MATLAB EXPO 2019

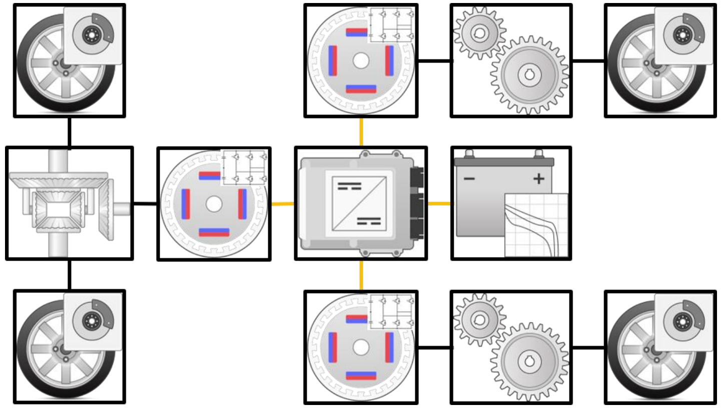
Virtual Vehicle Simulation

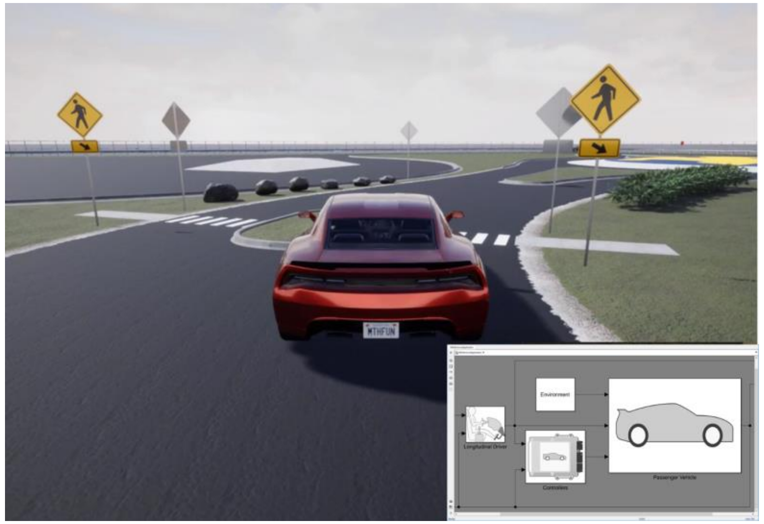
Electrified Powertrain Design

Kevin Oshiro
MathWorks Application Engineering

Key Takeaways

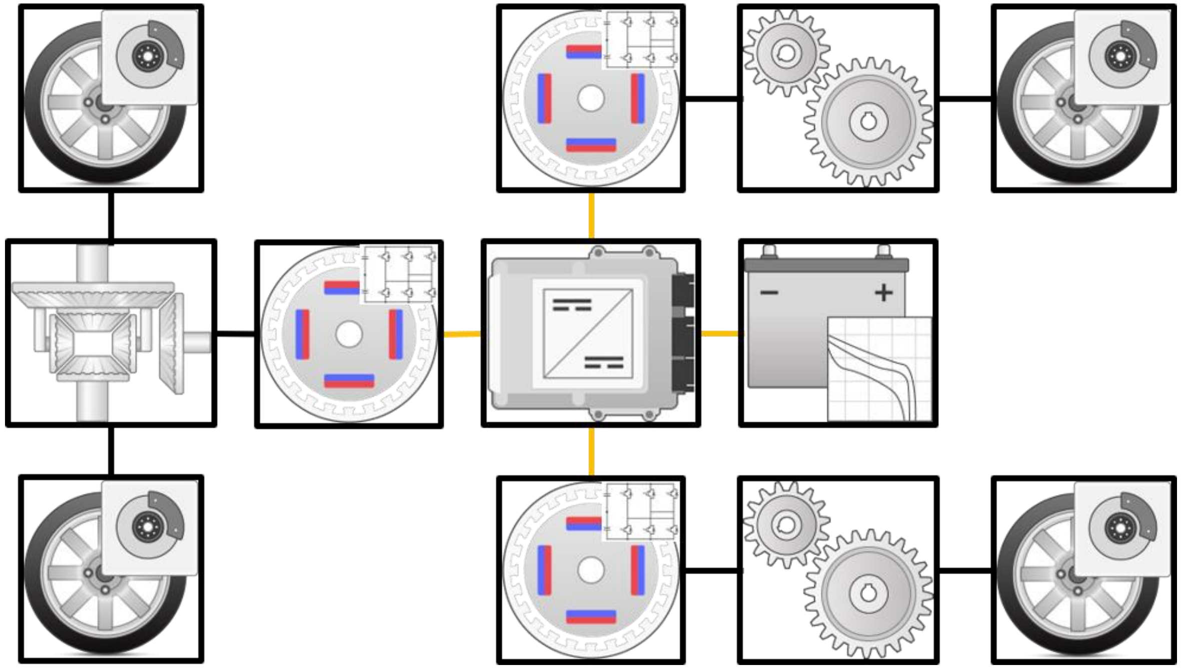
- MathWorks has Virtual Vehicle Simulation capabilities:
  - Rapidly assess electrified powertrain variants using Powertrain Blockset and Simulink
  - Integrate functionality with other tools using new Simulink features
  - Accelerate development of advanced powertrain and ADAS controllers using Vehicle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





Electrified Powertrain Selection Case Study

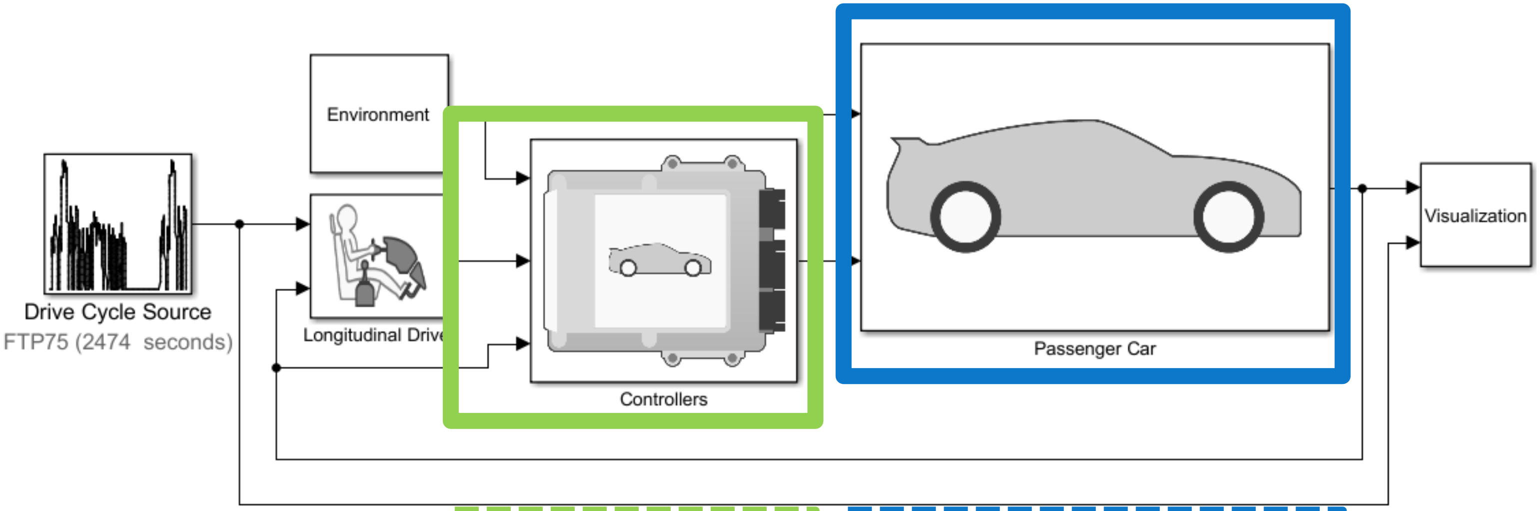
- Evaluate 3 motor Battery EV powertrain
  - What are the best gear ratios to use for the front / rear axles?
- Minimize:
  - Energy consumption (multiple drive cycles)
  - Acceleration time ($t_{0\text{-}60mph}$)
- Subject to constraints:
  - Operating limits for motor and battery
  - Velocity within 2 mph window of drive cycle target velocity



Electrified Powertrain Selection Case Study

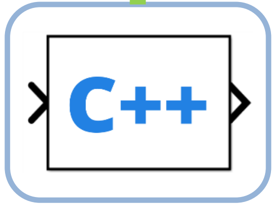
- Challenges
  - Need system level model of vehicle
    - Plant models
    - Controller models
  - Heterogenous modeling environment
    - Support for 3rd party simulation tools / legacy code

