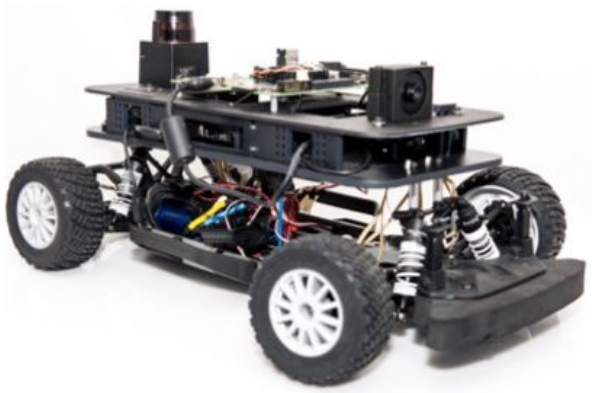
ESE 680 - F1/10: Autonomous Racing



Build

Build an autonomous race car from scratch! Starting from a stripped-down 1/10th-scale mechanical chassis capable of 40mph, add the sensors, the compute boards, and the electronic components needed for autonomous driving. In the process you will learn about different sensing modalities, their pros and cons, PID tuning, and principles of embedded computing.

Program

Now that you have the hardware ready, you will learn how to program an autonomous robot to turn the raw sensory data into actionable information, and ultimately into navigation and control commands for your car. Your code will leverage ROS capabilities, the widely popular OS for robotic applications.

Race!

Your team will compete against others for completing the track in the fastest time. In the second half of the class your team will independently develop racing algorithms that will give you the edge over the others. We will guide you in learning navigation algorithms such as pure pursuit and probabilistic roadmaps.

Instructors and TAs

Houssam Abbas

Matthew O'Kelly
Jack Harkins
Chris Kao
Rahul Mangharam

Logistics

Class will meet 5 hrs each week: 2.5hrs of lectures, and 2.5 hrs of lab. The TAs and instructors will be available during the lab time. You can also access the lab space at other times. Coordinate with lab supervisors.

Schedule

Note that the lecture topics are subject to change, based on class progress and class interest.

Date	Topic	Practice laps
8/28	Selection of teams. Introduction to the course and embedded control systems.	Getting a heartbeat: Assembling the car, wifi connection, and manual control
9/4	ROS basics	1. ROS tutorials 2. Monitor node creation
9/10	F1/10 Hardware	Setup the simulator, compute the largest gap, and the center of the gap (using true global coordinates)
9/11	--	
9/17	LiDAR	Setting up the LiDAR, visualizing in Rviz, and gap-finding on real lidar data from a saved bag
9/18	Pose representations and coordinate transformations	
9/24	Reference tracking: basics of Laplace domain analysis; PID	1) Vesc tuning 2) Wall-following in simulation
9/25	Reference tracking: PID; VESC tuning	
10/1	Electronic Speed Control [instructor: Jack Harkins]	Wall following and hard-coded turns on the car

10/2	No lecture - students work on their lab [H. Abbas and M. O'Kelly on travel]	
10/8 Drop Period Ends	Localization: scan matching I	Implement scan matching in simulation and on car
10/9	Localization: scan matching II (optimization)	
10/15	Mapping I: Describe the SLAM problem	Scan matching cont'd
10/16	Mapping II: SLAM continued	
10/22	Mapping III: describe the state-of-the-art packages [Instructor: Matt O'Kelly] (orbSlam, cartographer, elastic fusion, Hector SLAM)	Practice and run the SLAM and localization packages
10/23	Pure Pursuit	
10/29	How to manually generate and save trajectories [Instructor: Matt O'Kelly]	Pure pursuit and prepare for race
10/30	Path planning: A*	
11/5	Path planning: state-of-the-art or whole lecture on RRT?MPC?	Readings Mid-semester Report
11/6 Last day to withdraw: Nov. 9	Mid-semester race	
11/12	Moral decision-making I	No more assigned labs after this point. The teams start working on their racing algorithms, until the end.
11/13	Moral decision-making II	
11/19	Neural networks as pilots [Instructor: Chris Kao]	
11/20	Reinforcement Learning	

	[Instructor: Matt O'Kelly]	
11/26	Guest lecture: Stereo vision	
11/27	Guest lecture by CJ Taylor (UPenn): High performance embedded systems for high speed drones.	
12/3	Today's research Autonomous Vehicles [Instructor: Matt O'Kelly]	
12/4	End-of-semester race: Practice laps	
12/10	End-of-semester Race	

Course grading and evaluation

[50%] Labs
[30%] Competition performance (mid-semester and end-of-semester) and public communication
[5%] Competition document: a 10-page document summarizing your approach to the competition (software architecture, algorithms, hardware, tests, etc), examples of performance results, etc.
[5%] Peer review: an anonymous evaluation of your work performed by your teammates.
[10%] Participation in class and TA evaluation

Pre-requisites

The most important technical pre-requisite is good programming skills in C++ and Python. You will be coding or reading code in both languages. Python is easy to learn if you don't already know it, but you will have to do that on your own time.
You will also need knowledge of frequency transform concepts (e.g., Fourier or Laplace), basic matrix algebra and differential equations.

The F1/10 Reference Manual (August 2018 version) is available on www.f1tenth.org . It has lots of good information. Use it.