EECS461 Final Project Report

Fan Zhang & Bo Zhang[1]

Introduction

In this project, we are assigned tasks with using Simulink to design and simulate a Vehicle Model (milestone I), an Adaptive Cruise Control (ACC) system (milestone II), and an Automatic Steering Controller with PID (milestone III), and then run on a simulated road.

In milestone I, we implemented a simple Vehicle Model to achieve manual control, which is activated when ACC is disabled. In this model, potentiometer serves as the “pedal”, and haptic wheel serves as steering wheel. Additionally, self-aligning torque is acted to drive the steering angle back to zero when we turning steering wheel.

In milestone II, we designed Adaptive Cruise Control system. When ACC is turned on, there is a “speed control” mode that maintains the vehicle at the desired velocity unless another vehicle is too close, in which case a “position control” mode is implemented that keeps the vehicle a fixed distance from the car immediately in front. The communication between each station is through CAN controllers.

In milestone III, we implemented the automatic steering controller. When enabled, the controller should steer the car down the full length of the road by actuating the haptic wheel.

Under the Automatic Steering Controller, Velocity Controller and Position Controller, the car can manage its speed and position automatically. Furthermore, we can change between ACC mode and Manual mode easily anytime by pushing up or down a DIP switch on the microprocessor board.

Pick Lead Logic

We used an s-function block to implement the pick lead logic. Lead_speed and Lead-position is initialized as our car’s. The FOR loop is acted in this logic, to pick the position ahead of our car and closer to our car than the current lead-position, and replace the current lead_position and Lead_speed with the selected ones. Finally, after comparing all the 6 other cars, we can find the lead car’s s position and speed. The pseudo-code for this logic is listed in Table 1.

[1] fanum@umich.edu, gcbo@umich.edu

Table 1 Pseudocode of Pick Lead Logic

```
Pseudocode of Pick Lead Logic
Input: Array s of 7 float s direction position
       Array us of 7 float speed in s direction
Output: the s direction position and speed of the lead car
        Lead_speed = us[0]
        Lead_position = s[0]
        For i= 1 to 6
            If s[i] > s[0] and s[i]< lead_position
                Lead_position = s[i];
              Lead_speed = us[i];
            End if
        End for
```

Note: s[0] and us[0] are the s direction position and speed of our car respectively

Automatic Steering Controllers

To implement Automatic Steering Controllers, We use both PD and PID controllers in sequential to maintain the car can always run along the center of the road. Every time we set up a set of parameters for PD and PID, we observe the phenomenon in simulated car, and we adjust the five parameters accordingly. For example, if it oscillates so much, we will increase the differentiator parameter a bit according to oscillation intensity. If it does not turn enough when deviates from the central line of the road, we would increase the gain of difference control parameter. Our selected parameters are as follows:

Kd_PD	Kp_PD	Kp_PID	Kd_PID	Ki_PID
0.03	0.3	400	130	0.0001

Encountered Problems

One main problem we encountered was the self-aligning torque implementation in the manual control. We could only feel the torque when we turned the haptic wheel to the left and if we turned a big degree, then it would rotates. We spent about 3 to 4 hours in total to debugging this and finally we changed the position of the PWM configuration, extract the self-aligning torque block to the higher level, and made it work.

The other problem we encountered was implement CAN. When we added all the other 6 cars in the CAN, our car was frozen at the starting point of the simulated road. Then we restarted under the shoulder of milestone 1, only left one car in CAN, and finally it worked. Both the velocity control and position control worked well. We discussed this issue with GSI. Neither them nor us could figure out what caused the frozen when had other 6 cars in the CAN.

Work-load Distribution

We spent about 15 hours in total in this project. Fan and Bo are almost evenly distributed to the project.