- MathWorks has automotive focused design tools…



- Supervisory control
- Energy management

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

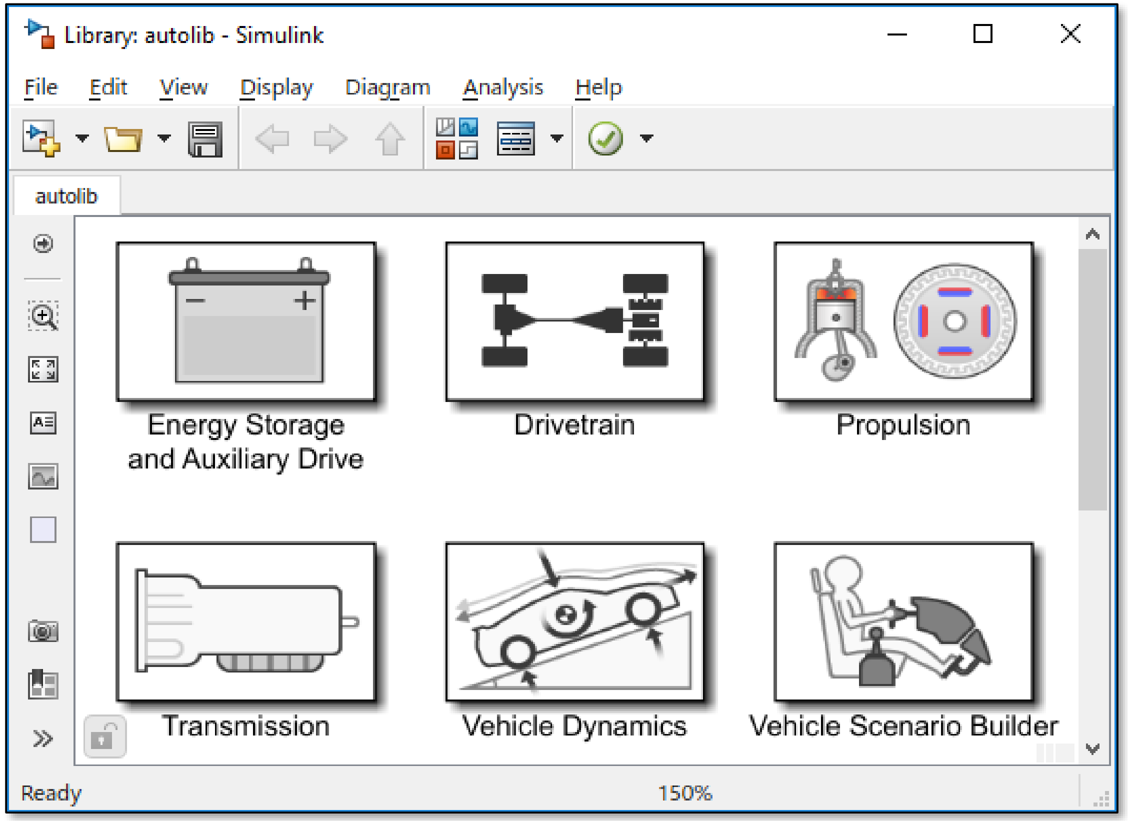


Powertrain Blockset

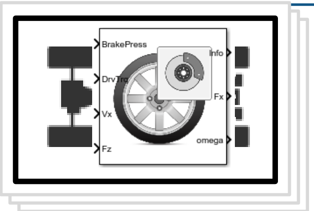
- Goals:
  - Provide starting point for engineers to build **good plant / controller models**
  - Provide **open** and documented models
  - Provide very **fast**-running models that work with popular HIL systems

R2016b

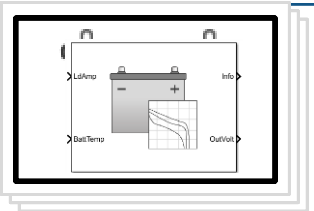
Library of blocks



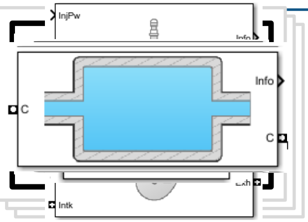
Lower the barrier to entry for Model-Based Design



