Course guide
240AR060 - 240AR060 - Introduction to Ros

Last modified: 08/04/2026

Unit in charge: Barcelona School of Industrial Engineering
Teaching unit: 707 - ESAII - Department of Automatic Control.

Degree: MASTER'S DEGREE IN AUTOMATIC CONTROL AND ROBOTICS (Syllabus 2012). (Optional subject).

Academic year: 2026 **ECTS Credits:** 4.5 **Languages:** English

LECTURER

Coordinating lecturer: Rosell Gratacos, Joan

Others: Rosell Gratacos, Joan
Palomo Avellaneda, Leopold

PRIOR SKILLS

The student should have basic skills in C++ programming as well as linux common tools and commands. An overall understanding of software processes involved in robotics will be welcomed.

TEACHING METHODOLOGY

The teaching methodology will combine lectures together with supervised exercises based on the current ROS version and tools. All classes will be organized with the theoretical sessions at the beginning and practical exercises and team work at the end. The initial part will consist on the explanation of theoretical concepts by the lecturer, promoting the active participation of students. The practical part will be focused on the student's solving skills. The main theoretical concepts will be shown in practical simulation examples and finally on a real robot test.

COVID'19: For the academic year 2019-2020 (Q2) all the lectures not able to be done face-to-face will be taught using Google Meet during the regular scheduled time slots. Moreover the introductory theoretical explanations of each session will be provided in video through ATENEA. Personal advice will be done by e-mail, through a virtual forum in ATENEA and with arranged personal meetings using Google Meet.

LEARNING OBJECTIVES OF THE SUBJECT

The objective of this course is to introduce students in the use of ROS as a powerful robotics tool. Specifically a familiarization with the middleware concept and the software structure of a robot. There will be a special emphasis on sensing and control of robots using ROS, both in simulation and in real environments.

Learning Outcomes:
Learn how to setup a Linux O.S. environment to work with ROS.
Understand the ROS communications architecture.
Use ROS in the different process layers, from sensing to control or actuation.
Implement simple ROS projects with both simulation and real robots.

Mandatory contents:
Install and setup ROS in a native O.S. Linux (Ubuntu).
Know and understand the internal procedures of ROS and its modules functionalities (master,nodes, and so on).
Identify and use the ROS tools and formats related to the internal communication between nodes (topics, actions, services,...).
Use ROS visualization and debugging tools.
Design and program C++ algorithms using ROS as a middleware.
Use debugging tools to verify the compilation and the algorithm functionalities.
Configure and use a simulation environment with the designed algorithms.
Managing acquisition, analysis and display of data obtained from different sensors using ROS.
Manage and send control commands to a robot using ROS, both using simulation and real settings.

STUDY LOAD

Type	Hours	Percentage
Hours small group	13,5	12.00
Hours large group	27,0	24.00
Self study	72,0	64.00

Total learning time: 112.5 h

CONTENTS

1. ROS Basic concepts

Description:
1.1. Introduction
1.2. ROS core components
1.3. Applications
1.4. Install instructions
1.5. ROS command-line tools

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

2. Development Tools

Description:
2.1. Programming
2.2. Building executables
2.3. The ROS build system
2.4. Good practices
2.5. Version control using GIT

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

3. Communications using topics

Description:
3.1. An example: The package agitr_chapter3
3.2. A publisher program
3.3. A subscriber program
3.4. Standard and common messages

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

4. The launch utility

Description:
4.1. Using launch files
4.2. Understanding launch files
4.3. Graph resource names
4.4. Managing names in launch files
4.5. ROS parameters

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

5. Communications using services

Description:
5.1. Services
5.2. The package agitr_chapter8
5.3. A client program
5.4. A server program
5.5. Standard services
5.6. Defining non-standard services

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

6. Tools

Description:
6.1. The tf tool
6.2. Robot Modeling and visualization tools
6.3. The rosbag Tool
6.4. The rqt tool

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

7. Communications using actions

Description:
7.1. Working with ROS actionlib
7.2. Building and running a simple example
7.3. The ROS action server
7.4. The ROS action client
7.5. The pan-tilt example

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

8. Simulation - basic issues

Description:
8.1. Gazebo basics
8.2. Integration to ROS
8.3. Configuring launch files
8.4. ROS-aware Gazebo plugins
8.5. Tunning URDF models

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

9. Simulation - sensors

Description:
9.1. Available ROS plugins
9.2. The camera ROS plugin
9.3. The depth camera ROS plugin
9.4. ROS plugins for some other sensors

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

10. Robot control

Description:
10.1. ros_control overview
10.2. Controllers
10.3. Hardware Abstraction Layer
10.4. Using ros_control in Gazebo

Full-or-part-time: 5h
Practical classes: 3h
Self study : 2h

Case study

Description:
Definition of the solution
Sensing module
Planning module
Action module

Full-or-part-time: 15h
Practical classes: 9h
Self study : 6h

GRADING SYSTEM

Any academic fraud, plagiarism or use or possession of unauthorised means in an assessment activity leads to a mark of zero (0) for the affected test or assignment. Furthermore, in accordance with the University's regulations, if disciplinary proceedings are initiated the subject is recorded as "Pending Assessment" until a decision is reached. These incidents are managed in accordance with the **Action Framework for Academic Integrity in Assessment at the UPC**.

The acquired competences and capabilities will be assessed on the basis of three qualification grades: exercises (20%), deliverable (20%) and final project (60%).
Re-evaluation: new final project (80%).

BIBLIOGRAPHY

Basic:
- Martín Rico, F. A concise introduction to robot programming with ROS2 [on line]. 1st. Boca Ratón, Florida: CRC Press, Taylor & Francis, 2023 [Consultation: 11/06/2025]. Available on: https://www-taylorfrancis-com.recursos.biblioteca.upc.edu/books/mono/10.1201/9781003289623/concise-introduction-robot-programming-ros2-francisco-mart%C3%ADn-rico. ISBN 9781032264653.

RESOURCES

Hyperlink:
- Introduction to ROS: online tutorials. https://sir.upc.edu/projects/rostutorials/index

Other resources:
ROS 2 Documentation: https://docs.ros.org/en/humble/index.html />Paper: Robot Operating System 2: Design, architecture, and uses in the wild.
Gazebo tutorials: https://gazebosim.org/docs/harmonic/tutorials />Git tutorial: https://try.github.io/ https://guides.github.com/activities/hello-world/ />