

Multi-View Geometry Based Visual Perception and Control of Robotic Systems



Jian Chen
Bingxi Jia • Kaixiang Zhang

*Multi-View Geometry
Based Visual Perception
and Control of Robotic
Systems*



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Jian Chen, Bingxi Jia, and Kaixiang Zhang

Multi-View Geometry Based Visual Perception and Control of Robotic Systems



CRC Press

Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

MATLAB[®] is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB[®] software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB[®] software.

Published in 2018 by CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2018 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-0-8153-6598-3 (hardback)

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

No part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access HYPERLINK “<http://www.copyright.com>” www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Names: Chen, Jian (Robotics engineer), author.
Title: Multi-view geometry based visual perception and control of robotic systems / Jian Chen.
Description: Boca Raton, FL : CRC Press/Taylor & Francis Group, 2017. | Includes bibliographical references and index.
Identifiers: LCCN 2018008526 | ISBN 9780815365983 (hb : acid-free paper) | ISBN 9780429489211 (ebook)
Subjects: LCSH: Robots—Control systems. | Computer vision.
Classification: LCC TJ211.35 .C44 2017 | DDC 629.8/92637—dc23
LC record available at <https://lccn.loc.gov/2018008526>

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Contents

Preface	xiii
Authors	xvii
PART I: FOUNDATIONS	1
1 Robotics	3
1.1 Pose Representation	3
1.1.1 Position and Translation	4
1.1.2 Orientation and Rotation	6
1.1.3 Homogeneous Pose Transformation	9
1.2 Motion Representation	10
1.2.1 Path and Trajectory	10
1.2.2 Pose Kinematics	13
1.3 Wheeled Mobile Robot Kinematics	15
1.3.1 Wheel Kinematic Constraints	15
1.3.2 Mobile Robot Kinematic Modeling	17
1.3.3 Typical Nonholonomic Mobile Robot	18
2 Multiple-View Geometry	21
2.1 Projective Geometry	22
2.1.1 Homogeneous Representation of Points and Lines	22
2.1.2 Projective Transformation	22
2.2 Single-View Geometry	24
2.2.1 Pinhole Camera Model	25
2.2.2 Camera Lens Distortion	26

2.3	Two-View Geometry	26
2.3.1	Homography for Planar Scenes	27
2.3.2	Epipolar Geometry for Nonplanar Scenes	31
2.3.3	General Scenes	32
2.4	Three-View Geometry	34
2.4.1	General Trifocal Tensor Model	34
2.4.2	Pose Estimation with Planar Constraint	36
2.5	Computation of Multiple-View Geometry	37
2.5.1	Calibration of Single-View Geometry	38
2.5.2	Computation of Two-View Geometry	39
2.5.2.1	Computation of Homography	39
2.5.2.2	Computation of Epipolar Geometry	40
2.5.3	Computation of Three-View Geometry	40
2.5.3.1	Direct Linear Transform	41
2.5.3.2	Matrix Factorization	41
2.5.4	Robust Approaches	43
3	Vision-Based Robotic Systems	47
3.1	System Overview	48
3.1.1	System Architecture	48
3.1.2	Physical Configurations	49
3.2	Research Essentials	50
	PART II: VISUAL PERCEPTION OF ROBOTICS	53
4	Introduction to Visual Perception	55
4.1	Road Reconstruction and Detection for Mobile Robots	55
4.1.1	Previous Works	55
4.1.2	A Typical Vehicle Vision System	59
4.1.2.1	System Configuration	59
4.1.2.2	Two-View Geometry Model	60
4.1.2.3	Image Warping Model	61
4.1.2.4	Vehicle-Road Geometric Model	61
4.1.2.5	More General Configurations	63
4.2	Motion Estimation of Moving Objects	63
4.3	Scaled Pose Estimation of Mobile Robots	65
4.3.1	Pose Reconstruction Based on Multiple-View Geometry	65
4.3.2	Dealing with Field of View Constraints	67
4.3.2.1	Key Frame Selection	68
4.3.2.2	Pose Estimation	69
4.3.3	Dealing with Measuring Uncertainties	71

4.3.3.1	Robust Image Feature Extraction	71
4.3.3.2	Robust Observers	72
4.3.3.3	Robust Controllers	72
4.3.3.4	Redundant Degrees of Freedom	72
5	Road Scene 3D Reconstruction	73
5.1	Introduction	73
5.2	Algorithm Process	74
5.3	3D Reconstruction	75
5.3.1	Image Model	75
5.3.2	Parameterization of Projective Parallax	76
5.3.3	Objective Function Definition and Linearization	78
5.3.4	Iterative Maximization	80
5.3.5	Post Processing	81
5.4	Road Detection	81
5.4.1	Road Region Segmentation	81
5.4.2	Road Region Diffusion	81
5.5	Experimental Results	84
5.5.1	Row-Wise Image Registration	84
5.5.2	Road Reconstruction	87
5.5.3	Computational Complexity	89
5.5.4	Evaluation for More Scenarios	93
6	Recursive Road Detection with Shadows	95
6.1	Introduction	95
6.2	Algorithm Process	96
6.3	Illuminant Invariant Color Space	97
6.3.1	Imaging Process	97
6.3.2	Illuminant Invariant Color Space	98
6.3.3	Practical Issues	98
6.4	Road Reconstruction Process	99
6.4.1	Image Fusion	99
6.4.2	Geometric Reconstruction	99
6.4.2.1	Geometric Modeling	99
6.4.3	Road Detection	103
6.4.4	Recursive Processing	104
6.5	Experiments	105
6.5.1	Illuminant Invariant Transform	105
6.5.2	Road Reconstruction	106
6.5.3	Comparisons with Previous Methods	110
6.5.4	Other Results	113

7	Range Identification of Moving Objects	117
7.1	Introduction	117
7.2	Geometric Modeling	118
7.2.1	Geometric Model of Vision Systems	118
7.2.2	Object Kinematics	120
7.2.3	Range Kinematic Model	121
7.3	Motion Estimation	122
7.3.1	Velocity Identification of the Object	122
7.3.2	Range Identification of Feature Points	122
7.4	Simulation Results	124
7.5	Conclusions	125
8	Motion Estimation of Moving Objects	127
8.1	Introduction	127
8.2	System Modeling	127
8.2.1	Geometric Model	127
8.2.2	Camera Motion and State Dynamics	129
8.3	Motion Estimation	131
8.3.1	Scaled Velocity Identification	131
8.3.2	Range Identification	134
8.4	Simulation Results	137
8.5	Conclusion	139
PART III: VISUAL CONTROL OF ROBOTICS		141
9	Introduction to Visual Control	143
9.1	Previous Works	143
9.1.1	Classical Visual Control Approaches	143
9.1.1.1	Position Based Methods	143
9.1.1.2	Image Based Methods	144
9.1.1.3	Multiple-View Geometry Based Method	144
9.1.2	Main Existing Problems	146
9.1.2.1	Model Uncertainties	146
9.1.2.2	Field of View Constraints	148
9.1.2.3	Nonholonomic Constraints	149
9.2	Task Descriptions	150
9.2.1	Autonomous Robot Systems	150
9.2.2	Semiautonomous Systems	151