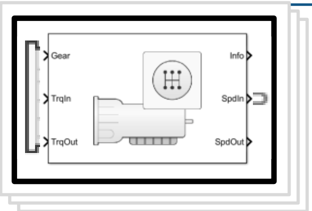
Drivetrain



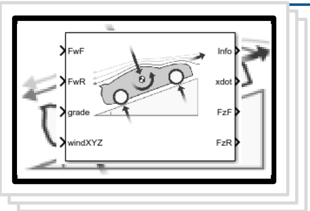
Energy Storage
and Auxiliary Drive



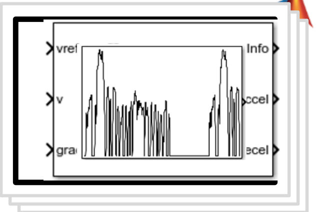
Propulsion



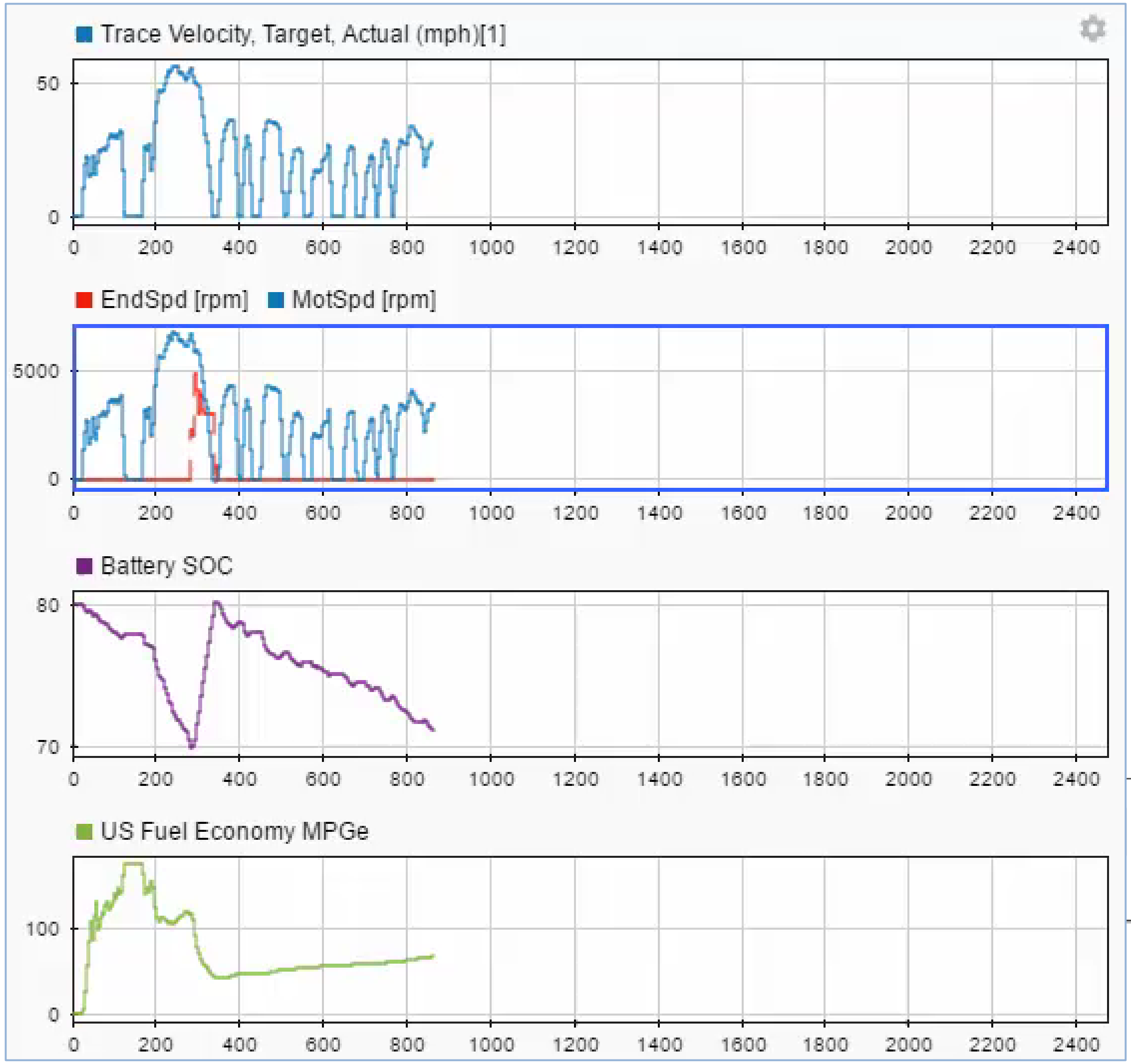
Transmission

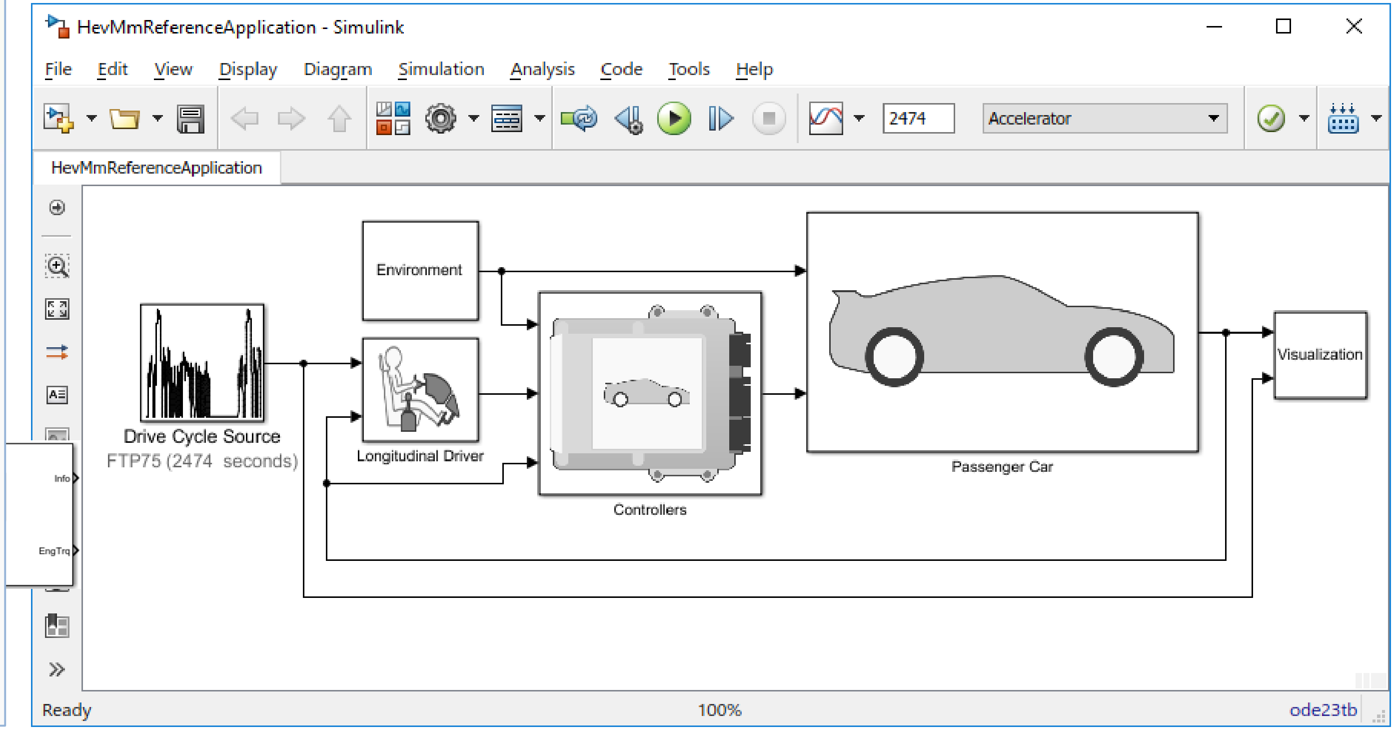


Vehicle Dynamics



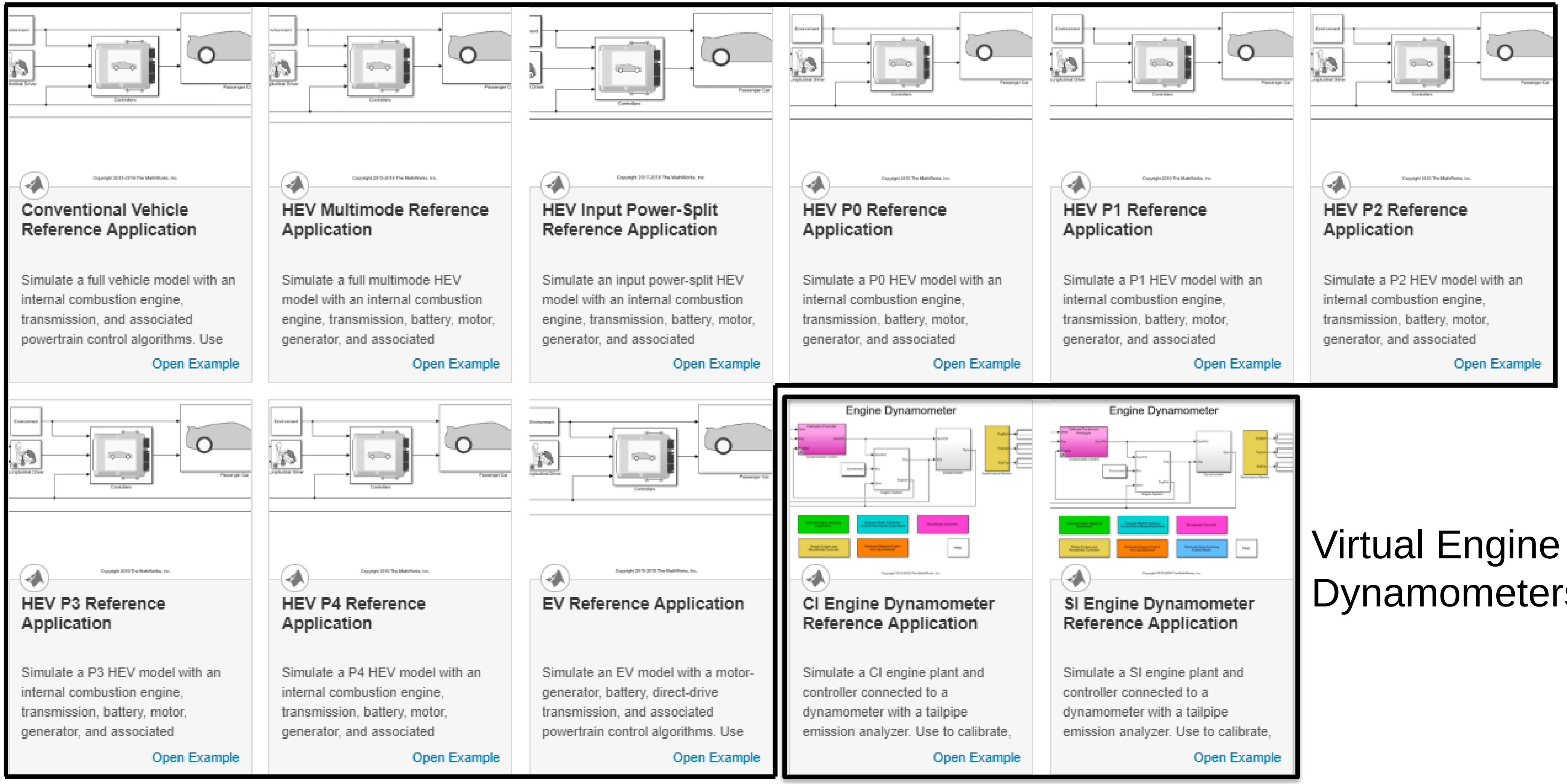
Vehicle Scenario Builder





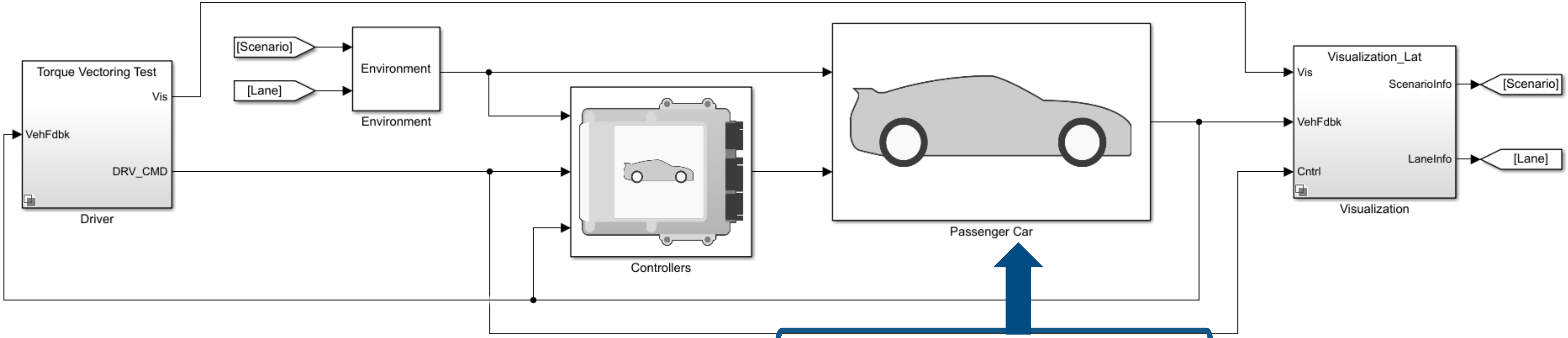
Reference Applications

Virtual Vehicle Models

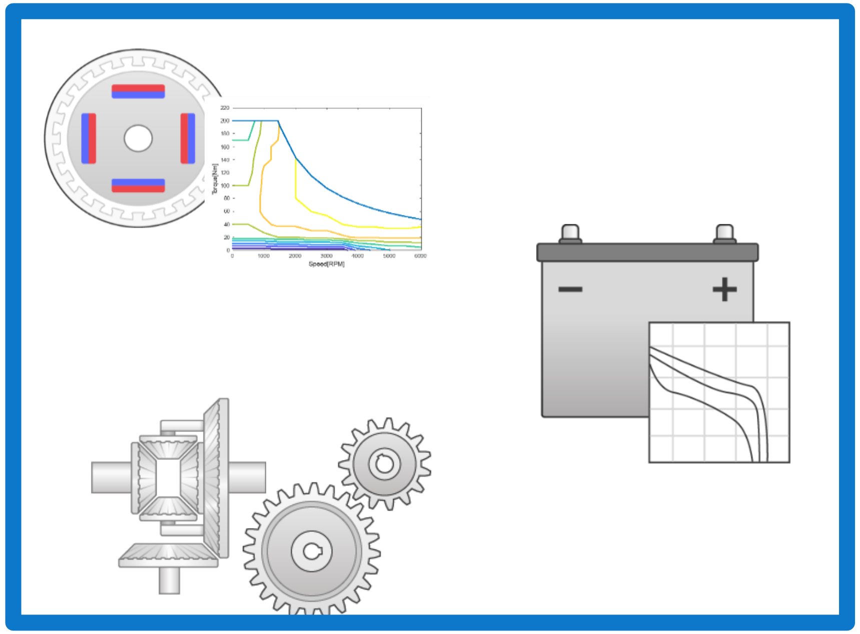


Virtual Engine Dynamometers

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?

