Overview

The modern automobile features more software components and electronics than ever before (Fig. 1.1). Unfortunately, many of these components (known as ECUs - Electronic Control Units) are “black boxes”—their internal states are difficult to determine. Thus, there is a need to remotely diagnose, update and certify automotive software for efficient warranty and safety management.

Recalls due to software errors have increased steadily as cars become more electronically complex. This results in huge costs for the manufacturers, the bulk of which could be avoided by better diagnostics for vehicles already on the road and more efficient repair methods for electronic errors. AutoPlug aims to provide a framework where these concepts can be implemented.

The AutoPlug testbed implements the following features that provide an inexpensive and efficient way for manufacturers to diagnose, repair, and recall vehicles:

- **Remote Diagnostics:** Manufacturers are able to interface with a vehicle remotely and determine if the car is operating properly.
- **Preemptive Maintenance:** Historical data gathered by the middleware layer can be used to warn the user about faults.
- **ECU Firmware Upgrades:** Upgrades to ECUs can be deployed without having to access the physical unit.
- **Recall Management:** The manufacturer would be able to access the vehicle and determine if the vehicle falls under a recall.

AutoPlug consists of three layers (Fig. 1.2) which validate the test bed through simulation, as well as provide an interface for the hardware manufacturers to access the vehicle’s information.

Implementation

**ECU Network Layer**

The ECU network (Fig. 2.1) consists of individual HC12 microcontrollers that perform each function of the car. Each controller is networked through the CAN protocol which allows the ECUs to communicate to each other. The ECU network in AutoPlug implements electronic functionality for basic options such as steering, throttle, and transmission as well as more complex functionality such as ABS and Cruise Control.

**Simulation Layer**

To simulate the operation of a car, AutoPlug uses the open-source racing simulator TORCS (Fig. 3.1). Using this physics–based racing simulator, we can generate the appropriate inputs and outputs to interact with the individual car ECUs, using a data acquisition board (DAQ) (Fig. 3.2).

**Middleware Layer**

The middleware layer (Fig. 4.1-2) provides the manufacturer an interface by which they can visualize the vehicle, perform diagnostics, upgrade ECUs and transmit data to remote devices. The dashboard validates the simulation and ECU network and displays vehicle state. The middleware also provides an interface for upgrading individual ECUs over CAN.

Results

AutoPlug is able to record simulation data from the ECU network over the CAN bus, qualitatively diagnose errors, and provide the ability to update individual ECU software over the CAN bus.

Example Scenario (Fig. 5): The cruise control module of the car is malfunctioning, and causes large errors with respect to the desired (set) speed.

Conclusions:

AutoPlug provides a framework for car manufacturer to remotely diagnose and correct ECU software errors. The large increase in electronic and software functions in vehicles has led to a need for manufacturers to efficiently address these problems without having to resort to traditional mechanical techniques (test and replace). AutoPlug provides a better and more cost effective method for dealing with software errors, giving the manufacturer better options at handling large scale recalls as well as isolated software errors.