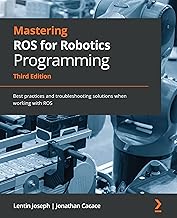
Mastering ROS for Robotics Programming – 3rd Edition

*Pack Publishing*



# 1. Section 1 – ROS Programming Essentials

# Chapter 1: Introduction to ROS

## 1.1 Technical requirements

## 1.2 Why should we use ROS?

## 1.3 Understanding the ROS filesystem level

ROS packages

ROS metapackages

ROS messages

The ROS services

## 1.4 Understanding the ROS computation graph level

ROS nodes

ROS messages

ROS topics

ROS services

ROS bagfiles

The ROS master

Using the ROS parameter

## 1.5 ROS community level

## 1.6 Prerequisites for starting with ROS

ROS distributions

Running the ROS master and the ROS parameter server

## 1.7 Summary

## 1.8 Questions

# Chapter 2: Getting Started with ROS Programming

## 2.1 Technical requirements

## 2.2 Creating a ROS package

Working with ROS topics

Creating ROS nodes

Building the nodes

## 2.3 Adding custom .msg and .srv files

## 2.4 Working with ROS services

Working with ROS actionlib

Building the ROS action server and client

## 2.5 Creating launch files

## 2.6 Applications of topics services and actionlib

## 2.7 Summary

## 2.8 Questions

# 2. Section 2 – ROS Robot Simulation

# Chapter 3: Working with ROS for 3D Modeling

## 3.1 Technical requirements

## 3.2 ROS packages for robot modeling

## 3.3 Understanding robot modeling using URDF

## 3.4 Creating the ROS package for the robot description

## 3.5 Creating our first URDF model

## 3.6 Explaining the URDF file

## 3.7 Visualizing the 3D robot model in RViz

Interacting with pan-and-tilt joints

## 3.8 Adding physical and collision properties to a URDF model

## 3.9 Understanding robot modeling using xacro 77

Using properties

Using the math expression

## 3.10 Converting xacro to URDF

## 3.11 Creating the robot description for a seven-DOF robot manipulator

Arm specification

## 3.12 Explaining the xacro model of the seven-DOF arm

Using constants

Using macro

Including other xacro files

Using meshes in the link

Working with the robot gripper

Viewing the seven-DOF arm in RViz

## 3.13 Creating a robot model for the differential drive mobile robot

## 3.14 Summary

## 3.15 Questions

# Chapter 4: Simulating Robots Using ROS and Gazebo

## 4.1 Technical requirements

## 4.2 Simulating the robotic arm using Gazebo and ROS

## 4.3 Creating the robotic arm simulation model for Gazebo

Adding colors and textures to the Gazebo robot model

Adding transmission tags to actuate the model

## 4.4 Adding the gazebo\_ros\_control plugin

Adding a 3D vision sensor to Gazebo

## 4.5 Simulating the robotic arm with Xtion Pro

Visualizing the 3D sensor data

## 4.6 Moving the robot joints using ROS controllers in Gazebo

Understanding the ros\_control packages

Different types of ROS controllers and hardware interfaces

How the ROS controller interacts with Gazebo

Interfacing the joint state controllers and joint position controllers with the arm

Launching the ROS controllers with Gazebo

Moving the robot joints

## 4.7 Simulating a differential wheeled robot in Gazebo

Adding the laser scanner to Gazebo

Moving the mobile robot in Gazebo

Adding joint state publishers to the launch file

## 4.8 Adding the ROS teleop node

## 4.9 Summary

## 4.10 Questions

# Chapter 5: Simulating Robots Using ROS, CoppeliaSim, and Webots

## 5.1 Technical requirements

## 5.2 Setting up CoppeliaSim with ROS

Understanding the RosInterface plugin

Working with ROS messages

## 5.3 Simulating a robotic arm using CoppeliaSim and ROS

Adding the ROS interface to CoppeliaSim joint controllers

## 5.4 Setting up Webots with ROS

Introduction to the Webots simulator

Simulating a mobile robot with Webots

## 5.5 Writing your first controller

Simulating the robotic arm using

Webots and ROS 148

## 5.6 Writing a teleop node using webots\_ros

Starting Webots with a launch file

## 5.7 Summary

## 5.8 Questions

# Chapter 6: Using the ROS MoveIt! and Navigation Stack

## 6.1 Technical requirements

## 6.2 The MoveIt! architecture

The move\_group node

Motion planning using MoveIt!

