A logo for a company

Description automatically generated

Final Project Report

**Adaptive Cruise Control System**

**Course Name: Computational Methods & Modeling**

**for Engineering Applications**

**Course ID: GENG 8030**

**Date of Submission: 28th June,2024**

**Instructor:**

**Prof. Dr. Rashid Rashidzadeh**

**Project Group No: 5**

|  |  |
| --- | --- |
| **Name** | **Student ID** |
| **Jas Prajapati** | **110147267** |
| **Nirav Polara** | **110153432** |
| **Jayeshkumar Thummar** | **110155671** |

**Table of Contents**

[**1) Project Objective 2**](#_Toc173177468)

[**2) Project Description 2**](#_Toc173177469)

[**3) Program Logic Design Flow Chart 3**](#_Toc173177470)

[**4) Function Explanation 4**](#_Toc173177471)

[**5) Components Details 6**](#_Toc173177472)

[**5.1) Connection Of Components 6**](#_Toc173177473)

[**6) Roles And Responsibilities 7**](#_Toc173177474)

[**7) Conclusion 8**](#_Toc173177475)

[**8) References 8**](#_Toc173177476)

**List of Figures**

[**Figure 1: Circuit Diagram 4**](#_Toc173061780)

# **1) Project Objective**

Our project aims to revolutionize automotive safety by integrating advanced features that reduce traffic accidents and enhance overall human safety. By addressing unsafe driving behaviors such as tailgating and sudden braking, we strive to make the roads safer for everyone. According to a World Health Organization study, nearly 1.35 million people die each year from traffic accidents, with road injuries being the leading cause of death for those aged 5 to 29 [1]. This statistic highlights the urgent need to improve vehicle safety to reduce the toll of traffic-related deaths and injuries.

One promising solution is the development of Adaptive Cruise Control (ACC) systems. ACC technology has made significant strides in recent years, automatically regulating a vehicle’s speed based on surrounding traffic [2]. These systems use distance-sensing technologies such as radar, lidar, or video cameras to measure the distance and relative speed between vehicles [3]. By allowing the vehicle to adjust its speed automatically, ACC not only enhances driving convenience but also addresses unsafe driving behaviors, making ACC a crucial milestone in the automotive industry's journey toward autonomous vehicles [2].

# **2) Project Description**

This project focuses on designing and simulating an Adaptive Cruise Control (ACC) system. We’ll use an Arduino Uno starter kit to build the necessary components and MATLAB to program and simulate the ACC system.

ACC helps drivers maintain a set speed, making driving more relaxed by eliminating the need for constant speed adjustments. Traditional cruise control systems don’t slow down when there’s a slower vehicle ahead, so drivers have to manually adjust their speed. Our research aims to improve this by adding adaptive cruise control, which automatically adjusts the car’s speed based on the distance to the vehicle in front, slowing down or speeding up as needed. The proposed system combines standard cruise control with a distance sensor to create this adaptive behavior [4]. It includes five buttons for setting speed, engaging adaptive speed control, increasing/decreasing speed, and canceling. This system operates in two modes: standard cruise control and adaptive cruise control [5].

In normal mode, the set speed button keeps the car at the selected speed, and the car accelerates or brakes as usual. You can turn off cruise control with the cancel button, which slows the car down gradually [4]. In adaptive mode, activated by the adaptive speed button, the car keeps the set speed unless it detects a slower vehicle ahead. When it does, it slows down to maintain a safe distance [4,6].

Once the road is clear, the car accelerates back to the set speed. Pressing the cancel button exits adaptive mode and starts slowing the car down gradually. To differentiate between the two modes, the system has a pulsing indicator [5]. The system uses an ultrasonic distance sensor and five buttons, which act as analog inputs. These inputs are processed by the programmed instructions in the Arduino Uno kit’s controller and displayed on an LCD screen [5,6].

# **3) Program Logic Design Flow Chart**

Flowcharts are excellent for outlining and documenting simple processes and programs. They make it easy to visualize each step. With the help of the flowchart above, we can understand the flow of the project and efficiently create the program in MATLAB.

A diagram of a flowchart

Description automatically generated

The circuit diagram details the hardware connections needed to bring the flowchart’s logic to life. Together, they ensure a comprehensive understanding of both the software and hardware aspects of the project.

A circuit board with wires and a screen

Description automatically generated

**Figure 1: Circuit Diagram**

There are five buttons on the circuit: cruise control, adaptive cruise control, cancel, and speed increase and decrease buttons. Each of these buttons has the following functions:

1. Start: The process begins.
2. Set Speed or Adaptive Cruise Control Button Pressed: If this button is not pressed, the system remains in normal mode at the starting speed (0 speed).
3. Normal Mode (0 Starting Speed): If the increase speed button is pressed, the speed will increase. If the increase button is released or the decrease button is pressed, the speed decreases to zero.
4. Cruise Control Activation: When the set speed or adaptive cruise control button is pressed, the system determines the mode.

