Simulink as a Platform for Full Vehicle Simulation

Mike Sasena (Product Manager)
Lars Krause (Application Engineer)
Ryan Chladny (Development)
Fuel Economy Simulation
Vehicle Dynamics Simulation

Ride & handling

Chassis controls

ADAS / AD
Agenda

- Product overview
  - Powertrain Blockset
  - Vehicle Dynamics Blockset

- Six Cool Things You Can Do
  - Engine modeling and calibration
  - Cooling circuit modeling
  - Aftertreatment modeling
  - Ride and handling analysis
  - Chassis controls development
  - ADAS / AD testing

Why are these cool?
- Reduce time on HIL, dyno, vehicle testing
- Account for complex thermal behavior
- Estimate tailpipe emissions more accurately
- Assess longitudinal / lateral dynamics
- Perform closed-loop testing
- Test in a virtual 3D environment
Agenda

- Product overview
  - Powertrain Blockset
  - Vehicle Dynamics Blockset

- Six Cool Things You Can Do
  - Engine modeling and calibration
  - Cooling circuit modeling
  - Aftertreatment modeling
  - Ride and handling analysis
  - Chassis controls development
  - ADAS / AD testing
Background

- **Context**
  - Automotive engineers need to evaluate powertrain systems as early as possible
    - What is the expected fuel economy, performance and emissions of my vehicle?
    - What is the impact of my controller on system efficiency?
    - Which electrification strategy should we develop?
  - **Model-Based Design** has become an important methodology for answering these questions and accelerating the development process

- **Challenges**
  - It’s hard to do good Model-Based Design without good models
Powertrain Blockset

- **Goals:**
  - Provide starting point for engineers to build good plant / controller models
  - Provide open and documented models
  - Provide very fast-running models that work with popular HIL systems

*Lower the barrier to entry for Model-Based Design*
Powertrain Blockset Features

Library of blocks

- Energy Storage and Auxiliary Drive
- Drivetrain
- Propulsion
- Transmission
- Vehicle Dynamics
- Vehicle Scenario Builder

Pre-built reference applications
Reference Applications

- Full vehicle models (conventional, EV, multi-mode HEV, input power-split HEV)
- Virtual engine dynamometers (compression ignition, spark ignition)
Four Use Cases. One Framework.

Use Cases:
1. System design and optimization
2. Controller parameter optimization
3. System integration test
4. Software-hardware integration test (HIL)
Powertrain Blockset Value Proposition

- **Open** and documented library of component and subsystem models
- Prebuilt vehicle models that you can parameterize and **customize**
- **Fast**-running models that are ready for HIL deployment
Agenda

- **Product overview**
  - Powertrain Blockset
  - **Vehicle Dynamics Blockset**

- **Six Cool Things You Can Do**
  - Engine modeling and calibration
  - Cooling circuit modeling
  - Aftertreatment modeling
  - Ride and handling analysis
  - Chassis controls development
  - ADAS / AD testing
Background

- **Context**
  - Automotive OEM’s and Tier 1 suppliers must assess vehicle’s dynamic performance
    - Will the vehicle roll over?
    - What’s the stopping distance of the vehicle?
    - Do the stability controls perform adequately?
  - Answer questions by building prototypes and / or running simulations

- **Challenges**
  - Prototypes are expensive, so must achieve a good design as early as possible
  - Specialized vehicle dynamics simulation software is quite expensive and difficult to use
  - Integrating 3rd party vehicle dynamics software with Simulink controls is cumbersome
Vehicle Dynamics Blockset
New product (R2018a)

- Model and simulate vehicle dynamics in a virtual 3D environment
- Use Vehicle Dynamics Blockset for:
  - **Ride & handling**: characterize vehicle performance under standard driving maneuvers
  - **Chassis controls**: design and test chassis control systems
  - **ADAS / AD**: create virtual 3D test ground for ADAS and automated driving features
Vehicle Dynamics Blockset Features

Library of blocks

Pre-built reference applications

Game engine
Game Engine Co-Simulation

**Simulink**
- Physics of vehicle
- Initialization of game engine camera

**Unreal Engine**
- Rendering / lighting
- Physics of non-Simulink objects
- Collision detection

Vehicle / camera location

Camera image, ground height, …
Reference Applications

**Vehicle Maneuvers**
Analyze ride and handling on driving maneuvers such as:
- Double-lane change
- Swept sine steering
- Slowly increasing steering

**Scene Interrogation**
Configure the interface to the 3D environment
Vehicle Dynamics Blockset Value Proposition

- **Open** and documented library of component and subsystem models
- Prebuilt vehicle models that you can parameterize and **customize**
- **Fast**-running models that are ready for HIL deployment
- Interface to **Unreal Engine**
Powertrain Blockset and Vehicle Dynamics Blockset: Flexible Modeling Framework

