die Position, Geschwindigkeit und Beschleunigung des Wagens mit Masse  $m_w$  werden mit  $x_w, \dot{x}_w, \ddot{x}_w$  bezeichnet. Hier kann ebenso wie für die gravitationsbedingte Beschleunigung die Bewegungsgleichung aufgestellt werden, diesmal unter der Annahme, dass die Erdbeschleunigung nicht wirkt. Der Massenpunkt des Pendels hat die Koordinaten  $(x_w + l \sin \theta, 0, l \cos \theta)^T$  und damit die Geschwindigkeit  $(\dot{x}_w + \dot{\theta} l \cos \theta, 0, -\dot{\theta} l \sin \theta)^T$ .

Damit gilt

$$\begin{aligned} P &= 0 + P_{\text{ref},2} \\ K &= 1/2m_w \dot{x}_w^2 + 1/2m_p (\dot{x}_w^2 + 2\dot{x}_w \dot{\theta} l \cos \theta + \dot{\theta}^2 l^2 \cos^2 \theta + \dot{\theta}^2 l^2 \sin^2 \theta) \\ &= 1/2(m_w + m_p) \dot{x}_w^2 + 1/2m_p \dot{\theta}^2 l^2 + m_p \dot{x}_w \dot{\theta} l \cos \theta. \end{aligned}$$

Nach Lagrange folgt weiter

$$\begin{aligned} 0 &= \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{\theta}} \right) - \frac{\partial K}{\partial \theta} + \frac{\partial P}{\partial \theta} \\ &= \frac{d}{dt} (m_p l^2 \dot{\theta} + m_p \dot{x}_w l \cos \theta) + m \dot{x}_w \dot{\theta} l \sin \theta + 0. \end{aligned}$$

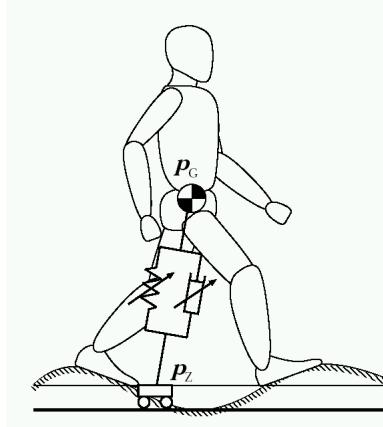
Damit ergibt sich schließlich als Beschleunigung  $\ddot{\theta}_w$  des Pendels, die auf die Beschleunigung des Wagens zurückzuführen ist

$$\ddot{\theta}_w = -(\ddot{x}/l) \cos \theta.$$

Zusammengefasst ergeben diese beiden Beschleunigungen die Bewegungsgleichung des invertierten Pendels als Näherung um die Nullauslenkung (d.h. aufrechte Haltung)

$$l \ddot{\theta} - g \theta = -\ddot{x}.$$

Dieses vereinfachte Modell eines zweibeinigen Roboters kann nun verwendet werden, um grundlegende Entscheidungen für das Vermeiden von Kippbewegungen zu treffen, ohne das komplette Gesamtmodell zu berücksichtigen, dessen Betrachtung häufig noch zu rechenintensiv wäre.



## Bemerkungen

- Wie ein solches System geregelt wird, d.h. wie mögliche Steuerungen (in diesem Fall die Beschleunigung des Wagens) gewählt werden müssen, um die Stabilität des Systems sicher zu stellen, wird in Robotik 2 in Kapitel 6 (Roboterregelungen) vorgestellt werden.
- Neben dem einfachen invertierten Pendel werden häufig auch erweiterte Modelle invertierter Pendel eingesetzt, beispielsweise mit nicht starrem Pendel sondern einem Pendel mit Feder-Dämpfer-Modul mit variabler Impedanz, das eine bessere Modellierung der Eigenschaften des Roboters beim Landen und Abheben des Fusses ermöglicht.

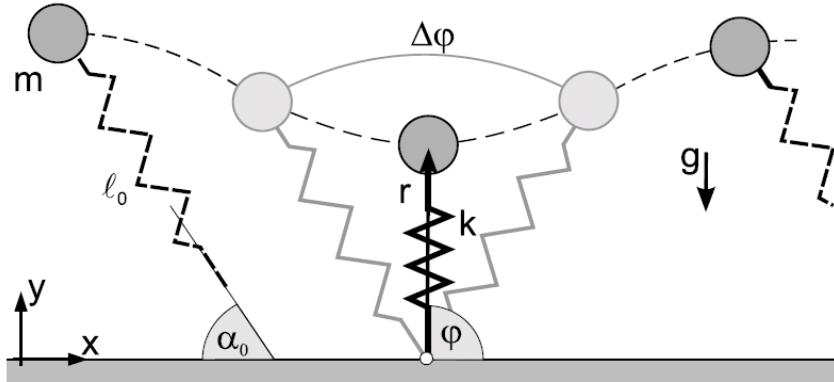


Figure 5.8: Einbeiniger Federhüpfer (aus H. Geyer: Simple Models of Legged Locomotion based on Compliant Limb Behavior, Dissertation, Uni Jena, 2005).

- Bereits bei schnellerem Gehen des Menschen liefert der ZMP keine nützliche Information mehr, da dieser dann häufig am Rande der konvexen Hülle der tragenden Bodenkontaktepunkte liegt. Für schnelles Gehen, Joggen oder Rennen müssen effektiv einsetzbare Kriterien für dynamische Stabilität erst noch entwickelt werden.

### 5.9.3 Erweiterte Modelle des menschlichen Gehens und Rennens

In konventionellen Industrierobotern ist Elastizität ein unerwünschter Effekt, der bereits in der Konstruktion so weit wie möglich vermieden werden soll. In biologischen Systemen, so auch im menschlichen Bewegungsapparat der Beine, ist Elastizität ein wesentliches Merkmal. Besonders deutlich wird dies bei der Untersuchung einfacher Modelle der menschlichen Beine, mit denen jedoch gemessene menschliche Bewegungen für einzelne Gangarten wie Gehen oder Rennen erstaunlich gut beschrieben werden können.

Die im vorherigen Abschnitt untersuchte Modellierung basierend auf Ansätzen vom Typ des invertierten Pendels liefert bei der Analyse menschlicher Gangarten brauchbare Ergebnisse nur für langsames Gehen. Beim Rennen dagegen müssen andere Modelle verwendet werden. Ein solches Modell soll im Folgenden vorgestellt werden. Anschliessend soll die Frage beantwortet werden, ob zur Untersuchung von Gehen und Rennen wirklich zwei grundlegend verschiedene Modelle verwendet werden müssen.

#### 5.9.3.1 Feder-Masse-Modell für menschliches Rennen

Das menschliche Bein wird modelliert als Punktmasse  $m$ , die sich auf einem Federpendel der Ruhelänge  $l_0$  und Federkonstante  $k$  hüpfend fortbewegt. Der Zustand des Hüpfers wird über Polarkoordinaten  $(r, \varphi)$  beschrieben, vgl. Abbildung 5.8. Für Flugphasen bewegt sich die Punktmasse mit trivialer Bewegungsgleichung auf einer ballistischen Flugbahn, während für die Kontaktphase eine genauere Herleitung der Bewegungsgleichung notwendig ist. Dafür wird die Lagrangefunktion  $\mathcal{L}$  aufgestellt. Die Feder ist der einzige Unterschied zum Ansatz des invertierten Pendels. Sie wirkt sich in zwei Anteilen auf die Bewegungsgleichung aus: in der kinetischen Energie durch den Summanden  $1/2mr^2$  (veränderliche Beinlänge), in der potentiellen Energie durch den Summanden  $1/2k(l_0 - r)^2$  (Energie der ausgelenkten Feder).

Als Anteil der kinetischen Energie ergibt sich

$$K = \frac{m}{2} (\dot{r}^2 + r^2 \dot{\varphi}^2).$$

Die potentielle Energie setzt sich aus der Lageenergie und der Federenergie zusammen

$$P = mgr \sin \varphi + \frac{k}{2} (l_0 - r)^2 + P_{ref}.$$

Damit ergibt sich die Lagrangefunktion zu

$$\mathcal{L}(r, \varphi) = K - P = \frac{m}{2} (\dot{r}^2 + r^2 \dot{\varphi}^2) - mgr \sin \varphi - \frac{k}{2} (l_0 - r)^2 + P_{ref}.$$

Die Bewegungsgleichungen für die beiden Zustände des Systems  $r$  und  $\varphi$  berechnen sich nun nach den bekannten Formeln.

### Bewegungsgleichung für $r$

$$\begin{aligned} & \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{r}} \right) - \frac{\partial \mathcal{L}}{\partial r} = 0 \\ \Leftrightarrow & \frac{d}{dt} (m\dot{r}) - m\dot{\varphi}^2 r + m g \sin \varphi - k(l_0 - r) = 0 \\ \Leftrightarrow & m\ddot{r} = k(l_0 - r) + mr\dot{\varphi}^2 - mg \sin \varphi \end{aligned}$$

### Bewegungsgleichung für $\varphi$

$$\begin{aligned} & \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\varphi}} \right) - \frac{\partial \mathcal{L}}{\partial \varphi} = 0 \\ \Leftrightarrow & \frac{d}{dt} (mr^2 \dot{\varphi}) + mgr \cos \varphi = 0 \\ \Leftrightarrow & mr^2 \ddot{\varphi} + 2mr\dot{r}\dot{\varphi} + mgr \cos \varphi = 0 \\ \Leftrightarrow & r\ddot{\varphi} = -2\dot{r}\dot{\varphi} - g \cos \varphi. \end{aligned}$$

Diese Gleichungen können nun numerisch integriert werden (vgl. Abschnitt 5.5.3), um den Bewegungsverlauf des Modells zu simulieren. Als Parameter für verschiedene Laufbewegungen erweist sich neben der Federsteifigkeit auch der Anstellwinkel der Feder beim Landen sowie die Gesamtenergie des Systems als sinnvoll. Stabilitätsuntersuchungen des Modells zeigen, dass das System „selbststabilisierend“ arbeitet. Die „Übereinstimmung“ der Fußkontaktekraftverläufe von menschlichen Rennbewegungen (vgl. Abb. 5.9 rechts) mit den durch Simulation dieses einfachen Modells erhältlichen ist erstaunlich gut.

#### 5.9.3.2 Feder-Masse-Modell für menschliches Gehen

Für langsames Gehen ist das invertierte Pendel ein oft verwendetes Modell für die menschlichen Beine. Der Mensch kann aber gehen und rennen mit den gleichen Beinen, daher stellt sich die Frage, ob wirklich zwei grundlegend unterschiedliche Modelle für Gehen und Rennen notwendig sind. Als naheliegendes Beinmodell für langsames Gehen erweist sich ein aus zwei Federn bestehendes, analog zum einbeinigen Hüpfer aufgebautes Modell (s. Abbildung 5.10). Tatsächlich zeigt sich bei genauerer Untersuchung des Modells, dass es die Bodenkontaktekräfte beim menschlichen Gehen (Abb. 5.9, links) gut nachbilden kann (Geyer, 2005).

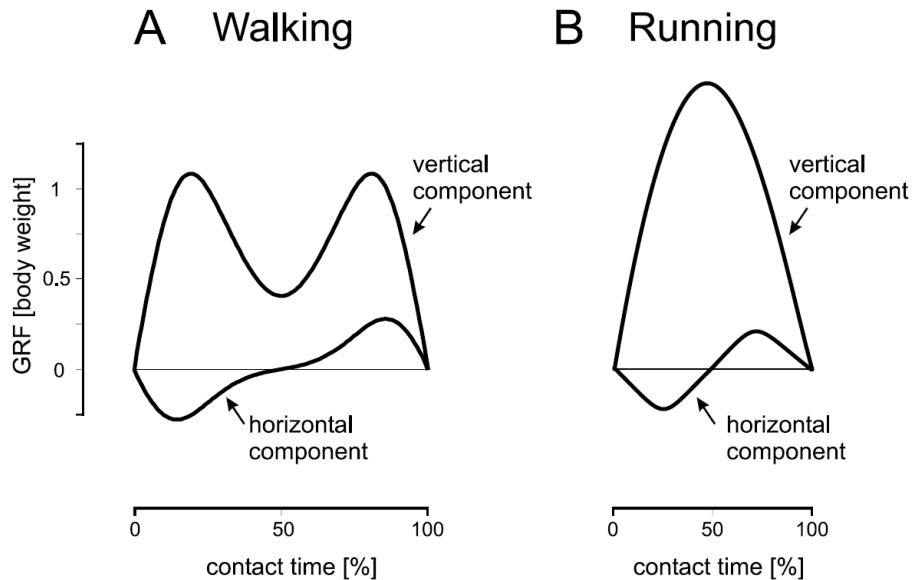


Figure 5.9: Typische Bodenreaktionskraftverläufe für menschliches Gehen (links) und Rennen (rechts) (H. Geyer, Dissertation, Uni Jena, 2005). Mit den vorgestellten Modellen können diese Verläufe sehr gut nachgebildet werden.