**Mode 1 - Cruise Control:** Set the desired speed by pressing the set speed button. The speed will increase if you press the increase speed button, while it remains the same if you don’t adjust it. Pressing the decrease speed button will lower the speed. The current speed will be displayed.

**Mode 2 - Adaptive Cruise Control:** Press the adaptive cruise control button to activate this mode. The speed will decrease if you hit the decrease speed button when an obstacle is detected, while pressing the increase speed button will speed up the vehicle. Once the route is clear, the speed will adjust back to the predetermined setting.

# **4) Function Explanation**

This MATLAB code establishes an adaptive cruise control system using an Arduino board, an ultrasonic sensor, and an LCD display. Here’s a detailed explanation of each part of the code:

**Initial Setup:**

* clc; clear all; Clears the command window and removes all workspace variables,
* arduinoConnection = arduino('COM7', 'Uno', 'Libraries', {'Ultrasonic', 'ExampleLCD/LCDAddOn'}, 'ForceBuildOn', true);
* Connects to an Arduino Uno on COM7 with Ultrasonic and LCD libraries.

**Hardware Initialization:**

* ultrasonicSensor = ultrasonic(arduinoConnection, 'D9', 'D8');: Initializes the ultrasonic sensor on pins D8 and D9.
* lcdDisplay = addon(arduinoConnection, "ExampleLCD/LCDAddOn", 'RegisterSelectPin', 'D7', 'EnablePin', 'D6', 'DataPins', {'D5', 'D4', 'D3', 'D2'}) Sets up the LCD display with specific pin configurations.
* initializeLCD(lcdDisplay);: Prepares the LCD display.
* printLCD (lcdDisplay, 'Welcome');: Shows "Welcome" on the LCD, waits 2 seconds, then clears the display.

**Main Loop:**

* Continuously monitors inputs and controls the system in an infinite loop(While true).
* Reads button presses from analog pins A1-A5.
* Reads distance data from the ultrasonic sensor.
* Adjusts speed based on button inputs and distance.

**LCD Display:**

* Displays messages on the LCD based on system status and button inputs.
* Uses printLCD function for showing messages.
* Shows speed information on the LCD with proper formatting and timing.

**Button Functionality:**

* increaseSpeedSignal and decreaseSpeedSignal adjust speed up and down, respectively.
* setSpeedSignal switches to manual speed adjustment mode (cnt = 1).
* adaptiveCruiseSignal activates adaptive cruise control mode (cnt = 2) and sets the current speed as the speed limit.
* cancelSignal cancels any mode and resets to default control.
* Speed adjustments are based on button inputs and distance readings.

Overall, this code sets up a system where an Arduino board manages the car's speed using inputs from buttons and an ultrasonic sensor. The current status and speed information are shown on an LCD screen.

# **5) Components Details**

|  |  |  |
| --- | --- | --- |
| **Components & Quantity** | **Function** | **Image** |
| **MATLAB SOFTWARE** | MATLAB® is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world**.** | MATLAB - Wikipedia |
| **Bread Board (1)** | A breadboard is a circuit board with holes for inserting components and wires without soldering. | Breadboard | Components Tools + Power Supply |
| **Ardiuno UNO (1)** | It’s the core of the project, featuring a programmable microcontroller for interacting with electrical components. | A close-up of a black circuit board  Description automatically generated |
| **Ultrasonic sensor (1)** | It measures distance using sound waves and can detect objects. | A small electronic device with two round speakers  Description automatically generated |
| **Liquid Crystal Displays**  **(LCD) (1)** | They display circuit status or add visual effects to a project. | A blue screen on a table  Description automatically generated |
| **Potentiometer (1)** | A device for adjusting resistance in a circuit, used to control LED brightness or motor speed. | A black and white object with a black handle  Description automatically generated |
| **Resistors (7)** | They manage current flow in circuits, come in various resistance values, and limit current through components. | Resistor Color Codes, Bands, Power Ratings, and Other Useful Information |  Volts 'n' Bolts |

## **5.1)** **Connection Of Components**

**Step 1: Powering the Circuit**

**Arduino Power Supply:**

• Powered via the barrel jack or USB.

• Provides 5V to the ultrasonic sensor and LCD.

• Ground (GND) connections ensure a common reference.

**Step 2: Ultrasonic Sensor Setup**

**Connections**

• Connect the VCC (Power) pin of the Ultrasonic Sensor to the Arduino’s 5V pin.

