**Intro to Computational Intelligence: Project 2 Report**

**Fuzzy Cruise Control**

1. Group Responsibilities
   * James (Alex) Hurt
2. Topic Brainstorming
3. Initial Abstract Report
4. Explaining “What is Fuzzy Logic?” in YouTube Video
5. Analysis of Results in Final Report
   * Will Morrison
6. YouTube Video Script Writing
7. YouTube Video Analysis
8. Explaining “Why Fuzzy Logic is Useful” in YouTube Video.
   * Holt Skinner
9. YouTube Video cinematography, editing, and production.
10. Explaining “How Fuzzy Logic Applies to Cruise Control” in YouTube Video
11. Documentation Editing & proofreading
    * Yilong Zhang
12. Cruise Control Fuzzy Systems Research
13. Creation of Prototype Model
14. Final Report Draft
15. Creation of Presentation Poster
16. Background

Cruise control systems have become a common feature in automobiles since the 1970’s when the United States was under an oil shortage. Instead of having the driver frequently checking the speedometer and adjusting pressure on the gas pedal or the brake, the cruise control system adjusts the speed of the car by maintaining a constant speed set by the driver. Therefore, cruise control system can help reduce driver’s fatigue in long road trips. This project models the control system behind modern cruise control using fuzzy logic. The project was implemented with a proportional–integral (PI) + fuzzy control cruise controller in Matlab Simulink.

1. Modeling of the cruise controller

Figure 1 shows an overview of the designed model of the cruise controller with the PI + fuzzy control methods. The input of this model is the target speed of the vehicle, which is defined by the user. The output of this model is the actual speed of the vehicle, as defined by the cruise controller.

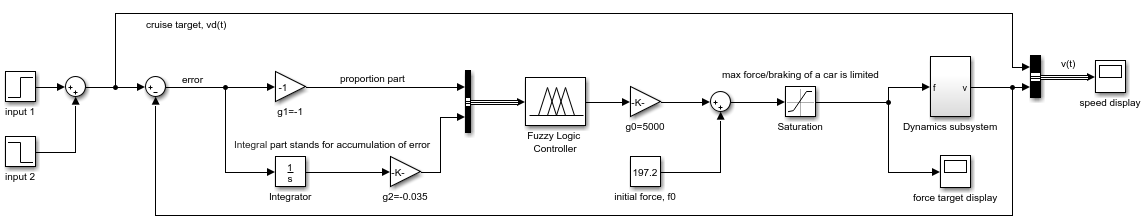


Figure 1 Overview of the model in Simulink

* 1. PI control

The PI control takes the error between the target speed and current speed as input, calculates the accumulation of this error over time, and sends both the current error and error accumulation to the fuzzy logic controller. There are different weights for the proportional path and the integral path, which values are the result of finetuning. In our final model, the weights are -1 and -0.6, respectively.

* 1. Fuzzy logic controller

The input to the fuzzy logic controller is exactly what comes out from the PI control part. In the fuzzy logic controller, there are defined membership functions for each variable – error of the speed, cumulation of the error of speed, and output force to the accelerator and brake to further control the vehicle. A positive number equates to acceleration, and a negative number equates to braking. The larger the absolute value of the number, the greater the force. Figure 2 shows an overview of the fuzzy logic controller in a block diagram. The model uses Zadeh’s definition for AND, OR, and Implication operators, and centroid defuzzification method to convert the output fuzzy set to a single numeric value.

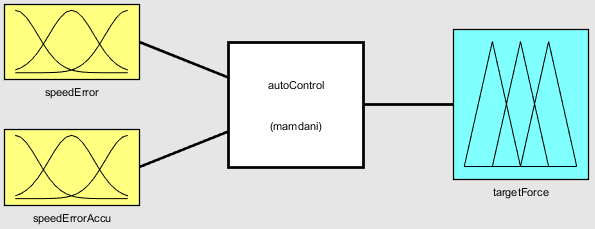


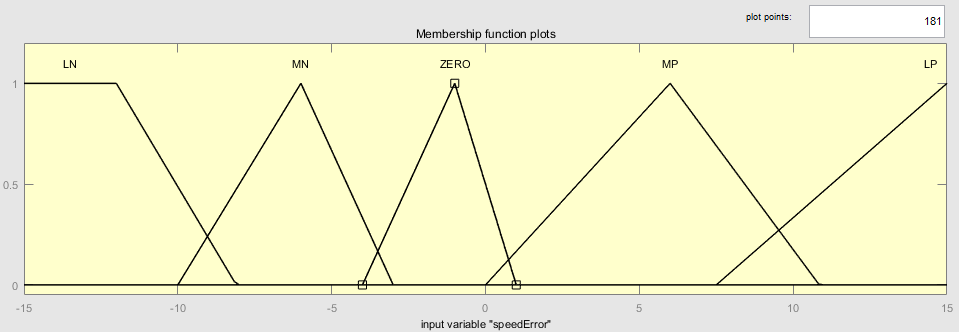
Figure 2 Interface of the fuzzy logic controller