Motion-planning request adapters

MoveIt! planning scene

MoveIt! kinematics handling

MoveIt! collision checking

## 6.3 Generating a MoveIt! configuration package using the Setup Assistant tool

Step 1 – Launching the Setup Assistant tool

Step 2 – Generating a self-collisionmatrix

Step 3 – Adding virtual joints

Step 4 – Adding planning groups

Step 5 – Adding the robot poses

Step 6 – Setting up the robot end effector

Step 7 – Adding passive joints

Step 8 – Author information

Step 9 – Generating configuration files

## 6.4 Motion planning of a robot in RViz using the MoveIt! configuration package

Using the RViz MotionPlanning plugin

Interfacing the MoveIt! configuration package to Gazebo

## 6.5 Understanding the ROS Navigation stack

ROS Navigation hardware requirements

Working with Navigation packages

Workings of the Navigation stack

## 6.6 Building a map using SLAM

Creating a launch file for gmapping

Running SLAM on the differential drive robot

Implementing autonomous navigation using amcl and a static map

Creating an amcl launch file

## 6.7 Summary

## 6.8 Questions

# Chapter 7: Exploring the Advanced Capabilities of ROS MoveIt!

## 7.1 Technical requirements

## 7.2 Motion planning using the move\_group C++ interface

Motion planning a random path using MoveIt! C++ APIs

Motion planning a custom path using MoveIt! C++ APIs

Collision checking with a robot arm using MoveIt!

## 7.3 Working with perception using MoveIt! and Gazebo

## 7.4 Performing object manipulation with MoveIt!

Working with a robot pick-and-place task using MoveIt!

