AuE 835
Automotive Electronics Integration

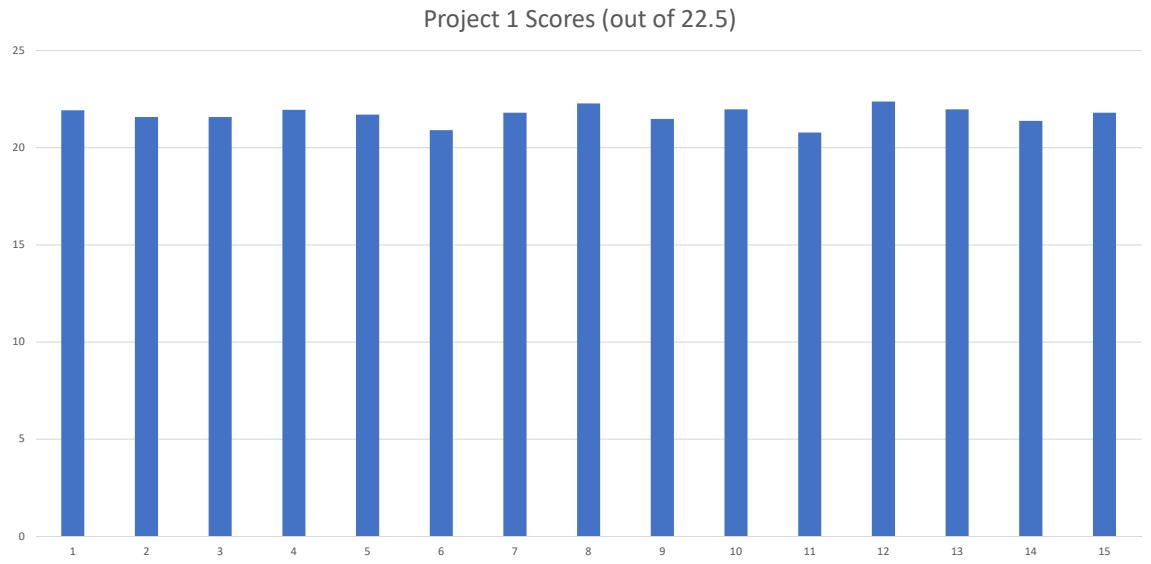
PROJECT 2: ADAPTIVE CRUISE CONTROL AND AUTONOMOUS LANE KEEPING

Project Schedule

Project 1

- **Nov 2 – Arduino and programming**
- **Nov 4 – Ultrasonic sensing & Project 1 Announcement**
- **Nov 9 – Signal processing review and hands-on teaching**
- **Nov 11 - Project 1 debugging, Q&As, Test details**
- **Nov 16 - Project 1 Presentation and Test**

Project Schedule



Project Schedule

- Nov 2 – Arduino and programming
- Nov 4 – Ultrasonic sensing & Project 1 Announcement
- Nov 9 – Signal processing review and hands-on teaching
- Nov 11 - Project 1 debugging, Q&As, Test details
- Nov 16 - Project 1 Presentation and Test

- **Nov 18 – Vehicle Control & Project 2 Announcement**
- **Nov 23 – Control review and hands-on teaching**
- **Nov 30 – Project 2 debugging and Q&As, Test details**
- **Dec 2 – Project 2 Presentation and Test**

- **Dec 10 - Project reports**

Project Name: Run Down The Hill

Project Name: Run Down The Hill

- Tasks

  Drive an RC vehicle to run down the ramp at CU-ICAR or a self-built indoor test track (depending on weather). Design and implement an embedded controller based on ultrasonic sensors to run down the hill as fast as you can while achieving the following functions:

  1. ***Adaptive Cruise Control***
     - Keep the vehicle 30 cm away from obstacles ahead
     - If obstacles ahead are stationary, vehicle should stop at 30 cm away from the obstacles
  2. ***Autonomous Lane Keeping***
     - Keep the vehicle along the center of a defined lane

  Note: You are free to use any learnt sensing signal processing approaches and control approaches in this project.

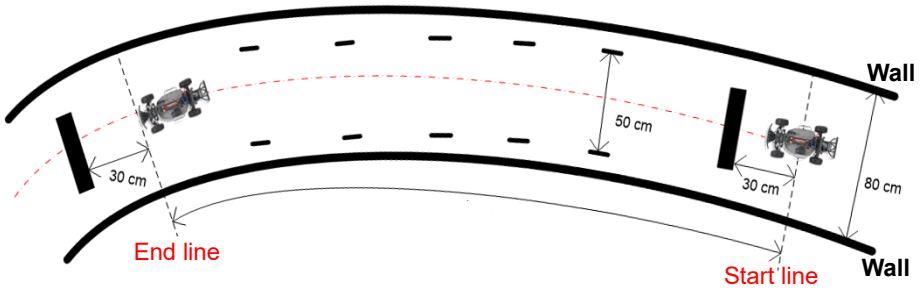
Project 2 Rules on Controls

- You can use any model-based or non-model based controllers
- You can combine the above controllers with simple logic (e.g., stop the vehicle when oscillation is small enough)
- But you cannot use purely simple logic to control the vehicle