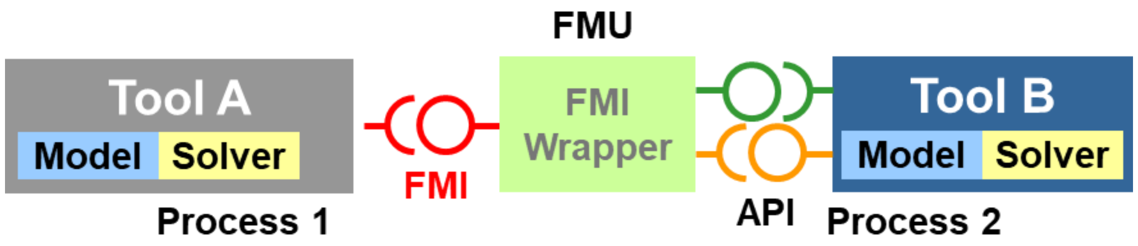
An FYI on FMI / FMU Configurations

- Model Exchange (One solver)

- Co-simulation (Separate solvers)
  - Standalone FMU

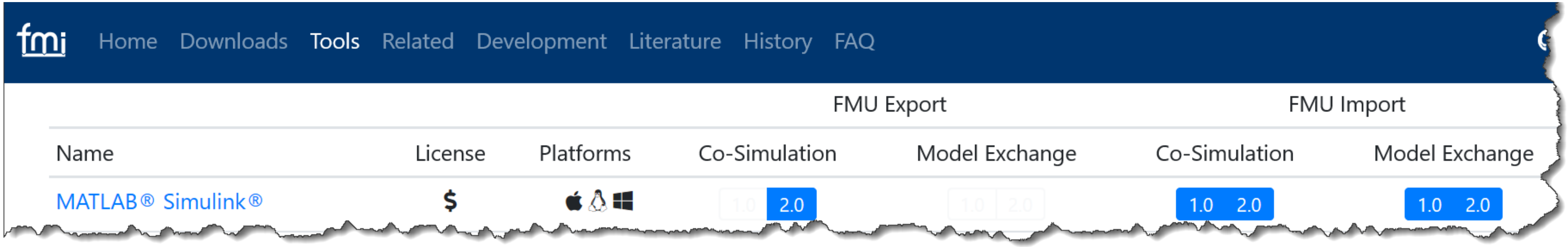
  - Tool coupling

(it is co-simulation in the traditional sense, both tools are needed during execution)

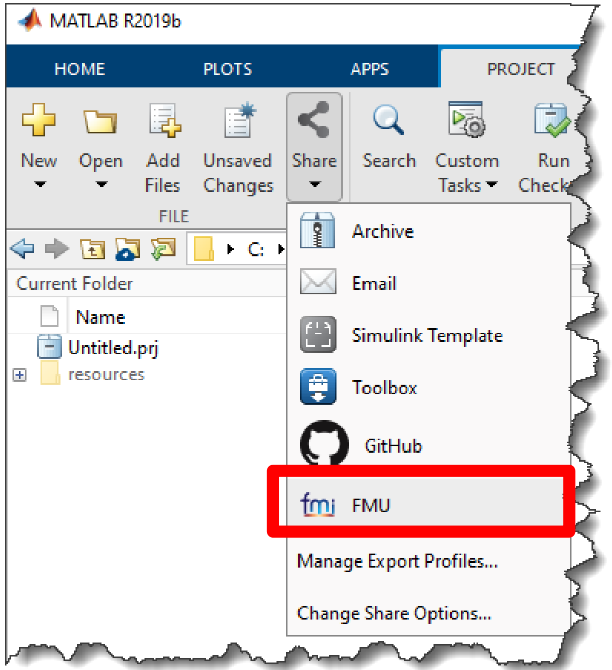


FMU Capabilities from MathWorks

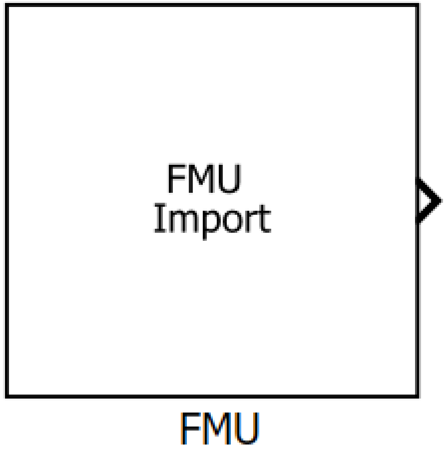
- https://fmi-standard.org/tools/



- Export a Model as a Tool-Coupling FMU



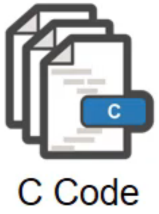
- Simulink Extras / FMU Import



What if I Have Controller Code from C / C++?

R2018b

- C Caller - “Just Call My Code”



EV Energy Management Strategy (EMS)

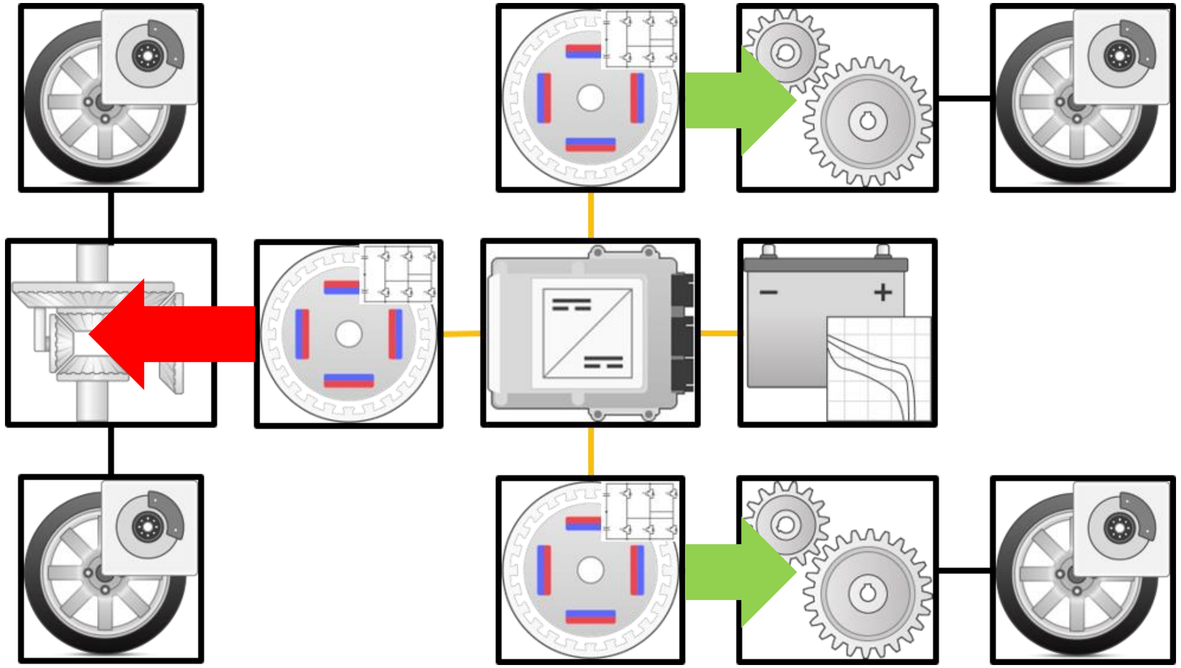
- Instantaneous torque (or power) command to actuators (electric machines)

- Subject to constraints:

$$\tau_{min}(\omega) \leq \tau_{act} \leq \tau_{max}(\omega)$$
$$P_{chg}(SOC) \leq P_{batt} \leq P_{dischg}(SOC)$$
$$I_{chg}(SOC) \leq I_{batt} \leq I_{dischg}(SOC)$$

- Attempt to minimize energy consumption, maintain drivability

$$T_{demand} = T_{mot,f} + T_{mot,r}$$



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)