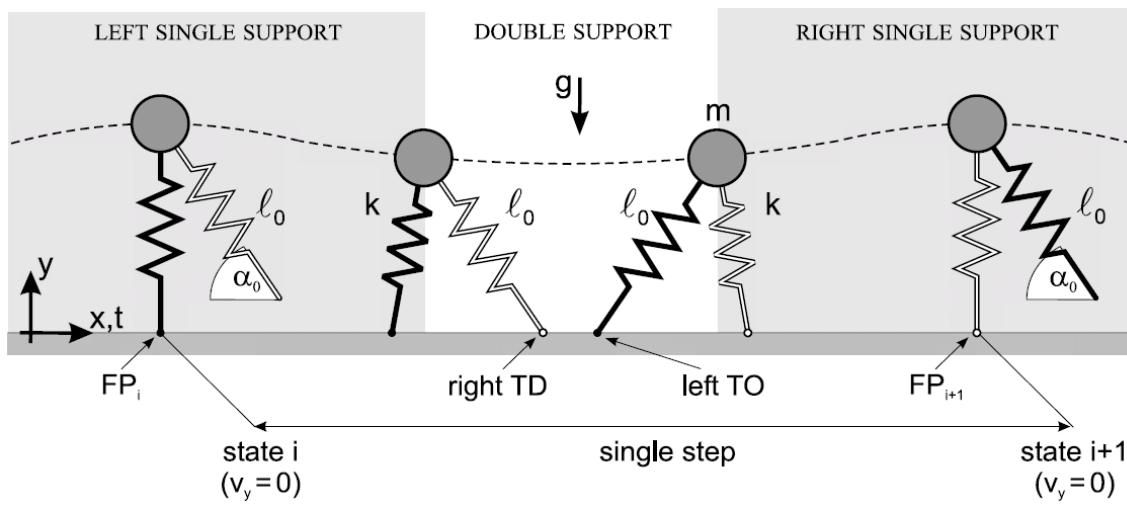


Figure 5.10: Zweibeiniger Federgeher (aus H. Geyer: Simple Models of Legged Locomotion based on Compliant Limb Behavior, Dissertation, Uni Jena, 2005).

## 5.9.4 Globale Stabilitätsbegriffe

Stabilitätskriterien wie der projizierte Massenschwerpunkt oder der ZMP liefern eine Aussage über die posturale Stabilität jeweils nur für einen einzelnen Zustand. Um die Stabilität z.B. einer Laufbewegung sicher zu stellen, ist das jeweilige Kriterium an jedem einzelnen Zeitpunkt der Bewegung auszuwerten. Im Folgenden werden einige Stabilitätsbegriffe vorgestellt, die periodische, dynamische Bewegungsmuster wie Gehen oder Laufen als Ganzes untersuchen.

Dazu sei im Folgenden stets ein allgemeines, dynamisches System  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ , ein Zustandsvektor im Phasenraum  $\mathbf{x} \in \mathbb{R}^n$  und  $\{t_k\}$  eine Menge von Zeitpunkten mit  $\lim_{k \rightarrow \infty} t_k = \infty$  gegeben.

**Definition:** Eine positive Grenzmenge (engl. positive limit set)  $\Omega$  einer beschränkten Trajektorie  $\mathbf{x}(t)$ , d.h.  $\exists \mu \in \mathbb{R}, \|\mathbf{x}(t)\| < \mu \forall t > 0$ , ist wie folgt definiert:

$$\Omega = \{\mathbf{p} \in \mathbb{R}^n \mid \forall \epsilon > 0 : \exists \{t_k\}, \text{ so dass } \|\mathbf{p} - \mathbf{x}(t_k)\| < \epsilon \forall k \in \mathbb{N}\}.$$

$\Omega$  ist die Menge aller Zustände, in deren beliebig kleiner Umgebung das System immer wieder Zustände annimmt. Damit ist klar, dass eine Laufbewegung, die zum Fallen führt, nur eine triviale positive Grenzmenge haben kann.

**Definition:** Eine Trajektorie heißt stabiler Orbit  $\Omega$ , wenn  $\forall \epsilon > 0 \exists \delta > 0$ , so dass gilt:

$$\|\mathbf{x}_0 - \Omega\| < \delta \Rightarrow \inf_{\mathbf{p} \in \Omega} \|\mathbf{x}(t) - \mathbf{p}\| < \epsilon \forall t > t_0$$

Alle Trajektorien, die in der Nähe eines stabilen Orbits  $\Omega$  starten, bleiben nahe bei der Trajektorie. Für die Laufbewegung heisst das, dass kleine Störungen ausgeglichen werden können.

Falls außerdem alle Trajektorien aus der Umgebung von  $\Omega$  sich für  $t \rightarrow \infty$  an  $\Omega$  annähern, heißt  $\Omega$  anziehend oder attraktiv (engl. attractive).

**Definition:** Eine Trajektorie  $\Omega$  heißt asymptotisch stabiler Orbit, falls sie ein stabiler Orbit ist und darüber hinaus gilt

$$\|\mathbf{x}_0 - \Omega\| < \delta \Rightarrow \liminf_{t \rightarrow \infty} \inf_{\mathbf{p} \in \Omega} \|\mathbf{x}(t) - \mathbf{p}\| = 0.$$

**Definition:** Die Poincare-Abbildung (Poincare map) ist die Abbildung, die einen Schnittpunkt der Trajektorie mit einer festen (Hyper-)Ebene im Phasenraum auf den nächsten Schnittpunkt der Trajektorie mit der gleichen (Hyper-)Ebene abbildet (vgl. Abb. 5.11).

Eine Untersuchung dieser Abbildung liefert Aufschluss darüber, wie weit zwei aufeinanderfolgende Trajektorien sich unterscheiden. Wenn das Bild der Abbildung nichttrivial beschränkt ist, liegt Stabilität vor.

Zur Untersuchung der periodischen oder orbitalen Stabilität (orbital stability) einer zweibeinigen Laufbewegung betrachtet man die Bewegungsstrajektorie  $\mathbf{x}(t), t_0 \leq t \leq t_f$ , einer Schrittfolge mit zugehöriger Bewegungsdynamik und periodischen Anfangs- und Endbedingungen

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$$

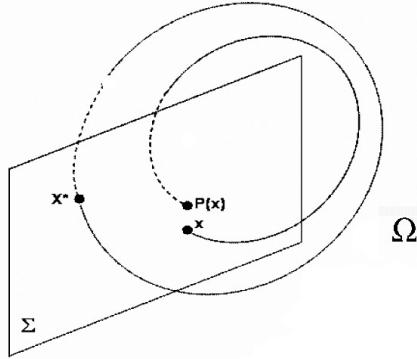


Figure 5.11: Poincare-Abbildung (aus N. Khraief, N.K. M'Sirdi, M.W. Spong: Nearly Passive Dynamic Walking of a Biped Robot, in Proc. European Control Conference, Cambridge, UK, September 1-4, 2003)

$$\mathbf{x}(t_f) = \mathbf{x}(t_0).$$

Wenn die Eigenwerte der Jacobi-Matrix (der in diesem Zusammenhang sogenannten *Monodromiematrix*)  $\partial f / \partial q$  ausgewertet entlang der Bewegungstrajektorie  $\mathbf{x}(t)$  alle kleiner als Null sind, so werden (hinreichend kleine) Abweichungen von der Bewegungstrajektorie kompensiert, so dass die periodische Laufbewegung orbital stabil ist.

## 5.9.5 Unterschiedliche Ansätze zu zweibeinigem (Roboter-)Gehen

## 5.9.6 Einige Referenzen zu Kapitel 5.9

- Geyer, H.: Simple Models of Legged Locomotion Based on Compliant Limb Behavior. Dissertation, Universität Jena, 2005.
- Goswami, A.: Postural stability of biped robots and the foot-rotation indicator (FRI) point. The International Journal of Robotics Research, Vol. 18, No. 6, 1999, S. 523–533.
- Harada, K., Kajita, S., Kaneko, K., Hirukawa, H.: Pushing manipulation by humanoid considering two-kinds of ZMPs, in Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan, September 14-19, 2003, S. 1627–1632.
- Raibert, M.: Legged Robots That Balance. MIT Press 1986/2000.
- Sugihara, T., Nakamura, Y.: Realtime humanoid motion generation through ZMP manipulation based on inverted pendulum control, in Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, May 2002, S. 1404–1409.
- Sugihara, T., Nakamura, Y.: Contact phase invariant control for humanoid robot based on variable impedance inverted pendulum model, in Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan, September 14-19, 2003, S. 51–56.
- Vukobratovic, M., Borovac, B.: Zero-moment point – Thirty five years of its life. International Journal of Humanoid Robotics, Vol. 1, No. 1 (2004) 157-173.

# Chapter 6

## Roboterregelungen

**Hinweis:** Bitte beachten Sie, dass der englische Fachbegriff „control“ im Deutschen je nach Kontext zwei unterschiedliche Bedeutungen hat:

Begriffe:	englisch	—	deutsch
(open loop / feedforward) control		—	Steuerung
(closed loop / feedback) control		—	Regelung

### 6.1 Antriebssysteme

Als Skript zu Kapitel 6.1 Antriebssysteme (dt. auch Aktoren oder Aktuatoren, engl. actuators) werden die Kopien der Folien online bereitgestellt.

### 6.2 Interne Sensoren

Als Skript zu Kapitel 6.2 werden die Kopien der Folien online bereitgestellt.

### 6.3 Lineare (Positions-)Regelung

#### Literatur zu Abschnitt 6.3:

Chapter 9 bei [Craig, 2004], Chapter 7 bei [Spong/Vidyasagar, 1989], Chapter 5 bei [Yoshikawa, 1990], Chapter 11 bei [McKerrow, 1991].

#### 6.3.1 Einführung und Vorbemerkungen

Wir betrachten einen Manipulator als kinematische Kette, die in jedem Gelenk mit Sensoren zur Messung des Gelenkwinkels (bei Drehgelenken) bzw. der Gelenkverschiebung (bei Schubgelenken) ausgestattet ist. Weiterhin nehmen wir an, dass jedes Gelenk mit einem Antrieb ausgestattet ist, mit dem entweder ein Drehmoment (bei einem Drehgelenk) oder eine Kraft (bei einem Schubgelenk) auf das nachfolgende Glied in der kinematischen Kette ausgeübt werden kann.

Das **(Positions-)Regelungsproblem** für einen Roboter besteht nun darin, dass der Roboter eine vorher geplante Bahn oder Trajektorie in kartesischen Weltkoordinaten

$${}^0\mathbf{r}_{nd}(t) \in \mathbb{R}^3, \quad \text{ggf. mit Orientierung } (\alpha_d(t), \beta_d(t), \delta_d(t))^T, \quad 0 \leq t \leq t_f$$

oder in Gelenkkoordinaten

$$\mathbf{q}_d(t) \text{ (oder } \mathbf{q}_{\text{soll}}(t)) \text{ und ggf. } \dot{\mathbf{q}}_d(t), \ddot{\mathbf{q}}_d(t) \in \mathbb{R}^n, \quad 0 \leq t \leq t_f$$

durchführen soll. Die Indizes „d“ stehen für „desired/demanded value“ bzw. „soll“ für „Sollwert“.

Anstelle einer kontinuierlichen Sollwerttrajektorie (engl. set-point trajectory) wird die durchzuführende Bewegung häufig

- in Gelenkkoordinaten als eine Folge aufeinanderfolgender Gelenkpositionen  $\mathbf{q}(t_i)$ , und ggf. Gelenkgeschwindigkeiten und -beschleunigungen  $\dot{\mathbf{q}}(t_i), \ddot{\mathbf{q}}(t_i), i = 1, \dots, n_m$ ,
- bzw. in kartesischen Weltkoordinaten als eine Folge aufeinanderfolgender Positionen  ${}^0\mathbf{r}_n(t_i), i = 1, \dots, n_m$ , und Orientierungen des Endeffektors, z.B.  $(\alpha(t_i), \beta(t_i), \delta(t_i))$ ,

angegeben.

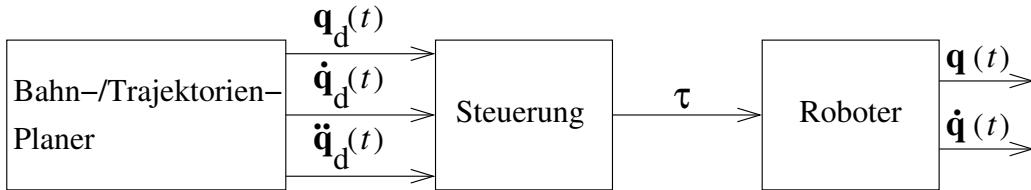


Figure 6.1: Abstraktes Blockschaltbild einer Robotersteuerung auf Gelenkebene (ohne Rückführung von Sensorwerten)

Der Zusammenhang zwischen Bahn- bzw. Trajektorienplanung (siehe Kapitel 7) und dem physikalischen Roboter ist in Abbildung 6.1 dargestellt. Die Antriebe des Roboters erwarten als Eingaben beispielsweise die Gelenkkräfte oder Drehmomente  $\tau$  oder die Motoreingangsspannungen.

