

Main Chapter Headings

Preface.....	v
Printed vs PDF Versions of the Book.....	vii
Changes Since ROS Hydro.....	xvii
1. Scope of this Volume.....	1
2. Installing the ros-by-example Code.....	3
3. Task Execution using ROS.....	7
4. Creating a URDF Model for your Robot.....	89
5. Controlling Dynamixel Servos: Take 2.....	147
6. Robot Diagnostics.....	167
7. Dynamic Reconfigure.....	189
8. Multiplexing Topics with mux & yocs.....	201
9. Head Tracking in 3D.....	213
10. Detecting and Tracking AR Tags.....	231
11. Arm Navigation using MoveIt!.....	251
12. Gazebo: Simulating Worlds and Robots.....	369
13. Rosbridge: Building a Web GUI for your Robot.....	403
Appendix: Plug and Play USB Devices for ROS using udev Rules.....	437
Notes.....	443
Links Referenced in the Text.....	445

Contents

Preface.....	v
Printed vs PDF Versions of the Book.....	vii
Changes Since ROS Hydro.....	xvii
1. Scope of this Volume.....	1
2. Installing the ros-by-example Code.....	3
3. Task Execution using ROS.....	7
3.1 A Fake Battery Simulator.....	8
3.2 A Common Setup for Running the Examples.....	10
3.3 A Brief Review of ROS Actions.....	11
3.4 A Patrol Bot Example.....	12
3.5 The Patrol Bot using a Standard Script.....	13
3.6 Problems with the Script Approach.....	16
3.7 SMACH or Behavior Trees?.....	16
3.8 SMACH: Tasks as State Machines.....	17
3.8.1 SMACH review.....	18
3.8.2 Patrolling a square using SMACH.....	19
3.8.3 Testing SMACH navigation in the ArbotiX simulator.....	23
3.8.4 Accessing results from a SimpleActionState.....	26
3.8.5 SMACH Iterators.....	27
3.8.6 Executing commands on each transition.....	30
3.8.7 Interacting with ROS topics and services.....	31
3.8.8 Callbacks and Introspection.....	36
3.8.9 Concurrent tasks: Adding the battery check to the patrol routine.....	36
3.8.10 Comments on the battery checking Patrol Bot.....	44
3.8.11 Passing user data between states and state machines.....	44
3.8.12 Subtasks and hierarchical state machines.....	48
3.8.13 Adding the battery check to the house cleaning robot.....	54
3.8.14 Drawbacks of state machines.....	54
3.9 Behavior Trees.....	55
3.9.1 Behavior Trees versus Hierarchical State Machines.....	56
3.9.2 Key properties of behavior trees.....	57
3.9.3 Building a behavior tree.....	58
3.9.4 Selectors and sequences.....	60
3.9.5 Customizing behaviors using decorators (meta-behaviors).....	61
3.10 Programming with Behavior Trees and ROS.....	62
3.10.1 Installing the pi_trees library.....	62
3.10.2 Basic components of the pi_trees library.....	63
3.10.3 ROS-specific behavior tree classes.....	69
3.10.4 A Patrol Bot example using behavior trees.....	73
3.10.5 A housing cleaning robot using behavior trees.....	79
3.10.6 Parallel tasks.....	85
3.10.7 Adding and removing tasks.....	87