9.3	Typical Visual Control Systems	152
9.3.1	Visual Control for General Robots	152
9.3.2	Visual Control for Mobile Robots	154
10	Visual Tracking Control of General Robotic Systems	157
10.1	Introduction	157
10.2	Visual Tracking with Eye-to-Hand Configuration	158
10.2.1	Geometric Modeling	158
10.2.1.1	Vision System Model	158
10.2.1.2	Euclidean Reconstruction	159
10.2.2	Control Development	160
10.2.2.1	Control Objective	160
10.2.2.2	Open-Loop Error System	163
10.2.2.3	Closed-Loop Error System	164
10.2.2.4	Stability Analysis	165
10.3	Visual Tracking with Eye-in-Hand Configuration	166
10.3.1	Geometric Modeling	166
10.3.2	Control Development	167
10.3.2.1	Open-Loop Error System	167
10.3.2.2	Controller Design	168
10.4	Simulation Results	168
10.5	Conclusion	180
11	Robust Moving Object Tracking Control	181
11.1	Introduction	181
11.2	Vision System Model	182
11.2.1	Camera Geometry	182
11.2.2	Euclidean Reconstruction	182
11.3	Control Development	184
11.3.1	Open-Loop Error System	184
11.3.2	Control Design	185
11.3.3	Closed-Loop Error System	186
11.4	Stability Analysis	187
11.4.1	Convergence of the Rotational Error	187
11.4.2	Convergence of the Translational Error	190
11.5	Simulation Results	192
11.5.1	Simulation Configuration	192
11.5.2	Simulation Results and Discussion	193
11.6	Conclusion	197

12	Visual Control with Field-of-View Constraints	201
12.1	Introduction	201
12.2	Geometric Modeling	202
12.2.1	Euclidean Homography	202
12.2.2	Projective Homography	204
12.2.3	Kinematic Model of Vision System	205
12.3	Image-Based Path Planning	206
12.3.1	Pose Space to Image Space Relationship	206
12.3.2	Desired Image Trajectory Planning	207
12.3.3	Path Planner Analysis	209
12.4	Tracking Control Development	210
12.4.1	Control Development	210
12.4.2	Controller Analysis	211
12.5	Simulation Results	213
12.5.1	Optical Axis Rotation	214
12.5.2	Optical Axis Translation	218
12.5.3	Camera y -Axis Rotation	220
12.5.4	General Camera Motion	220
12.6	Conclusions	224
13	Visual Control of Mobile Robots	225
13.1	Introduction	225
13.2	Geometric Reconstruction	226
13.2.1	Eye-to-Hand Configuration	226
13.2.2	Eye-in-Hand Configuration	229
13.2.2.1	Geometric Modeling	229
13.2.2.2	Euclidean Reconstruction	230
13.3	Control Development for Eye-to-Hand Configuration	233
13.3.1	Control Objective	233
13.3.2	Open-Loop Error System	234
13.3.3	Closed-Loop Error System	235
13.3.4	Stability Analysis	236
13.3.5	Regulation Extension	237
13.4	Control Development for Eye-in-Hand Configuration	239
13.4.1	Open-Loop Error System	239
13.4.2	Closed-Loop Error System	241
13.4.3	Stability Analysis	241
13.5	Simulation and Experimental Verifications	243
13.5.1	Eye-to-Hand Case	243
13.5.2	Eye-in-Hand Case	244
13.5.2.1	Experimental Configuration	244

13.5.2.2	Experimental Results	247
13.5.2.3	Results Discussion	249
13.6	Conclusion	253
14	Trifocal Tensor Based Visual Control of Mobile Robots	255
14.1	Introduction	255
14.2	Geometric Modeling	256
14.3	Control Development	257
14.3.1	Error System Development	258
14.3.2	Controller Design	260
14.3.3	Stability Analysis	261
14.4	Simulation Verification	263
14.4.1	Pose Estimation	264
14.4.2	Visual Trajectory Tracking and Pose Regulation	268
14.4.3	Trajectory Tracking with Longer Range	271
14.5	Conclusion	273
15	Unified Visual Control of Mobile Robots with Euclidean Reconstruction	275
15.1	Introduction	275
15.2	Control Development	276
15.2.1	Kinematic Model	276
15.2.2	Open-Loop Error System	277
15.2.3	Controller Design	278
15.2.4	Stability Analysis	279
15.3	Euclidean Reconstruction	282
15.4	Simulation Results	283
15.5	Conclusion	286
PART IV:	APPENDICES	287
Appendix A:	Chapter 7	289
A.1	Proof of Theorem 7.1	289
Appendix B:	Chapter 10	291
B.1	291
B.2	291
B.3	292
Appendix C:	Chapter 11	295
C.1	295
C.2	Proof of Property 11.1	296

C.3	Proof of Lemma 11.1	297
C.4	Proof of Property 11.2	298
Appendix D: Chapter 12		301
D.1	Open-Loop Dynamics	301
D.2	Image Jacobian-Like Matrix	301
D.3	Image Space Navigation Function	302
Appendix E: Chapter 13		305
Appendix F: Chapter 14		307
F.1	307
F.2	Proof of $\frac{1}{2} \ln(\bar{e}_1^2 + \mu) > \frac{\bar{e}_1}{\bar{e}_1^2 + \mu}$ for $\mu > 1.5$ and $\bar{e}_1 \in \mathbb{R}$	307
F.3	Proof of $\rho(t)e_3(t) \in \mathcal{L}_1$	308
F.4	Proof of $e_3(t) \in \mathcal{L}_1$ in the Closed-Loop System (14.19)	308
References		311
Index		331

Preface

Over the past decade, there has been a rapid development in the vision-based perception and control of robotic systems. Especially, multiple-view geometry is utilized to extract low-dimensional geometric information from abundant and high-dimensional image space, making it convenient to develop general solutions for robot perception and control tasks. This book aims to describe possible frameworks for setting up visual perception and control problems that need to be solved in the context of robotic systems.

The visual perception of robots provides necessary feedback for control systems, such as robot pose information, object motion information, and drivable road information. Since 3D information is lost and image noise exists in the imaging process, the effective pose estimation and motion identification of objects are still challenging. Besides, mobile robots are generally faced with complex scenes making it difficult to robustly detect drivable road space for safe operation. In this book, multiple-view geometry is exploited to describe the scene structure and maps from image space to Euclidean space. Optimization and estimation theories are applied to reconstruct the geometric information of the scene. Then, it is convenient to identify the real-time states of the robot and objects and to detect drivable road region based on geometric information.