Als „Steuerungsmethode“ in Abbildung 6.1 bietet sich zunächst die Verwendung des inversen Dynamikmodells aus Kapitel 5

$$\text{INV DYN: } \tau = M(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$$

an zur Berechnung der von den Antrieben aufzubringenden Drehmomente bzw. Kräfte in den Gelenken.

Jedoch bereits kleine, unvermeidbare Abweichungen zwischen Robotermodell und physikalischer Realität, spezielle Eigenschaften der Antriebe (z.B. sind die Ansteuerungen nicht kontinuierlich sondern typischerweise in Taktraten von Millisekunden möglich) sowie nicht vorhersehbare äußere Einwirkungen führen dazu, dass trotz bester Robotermodellierung und Trajektorienplanung im Laufe der Durchführung der Bewegung die tatsächliche Bahn bzw. Trajektorie  $\mathbf{q}(t)$  bzw.  $\mathbf{q}_{\text{ist}}(t)$  immer stärker von der vorgegebenen Sollbahn  $\mathbf{q}_d(t)$  abweicht. Je besser die Modellierung und Trajektorienplanung umso geringer die resultierenden Abweichungen, dennoch lassen diese sich nicht völlig vermeiden.

Um Abweichungen der Ist-Bewegung von der Soll-Bewegung zu vermeiden, werden mit Hilfe interner Sensoren z.B. die tatsächlichen Gelenkvariablen  $\mathbf{q}(t)$  (bzw.  $\mathbf{q}_{\text{ist}}(t)$ ) und ggf. die tatsächlichen Gelenkgeschwindigkeiten  $\dot{\mathbf{q}}(t)$  (bzw.  $\dot{\mathbf{q}}_{\text{ist}}(t)$ ) gemessen. Die Aufgabe des Regelungssystems ist es (Abbildung 6.2), mit Hilfe der vorgegebenen Soll- und den

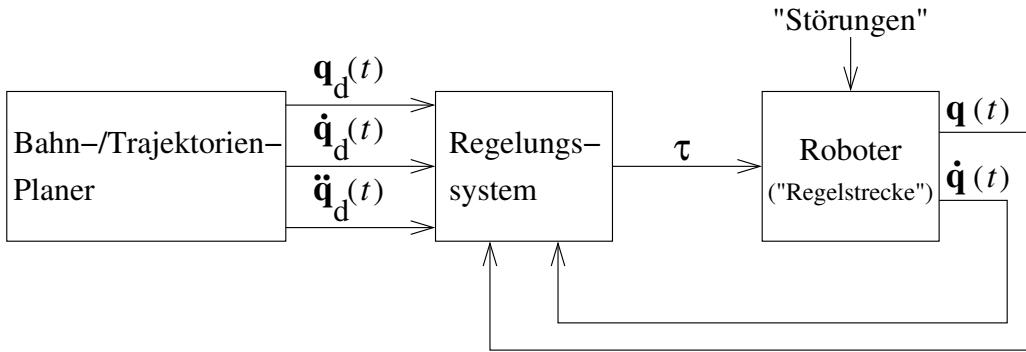


Figure 6.2: Abstraktes Blockschaltbild eines Roboterregelungssystems auf Gelenkebene

gemessenen Ist-Werten die Eingangswerte an die Roboterantriebe so zu bestimmen, dass die **Regeldifferenz**<sup>1</sup>

$$\begin{aligned} e &= q_d - q \\ \dot{e} &= \dot{q}_d - \dot{q} \end{aligned}$$

(Buchstabe „e“ für „error“) reduziert wird.

Unsere Aufgabe ist es nun, ein Regelungssystem zu entwerfen, das gewisse Gütekriterien erfüllt. Das elementarste Gütekriterium ist die **Stabilität** des Gesamtsystems. Es gibt mehrere Definitionen der Stabilität eines Regelkreises. Für die Zwecke hier betrachten wir ein System als „stabil“, wenn die Regelabweichung bei Ausführung unterschiedlicher Sollwerttrajektorien und unter Einwirkung „moderer“ Störungen „klein“ bleibt.

In einem schlecht entworfenen Regelungssystem kann es zu Instabilität kommen, so dass Regelabweichungen verstärkt statt reduziert werden. Bei Entwurf einer Regelungsmethode ist daher als erstes nachzuweisen, dass der resultierende Regelkreis stabil ist, und als zweites, dass die Systemperformanz befriedigend ist. Der Nachweis der Stabilität einer Regelungsmethode wird je nach Verfahren und Anwendung unterschiedlich erbracht. Beispielsweise können strikte mathematische Beweise unter gewissen Annahmen auf spezielle Modelle angewendet werden oder empirische Resultate werden durch Experimente oder Simulationen ermittelt.

In Abbildung 6.2 entsprechen alle Linien mit Pfeilen  $n$ -dimensionalen Signalen. Die Aufgabe der Bewegungsregelung eines  $n$ -gelenkigen Roboters ist folglich ein „multi-input multi-output (MIMO)“-Regelungsproblem (Mehrgrößenregelung, multivariable control system). Wir werden das Bewegungsregelungsproblem zunächst vereinfachen, indem wir  $n$  unabhängige „single-input single-output (SISO)“-Regelungsprobleme betrachten (independent joint control). Dieser Ansatz wird z.B. bei Industrierobotern oder Unterhaltsrobotern am häufigsten verwendet.

**Bemerkung:** Die Güte der resultierenden Roboterbewegung hängt nicht allein von den eingesetzten Regelungsverfahren sondern auch vom mechanischen Roboterdesign und den Eigenschaften der verwendeten Antriebe ab („good software cannot overcome poor electro-mechanical design“). Daher werden für besonders anspruchsvolle Roboteranwendungen (z.B. für den DLR-KUKA Leichtbauroboterarm, die humanoiden Roboter von Honda, Kawada, KAIST, ROBOTIS und PAL Robotics) neue, besonders kompakte, leistungsfähige und spielfreie Drehgelenksantriebe entwickelt.

<sup>1</sup>Nach DIN 19226 wird  $q - q_d$  als **Regelabweichung** und  $q_d - q$  als **Regeldifferenz** bezeichnet.

### 6.3.2 Lineare Systemdynamik

Das zeitveränderliche, dynamische Verhalten des Roboters in Abbildung 6.2 wird näherungsweise durch das nichtlineare inverse Dynamikmodell

$$M(\mathbf{q}) \cdot \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) + F(\mathbf{q}, \dot{\mathbf{q}}) = \tau$$

beschrieben. Zunächst betrachten wir den Spezialfall eines einfachen mechanischen Systems (Abbildung 6.3).

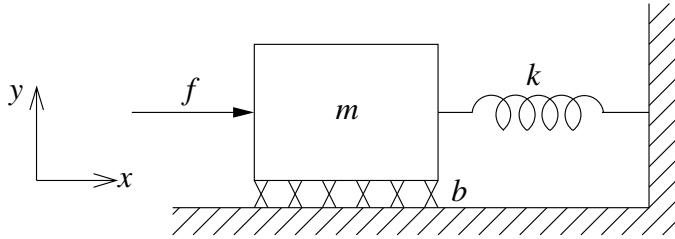


Figure 6.3: Feder-Masse-System mit Reibung und Antriebskraft

Eine Masse  $m > 0$  ist mit einer Feder der Steifigkeit  $k > 0$  verbunden und bewegt sich entlang der  $x$ -Achse, wobei eine Reibungskraft mit Koeffizient  $b > 0$  proportional zur Geschwindigkeit der Masse wirkt. Über einen Antriebsmechanismus kann eine Antriebs- oder Bremskraft  $f$  erzeugt werden. Die Bewegungsdifferentialgleichung des Feder-Masse-Systems lautet somit

$$m \ddot{x}(t) + b \dot{x}(t) + k x(t) = f(t).$$

Dies ist eine lineare Differentialgleichung zweiter Ordnung mit konstanten Koeffizienten. Im folgenden wird zunächst die Bewegung ohne Antriebskraft (d.h.  $f = 0$ ) untersucht. Die Ruhelage des Systems sei bei  $x = 0$ .<sup>2</sup>

Wie bei der Simulation dynamischer Roboterbewegungen in Abschnitt 5.5.3 kann bei gegebener Anfangsposition  $x(0)$  und Anfangsgeschwindigkeit  $\dot{x}(0)$  der Masse (und bei gegebenem Antriebskraftverlauf  $f(t)$ ,  $0 \leq t \leq t_f$ ) der zeitliche Verlauf der Bewegung der Masse  $x(t)$ ,  $0 \leq t \leq t_f$ , eindeutig berechnet werden.

Was für ein Bewegungsverhalten kann intuitiv erwartet werden?

- Bei einer sehr starken Feder (d.h.  $k$  „groß“) und sehr kleiner Reibungskraft (d.h.  $b$  „klein“) wird die Masse mehrere Male hin und her oszillieren, bis sie nach einer Anfangsauslenkung in die Ruhelage bei  $x = 0$  zurückkehrt.
- Andererseits wird bei einer sehr schwachen Feder (d.h.  $k$  „klein“) und sehr großer Reibungskraft (d.h.  $b$  „groß“) die Masse nach einer Anfangsauslenkung nur langsam und schwerfällig in die Ruhelage bei  $x = 0$  zurückkehren.

<sup>2</sup>Die Ruhelage  $\mathbf{x}_{\text{stat}}$  eines allgemeinen Differentialgleichungssystems erster Ordnung  $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$  erhält man als Lösung der stationären Gleichung  $\mathbf{0} = \mathbf{f}(\mathbf{x}_{\text{stat}})$ . Für eine Reihe mechanischer Systeme (wie für das Feder-Masse-System) gilt, dass  $\dot{\mathbf{x}}(t) \rightarrow 0$  und damit  $\mathbf{x}(t) \rightarrow \mathbf{x}_{\text{stat}}$  für  $t \rightarrow \infty$ , falls keine weiteren Antriebs- oder Störungskräfte auf das System wirken.

Die Lösung einer linearen Differentialgleichung mit konstanten Koeffizienten kann explizit berechnet werden. Zur Anwendung der Berechnungsformeln wird die Differentialgleichung zunächst auf „Normalform“ gebracht:

$$\begin{aligned} m\ddot{x}(t) + b\dot{x}(t) + kx(t) &= 0 \\ \Leftrightarrow \quad \ddot{x}(t) + \frac{b}{m}\dot{x}(t) + \frac{k}{m}x(t) &= 0. \end{aligned}$$

Durch Einsetzen des allgemeinen Lösungsansatzes  $x(t) = ce^{\lambda t}$  mit geeigneten Konstanten  $c$  und  $\lambda$  erhält man die charakteristische Gleichung

$$\lambda^2 + \frac{b}{m}\lambda + \frac{k}{m} = 0$$

deren Nullstellen sich ergeben als

$$\lambda_{1,2} = -\frac{b}{2m} \pm \frac{\sqrt{b^2 - 4mk}}{2m}.$$

Die Lage der Nullstellen  $\lambda_1, \lambda_2$  („Pole“) bestimmt das Lösungsverhalten und damit das resultierende Bewegungsverhalten der Masse:

- (i)  $\lambda_{1,2}$  einfache, reelle Nullstellen, wenn

$$\begin{aligned} b^2 - 4mk &> 0 \quad \Leftrightarrow \quad b^2 > 4mk \\ \Rightarrow \quad \lambda_{1,2} &< 0. \end{aligned}$$

Im Fall „hoher Reibungskraft“ lautet somit die allgemeine Lösung der Bewegungsdifferentialgleichung

$$x(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}.$$

Die unbekannten, reellen Konstanten  $c_1, c_2$  werden durch Vorgabe von Anfangswerten für  $x(0)$  und  $\dot{x}(0)$  im Einzelfall bestimmt. Wegen  $\lambda_{1,2} < 0$  gilt

$$x \rightarrow 0 \quad \text{für} \quad t \rightarrow \infty.$$

Dieses Systemverhalten nennt man **überkritisch gedämpft** (overdamped).

- (ii)  $\lambda_{1,2}$  eine doppelte, reelle Nullstelle, wenn

$$\begin{aligned} b^2 - 4mk &= 0 \quad \Leftrightarrow \quad b^2 = 4mk \\ \Rightarrow \quad \lambda_{1,2} &= -\frac{b}{2m} < 0. \end{aligned}$$

Im Fall „ausgewogener Reibung und Federsteifigkeit“ erhält man die schnellstmögliche, nicht oszillierende Systemantwort:

$$x(t) = c_1 e^{\lambda_1 t} + c_2 t e^{\lambda_1 t}.$$

Wegen

$$x \rightarrow 0 \quad \text{für} \quad t \rightarrow \infty$$

ist die Lösung wiederum abklingend.

Dieses Systemverhalten nennt man **kritisch gedämpft** (critically damped).

