Robotics 2018/AMR 2019

http://www.rsl.ethz.ch/education-students/lectures/ros.html

Abstract: This course gives an introduction to the AMR/ Robot Operating System (ROS) including many of the available tools that are commonly used in robotics. With the help of different examples, the course should provide a good starting point for students to work with robots. They learn how to create software including simulation, to interface sensors and actuators, and to integrate control algorithms.

Robotics 2018

- Objective:
- ROS architecture: Master, nodes, topics, messages, services, parameters and actions
- Console commands: Navigating and analyzing the ROS system and the catkin workspace
- Creating ROS packages: Structure, launch-files, and best practices
- ROS C++ client library (roscpp): Creating your own ROS C++ programs
- Simulating with ROS: Gazebo simulator, robot models (URDF) and simulation environments (SDF)
- Working with visualizations (RViz) and user interface tools (rqt)
- Inside ROS: TF transformation system, time, bags
-

Robotics 2018

- **Content:** This course consists of a guided tutorial and exercises with increasing level of difficulty when working with an autonomous robot. You learn how to setup such a system from scratch using ROS, how to interface the individual sensors and actuators, and finally how to implement first closed loop control systems.

Robotics 2018 : Lecture 1

- ROS architecture & philosophy
- ROS master, nodes, and topics
- Console commands
- Catkin workspace and build system
- Launch-files
- Gazebo simulator
-

Robotics 2018 : Lecture 2

- ROS package structure
- Integration and programming with Eclipse
- ROS C++ client library (roscpp)
- ROS subscribers and publishers
- ROS parameter server
- RViz visualization

Robotics 2018 : Lecture 3

- TF Transformation System
- rqt User Interface
- Robot models (URDF)
- Simulation descriptions (SDF)

Robotics 2018 : Lecture 4

- ROS services
- ROS actions (actionlib)
- ROS time
- ROS bags
- Debugging strategies

Robotics 2018 : Preparation

In this course, we will work with Ubuntu 16.04 and ROS Kinetic Kame. We highly recommend you to install the following software:

- Ubuntu 16.04 https://www.ubuntu.com/download/desktop
- ROS Kinetic Kame http://wiki.ros.org/kinetic/Installation
- Qt IDE https://www.qt.io/download
- Qt ROS Plugin https://github.com/ros-industrial/ros_qtc_plugin
- Catkin Command Line Tools sudo apt-get install ros-kinetic-catkin
- Terminator elkepon.com/install-terminator-terminal-emulator-in-ubuntu/
- Git sudo apt-get install git-core

Robotics 2018 : Missing topics

- Robot Dynamics
- Robot Kinematics
- Robot Navigation A* Djikstra
- SLAM simultaneous localization and Mapping
- Industrial ROS
- MoveIt
- Mapping using gmapping, hector mapping
- Creating robot in solid work and urdf pluggin

Robotics 2018 :
Final Exam (Lecture 5)

- Navigate you Robot
- Autonomous : robo jacket
- Autonomous : Husky
- Waypoints : move to points, Husky Turtlebot3
- MoveIt : Plan and Execute. Arm robot
- Mapping using gmapping
- Creating World and navigate, Husky turtlebot
- Case study: Using ROS in complex real-world applications

Robotics 2018 :
Final Exam (Lecture 5)

- MIR100 (gio, fendi)
- ROS-AMR (Dwi, Kevin Sukamto)
- HUSKY (Edrick, Wilvan)
- TURTLEBOT3 (Ezra), aveka, daffa)
- LINOROBOT (Michael)(Rich, William)
- ROBOJACKET (Fredric, marvin)
- ROBOTNIKAUTOMATION :guardian\ ()
- Autoware.ai (Kris,)
- Waymo (fabio)