Figure 3 defines the membership functions for each variable. In (a) and (b), representations of the abbreviation are as follows:

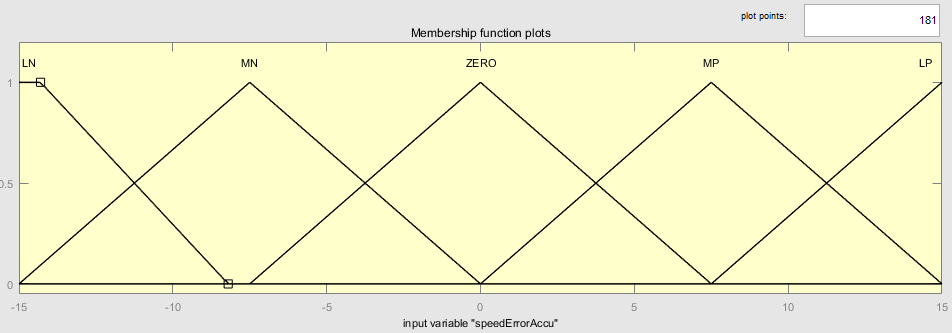
**LN**=Large Negative **MN**=Medium Negative **Zero**=Zero

**MP**=Medium Positive **LP**=Large Positive

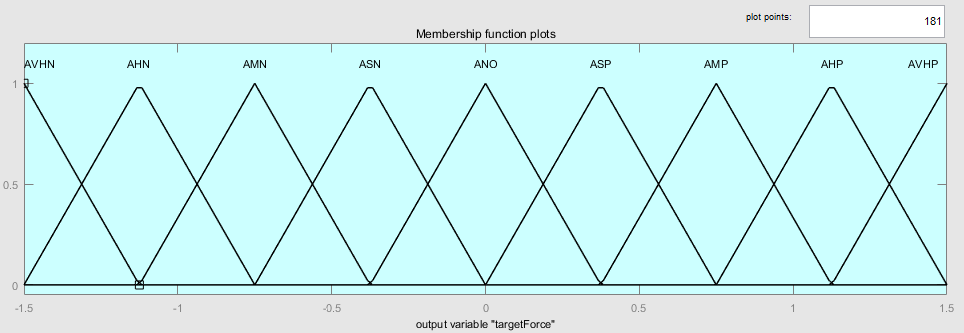
A negative number means the current speed is lower than the target speed, and a positive number means the current speed is higher than the target speed. For the sake of safety, the models set a peak of **ZERO** to be a little lower than 0. If current speed is lower than the target, drivers may complain about the poor cruise controlling system. So, **LN** gets its peak at about -12 mph rather than mirroring **LP**, at 15 mph.



1. membership definition for error in speed



1. membership definition for error cumulation in speed



1. membership definition for force output to accelerator/ brake

Figure 3 Definitions of the membership functions for each variable

In (c), meanings of the abbreviations go as follows:

