Contents

Preface vii

Printed vs PDF Versions of the Book ix

Changes Since Groovy xi

1. Purpose of this Book 1

2. Real and Simulated Robots 3

2.1 Gazebo, Stage, and the ArbotiX Simulator 3
2.2 Introducing the TurtleBot, Maxwell and Pi Robot 4

3. Operating Systems and ROS Versions 5

3.1 Installing Ubuntu Linux 5
3.2 Getting Started with Linux 6
3.3 A Note about Updates and Upgrades 7

4. Reviewing the ROS Basics 9

4.1 Installing ROS 9
4.2 Installing rosinstall 9
4.3 Building ROS Packages with Catkin 10
4.4 Creating a catkin Workspace 10
4.5 Doing a "make clean" with catkin 11
4.6 Rebuilding a Single catkin Package 11
4.7 Mixing catkin and rosbuild Workspaces 12
4.8 Working through the Official ROS Tutorials 13
4.9 RViz: The ROS Visualization Tool 13
4.10 Using ROS Parameters in your Programs 14
4.11 Using rqt_reconfigure (formerly dynamic_reconfigure) to set ROS Parameters 14
4.12 Networking Between a Robot and a Desktop Computer 16
4.12.1 Time Synchronization 16
4.12.2 ROS Networking using Zeroconf 16
4.12.3 Testing Connectivity 17
4.12.4 Setting the ROS_MASTER_URI and ROS_HOS TNAME Variables 17
4.12.5 Opening New Terminals 18
4.12.6 Running Nodes on both Machines 19
4.12.7 ROS Networking across the Internet 20
4.13 ROS Recap 21
4.14 What is a ROS Application? 21
4.15 Installing Packages with SVN, Git, and Mercurial 22
4.15.1 SVN 23

4.15.2 Git 23
4.15.3 Mercurial 24
4.16 Removing Packages from your Personal catkin Directory 24
4.17 How to Find Third-Party ROS Packages 25
4.17.1 Searching the ROS Wiki 25
4.17.2 Using the roslocate Command 25
4.17.3 Browsing the ROS Software Index 27
4.17.4 Doing a Google Search 27
4.18 Getting Further Help with ROS 27

5. Installing the ros-by-example Code 29

5.1 Installing the Prerequisites 29
5.2 Cloning the Hydro ros-by-example Repository 29
5.2.1 Upgrading from Electric or Fuerte 29
5.2.2 Upgrading from Groovy 30
5.2.3 Cloning the rbx1 repository for Hydro 30
5.3 About the Code Listings in this Book 31

6. Installing the Arbotix Simulator 33

6.1 Installing the Simulator 33
6.2 Testing the Simulator 33
6.3 Running the Simulator with Your Own Robot 35

7. Controlling a Mobile Base 37

7.1 Units and Coordinate Systems 37
7.2 Levels of Motion Control 37
7.2.1 Motors, Wheels, and Encoders 38
7.2.2 Motor Controllers and Drivers 38
7.2.3 The ROS Base Controller 38
7.2.4 Frame-Base Motion using the move_base ROS Package 39
7.2.5 SLAM using the gmapping and amcl ROS Packages 39
7.2.6 Semantic Goals 40
7.2.7 Summary 40
7.3 Twisting and Turning with ROS 41
7.3.1 Example Twist Messages 41
7.3.2 Monitoring Robot Motion using RViz 42
7.4 Calibrating Your Robot's Odometry 44
7.4.1 Linear Calibration 45
7.4.2 Angular Calibration 46
7.5 Sending Twist Messages to a Real Robot 47
7.6 Publishing Twist Messages from a ROS Node 49
7.6.1 Estimating Distance and Rotation Using Time and Speed 49
7.6.2 Timed Out-and-Back in the ArbotiX Simulator 49
7.6.3 The Timed Out-and-Back Script 50
7.6.4 Timed Out and Back using a Real Robot 55
7.7 Are We There Yet? Going the Distance with Odometry 57
7.8 Out and Back Using Odometry 59
7.8.1 Odometry-Based Out and Back in the ArbotiX Simulator 59

7.8.2 Odometry-Based Out and Back Using a Real Robot 60
7.8.3 The Odometry-Based Out-and-Back Script 62
7.8.4 The /odom Topic versus the /odom Frame 67
7.9 Navigating a Square using Odometry 68
7.9.1 Navigating a Square in the ArbotiX Simulator 68
7.9.2 Navigating a Square using a Real Robot 69
7.9.3 The nav_square.py Script 71
7.9.4 The Trouble with Dead Reckoning 71
7.10 Teleoperating your Robot 71
7.10.1 Using the Keyboard 72
7.10.2 Using a Logitech Game Pad 73
7.10.3 Using the ArbotiX Controller GUI 73
7.10.4 TurtleBot Teleoperation Using Interactive Markers 74
7.10.5 Writing your Own Teleop Node 74

