Key Takeaways

- MathWorks has Virtual Vehicle Simulation capabilities:
  - Rapidly assess electrified powertrain variants using Powertrain Blockset and Simulink
  - Integrate functionality with other tools using new Simulink features
  - Accelerate development of advanced powertrain and ADAS controllers using Vehicle Dynamics Blockset
Electrified Powertrain Selection Case Study

- Evaluate 3 motor Battery EV powertrain
  - What are the best gear ratios to use for the front / rear axles?

- Minimize:
  - Energy consumption (multiple drive cycles)
  - Acceleration time ($t_{0-60mph}$)

- Subject to constraints:
  - Operating limits for motor and battery
  - Velocity within 2 mph window of drive cycle target velocity
Electrified Powertrain Selection Case Study

- **Challenges**
  - Need system level model of vehicle
    - Plant models
    - Controller models
  - Heterogenous modeling environment
    - Support for 3rd party simulation tools / legacy code

- **MathWorks has automotive focused design tools...**

- **Accurate for use case**
- **Fast for design optimization (> real-time)**
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*
Reference Applications

Virtual Vehicle Models

Conventional Vehicle Reference Application
Simulate a full vehicle model with an internal combustion engine, transmission, and associated powertrain control algorithms. Use Open Example

HEV Multimode Reference Application
Simulate a full multimode HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated Open Example

HEV Input Power-Split Reference Application
Simulate an input power-split HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated Open Example

HEV P0 Reference Application
Simulate a P0 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated Open Example

HEV P1 Reference Application
Simulate a P1 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated Open Example

HEV P2 Reference Application
Simulate a P2 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated Open Example

Virtual Engine Dynamometers

CI Engine Dynamometer Reference Application
Simulate a CI engine plant and controller connected to a dynamometer with an tailpipe emission analyzer. Use to calibrate. Open Example

SI Engine Dynamometer Reference Application
Simulate a SI engine plant and controller connected to a dynamometer with a tailpipe emission analyzer. Use to calibrate. Open Example
Powertrain Modeling

- Modify EV Virtual Vehicle Model
  - 3 Permanent Magnet Machines
  - Battery / DC-DC Converter
  - Front Differential / Rear Axle Gears

What if I have plant models from other tools?
An FYI on FMI / FMU Configurations

- Model Exchange (One solver)

- Co-simulation (Separate solvers)
  - Standalone FMU
  - Tool coupling
    (it is co-simulation in the traditional sense, both tools are needed during execution)
FMU Capabilities from MathWorks


<table>
<thead>
<tr>
<th>Name</th>
<th>License</th>
<th>Platforms</th>
<th>Co-Simulation</th>
<th>Model Exchange</th>
<th>Co-Simulation</th>
<th>Model Exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB® Simulink®</td>
<td>$</td>
<td>macOS, Windows, Linux</td>
<td>10</td>
<td>2.0</td>
<td>10</td>
<td>2.0</td>
</tr>
</tbody>
</table>

- Export a Model as a Tool-Coupling FMU

- Simulink Extras / FMU Import

![Diagram of FMU Import and Export]
What if I Have Controller Code from C / C++?

- **C Caller - “Just Call My Code”**
EV Energy Management Strategy (EMS)

- Instantaneous torque (or power) command to actuators (electric machines)

- Subject to constraints:

  \[ \tau_{\text{min}}(\omega) \leq \tau_{\text{act}} \leq \tau_{\text{max}}(\omega) \]
  \[ P_{\text{chg}}(SOC) \leq P_{\text{batt}} \leq P_{\text{disch}}(SOC) \]
  \[ I_{\text{chg}}(SOC) \leq I_{\text{batt}} \leq I_{\text{disch}}(SOC) \]

- Attempt to minimize energy consumption, maintain drivability

\[ T_{\text{demand}} = T_{\text{mot,f}} + T_{\text{mot,r}} \]
EV Energy Management Strategy (EMS) Process

Performed every controller time step

1. Create torque split vector

2. Check constraints, determine infeasible conditions

3. Calculate and minimize cost function (Battery Power)

\[\begin{align*}
\tau_{\text{min}}(\omega) & \leq \tau_{\text{act}} \leq \tau_{\text{max}}(\omega) \\
P_{\text{chg}}(SOC) & \leq P_{\text{batt}} \leq P_{\text{disch}}(SOC) \\
I_{\text{chg}}(SOC) & \leq I_{\text{batt}} \leq I_{\text{disch}}(SOC) \\
\tau_{\text{demand}} & = \tau_{\text{front}} + \tau_{\text{rear}} \\
\end{align*}\]
EV Energy Management Strategy (EMS) Process

