GENG8030 Computational Methods and Modeling for Engineering Applications Course Project – Fall 2022

Adaptive Cruise Control with Cloud Connectivity

Objective

Use **MATLAB** to develop and implement an adaptive cruise control system for vehicles that automatically adjusts the speed to maintain a safe distance from vehicles ahead. The speed information has to be uploaded on the cloud to support connected vehicles and intelligent transport.

Hardware

Arduino uno x 1, button x 5, distance sensor x 1, 4 digit 7-segment anode display x 1

Description

Implement an adaptive cruise control system using the Arduino UNO board. The system is controlled by five buttons of (1) Set_speed, (2) Adaptive_speed, (3) Cancel, (4) Increase_speed, and (5) Decrease_speed. After power on, the display has to show the initial speed of 0.

1. When the Increase_speed button is pressed, the speed increases, and when the Decrease_speed button is pressed the speed decreases however without pressing the Set_speed button the speed will not remain constant and it changes slowly over time.
2. When the Set_speed button is pressed the system enters the cruise control mode where the speed is held constant. In this mode, the Increase_speed button and the Decrease_speed button are still functional and can be used to change the Set_speed. If the Cancel button is pressed, the system quits the cruise control mode where the speed decreases slowly.
3. If the Adaptive_speed button is pressed, the speed is set and held constant until a vehicle shows up at the front or an object is detected where the speed automatically decreases. When the road becomes clear, the speed increases to reach the set speed again. In the adaptive cruise control mode, the display has to blink to differentiate this mode from the cruise mode. In this mode, the Increase_speed button and the Decrease_speed button do not function but the Cancel button can still be used to quit the adaptive cruise mode. If the Cancel button is pressed, the display stops blinking, and the vehicle speed begins to slow down.
4. The implemented system has to be able to communicate with the ThingSpeak platform to store the speed data on the cloud. ThingSpeak is an Internet of Things (IoT) application that supports MATLAB to analyze and visualize data

Test procedure

The implemented project will be fully tested to ensure that the project objectives are met. A sample test may include the following steps.

1. When the system is initialized, the display system shows: 0
2. Press the Increase_speed button to increase the speed to over 40 and then release the button, the speed has to decrease gradually.
3. Press the Decrease_speed button to decrease the speed until the speed reaches 0.
4. Press the Increase_speed button to increase the speed to about 20, press the Set_speed button, and then press the Increase_speed button and the Decrease_speed button to change the speed. Press the Cancel button to quit the cruise control mode.
5. Press the Increase_speed button to increase the speed to about 30 and then press the Set_speed button to lock the speed. Press the Adaptive_speed button; use a model car to trigger the distance sensor and watch that the speed decreases. Remove the model car and watch that the speed increases and locks again at 30.