The visual control of robots exploits visual information for task description and controls the robots through appropriate control laws using visual feedback. Since depth information is lost in the imaging process of monocular cameras, there exist model uncertainties in the control process. Besides, the limited field of view of the camera and the physical constraints (e.g., nonholonomic constraints) of the robots also have great influences on the stability and robustness of the control process. In this book, multiple-view geometry is used for geometric modeling and scaled pose estimation. Then Lyapunov methods are applied to design stabilizing control laws in the presence of model uncertainties and multiple constraints.

The book is divided into four parts.

- Part I consists of Chapter 1–3. This part is more tutorial than the others and introduces the basic knowledge of the robotics and the multiple-view geometry.
- In Part II, visual perception of robotics is presented. Specifically, Chapter 4 gives an overview of the perception problems. Chapter 5 describes a road construction approach based on iterative optimization and two-view geometry. Considering the complex lighting conditions, an illuminant invariant transform method is presented in Chapter 6 to eliminate the effects of shadows and recover the textures in shadows. After that, a nonlinear estimation strategy is proposed in Chapter 7 to identify the velocity and range of a moving object with a static monocular camera. To eliminate the motion constraint and prior geometric knowledge of the moving object, a static-moving camera system is exploited in Chapter 8 to achieve the velocity and range identification tasks.
- Visual control of robotics is given in Part III. First, Chapter 9 introduces the visual control problems of a robotic system. Then, for the general fully actuated robots with six DOFs, the trajectory tracking control is considered in Chapter 10 for both eye-in-hand and eye-to-hand configurations. In the presence of the system uncertainties, a robust control law is designed in Chapter 11 to asymptotic tracking of a moving object. Considering the field-of-view constraints, i.e., the target should remain visible in the camera view, a trajectory planning method based on navigation function is proposed in Chapter 12 to generate a desired trajectory, moving the camera from the initial pose to a goal pose while ensuring all the feature points remain visible. Besides, for the wheeled mobile robots, adaptive control laws are presented in Chapter 13 to achieve the trajectory tracking and pose regulation tasks, respectively. In Chapter 14, with the existence of unknown camera installing position, a trifocal tensor based controller is designed to address the trajectory tracking and regulation problems. Moreover, to identify the depth information during the control process, a unified controller with online depth estimation is developed in Chapter 15.
- Appendices are provided in Part IV to describe further details of Parts II and III.

The content in Parts II and III of this book (unless noted otherwise) has been derived from the authors' research work during the past several years in the area of visual perception and control. This book is aimed at researchers who are interested in the application of multiple-view geometry and Lyapunov-based techniques to emerging problems in perception and control of robotics.

The authors are grateful to the National Natural Science Foundation of China (Grant 61433013) for supporting our research on control of fuel cell intelligent vehicles. We would like to thank the colleagues and collaborators for their valuable contributions and support, without them this work would not have been possible. We would also like to acknowledge the support from the following members of the Sustainable Energy and Intelligent Vehicles Laboratory in the College of Control Science and Engineering at Zhejiang University: Xinfang Zhang, Qi Wang, Yanyan Gao, Yang Li, and Guoqing Yu.

If the reader finds any errors in the book, please send electronically to

jchen@zju.edu.cn

which will be sincerely appreciated. An up-to-date errata list will be provided at the homepage of the book:

<http://person.zju.edu.cn/jchen/714344.html>

MATLAB[®] is a registered trademark of The MathWorks, Inc. For product information, please contact:

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 USA
Tel: 508-647-7000
Fax: 508-647-7001
E-mail: info@mathworks.com
Web: www.mathworks.com



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Authors

Jian Chen received his BE and ME degrees from Zhejiang University, Hangzhou, China, in 1998 and 2001, respectively, and his PhD degree in electrical engineering from Clemson University, Clemson SC, in 2005. He was a research fellow in the University of Michigan, Ann Arbor, MI from 2006 to 2008, where he worked on fuel cell modeling and control. He joined IdaTech LLC, Bend, OR, in 2008, where he worked on fuel cell back power systems. Later in 2012, he joined Proterra Inc., Greenville, SC, where he was involved in the National Fuel Cell Bus Program. In 2013, he went back to academia and joined the College of Control Science and Engineering, Zhejiang University as a full professor. Moreover, he was recruited by the Chinese Recruitment Program of Global Youth Experts. The research interests of Jian Chen include visual servo techniques, control of fuel cell vehicles, battery management, and nonlinear control.

Bingxi Jia received his BE and PhD degrees both in control science and engineering from the College of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2012 and 2017, respectively. He joined Zhuying Tech. Inc., Hangzhou, China, in 2017, serving as the senior engineer responsible for the motion planning and control of autonomous vehicles. His research interests include vision-based control, computer vision, and motion planning for intelligent vehicles.

Kaixiang Zhang received his BE degree in automation from Xiamen University, Xiamen, China, in 2014. He is currently pursuing his PhD degree in the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. He is a visiting student in the National Institute for Research in Computer Science and Control, Rennes, France. His current research interests include vision-based control and vision-based estimation.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

FOUNDATIONS

I



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Chapter 1

Robotics

Robotics is concerned with the study of those machines that can replace human beings in the execution of a task, including physical activity, decision-making, and human interaction. This book is devoted to the studies on the robots that accomplish the physical tasks via locomotion. Generally, these types of robots can be divided into robot manipulators and mobile robots. Robot manipulators are also called fixed robots as they are generally mounted in industrial fields and the motion of end effectors are actuated by the motion of joints. The end effectors generally move in 3D space to perform manipulation tasks. Mobile robots are featured with the free mobility in large workspace, including wheeled mobile robots (WMRs), legged mobile robots, unmanned aerial vehicles, and autonomous underwater vehicles. WMRs are widely used in both industrial and daily life scenarios because they are appropriate for typical applications with relatively low mechanical complexity and energy consumption [107].

The key feature of a physical robot is its locomotion via specific mechanical structure. For control development, it should be modeled mathematically to describe its motion characteristics. In the autonomous navigation and manipulation tasks, the motion of robot bases and end effectors can be expressed mathematically by the notion of pose information. In this chapter, the general pose representation of rigid bodies is introduced, and then the robot kinematics is developed.

1.1 Pose Representation

A rigid body in physical space can be described by the position and orientation, which are collectively named as the pose information. Then, the robot

locomotion can be described by the pose information, which serves as the output of the robot model. The motion of a rigid body consists of translations and rotations, resulting in the position and orientation descriptions with respect to the reference coordinate system. As shown in Figure 1.1, two coordinate frames \mathcal{F} and \mathcal{F}' exist in 3D Euclidean space. The motion from \mathcal{F} to \mathcal{F}' can be described by a translation and a rotation. Then, the pose of frame \mathcal{F}' with respect to frame \mathcal{F} can be described by the position and the orientation.

