UNLV | Howard R. Hughes College of Engineering

DASL

Drones and Autonomous Systems Lab @ UNLV

DASL 107 – Introduction to ROS

Instructor: Keitaro Nishimura

B.S. Mechanical Engineering (‘18)

Cell: (310)729-8216, Email: nishik1@unlv.nevada.edu

Office Hours: TBD or by appointment

Classroom: Drones and Autonomous Systems Lab classroom

Textbook: There is no textbook, but lecture notes will be published on class page on the daslhub (www.daslhub.org) wiki page.

Pr-requisites: Having access to a computer with internet access. Basic understanding of C++ or python.

Homework: Homework is given weekly and is expected to be finished before the next class.

Course Description:

This course is designed to teach students without any prior experience with Robot Operating System (ROS) so that they can use it for future projects and research. This is a homework intensive course and expects the student to learn a lot outside of class. This course is intended to be taught over an 8 week period.

Learning Outcomes:

- Be able to understand how a ROS package works and use it
- Be able to make their own ROS package
- Be prepared to learn more advanced usages of ROS for research

Grading:

Homework	60%
Final	40%
Participation (extra credit)	Max~10%
Total	100%

A (90%~100~), B (80%~89%), C (70%~79%), D (60%~69%), F (~59%)

Schedule:

Week	Topic	HW
1	-Overview of class • What is the point of the class? ○ Why is this class important to robotics? • Explanation of the software used for the class ○ What is ROS? (history of why it was made and what it is today) ○ What is Github ○ What is ■ gedit ■ vim ■ sublime ■ Atom, etc. ■ latex(optional) • What I expect from the students ○ To not only follow class but to work on their own independently or in groups on understanding the material throughout the week.	1. Get ROS-Kinetic 2. Get Ubuntu 16.04 3. Create Github repo to submit hw to and email me the link to it. 4. Decide on what text editor to use and learn it if need be
2	-How to use ROS commands to find out how a package works. -ROS beginner tutorials 1-9	-Download 2 simple or 1 complex ros package(s) that students haven't used before and using command line tools and their understanding of ros explain to students how it works. (a high level explanation based on ros nodes) -upload txt file with explation to github
3	-How to make and trouble shoot ros nodes (services, msgs, subscribers, publishers) -ROS beginner tutorials 10-20	-Make 2 different ROS packages that use 2 different ros msgs with at least 1 subscriber and 1 publisher each. -upload packages to github with rosbag files

4	-ROS intermediate tutorials 1-8	-Make 2 ros packages to communicate across 2 different machines. Download the snap file from github to install on second computer. (this may require working in a group) -upload both packages to github with rosbag files
5	-Lecture on why ROS2/G/M/I exist. -introduction to ros2 specifically and the benefits that it has over ros1 where it is going	-download and install ros2 from source -run listener/talker to ensure installation was successful -upload txt file to github to declare success
6	-Going over current tutorials on ROS2 and the ros 1 bridge.	-using ros 1-2bridge to communicate between a windows ros 2 and ubuntu ros 1 nodes. -upload packages (ros2&1) to github with rosbag files
7	Extra day to go over material or as an extra day if some topic takes more time than expected	-go over past material and hw
8	Final	-2 hr exam -(something to do with both ros1&2) -You will have internet access and be able to use past code