4. Creating a URDF Model for your Robot.....	89
4.1 Start with the Base and Wheels.....	90
4.1.1 The robot_state_publisher and joint_state_publisher nodes.....	91
4.1.2 The base URDF/Xacro file.....	92
4.1.3 Alternatives to using the /base_footprint frame.....	97
4.1.4 Adding the base to the robot model.....	97
4.1.5 Viewing the robot's transform tree.....	98
4.1.6 Using a mesh for the base.....	99
4.2 Simplifying Your Meshes.....	104
4.3 Adding a Torso.....	104
4.3.1 Modeling the torso.....	105
4.3.2 Attaching the torso to the base.....	106
4.3.3 Using a mesh for the torso.....	107
4.3.4 Adding the mesh torso to the mesh base.....	108
4.4 Measure, Calculate and Tweak.....	110
4.5 Adding a Camera.....	110
4.5.1 Placement of the camera.....	111
4.5.2 Modeling the camera.....	112
4.5.3 Adding the camera to the torso and base.....	114
4.5.4 Viewing the transform tree with torso and camera.....	115
4.5.5 Using a mesh for the camera.....	117
4.5.6 Using an Asus Xtion Pro instead of a Kinect.....	118
4.6 Adding a Laser Scanner (or other Sensors).....	119
4.6.1 Modeling the laser scanner.....	119
4.6.2 Attaching a laser scanner (or other sensor) to a mesh base.....	120
4.6.3 Configuring the laser node launch file.....	121
4.7 Adding a Pan and Tilt Head.....	122
4.7.1 Using an Asus Xtion Pro instead of a Kinect.....	124
4.7.2 Modeling the pan-and-tilt head.....	124
4.7.3 Figuring out rotation axes.....	127
4.7.4 A pan and tilt head using meshes on Pi Robot.....	128
4.7.5 Using an Asus Xtion Pro mesh instead of a Kinect on Pi Robot.....	129
4.8 Adding One or Two Arms.....	129
4.8.1 Placement of the arm(s).....	129
4.8.2 Modeling the arm.....	129
4.8.3 Adding a gripper frame for planning.....	132
4.8.4 Adding a second arm.....	133
4.8.5 Using meshes for the arm servos and brackets.....	135
4.9 Adding a Telescoping Torso to the Box Robot.....	137
4.10 Adding a Telescoping Torso to Pi Robot.....	138
4.11 A Tabletop One-Arm Pi Robot.....	138
4.12 Testing your Model with the ArbotiX Simulator.....	140
4.12.1 A fake Box Robot.....	141
4.12.2 A fake Pi Robot.....	143
4.13 Creating your own Robot Description Package.....	144
4.13.1 Copying files from the rbx2_description package.....	145
4.13.2 Creating a test launch file.....	145
5. Controlling Dynamixel Servos: Take 2.....	147

5.1	Installing the ArbotiX Packages.....	147
5.2	Launching the ArbotiX Nodes.....	148
5.3	The ArbotiX Configuration File.....	152
5.4	Testing the ArbotiX Joint Controllers in Fake Mode.....	158
5.5	Testing the Arbotix Joint Controllers with Real Servos.....	160
5.6	Relaxing All Servos.....	163
5.7	Enabling or Disabling All Servos.....	166
6.	Robot Diagnostics.....	167
6.1	The DiagnosticStatus Message.....	168
6.2	The Analyzer Configuration File.....	169
6.3	Monitoring Dynamixel Servo Temperatures.....	170
6.3.1	Monitoring the servos for a pan-and-tilt head.....	170
6.3.2	Viewing messages on the /diagnostics topic.....	173
6.3.3	Protecting servos by monitoring the /diagnostics topic.....	175
6.4	Monitoring a Laptop Battery.....	179
6.5	Creating your Own Diagnostics Messages.....	180
6.6	Monitoring Other Hardware States.....	186
7.	Dynamic Reconfigure.....	189
7.1	Adding Dynamic Parameters to your own Nodes.....	190
7.1.1	Creating the .cfg file.....	190
7.1.2	Making the .cfg file executable.....	191
7.1.3	Configuring the CMakeLists.txt file.....	192
7.1.4	Building the package.....	192
7.2	Adding Dynamic Reconfigure Capability to the Battery Simulator Node.....	192
7.3	Adding Dynamic Reconfigure Client Support to a ROS Node.....	196
7.4	Dynamic Reconfigure from the Command Line.....	199
8.	Multiplexing Topics with mux & yocs.....	201
8.1	Configuring Launch Files to use mux Topics.....	202
8.2	Testing mux with the Fake TurtleBot.....	203
8.3	Switching Inputs using mux Services.....	204
8.4	A ROS Node to Prioritize mux Inputs.....	205
8.5	The YOCS Controller from Yujin Robot.....	208
8.5.1	Adding input sources.....	211
9.	Head Tracking in 3D.....	213
9.1	Tracking a Fictional 3D Target.....	214
9.2	Tracking a Point on the Robot.....	215
9.3	The 3D Head Tracking Node.....	218
9.3.1	Real or fake head tracking.....	218
9.3.2	Projecting the target onto the camera plane.....	219
9.4	Head Tracking with Real Servos.....	222
9.4.1	Real servos and fake target.....	223
9.4.2	Real servos, real target.....	224
9.4.3	The nearest_cloud.py node and launch file.....	226

