F1/10th Onboarding:

Introduction to Autonomous Vehicles with ROS

CRSN-151C

The Introduction to Autonomous Vehicles with ROS module of CRSN-151C aims to introduce you to the theory and software behind controlling vehicles autonomously. This module introduces the software and engineering concepts for development of custom hardware applications exemplified through the use case of scaled autonomous vehicles. Throughout the course, we will cumulatively develop the skills required to achieve safety-minded autonomous driving algorithms. By learning how to design systems with the Robot Operating System (ROS), students will use LIDAR data to interact with the world around a simulated autonomous vehicle. By the end of the course, the students will implement multiple autonomous driving methods, namely “Wall Following” and “Follow the Gap”. Students completing this course will be positioned to implement autonomous algorithm development on their own projects or may elect to further develop on the car that will be competing in the Formula 1/10th competition as part of the team. It is highly recommended that incoming students are familiar with an object-oriented programming language, as we use Python extensively in this course.

Meeting Information

Tuesdays and Thursdays @ 11:40 - 1:15 pm:
Meeting ID:
Passcode:

This course is asynchronous (attendance is encouraged), and recordings can be found on canvas

Lab Instructors Contact Info

Neal Chokshi          Sriram Venkataraman
Slack: @nkchoksh       Slack: @svenka12
Email: nkchoksh+151@ucsc.edu  Email: svenka12+151@ucsc.edu

Office Hours and Communication

Canvas will be used as the primary repository for course content and announcements. Slack will be used for more informal supplementary communication as well as Q&A. In addition to the regularly scheduled Class lectures/labs, each student-instructor will also have regular office hours that are schedulable through canvas appointments.

https://intro2auto.slack.com

If you need to speak with a student-instructor outside of normal office hours, feel free to email or DM them in Slack to set up a meeting.
Learning Objectives
- Good laboratory Environment, Health, and Safety practices
- Developing software in a Linux (Ubuntu) environment
- Designing multi-sensor systems with the ROS framework
- Using LIDAR as a means of vehicle perception
- Using PID to achieve an autonomous wall following algorithm
- Implementing a reactive control method (follow the gap) for better autonomous control
- Designing autonomous systems with Planning, Perception, and Control as major components

Prerequisites
- Working knowledge of at least one object-oriented programming language, preferably Python

Materials
All students participating in this section of CRSN 151C must possess the following pieces of equipment. **Note:** a microphone and/or webcam are highly recommended but not required.

- One composition notebook for diagrams, pseudocode, flow charts, and logging course progress.
- A computer meeting the following requirements:
  - BSOE Laptop Requirements
    - 8GB RAM Minimum (16GB recommended)
    - 250GB Hard Disk or SSD
    - 2Ghz or greater, dual core or greater x86-64 processor
  - Able to run virtualization software
    - For Intel, VT-d is accessible and enabled in the BIOS
    - For AMD, AMD-V is accessible and enabled in the BIOS
  - Administrator access

If your device does not satisfy these requirements, you will be able to remotely connect to pre-configured machines

*Note: Please ensure your hardware is running an x86-64 processor. ARM devices, such as Macs with Apple Silicon, Raspberry Pi, or Pine64 devices will not be able to run the provided virtual machine software.*

Course Software
All students participating in this section of the CRSN 151C must have the following list of software tools installed on their machines. Please follow the official software instruction guide to get the development environment set up.

- **WSL2 (For Windows users only)** - A feature in windows allowing to run fully featured Linux instances without the overhead of virtualization
- **Docker** - A system to package development environments for easy deployment. Our entire development stack has been packaged into a docker container which you will need to pull
- **TigerVNC** - A Virtual Network Computing system to be able to see the graphical interface within our Docker Container.
- **Visual Studio Code** - An IDE which has software which enables us to seamlessly connect to our development environment

Additionally, there will be some software needed to access course material and attend lectures:
- **Zoom**: Video conferencing software used to host and attend lectures and office horse
- **Canvas**: Academic organization app used by UCSC
- **Slack**: Communication and collaboration environment allowing for easy access to communication between instructors and students.

---

**Class Structure and Zoom**

**Tuesdays:**

The lectures introduce the fundamentals of ROS, programming concepts, and equipment behind autonomous driving. Each week we will incrementally develop the skills needed to design fully autonomous vehicles.

**Thursdays:**

Reserved for tutorial and prelab workshops in which we will cover prelab material needed to be prepared for the lab component. If there is a short prelab activity, the rest of the meeting will be devoted to office hours to answer any questions you may have about the lab. There may also be lecture content corresponding to the current modules’ lab.

