Overview of Scalable Cruise Control
Software Engineering CSE 435
Michigan State University
Fall 2016

Team members:
- Project Manager: Tyler Armstrong
- Facilitator: Rahul Patel
- Customer Liaison: Evan Brazen
- Configuration Manager: Gustavo Fernandes
- Safety/Security Engineer: Anthony Capriglione

Customer: Mr. Eric Winder
Instructor: Dr. Betty H.C. Cheng*

*Please direct all inquiries to the instructor.
Project Overview

• The system’s functionality is to avoid collisions by always keeping the vehicle at a safe distance from traffic ahead and assisting the driver with braking incase of emergencies.

• Motivation for project
  – Focus the driver’s attention on the road rather than the vehicle’s speed and distance to the car in front.
  – To facilitate traffic flow
  – Reduce accidents
Overview of Features

• Simple Cruise Control
• Following Distance Management
• Automatic Emergency Brake
Simple Cruise Control

- Driver enables the feature and sets a maximum speed. If disabled, clears the set speed.
- Set speed must be greater than 25 mph.
- Feature can be canceled through button press or by tapping on the brake.
- If canceled, feature can be resumed to the previous set speed through button press.
- Driver can increase/decrease the set speed through button presses.
- Driver can exceed the set speed by pressing the throttle and resumes set speed when let go.
Following Distance Management

• Driver enables the feature and sets a following distance from a leading vehicle. Distance is represented to driver in four steps:
  – Close
  – Medium
  – Far
  – Extra Far

• Distances are calculated based on position and relative speed of leading and current vehicles.

• Alerts driver if vehicle comes too close.
Automatic Emergency Brake

- Functions with or without Simple Cruise enabled.
- Inputs come from vehicle speed and camera/radar object tracking.
- Driver enables this feature and when the car is rapidly approaching an object, the brakes are automatically triggered at maximum pressure.
- Works well in situations where the driver would not have been able to manually stop in time.
Domain Research

• Investigated available adaptive cruise control systems in today’s market.
  – Chrysler Adaptive Cruise
  – Nissan Intelligence Cruise Control
  – Uber Self-Driving Car

• Needed to apply domain knowledge of common cruise control functionalities and architecture to our system in regards to our requirements.

• Project Constraints
  – If any of the functions are not functioning properly, the system will shut off.
  – The set speed for the driver to enable SCC must be greater than 25 mph.
  – The system should not irritate the driver with the warnings.
  – Driver should be familiar with the functions of the system and the warnings it displays.
Part II: Model-based View of System

Cruise Control ECU
- vehicle_speed
- safe_distance
+ retrieve_radar_data()
+ retrieve_camera_data()
+ engage_simple_CC()
+ engage_follow_distance_management()
+ engage_AEB()
+ disengage_CC()
+ cancel_CC()
+ resume_CC()
+ set_vehicle_speed()
+ set_follow_distance()
+ increase_speed()
+ decrease_speed()

Object Sensor
- object_detected
- object_distance

Monitors

Camera
+ detect_object()

Radar Sensor
+ detect_object_distance()

Controls

Driver Interface
- cruise_speed
- follow_distance
+ enable_simple_CC()
+ enable_follow_distance_management()
+ enable_AEB()
+ disable_simple_CC()
+ disable_AEB()
+ cancel_CC()

Requests Actuation

Vehicle ECU
+ request_throttle()
+ request_brake()
+ request_audio_alert()
+ request_visual_alert()
Sequence Diagrams

- Depict the ordering of interactions between objects over time.
- Each object involved in the system is represented as a lifeline with a box and vertical dashed line for each lifeline.
- Events or operations between objects in the scenario are represented as labeled horizontal dashed or solid lines starting at the initiator and ending with an arrow at the lifeline in which the event or operation is to occur.
- Brackets represent a guard or a condition that must be met for an event to occur.
Simple Cruise Control

- In each scenario, the sequence of events begins with an external actor that acts on the system (the driver).

System maintains set speed through throttle actuator
Following Distance Management

Continuous Events

Driver Interface

Driver

Cruise Control ECU

Radar Sensors

Camera

Vehicle ECU

enable_following_distance_management()

engage_following_distance_management()

[simple_cruise_control == false]

engage_simple_CC()

set_follow_distance()

detect_object_distance()

retrieve_radar_data()

detect_object()

retrieve_camera_data()

request_brake()

request_throttle()

set_follow_distance()
Automatic Emergency Brake

Continuous Events

Driver Interface
Driver
Cruise Control ECU
Radar Sensors
Camera
Vehicle ECU

enable_AEB()
engage_AEB()

disable_AEB()
disengage_AEB()
detect_object_distance()
retrieve_radar_data()
detect_object()
retrieve_camera_data()

[safe_distance == false]
request_audio_alert()

[safe_distance == false]
request_brake()

[request_brake()== true]
request_visual_alert()
Part III: Demonstration

- Our Website
Acknowledgements

• We gratefully acknowledge and appreciate the participation of our customer, Eric Winder from Ford Motor Company.