10. Detecting and Tracking AR Tags.....	231
10.1 Installing and Testing the ar_track_alvar Package.....	232
10.1.1 Creating your own AR Tags.....	232
10.1.2 Generating and printing the AR tags.....	234
10.1.3 Launching the camera driver and ar_track_alvar node.....	234
10.1.4 Testing marker detection.....	236
10.1.5 Understanding the /ar_pose_marker topic.....	236
10.1.6 Viewing the markers in RViz.....	237
10.2 Accessing AR Tag Poses in your Programs.....	238
10.2.1 The ar_tags_cog.py script.....	239
10.2.2 Tracking the tags with a pan-and-tilt head.....	242
10.3 Tracking Multiple Tags using Marker Bundles.....	243
10.4 Following an AR Tag with a Mobile Robot.....	244
10.4.1 Running the AR follower script on a TurtleBot	247
10.5 Exercise: Localization using AR Tags.....	248
11. Arm Navigation using MoveIt!.....	251
11.1 Do I Need a Real Robot with a Real Arm?.....	252
11.2 Degrees of Freedom.....	252
11.3 Joint Types.....	253
11.4 Joint Trajectories and the Joint Trajectory Action Controller.....	254
11.5 Forward and Inverse Arm Kinematics.....	257
11.6 Numerical versus Analytic Inverse Kinematics.....	258
11.7 The MoveIt! Architecture.....	258
11.8 Installing MoveIt!.....	260
11.9 Creating a Static URDF Model for your Robot	260
11.10 Running the MoveIt! Setup Assistant.....	261
11.10.1 Load the robot's URDF model.....	262
11.10.2 Generate the collision matrix.....	263
11.10.3 Add the base_odom virtual joint.....	264
11.10.4 Adding the right arm planning group.....	265
11.10.5 Adding the right gripper planning group.....	268
11.10.6 Defining robot poses.....	270
11.10.7 Defining end effectors.....	272
11.10.8 Defining passive joints.....	272
11.10.9 Generating the configuration files.....	272
11.11 Configuration Files Created by the MoveIt! Setup Assistant.....	274
11.11.1 The SRDF file (robot_name.srdf).....	274
11.11.2 The fake_controllers.yaml file.....	275
11.11.3 The joint_limits.yaml file.....	276
11.11.4 The kinematics.yaml file.....	277
11.12 The move_group Node and Launch File.....	279
11.13 Testing MoveIt! in Demo Mode.....	279
11.13.1 Exploring additional features of the Motion Planning plugin.....	283
11.13.2 Re-running the Setup Assistant at a later time.....	284
11.14 Testing MoveIt! from the Command Line.....	285
11.15 Determining Joint Configurations and End Effector Poses.....	288
11.16 Using the ArbotiX Joint Trajectory Action Controllers.....	290