**AVHN** = Add Very High Negative **AHN** = Add High Negative

**AMN** = Add Medium Negative **ASN** = Add Small Negative

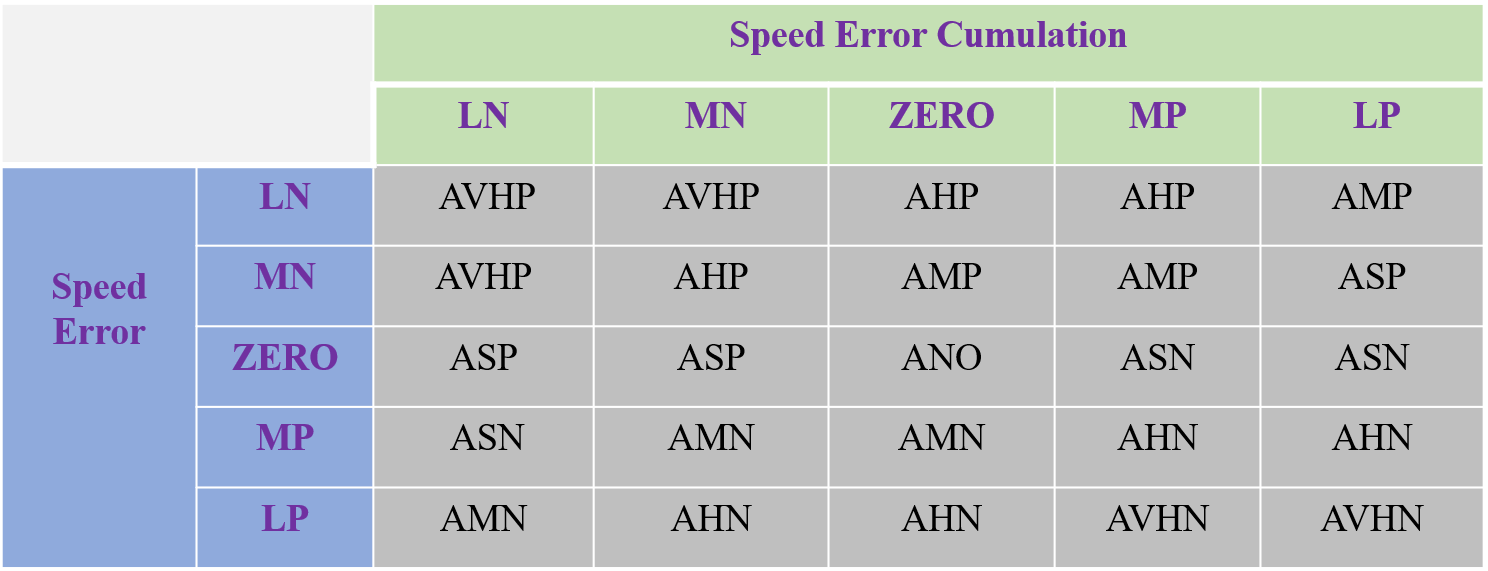
**ANO** = Add No **ASP** = Add Small Positive

**AMP** = Add Medium Positive **AHP** = Add High Positive

**AVHP** = Add Very High Positive

We define 9 membership functions for force rather than 5, since we expect to make the output force, such that the result speed will be as near the target speed as possible. Note that in reality, force generated from the engine or brake cannot be infinitely large, and in our experiments, we have set its limit value to .

Table 1 Fuzzy rule definition



Fuzzy rules are shown in Table 1. We take crisp values as input and follow the methods in Section 7.4 of the textbook to calculate output of the fuzzy logic controller.

* 1. The dynamics subsystem

The dynamics subsystem takes the force exerted on the accelerator and brake as input, and outputs the current speed of the vehicle. We say the input force is a target force rather than the real force, since it takes a moment of time for the vehicle to respond. Governing equations in the dynamics subsystem is rather simple.

(1)

(2)

where,

, is the aerodynamic drag

– vehicle speed at time

, is a constant frictional force

is the driving/braking force at time

is the driving/braking force target, output from the fuzzy controller

, is the time slot given to the vehicle to adjust its force

Eq. (1), in which will be a variable if the vehicle goes up or down a slope, comes from Newton's second law of motion. Eq. (2) represents the change of force over time, meaning response time of the vehicle. Realization of the dynamics subsystem is shown in Figure 4, in which the two equations are realized.

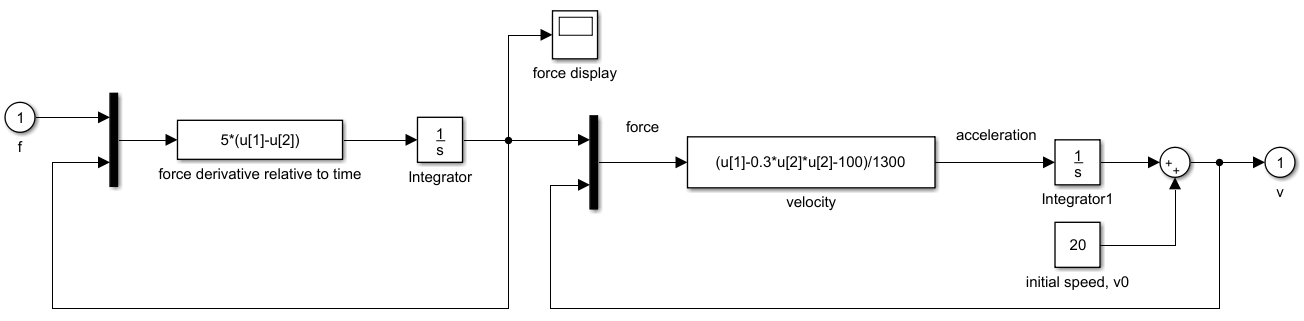


Figure 4 Realization of the dynamics subsystem in Simulink

1. Experiments

For the initial state of the experiments, suppose the vehicle is running at 20 mph, which is also the speed target. During the experiment, we monitored the real-time speed, and the force output of the accelerator/ brake.

