Simulating Highly Complex Systems to Deliver the Next Generation of Jaguar Land Rover Vehicles

Andy Richardson
Head of Simulation
Jaguar Land Rover

6th Nov 2012
The All New Range Rover
Jaguar Land Rover Product
Business Drivers and Benefits for Simulation

Development and Delivery of Leading Edge Product Technology
   Development of New Concepts
   Develop and Deliver New Technology

Improved Design Decisions
   Improved Data to Support Robust Decisions

Improved Design Robustness
   Identify Failure Modes and Establish Countermeasures
   Reduced Sensitivity to Noise Factors

Optimum Product Performance
   Class leading attribute performance
   Optimised product (weight / CO2)
   Design Efficiency, Product Cost

Business Efficiency
   *Right First Time* Design – Reduced redesign, retool, rebuild, retest
   Increase Virtual Test – Reduce Physical Test
   Improved Design Process Efficiency
   Faster Time to Market

Customer Experience and Satisfaction
   Reduced Failure in Service, Reduced Warranty
   Improved Customer Satisfaction

Typical Approx Costs:
   - Cost of Late Change = £10m's
   - Prototype Vehicle =~£0.5m
   - Testing a Prototype =~£0.5m
   - 1 week delay on a vehicle sign-off = £1.0m (resources)
   - Lost revenue due to 1 wk launch delay = £50m
Typical Automotive Control System
System Complexity – High Speed

Chassis Systems:
- Adaptive Damping Module
- Air Suspension Module
- Brake Control Module
- Steering Angle Sensor
- Suspension Module
- Steering Wheel Module
- Terrain Optimisation Switch
- Active Aerodynamics

Powertrain Systems:
- All Wheel Drive Module
- Combustion Preheater Module
- Differential Electronic Module
- Engine Management System
- Engine Control Module
- Fuel Additive Control Module
- Gear Shift Module
- Start Control Unit
- Transmission Control Module
System Complexity – Medium Speed

Comfort Systems:
- Climate Air Quality Sensor
- Climate Control Module
- Defroster Control
- Panel & Floor Air Distribution Flap
- Recirculation Control
- Temperature Control Left
- Temperature Control Right

Convenience Systems and Features:
- Accessory Electronic Module
- Advanced Front Lighting System
- Battery Backed Siren
- Central Electronic Module
- Light Switch Module
- Rear Electronic Module
- Rear Smart Junction Box
- Trailer Module
- Upper Electronic Module
- Wiper Motor Module
- Driver Door Module
- Occupant Weight Sensor
- Passenger Seat Module
- Power Seat Module
- Seat Heating Module Left
- Seat Heating Module Right
- Sun Roof Module
System Complexity – High Data

Infotainment Systems:
- Audio Module
- Bluetooth Phone Module
- Digital Audio Broadcast Radio
- Driver Information Module
- DVD Player
- High Level Display Front
- High Level Display Rear Left/Right
- Instrument Cluster
- Infotainment Control Panel
- Integrated Head Unit
- Multimedia Module
- MD Player
- TeleVision Module
- Vehicle Emergency Messaging System
- Vehicle Information Channel
- RF Receiver
- Subwoofer Module
- Telematics Module
- Phone Module
- Radio Data Channel, Traffic Message Channel.
System Interdependencies

PED -> T&D
EMS -> 1D Base Eng
PiVA

SCS -> SST
RSC

V Dynamics
Driving Sim

Door Mech -> Wipers
Climate

Body ViTAL
Comfort

IPM

HEV Sim
HEV Cont
What makes luxury cars complex systems?

- Up to 100 Electronic Control Units
- Different networks such as HSCAN, MSCAN, LIN, FlexRay and MOST
- ~3,000 data signals Updated upto 100 * /second
- More than 20 LIN buses
- More than 1,500 features and 15,000 functions
- Up to 9 operational power modes
- Introduction of new technology e.g. hybrid vehicles
- Software Variants for different markets
Aerospace vs. automotive control system complexity

F-22 Raptor  
The avionics system in the F-22 Raptor, Air Force frontline jet fighter consists of about **1.7 million** lines of code

Boeing 777  
4 million lines of code implemented in 79 different systems to operate its avionics and onboard support systems

F-35 Joint Strike  
F-35 joint strike fighter requires about **5.7 million** lines of code to operate its onboard systems