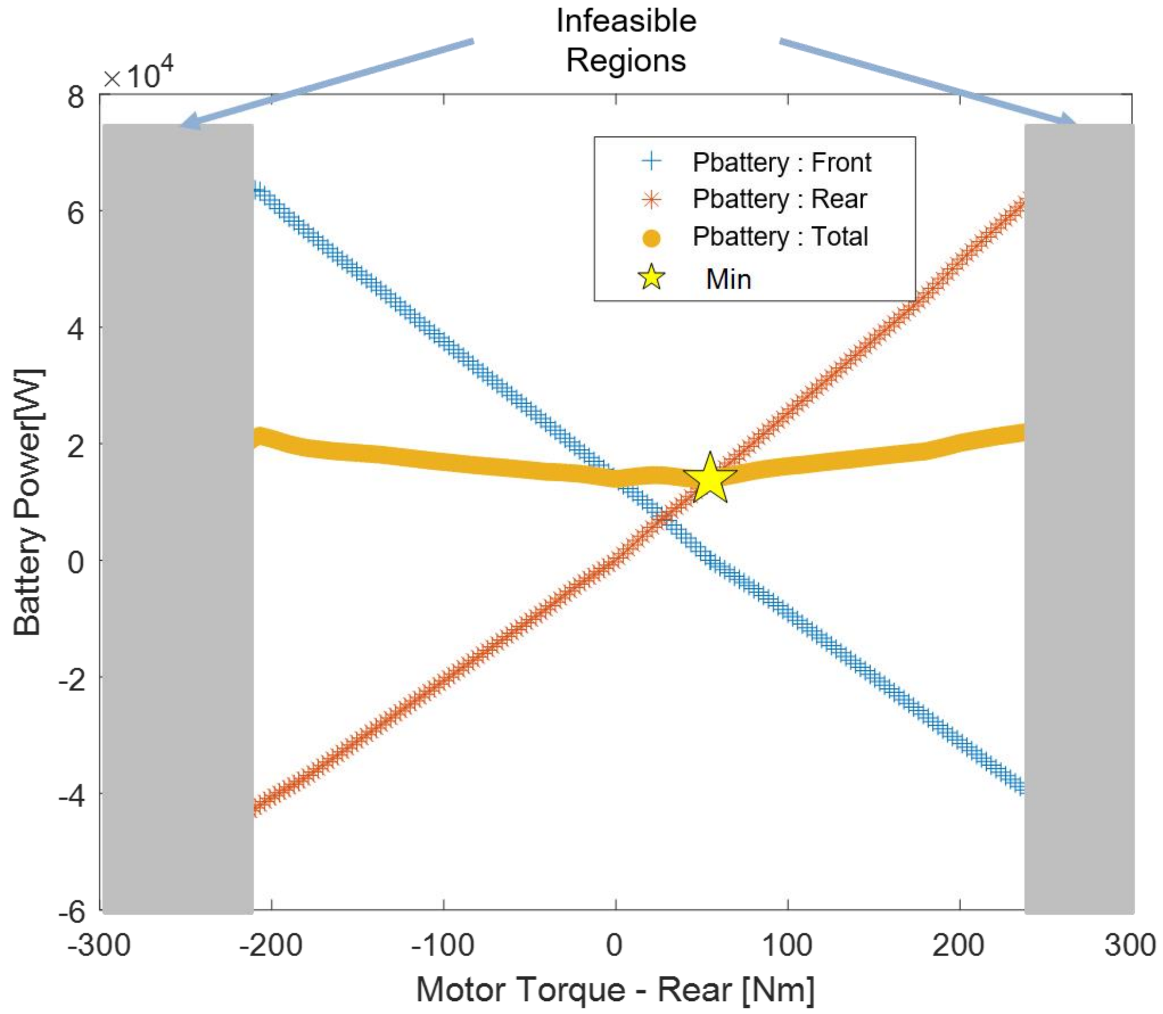
$$\begin{bmatrix} -Min\ Rear\ Torque \\ \vdots \\ +Max\ Rear\ Torque \end{bmatrix}$$

$$\tau_{min}(\omega) \le \tau_{act} \le \tau_{max}(\omega)$$
$$P_{chg}(SOC) \le P_{batt} \le P_{dischg}(SOC)$$
$$I_{chg}(SOC) \le I_{batt} \le I_{dischg}(SOC)$$
$$\tau_{demand} = \tau_{front} + \tau_{rear}$$

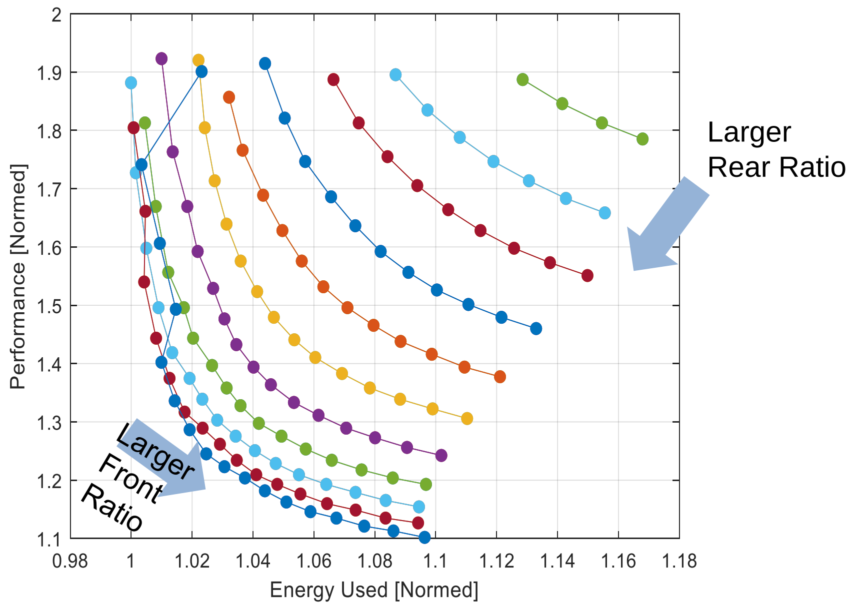
$$\min_{\tau_{rear}} P_b(\tau_{rear})$$

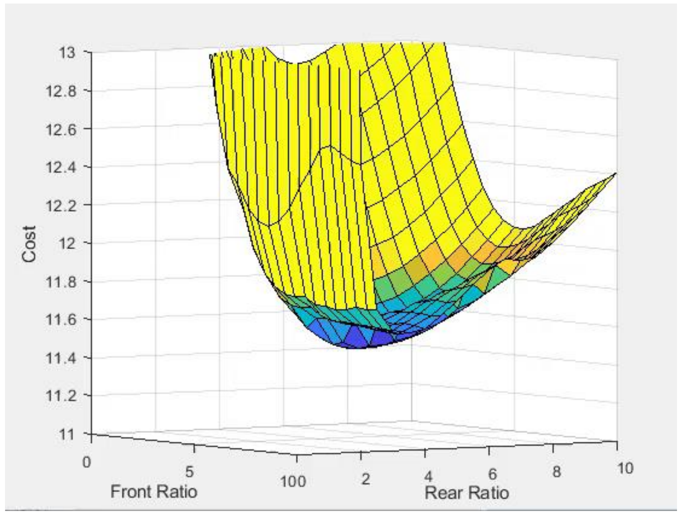
EV Energy Management Strategy (EMS) Process