**Scenario 1**: Set target speed with step functions. That means, the target speed jumps to a different value in an instant. In our experiment, at the target jumps up to 70 mph, and at the target jumps down to 40 mph.

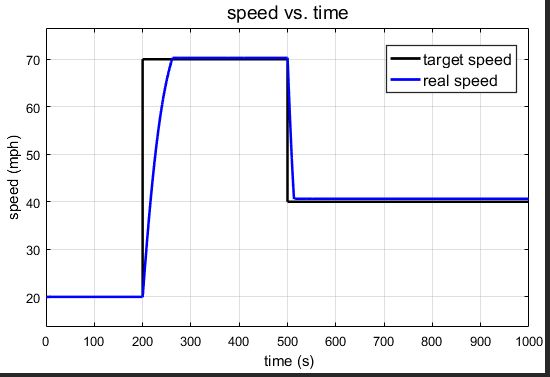
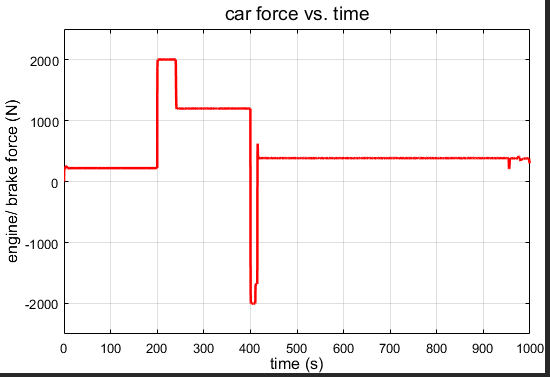
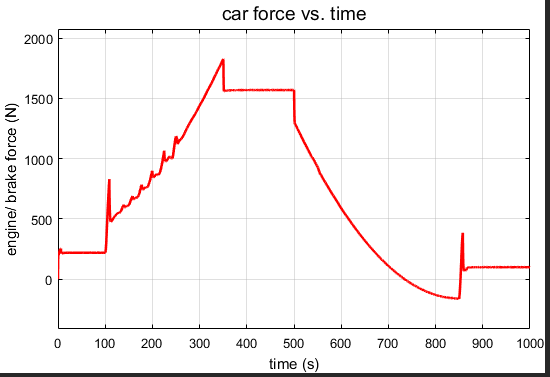
 

Figure 5 speed change and force change in scenario 1

From Figure 5 we notice that it takes the vehicle about 60s to accelerate up to 70 mph, and about 10s to slow down to 40 mph. (It seems our vehicle’s horsepower is quite low since we have limited the engine output to 2000 *N*) No overshoot or undershoot in speed occurs, which is good for the rider. Notice there is a sudden change in the acceleration and brake forces of the vehicle, which tells us that our cruise controller is quite sensitive to change in the target speed.

**Scenario 2**: Linearly change the target speed, which is very common for some vehicles. In our experiment, from the target gradually goes up to 70 mph, and from the target gradually goes down to 0.



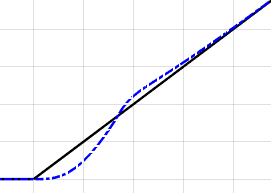
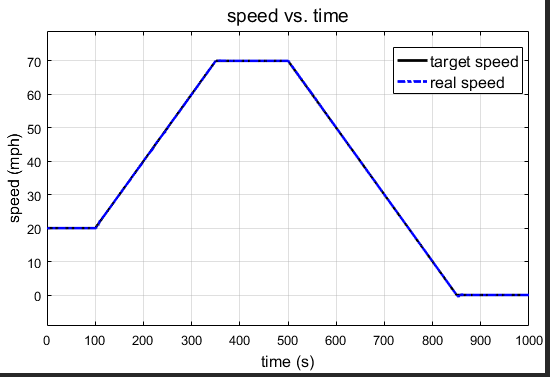


Figure 6 speed change and force change in scenario 2

From Figure 6 we notice that the real-time speed changes conformably with the target speed, which should be satisfactory from the driver’s view. In the process of synchronization, the real-time speed is adjusting dynamically by changing the force output. We can see that there are several spikes in the force curve, which we believe that there should be a change in the dominant membership of force along with it.