(iii)  $\lambda_{1,2}$  einfache, komplexe Nullstellen, wenn

$$\begin{aligned} b^2 - 4mk &< 0 \quad \Leftrightarrow \quad b^2 < 4mk \\ \Rightarrow \quad \lambda_{1,2} &= -\frac{b}{2m} \pm i\sqrt{\frac{|b^2 - 4mk|}{2m}} := \lambda_r \pm i\lambda_i. \end{aligned}$$

Im Fall „dominierender Federsteifigkeit“ lautet der Realteil der allgemeinen Lösung der Bewegungsdifferentialgleichung

$$x(t) = c_1 e^{\lambda_r t} \cos(\lambda_i t) + c_2 e^{\lambda_r t} \sin(\lambda_i t).$$

Wegen  $\lambda_r < 0$  gilt zwar auch

$$x \rightarrow 0 \quad \text{für} \quad t \rightarrow \infty$$

allerdings mit einer um die Ruhelage  $x = 0$  oszillierenden Bewegung.

Dieses Systemverhalten nennt man **unterkritisch gedämpft** (underdamped).

Das Lösungsverhalten (ii) (kritische Dämpfung) ist die im allgemeinen wünschenswerte Situation, da hier das Systemverhalten von der Ruhelage verschiedene Anfangsbedingungen durch schnellstmögliche Rückkehr in die Ruhelage (Nominalposition) ausgleicht, ohne dass die Bewegung dabei oszillatorisch ist.

**Bemerkung:** Zur Untersuchung von Regelungen bei linearer Systemdynamik werden in der Regelungstechnik üblicherweise mathematische Techniken wie Laplace-Transformationen zur Definition von Übertragungsfunktionen und der Untersuchung von Polen eingesetzt<sup>3</sup>. Da die Kenntnis dieser Techniken hier nicht vorausgesetzt werden kann, beschränken sich die Untersuchungen hier auf den Zustandsraum und setzen dafür lediglich elementare Kenntnisse bei gewöhnlichen Differentialgleichungen voraus.

### 6.3.3 PD-Regelung linearer Systeme zweiter Ordnung

Möglicherweise ist das im vorangehenden Abschnitt untersuchte natürliche Bewegungsverhalten der Masse bei Auslenkung nicht wie es sein sollte, z.B.

- es ist untermittisch gedämpft und oszillatorisch und sollte besser kritisch gedämpft sein oder
- die Feder fehlt komplett (d.h.  $k = 0$ ), so dass die Masse bei Auslenkung niemals in die Nominalposition  $x = 0$  zurückkehrt.

Mit Hilfe von Sensoren zur Messung der Position  $x$  und Geschwindigkeit  $\dot{x}$ , der Verwendung der Antriebskraft  $f$  als Stellgröße bzw. Steuerung sowie eines Regelungssystems soll nun das natürliche Systemverhalten so modifiziert werden, dass das gewünschte Verhalten resultiert, d.h. Einnahme des Sollwerts für den stationären Zustand

$$x_d = 0.$$

Zu diesem Zweck wird ein **Regelgesetz** (control law) untersucht, das die anzuwendende

---

<sup>3</sup>siehe z.B. R. Isermann: Regelungstechnik I, Einführung in Skriptform, Shaker-Verlag, Aachen, 2002

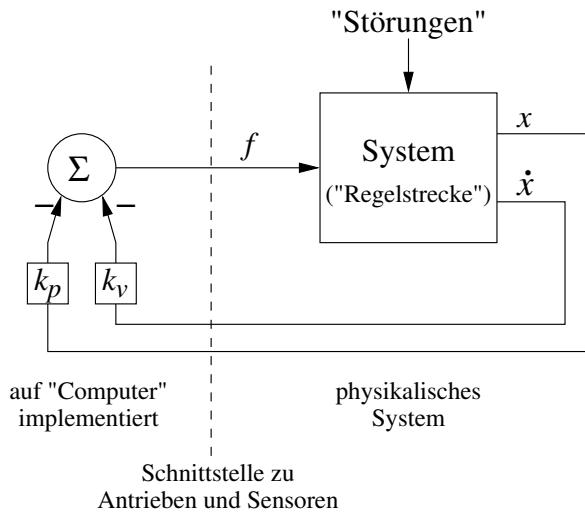


Figure 6.4: PD-Regelungssystem für Feder-Masse-System.

Antriebskraft  $f$  als Funktion der gemessenen Zustandswerte bestimmt:

$$\begin{aligned} f &= -k_p x - k_v \dot{x}, \\ (\text{bzw. } f &= k_p (x_d - x) + k_v (\dot{x}_d - \dot{x}) \quad \text{mit } x_d = 0,) \end{aligned}$$

mit geeigneten, reellen Konstanten  $k_p, k_v$ . Dabei bezeichnet  $-k_p x$  den Proportionalanteil (P) und  $-k_v \dot{x}$  den Differentialanteil (D). Ein Blockschaltbild bzw. Signal-Fluss-Diagramm des solchermaßen geschlossenen Regelkreises ist in Abbildung 6.4 dargestellt. Links von der gestrichelten Linie befindet sich das überlicherweise mit Hilfe eines Rechners (PC, eingebetteter Prozessor o.Ä.) implementierte Regelungssystem und rechts davon das physikalische System (die Regelstrecke). Die erforderlichen Schnittstellen zwischen Rechner, der Ansteuerung der Antriebe und dem Auslesen der Sensorwerte sind nicht explizit dargestellt.

Das vorgeschlagene Regelungssystem versucht die Masse in einer konstanten Normalposition (der Ruhelage  $x = 0$ ) zu halten unabhängig von auf die Bewegung der Masse wirkenden Störeinflüssen (Festwertregler, set-point control). Eine Sollwerttrajektorien-Folgeregelung wird später in Abschnitt 6.3.5 untersucht werden.

Die Anwendung des Regelungsgesetzes auf die Bewegungsdifferentialgleichung des Systems ergibt:

$$\begin{aligned} m \ddot{x}(t) + b \dot{x}(t) + k x(t) &= -k_p x(t) - k_v \dot{x}(t) \\ \Leftrightarrow m \ddot{x}(t) + \underbrace{(b + k_v)}_{:= \hat{b}} \dot{x}(t) + \underbrace{(k + k_p)}_{:= \hat{k}} x(t) &= 0 \end{aligned}$$

Durch geeignete Wahl der Reglerparameter (control gains)  $k_p, k_v$  kann offensichtlich jedes beliebige, der im Abschnitt 6.3.2 ermittelten Systemverhalten erreicht werden, z.B. kritische Dämpfung mit  $\hat{b} = 2\sqrt{m \hat{k}}$ .

**Bemerkung:** Die Verstärkungsfaktoren  $k_p, k_v$  können prinzipiell positiv oder negativ werden je nach Parametern  $m, b, k$  des Ausgangssystems. Wenn allerdings  $\hat{b}$  oder  $\hat{k}$  negativ werden, würde der resultierende Regelkreis instabil werden (siehe explizite Lösung der Systemdifferentialgleichung in diesen Fällen). Das bedeutet, dass dann Abweichungen in Positions- oder Geschwindigkeits-Sollwerten eher verstärkt als reduziert werden würden.

### 6.3.4 Partitionierung des Regelgesetzes durch Feedback-Linearisierung

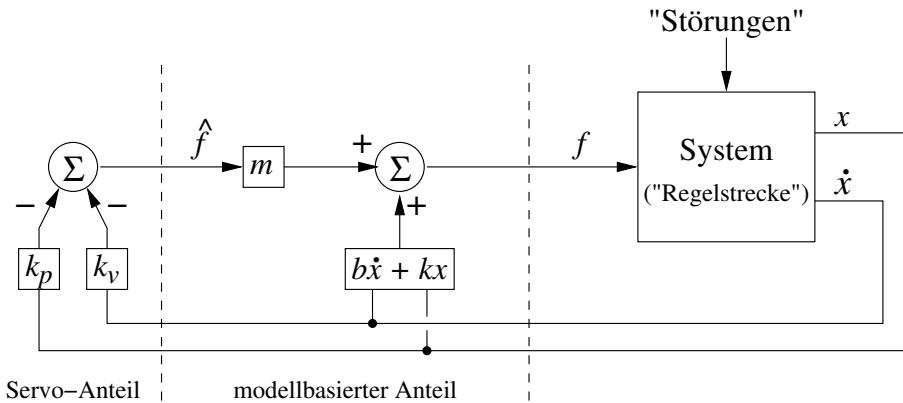


Figure 6.5: Partitionierung des Positions-Reglers für das Feder-Masse-System in modellbasierten und Servo-Anteil.

Zur Vorbereitung von Reglern für komplexere Systeme mit nichtlinearer Systemdynamik wird eine modifizierte Struktur des Reglers aus dem vorangegangenen Abschnitt für das Feder-Masse-Schwinger-System betrachtet. Die Systemdynamik des angetriebenen Systems lautete:

$$m \ddot{x}(t) + b \dot{x}(t) + k x(t) = f(t).$$

Es wird nun eine Aufteilung des Reglers (control law partitioning) in zwei Teile betrachtet:

- a) den („inneren“) **modellbasierten Anteil**, welcher Kenntnis der Systemdynamik ausnutzt,
- b) den („äußerem“) **Servo-Anteil**, welcher Sensormesswerte rückkoppelt zur Anpassung des dynamischen Systemverhaltens.

Durch a) wird die Systemdynamik transformiert bzw. „reduziert“ (und im Falle einer nichtlinearen Systemdynamik „linearisiert“), so dass der Entwurf des Servoanteils b) sehr viel einfacher wird.

Für den modellbasierten Regleranteil wird der Ansatz

$$f = \alpha \hat{f} + \beta$$

betrachtet. Die Größen  $\alpha$  und  $\beta$  sind Funktionen, die möglichst so zu wählen sind, dass die Systemdynamik für eine Einheitsmasse ohne Reibung und Feder resultiert:

$$\ddot{x} = \hat{f}.$$

Ein solches Vorgehen zur Aufteilung des Reglers nennt man „Feedback-Linearisierung“ (feedback linearization).

Aus der Bedingung

$$\underbrace{m \ddot{x}}_{=: \alpha} + \underbrace{b \dot{x} + k x}_{=: \beta} = \alpha \hat{f} + \beta$$

lassen sich  $\alpha$  und  $\beta$  sofort bestimmen.

Den Servo-Anteil bestimmt man als PD-Regler für das System  $\ddot{x} = \hat{f}$ :

$$\hat{f} = -k_p x - k_v \dot{x}.$$

Die Anwendung des Regelungsgesetzes auf die Systemdynamik ergibt (analoges Vorgehen wie im vorangegangenen Abschnitt):

$$\begin{aligned}\ddot{x} &= -k_p x - k_v \dot{x} \\ \Leftrightarrow \ddot{x} + k_p x + k_v \dot{x} &= 0.\end{aligned}$$

Die resultierende charakteristische Gleichung

$$\lambda^2 + k_v \lambda + k_p = 0$$

hat die Nullstellen

$$\lambda_{1,2} = -\frac{k_v}{2} \pm \sqrt{\left(\frac{k_v}{2}\right)^2 - k_p}$$

und man erhält kritische Dämpfung für:

$$\left(\frac{k_v}{2}\right)^2 - k_p = 0 \Rightarrow k_v = 2\sqrt{k_p}.$$

Das zugehörige Blockschaltbild ist in Abbildung 6.5 dargestellt. Dies ist auch ein erstes Beispiel für eine Kaskadenregelung.

### 6.3.5 Sollwerttrajektorien-Folgeregelung

Statt der Einhaltung einer konstanten Nominalposition ( $x = 0$ ) soll die Regelung der Systemdynamik so erfolgen, dass der Bewegungsverlauf der Masse  $x(t)$  einer vorgegebenem Trajektorie  $x_d(t)$  von Sollwerten (z.B. des Trajektorienplaners) folgt. Die Sollwerttrajektorie (set-point trajectory)  $x_d(t)$  sei „glatt“, d.h.  $\dot{x}_d(t)$  und  $\ddot{x}_d(t)$  existieren und werden zusammen mit  $x_d(t)$  vorgegeben.

Gemäß Abschnitt 6.3.4 sei die Systemdynamik transformiert auf

$$\ddot{x} = \hat{f}.$$

Durch etwas Überlegung erhält man als Ansatz für das Regelgesetz

$$\hat{f} = \ddot{x}_d + k_v \underbrace{(\dot{x}_d - \dot{x})}_{=\dot{e}} + k_p \underbrace{(x_d - x)}_{=e}.$$

Somit lautet die Dynamik des geregelten Systems:

$$\begin{aligned}\ddot{x} &= \ddot{x}_d + k_v \underbrace{(\dot{x}_d - \dot{x})}_{=\dot{e}} + k_p \underbrace{(x_d - x)}_{=e} \\ \Leftrightarrow 0 &= \ddot{e} + k_v \dot{e} + k_p e.\end{aligned}$$

Nach Abschnitt 6.3.2 kann für diese Dynamik im Fehlerraum durch entsprechende Wahl von  $k_v$  und  $k_p$  ein gewünschtes Systemverhalten erreicht werden, z.B. kritische Dämpfung durch

$$k_v = 2\sqrt{k_p},$$

so dass Anfangsfehler  $e(0) \neq 0$  möglichst schnell abklingen. Das zugehörige Blockschaltbild ist in Abbildung 6.6 dargestellt.