Pick-and-place actions in Gazebo and real robots

## 7.5 Understanding DYNAMIXEL ROS servo controllers for robot hardware interfacing

DYNAMIXEL servos

DYNAMIXEL-ROS interface

## 7.6 Interfacing a 7-DOF DYNAMIXEL-based robotic arm with ROS MoveIt!

Creating a controller package for a COOL arm robot

MoveIt! configuration of the COOL arm

## 7.7 Summary

## 7.8 Questions

# Chapter 8: ROS for Aerial Robots

## 8.1 Technical requirements

## 8.2 Using aerial robots

UAV hardware

Pixhawk autopilot

## 8.3 Using the PX4 flight control stack

PX4 firmware architecture

PX4 SITL

## 8.4 PC/autopilot communication

The mavros ROS package

## 8.5 Writing a ROS-PX4 application

Writing a trajectory streamer

External pose estimation for PX4

## 8.6 Using the RotorS simulation framework

Installing RotorS

RotorS packages

Creating a new UAV model

Interacting with RotorS motor models

## 8.7 Summary

## 8.8 Questions

# Chapter 9: Interfacing I/O Board Sensors and Actuators to ROS

## 9.1 Technical requirements

## 9.2 Understanding the Arduino-ROS interface

## 9.3 What is the Arduino-ROS interface?

Understanding the rosserial package in ROS

Understanding ROS node APIs in Arduino

ROS-Arduino Publisher and Subscriber example

Arduino-ROS example – blinking an LED with a push button

Arduino-ROS example – Accelerometer ADXL 335

Arduino-ROS example – ultrasonic distance sensor

Arduino-ROS example – odometry data publisher

## 9.4 Interfacing non-Arduino boards to ROS

Setting up the Odroid-C4, Raspberry Pi 4, and Jetson Nano for installing ROS

Blinking the LED using ROS on the Raspberry Pi 4

A push button and a blinking LED using ROS on the Raspberry Pi 2

Running examples on the Raspberry Pi 4

## 9.5 Interfacing DYNAMIXEL actuators to ROS

## 9.6 Summary

## 9.7 Questions

# Chapter 10: Programming Vision Sensors Using ROS, OpenCV, and PCL

## 10.1 Technical requirements

## 10.2 Understanding ROS–OpenCV interfacing packages

## 10.3 Understanding ROS–PCL interfacing packages

Installing ROS perception

## 10.4 Interfacing USB webcams in ROS

## 10.5 Working with ROS camera calibration

Converting images between ROS and OpenCV using cv\_bridge

## 10.6 Interfacing Kinect and Asus Xtion Pro with ROS

## 10.7 Interfacing the Intel RealSense camera with ROS

Converting point cloud to a laser scan

## 10.8 Interfacing Hokuyo lasers with ROS

Interfacing RPLIDAR and YDLIDAR with ROS

## 10.9 Working with point cloud data

How to publish a point cloud

How to subscribe and process a point cloud

Reading and publishing a point cloud from a PCD file

## 10.10 Summary

## 10.11 Questions

# Chapter 11: Building and Interfacing Differential Drive Mobile Robot Hardware in ROS

## 11.1 Technical requirements

Software requirements

Network setup

Hardware requirements

## 11.2 Introduction to the Remo robot – a DIY autonomous mobile robot

Remo hardware components

Software requirements for the ROS Navigation Stack

## 11.3 Developing a low-level controller and a high-level ROS Control hardware interface for a differential drive robot

Implementing the low-level base controller for Remo

ROS Control high-level hardware interface for a differential drive robot

Overview of ROS nodes and topics for the Remo robot

## 11.4 Configuring and working with the Navigation Stack

Configuring the gmapping node and creating a map

Working with the gmapping node

Configuring the move\_base node

Configuring the AMCL node

AMCL planning

Working with Remo robot in simulation

## 11.5 Summary

## 11.6 Questions

# Chapter 12: Working with pluginlib, nodelets, and Gazebo Plugins

## 12.1 Technical requirements

## 12.2 Understanding pluginlib

Implementing a calculator plugin using pluginlib

## 12.3 Understanding ROS nodelets

Implementing a sample nodelet

## 12.4 Understanding and creating a Gazebo plugin

Creating a basic world plugin

## 12.5 Summary

## 12.6 Questions

# Chapter 13: Writing ROS Controllers and Visualization Plugins

## 13.1 Technical requirements

## 13.2 Understanding ros\_control packages

The controller\_interface package

## 13.3 Writing a basic joint controller in ROS

Step 1 – Creating the controller package

Step 2 – Creating the controller header file

Step 3 – Creating the controller source file

Step 4 – Detailed explanation of the controller source file

Step 5 – Creating the plugin description file

Step 6 – Updating package.xml

Step 7 – Updating CMakeLists.txt

Step 8 – Building the controller

Step 9 – Writing the controller configuration file

Step 10 – Writing the launch file for the controller

Step 11 – Running the controller along with the seven-DOF arm in Gazebo

## 13.4 Understanding the RViz tool and its plugins

The Displays panel

The RViz toolbar

The Views panel

The Time panel

Dockable panels

## 13.5 Writing an RViz plugin for teleoperation

The methodology of building an RViz plugin

## 13.6 Summary

## 13.7 Questions

# Chapter 14: Using ROS in MATLAB and Simulink

## 14.1 Technical requirements

## 14.2 Getting started with MATLAB

## 14.3 Getting started with ROS Toolbox and MATLAB

Starting with ROS topics and MATLAB callback functions

## 14.4 Developing a robotic application using MATLAB and Gazebo

## 14.5 Getting started with ROS and Simulink

Creating a wave signal integrator in Simulink

Publishing a ROS message in Simulink

Subscribing to a ROS topic in Simulink

## 14.6 Developing a simple control system in Simulink

Configuring the Simulink model

## 14.7 Summary

## 14.8 Questions

# Chapter 15: ROS for Industrial Robots

## 15.1 Technical requirements

## 15.2 Understanding ROS-Industrial packages

## 15.3 Goals of ROS-Industrial

## 15.4 ROS-Industrial – A brief history

## 15.5 Installing ROS-Industrial packages

Block diagram of ROS-Industrial packages

## 15.6 Creating a URDF for an industrial robot

## 15.7 Creating the MoveIt configuration for an industrial robot

## 15.8 Updating the MoveIt configuration files

## 15.9 Installing ROS-Industrial packages for Universal Robots arms

## 15.10 Installing the ROS interface for Universal Robots

## 15.11 Understanding the MoveIt configuration of a Universal Robots arm

## 15.12 Getting started with real Universal Robots hardware and ROS-I

## 15.13 Working with MoveIt configuration for ABB robots

## 15.14 Understanding the ROS-Industrial robot support packages

## 15.15 The ROS-Industrial robot client package

## 15.16 Designing industrial robot client nodes

## 15.17 The ROS-Industrial robot driver package

## 15.18 Understanding the MoveIt IKFast plugin

## 15.19 Creating the MoveIt IKFast plugin for the ABB IRB 6640 robot

Prerequisites for developing the MoveIt IKFast plugin

## 15.20 The OpenRave and IKFast modules

MoveIt IKFast

Installing the MoveIt IKFast package

Installing OpenRave on Ubuntu 20.04

## 15.21 Creating the COLLADA file of a robot to work with OpenRave

## 15.22 Generating the IKFast CPP file for the IRB 6640 robot

## 15.23 Creating the MoveIt IKFast plugin

## 15.24 Summary

## 15.25 Questions

# Chapter 16: Troubleshooting and Best Practices in ROS

## 16.1 Setting up Visual Studio Code with ROS

Installing/uninstalling Visual Studio Code

Getting started with Visual Studio Code

Installing new Visual Studio Code extensions

Getting started with the Visual Studio Code ROS extension

Inspecting and building the ROS workspace

Managing ROS packages using Visual Studio Code

Visualizing the preview of a URDF file

## 16.2 Best practices in ROS

ROS C++ coding style guide

## 16.3 Best coding practices for the ROS package

## 16.4 Important troubleshooting tips in ROS

Using roswtf

## 16.5 Summary

## 16.6 Questions