Software Design

This document describes the overall software structure designed and developed as part of the TO22: Speed Harmonization Fundamental Research experimental framework. The software structure is illustrated in Figure 1. The server, on the left side of the diagram, serves as the surrogate traffic management center (TMC). The vehicle components, on the right side of the diagram, utilize an existing connected and automated vehicle (CAV) fleet platform to transmit current vehicle position and speed, and receive and respond to speed commands via the cellular network.



Figure 1. Diagram. Overall Software Structure.

# Surrogate Traffic Management Center

The surrogate TMC performs the following functions:

* Collect point detector data using a WaterSQL database
* Run the speed control algorithm
* Communicate with each participating CAV via its application program interface (API)
  + Receive speed and position data
  + Transmit speed commands
* Store experimental data (i.e., time and location of the experiment, the point detectors and CAVs involved, the measurements gathered from those CAVs, and the commands sent to them) in a SQL Server relational database
* Provide a rudimentary command-line user interface that allows an experiment manager to define a new experiment, assign vehicles to it, assign algorithms to vehicles, and monitor interactions of these elements during the experiment

# CAV Fleet Platform

An existing CAV fleet software platform was modified to manage several project-specific functions. Each CAV in the fleet was outfitted with:

* A longitudinal controller, a set of custom electronic control units that enable fully automatic control of vehicle acceleration and braking by integrating directly with the existing vehicle adaptive cruise control system
* A dSPACE MicroAutoBox controller, a specialized real-time computing platform that provides commands to the longitudinal controller, accessed via dSPACE ControlDesk through a MATLAB/Simulink library
* A Linux-based auxiliary computing platform that processes data the MicroAutoBox cannot directly manage
* A positioning system that uses dual-GPS receivers to provide real-time vehicle position data
* A wireless broadband modem connected to the cellular network, used to enable external communications

The Linux-based auxiliary computing platform performs the following functions:

* Communicate with the surrogate TMC
  + Request and receive speed commands
  + Request and receive status of other participating CAVs in experiment
  + Transmit vehicle ID, speed, acceleration, position, and other state variables
* Communicate with the MicroAutoBox controller
  + Receive vehicle ID, speed, acceleration and other state variables
  + Transmit speed commands
* Receive position data from the positioning system
* Compute a speed command confidence based on the age of the latest command
* Communicate with the HMI
  + Transmit speed command and speed command confidence
  + Transmit status of other participating CAVs in experiment

In addition, a tablet computer serves as a human machine interface (HMI) to display information on current speed, commanded speed, confidence in the commanded speed, and the status of surrounding CAVs to the driver of the CAV. The HMI is shown in Figure 2.

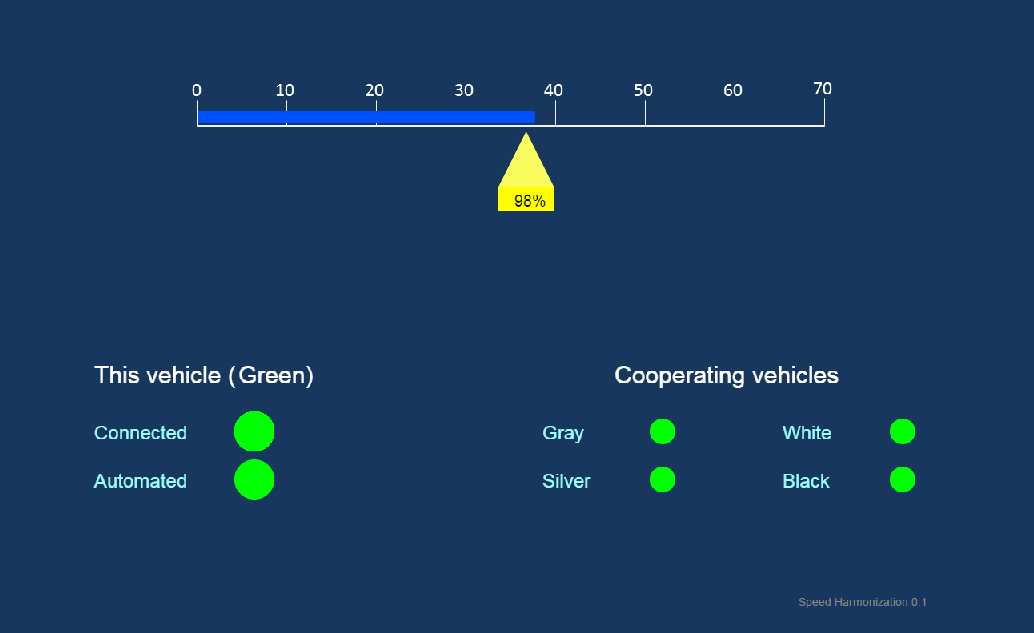


Figure . Screen Capture. Human Machine Interface.