1.1.1 Position and Translation

In 3D Euclidean space, the position of the origin of coordinate frame \mathcal{F}' relative to coordinate frame \mathcal{F} can be denoted by the following: 3×1 vector $[x \ y \ z]^T$. The components of this vector are the Cartesian coordinates of \mathcal{F}' in the \mathcal{F} frame, which are the projections of the vector x_f onto the corresponding axes. Besides the Cartesian coordinate system, the position of a rigid body can also be expressed in spherical or cylindrical coordinates. Such representations are generally used for the analysis of specific mechanisms such as the spherical and cylindrical robot joints as well as the omnidirectional cameras. As shown in Figure 1.2, the spherical coordinate system can be viewed as the three-dimensional version of the polar coordinate system, and the position of a point is specified by three numbers: the radial distance r from the point to the coordinate origin, its polar angle θ measured from a fixed zenith direction, and the azimuth angle ϕ of its orthogonal projection on a reference plane that passes through the origin and is orthogonal to the zenith. As shown in Figure 1.3, the position of a point in the cylindrical coordinate system is specified by three numbers: the distance ρ from the point to the chosen reference axis (generally called cylindrical or longitudinal axis), the direction angle ϕ from the axis relative to a chosen reference direction, and the distance z from a chosen reference plane

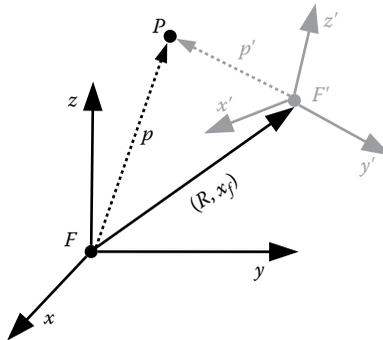


Figure 1.1: Pose representation.

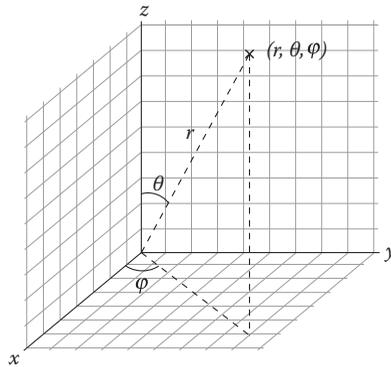


Figure 1.2: Spherical coordinate system.

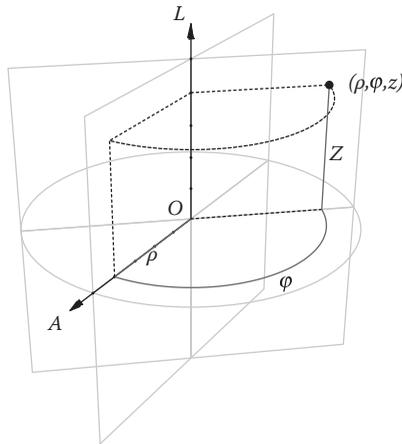


Figure 1.3: Cylindrical coordinate system.

perpendicular to the axis. The latter distance is given as a positive or negative number depending on which side of the reference plane faces the point.

A translation is a displacement in which no point in the rigid body remains in its initial position and all straight lines in the rigid body remain parallel to their initial orientations. The translation of a body in 3D Euclidean space can be represented by the following 3×1 vector:

$$x_f = \begin{bmatrix} x_{fx} \\ x_{fy} \\ x_{fz} \end{bmatrix}. \quad (1.1)$$

Conversely, the position of a body can be represented as a translation that takes the body from a starting position (the origin of frame \mathcal{F}) to the current position (the origin of frame \mathcal{F}'). As a result, the representation of position can be used to express translation and vice versa.

1.1.2 Orientation and Rotation

The orientation expresses the axis directions of frame \mathcal{F}' with respect to frame \mathcal{F} . A rotation is a displacement that the origin of frame \mathcal{F}' coincides with the origin of frame \mathcal{F} and not all the axes of \mathcal{F}' parallel to those of \mathcal{F} . As in the case of position and translation, any representation of orientation can be used to create a representation of rotation and vice versa. In the following descriptions in this book, the positive direction of rotational angle is defined by the right-hand rule, as indicated in Figure 1.4.

Rotation matrix

The orientation of frame \mathcal{F}' with respect to frame \mathcal{F} can be denoted by expressing the basis vectors $(\vec{x}', \vec{y}', \vec{z}')$ (the axis directions of \mathcal{F}') in terms of the basis vectors $(\vec{x}, \vec{y}, \vec{z})$ (the axis directions of \mathcal{F}). The rotation matrix R is used to express the rotation from \mathcal{F} to \mathcal{F}' . The components of R are the dot products of the basis vectors of the two coordinate frames:

$$R = \begin{bmatrix} \vec{x} \cdot \vec{x}' & \vec{y} \cdot \vec{x}' & \vec{z} \cdot \vec{x}' \\ \vec{x} \cdot \vec{y}' & \vec{y} \cdot \vec{y}' & \vec{z} \cdot \vec{y}' \\ \vec{x} \cdot \vec{z}' & \vec{y} \cdot \vec{z}' & \vec{z} \cdot \vec{z}' \end{bmatrix}. \quad (1.2)$$

Since the basis vectors are unit vectors, the dot product of any two unit vectors is actually the cosine of the angle between them, the components are commonly referred to as direction cosines.

The rotation matrix R contains nine elements but only three Degrees of Freedom (DOF) to express the rotation and orientation in the 3D Euclidean space.

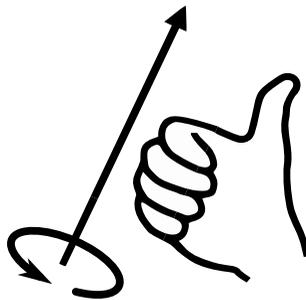


Figure 1.4: The right-hand rule for rotation.

Therefore, six auxiliary relationships exist among the elements of the matrix. Because the basis vectors of coordinate frame \mathcal{F} are mutually orthonormal, as are the basis vectors of coordinate frame \mathcal{F}' , the columns of R formed from the dot products of these vectors are also mutually orthonormal. The rotation matrix is an orthogonal matrix and has the property that its inverse is simply its transpose. This property provides the six auxiliary relationships. Three require the column vectors to have unit length, and three require the column vectors to be orthogonal. Besides, the orthogonality of the rotation matrix can be seen from the expression in (1.2) by considering the frames in reverse order. The orientation of coordinate frame \mathcal{F}' relative to coordinate frame \mathcal{F} is the rotation matrix whose rows are clearly the columns of the matrix R .