Project Name: Run Down The Hill

❖ Detailed Test Tasks

1. You need to run down the hill as fast as you can without crashes
2. At the beginning, a board is put 30 cm away from start line and the vehicle crosses the line. Then, the vehicle should move backward to stop at the start line.
3. After the board is removed, the vehicle should start to run down the ramp along the center of the defined lane (do not hit boundaries and pass through 5 check points)
4. A board is placed 30 away from the end line and the vehicle should stop at the end line



Project 2 Hours

- Project hours are as follows:
  - Monday and Wednesday 4-5 PM
  - Sunday (11/21): 1-5 PM
  - Sunday (11/28): 1-5 PM
  - Wednesday (12/1): 10AM-5 PM
  - If you would like to meet in other time, please send an email to the instructor or TAs to make an appointment.
- Office hours will be mainly in Zoom: https://clemson.zoom.us/my/haotiansu
- However, in-person assistance will be available too when really needed

Project 2 Test Details

Project 2 Test Details

1. Each group must upload your codes to Canvas before test class
2. Download your codes to Arduino and use your Arduino-sensor-vehicle setup to perform the task
3. If your vehicle cannot run well, you will have a free run after all groups are finished
4. For any extra run after the free run, you will lose 10% of the project 2 test score.
5. The test order will be from group 15 to group 1

Work After Test

- Due to class time limit, each group need to record a 5-min presentation video (including problem, approach, results and conclusions) and upload it to Google Drive by December 6 (Monday) Noon: https://drive.google.com/drive/folders/1tah3MOHooiouDTh330QoYr55i69-w4cV?usp=sharing
- Each student has to submit peer-review scores to Canvas to evaluate all your other teammates (score: 1-10) based on their contributions after the test (confidential) by December 6 (Monday) Noon.

Project 2: Grading Details

❖ Grading of Presentation (2.5 pts)

- 50% – Slides
- 40% – Presentation
- 10% – Time

❖ Grading of Project 2 Test (20 pts)

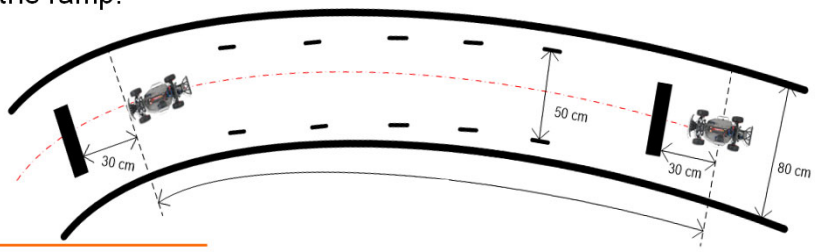
- 60% - Test results, including:
  - 20% - Stop the vehicle 30 cm from the start line and end line with minimum vibrations (you have to stop the vehicle!)
  - 20% - Running down time: from start line to the end line
  - 20% - No boundary hits and passing through 5 check points on the ramp
- 40% - Submitted codes

- -10% - Penalty
  - Up to 10% penalty by peer evaluations

Project 1.2 Test Details

Important Notes

1. Stop the vehicle at the start line 30 cm away from the board [You have to stop the vehicle!]
2. Stop the vehicle at the end line 30 cm away from the board [You have to stop the vehicle!]
3. Run down the hill with appropriately fast speed. Total running down time counts from start line to end line.
4. Do not hit the boundaries and pass through 5 check points on the ramp.

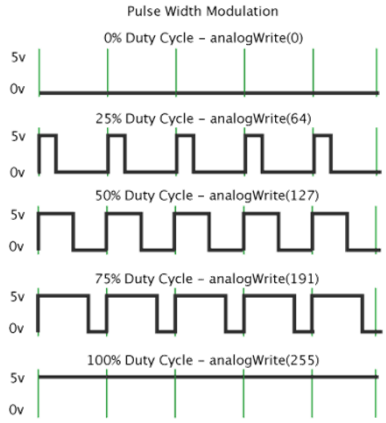


Project 2 Notes

- Recommend PID control for autonomous lane keeping task
- Tune the maximal speed to tradeoff between the time cost and control performance in adaptive cruise control and autonomous lane keeping
- Perform the calibration process when the open-loop throttle/speed control of RC vehicle does not work as expected
- Use writeMicrosecond() function when the vehicle throttle/speed control is not smooth
- You can also create Pulse width modulation (PWM) for throttle/speed if necessary

Project 2 Notes

Pulse Width Modulation (PWM) is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off.



PWM Throttle Control Example:

```
……..
setVehicle(steering, throttle);
delay(75);
setVehicle(steering, 90 or 1500);
delay(25);
……..

OR

……..
cycle = 100;
PWM_On = cycle*throttle/max_throttle;
setVehicle(steering, max_throttle);
delay(PWM_On);
setVehicle(steering, 90 or 1500);
delay(cycle-PWM_On);
……..
```

Project Report

- **Each student** has to write a report to introduce your achievements and findings in project 2 in addition to project 1 report. (**Record data now!**)
- The report is due on **December 10 (Friday) at 11:59 PM**
- Recommended template for project 2 is uploaded to Canvas. The format is similar to a technical paper as following:
  - Problem Statement
  - Technical Approach
  - Hardware and Software Implementation
  - Experimental Results
  - Discussions and Conclusions