1. Choose a vehicle configuration
   - Select a reference application as a starting point

2. Customize the plant model
   - Parameterize the components
   - Customize existing subsystems
   - Add your own subsystem variants

3. Customize the controllers
   - Parameterize the controllers
   - Customize supervisory control logic
   - Add your own controller variants

4. Perform closed-loop system testing
   - Design optimization
   - Sensitivity analyses
   - MIL / SIL / HIL testing
Agenda

- Product overview
  - Powertrain Blockset
  - Vehicle Dynamics Blockset

- Six Cool Things You Can Do
  - Engine modeling and calibration
  - Cooling circuit modeling
  - Aftertreatment modeling
  - Ride and handling analysis
  - Chassis controls development
  - ADAS / AD testing
  - Reduce time on HIL, dyno, vehicle testing
Engine Dynamometer Reference Application

- Powertrain Blockset includes virtual engine dynamometer reference applications
- These can be used for a variety of engine controls development and calibration activities
- Includes several pre-defined experiments
Pre-defined Experiments for Automating Analyses

Simulate engine over grid of **steady state** operating points

- Import engine dynamometer **test data**
- Follow **transient** torque / speed profile
- Calibrate controller to match torque command
- Scale engine and recalibrate controller

**Execute Engine Mapping Experiment**
**Execute Model Predictive Control Plant Model Experiment**
**Recalibrate Controller**
**Resize Engine and Recalibrate Controller**

Dynamometer Control

Environment

Engine System
Automated Calibration Experiment
Executable Test Specification

- Describe the calibration procedure as a Stateflow chart (not a Word doc)
- Test the procedure virtually
- Validate / plan calibration procedure with test engineers
- Start testing on real hardware with refined procedure
Flexible Testing Framework

Use Powertrain Blockset mapped engine blocks with your own data

Create custom engine models using Powertrain Blockset library components

Connect in your own engine model (e.g., 3rd party CAE tool)
Controls Validation with Engine Model Co-Simulation
Controls-oriented Model Creation

- Detailed, design-oriented model

Fast, but accurate controls-oriented model
How Accurate is the Mapped Engine Model?

Auto-generated mapped engine model vs. co-simulation with design-oriented CAE model:
- 0.3% fuel economy difference
- 50x faster
Engine Test Data Import

Engine Dynamometer

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Torque</td>
<td>EngSpd</td>
<td>AirMassFlwRate</td>
<td>FuelMassFlwRate</td>
<td>ExhTemp</td>
<td>BSFC</td>
</tr>
<tr>
<td>Unit:</td>
<td>N*m</td>
<td>rpm</td>
<td>kg/s</td>
<td>kg/s</td>
<td>K</td>
<td>g/(kW*h)</td>
</tr>
<tr>
<td>Data:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.598</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45.847</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56.508</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68.245</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76.223</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.544</td>
<td>1053.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62.851</td>
<td>1053.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74.31</td>
<td>1053.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85.738</td>
<td>1053.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95.025</td>
<td>1053.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.676</td>
<td>1357.1</td>
<td>0.007204383</td>
<td>921.4745</td>
<td>338.2008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48.412</td>
<td>1357.1</td>
<td>0.008373948</td>
<td>926.201</td>
<td>300.2074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55.84</td>
<td>1357.1</td>
<td>0.009733333</td>
<td>941.4953</td>
<td>282.7931</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71.209</td>
<td>1357.1</td>
<td>0.011219721</td>
<td>955.8058</td>
<td>273.1425</td>
<td></td>
<td></td>
</tr>
<tr>
<td>82.697</td>
<td>1357.1</td>
<td>0.012888208</td>
<td>969.9667</td>
<td>268.1969</td>
<td></td>
<td></td>
</tr>
<tr>
<td>94.126</td>
<td>1357.1</td>
<td>0.014331802</td>
<td>1007.937</td>
<td>236.8019</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Engine Modeling and Calibration

- Calibrate engine control inputs to match torque command
- Define and simulate custom calibration procedures
- Generate engine maps from CAE models or engine dyno data
Agenda

- **Product overview**
  - Powertrain Blockset
  - Vehicle Dynamics Blockset

- **Six Cool Things You Can Do**
  - Engine modeling and calibration
  - *Cooling circuit modeling*
  - Aftertreatment modeling
  - Ride and handling analysis
  - Chassis controls development
  - ADAS / AD testing
  - Account for complex thermal behavior
Engine Cooling System