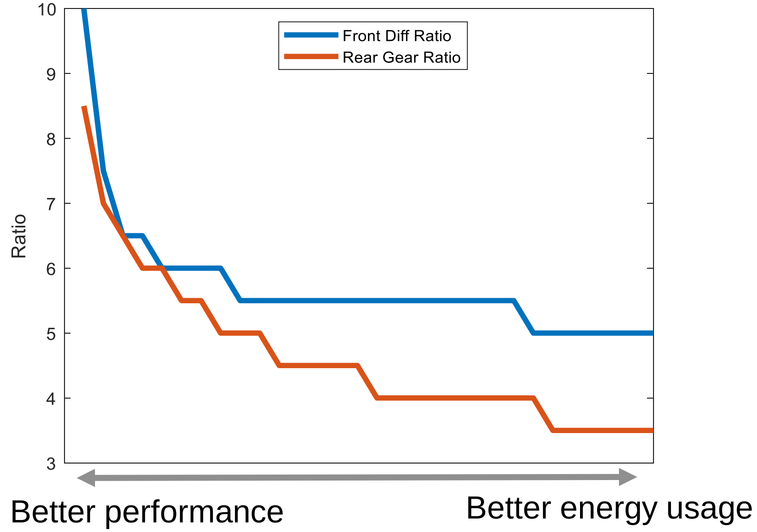
Snapshot of EMS at a point in time



Optimizing Front and Rear Gear Ratios





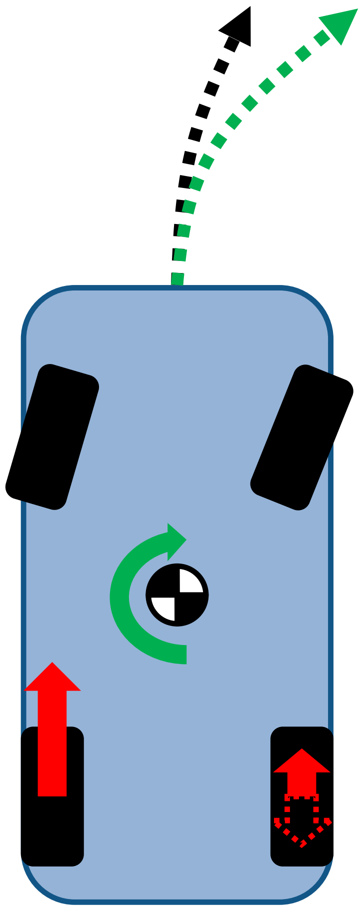


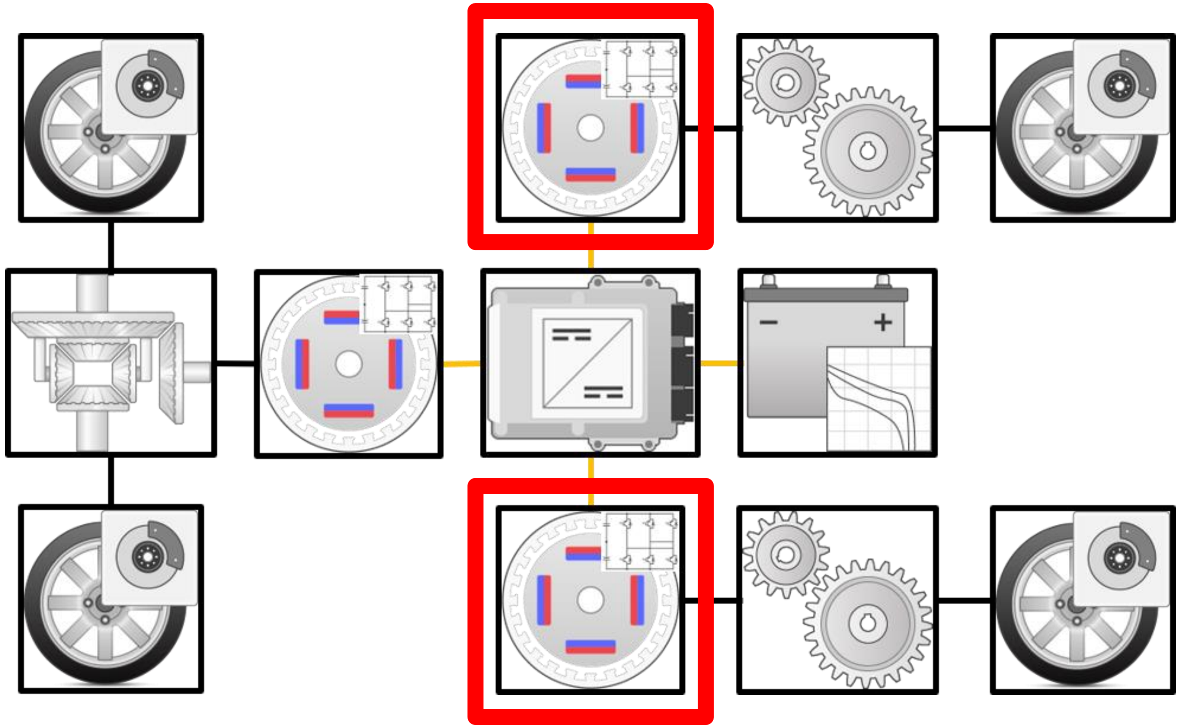
$$\min_{N_f, N_r} (0.55E_{FTP} + 0.45E_{HWY})\, W_1 + W_2(T_{0-120KPH})$$

- 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

Advanced Powertrain Control – Torque Vectoring

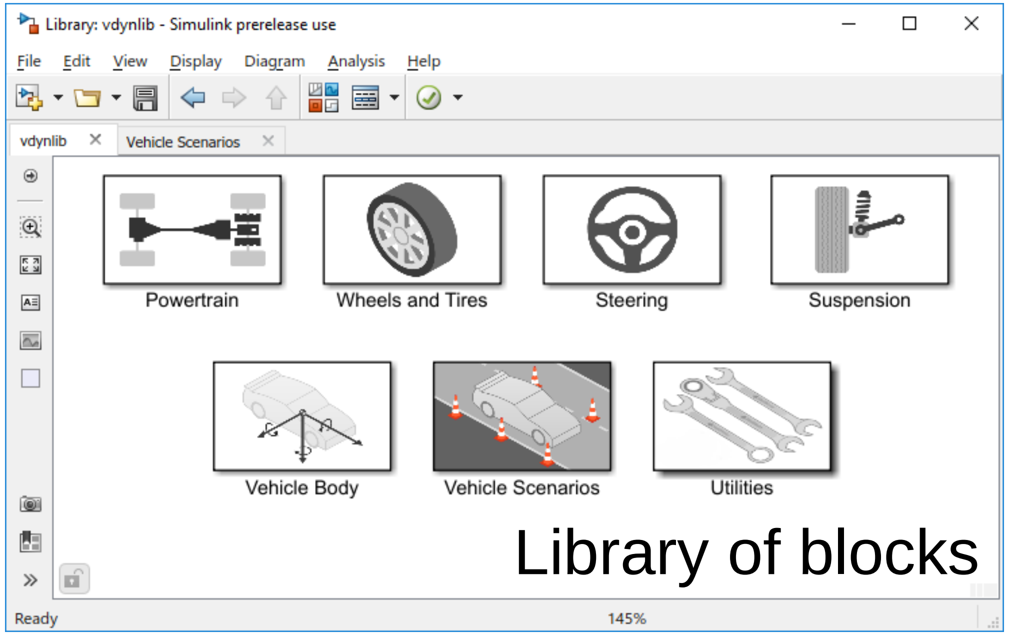
- Dual Rear Motors

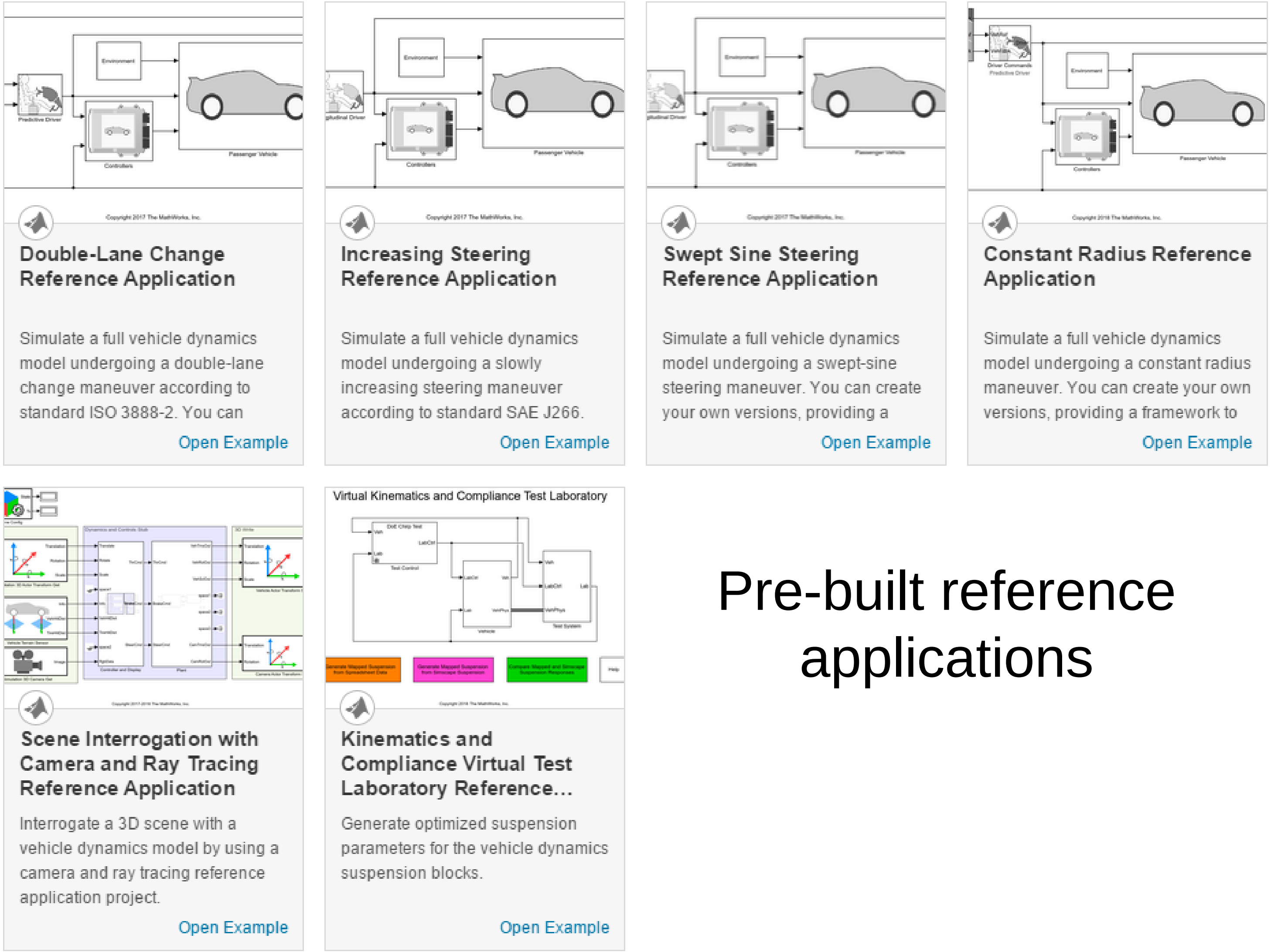




- Improved handling
- Longitudinal <u>and</u> lateral vehicle dynamics models are needed to simulate torque vectoring
→ Need appropriate tool…

Vehicle Dynamics Blockset R2018a



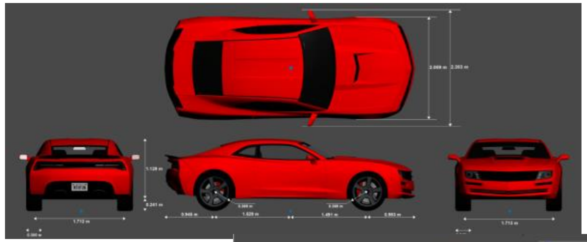


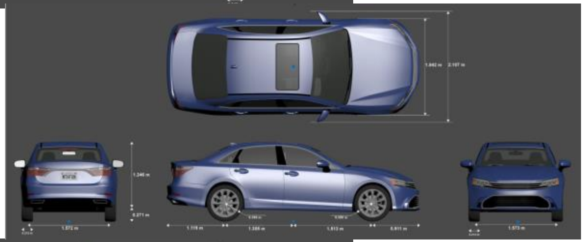
Pre-built reference applications

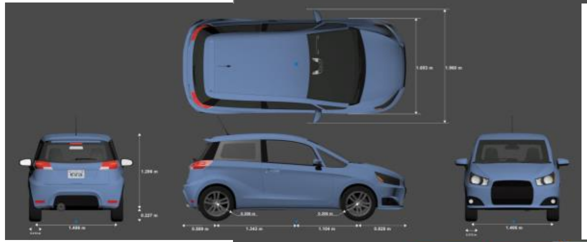
Vehicle Dynamics Blockset

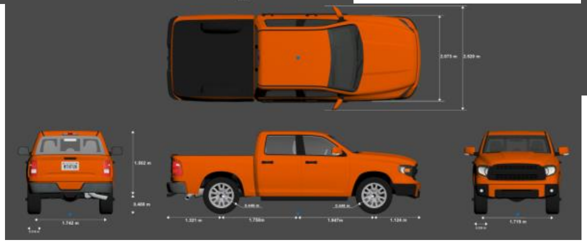
- 3D photo-realistic environment using the Unreal Engine (Epic Games)
- Pre-built scenes and vehicle types

