

240AR060 - Introduction to Ros

Coordinating unit: 240 - ETSEIB - Barcelona School of Industrial Engineering
Teaching unit: 915 - IRI - Institute of Robotics and Industrial Informatics
Academic year: 2017
Degree: MASTER'S DEGREE IN AUTOMATIC CONTROL AND ROBOTICS (Syllabus 2012). (Teaching unit Optional)
ECTS credits: 4,5 Teaching languages: English

Teaching staff

Coordinator: RAMON COSTA CASTELLO - GUILLEM ALENYÀ RIBAS
Others: Segon quadrimestre:
SERGIO FOIX SALMERON - 10

Opening hours

Timetable: To be defined

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 requiring the use of their own computers running Ubuntu and ROS. The main theoretical concepts will be shown in practical simulation examples and finally on a real robot test.

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 a robot using ROS, both in simulation and 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,...).

240AR060 - Introduction to Ros

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 (cameras, IMU, and so on), both using simulation and real settings.

Manage and send control commands to a robot using ROS (parrot ARdrone), both using simulation and real settings.

Study load

Total learning time: 0h	Hours large group:	0h	0%
	Hours medium group:	0h	0%
	Hours small group:	0h	0%
	Guided activities:	0h	0%
	Self study:	0h	0%

240AR060 - Introduction to Ros

Content

<p>ROS introduction</p>	<p>Learning time: 5h Theory classes: 3h Guided activities: 0h Self study : 2h</p>
<p>Description: What is ROS. Why, when and where to use ROS. Introduction to basic concepts of software processes, system and design of robotic architectures. ROS community. Pros and cons of using ROS in a robot. Overview of current famous robots using ROS.</p> <p>Related activities: Lectures and ROS wiki page.</p> <p>Specific objectives: Specific introduction to ROS characteristics. Presentation of the ROS community, forum and resources.</p>	
<p>ROS environment configuration in a Linux O.S.</p>	<p>Learning time: 5h Theory classes: 1h 30m Guided activities: 1h 30m Self study : 2h</p>
<p>Description: Installation of ROS in a Linux environment (Ubuntu).</p> <p>Related activities: Lectures and ROS installation tutorial.</p> <p>Specific objectives: How ROS uses the system, Which part is done by ROS and which by the underlying operating system. Installation of ROS main packages and dependencies as well as those uses during the course.</p>	
<p>Introduction to ROS tools. Visualisation, analysis and debug. Practical application (using ?.bag? files).</p>	<p>Learning time: 16h Theory classes: 6h Guided activities: 6h Self study : 4h</p>
<p>Description: Introduction to ROS developer tools. Command line tools and GUIs</p> <p>Related activities: Lectures and guided exercises.</p> <p>Specific objectives: Frame transformations (tf). 3D visualization (rviz). Logging and visualization sensor data (roscap and rqt_bag). Live plotting (rqt_plot). System visualization (rqt_graph). Command line tools. Move it!. ros_control.</p>	

240AR060 - Introduction to Ros

<p>Introduction to ROS nodes and communications.</p>	<p>Learning time: 24h Theory classes: 6h Guided activities: 8h Self study : 10h</p>
<p>Description: Description of ROS communications scheme. ROS package (node) creation.</p> <p>Related activities: Lectures and guided exercises.</p> <p>Specific objectives: ROS package filesystem and architecture. Topics. Publishers and Subscribers. Messages. Services. Actions. Master. Parameter Server. Dynamic reconfigure. Creation of two nodes. Addition of message publishers and subscribers. Addition of an action and a service (server and client). Callbacks and spinning (single threaded, multi-threaded or asynch-spinner). Multiple machines with unique master.</p>	
<p>Configuration and use of a simulation environment.</p>	<p>Learning time: 9h Theory classes: 1h Guided activities: 4h Self study : 4h</p>
<p>Description: Presentation of the simulated environment to be used during the subject. Introduction to Gazebo. Overview of a simulated robot.</p> <p>Related activities: Lectures and guided exercises.</p> <p>Specific objectives: Use of Gazebo with a simulated robot and world. Definition of a robot (urdf) and sensors (plugins). Interaction between ROS and Gazebo. Robot control using created nodes.</p>	

240AR060 - Introduction to Ros

<p>Managing acquisition, analysis and display of data obtained from different sensors.</p>	<p>Learning time: 7h Theory classes: 1h Guided activities: 2h Self study : 4h</p>
<p>Description: Use of ROS tools to acquire data from sensors (simulation and real setting).</p> <p>Related activities: Lectures and guided exercises.</p> <p>Specific objectives: Display of sensor data using a terminal or visualization tools. Use a simulated camera and detect markers in the image.</p>	
<p>Using ROS in a real setting (Robot).</p>	<p>Learning time: 20h Theory classes: 2h Guided activities: 2h Self study : 16h</p>
<p>Description: Final project presentation. Apply of acquired concepts during the course.</p> <p>Related activities: Lectures, guided exercises and final project statement.</p> <p>Specific objectives: Robot introduction. Communications between robot and laptops. Sensor acquisitions. Robot control. Final project with public presentation.</p>	

Qualification system

The acquired competences and capabilities will be assessed on the basis of four qualification grades: practical work with periodic reports that must be delivered during the course (20%), a formal final project report (40%), the corresponding project exhibition (20%), and a discretionary teamwork evaluation (20%).

Re-evaluation: new final project (40%).

240AR060 - Introduction to Ros

Bibliography

Others resources:

Basic:

Lectures slides.

Description of case studies, exercises and guides.

ROS wiki page: <http://wiki.ros.org/>

Complementary:

ROS tutorials. <http://wiki.ros.org/ROS/Tutorials>

Representing Robot Pose, the good, the bad and the ugly. Paul Furgale, ETH Zürich. <http://goo.gl/gcQsXn>

A gentle Introduction to ROS, Jason M. O'Kane, 2013. <http://www.cse.sc.edu/~jokane/agitr/>

ROS Cheat Sheet (from Hydro version and catkin). <http://clearpath.wpengine.netdna-cdn.com/wp-content/uploads/2014/01/ROS-Cheat-Sheet-Landscape-v2.pdf>

Hyperlink

Nom recurs

Resource