8. Navigation, Path Planning and SLAM 75

8.1 Path Planning and Obstacle Avoidance using move_base 75
8.1.1 Specifying Navigation Goals Using move_base 76
8.1.2 Configuration Parameters for Path Planning 77
8.1.2.1 base_local_planner_params.yaml 78
8.1.2.2 costmap_common_params.yaml 79
8.1.2.3 global_costmap_params.yaml 80
8.1.2.4 local_costmap_params.yaml 80
8.2 Testing move_base in the ArbotiX Simulator 81
8.2.1 Point and Click Navigation in RViz 86
8.2.2 Navigation Display Types for RViz 87
8.2.3 Navigating a Square using move_base 87
8.2.4 Avoiding Simulated Obstacles 94
8.2.5 Setting Manual Goals with Obstacles Present 97
8.3 Running move_base on a Real Robot 97
8.3.1 Testing move_base without Obstacles 97
8.3.2 Avoiding Obstacles using a Depth Camera as a Fake Laser 99
8.4 Map Building using the gmapping Package 101
8.4.1 Laser Scanner or Depth Camera? 102
8.4.2 Collecting and Recording Scan Data 104
8.4.3 Creating the Map 106
8.4.4 Creating a Map from Bag Data 106
8.4.5 Can I Extend or Modify an Existing Map? 108
8.5 Navigation and Localization using a Map and amcl 108
8.5.1 Testing amcl with Fake Localization 108
8.5.2 Using amcl with a Real Robot 110
8.5.3 Fully Autonomous Navigation 113
8.5.4 Running the Navigation Test in Simulation 113
8.5.5 Understanding the Navigation Test Script 115
8.5.6 Running the Navigation Test on a Real Robot 120
8.5.7 What's Next? 122

9. Speech Recognition and Synthesis 123

9.1 Installing PocketSphinx for Speech Recognition 123

9.2 Testing the PocketSphinx Recognizer 123
9.3 Creating a Vocabulary 125
9.4 A Voice-Control Navigation Script 127
9.4.1 Testing Voice-Control in the ArbotiX Simulator 132
9.4.2 Using Voice-Control with a Real Robot 133
9.5 Installing and Testing Festival Text-to-Speech 134
9.5.1 Using Text-to-Speech within a ROS Node 136
9.5.2 Testing the talkback.py script 138

10. Robot Vision 139

10.1 OpenCV, OpenNI and PCL 139
10.2 A Note about Camera Resolutions 140
10.3 Installing and Testing the ROS Camera Drivers 140
10.3.1 Installing the OpenNI Drivers 140
10.3.2 Installing Webcam Drivers 140
10.3.3 Testing your Kinect or Xtion Camera 141
10.3.4 Testing your USB Webcam 142
10.4 Installing OpenCV on Ubuntu Linux 143
10.5 ROS to OpenCV: The cv_bridge Package 144
10.6 The ros2opencv2.py Utility 149
10.7 Processing Recorded Video 151
10.8 OpenCV: The Open Source Computer Vision Library 152
10.8.1 Face Detection 152
10.8.2 Keypoint Detection using GoodFeaturesToTrack 158
10.8.3 Tracking Keypoints using Optical Flow 164
10.8.4 Building a Better Face Tracker 170
10.8.5 Dynamically Adding and Dropping Keypoints 173
10.8.6 Color Blob Tracking (CamShift) 175
10.9 OpenNI and Skeleton Tracking 181
10.9.1 Checking your OpenNI installation for Hydro 182
10.9.2 Viewing Skeletons in RViz 183
10.9.3 Accessing Skeleton Frames in your Programs 183
10.10 PCL Nodelets and 3D Point Clouds 191
10.10.1 The PassThrough Filter 191
10.10.2 Combining More than One PassThrough Filter 193
10.10.3 The VoxelGrid Filter 194

11. Combining Vision and Base Control 197

11.1 A Note about Camera Coordinate Axes 197
11.2 Object Tracker 197
11.2.1 Testing the Object Tracker with rqt_plot 197
11.2.2 Testing the Object Tracker with a Simulated Robot 198
11.2.3 Understanding the Object Tracker Code 199
11.2.4 Object Tracking on a Real Robot 205
11.3 Object Follower 206
11.3.1 Adding Depth to the Object Tracker 206
11.3.2 Testing the Object Follower with a Simulated Robot 210
11.3.3 Object Following on a Real Robot 211

11.4 Person Follower 212
11.4.1 Testing the Follower Application in Simulation 213
11.4.2 Understanding the Follower Script 213
11.4.3 Running the Follower Application on a TurtleBot 217
11.4.4 Running the Follower Node on a Filtered Point Cloud 218

12. Dynamixel Servos and ROS 219

12.1 A TurtleBot with a Pan-and-Tilt Head 220
12.2 Choosing a Dynamixel Hardware Controller 220
12.3 A Note Regarding Dynamixel Hardware 221
12.4 Choosing a ROS Dynamixel Package 221
12.5 Understanding the ROS JointState Message Type 221
12.6 Controlling Joint Position, Speed and Torque 222
12.6.1 Setting Servo Position 223
12.6.2 Setting Servo Speed 224
12.6.3 Controlling Servo Torque 224
12.7 Checking the USB2Dynamixel Connection 225
12.8 Setting the Servo Hardware IDs 226
12.9 Configuring and Launching dynamixel_controllers 227
12.9.1 The dynamixel_controllers Configuration File 227
12.9.2 The dynamixel_controllers Launch File 228
12.10 Testing the Servos 230
12.10.1 Starting the Controllers 230
12.10.2 Monitoring the Robot in RViz 231
12.10.3 Listing the Controller Topics and Monitoring Joint States 231
12.10.4 Listing the Controller Services 233
12.10.5 Setting Servo Position, Speed and Torque 234
12.10.6 Using the relax_all_servos.py Script 235
12.11 Tracking a Visual Target 236
12.11.1 Tracking a Face 236
12.11.2 The Head Tracker Script 237
12.11.3 Tracking Colored Objects 244
12.11.4 Tracking Manually Selected Targets 245
12.12 A Complete Head Tracking ROS Application 246

13. Where to Go Next? 249