Snapshot of EMS at a point in time

![Graph showing battery power vs. motor torque with infeasible regions marked.](image-url)
Optimizing Front and Rear Gear Ratios

- A pareto curve exists between energy usage and acceleration performance
- A cost function can be used to help determine the best set of ratios
- Higher weight towards system efficiency leads to lower over all gear ratios

\[
\min_{N_f, N_r} (0.55E_{FTP} + 0.45E_{HWY}) W_1 + W_2(T_{0-120KPH})
\]
Advanced Powertrain Control – Torque Vectoring

- Dual Rear Motors

- Improved handling
- Longitudinal and lateral vehicle dynamics models are needed to simulate torque vectoring → Need appropriate tool…
Vehicle Dynamics Blockset

Library of blocks

Game engine

Pre-built reference applications
Vehicle Dynamics Blockset

- 3D photo-realistic environment using the Unreal Engine (Epic Games)
- Pre-built scenes and vehicle types
Vehicle Dynamics Modeling

- Add components to Virtual Vehicle Model
  - 6 DOF Vehicle
    - 14-DOF
  - 2 DOF “Magic Tire” + Brakes
  - Suspension
  - Steering
Vehicle Dynamics Control – Torque Vectoring

Tire Slip Angle = \frac{(a+b)r}{a}

Average Steer Angle

PID(z) → TV Torque

A Torque Vectoring Strategy for Improving the Performance of a Rear Wheel Drive Electric Vehicle

Jyotishman Ghosh, Andrea Tonoli, Nicola Amati
Department of Mechanical and Aerospace Engineering
Politecnico di Torino
Turin, Italy
Email: jyotishman.ghosh@polito.it

Greater lateral acceleration with 8.7% less steering input

Longer linear tire slip angle region and 5.7% greater lateral acceleration
Vehicle Dynamics Control – Torque Vectoring

Steering = 45° Right
WOT
Red = TV On
Blue = TV Off
Driver-In-Loop Functionality

- Open-loop control w/ external steering wheel and pedals
  - Setup for Logitech G29 steering wheel / pedals
  - Could use any “joystick” device

- “One Pedal Driving” algorithm
  - Maps accelerator pedal to be used for both acceleration and regen braking
  - Zone calibration effects drivability behavior and “Fun To Drive” characteristics
Driver-In-Loop Functionality
Vehicle Dynamics can be used for ADAS Applications

- Example from University of Alabama EcoCAR team
  - David Barnes, Engineering Manager for UA team
  - Graduate summer intern for MathWorks
  - MathWorks Racing Lounge article
Vehicle Dynamics for ADAS Applications

- Perception System for SAE Level 2
  - Utilize Automated Driving Toolbox Driving Scenario Designer app
  - Configure vision and mid-range radar (MRR) sensors on ego vehicle
  - Red areas are blind spots
  - Integrated sensors into virtual vehicle model
Vehicle Dynamics for ADAS Applications

- Lane Keep Assist (LKA) and Adaptive Cruise Control (ACC)
  - 14DOF vehicle model
  - Model Predictive Controller (MPC) Toolbox
    LKA, ACC blocks
To Learn More about MathWorks Automated Driving Capabilities…

Design and Test of Automated Driving Algorithms
12:00 p.m.–12:30 p.m.

In this talk, you will learn how MathWorks helps you design and test automated driving algorithms, including:

• Perception: Design LiDAR, vision, radar, and sensor fusion algorithms with recorded and live data
• Planning: Visualize street maps, design path planners, and generate C/C++ code
• Controls: Design a model-predictive controller for traffic jam assist, test with synthetic scenes and sensors, and generate C/C++ code
• Deep learning: Label data, train networks, and generate GPU code
• Systems: Simulate perception and control algorithms, as well as integrate and test hardware
Model-Based Design == Model Reuse
Key Takeaways

- MathWorks has Virtual Vehicle Simulation capabilities:
  - Rapidly assess electrified powertrain variants using Powertrain Blockset and Simulink
  - Integrate functionality with other tools using new Simulink features
  - Accelerate development of advanced powertrain and ADAS controllers using Vehicle Dynamics Blockset
Q & A

Please follow up with us if you have any interest in the material presented
MATLAB EXPO 2019

Thank You!

Kevin Oshiro
Principal Application Engineer
koshiro@mathworks.com