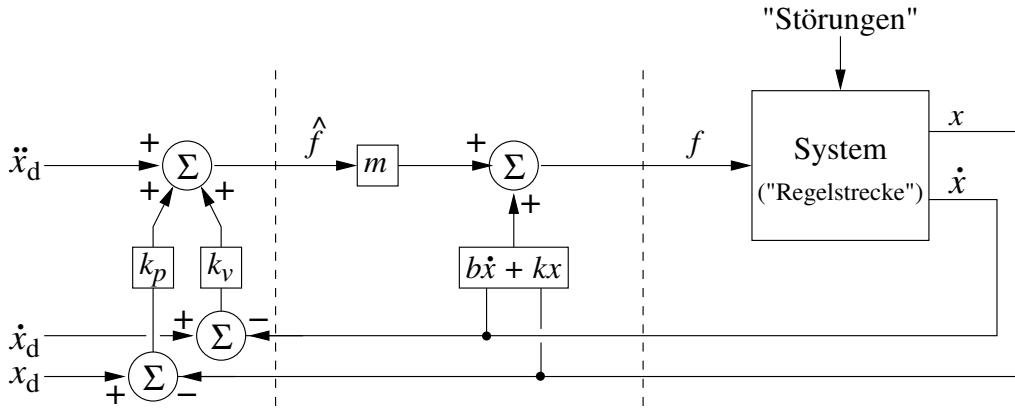


Figure 6.6: Sollwerttrajektorien-Folgeregelung für das Feder-Masse-System.

### 6.3.6 PID-Regelung linearer Systeme (mit Anti-Windup)

Eine Aufgabe des Regelungssystems ist es, auch im Fall externer Störungen oder Rauschens ein gutes Systemverhalten (d.h. nur kleine Regelfehler) zu ermöglichen. In Abbildung 6.7 ist das Blockschaltbild des PD-Sollwerttrajektorienreglers dargestellt mit expliziter Berücksichtigung einer zusätzlich wirkenden Störungskraft  $f_s$  (z. B. beim Feder-Masse-System verursacht durch einen Rücken- oder Gegenwind).

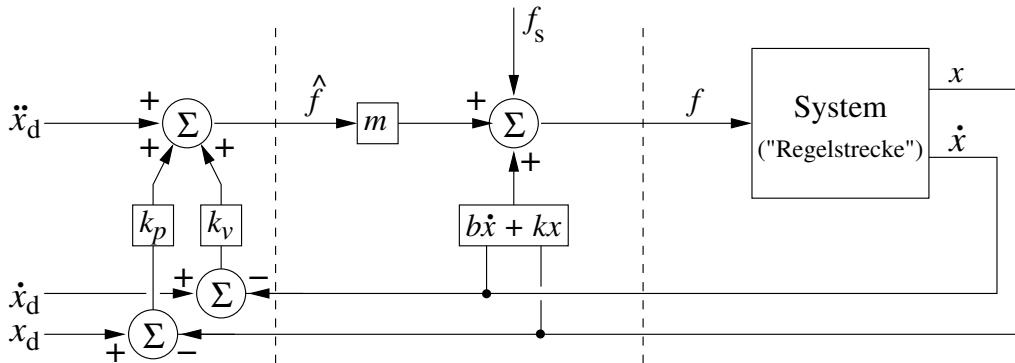


Figure 6.7: Sollwerttrajektorien-Folgeregelung für das Feder-Masse-System mit Berücksichtigung einer Störungskraft.

Die Regelabweichung  $e$  dieses Systems genügt der Differentialgleichung

$$\ddot{e} + k_v \dot{e} + k_p e = f_s.$$

Diese Differentialgleichung ist von der Form einer linearen Systemdynamik zweiter Ordnung mit einer „Antriebskraft“  $f_s$ . Falls die Störungskraft beschränkt ist, d.h. es existiert eine reelle Konstante  $c_s$  so dass gilt

$$\max |f_s(t)| < c_s, \quad \text{für alle } t,$$

dann ist gemäß Abschnitt 6.3.2  $e(t)$  als Lösung der Differentialgleichung ebenfalls beschränkt. D.h. ein stabiles, lineares dynamisches System bleibt für eine große Menge möglicher Störungen garantiert stabil.

Der einfachste Fall einer Störung ist eine konstante Störung, d.h.  $f_s = \text{const.}$  Wird das System im stationären Zustand (eingeschwungener Zustand, Ruhelage) betrachtet (d.h.  $\dot{e} = \ddot{e} = 0$ ), so gilt

$$k_p e = f_s \Leftrightarrow e = \frac{f_s}{k_p}.$$

Dieser Wert von  $e$  stellt den stationären Zustandsfehler dar. Folglich gilt: Je größer der Positionsregelparameter  $k_p$  ist, desto kleiner wird der stationäre Zustandsfehler.

Um den stationären Zustandsfehler zu eliminieren, kann ein modifiziertes Regelgesetz verwendet werden, wobei ein Integralterm hinzugefügt wird:

$$\hat{f} = \ddot{x}_d + k_v \dot{e} + k_p e + k_i \int e dt.$$

Dieses führt auf die neue Differentialgleichung für  $e$ :

$$\ddot{e} + k_v \dot{e} + k_p e + k_i \int e dt = f_s.$$

Durch Hinzufügen des Integralterms soll das System im Falle konstanter Störkraft  $f_s$  keinen stationären Zustandsfehler haben.

Warum kann dies erreicht werden? Um für die Untersuchung den Integralterm zu eliminieren, betrachtet man das einmal total nach  $t$  abgeleitete System

$$\ddot{e} + k_v \dot{e} + k_p e + k_i e = \dot{f}_s,$$

für dessen stationären Zustand folgt

$$k_i e = 0 \Leftrightarrow e = 0.$$

Wie geht man allgemein vor? Durch dieses Regelgesetz wird das dynamische System eine lineare Differentialgleichung **dritter Ordnung**. Diese kann ähnlich wie die Differentialgleichung zweiter Ordnung im Abschnitt 6.3.2 gelöst werden, um das resultierende Systemverhalten in Abhängigkeit der Anfangswerte  $e(0)$ ,  $\dot{e}(0)$ ,  $\ddot{e}(0)$  zu bestimmen und dann die Regelparameter  $k_p$ ,  $k_v$ ,  $k_i$  entsprechend zu wählen. Häufig wird  $k_i$  „klein“ gewählt, so dass das Verhalten der Lösung der Differentialgleichung dritter Ordnung „nahe“ beim Verhalten der Lösung der Differentialgleichung zweiter Ordnung ohne den Integralterm liegt.

Die in diesem Abschnitt untersuchte Form des Regelgesetzes besitzt einen Proportionalanteil (P), einen Integralanteil (I) und einen Differentialanteil (D) und wird als **PID-Regelung** bezeichnet.

Weitere Anmerkungen, u.a. zu Windup-Effekt und Anti-Windup, siehe Kopien der Folien.

### 6.3.7 Kaskadenregelung

Hierzu werden die Kopien der Folien online bereitgestellt.

### 6.3.8 Stabilität als Sprungantwortverhalten und Wahl der PID-Reglerparameter

Hierzu werden die Kopien der Folien online bereitgestellt.

## 6.4 Nichtlineare Regelung

Für dieses und alle nachfolgenden Abschnitte dieses Kapitels werden die Kopien der Folien online bereitgestellt.

### 6.4.1 Systemlinearisierung

### 6.4.2 Modellbasierte Manipulatorregelung (Feedback Linearization)

### 6.4.3 Adaptive Manipulatorregelung

### 6.4.4 Bahnregelung in Weltkoordination (Kartesische Bahnregelung)

## 6.5 Kraft-/Momenten-Regelung

## 6.6 Hybride Bahn-/Kraft-Regelung

## 6.7 Nachgiebigkeitsregelungen (Compliant Control)

### 6.7.1 Impedanzregelung

### 6.7.2 Admittanzregelung

### 6.7.3 Aktiv-passive Konzepte für Impedanz-/Admittanzregelungen

## 6.8 Bildgeführte Regelung

## 6.9 Multimodale Regelung physikalischer Interaktion

## 6.10 Roboterhände und Greifen

## 6.11 Elementare Roboterbewegungen

## 6.12 Regelung und Steuerung von Bewegungen bei Mensch und Tier

# Chapter 7

## Bahnplanung

Das Skript zu Kapitel 7 sind die Kopien der Folien.

### 7.1 Einleitung und Übersicht

### 7.2 Topologische Wegeplanung

### 7.3 Konfigurationsraum

### 7.4 Geometrische Bahnplanung

#### 7.4.1 Roadmap-Verfahren

#### 7.4.2 Exakte Zellzerlegung

#### 7.4.3 Approximative Zellzerlegung

#### 7.4.4 Potentialfeldmethoden

#### 7.4.5 Neuere Verfahren: Stichprobenverfahren, RRTs

#### 7.4.6 Komplexität der geometrischen Bahnplanung

#### 7.4.7 Anwendungsbeispiele

### 7.5 Kinematische und dynamische Trajektorienplanung

### 7.6 Greifplanung

# Chapter 8

## Externe Sensorsysteme

Das Skript zu Kapitel 8 sind die Kopien der Folien.

### 8.1 Nichtvisuelle externe Sensoren

### 8.2 Visuelle Sensoren und Algorithmen

#### 8.2.1 Einleitung

#### 8.2.2 Bilderzeugung

#### 8.2.3 Grundlagen der Bildverarbeitung

#### 8.2.4 Bildvorverarbeitung

#### 8.2.5 Bildverarbeitung

#### 8.2.6 Merkmalsextraktion

#### 8.2.7 Objektklassifikation

### 8.3 3D-Sensoren, -Perzeption und Umgebungsrepräsentation

# Chapter 9

## Navigation mobiler Roboter

Das Skript zu Kapitel 9 sind die Kopien der Folien.

### 9.1 Lokalisierung und Positionierung

#### 9.1.1 Relative Positionsbestimmung

#### 9.1.2 Absolute Positionsbestimmung

#### 9.1.3 Stochastische Positionsbestimmung

### 9.2 Selbstlokalisierung und Navigation

#### 9.2.1 Einleitung und Übersicht

#### 9.2.2 Lokalisierung mit einer Hypothese (Kalman-Filter)

#### 9.2.3 Lokalisierung mit mehreren Hypothesen

##### 9.2.3.1 Verwendung mehrerer Kalman-Filter

##### 9.2.3.2 Diskretisierte Wahrscheinlichkeitsverteilung

##### 9.2.3.3 Monte-Carlo Lokalisierung

#### 9.2.4 SLAM (Simultaneous Localization and Mapping)

# Chapter 10

## Middleware und Simulation

Das Skript zu Kapitel 10 sind die Kopien der Folien.

### 10.1 Middleware für Robotersysteme

### 10.2 Simulation autonomer Roboter

- Simulation des physikalischen Bewegungsverhaltens (Kinematik, Dynamik, ODE-/DAE-Integration); Aktuatorsimulation
- Simulation der internen und externen Sensorsysteme
- Simulation der Umwelt und der Interaktion mit der Umwelt
- Systementwicklung mit Hardware-/Software-in-the-Loop Simulation

### 10.3 Robot Operating System (ROS)

# Chapter 11

## Steuerung autonomer Roboter

Das Skript zu Kapitel 11 sind die Kopien der Folien.

### 11.1 Steuerungsarchitekturen

### 11.2 Programmierung von autonomem Verhalten

### 11.3 Roboterverhalten durch generative KI

## **Chapter 12**

# **Herausforderung und Verantwortung der Robotik**

Das Skript zu Kapitel 12 sind die Kopien der Folien.

# Anhang

## A.1 Orientierungsrepräsentation mit Quaternionen<sup>1</sup>

Quaternionen werden in der Vorlesung Robotik 1 nicht verwendet. Da diese in speziellen Fällen auch in der Robotik eine Rolle spielen, sollen an dieser Stelle einige, ergänzende Informationen mitgeteilt werden.

### Einleitung

Ein Quaternion  $\mathbf{q} = \begin{pmatrix} s \\ \mathbf{v} \end{pmatrix}$  besteht aus einem skalaren Anteil  $s \in \mathbb{R}$  und einem vektoriellen Anteil  $\mathbf{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3$ .

Wenn das Quaternion normiert ist, also  $\|\mathbf{q}\| = 1$  gilt, wird es als Einheitsquaternion bezeichnet. Einheitsquaternionen repräsentieren Drehungen im Raum. Wird mit  $\phi$  der Drehwinkel und mit  $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$  der normierte Drehvektor bezeichnet, so gilt

$$\mathbf{q} = \begin{pmatrix} \cos(\frac{\phi}{2}) \\ \sin(\frac{\phi}{2}) \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \end{pmatrix}$$

Die Nullrotation eines Quaternions wird durch  $[1, (0, 0, 0)]^T$  beschrieben (vgl. bei Drehmatrizen z. B. Drehung um eine Achse um  $0^\circ$ :  $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ ).