Using the above definition, an elementary rotation of frame \mathcal{F}' about the z axis through an angle θ is

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1.3)$$

and the same rotation about the y axis is

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \quad (1.4)$$

and the same rotation about the x axis is

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}. \quad (1.5)$$

Three angle representations

As mentioned above, the minimum representation of a rotation in 3D Euclidean space is of three numbers. As a result, a common representation for the orientation of coordinate frame \mathcal{F}' with respect to coordinate frame \mathcal{F} can be denoted as a vector of three angles. Then, the overall rotation can be decomposed into a sequence of rotations by the three angles about specific axes. For example, the rotation matrix R can be expressed by the multiplication of the three matrices $R = R_1 R_2 R_3$ with the rotational order R_1, R_2, R_3 . Two commonly used angle representations are the Euler angles and the fixed angles, as described as follows:

- **Euler angles:** The Euler angles have three components; each represents a rotation about the axes of a moving coordinate frame (i.e., frame \mathcal{F}'). As a result, the rotation matrix depends on the order of rotations. In this book, the indication of Euler angles (α, β, γ) expresses the rotation order $Z - Y - X$. Assuming the frames \mathcal{F} and \mathcal{F}' are initially coincident, the

3D rotation can be decomposed into three steps: rotate angle α about the z axis of frame \mathcal{F}' , rotate angle β about the y axis of frame \mathcal{F}' , and then rotate angle γ about the z axis of frame \mathcal{F}' . The rotational matrix can be expressed by the Euler angles as follows:

$$R = \begin{bmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{bmatrix} \quad (1.6)$$

Based on the above expression, the Euler angles can be decomposed by the rotation matrix as follows:

$$\begin{aligned} \beta &= \text{atan2} \left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2} \right) \\ \alpha &= \text{atan2} \left(\frac{r_{21}}{\cos \beta}, \frac{r_{11}}{\cos \beta} \right) \\ \gamma &= \text{atan2} \left(\frac{r_{32}}{\cos \beta}, \frac{r_{33}}{\cos \beta} \right). \end{aligned} \quad (1.7)$$

- **Fixed angles:** Except for the Euler angles, a vector of three angles can also denote the rotational relationship, with each angle representing a rotation about an axis of a fixed frame. As shown in Figure 1.1, \mathcal{F} is the fixed frame and \mathcal{F}' is the moving frame. Take these two frames to be initially coincident, ψ is the yaw rotation about the x axis of \mathcal{F} , θ is the pitch rotation about the y axis of \mathcal{F} , and ϕ is the roll rotation about the z axis of \mathcal{F} . Actually, a set of $X-Y-Z$ fixed angles is equivalent to the same set of $Z-Y-X$ Euler angles ($\alpha = \phi$, $\beta = \theta$, and $\gamma = \psi$).

The representations using three angles are intuitive but suffer from singularity. The singularity occurs when the rotational axis of the second term in the sequence becomes parallel to the rotational axis of the first or the third term. In that case, there are only two effective rotational axes instead of the original three axes, i.e., one DOF is lost. This issue occurs when $\beta = \pm \frac{\pi}{2}$ for Euler angles and $\theta = \pm \frac{\pi}{2}$ for fixed angles.

Angle-vector representation

Different from the three angle representations that decompose the overall rotation into three sequential rotations along specific axes, the angle-vector representation expresses the rotation by a rotational angle θ about a rotational vector \mathbf{v} . The rotation matrix is expressed by the angle-vector representation by the Rodrigues' rotation formula as follows:

$$R = I_{3 \times 3} + \sin \theta [\mathbf{v}]_{\times} + (1 - \cos \theta)(\mathbf{v}\mathbf{v}^T - I_{3 \times 3}), \quad (1.8)$$

where $[\mathbf{v}]_{\times}$ is the skew-symmetric matrix defined as

$$[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}. \quad (1.9)$$

The rotation angle and vector are encoded in the eigenvalues and eigenvectors of the rotation matrix R . The orthonormal rotation matrix R has one real eigenvalue $\lambda = 1$ and two complex eigenvalues $\lambda = \cos \theta \pm i \sin \theta$. According to the definition,

$$R\mathbf{v} = \lambda\mathbf{v}, \quad (1.10)$$

where \mathbf{v} is the eigenvector corresponding to the eigenvalue λ . For the case of $\lambda = 1$,

$$R\mathbf{v} = \mathbf{v}, \quad (1.11)$$

which implies that the vector is not changed by the rotation. Actually, the rotational axis is the only vector in 3D space satisfying this feature. As a result, the eigenvector corresponding to the eigenvalue $\lambda = 1$ is the rotational vector \mathbf{v} .

Unit quaternion representation

The quaternions are actually the extension of complex numbers and were first described by W. R. Hamilton. The concept of quaternions is widely applied to robotics and mechanics in 3D space. A quaternion \mathcal{E} is written as a scalar \mathcal{E}_0 plus a vector \mathcal{E}_v as follows:

$$\begin{aligned} \mathcal{E} &= \mathcal{E}_0 + \mathcal{E}_v \\ &= \mathcal{E}_0 + \mathcal{E}_1 \cdot i + \mathcal{E}_2 \cdot j + \mathcal{E}_3 \cdot k, \end{aligned} \quad (1.12)$$

where $\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2$, and \mathcal{E}_3 are scalars, and i, j , and k are operators satisfying the following rules:

$$\begin{aligned} ii &= jj = kk = -1 \\ ij &= k, jk = i, ki = j \\ ji &= -k, kj = -i, ik = -j. \end{aligned} \quad (1.13)$$

The unit quaternion has the property that $\mathcal{E}_0^2 + \mathcal{E}_1^2 + \mathcal{E}_2^2 + \mathcal{E}_3^2 = 1$. The unit quaternion can be used to express a rotation of angle θ about the unit rotational vector \mathbf{v} as follows:

$$\mathcal{E}_0 = \cos \frac{\theta}{2}, \mathcal{E}_v = \sin \frac{\theta}{2} \cdot \mathbf{v}. \quad (1.14)$$

1.1.3 Homogeneous Pose Transformation

The preceding sections have addressed representations of position and orientation separately. The motion in 3D Euclidean space can be decomposed into a translation and a rotation. As shown in Figure 1.1, due to the effect of motion,

there exists a transformation between the coordinates of point P with respect to frames \mathcal{F} and \mathcal{F}' . Denote the coordinate of P with respect to frame \mathcal{F} and \mathcal{F}' as p and p' , respectively; the transformation from p' to p is defined as follows:

$$p = R \cdot p' + x_f, \quad (1.15)$$

where R and x_f are the rotation and translation from frame \mathcal{F} to frame \mathcal{F}' expressed in \mathcal{F} . To be compact, define the homogeneous coordinates of point P with respect to frames \mathcal{F} and \mathcal{F}' as \bar{p} and \bar{p}' , respectively, as follows:

$$\bar{p} = [x \ y \ z \ 1]^T, \bar{p}' = [x' \ y' \ z' \ 1]^T. \quad (1.16)$$

Then, the homogeneous transformation can be rewritten as the following:

$$\bar{p} = \begin{bmatrix} R & x_f \\ 0_{1 \times 3} & 1 \end{bmatrix} \bar{p}' = T \bar{p}', \quad (1.17)$$

where T is the 4×4 homogeneous transformation.

1.2 Motion Representation

In the previous section, the pose representation of rigid bodies in 3D Euclidean space is introduced. This section extends the concepts to the moving objects whose pose is varying, which is a basic description for robots. For robots, the tasks are generally accomplished by following specific paths or trajectories, or by regulating to specific target poses. In this section, the general concepts of path and trajectory are introduced for task description; then, the pose kinematics are described for motion analysis and further control development of robots.