• Connect the GND (Ground) pin of the Ultrasonic Sensor to the Arduino’s GND.

• Connect the Trig pin of the Ultrasonic Sensor to a digital pin D9 on the Arduino

• Connect the Echo pin of the Ultrasonic Sensor to D8 digital pin on the Arduino.

**Operation**:

• The Arduino sends a 10-microsecond HIGH pulse to the Trig pin to start the ultrasonic burst.

• The sensor emits a burst of ultrasonic waves and waits to receive the reflected waves.

• Once the reflected wave is detected, the Echo pin goes HIGH for a duration proportional to the time taken for the wave to return.

**Step 3: Displaying Distance on LCD**

**Connections:**

• VSS (GND): Connect to the Arduino’s GND.

• VDD (Power): Connect to the Arduino’s 5V.

• V0 (Contrast Adjustment): Connect to the middle pin of the potentiometer.

• RS (Register Select): Connect to a digital pin D7 on the Arduino.

• RW (Read/Write): Connect to GND (set to write mode).

• E (Enable): Connect to a digital pin D6 on the Arduino

• D4 to D7 (Data Pins): Connect to digital pins D5 to D2 on the Arduino.

**Operation:**

• The Arduino initializes the LCD using the LiquidCrystal library.

• It sends the calculated distance data to the LCD, updating the display.

• The potentiometer is used to adjust the contrast of the LCD for better readability.

**Step 4: Continuous Operation**

* The Arduino continuously Sends pulses from the Trig pin.
* Measures the duration of the echo.
* Calculates the distance based on the duration.
* Updates the LCD display with the new distance measurement.
* This cycle repeats every 500 milliseconds (or as specified in the delay function), ensuring real-time distance measurement and display.

# **6) Roles And Responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles and Responsibilities** | **Jas Prajapati** | **Nirav Polara** | **JayeshKumar Thummar** |
| **Managing Meeting Schedule** |  |  |  |
| **Making Records of Minutes of Meeting** |  |  |  |
| **Research on Previous Work** |  |  |  |
| **Understanding the Integration of Components** |  |  |  |
| **Flowchart and Process Formation** |  |  |  |
| **Primary Project Base Structure** |  |  |  |
| **Circuit Diagram & Coding** |  |  |  |
| **Testing** |  |  |  |
| **Quality Assurance and Final Testing** |  |  |  |
| **Documentation and Report Compilation** |  |  |  |
| **Preparation of Final Report** |  |  |  |

# **7) Conclusion**

In conclusion, our Adaptive Cruise Control (ACC) system effectively integrates advanced safety features to enhance vehicle automation and driver convenience. By utilizing an Arduino board, ultrasonic sensor, and LCD display, we successfully demonstrated a practical solution to improve driving safety and comfort. Our system's ability to automatically adjust speed based on real-time distance measurements addresses key issues related to unsafe driving behaviors, contributing to safer roadways. This project highlights the potential of adaptive technologies in advancing automotive safety and sets a solid foundation for future enhancements in vehicle automation.

# **8) References**

[1] World Health Organization, "Global status report on road safety 2018," 2018. [Online]. Available: https://[www.who.int/publications/i/item/9789241565684.](http://www.who.int/publications/i/item/9789241565684) [Accessed: Feb. 23, 2023].

[2] “What is Adaptive Cruise Control? | TomTom Newsroom,” TomTom. https://www.tomtom.com/newsroom/explainers-and-insights/what-is-adaptive-cruise-control/

[3] “Automatic cruise control – the Canadian Association of Road Safety Professionals (CARSP),” Apr. 20, 2023. <https://carsp.ca/en/news-and-resources/advanced-vehicle-safety-systems/automatic-cruise-control/>

[4] G. Marsden, M. McDonald, and M. Brackstone, “Towards an understanding of adaptive cruise control,” Transportation Research. Part C, Emerging Technologies, vol. 9, no. 1, pp. 33–51, Feb. 2001, doi: 10.1016/s0968-090x(00)00022-x. Available: <https://doi.org/10.1016/s0968-090x(00)00022-x>

[5] S. Naylor, “What is ACC (Adaptive Cruise Control)?,” Parkers, 05-Apr-2021. [Online]. Available: https://www.parkers.co.uk/what-is/acc-adaptive-cruise-control/. [Accessed: 21-June-2024].

[6] W. Pananurak, M. Parnichkun, and S. Thanok, "Adaptive cruise control for an intelligent vehicle," 2008 IEEE International Conference on Robotics and Biomimetics, Bangkok, 2009, pp. 1794-1799. [Online]. doi: 10.1109/ROBIO.2009.4913274. [Accessed: 20-June-2024]