- Take customization one step further
- Start with “Custom Driveline” variant
- Add Engine Cooling subsystem adapted from sscfluids_engine_cooling_system
Conventional Vehicle with Simscape Engine Cooling

1. Heat rejection calculation

2. Heat distributed between oil and coolant

3. Temperature of cylinder used to validate cooling system performance
Cooling Circuit Modeling

- Create detailed, multi-domain subsystem models with Simscape
- Incorporate them into system level vehicle models from Powertrain Blockset
- Validate subsystem performance with closed loop simulation
Agenda

- Product overview
  - Powertrain Blockset
  - Vehicle Dynamics Blockset

- Six Cool Things You Can Do
  - Engine modeling and calibration
  - Cooling circuit modeling
  - Aftertreatment modeling
  - Ride and handling analysis
  - Chassis controls development
  - ADAS / AD testing
  - Estimate tailpipe emissions more accurately
Exothermia

- *Exothermia* is a MathWorks Connections Partner
- **axisuite®**: modular software for the simulation of exhaust aftertreatment devices and systems
  - axitrap: for wall-flow particulate filters
  - axicat: for flow-through catalytic converters with any kind of catalytic coating
  - axifoam: for foam-based or fiber-based filters and catalysts, with any type of catalytic coating
  - axiheat: for connecting pipes
- Models can be exported as S-functions for coupling with Simulink-based software, e.g. Powertrain Blockset

*axisuite®* is a trademark of Exothermia
www.exothermia.com
Catalyst simulation in the ‘V-shape’ development process

- **Requirement analysis**: Catalyst experts
- **Preliminary design**: Concept analysis
- **Detailed design**: Model calibration with *axisuite*: obtain the reaction rate parameters of each coating technology
- **Module test**: CFD, FEA experts
- **Integration test**: Controls department
- **System test**: Systems integration

CFD and control experts use their preferred simulation platforms. No need for recalibration of chemistry.
Catalyst Modeling Scales

- Dedicated simulation software for catalytic exhaust aftertreatment
- Extensively validated and applied by most automotive OEMs and suppliers
Overview of Flow-through Catalyst Model Equations in axisuite

Pontikakis et al., Top. In Catal, 2001
Use Cases: Aftertreatment System Design

- Use axisuite to design aftertreatment system
  - Determine required flow rates, thermal properties, etc.
  - Estimate conversion efficiency, O2 storage dynamics, etc.

- Couple with Powertrain Blockset to evaluate at vehicle level
  - Test on different drive cycles, ambient conditions, etc.
  - Perform design studies, sensitivity analysis, etc.

Catalyst experts

Concept analysis

CFD, FEA experts
Use Cases: Aftertreatment Controls Development

- **Closed-loop testing of controls features**
  - AFR control (rear trim), SCR control, etc.
  - Catalyst light-off calibration, thermal management, etc.

- **Diagnostics and predictive maintenance**
  - OBD catalyst monitoring
  - Front / rear O2 sensor feedback
  - DPF regeneration feedback

Controls department

Systems integration
Example: Conventional Vehicle + TWC / GPF

- Started with ConVeh project
- Replaced default TWC with axisuite model
Results

Vehicle speed

Exhaust gas flow rate

Catalyst temp

Tailpipe emissions

Lambda (post TWC)

Conversion efficiency
Aftertreatment Modeling

- Account for system level interactions (driver, vehicle, engine, aftertreatment, etc.) in a single environment

- Study impact of design and control changes on overall vehicle performance

- Couple high-fidelity aftertreatment model with real driving conditions
Agenda

▪ Product overview
  – Powertrain Blockset
  – Vehicle Dynamics Blockset

▪ Six Cool Things You Can Do
  – Engine modeling and calibration
  – Cooling circuit modeling
  – Aftertreatment modeling
  – **Ride and handling analysis**
  – Chassis controls development
  – ADAS / AD testing

  – Assess longitudinal / lateral dynamics
Reference Application: Double Lange Change
Reference Application: Double Lange Change (Maneuver)

Set target velocity and lateral position
Reference Application: Double Lange Change (Driver)

PI controller sets throttle / brake command

Predictive driver model sets steering wheel angle command
- Simple open-loop commands, by default
- Customize the subsystem as needed:
  - Maneuver-specific inputs
  - Closed-loop feedback
  - Test data playback
Reference Application: Double Lange Change (Controllers)

- Basic controllers provided for engine, transmission and brakes
- Incorporate your own variants, as needed
Reference Application: Double Lange Change (Plant)

- Use default plant model provided
- Select variants of interest
- Customize subsystems