**Scenario 3**: The vehicle goes up and down slopes while keeping the target speed, which is common in hilly areas. In our experiment, at the frictional force steps up from initial value of 100 *N* to 300 *N*, and then from to , it linearly goes down back to 100 *N*. From Figure 7 we notice very little change in real-time speed, which gives the driver a comfortable driving experience. Since at the frictional force has a sudden change, the engine responds accordingly, and it results in a spike in the graph. Overall, the model performs well.

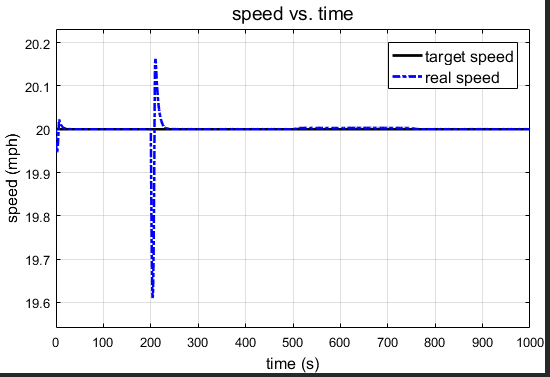
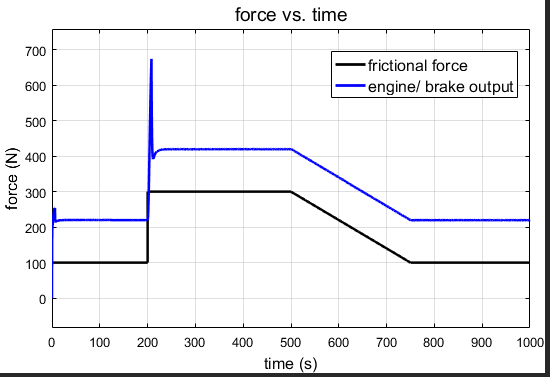
 

Figure 7 speed change and force change in scenario 3

1. Analysis

The PI + fuzzy cruise controller, has been to have properly simulate several different scenarios in driving. As expected, it was found that gradual changes in acceleration and deceleration lead to a smoother driving experience, with smoothness determined as a function of force over time. The application of fuzzy rules provides a smoother driving experience in contrast with accelerating and braking through discrete amounts of force and speed because of the continuous nature of the fuzzy membership functions.

**Scenario 1**

In the speed graph, we see an expected chart. It makes sense for there to be a delay between when the target speed is set to 70mph, and when the vehicle reaches 70mph. A similar event occurs in a physical car as well. When jumping from 20 to 70mph, there is a delay as the car accelerates to desired speed. The same logic applies to the jump down to 40mph.

The right graph is also an expected result because there would need to be a large change in direction and magnitude of force to attempt to slow the vehicle from 70mph to 40mph quickly. The design of the scenario attempts to instantaneously change speeds, which means a very large amount of force in an attempt to minimize time to get to the target speed.

**Scenario 2**

The left graph makes sense, because rather than attempting to jump instantaneously, the jump happens over a greater time period and is defined to occur linearly. The small differences between real speed and target speed tell us that an exact cruise control system is very difficult to achieve, but one very close to perfect is possible.

The force graph is slightly surprising during the linear increase. One may expect the force curve during a linear increase to be more smooth, due to the physical feeling of linearly speeding up in a car. In addition, the spike at t=850 is surprising, as the car has stopped moving, so a large increase in force is not what we expected to see.

**Scenario 3**

The results of this scenario are fairly consistent with our previous thoughts on how this scenario would play out. The target and real speed curves are practically identical and the frictional force has a strong correlation with the engine/brake output. These results are comparable to our expectations prior to running this scenario.

1. YouTube Video

A YouTube video describing fuzzy logic and how it is used in a cruise controller was developed for the project. The video was developed with the intention of describing fuzzy cruise control research to someone who was unfamiliar with computational intelligence. Developing the video helped the members of the group better understand fuzzy logic and how it is used in cruise controllers. The video also allowed the group to express itself in another way which was not programmatic, written, or poster based. The members of the group acknowledge a lack of fuzzy logic videos on YouTube and hope the accessibility of the video helps people familiarize themselves with a new field of computer science. The video can be found at <https://www.youtube.com/watch?v=JKr2ZYrP5KE>