Boeing 787  
Boeing’s new Dreamliner requires about **6.5 million** lines of code to operate its avionics and onboard support systems

Premium car  
Modern luxury car today requires about **100 million** lines of code to operate its control and comfort systems

Within a few years cars could require **200 MILLION to 300 MILLION** lines of software

Source: IEEE Spectrum, (2009). This car runs on code
1. Strategic Approach

- 2D CAD
- Ad-hoc Process & Data Mgmt
- In-House Tools
- Limited Integration

- 3D CAD
- PDM focus on Mechanical BOM
- Organisation & Process Improvement for Geometry Build & Analysis
- Limited System Design focus
- Software development tools mixture of in-house & commercial with limited data management

- Focus on System Design & Interaction
- Focus on Full Requirements Traceability across Lifecycle
- Tailoring Processes to Complexity

2. Systems Engineering

3. Strategic Relationship

Product Complexity Challenges
1. Strategic Approach Framework

- Consistent engagement with capable suppliers
- Capable, and performant hardware tools
- Technically capable and compatible tools
- Recruit and develop systems simulation skills
- Develop and share best practice techniques
- Capable and robust simulation methods
- Aligned standards and conventions
- Standardised and efficient (lean) processes. Automated where possible.
## 2. Systems Engineering Approach

### Information Framework

<table>
<thead>
<tr>
<th>Business Context</th>
<th>Customer Concept</th>
<th>Logical Systems</th>
<th>Physical Systems</th>
<th>Component Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Strategy / Direction</strong></td>
<td><strong>Customer Experience</strong></td>
<td><strong>System Behaviour</strong></td>
<td><strong>Allocate System Behaviour</strong></td>
<td><strong>Component Detail</strong></td>
</tr>
<tr>
<td>“Market/Legal, Programme, Product, Portfolio &amp; Ideas”</td>
<td>“Handbook Functionality”</td>
<td>“Function Without Form”</td>
<td>“Component System Interfaces”</td>
<td>“Hardware &amp; Software”</td>
</tr>
</tbody>
</table>
2. Systems Engineering Approach Process

- Feature & Attribute Defined & Verified
- Customer Functionality Defined & Verified
- Control System Defined & Verified
- Physical System Defined & Verified
- Hardware and Software Defined & Verified
2. Systems Engineering Approach
Model Based Design

Concept Control Design
- Desktop Simulation
- Component Sizing
- Model Based Calibration

Driver Models

Environment Models

Requirements

System Verification & Test
- Hardware-in-the-Loop (HIL)

Component Software Generation
- Auto-Code Generation (ACG)
- Software-in-the-Loop (SIL)
3. Strategic Partnerships 
Matlab/Simulink

- System Modeling environment from Mathworks Inc
- JLR Core Environment for fully Integrated System Simulation
- Supports Model to Code to Validation Methods
- Incorporate Event Driven Control & Simulation

Mathworks identified as strategic simulation tool supplier
Event Driven Systems
e.g. Wash/Wipe System

Wipers
Screen Wash
Wiper Motor
Head Lamp Wash
Stalks
Rain Sensor
Rear Wiper
Screen Wash
Ignition
ECU
Gear Shift
Wash/Wiper controller development using Stateflow

Requirements document or Functional Area Description (FAD) no more than 15 pages

2 Wash Wipe System
The front wash/wipe function consists of the following features:

- Two Speed Wiper Control (LOW and HIGH)
- Automatic Parking of the Wipers
- Intermittent Wiper mode
- Moisture Sensitive Wiping (Rain sense)
- Programmed Screen Wash and Wipe Sequence

Whenever the wiper motor is driven (due to a command from the wiper switch), the wiper arms will complete a minimum of one complete wipe.

The wiper system will only be operational when the ignition is in IGN. If the ignition switch is moved out of the IGN position the front wiper system will be disabled and the wipers will return to the PARK position, ONLY AFTER COMPLETING THE CURRENT WIPE. This is also the case when the wiper switch is turned to OFF.