**Bemerkung:** Quaternionen spielen in der Algebra eine große Rolle, z.B. bildet die Menge der Quaternionen einen Schiefkörper. Quaternionen werden dort auch in einer erweiterten „Komplexe Zahlen“-Notation dargestellt:

$$\begin{aligned} \mathbf{q} = s + xi + yj + zk \quad \text{mit} \quad i^2 = j^2 = k^2 = -1 \\ ij = k = -ji \\ s, x, y, z \in \mathbb{R} \end{aligned}$$

An dieser Stelle soll aber nicht weiter darauf eingegangen werden.

### Rechenregeln

Wichtig ist die Multiplikation von Quaternionen, denn dadurch kann die Hintereinanderausführung von Rotationen dargestellt werden. Darüber hinaus spielen für allgemeine Beobachtungen noch weitere Operationen eine Rolle, von denen die wichtigsten angegeben sind.

Es sei  $\mathbf{q} = \begin{pmatrix} s \\ \mathbf{v} \end{pmatrix}$ ,  $\mathbf{q}_1 = \begin{pmatrix} s_1 \\ \mathbf{v}_1 \end{pmatrix}$ ,  $\mathbf{q}_2 = \begin{pmatrix} s_2 \\ \mathbf{v}_2 \end{pmatrix}$  und  $\mathbf{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3$ .

Addition:	$\mathbf{q}_1 + \mathbf{q}_2 = [s_1 + s_2, \mathbf{v}_1 + \mathbf{v}_2]^T$
Multiplikation:	$\mathbf{q}_1 \mathbf{q}_2 = [s_1 s_2 - \mathbf{v}_1^T \mathbf{v}_2, \mathbf{v}_1 \times \mathbf{v}_2 + s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1]^T$
Konjugiertes:	$\mathbf{q}^* = [s, -\mathbf{v}]^T$
Norm:	$N(\mathbf{q}) = s^2 + x^2 + y^2 + z^2$
Inverses:	$\mathbf{q}^{-1} = \frac{1}{N(\mathbf{q})} \cdot \mathbf{q}^*$

(Q1)

---

<sup>1</sup>Dieses Kapitel beruht auf einer Ausarbeitung von Jutta Kiener.

## Umrechnungen: Quaternionen $\leftrightarrow$ Rotationsmatrizen

Ein Einheitsquaternion  $\mathbf{q}$  ist eine alternative Darstellung für eine Drehung, wie sie auch durch eine Rotationsmatrix  $R$  ausgedrückt wird. Deshalb ist es auch möglich, die eine Darstellung durch die andere auszudrücken. Gründe, warum man sich nicht auf eine der beiden Darstellungsarten beschränkt, sind beispielsweise, dass die Berechnungen mit Quaternionen effizient sind, diese jedoch nicht so intuitiv sind wie Rotationsmatrizen. Programme wie OpenGL oder Direct3D unterstützen Drehmatrizen, die Hintereinanderausführung von vielen Drehungen kann aber trotzdem mit Quaternionen effizienter, d.h. schneller, sein. Aus diesem Grund sind effiziente Funktionen zur Umrechnung zwischen Quaternionen und Drehmatrizen nötig.

Sei  $\mathbf{q} = \begin{pmatrix} s \\ x \\ y \\ z \end{pmatrix}$  ein Einheitsquaternion. Dann lässt sich die Rotationsmatrix  $R \in \mathbb{R}^{3 \times 3}$  beschreiben durch

$$R = \begin{pmatrix} 1 - 2(y^2 + z^2) & 2(xy - zs) & 2(xz + ys) \\ 2(xy + zs) & 1 - 2(x^2 + z^2) & 2(yz - xs) \\ 2(xz - ys) & 2(yz + xs) & 1 - 2(x^2 + y^2) \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}. \quad (\text{Q2})$$

Andererseits lässt sich mittels der Rotationsmatrix  $R$  ein Einheitsquaternion beschreiben als

$$\begin{aligned} s &= \cos\left(\frac{\phi}{2}\right) = \frac{1}{2}\sqrt{\text{Spur}(R) + 1}, \quad \text{mit } \text{Spur}(R) = r_{11} + r_{22} + r_{33}, \\ \mathbf{v} &= \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{2\sqrt{\text{Spur}(R) + 1}} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}. \end{aligned} \quad (\text{Q3})$$

## Verkettung von Drehungen

Sollen mehrere Drehungen hintereinander ausgeführt werden, so werden die dazugehörigen Einheitsquaternionen miteinander multipliziert. Sei beispielsweise die erste Drehung mit Quaternion  $\mathbf{q}_1$ , die zweite mit  $\mathbf{q}_2$ , so lautet die Gesamtdrehung  $\mathbf{q}_2\mathbf{q}_1$  mit der in (Q1) definierten Multiplikation von Quaternionen. Diese Berechnungsweise ist sehr effizient. Wie auch bei Drehmatrizen ist die Multiplikation von Quaternionen nicht kommutativ.

Um die Normierung des Quaternions der Gesamtdrehung zu gewährleisten, muss in der Praxis gegebenenfalls nach der Multiplikation am Rechner noch eine zusätzliche Normierung vorgenommen werden, falls das Produktquaternion wegen Rundungsfehlern nicht mehr die Norm 1 hat<sup>2</sup>.

## Repräsentation der Koordinaten eines Punktes bei Rotation

Die Koordinatentransformation eines Punktes  ${}^a\mathbf{p}$  bzw.  ${}^b\mathbf{p} \in \mathbb{R}^3$  bei Anwendung von Rotationsmatrizen lässt sich darstellen mittels

$${}^a\mathbf{p} = {}^aR_b {}^b\mathbf{p}.$$

---

<sup>2</sup>Diese „Nachnormierung“ ist immer noch billiger als die Korrektur bzgl. der Drehung bei Drehmatrizen.

Bei Quaternionen ist die Darstellung etwas umständlicher, hier wird noch das konjugierte Quaternion miteinbezogen:

$${}^a\tilde{\mathbf{p}} = {}^a\mathbf{q}_b {}^b\tilde{\mathbf{p}} {}^a\mathbf{q}_b^*.$$

Dabei sind  $\tilde{\mathbf{p}}^T = (1, \mathbf{p}^T) \in \mathbb{R}^4$  der homogenisierte Positionsvektor,  ${}^a\mathbf{q}_b$  das der Rotationsmatrix  ${}^aR_b$  gemäß (Q2) bzw. (Q3) entsprechende Quaternion,  ${}^a\mathbf{q}_b^*$  das gemäß (Q1) konjugierte Quaternion und die Multiplikation ist gemäß (Q1) durchzuführen.

Die Verwendung des Konjugierten macht die explizite Darstellung von Punkten mittels Quaternionen umständlich, deshalb werden dafür eher andere Darstellungen von Drehungen, eben z. B. Rotationsmatrizen, gewählt.

## Anwendung

Quaternionen werden wegen ihres geringen Rechenaufwands sehr gerne in der Computergraphik für Animationen verwendet. So sind Quaternionen z. B. bei der Programmierung des Spiels *Tomb Raider* zum Einsatz gekommen. Verwendung finden sie auch bei Spieltypen wie Flugsimulatoren. Ein Grund dafür ist, dass mit Quaternionen sehr gut zwischen verschiedenen Positionen interpoliert werden kann.

## Vergleich mit anderen Darstellungsarten

Vorteile von Quaternionen:

- Beschreibung von Rotationen mittels einer Rotationsachse ist geometrisch intuitiv,
- kompakte Darstellung mit wenig Speicherbedarf
- geringe Anzahl arithmetischer Operationen (rechenzeiteffizient)
- höhere numerische Genauigkeit durch weniger Rundungsfehlereinfluss bei weniger arithmetischen Operationen (insbesondere relevant, wenn die Anzahl der Subtraktionen und Divisionen reduziert wird)
- schnelle Berechnungsmöglichkeit für Winkelgeschwindigkeiten
- effiziente Berechnung von verketteter Drehungen
- keine „Singularitäten“ (z.B. bei Verwendung von Euler-Winkeln kann es vorkommen, dass nach zwei Drehungen die Achse, um die noch nicht gedreht worden ist, zusammenfällt mit einer Achse, um die schon gedreht worden ist, und somit die komplexe Drehung nicht ausführbar ist. Dies kann bei einer Rotationsdarstellung mit Quaternionen nicht vorkommen.)

Nachteile:

- Arithmetische Operationen mit Quaternionen wirken kompliziert,
- Translationen sind mit Quaternionen nicht so elegant beschreibbar wie mit homogenen Transformationen.

Verweise auf weitere Informationen im WWW:

- <http://mathworld.wolfram.com/Quaternion.html>
- <http://www.opengl.org>
- [https://www.gamasutra.com/view/feature/131686/rotating\\_objects\\_using\\_quaternions.php](https://www.gamasutra.com/view/feature/131686/rotating_objects_using_quaternions.php)
- <http://www.gamedev.net, speziell https://www.gamedev.net/articles/programming/math-and-physics/quaternion-powers-r1095/>
- und viele andere mehr.

## A.2 Zusammenhang zwischen Rotationsmatrix und Drehvektor und Drehwinkel<sup>3</sup>

### A.2.1 Berechnung der Rotationsmatrix aus Drehvektor und Drehwinkel

Eine Drehung  $\mathbf{dr} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  mit  $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \neq 0$  als Drehvektor und  $\phi \geq 0$  als Drehwinkel ist gegeben durch

$$\mathbf{dr}(\mathbf{x}) = (\cos \phi) \mathbf{x} + (1 - \cos \phi) \frac{\mathbf{x}^T \mathbf{a}}{|\mathbf{a}|^2} \mathbf{a} + \frac{\sin \phi}{|\mathbf{a}|} \mathbf{a} \times \mathbf{x}$$

(Nachweis z. B. Meyberg, Vachenauer: Höhere Mathematik 1, Springer 1993, S. 319)

Der Drehvektor  $\mathbf{a} = (a_1, a_2, a_3)^T$ , die gerichtete, normierte (d. h.  $|\mathbf{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2} = 1$ ) Drehachse, beschreibt zusammen mit dem Drehwinkel  $\phi$  die zu der oben beschriebenen Drehung  $\mathbf{dr}$  gehörende Rotationsmatrix  $D$

$$\mathbf{dr}(\mathbf{x}) = D \cdot \mathbf{x}$$

mit

$$D = \begin{pmatrix} \cos \phi + (1 - \cos \phi)a_1^2 & (1 - \cos \phi)a_1a_2 - a_3 \sin \phi & (1 - \cos \phi)a_1a_3 + a_2 \sin \phi \\ (1 - \cos \phi)a_1a_2 + a_3 \sin \phi & \cos \phi + (1 - \cos \phi)a_2^2 & (1 - \cos \phi)a_2a_3 - a_1 \sin \phi \\ (1 - \cos \phi)a_1a_3 - a_2 \sin \phi & (1 - \cos \phi)a_2a_3 + a_1 \sin \phi & \cos \phi + (1 - \cos \phi)a_3^2 \end{pmatrix}$$

Dies ist die allgemeine Darstellung einer Rotationsmatrix mittels des Drehvektors  $\mathbf{a}$  und des Drehwinkels  $\phi$ . Diese Beschreibung gilt für alle Rotationsmatrizen. Durch Nachrechnen kann man sich davon überzeugen, dass die Bedingungen  $DD^T = I$  und  $\det(D) = 1$  erfüllt sind.

### A.2.2 Berechnung von Drehvektor und Drehwinkel aus der Rotationsmatrix

Die Drehachse und damit auch der Drehvektor  $\mathbf{a}$  haben die Eigenschaft, dass sie unter der Drehung, zu der sie gehören, invariant sind, also  $\mathbf{a} = D\mathbf{a}$ . Durch Umformung ergibt sich

$$(D\mathbf{a} - I\mathbf{a}) = 0 \Leftrightarrow (D - I)\mathbf{a} = 0 \quad \text{mit } I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.2})$$

Der Drehvektor kann somit auch als Eigenvektor der Matrix  $D$  zum Eigenwert 1 gesehen werden.<sup>4</sup>

Sei  $M := (D - I) = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix}$  mit den Zeilenvektoren  $\mathbf{m}_1, \mathbf{m}_2$  und  $\mathbf{m}_3$ .

<sup>3</sup>Dieses Kapitel beruht auf einer Ausarbeitung von Markus Glocker und Jutta Kiener.

<sup>4</sup>Aus der Linearen Algebra ist bekannt, dass für einen beliebigen Eigenwert  $\lambda$  zu einer Matrix  $A$  gilt:  $\det(A - \lambda I) = 0$ . Dieser Sachverhalt gilt natürlich auch hier. D. h.  $\det(D - I) = 0$ , also sind die Zeilenvektoren von  $(D - I)$  linear abhängig und liegen für eine Rotationsmatrix  $D$  in einer Ebene und es lässt sich genau ein Vektor (bis auf Vielfache) bestimmen, der auf dieser Ebene senkrecht steht. Dieser Vektor ist  $\mathbf{a}$ .

