

1. Record Nr.	UNINA9910253961403321
Titolo	Robot Operating System (ROS) : The Complete Reference (Volume 1) // edited by Anis Koubaa
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2016
ISBN	3-319-26054-5
Edizione	[1st ed. 2016.]
Descrizione fisica	1 online resource (XIII, 728 p. 352 illus., 266 illus. in color.)
Collana	Studies in Computational Intelligence, , 1860-949X ; ; 625
Disciplina	006.3
Soggetti	Robotics Automation Computational intelligence Artificial intelligence Robotics and Automation Computational Intelligence Artificial Intelligence
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di contenuto	Intro -- Preface -- Acknowledgements -- Acknowledgements to Reviewers -- Contents -- Part I ROS Basics and Foundations -- MoveIt!: An Introduction -- 1 Introduction -- 2 A Brief History -- 3 MoveIt! Architecture -- 3.1 Collision Checking -- 3.2 Kinematics -- 3.3 Motion Planning -- 3.4 Planning Scene -- 3.5 3D Perception -- 3.6 Trajectory Processing -- 3.7 Using This Tutorial -- 3.8 Installing MoveIt! -- 4 Starting with MoveIt!: The Setup Assistant -- 4.1 Start -- 4.2 Generating the Self-Collision Matrix -- 4.3 Add Virtual Joints -- 4.4 Planning Groups -- 4.5 Robot Poses -- 4.6 Passive Joints -- 4.7 Adding End-Effectors (Optional) -- 4.8 Configuration Files -- 5 Using the Rviz Motion Planning Plugin -- 5.1 Visualization and Interaction -- 5.2 Useful Hints -- 6 The move_group_interface -- 6.1 Planning to a Pose Goal -- 6.2 Planning to a Joint Goal -- 6.3 Move to Joint or Pose Goals -- 6.4 Adding Objects into the Environment -- 6.5 Helpful Hints -- 6.6 Additional Resources -- 7 Connecting to a Robot -- 7.1 Configuring the Controller Interface -- 7.2 Debugging Hints -- 7.3 Integrating 3D

