

Robotic Operating System

Chapter 1: Introduction to ROS

1.1 Overview of ROS

1.2 Installing ROS (Robot Operating System) on Linux (Ubuntu)

1.3 ROS File System

1.4 ROS Core Concepts

1.5 Introduction to ROS and its Command

1.6 Line Tools

1.7 Writing and Running a Simple ROS Node in C/C++

Chapter 2: ROS Communication and Parameter Management

2.1 ROS Communication: Topics, Publishers, and Subscribers

2.2 Creating Custom Messages and Services

2.3 Real-Time Implementation of ROS Communication

2.4 ROS Parameter Server: Setting and Accessing Parameters

2.5 Real-Time Example of Setting and Accessing Parameters in ROS

2.6 Working with ROS Services and Actions

2.7 Introduction to ROS Launch Files: Automating Node and Configuration

2.8 ROS Logging and Debugging Tools

Chapter 3: Sensors and Actuators in ROS

3.1 Integration of Sensors with ROS

3.2 ROS Packages for Common Sensor Types

3.3 Introduction to Motor Control with ROS

3.4 Creating URDF Models to Describe Robot Structures

3.5 Visualizing Robots in ROS Using Rviz

3.6 Controlling Simulated Robots and Sensors in Gazebo

Chapter 4: Robot Motion and Path Planning

4.1 ROS Navigation Stack: Overview and Configuration

4.2 Localization with ROS

4.3 Mapping (SLAM) with ROS

4.4 A* (A-Star) Path Planning Algorithm

4.5 Dijkstra's Path Planning Algorithm

4.6 Obstacle Detection and Avoidance in Robotic Motion and Path Planning

4.7 Introduction to MoveIt! for Robotic Arm Manipulation

4.8 Creating an Autonomous Navigation System for a Mobile Robot

Chapter 5: Advanced ROS and Project Implementation

5.1 Introduction to ROS 2

5.2 ROS in Multi-Robot Systems: Communication and Coordination

5.3 Using OpenCV for Computer Vision Tasks in ROS

5.4 ROS-Industrial: Applications in Manufacturing and Industrial Robotics

5.5 Capstone Project: Develop a ROS-Based Solution Integrating Navigation, Manipulation, and Sensor Data

5.6 Case Studies of Real-World Robotic Applications Using ROS