Nach (A.2) gilt  $\mathbf{m}_1 \mathbf{a} = 0$  und  $\mathbf{m}_2 \mathbf{a} = 0$ , also  $\mathbf{a}$  steht rechtwinklig sowohl auf  $\mathbf{m}_1^T$  als auch auf  $\mathbf{m}_2^T$ . Der **Drehvektor**  $\mathbf{a}$  kann also mit dem Vektorprodukt ausgedrückt werden,

$$\mathbf{a} = \frac{\mathbf{m}_1^T \times \mathbf{m}_2^T}{|\mathbf{m}_1^T \times \mathbf{m}_2^T|}.$$

Der Nenner  $|\mathbf{m}_1^T \times \mathbf{m}_2^T|$  dient dabei zur Normierung von  $\mathbf{a}$ .

Zur Berechnung des dazugehörigen **Drehwinkels**  $\phi$  addiert man die Diagonalelemente der Rotationsmatrix. Die Summe der Diagonalelemente einer Matrix wird als deren *Spur* bezeichnet. Wegen der Normierung von  $\mathbf{a}$  lässt sich die Spur vereinfachen zu

$$\begin{aligned} \text{Spur}(D) &= d_{11} + d_{22} + d_{33} = \\ &= \cos \phi + (1 - \cos \phi)a_1^2 + \cos \phi + (1 - \cos \phi)a_2^2 + \cos \phi + (1 - \cos \phi)a_3^2 = \\ &= 3\cos \phi + (1 - \cos \phi)(a_1^2 + a_2^2 + a_3^2) = \\ &= 3\cos \phi + (1 - \cos \phi) = \\ &= 2\cos \phi + 1 \end{aligned}$$

bzw.

$$\cos \phi = \frac{1}{2}(\text{Spur}(D) - 1) \quad (\text{A.3})$$

Der **Drehwinkel**  $\phi$  ergibt sich aus (A.3):

$$\phi = \arccos \left( \frac{1}{2}(\text{Spur}(D) - 1) \right).$$

Da  $\arccos$  im Bereich  $[0, 2\pi]$  nicht eindeutig ist, wählt man einen beliebigen Vektor  $\mathbf{v}$  mit  $\mathbf{v} \perp \mathbf{a} \Leftrightarrow \mathbf{v}^T \mathbf{a} = 0$ . Man bilde den Term

$$\mathbf{v} \times D \mathbf{v} = \mu \mathbf{a},$$

der entscheidet, in welche Richtung der Drehvektor zeigt. Ist  $\mu > 0$ , so ist  $0 \leq \phi < \pi$ ; ist  $\mu \leq 0$ , dann liegt  $\phi$  im Intervall  $\pi \leq \phi < 2\pi$ .

Eine mathematisch nicht so schöne, aber einfachere Darstellung für den Drehvektor  $\mathbf{a}$  erhält man durch eine zum Drehwinkel analoge Überlegung:

Aus (A.1) ergibt sich mit

$$d_{32} = (1 - \cos \phi)a_2a_3 + a_1 \sin \phi \text{ und } d_{23} = (1 - \cos \phi)a_2a_3 - a_1 \sin \phi$$

$$\begin{aligned} d_{32} - d_{23} &= (1 - \cos \phi)a_2a_3 + a_1 \sin \phi - [(1 - \cos \phi)a_2a_3 - a_1 \sin \phi] = 2a_1 \sin \phi \\ &\Leftrightarrow a_1 = \frac{1}{2 \sin \phi} (d_{32} - d_{23}) \quad \text{für } \phi \neq \pi \end{aligned}$$

Überträgt man diese Überlegung auch auf die anderen beiden Elemente von  $\mathbf{a}$ , so ergibt sich insgesamt für die Drehachse die Beschreibung

$$\begin{pmatrix} d_{32} - d_{23} \\ d_{13} - d_{31} \\ d_{21} - d_{12} \end{pmatrix} \quad \text{für } \phi \neq \pi.$$

und für den **Drehvektor a**

$$\mathbf{a} = \frac{1}{2\sin\phi} \begin{pmatrix} d_{32} - d_{23} \\ d_{13} - d_{31} \\ d_{21} - d_{12} \end{pmatrix} \quad \text{für } \phi \neq \pi.$$

Allerdings muss für diese Vorgehensweise der Drehwinkel  $\phi$  schon bekannt sein, da sonst  $\sin\phi$  nicht ausgewertet werden kann. Dafür wählt man, wie oben beschrieben, den Drehwinkel so, dass er im Intervall  $[0; \pi[$  liegt<sup>5</sup>.

Die beiden Darstellungen für den Drehvektor **a** sind natürlich äquivalent.

---

<sup>5</sup>Wählt man den Drehwinkel  $\phi$  im Intervall  $[\pi, 2\pi[,$  so hat  $\sin(\phi)$  das negative Vorzeichen, der Drehvektor ist in die Gegenrichtung gerichtet. Die Drehung insgesamt bleibt damit wieder gleich.

## A.3 Vergleich einiger Notationen im Skript mit anderen Lehrbüchern

Beschreibung	Skript	Spong/Vid. (1989)	Craig (1989)	Yoshikawa, 1990
lokale Koordinatensysteme Koordinatenachsen	$S_a$ $x_a, y_a, z_a$ $S_b$ $x_b, y_b, z_b$	$o_a x_a y_a z_a$	$\{A\}$ $\hat{X}_A, \hat{Y}_A, \hat{Z}_A$ $\{B\}$	$\Sigma_A$ $X_A, Y_A, Z_A$ $\Sigma_B$
Einheitsvektoren $\in \mathbb{R}^3$ eines Rechtskoordinaten-systems	$\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ ${}^i \mathbf{e}_{x_i}, {}^i \mathbf{e}_{y_i}, {}^i \mathbf{e}_{z_i}$ ${}^i \mathbf{e}_{z_i} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ ${}^0 \mathbf{e}_{z_i} = {}^0 R_i {}^i \mathbf{e}_{z_i}$	$i, j, k$	${}^i \hat{X}_i, {}^i \hat{Y}_i, {}^i \hat{Z}_i$	
Translationsvektor $\in \mathbb{R}^3$	${}^0 \mathbf{r}_n$	$\mathbf{d}_0^n$		${}^0 p_n$
Koordinatenvektor $\in \mathbb{R}^3$	${}^0 \mathbf{p}$ ${}^a \mathbf{p}$ ${}^b \mathbf{p}$	$\mathbf{p}_0$ $\mathbf{p}_a$ $\mathbf{p}_b$	${}^0 P$ ${}^A P$ ${}^B P$	${}^0 r$ ${}^A r$ ${}^B r$
( $3 \times 3$ )-Rotationsmatrizen	${}^a R_b$ ${}^0 R_n$	$R_a^b$ $R_0^n$	${}^A B R$	${}^A R_B$
homogene ( $4 \times 4$ )-Transformationsmatrizen	${}^a T_b$ ${}^0 T_n$	$H_a^b$ $T_0^n$	${}^A B T$	
homogene Elementarrotationsmatrix $\in \mathbb{R}^{4 \times 4}$	$\text{Rot}(x; \theta_x)$	$R_{x, \theta}$		
schiefsymmetrische Matrix $\in \mathbb{R}^{3 \times 3}$	$B$	$S$	$S$	
Einheitsmatrix $\in \mathbb{R}^{3 \times 3}$ bzw. $\in \mathbb{R}^{n \times n}$	$E, E_3, E_n$			
Trägheitstensor $\in \mathbb{R}^{3 \times 3}$	$I$	$I$	$I$	
Winkelgeschwindigkeitsvektor des $i$ -ten Gelenks bezüglich Frame $S_{i-1}$	${}^{i-1} \omega_i$	$\omega_{i-1}^i$		
Winkelgeschwindigkeitsvektor des $i$ -ten Gelenks bezüglich Basis-Frame $S_0$	${}^0 \omega_i$	$\omega_0^i$		
linearer Geschwindigkeitsvektor des Ursprungs von $S_n$ (oft TCP) bezüglich festem Basis-Frame $S_0$	${}^0 \mathbf{v}_n$	$\mathbf{v}_0^n$		

**Hinweis:** Die Nummerierung von Gelenken und zugehörigen Koordinatensystemen ist in der Literatur nicht einheitlich. Nach der Denavit-Hartenberg Konvention, die im Skript sowie in den Lehrbüchern von [Spong/Vidyasagar, 1989], [Fu/Gonzalez/Lee, 1987] ver-

wendet wird, befindet sich im ***i*-ten Gelenk** das Koordinatensystem  $S_{i-1}$ . In den Lehrbüchern von [Craig, 1989] sowie von [Yoshikawa, 1990] befindet sich dagegen im ***i*-ten Gelenk** das Koordinatensystem  $S_i$ .

## A.4 Typische Notationen und Symbole in Robotik 1

### A.4.1 Griechisches Alphabet

klein geschrieben	griechischer Buchstabe	groß geschrieben	Bezeichnung
$\alpha$	$A$		Alpha
$\beta$	$B$		Beta
$\gamma$	$\Gamma$		Gamma
$\delta$	$\Delta$		Delta
$\epsilon$ bzw. $\varepsilon$	$E$		Epsilon
$\zeta$	$Z$		Zeta
$\eta$	$H$		Eta
$\theta$ bzw. $\vartheta$	$\Theta$		Theta
$\iota$	$I$		Iota
$\kappa$	$K$		Kappa
$\lambda$	$\Lambda$		Lambda
$\mu$	$M$		My
$\nu$	$N$		Ny
$\xi$	$\Xi$		Xi
$\o$	$O$		Omicron
$\pi$ bzw. $\varpi$	$\Pi$		Pi
$\rho$ bzw. $\varrho$	$P$		Rho
$\sigma$ bzw. $\varsigma$	$\Sigma$		Sigma
$\tau$	$T$		Tau
$v$	$\Upsilon$		Ypsilon
$\phi$ bzw. $\varphi$	$\Phi$		Phi
$\chi$	$X$		Chi
$\psi$	$\Psi$		Psi
$\omega$	$\Omega$		Omega

### A.4.2 Typische Symbole und ihre Bedeutung

- Kleine, kursive, römische Buchstaben bezeichnen meistens skalare Größen, z.B.  $t, x, f, g, h$ .
- Große, kursive Buchstaben bezeichnen meistens Matrizen<sup>6</sup>, z.B.  $A, B, R, T$ .
- Fettgedruckte römische Buchstaben kennzeichnen Vektoren bzw. vektorwertige Größen, z.B.  $\mathbf{x}, \mathbf{q}, \mathbf{F}$ .

<sup>6</sup>Achtung: Der Singular zu „Matrizen“ ist „Matrix“ und **nicht** „Matrize“!

**Allgemein**

$f, g, h$	(skalare) Funktionsnamen
$t$	Zeit
$t_a, t_s$	Startzeitpunkt
$t_e, t_f$	Endzeitpunkt
$x, y, z$	Koordinatenachsen
$i, j, k$	Laufindex (z. B. in Schleifen)
$l$	Länge
$m$	Masse
$\rho$	Dichte
$g$	Gravitationskonstante
$\delta$	kleiner Abstand
$\frac{df}{dx} = f'(x)$	(totale) Ableitung nach der unabhängigen Variablen $x$
$\frac{dx}{dt} = \dot{x}(t)$	(totale) Ableitung nach der unabhängigen Variablen Zeit $t$

**Räumliche Darstellung und Transformationen**

$p$	Koordinatenvektor eines Punkts, meist (zwei- oder) dreidimensional
$\vartheta$	Winkel
$(\alpha, \beta, \gamma)$	Winkeltripel
$(\psi, \vartheta, \varphi)$	Winkeltripel
$E$	Einheitsmatrix
$R$	Rotationsmatrix $\in \mathbb{R}^{3 \times 3}$
$r$	Translationsvektor $\in \mathbb{R}^3$
$T$	homogene Transformationsmatrix $\in \mathbb{R}^{4 \times 4}$

**Manipulatorkinematik**

$S$	Koordinatensystem
$n$	Anzahl, z. B. der Freiheitsgrade
$\mathbf{q} = (q_1, q_2, \dots, q_n)^T$	Zustandsvektor der Gelenkvariablen $\in \mathbb{R}^n$

**Geschwindigkeit, Jacobi-Matrix, statische Kräfte**

$\omega$	Winkelgeschwindigkeitsvektor $\in \mathbb{R}^3$
$\mathbf{v}$	linearer Geschwindigkeitsvektor $\in \mathbb{R}^3$
$J$	Jacobi-Matrix $\in \mathbb{R}^{6 \times n}$

**Manipulatordynamik**

$I$	Trägheitstensor
$n, N$	(skalares) Drehmoment (Achtung: <i>der</i> Moment ist ein kurzer Zeitabschnitt)
$f, F$	(skalare) Kraft
$\tau$	(skalare oder vektorwertige) Kraft oder Drehmoment
$M$	Massenmatrix
$C$	Vektorfunktion der Zentrifugal- und Coriolisanteile $\in \mathbb{R}^n$
$G$	Vektorfunktion der Gravitationsanteile $\in \mathbb{R}^n$
$K$	kinetische Energie
$P$	potentielle Energie