Perception -- 7.4 Helpful Hints -- 8 Building Applications with MoveIt!
 -- 9 Conclusion -- References -- Hands-on Learning of ROS Using
 Common Hardware -- 1 Introduction -- 2 Background -- 3 ROS
 Environment Configuration -- 4 Camera Sensors: Driver, Use and
 Calibration -- 5 Custom Node and Messages for Image Processing with
 OpenCV -- 6 RGB-D Sensors and PCL -- 7 Actuator Control: Dynamixel
 and ROS Control -- 8 Robot Description with URDF -- 9 Motion
 Planning with MoveIt! -- 10 Robot Simulation with Gazebo -- Reference
 -- Threaded Applications with the roscpp API -- 1 Introduction -- 2
 ROS Environment Configuration -- 3 Catkin Build System -- 3.1
 Creating package.xml -- 3.2 CMake Setup -- 3.3 Message Generation
 -- 4 ROS Callback Functions.
 4.1 Basic Callback Functions -- 4.2 Robots as a Thread -- 5 GUI
 Programming with Qt5 and ROS -- 5.1 An Overview of Qt5 GUI
 Programming -- 5.2 Connecting the Robot to the GUI -- 5.3 Publishing
 the Velocity Messages -- 5.4 Results -- References -- Part II
 Navigation, Motion and Planning -- Writing Global Path Planners
 Plugins in ROS: A Tutorial -- 1 Introduction -- 2 ROS -- 2.1 ROS
 Navigation Stack -- 3 Relaxed A* -- 4 Integration Steps -- 4.1 Writing
 the Path Planner Class -- 4.2 Writing Your Plugin -- 4.3 Running the
 Plugin -- 4.4 Testing the Planner with RVIZ -- 5 ROS Environment
 Configuration -- 6 Experimental Validation -- References -- A
 Universal Grid Map Library: Implementation and Use Case for Rough
 Terrain Navigation -- 1 Introduction -- 2 Overview -- 2.1 Prerequisites
 and Installation -- 2.2 Software Components -- 2.3 A Simple Example
 -- 3 Package Description -- 3.1 Adding, Accessing, and Removing
 Layers -- 3.2 Setting the Geometry and Position -- 3.3 Accessing Cells
 -- 3.4 Moving the Map -- 3.5 Basic Layers -- 3.6 Iterating over Cells --
 3.7 Using Eigen Functions -- 3.8 Creating Submaps -- 3.9 Converting
 from and to ROS Data Types -- 3.10 Adding Data from Images -- 4 Use
 Case: Elevation Mapping -- 4.1 Background -- 4.2 Implementation --
 4.3 Results -- 5 Summary and Conclusion -- References -- ROS
 Navigation: Concepts and Tutorial -- 1 Introduction -- 2 Background
 -- 3 ROS Environment -- 3.1 Configuring the Kinect Sensor -- 3.2 Sick
 S300 Laser Sensor -- 3.3 Transformations -- 3.4 Creating a Package --
 3.5 The Navigation Stack---System Overview -- 3.6 The Navigation
 Stack---Getting It to Work -- 3.7 Layered Costmaps -- 4 Starting with
 a Test -- 4.1 Using Rviz -- 4.2 Multiple Machines Communication --
 4.3 Real Tests on Pioneer 3-AT -- References.
 Localization and Navigation of a Climbing Robot Inside a LPG Spherical
 Tank Based on Dual-LIDAR Scanning of Weld Beads -- 1 Introduction
 -- 2 AIR---Autonomous Inspection Robot -- 3 Localization Problems
 in Spherical Tanks -- 4 Weld Beads Scanning with LIDAR Sensors -- 5
 Localization Inside LPG Spheres -- 5.1 LIDAR Based Odometry -- 6
 Experimental Results -- 6.1 Experiments on Planar Environment -- 6.2
 Experiments in LPG Spherical Tank -- 7 Navigation -- 7.1 Wheel
 Odometry -- 7.2 EKF -- 7.3 Simultaneous Localization and Mapping---
 SLAM -- 8 Conclusions -- 9 Compliance with Ethical Standards --
 References -- Part III Service and Experimental Robots -- People
 Detection, Tracking and Visualization Using ROS on a Mobile Service
 Robot -- 1 Introduction -- 2 Background of the SPENCER Project -- 2.1
 Robot Hardware and Sensory Setup -- 2.2 People Tracking Pipeline -- 3
 People Detection -- 3.1 ROS Message Definitions -- 3.2 Person
 Detection in 2D Laser Data -- 3.3 Person Detection in RGB-D -- 3.4
 Person Detection in Monocular Vision -- 4 People Tracking -- 4.1 ROS
 Message Definitions -- 4.2 People Tracking Algorithms Used in Our
 Experiments -- 4.3 Example: Nearest-Neighbor Tracker -- 4.4
 Improving Robustness of Tracking -- 4.5 Tracking Metrics -- 5 Group