UNREAL ENGINE

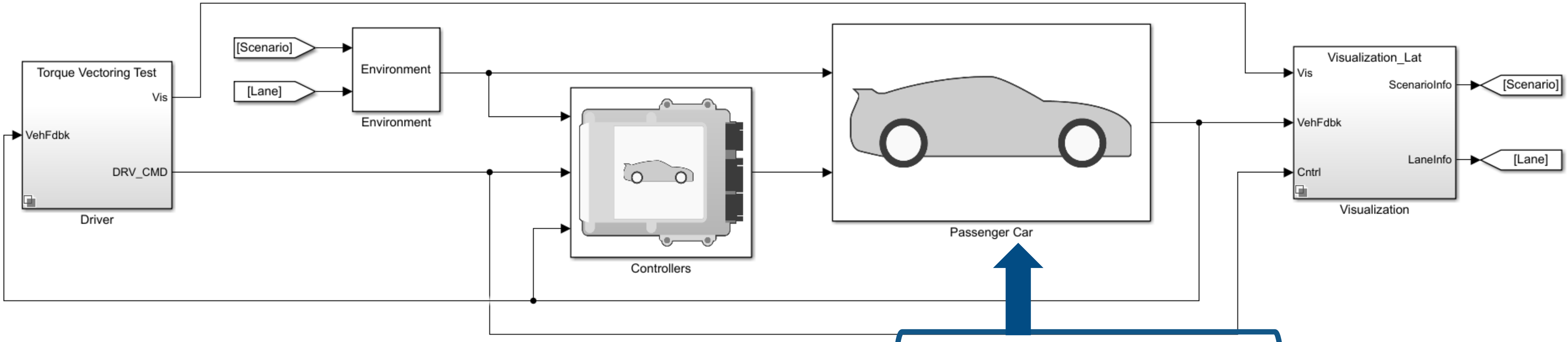








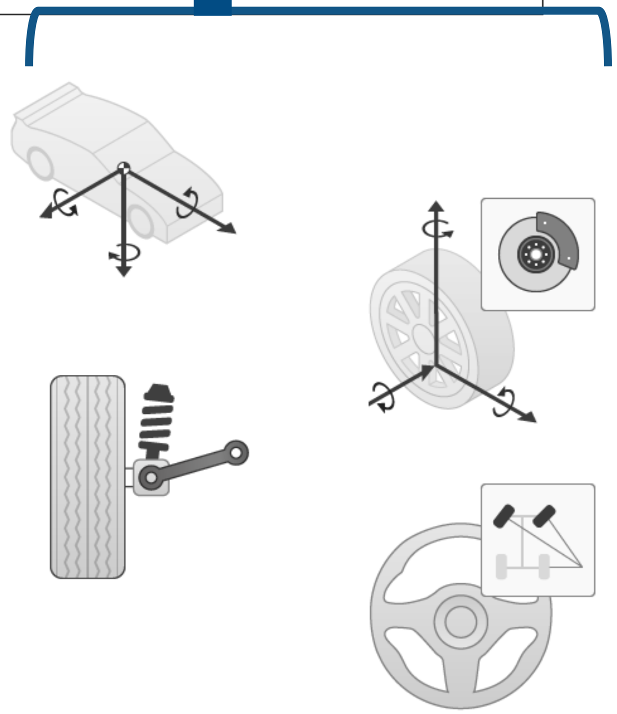
Vehicle Dynamics Modeling



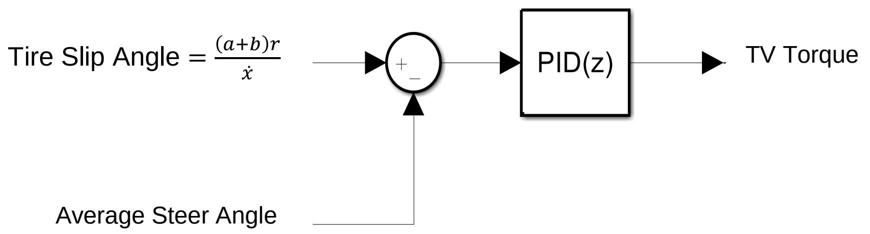
- Add components to Virtual Vehicle Model
  - 6 DOF Vehicle
  - 2 DOF "Magic Tire" + Brakes

  14-DOF

  - Suspension
  - Steering

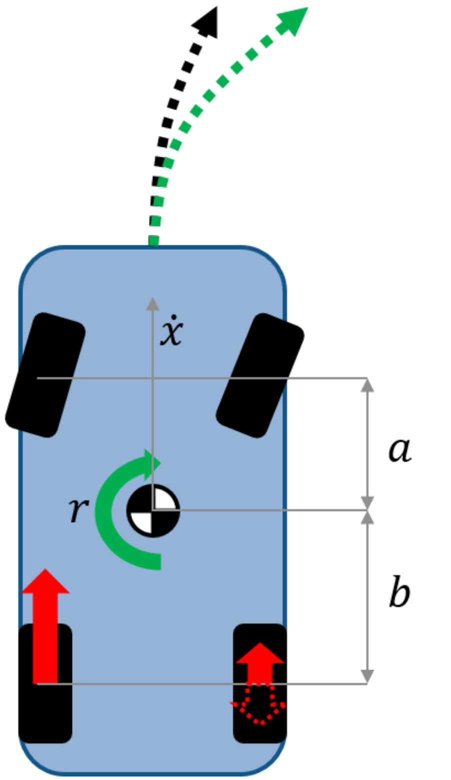


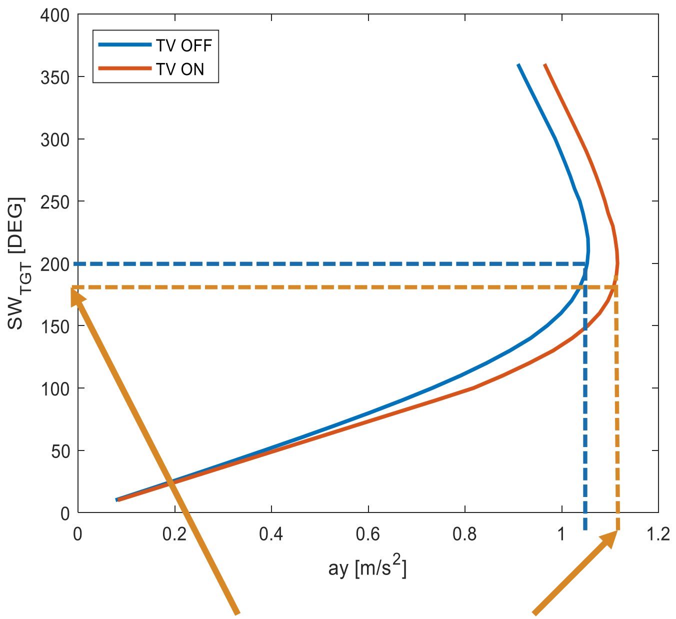
Vehicle Dynamics Control – Torque Vectoring



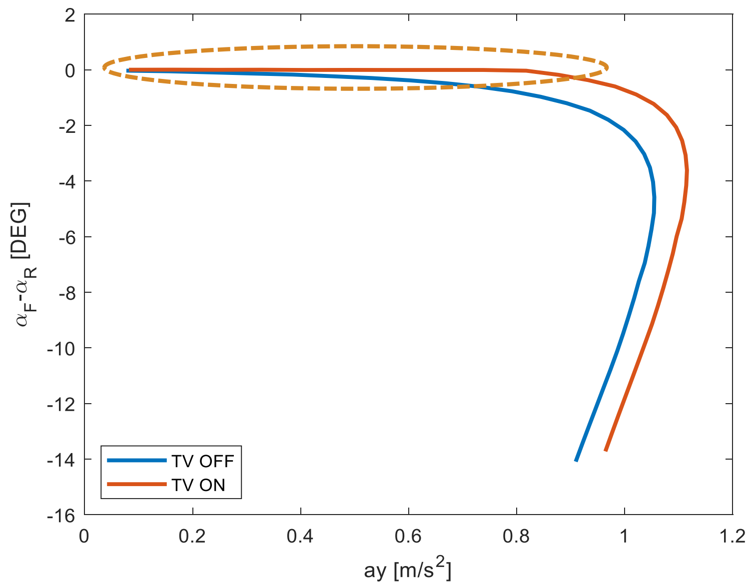
A Torque Vectoring Strategy for Improving the Performance of a Rear Wheel Drive Electric Vehicle