Attendance at both meetings is crucial if you want to succeed in this course, however it is not mandatory. All lectures and workshops will be recorded and available on Canvas.

The course is broken down into a three components:

**Labs** (Labs are worth 70% of your final grade in the course.)

This is the main component of the course, the goal of which is to apply fundamental concepts learned during lectures in ROS. Labs are done in the dedicated development environment and involve writing software to enable the simulated vehicle to perform tasks autonomously. There will be starter code provided for each lab. Labs are assigned on Tuesday and are due the following Tuesday before the start of class, unless specified otherwise.
The labs will be submitted as zip files on Canvas

**Prelabs and Quizzes** (Prelabs are worth 15% of your final grade in the course.)
Most labs will have an associated prelab component, to be completed before starting the lab. These prelab sections will be walked through during Thursday’s class meetings to help boost your understanding of the material before you start the lab, though you may start on that section earlier.

Additionally, there will be prelab quiz due before class begins on Tuesday. Late quizzes are not accepted.

**Notebook** (Notebooks are worth 15% of your final grade in the course.)
The engineering notebook is an integral part of this course. Here you are highly encouraged to take lecture notes and jot down your thought process, flowcharts, pseudocode, and any other information while completing the lab assignments. There are also notebook questions in the lab manuals which are to be completed in your notebooks. Please make a habit to make entries everytime you are working on course material. Notebooks will be graded based on quality and consistency of the content via submission of a scanned copy to the course Canvas. This handout helps to articulate the minimum expectations for the course notebook.

---

**Grading**

- Labs 70%
- Prelabs 15%
- Notebook Checks 15%

**Late Policy:**
Due to the fact that projects will be building on prior content throughout the quarter, it is important that all students **submit their projects on time**. This allows you to gain the ability to improve towards the next assignment based on timely feedback. Time management is important, and the due dates of each assignment have been specifically crafted such that you have time to work on each and improve without needing to cram.

Nevertheless, situations may arise throughout the quarter outside of your control, if such events do occur, please contact one of the two instructors **before** the due date of the assignment has occurred, and individual situations will be considered and adjusted accordingly.

The prelabs and quizzes will not be accepted late.

---

**Schedule**
- **Week 0: (September 19-25)**
  - September 23rd: (General Meeting involving all sections)
- **Week 1: Introduction to ROS (September 26 - October 2)**
  - September 28th: Introduction to course and ROS
  - Review System Software Requirements for your computers
  - Development Environment Setup assigned, due
- **Week 2: F1/10 Simulator and the Perception Stack with LIDAR(October 3 - 9)**
  - Introductory lecture covering the anatomy of ROS
  - Deep dive into each component in ROS with follow along tutorial section pertaining to the week’s lab
  - Lab 1 intro assigned, due one week after
- **Week 3-4: F1/10 Simulator and the Perception Stack with LIDAR(October 10 - 23)**
  - Lectures on the perception stack and introduction of the LIDAR sensor
  - Week 3
    - Introduction to provided simulator system within ROS
    - Intro to the LIDAR sensor within the simulator environment
    - Learn how to interact with the components in the simulator
    - Lab 1 due
    - Lab 2 - LaserScan assigned, due one week after
  - Week 4
    - Exploring safety systems using the LIDAR scanner
    - Examining various safety systems present in industry
    - Lab 2 due
    - Lab 3 - Automatic Emergency Braking assigned, due one week after
- **Week 5-6: Introduction Reactive Methods for Wall Following (October 24-November 6)**
  - Week 5
    - Overview of PID and its various controllers and math in relation to an implementation of a wall following algorithm
    - Lab 3 due
    - Lab 4 - Wall Following assigned, due two weeks after
  - Week 6
    - Bonus control theory, Kalman Filters
- **Week 7-8: Follow the Gap (November 7 - 20)**
  - Week 7
    - Introduction to a few Follow the Gap techniques
    - Exploration of the benefits and limitations of Follow the Gap
    - Lab 4 due
    - Lab 5 - Follow the Gap, due two weeks after
  - Week 8
    - Potential improvements to the Follow the Gap algorithm
- **Week 9: Further Topics in Autonomous Vehicles (November 21-27)**
  - Lab 5 due
- Final Project- Optimize Follow the Gap, due two weeks after
- Discuss more advanced topics such as localization and mapping, and pure pursuit

- **Week 10: Conclusion + All Questions Answered (November 28 - December 4)
  - Final Project Due at End of Finals Window (December 8th 10:30 PM)**