# Index

- $E$ , Einheitsmatrix, 12  
 $O$ , Nullmatrix, 12  
 $S_0, S_F$ , Basiskoordinatensystem, 30  
 $S_i$ , gliedfestes Koordinatensystem am ( $i + 1$ )-ten Gelenk, 30  
 $S_n, S_E$ , Endeffektorkoordinatensystem, 30  
 $S_{c_i}$ , gliedfestes Koordinatensystem im Schwerpunkt, 103  
 $\mathcal{A}$ , erreichbarer Arbeitsraum, 41  
 $\mathcal{A}$ , reachable workspace, 79  
 $\mathbf{0}$ , Nullvektor, 11  
 $\omega$ , Winkelgeschwindigkeitsvektor, 62  
 $R(x; \theta_x)$ , 18  
 $R(y; \theta_y)$ , 18  
 $R(z; \theta_z)$ , 18
- ABA (Articulated Body Algorithm), 141  
acceleration, 104  
accuracy, 50  
actuator, 1, 164  
ankle joints, 52  
Anti-Windup, 174  
anziehend, 162  
Arbeitsraum, erreichbarer, 41  
Arbeitsraum, vollmanipulierbarer, 41  
Articulated Body Algorithm (ABA) , 141  
asymptotisch stabiler Orbit, 162  
atan2, 47  
attractive, 162  
attraktiv, 162  
Auflösungsgenauigkeit, 50
- backlash, 143  
Bahn, 56  
base frame, 30  
Baumstruktur, 97  
Bewegungsgleichungen, 98, 125  
Blockschaltbild, 165, 166  
Bodenkontaktpunkte, 154  
Bodenreaktionskräfte, 154, 161
- caster wheel, 87  
charakteristische Gleichung, 168  
Christoffel-Symbol, 127  
closed kinematic chains, 142  
closed loop control, 164  
Composite Rigid Body Algorithm (CRBA), 141  
configuration, 6  
control, 164  
control gain, 170  
control law partitioning, 171  
Coulomb friction, 144  
Coulombsche Reibung, 144  
CRBA (Composite Rigid Body Algorithm), 139, 141  
critically damped, 168
- dead reckoning, 85  
degressive Federkennlinie, 146  
Denavit-Hartenberg Konventionen, DH, 31  
Determinante einer  $(3 \times 3)$ -Matrix, 12  
DH (Denavit-Hartenberg), 31  
differential algebraic equations, 142  
differential-algebraisches Gleichungssystem, 142  
differentialgetriebenes Fahrzeug, 87  
DIR DYN, 97, 123, 138  
DIR JAC, 75  
DIR KIN, 39, 75  
DOF (degrees of freedom), 7  
Drake, 44  
Drallsatz, 104  
Drehgelenke, 29  
Drehvektor und Drehwinkel, 186  
Dynamik, 8, 96  
dynamisch stabiles Gehen, 153
- Einheitsvektoren, 14  
EL (Euler-Lagrange), 98  
elasticity, 143  
elastisches Beinmodell, 159

- Elastizität, 143, 145  
 embedded robotics, 1  
 Endeffektor  $E$ , 30  
 equations of motion, 98  
 Euler-Lagrangesche Differentialgleichungen, 127  
 Euler-Winkel, 21, 23  
 Eulersche Gleichung, 104  
 Fahrzeug-RAN, 86  
 Feder-Masse-System, 167  
 Federkennlinie, 146  
 Federkonstante, 146  
 Federkraft, 146  
 Federmoment, 146  
 Federrate, 146  
 Federsteifigkeit, 146  
 Federweg, 146  
 feedback control, 164  
 feedback linearization, 171  
 Feedback-Linearisierung, 171  
 feedforward control, 164  
 Festwertregelung, 170  
 force, 103  
 frame, 5, 30  
 Frame-Konzept, 16  
 friction, 143  
 Fußpunkt  $F$ , 30  
 Gehen, dynamisch stabiles, 153  
 Gehen, Feder-Masse-Modell, 160  
 Gehen, statisch stabiles, 153  
 Gelenk, 29  
 Gelenke, elastische, 147  
 Gelenke, starre, 28  
 Gelenkkoordinaten, 6  
 Gelenkvariable  $q$ , 30  
 Geschwindigkeitssteuerung, 75  
 Gierwinkel, 22  
 Glied, 29  
 Glieder, starre, 28  
 Glieder, elastische, 147  
 Hamiltonsches Prinzip, 127  
 Hauptträgheitsachsen, 100  
 Hauptträgheitsmomente, 100  
 hip joints, 52  
 holonome Zwangsbedingung, 86  
 homogene Transformationsmatrix  ${}^aT_b$ , 24  
 Hookesches Gesetz, 146  
 humanoide Roboter, 52, 54  
 humanoider Roboterarm, 54  
 humanoides Roboterbein, 52  
 Hüftgelenke, 52  
 ICC (instantaneous center of curvature), 85  
 ICR (instantaneous center of rotation), 85  
 ikfast, 49  
 Impulssatz, 104  
 inertia tensor, 99  
 inertial frame, 30  
 instantaneous center of curvature (ICC), 85  
 instantaneous center of rotation (ICR), 85  
 INV DYN, 97, 105, 122, 123, 142, 165  
 INV JAC, 75, 76  
 INV KIN, 40, 75  
 inverse Dynamik, 97, 114, 127  
 Inverse Kinematik, 8  
 inverse Matrix, 12  
 inverses Jacobi-Modell, 75  
 inverses Kinematikmodell, 40  
 inverses Kinematikproblem, 8  
 inverted pendulum, 157  
 invertiertes Pendel, 156, 157  
 Jacobi-Matrix, 57  
 Jacobi-Matrix eines Manipulators, 8, 66, 72  
 Jacobian, 57  
 joint, 29  
 joints, elastic, 147  
 joints, rigid, 28  
 Kardanwinkel, 24  
 Kaskadenregelung, 172, 174  
 KDL, 44  
 Kettenfahrzeug, 93  
 Kinematik, 6, 28  
 kinematisch redundant, 76, 77  
 kinematische Kette, 6, 28  
 kinematische Kette, geschlossene, 142  
 kinematische Kette, offene, 28  
 kinematische Schleife, 142  
 kinematische Singularität, 8, 79  
 kinetic energy, 125  
 Kinetik, 8, 96  
 kinetische Energie, 125  
 kinetische Energie, 125, 126  
 knee joint, 52

Knickarmroboter, 147  
 Kniegelenk, 52  
 Knöchelgelenke, 52  
 koinzidente Koordinatensysteme, 19  
 Konfiguration, 6  
 Koordinatensystem, Basis-, 30  
 Koordinatensystem, inertiales, 30  
 Kraftregelung, 9  
 Kreuzprodukt zweier Vektoren, 11  
 kritisch gedämpft, 168  
 KS (Koordinatensystem), 5, 30  
  
 Lagrangefunktion, 126  
 Lagrangian, 126  
 linear joints, 29  
 link, 29  
 links, elastic, 147  
 links, rigid, 28  
 Linkskoordinatensystem, 19  
 Lose, 50, 143  
  
 manipulability measure, 77  
 Manipulator Jacobian, 57, 72  
 Manipulierbarkeitsmaß, 77  
 mass matrix, 126  
 Massenmatrix, 123, 126  
 Massenträgheitsmomente, 99  
 Massenträgheitsprodukte, 99  
 Massenverteilung, 99  
 Matlab Symbolic Math Toolbox, 49  
 MBS dynamics, 97  
 MBSlib, 98, 114, 142  
 Mehrgrößenregelung, 166  
 Mehrkörpersystemdynamik, 97  
 menschliches Bein, 159  
 MIMO, 166  
 momentanes Drehzentrum, 85  
 Monodromiematrix, 163  
 Motornennmoment, 143  
 Motorspitzenmoment, 143  
 multibody systems dynamics, 97  
 Muskelmodell, 152  
  
 Nachmultiplikation, 21, 23  
 NE (Newton-Euler), 98  
 Newton-Verfahren, 43  
 Newtonsche Gleichung, 104  
 nicht-holome Zwangsbedingung, 86  
 Nickwinkel, 22

Nullstellung, 32, 36  
 Odometrie, 85  
 odometry, 85  
 ONB (Orthonormalitätsbedingungen), 19  
 open loop control, 164  
 optimal control problem, 86  
 optimales Steuerungsproblem, 86  
 Orbit, 162  
 orbital stability, 162  
 orbitale Stabilität, 162  
 Orocos, 44  
 orthogonale Matrix, 13  
 orthogonale Vektoren, 11  
 Orthonormalitätsbedingungen (ONB), 19  
 overdamped, 168  
  
 path, 56  
 PD-Regelung, 169  
 Pendel, 156  
 periodische Stabilität, 162  
 Pfad, 56  
 PID-Regelung, 173  
 Pieper, Kriterium von, 45  
 Pioneer P3-DX, 87  
 pitch angle, 22  
 Poincare map, 162  
 Poincare-Abbildung, 162  
 Pole, 168  
 pose, 86  
 Positionsgenauigkeit, 50  
 Positionsregelung, 9, 164  
 positive Grenzmenge, 162  
 positive limit set, 162  
 positiver Drehwinkel, 18  
 postural stability, 153  
 posturale Stabilität, 153  
 potential energy, 126  
 potentielle Energie, 125, 126, 147  
 principal moments of inertia, 100  
 Principle of Least Action, 127  
 Prinzip der kleinsten Wirkung, 127  
 prismatic joints, 29  
 progressive Federkennlinie, 146  
  
 Quaternionen, 24, 183  
  
 Radnormale, 85  
 RAN (räumliche Anordnung), 16

- Rechte Hand Regel, 14, 18  
 Rechtskoordinatensystem, 19  
 Regelabweichung, 166  
 Regeldifferenz, 166  
 Regelung, 164  
 Reglerparameter, 170  
 reguläre Matrix, 13  
 Reibung, 143  
 Rennen, 159  
 repeatability, 50  
 resolution, 50  
 revolute joints, 29  
 Richtungskosinusmatrix, 19  
 Roboterdynamik, 8, 96  
 Robotics Toolbox, 44  
 roll angle, 22  
 Roomba, 87  
 ROS, 44  
 $\text{Rot}(x; \theta_x)$ , 25  
 rotary joints, 29  
 Rotationsmatrix  ${}^aR_b$ , 19  
 RPY-Winkel, 22  
 Rückwärts Kinematik-Modell, 40  
 Ruhelage, 167  
 RWL, 40  
 RWL, Rückwärtlösung, 39  
 räumliche Anordnung (RAN), 16  
 Satz von Steiner, 101  
 SCARA (Selectively Compliant Assembly Robot Arm), 36  
 SCARA-Manipulator, 36, 41, 46, 73, 77, 80, 115, 128  
 schiefsymmetrisch, 60  
 Schubgelenke, 29  
 Segway, 87  
 set-point control, 170  
 set-point trajectory control, 172  
 set-point trajectory, 165  
 singular configuration, 79  
 singularity, 79  
 singuläre Konfiguration, 79  
 singuläre Matrix, 13  
 SISO, 166  
 Skalarprodukt, 11  
 skew symmetric, 60  
 skid, 94  
 skid steering, 94  
 slip, 94  
 Sollwert, 165  
 Sollwerttrajektorie, 165  
 Sollwerttrajektorien-Folgeregelung, 172  
 Spiegelung, 13  
 stabiler Orbit, 162  
 stability, orbital, 162  
 stability, postural, 153  
 Stabilität eines Regelkreises, 166  
 Stabilität, orbitale, 162  
 Stabilität, periodische, 162  
 Stabilität, posturale, 153  
 statisch stabiles Gehen, 153  
 Steiner, Satz von, 101  
 Steuerung, 164  
 TCP (tool center point), 34, 50  
 tool, 34  
 tool center point (TCP), 34, 50  
 tool frame, 34  
 torque, 103  
 trajectory, 56  
 Trajektorie, 56  
 $\text{Trans}(r_x, r_y, r_z)$ , 25  
 transponierte Matrix, 12  
 Trägheitstensor, 99, 100  
 Trägheitstensoren, Transformation von, 101  
 TurtleBot, 87  
 Uhrzeigersinn, 18  
 underactuated, 6  
 underdamped, 169  
 unteraktuiert, 6  
 unterkritisch gedämpft, 169  
 Variationsproblem, 127  
 Vektorprodukt zweier Vektoren, 11  
 Via-Punkte, 10  
 viscous friction, 144  
 viskose Gleitreibung, 144  
 Vormultiplikation, 21, 22  
 Vorwärtskinematik, 6, 28  
 Vorwärtsdynamik, 97  
 Vorwärtskinematik-Modell, 39  
 Vorwärtskinematikproblem, 6  
 VWL, Vorwärtlösung, 39  
 Wankwinkel, 22  
 Werkzeug, 34

Wiederholgenauigkeit, 50  
Windup, 174  
Winkelgeschwindigkeit eines Fahrzeugs, 88  
Winkelgeschwindigkeitsvektor, 62  
  
X-Y-Z, 22  
X-Y-Z Fixed Angles, 22  
  
yaw angle, 22  
  
Z-Y-X Euler-Winkel, 23  
Z-Y-Z, 22  
Z-Y-Z Euler-Winkel, 23  
zero position, 32  
Zero-Moment-Point (ZMP), 154  
ZMP (zero moment point), 154  
  
überkritisch gedämpft, 168