A Real-time Hierarchical Software Architecture for Coordinated Vehicle Control

Adam Howell, Anouck Girard, J. Karl Hedrick, Pravin Varaiya

University of California, Berkeley
Overview

• Model Based Integration of Electronic Systems (MoBIES)
  – Investigation of improved software development tools for control software
  – Automotive applications for powertrain, adaptive cruise control, etc.

• Outline
  – Background / Problem Description
  – Design Methodology
  – Modeling
  – Control Architecture
  – Maneuver Design
  – Experimental Implementation
  – Concluding remarks
Intelligent Cruise Control

Cooperative Adaptive (Intelligent) Cruise Control (CACC):
Cruise at given speed when the road is clear (cruise control), otherwise follow the car in front, using radar (adaptive) and/or communications (cooperative).
Why vehicle following is hard...

- Around curve
- Elevation changes
- Cut-ins
- Rapid deceleration
- Dense traffic
Background: AHS

- Requirements
  - increase throughput
  - full automation
- Features of the proposed solution
  - hierarchical decomposition
- Motion coordination
  - string stability issues
- Simulation and Implementation
MoBIES ACC/CACC

- Requirements
  - increase driver comfort
  - partial automation
  - management of complexity
  - human factors
  - increase throughput

- Features of the proposed solution
  - hierarchical decomposition
  - layered theories

- Motion coordination
  - string stability issues

- Simulation and Implementation
  - TEJA
Design Methodology
MoBIES Experiment Goal: Show an end-to-end example of tool integration for the Automotive Electronics Open Experimental Platform. In this case, design an ACC/CACC using hybrid control theory and produce code that runs on a specific CPU board. Measure results in terms of toolset performance, not code performance.

## Stages and Flow of the Conventional Design Process:

<table>
<thead>
<tr>
<th>Modeling</th>
<th>Controller Synthesis</th>
<th>System Analysis</th>
<th>Code Synthesis</th>
<th>Validation Verification</th>
<th>Target Analysis</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulink</td>
<td>Plant and Control Model</td>
<td>Valid Control Model</td>
<td>Code</td>
<td>Design Feedback</td>
<td>Few tools available; little automation; weak interoperability</td>
<td></td>
</tr>
<tr>
<td>Stateflow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COTS development tools

- **Generic modeling and synthesis tools:**
  - Not customizable
  - Proprietary internal data formats

- **Model Checking and Performance Analysis**
  - Model checkers don’t understand the application.
  - Most modeling tools don’t have built-in verification tools

- **Generate code for target CPU**
  - Neglects distributed processing
  - Considers CPU only; not rest of system
  - Code produced is unstructured and un-verifiable by automated means: detailed hand-verification is required

- **Simulink Stateflow**
- **Matlab Simulator**
- **R-T Workshop**
- **Manual Testing**
- **Manual Testing**
Development Process

- **Plant Library**
  - Teja

- **Simulation**
  - Teja

- **Hybrid System Verification**
  - Third party tools

- **QNX Machine**
  - Low-level C code
  - Device drivers
  - P/S database

- **QNX Machine**
  - C++ code

- **Schedulability Analysis**
  - Third party tools

- **Car, Pentiums**

- **Until HSIF matures**
Modeling
Abstractions

- Hybrid automata

\[ a = -v + 100q \]

- Dynamic reconfiguration
  (a challenge to the mathematical theory of control)
ICC Model

CNG/Diesel Engine

- Uses 1\textsuperscript{st} C.T. nonlinear model for engine torque dynamics
- 2 External Data Maps are used which require both 1 and 2-d interpolation.

Vehicle State Dynamics

- Includes vehicle mass, air drag, rolling resistance, no slip tire model, etc.

Transmission

- Discrete transitions are taken during gear changes based on vehicle speed.
- Abrupt gear changes cause abrupt gear ratio changes, so a filter is added which includes 1 C.T. state.

Torque Converter

- No C.T. states. 2 Hybrid states: Coupled & Uncoupled

Pneumatic Brakes

- 2 C.T. State representing time response lag for empty/fill

Transmission Diagram

Vehicle State Dynamics Diagram

Torque Converter Diagram

Pneumatic Brakes Diagram

CNG/Diesel Engine Diagram
Control Architecture
Hierarchical Organization

- Each layer provides different kinds of services and behaviors
- Layers may or may not be centralized
- Models of computation and communication need to be different at different levels in the architecture.
  - Lower levels = time driven or event driven (such as camshaft based computations); synchronous communications
  - Higher levels = state machines and/or discrete time abstractions and benefit from using asynchronous, very reliable, but possibly “slow”, communications.
Transit Bus Architecture

- **Supervision Layer**
  - Driver interface (DVI)
  - Scripting for event/time based scenarios

- **Coordination Layer**
  - Manual/automatic
  - Leader/follower

- **Regulation**
  - Speed/distance tracking
  - Fuel/brake control

- **Database**
  - Interface to low level drivers/hardware
Difficulties with Architectures

- More than interfaces - Exhibit structural properties that determine the performance of the design and possibly the project

- Lack of formal representation
  - No good description language
  - Design patterns gaining acceptance
  - No information about connecting patterns
  - No information about requirements to patterns

- Little formal basis to study properties at the level of the architecture
  - Static vs. dynamic
  - Structural (connection) properties
  - “Safety” properties (reachability)
  - Deadlock, liveness,…
  - Scalability is an issue (most problems are combinatorial)
Maneuver Design
Lead Vehicle Trajectory Design

- Different profiles for acceleration and deceleration due to differing vehicle response
  - Acceleration uses first order system
  - Deceleration uses second order polynomial on acceleration
- All profiles are parameterized by max absolute acceleration, initial velocity, and final velocity.
Following Vehicle Trajectory Design

- Distance profile generated as fifth order polynomial
- Parameterized by max absolute acceleration, initial and final distances
- Higher order polynomial used to ensure smoother transitions at beginning and end of maneuvers
- Can be implemented as a set of algebraic equations
Experimental Implementation
Experimental Platform
Transit Bus Video
Research directions

Autonomy: Bringing intelligence into vehicles

- **Control architectures**
  - Formal descriptions
  - Verification of architectural properties
  - Requirements traceability and acceptance testing

- **Coordinated Vehicle Control**
  - Mesh-stable control algorithms
  - Maneuver design and switching
  - Fault detection/tolerance issues
  - Wireless communication and control

- **Implementation**
  - Model-based approaches
  - Hardware/software co-design
  - Integration of tools (hybrid system interchange format)