ECE 303-Term Project

May 1, 2020

Motivation

In this half-term project, you will be developing an electric vehicle test-bed. This will utilize just about everything you have learned throughout the term. If you think about how to structure the project before just writing code and building circuits, it will make your life easier.

This project has the following areas:

1. Collision avoidance testing
2. Dashboard development
3. Security and Remote Control
4. Sensor implementation

Project 1-Collision Avoidance

Motivation

Collision avoidance is used in modern automobiles. It uses sensors (light, sound, etc.) to determine if there is something in front of the car. If the car needs to slow down, signals will control servos to reduce fuel into the engine, and autonomously control the brakes.

In this part of the project, you will be simulating a collision avoidance system using the ultrasonic sensor and servo motor in your kit. The closer an object gets to the ultrasonic sensor, the servo rotates to a different position.

Suggested Reading

- Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino: `https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/`

Part 1-Ultrasonic Sensor Installation and Verification

Install and verify you can properly use the ultrasonic sensor. Take and place an object directly in front of the sensor. from 1 to 10 cm. Make sure the sensor is not on the ground, as you don't want ground bounce.

Part 2-Map the Angle of Acceptance

Place an flat object 30 cm directly in front of the sensor. Using a protractor phone app of your choosing, position the object from 90 degrees (directly in front of the sensor). Take a reading. Move the object 5 degrees to the left or the right of the sensor. Take a reading. Continue to do this until you no longer get a reading. This is a rough estimate of the angular acceptance.

Part 3-DC Motor Installation and Verification

On the same board as your ultrasonic sensor, connect the DC motor (with fan blades). Make sure you're using the H-bridge and the power supply in your kit.

Set up your motor so it only moves in one direction, and can be speed controlled.

Part 4-Servo Motor Installation and Verification

Install and verify operation of the servo motor.

Part 5-LED Installation and Verification

Install three different LEDs on the same board as the DC motor and ultrasonic sensor. Ensure you have resistors in series with the LEDs (and make sure they are the right value).

Part 6-Combining It All Together

Now that you have all of the parts individually working, you need to integrate them into a system. The DC motor will simulate the engine, the servo motor will act as the brake control, and the LEDs as display signs.

Set up your system such that the following are a function of distance:

- Servo motor: When the object is past a certain distance from the sensor, the servo is positioned at one extreme. As the object moves closer, the servo moves towards the other direction. When the object is at a certain distance, the servo is positioned at the opposite end than when it is far away.
- DC motor: When the object is past a certain distance from the sensor, the DC motor is at full speed. As the object moves closer, the DC motor reduces speed. At a certain distance (same distance as the servo motor threshold), the motor stops.

- LEDs: Configure the LEDs such that no lights will be on when the distance is far away, with more LEDs turning on successfully at given distances. NOTE: you will be removing the LEDs in the next part of the project. You are using these at a temporary place holder for the graphical user interface.

Part 2-Graphical User Interface

Motivation and Direction

When implementing a testbed, it is often important to display the data live-time to see system performance. For this part of your project, you are to use a graphical user interface. You can use either MATLAB or Python to produce your interface. On your GUI, you should have the following:

- Distance to object (in meters)
- Distance LEDs based on distance (Red, Yellow, and Green)-this will replace the LEDs from the previous part
- Headlights off/dim/bright - this will be used when you implement the IR remote
- Motor "Speed", in percent - this will be used when you implement the IR remote
- Low coolant alarm - this will be used when you implement the water level sensor
- Temperature display and alarm - this will be used when you implement the temperature detector

Part 3-Security and Remote Control

In this part of the project, you are to implement a security measure to prevent unauthorized personnel from starting the system. Additionally, using remote control of the system is an attractive feature.

Security Measure

Your system needs to have a security measure. It should only allow proper personnel to have access to operating the system. Use a buzzer to use different sounds to inform the user whether or not they are authorized or unauthorized.

Remote Control

You need to have the system respond to the IR remote, controlling the following:

- Motor speed. The remote should change speed by increasing/speed of the engine motor in 10% increments.

- Lights-Use two LEDs as headlight. The remote should be able to set the LEDs to off/dim/high. Note you cannot use one button to control all three states.

Note that both motor speed and headlight status should be sent to the GUI.

Displays

Using the LCD screen, you need to display the current motor speed, coolant level, and battery temperature.

Video

As the grand finale, you are to produce a video. The report should discuss your project, including:

- Project description
- Displaying the test bed
- Functionality of the test bed
- Current budget to validate you are not drawing too much current
- Your method/data format to transfer all data to the computer for display.