1.2.1 Path and Trajectory

In robotics, a path is a spatial concept in space that leads from an initial pose to a final pose. A trajectory is a path with specified time constraints, i.e., it describes the poses at each specified time instants.

Parameterization

Due to the nature of trajectory, it is straightforward to parameterize the trajectory with time. Then, a trajectory can be denoted as $\mathbf{X}(t)$, where t is the time and \mathbf{X} is the vector representing pose information. For the parameterization of the path, there should be a parameter that varies smoothly and differs everywhere along the path. A commonly used parameter is the arc length accumulating from the start pose. Then, a path can be expressed as $\mathbf{X}(s)$, where s denotes the arc length.

The pose information \mathbf{X} includes position and orientation information. For robots moving in 3D Euclidean space, the pose information includes three DOF position and three DOF orientation. For mobile robots moving on planes, the

pose information includes two DOF position and one DOF orientation. For example, a common definition of six DOF pose vector is

$$\mathbf{X} = [x \ y \ z \ \psi \ \theta \ \phi]^T, \quad (1.18)$$

where the coordinates x, y, z denote the position and the angles ψ (yaw angle), θ (pitch angle), ϕ (roll angle) represent the orientation.

Due to the existence of kinematic constraints, the physically feasible trajectories are only a subset of the entire space. For a robot end effector that moves freely with six DOF, the elements in the pose vector vary independently from each other. While for a differential driving wheeled mobile robot, the variation of position is determined by the current orientation; thus, not all trajectories are feasible for this type of robots.

Curve fitting

For the application of control development, the desired path and trajectory generally need to be continuous, i.e., the position, velocity, and acceleration are required to be continuous.

For the case of pose regulation, the initial and final poses are given as \mathbf{X}_0 and \mathbf{X}^* . The polynomial function is an obvious candidate for curve fitting to parameterize paths or trajectories. For the general case that the position, velocity, and acceleration are given as initial and final conditions, fifth order polynomial function is the minimal structure to satisfy the conditions. For a robot without kinematic constraints (moves freely with six DOF), the elements in the pose vector \mathbf{X} can be designed separately. For example, take one dimension x ; the initial conditions $x_0, \dot{x}_0, \ddot{x}_0$ at t_0 and the final conditions $x^*, \dot{x}^*, \ddot{x}^*$ at t^* are given; then, the trajectory can be fitted by the function $x(t) = c_5t^5 + c_4t^4 + c_3t^3 + c_2t^2 + c_1t + c_0$, and the coefficients are determined by solving the least square solution of the following equation:

$$\begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\ 10t_0^3 & 12t_0^2 & 6t_0 & 2 & 0 & 0 \\ t^{*5} & t^{*4} & t^{*3} & t^{*2} & t^* & 1 \\ 5t^{*4} & 4t^{*3} & 3t^{*2} & 2t^* & 1 & 0 \\ 10t^{*3} & 12t^{*2} & 6t^* & 2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ x^* \\ \dot{x}^* \\ \ddot{x}^* \end{bmatrix} \quad (1.19)$$

For the robots with kinematic constraints, the constraints among the elements in the pose information should be considered in the curve fitting. For example, take the differential driving wheeled mobile robot; it has two position DOFs (x, y) and one orientation DOF θ . It has the feature that the direction of instantaneous velocity coincides with the orientation, i.e., $\theta = \text{atan2}(\dot{y}, \dot{x})$. As a result, only the trajectories of x and y need to be designed, and the trajectory of θ can be calculated accordingly.

For the case of path following or trajectory tracking, a set of points are generally given to represent the desired path or trajectory. For control development, a sufficient smooth curve representation is required, i.e., the first and second derivatives exist and are bounded. High order polynomial functions can be used for curve fitting by solving the least square solution of conditions formed by the points. However, the numerical stability decreases with higher order and the fitting accuracy decreases with lower order. Besides, the Runge's phenomenon exists which is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points. It shows that going to higher order does not always improve accuracy of curve fitting. Considering the flexibility and accuracy of curve fitting, the spline curves can be used for representation. A spline is a special function defined piecewise by polynomials. It uses low order polynomials in each interval and ensures the continuity between the intervals. Considering the fundamental case of one-dimensional fitting, a set of points $\{x_i\}_{i \in [1, N]}$ at time instants $\{t_i\}_{i \in [1, N]}$ are given. There are various forms of spline functions that can be used for curve fitting. Without generality, the following two cubic spline functions are used:

- One type of spline function directly uses the values at each data points for interpolation, and the resulting spline curve passes the data points. The spline curve is composed of a series of piecewise cubic polynomials $\{f_i\}_{i \in [1, N-1]}$ with $f_i : [t_i, t_{i+1}] \rightarrow \mathbb{R}$ defined as the following:

$$f_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + x_i. \quad (1.20)$$

To ensure the smoothness, the following boundary conditions are required:

$$\begin{aligned} f_i(t_{i+1}) &= x_{i+1} \\ \dot{f}_{i-1}(t_i) &= \dot{f}_i(t_i) \\ \ddot{f}_{i-1}(t_i) &= \ddot{f}_i(t_i). \end{aligned} \quad (1.21)$$

Then, the curve parameters in each interval can be solved by the above boundary conditions.

- Another type of spline function interpolates the function values based on the values at control points $\{\phi_j\}_{j \in [-1, M-2]}$, which are defined uniformly in the region of time horizon. The distribution of control points is generally sparser than the raw data points ($M < N$) to be smoother and more computationally efficient. The interval between neighboring control points is $\kappa = \frac{t^* - t_0}{M-3}$. The approximation function is defined in terms of the control points as follows:

$$f(t) = \sum_{k=0}^3 B_k(s) \phi_{j+k}, \quad (1.22)$$

where $j = \lfloor \frac{t-t_0}{\kappa} \rfloor - 2$ and $s = \frac{t-t_0}{\kappa} - \lfloor \frac{t-t_0}{\kappa} \rfloor$. The terms $\{B_k(s)\}_{k \in \{0,1,2,3\}}$ denote the spline basis functions defined as follows:

$$\begin{aligned}
 B_0(s) &= \frac{(1-s)^3}{6} \\
 B_1(s) &= \frac{(3s^3 - 6s^2 + 4)}{6} \\
 B_2(s) &= \frac{-3s^3 + 3s^2 + 3s + 1}{6} \\
 B_3(s) &= \frac{s^3}{6}.
 \end{aligned} \tag{1.23}$$

1.2.2 Pose Kinematics

For the development of autonomous navigation and manipulation for robots, the task generally described in terms of pose information as introduced above. The control objective is to regulate the robot motion to perform the tasks. In this section, the general six DOF robot is considered which has linear velocities $v = [v_x, v_y, v_z]^T$ and angular velocities $\omega = [\omega_x, \omega_y, \omega_z]^T$ in the frame \mathcal{F} with respect to frame \mathcal{F}' . Similarly, denote the linear and angular velocities of frame \mathcal{F}' with respect to \mathcal{F} as $v' = [v'_x, v'_y, v'_z]^T$ and angular velocities $\omega' = [\omega'_x, \omega'_y, \omega'_z]^T$. The positive directions of the linear velocities are the same as the directions of corresponding axes, and the positive directions of the angular velocities are determined by the right-hand rule.