Tracking -- 5.1 Social Relation Estimation -- 5.2 Group Detection and Tracking -- 6 Multi-Modal Tracking -- 6.1 ROS Message Definitions -- 6.2 Strategies for Fusion at the Detection Level -- 6.3 Post-Processing Filters -- 6.4 Multi-Modal People Tracking Setup on the Robot -- 6.5 Exemplary Launch File -- 7 Visualizing the Outputs of the Perception Pipeline -- 7.1 Custom RViz Visualization Plugins -- 7.2 URDF Model for Robot Visualization -- 7.3 ROS-based SVG Exporters -- 8 Integration with 3rd-Party Simulation Tools -- 8.1 Integration with the Robot Simulator Gazebo. 8.2 Integration with the Pedestrian Simulator PedSim -- 9 Results -- 9.1 Qualitative Results -- 9.2 Runtime Performance -- 9.3 Lessons Learned -- 10 Conclusion -- References -- A ROS-Based System for an Autonomous Service Robot -- 1 Introduction -- 2 ROS Environment Configuration -- 3 Graphical User Interface -- 3.1 Background -- 3.2 ROS Environment Configuration -- 3.3 Quick Start and Example -- 3.4 Package Description -- 4 Mapping and Navigation -- 4.1 Background -- 4.2 ROS Environment Configuration -- 4.3 Quick Start and Example -- 4.4 Using the homer_gui for Mapping and Navigation -- 4.5 Package Description and Code Examples -- 5 Object Recognition -- 5.1 Background -- 5.2 ROS Environment Configuration -- 5.3 Quick Start and Example -- 5.4 Using the homer_gui for Object Learning and Recognition -- 5.5 Package Description and Code Examples -- 6 Human Robot Interaction -- 6.1 Background -- 6.2 ROS Environment Configuration -- 6.3 Quick Start and Example -- 6.4 Package Description and Code Examples -- 7 Conclusion -- References -- Robotnik---Professional Service Robotics Applications with ROS -- 1 Contributions of the Book Chapter -- 2 RESCUER: Robot for CBRN Intervention -- 2.1 Brief Description of the System -- 2.2 Challenges -- 3 R-INSPECT: Mobile Robot for Tunnel Inspection -- 3.1 Brief Description of the System -- 3.2 Challenges -- 4 CROM: Upper Body Torso Robot -- 4.1 Brief Description of the System -- 4.2 Main Topics Covered -- 5 AGVS: Indoor Healthcare Logistics Transport Robot -- 5.1 Brief Description of the System -- 5.2 Main Topics Covered -- 5.3 Navigation -- 5.4 Challenges -- 6 VINBOT: Robot for Precision Viticulture -- 6.1 Brief Description of the System -- 6.2 Challenges -- 7 Summary and Conclusions -- References -- Standardization of a Heterogeneous Robots Society Based on ROS -- 1 Introduction -- 2 Robot Description. 2.1 MariSorgin -- 2.2 Tartalo and Galtxagorri -- 2.3 Robotino-s -- 2.4 NAO -- 2.5 Kbot-I -- 3 Working Areas of RSAIT Research Group -- 4 Case Study 1: Setup of the Navigation Stack -- 5 Case Study 2: Kinect Based Teleoperation -- 5.1 The robotino_teleop_gesture Package -- 5.2 The nao_teleop_gesture Package -- 6 Case Study 3: Speech Based Teleoperation in Basque -- 6.1 Speech-Based Teleoperation in MariSorgin -- 6.2 The nao_teleop_speech_eus Package -- 7 Conclusions -- References -- Part IV Real-World Applications Deployment -- ROS-Based Cognitive Surgical Robotics -- 1 Introduction -- 2 Background -- 3 ROS Environment Configuration -- 4 Components -- 4.1 Robots -- 4.2 Endoscope Cameras -- 4.3 OR Perception System -- 4.4 Marker-Based Optical Tracking -- 4.5 Time-of-Flight Cameras -- 4.6 RGB-D Cameras -- 4.7 Input Devices -- 4.8 OpenIGTLink-ROS-Bridge -- 4.9 Ultrasound Imaging -- 4.10 Surgical Instruments -- 4.11 Augmented Reality -- 5 Subsystems -- 5.1 Telemanipulation -- 5.2 Multi-RGBD People Tracking -- 5.3 Human-Robot-Interaction -- 5.4 Endoscope Guidance -- 5.5 Ultrasound Tomography -- 5.6 Simulation -- 5.7 Software Frameworks -- 6 Organization and Software Engineering -- 6.1 Registration and Calibration -- 6.2 TF and Pose Topics -- 6.3 Windows/Matlab

Integration -- 6.4 Software Repositories and Configuration Management -- References -- ROS in Space: A Case Study on Robonaut 2 -- 1 Introduction -- 2 Architecture Overview -- 2.1 System Architecture -- 2.2 Software Architecture -- 3 Control Software -- 3.1 RoboNet -- 3.2 JointApi -- 3.3 Robodyn -- 4 Safety System and Certification -- 4.1 Static Force -- 4.2 Dynamic Force -- 4.3 Crushing Force -- 4.4 Health Monitoring -- 4.5 Trajectory Monitoring -- 4.6 Inadvertent Release -- 4.7 Certification -- 5 Vision and Supervisory Elements -- 6 Simulation and User Interfaces -- 6.1 Simulation. 6.2 Affordance Templates.

Sommario/riassunto

The objective of this book is to provide the reader with a comprehensive coverage on the Robot Operating Systems (ROS) and latest related systems, which is currently considered as the main development framework for robotics applications. The book includes twenty-seven chapters organized into eight parts. Part 1 presents the basics and foundations of ROS. In Part 2, four chapters deal with navigation, motion and planning. Part 3 provides four examples of service and experimental robots. Part 4 deals with real-world deployment of applications. Part 5 presents signal-processing tools for perception and sensing. Part 6 provides software engineering methodologies to design complex software with ROS. Simulations frameworks are presented in Part 7. Finally, Part 8 presents advanced tools and frameworks for ROS including multi-master extension, network introspection, controllers and cognitive systems. This book will be a valuable companion for ROS users and developers to learn more ROS capabilities and features.