---

- Engine
- Transmission
- Driveline
- Steering
- Suspension
- Vehicle body
- Tire

- TrqCmd
- EngSpd
- EngTrq
- Suspension
- Vehicle
- Wheels
- Tires
- Brake
- Propeller
- Propulsion
- Drive
- Gear
- Motor
- Shaft
Reference Application: Double Lange Change (Visualization)

- Scopes, gauges, plotters, logs
- 3D engine interface
Ride and Handling Study: Double Lane Change at 30 mph
Ride and Handling Study: Double Lane Change at 50 mph
Ride and Handling

- Analyze ride and handling metric of interest
  - Lateral acceleration
  - Roll-over propensity
  - Understeer / oversteer

- Simulate the vehicle over various driving maneuvers
  - Double lane change
  - Slowly increasing steering
  - Swept sine steering
  - Customer maneuver
Agenda

- Product overview
  - Powertrain Blockset
  - Vehicle Dynamics Blockset

- Six Cool Things You Can Do
  - Engine modeling and calibration
  - Cooling circuit modeling
  - Aftertreatment modeling
  - Ride and handling analysis
  - **Chassis controls development**
  - Perform closed-loop testing
  - ADAS / AD testing
Chassis Controls Study: Braking Test

Open loop brake controller simply passes through brake pressure command

Disc brakes
Chassis Controls Study: Braking Test
Added custom MPC variant to brake controller subsystem
At each time step, finds optimal brake pressure for target slip ratio
Chassis Controls Study: ABS Controller

Vehicle Speed

- Open-loop brakes
- MPC-based ABS

Tire lock-up

Ideal slip ratio
Chassis Controls Study: Braking Test with ABS
Split Mu Test
Chassis Controls Study: Split Mu Test
Chassis Controls Development

- Study the impact of controller on vehicle behavior
- Incorporate custom control features
- Test the closed-loop system over a wide range of scenarios
Agenda

- Product overview
  - Powertrain Blockset
  - Vehicle Dynamics Blockset

- Six Cool Things You Can Do
  - Engine modeling and calibration
  - Cooling circuit modeling
  - Aftertreatment modeling
  - Ride and handling analysis
  - Chassis controls development
  - ADAS / AD testing
  - Test in a virtual 3D environment
ADAS / AD Testing: Virtual 3D Scene

Camera sensor sends video to Simulink

Synthetic video used for testing vision-based algorithms (e.g., lane detection)
Stop Sign Detection and Braking
Customizing Scene with Support Package

- Create your own scenes with Unreal Editor and our Simulink plug-in
- Unreal Editor project files available in our Support Package: “Vehicle Dynamics Blockset interface for Unreal Engine 4”
Editing Support Package Scene to Add Stop Sign
Training Stop Sign Detector

- Train a stop sign detector as an ACF object detector

- The detector is trained based on the CVST example and saved as a MAT-file
Implementing Detector as System Object

- Implement the stop sign detector as a system object in Simulink
- Simulate using “Interpreted execution” mode

```matlab
classdef StopSignDetector < matlab.System & ...
    matlab.system.mixin.Propagates
    % Stop Sign Detector
    % Pre-computed constants
    properties (Access = private)
        detector
    end

    methods (Access = protected)
        function setupImpl(obj)
            d = load('acfDetector');
            obj.detector = d.acfDetector;
        end

        function [bbox,score]  = stepImpl(obj,img)
            [bboxes,scores] = detect(obj.detector,img);
            if ~isempty(bboxes)
                [score,idx] = max(scores);
                bbox = bboxes(idx,:);
            else
                bbox = nan(1,4);
                score = nan(1,1);
            end
        end
    end
```
Implementing Braking Logic

- Start with Scene Interrogation reference application
- Add braking logic to stop when the stop sign appears

Add switching logic

Add stop sign detector as MATLAB System Object
Changing the Lighting to Night Conditions
ADAS / AD Testing

- Use Unreal Engine as a virtual test environment for ADAS / AD control features
- Incorporate and test custom sensor models
- Create custom scenes for exercising the system
Value Proposition

- MathWorks provides vertical products to serve automotive industry, including
  - **Powertrain Blockset**: powertrain controls, fuel economy and performance simulation
  - **Vehicle Dynamics Blockset**: ride and handling, chassis controls, AD / ADAS testing

- These products offer
  - **Open** and documented library of component and subsystem models
  - Prebuilt vehicle models that you can parameterize and **customize**
  - **Fast**-running models that are ready for HIL deployment
  - **Framework** that supports integration with 3rd party software
Thank You

Mike Sasena, PhD
Product Manager
mike.sasena@mathworks.com