11.16.1	Testing the ArbotiX joint trajectory action controllers in simulation.....	291
11.16.2	Testing the ArbotiX joint trajectory controllers with real servos.....	298
11.17	Configuring MoveIt! Joint Controllers.....	299
11.17.1	Creating the controllers.yaml file.....	300
11.17.2	Creating the controller manager launch file.....	302
11.18	The MoveIt! API.....	303
11.19	Forward Kinematics: Planning in Joint Space.....	304
11.20	Inverse Kinematics: Planning in Cartesian Space.....	312
11.21	Pointing at or Reaching for a Visual Target.....	320
11.22	Setting Constraints on Planned Trajectories.....	322
11.22.1	Executing Cartesian Paths.....	322
11.22.2	Setting other path constraints.....	328
11.23	Adjusting Trajectory Speed.....	332
11.24	Adding Obstacles to the Planning Scene.....	336
11.25	Attaching Objects and Tools to the Robot.....	345
11.26	Pick and Place.....	347
11.27	Adding a Sensor Controller.....	359
11.28	Running MoveIt! on a Real Arm.....	363
11.28.1	Creating your own launch files and scripts.....	364
11.28.2	Running the robot's launch files.....	364
11.28.3	Forward kinematics on a real arm.....	365
11.28.4	Inverse kinematics on a real arm.....	365
11.28.5	Cartesian paths on a real arm.....	366
11.28.6	Pick-and-place on a real arm.....	366
11.28.7	Pointing at or reaching for a visual target.....	367

12. Gazebo: Simulating Worlds and Robots.....369

12.1	Installing Gazebo.....	370
12.2	Hardware Graphics Acceleration.....	371
12.3	Installing the ROS Gazebo Packages.....	372
12.4	Installing the Kobuki ROS Packages.....	373
12.5	Installing the Fetch Robot ROS Packages.....	373
12.6	Using the Gazebo GUI.....	373
12.7	Testing the Kobuki Robot in Gazebo.....	375
12.7.1	Accessing simulated sensor data.....	377
12.7.2	Adding safety control to the Kobuki.....	381
12.7.3	Running the nav_square.py script from Volume 1.....	383
12.8	Loading Other Worlds and Objects.....	384
12.9	Testing the Fetch Robot in Gazebo.....	386
12.9.1	Fetch joint trajectories.....	386
12.9.2	Fetch and MoveIt!.....	387
12.9.3	Fetch Pick-and-Place.....	389
12.10	Real Pick-and-Place using the simple_grasping Perception Pipeline.....	390
12.10.1	Limitations of depth cameras.....	391
12.10.2	Running the demo.....	391
12.10.3	Understanding the real_pick_and_place.py script.....	397
12.11	Running Gazebo Headless + RViz.....	400

13. Rosbridge: Building a Web GUI for your Robot.....	403
13.1 Installing the rosbridge Packages.....	403
13.2 Installing the web_video_server Package.....	404
13.3 Installing a Simple Web Server (mini-httpd).....	406
13.4 Starting mini-httpd, rosbridge and web_video_server.....	407
13.5 A Simple rosbridge HTML/Javascript GUI.....	409
13.6 Testing the GUI with a Fake TurtleBot.....	411
13.7 Testing the GUI with a Real Robot.....	412
13.8 Viewing the Web GUI on another Device on your Network.....	412
13.9 Using the Browser Debug Console.....	413
13.10 Understanding the Simple GUI.....	414
13.10.1 The HTML layout: simple_gui.html.....	415
13.10.2 The JavaScript code: simple_gui.js.....	420
13.11 A More Advanced GUI using jQuery, jqWidgets and KineticJS.....	430
13.12 Rosbridge Summary.....	435
Appendix: Plug and Play USB Devices for ROS using udev Rules.....	437
13.13 Adding yourself to the dialout Group.....	437
13.14 Determining the Serial Number of a Device.....	438
13.15 UDEV Rules.....	439
13.16 Testing a UDEV Rule.....	440
13.17 Using a UDEV Device Name in a ROS Configuration File.....	440
Notes.....	443
Links Referenced in the Text.....	445