Scalable Cruise Control

Customer: Mr. Eric Winder, Ford Motor Company

Motivation
As software-driven features have continued to proliferate in modern vehicles, systems which previously provided only driver notifications or convenience and comfort have begun to take a more active role in ensuring the safety of occupants and others. In addition, vehicle systems which provide more advanced features, like adaptive cruise control and crash mitigation, may use many of the same inputs and may even attempt to provide control commands which are contradictory. Prioritization of these command inputs is key to controlling the vehicle dynamics.

Many of these systems rely on distributed sensor inputs (yaw/roll, radar, camera, vehicle speed, throttle position, steering angle, etc.), with raw data and control information arriving over vehicle networks which may be open for the connection of other devices and tools. Another consideration is that cars have become further connected to the outside world, through various wireless technologies (Wifi, LTE, 802.11p V2x, etc.), and the attack surface for hackers has expanded. Security has become a major concern with high profile hacks in the news. The adage is proving true in these cyber physical systems that there is no safety without security, and the ability to remotely command throttle and brake behavior as these vehicle functions have moved under software control is a top safety concern.

Another major consideration for software design is the architecture. Short-sighted decisions at the time of architecture definition can reduce or eliminate the advantages of other such key non-functional factors as debuggability, extensibility and scalability, testability, and portability/re-usability. Some vehicle features scale from the very basic, available in the barest entry-level vehicles, to advanced driver assistance which has begun to cascade downward into well-equipped, high-volume production vehicles. Developing entirely separate systems or software variants to support features which exhibit a high level of functional overlap will result in duplicate design, implementation, maintenance, and testing effort and will increase cost, as well as the probability of issues across the different variants.

Scalable Cruise Control Description
The idea of speed control dates back hundreds of years, and the direct predecessors to modern cruise control designs are more than 50 years old. A selection of several speed control designs can be categorized as follows:

- Simple Cruise:
  - Driver enables the feature and sets a maximum speed to be maintained. Set speed must be greater than 25mph.
  - Vehicle maintains the set speed through throttle control and vehicle speed feedback.
  - Driver may exceed the set speed through direct throttle inputs.
  - Driver may cancel (suspend) the feature through a button press or by depressing the brake pedal.
  - Driver may resume the previously set speed by a button press, and the vehicle must accelerate or decelerate at a safe rate to the set speed (in the absence of throttle input from the driver).
  - Driver may increase/decrease the set speed while active, through button presses.
  - Driver may turn the feature off, which clears the previous set speed.
• Following Distance Management: Adds the following to simple cruise feature:
  o Driver may set a following distance from a leading vehicle through button controls. Distance is represented to the driver in 4 relative steps (i.e. “small” up to “large”) without units, and is maintained internally by the cruise controller as a measurement of “minimum time to intercept,” based on the position and relative speed of the leading vehicle. In this way, the driver setting scales over a range of speeds without compromising safety, but the speed control must take the current vehicle speed and distance into account.
  o Vehicle maintains maximum possible safe speed through throttle and brake control, with real-time inputs from vehicle speed and lead vehicle tracking information from camera and radar, within the constraints of set speed, minimum following distance/time, driver throttle input
  o Feature alerts driver in case of emergency (hard braking necessary)

• Automatic Emergency Brake (AEB, standard feature in calendar year 2022)
  o Functions with or without cruise enabled
  o Inputs come from vehicle speed, and camera/radar object tracking
  o Applies maximum braking pressure to minimize stopping distance, if driver cannot react in time or with sufficient force

The design should be done such that either simple cruise or cruise with following distance management are possible, with or without AEB. Common functionality should be reused without duplication. The design should also be extensible to take into account additional future features which may use the same base functionality to support such innovations as curve speed assist, DSRC-based platooning, DSRC-based traffic signal awareness (SPaT), pedestrian detection, etc.

Scenarios

Scenario 1: Given a system with simple cruise and AEB, a driver is driving along a highway at night with the cruise control engaged. Her feet are off the pedals. As she crests a hill, she sees a disabled vehicle in her lane, and has little time to react.

Scenario 2: Given a system with cruise supporting following distance management and AEB, a driver is driving through heavy rush hour freeway traffic with cruise control engaged. Traffic speed is varying in the right lane between 25-40mph, due to entering/exiting traffic. As traffic merges in, there are step changes in the measured distance to the lead vehicle.

Scenario 3: Consider the security of the system, which has inputs received and output commands transmitted via a vehicle network. Driver assistance features can take the role of driver, depending on their inputs. A hacker connects through a compromised LTE-connected ECU on the network and sends false sensor and system state data, as well as unauthorized throttle and braking commands.

Scenario 4: Consider the robustness of the distributed system. Sensor and control data is sent over networks in the vehicle. A hacker connects through a compromised LTE-connected ECU and sends many high-priority frames on the network which inhibit normal message traffic (sensor inputs or command outputs).