Rotational motion

The direction of the angular velocity vector ω defines the instantaneous axis of rotation, that is, the axis about which the coordinate frame is rotating at a particular instant of time. In general, this axis changes with time. The magnitude of the vector is the rate of rotation about the axis; in this respect, it is similar to the angle-axis representation for rotation introduced above. From mechanics, there is a well-known expression for the derivative of a time-varying rotation matrix

$$\dot{R}(t) = [R\omega]_{\times} \cdot R(t). \tag{1.24}$$

Since R is redundant to describe the rotation, it is not appropriate for control development. As mentioned above, the angle-vector representation is widely used in robot applications, which is denoted as $\Theta(t) = u(t)\theta(t)$. The rotation matrix can be expressed in terms of the angle-vector representation as follows:

$$R = I_3 + \sin \theta [u]_{\times} + 2 \sin^2 \frac{\theta}{2} [u]_{\times}^2 \tag{1.25}$$

Taking the derivative of (1.25) and using (1.24), the following expression is obtained:

$$[R\omega] = \sin \theta [\dot{u}]_{\times} + [u]_{\times} \dot{\theta} + (1 - \cos \theta) [[u]_{\times} \dot{u}]_{\times}, \quad (1.26)$$

where the following properties are utilized:

$$[u]_{\times} \zeta = -[\zeta]_{\times} u \quad (1.27)$$

$$[u]_{\times}^2 = uu^T - I_3 \quad (1.28)$$

$$[u]_{\times} uu^T = 0 \quad (1.29)$$

$$[u]_{\times} [\dot{u}]_{\times} [u]_{\times} = 0 \quad (1.30)$$

$$[[u]_{\times} \dot{u}]_{\times} = [u]_{\times} [\dot{u}]_{\times} - [\dot{u}]_{\times} [u]_{\times}. \quad (1.31)$$

Taking the derivative of $\Theta(t)$, the following expression is obtained:

$$\dot{\Theta} = \dot{u}\theta + u\dot{\theta}. \quad (1.32)$$

Multiplying both sides in (1.32) by $(I_3 + [u]_{\times}^2)$, the following expression is obtained:

$$(I_3 + [u]_{\times}^2)\dot{\Theta} = u\dot{\theta}, \quad (1.33)$$

where the following properties are utilized;

$$[u]_{\times}^2 = uu^T - I_3$$

$$u^T u = 1 \quad (1.34)$$

$$u^T \dot{u} = 0.$$

Multiplying both sides in (1.32) by $-[u]_{\times}^2$, the following expression is obtained:

$$-[u]_{\times}^2 \dot{\Theta} = \dot{u}\theta, \quad (1.35)$$

where the following properties are utilized:

$$u^T u = 1 \quad (1.36)$$

$$u^T \dot{u} = 0.$$

From the expression in (1.26) and expressions in (1.32), (1.33), (1.35), the following expression can be obtained

$$\dot{\Theta} = L_w R\omega \quad (1.37)$$

where the Jacobian-like matrix $L_w \in \mathbb{R}^{3 \times 3}$ is defined as

$$L_w = I_3 - \frac{\theta}{2} [u]_{\times} + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})} \right) [u]_{\times}^2 \quad (1.38)$$

with $\text{sinc}(\theta) = \sin \theta / \theta$.

Translational motion

Denote the rotational and translational matrices from \mathcal{F}' to \mathcal{F} expressed in \mathcal{F}' as R' and x'_f , respectively. The translational vector x_f expressed in \mathcal{F} can be expressed as the following:

$$x_f = -R'^T x'_f. \quad (1.39)$$

Taking the time derivative of (1.39), it can be obtained that

$$\dot{x}_f = -\dot{R}'^T x'_f - R'^T \dot{x}'_f \quad (1.40)$$

Similar to (1.24), the above expression can be rewritten as the following:

$$\dot{x}_f = -[R'^T \omega']_{\times} \cdot R'^T x'_f - R'^T v'. \quad (1.41)$$

Using the fact that $\omega = R'^T \omega'$ and $v = R'^T v'$:

$$\dot{x}_f = -[\omega]_{\times} \cdot R'^T x'_f v. \quad (1.42)$$

Then, the time derivative of translational vector x_f can be expressed as the following:

$$\dot{x}_f = -[\omega]_{\times} \cdot x_f - v \quad (1.43)$$

1.3 Wheeled Mobile Robot Kinematics

In real applications, wheeled mobile robots are widely used for intelligent transportation and manipulation. They are actuated by wheels and generally move on the ground. As a result, the mobile robots have three DOFs, i.e., two position DOFs and one orientation DOF.

1.3.1 Wheel Kinematic Constraints

The first step to a kinematic model of the robot is to express constraints on the motions of individual wheels. The motions of individual wheels would be combined to compute the motion of the robot as a whole. As shown in Figure 1.5, there are four basic wheel types with varying kinematic properties in applications, i.e., fixed wheel, steered wheel, castor wheel, and Swedish wheel.

In the following developments, it is assumed that the plane of the wheel always remains vertical and that there is one single point of contact between the wheel and the ground plane. Besides, there is no sliding at this single point of contact. That is, the wheel undergoes motion only under conditions of pure rolling and rotation about the vertical axis through the contact point. Under these assumptions, we present two constraints for every wheel type. The rolling constraint enforces the concept of rolling contact that the wheel must roll when

