Vehicle-to-Vehicle Control Challenge

Ken Henry, General Motors R&D and Planning
January 24, 2001

Show that an integrated, virtual environment can be used to develop a set of control algorithms with vehicle scope that can be deployed to a network of processors on a particular vehicle and further deployed to a small group of vehicles to meet a multiple-vehicle control objective. Within the virtual environment, model the effects of proposed deployments ahead of time to assure a well-behaved system and then confirm this behavior experimentally. Moreover, demonstrate that algorithm models captured in disparate modeling tools can be composed into a larger system model without loss of information.

Application to the SmartVehicle OEP

The vehicle-to-vehicle challenge is intended to stress continuity with the partitioned powertrain control challenge. We propose to build upon the models for the vehicle and the powertrain control algorithms and a separate model for cruise control algorithms.

In this challenge we wish to reuse the vehicle model and powertrain control algorithm model from the companion challenges as expressed in Matlab/Simulink/Stateflow. However, the adaptive cruise control algorithm has been designed by different group in a different modeling environment (SHIFT or Teja). This situation, where designs and models from multiple sources must be composed into a coherent system, is common within the automotive industry.

> Given the vehicle, powertrain, and throttle system models in Matlab/Simulink/Stateflow and the adaptive cruise control algorithm modeled in Shift or Teja, validate and realize the behavior of the complete adaptive cruise control system in the virtual environment. In this context, validation and realization includes:
>
> - System simulation
> - Hybrid system analysis (e.g. Show that, given a radar sensor failure that yields a stepwise change in distance-to-the-lead-vehicle-measurement error from 0 to > 100 m, no collision occurs when the lead vehicle is traveling at 100 KPH and its initial separation is 20 m.
> - Embedded software analysis and functional allocation
> - Embedded software synthesis

The first physical demonstration is a well-behaved vehicle cruise control algorithm. The cruise control algorithm should be developed with the vehicle simulation, using the engine-dynamometer OEP for verification..

The second physical demonstration is a well-behaved adaptive cruise control (ACC) algorithm for two vehicles. This should be done by instantiating a second, independent vehicle /control algorithm simulation within the virtual environment and adding the necessary sensor models to "couple" the two simulations. The ACC could then be developed/verified on the engine-dynamometer OEP where one vehicle is simulated and the other is hardware-in-the-loop, and then moved to a two-vehicle OEP for final validation.