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.

<table>
<thead>
<tr>
<th>Study load</th>
<th>Hours large group:</th>
<th>Hours medium group:</th>
<th>Hours small group:</th>
<th>Guided activities:</th>
<th>Self study:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total learning time:</strong> 0h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0%</td>
</tr>
</tbody>
</table>
## 240AR060 - Introduction to Ros

### Content

<table>
<thead>
<tr>
<th>ROS introduction</th>
<th>Learning time: 5h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>Theory classes: 3h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study : 2h</td>
</tr>
</tbody>
</table>

**Related activities:**
- Lectures and ROS wiki page.

**Specific objectives:**
- Specific introduction to ROS characteristics. Presentation of the ROS community, forum and resources.

<table>
<thead>
<tr>
<th>ROS environment configuration in a Linux O.S.</th>
<th>Learning time: 5h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>Theory classes: 1h 30m</td>
</tr>
<tr>
<td>Installation of ROS in a Linux environment (Ubuntu).</td>
<td>Guided activities: 1h 30m</td>
</tr>
<tr>
<td></td>
<td>Self study : 2h</td>
</tr>
</tbody>
</table>

**Related activities:**
- Lectures and ROS installation tutorial.

**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.

<table>
<thead>
<tr>
<th>Introduction to ROS tools. Visualisation, analysis and debug. Practical application (using .bag files).</th>
<th>Learning time: 16h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>Theory classes: 6h</td>
</tr>
<tr>
<td>Introduction to ROS developer tools. Command line tools and GUIs</td>
<td>Guided activities: 6h</td>
</tr>
<tr>
<td></td>
<td>Self study : 4h</td>
</tr>
</tbody>
</table>

**Related activities:**
- Lectures and guided exercises.

**Specific objectives:**
- Frame transformations (tf). 3D visualization (rviz). Logging and visualization sensor data (rospack and rqt_bag).
## Introduction to ROS nodes and communications.

**Learning time:** 24h  
- Theory classes: 6h  
- Guided activities: 8h  
- Self study: 10h

**Description:**  
Description of ROS communications scheme. ROS package (node) creation.

**Related activities:**  
Lectures and guided exercises.

**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 async-spinner)  
- Multiple machines with unique master

## Configuration and use of a simulation environment.

**Learning time:** 9h  
- Theory classes: 1h  
- Guided activities: 4h  
- Self study: 4h

**Description:**  
Presentation of the simulated environment to be used during the subject. Introduction to Gazebo. Overview of a simulated robot.

**Related activities:**  
Lectures and guided exercises.

**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
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 discreetional 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:

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/

Hyperlink
Nom recurs
Resource