When changing between wiper modes or wiper dwell times (interval wipe) the wiper arms will maintain a continuous movement without stutter.
Wash/Wiper Control System

- Wiper Controller Development
- Model to Production Code Gen
- Entirely Event Driven
- Stateflow Modelled
- Executable Specification

Main Controller (Stateflow Model Suitable for Auto-coding)

~150 states
~180 variables

Wiper System (Electric motor and mechanical system)
Real Time Continuous Systems

e.g. Vehicle System Dynamic controllers

- Stability Control System
- Roll Stability Control
- Active Dampers
- Air Suspension
- Active Differential
- Powertrain Controller
- Active Roll Control
Stability Control System

On vehicle control systems

Passive Vehicle

Steady State Tyres

Simulation Environment

Road / Driver

On vehicle control systems
## Model Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Component</th>
<th>Requirement</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steady State Tyres</td>
<td>RSC</td>
<td>Engine</td>
<td></td>
</tr>
<tr>
<td>SCS Controller</td>
<td>Foundation Brakes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVD</td>
<td>Active Damping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2ARC</td>
<td>EPAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Road profile</td>
<td>C-DIFF</td>
<td>Transient Engine</td>
<td></td>
</tr>
<tr>
<td>Driver Model</td>
<td>E-DIFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential</td>
<td>Transmission</td>
<td>Driveline disconnect</td>
<td></td>
</tr>
<tr>
<td>Airspring</td>
<td>Auto TR</td>
<td>Hybrid Powertrain</td>
<td></td>
</tr>
<tr>
<td>ABS Hydraulics</td>
<td>Transient Tyres</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transient Drivetrain</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example Vehicle System
SCS Model Integration Architecture

- Created Re-Useable Simulation Architecture
- Minimises the Simulink model changes for ECU software updates
  - Utilising Simulink Bus Creator & Selecton
Example Vehicle System
Customer SW Model Integration

- Roll Stability Control (JLR) Code integration with Stability Control System (supplier)

- JLR RSC Strategy Model:
  - 250 inputs, 40 outputs, 2500 Internal variables
  - within C-code (~50k lines)

- Main ECU Supplier Model:
  - 300 inputs, 500 outputs

- Test Automation via Matlab for:
  - Initialisation & Post processing
  - Complete Simulink Environment + 3rd Party Integration
  - Functional & Robustness Evaluation
Example Vehicle System
JLR controller on/off animation
Modelling Complex Systems
e.g. Hybrid Vehicle Simulation

What is modelled?

Conventional Vehicle
- Engine Emissions, Heat & Fuel Consumption
- Transmission Losses & Heat

Parallel Hybrid
- Electrified FEAD loads
- Disconnect Clutch Drag
- Electric Motor Torque/Power Limits & Efficiency
- Power Electronics Limits & Efficiency
- Hybrid Supervisory Controller

Driver Control
- Driveline Losses & Heat
- Chassis Dynamics & Roadload
- Electro-Hydraulic Brakes Torque Control
- Hi-Volt Battery Efficiency & Charge Acceptance
Hybrid Vehicle Modelling

• Standardised Vehicle Modelling Architecture deployed across JLR
  > Re Use Vehicle Sub Systems
  > Architecture supports offline and real-time applications
  > Run Scenarios, Adjust Parameters and Visualise Outputs
Vehicle Supervisory Controller Model Development

- Jaguar Land Rover hybrid vehicle supervisory control software is developed in-house in the Simulink environment

- Simulink Library-Links utilised to allow parallel independent control function development
  > ~25 developers at any one time

- Highly complex controller model:
  > 244 separate control components
  > 8000 subsystems
  > 70000 blocks
  > Software changes are immediately tested in the offline & HIL environments
Hybrid Vehicle Controller Calibration Optimisation

- Automated optimisation process for calibrating the vehicle supervisory controller
  > This process uses a combination of Simulink vehicle models and optimisation algorithms coded in Matlab

Optimisation Challenge
Benefits of a Systematic Approach to Model Based Design

Simulation Capability:
1. Strategic Approach
2. Systems Engineering
3. Strategic Relationships

- Improved control development process
- Use models to describe the system
- Reduce document based specification
- Improved requirements capture process
- Auto-coding
- More commonality
- Smarter Testing & Automation
Future Challenges

Demand:

• Demand for more new product faster
• Pace of introduction and change of complex new technology
• Increasing complexity and interdependency of vehicle systems

Delivery:

• Managing increasingly complex modelling environment
• Managing 300million lines of code!
• Finding Enough Skilled Systems Simulation Engineers to develop these future systems
The All New Jaguar F-Type
Thank You