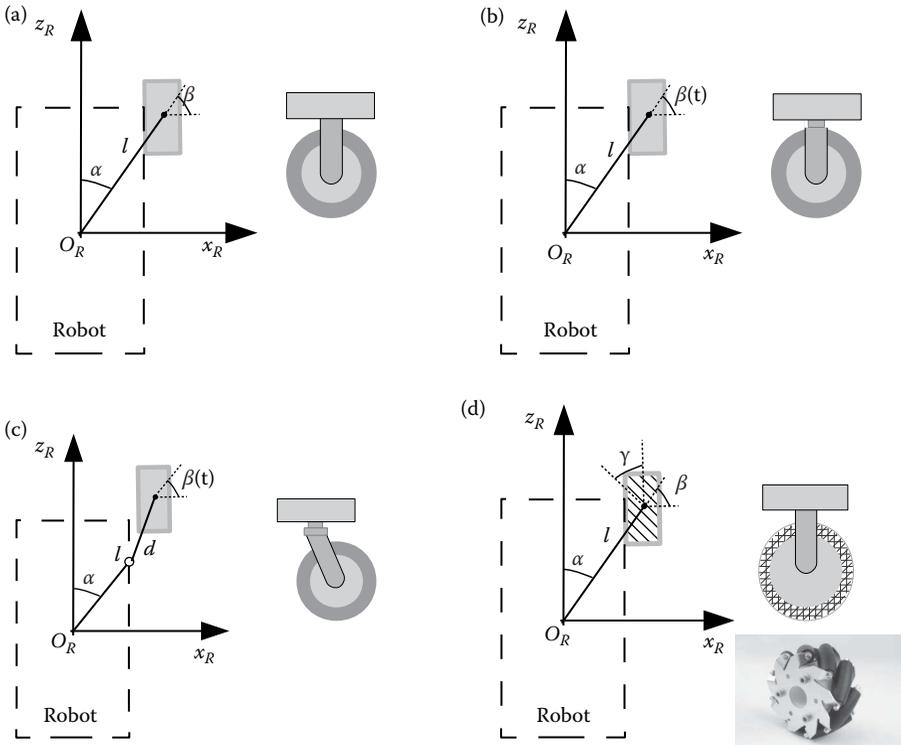


Figure 1.5: Standard basic wheels. (a) Fixed wheel, (b) Steered wheel, (c) Castor wheel, and (d) Swedish wheel.

motion takes place in the appropriate direction. The sliding constraint enforces the concept of no lateral slippage that the wheel must not slide orthogonal to the wheel plane.

As shown in Figure 1.5, the position of the wheel is expressed in polar coordinates by distance l and angle α . The angle of the wheel plane relative to the chassis is denoted by β , which is fixed since the fixed standard wheel is not steerable. The wheel, which has radius r , can spin over time with angular velocity $\omega(t)$. The fixed wheel has no vertical axis of rotation for steering. Its angle to the chassis is thus fixed, and it is limited to motion back and forth along the wheel plane and rotation around its contact point with the ground plane. The steered wheel differs from the fixed standard wheel only in that there is an additional DOF; the wheel may rotate around a vertical axis passing through the center of the wheel and the ground contact point. Castor wheels are able to steer around a vertical axis. However, unlike the steered standard wheel, the vertical axis of rotation in a castor wheel does not pass through the ground contact point. Swedish

wheels have no vertical axis of rotation, yet are able to move omnidirectionally like the castor wheel. This is possible by adding a DOF to the fixed standard wheel. Swedish wheels consist of a fixed standard wheel with rollers attached to the wheel perimeter with axes that are antiparallel to the main axis of the fixed wheel component. The rolling and sliding constraints are shown in Tables 1.1 and 1.2, respectively. It should be noted that β is free for the castor wheel and ω_{sw} is free for the Swedish wheel; thus, the sliding constraints of the castor and the Swedish wheels do not affect the motion direction of the robots.

1.3.2 Mobile Robot Kinematic Modeling

For a mobile robot with several wheels, its kinematic constraints can be computed by combining all of the kinematic constraints of each wheel. From the above, the castor and Swedish wheels have no impact on the kinematic constraints of the robot. Therefore, only fixed wheels and steerable standard wheels have impact on robot kinematics, and therefore require consideration when computing the robot's kinematic constraints.

Suppose that the robot has a total of N standard wheels, comprising N_f fixed wheels and N_s steerable wheels. Denote B_f and $B_s(t)$ as the fixed steering angles of fixed wheels and the varying steering angles of the steerable wheels, respectively. Denote W_f and $W_s(t)$ as the angular velocities of fixed and steerable wheels, respectively. The rolling constraint of the robot can be obtained by combining the effects of wheels as follows:

$$J_1(B_s)R(\theta)\dot{X} = J_2W, \quad (1.44)$$

Table 1.1 Wheel Kinematics: Rolling Constraints

<i>Wheel Type</i>	<i>Rolling Constraint</i>
Fixed wheel	
Steered wheel	$[\sin(\alpha + \beta) \quad -\cos(\alpha + \beta) \quad -l \cos \beta] R(\theta) \dot{X} = r\omega$
Castor wheel	
Swedish wheel	$[\sin(\alpha + \beta + \gamma) \quad -\cos(\alpha + \beta + \gamma)]$ $[-l \cos(\beta + \gamma)] R(\theta) \dot{X} = r \cos \gamma \cdot \omega$

Table 1.2 Wheel Kinematics: Sliding Constraints

<i>Wheel Type</i>	<i>Sliding Constraint</i>
Fixed wheel	$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad l \sin \beta] R(\theta) \dot{X} = 0$
Steered wheel	
Castor wheel	$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad d + l \sin \beta] R(\theta) \dot{X} = -d\dot{\beta}$
Swedish wheel	$[\cos(\alpha + \beta + \gamma) \quad \sin(\alpha + \beta + \gamma) \quad l \sin(\beta + \gamma)]$ $R(\theta) \dot{X} = r\omega \sin \gamma + r_{sw} \omega_{sw}$

where $W = [W_f, W_s]^T$ is the angular velocities of all wheels. J_2 is a constant diagonal $N \times N$ matrix whose entries are the corresponding radius of the wheels. J_1 defines the projections for the wheels to their motions along their individual wheel planes:

$$J_1(B_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(B_s) \end{bmatrix}. \tag{1.45}$$

Here, $J_{1f} \in \mathbb{R}^{N_f \times 3}$ and $J_{1s} \in \mathbb{R}^{N_s \times 3}$ are Jacobian-like matrices with each row consisting of the three terms in the equation in Table 1.1.

The sliding constraints can be also obtained as follows:

$$C_1(B_s)R(\theta)\dot{X} = 0, \tag{1.46}$$

where $C_1(B_s) = [C_{1f} \ C_{1s}(B_s)]^T$. The matrices $C_{1f} \in \mathbb{R}^{N_f \times 3}$ and $C_{1s}(B_s) \in \mathbb{R}^{N_s \times 3}$ are composed of rows consisting of the three terms in the equation in Table 1.2.

1.3.3 Typical Nonholonomic Mobile Robot

The above subsection provides a general tool for kinematic modeling of mobile robots. In the following, the typical differential driving mobile robot is considered as shown in Figure 1.6.

As shown in Figure 1.6, the differential driving mobile robot consists of two independent driving fixed wheels and one castor wheel. The origin of the robot coordinate is located at P . Then, the kinematic model of the robot can be obtained by combining the rolling and sliding constraints as follows:

$$\begin{bmatrix} J_1(B_s) \\ C_1(B_s) \end{bmatrix} R(\theta)\dot{X} = \begin{bmatrix} J_2W \\ 0 \end{bmatrix}. \tag{1.47}$$

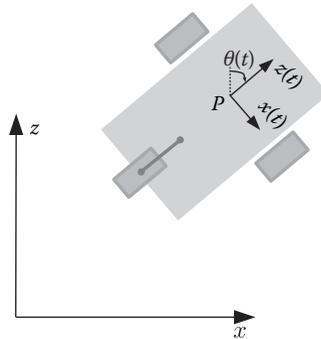


Figure 1.6: Differential driving mobile robot.