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



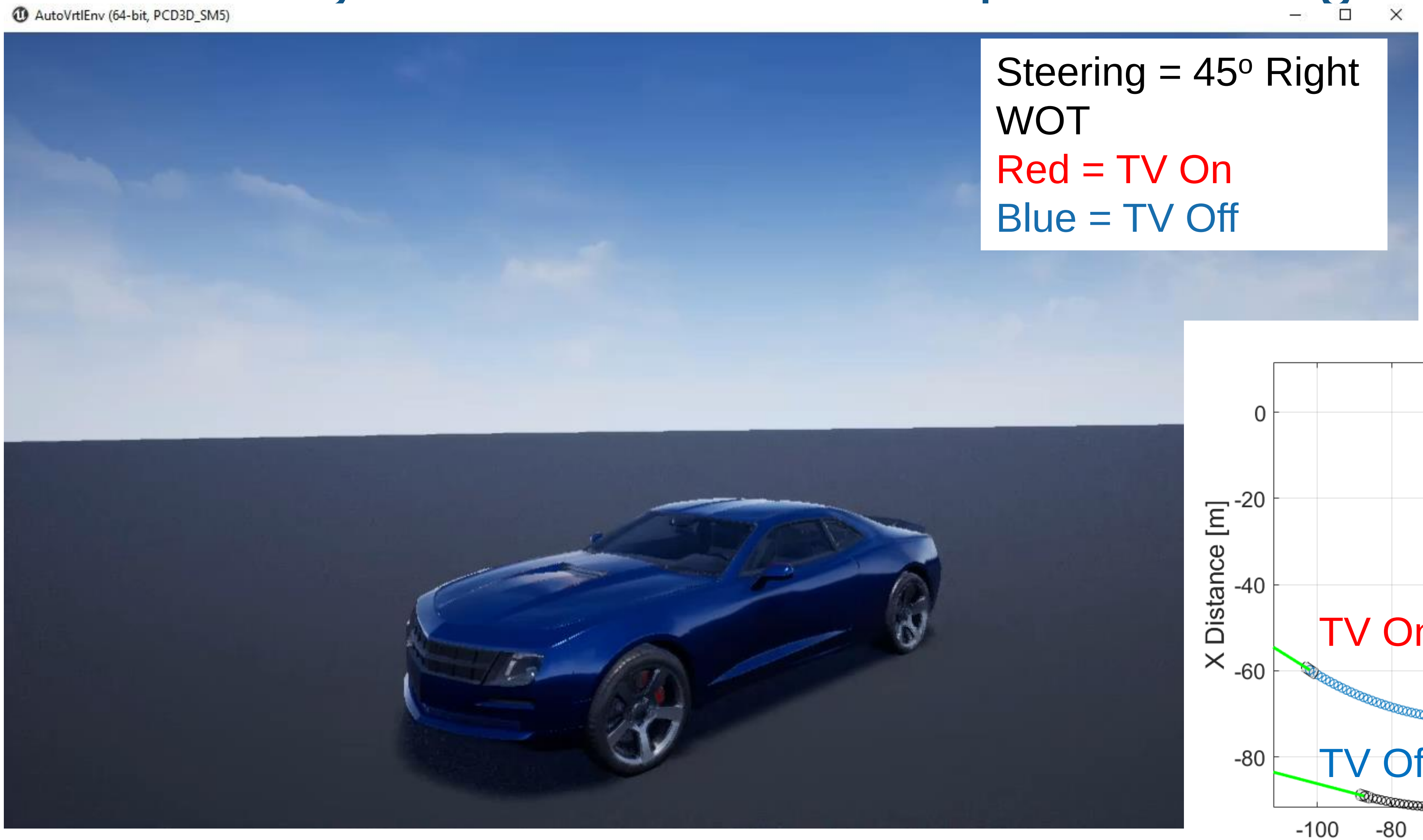


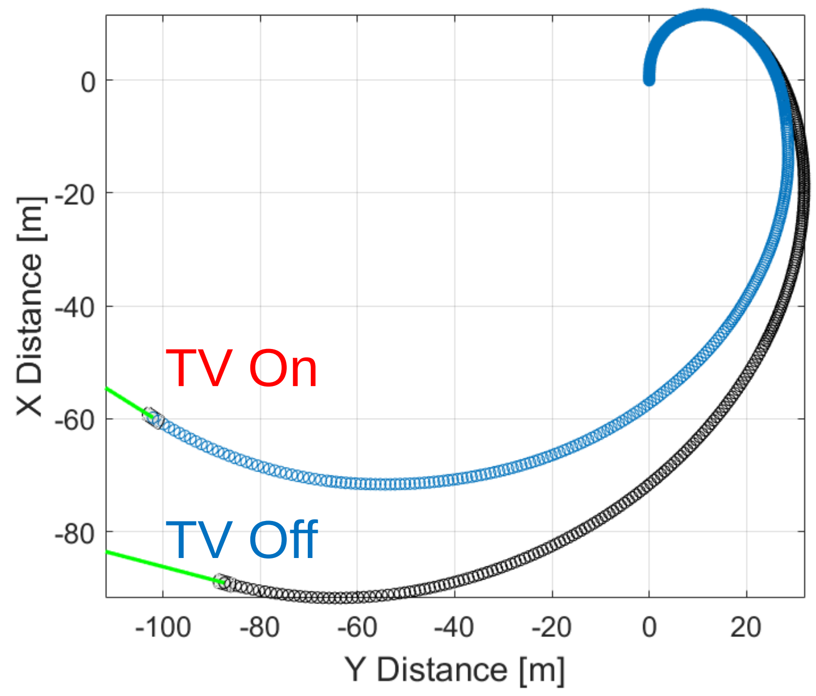
Greater lateral acceleration with 8.7% less steering input



Longer linear tire slip angle region and 5.7% greater lateral acceleration

Vehicle Dynamics Control – Torque Vectoring

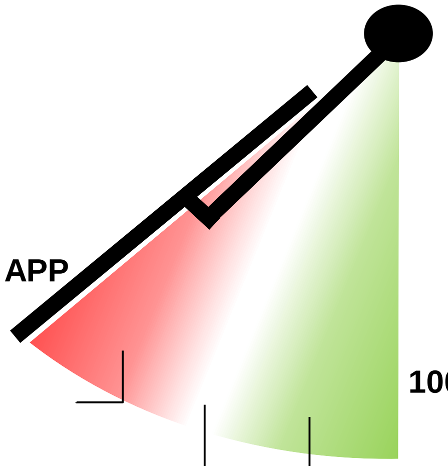


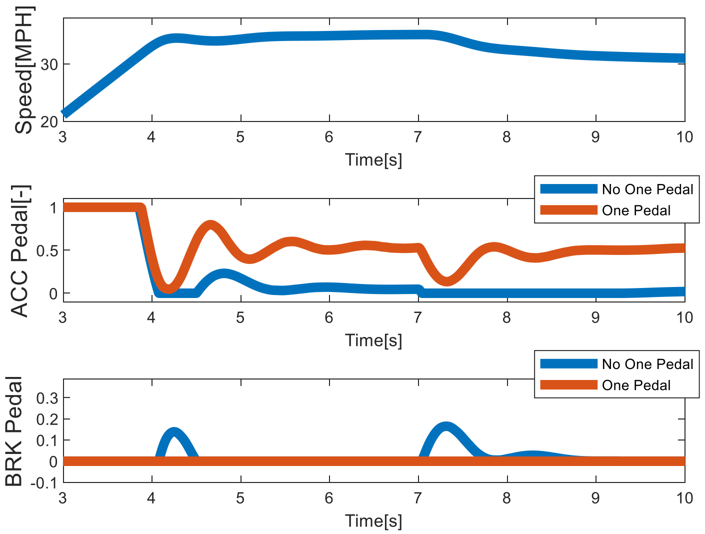


Driver-In-Loop Functionality

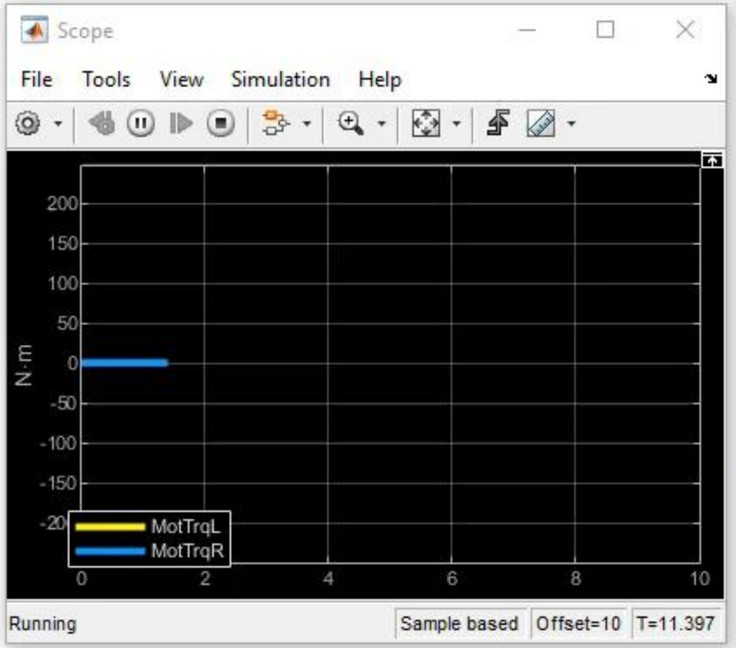
- Open-loop control w/ external steering wheel and pedals
  - Setup for Logitech G29steering wheel / pedals
  - Could use any “joystick” device

- “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



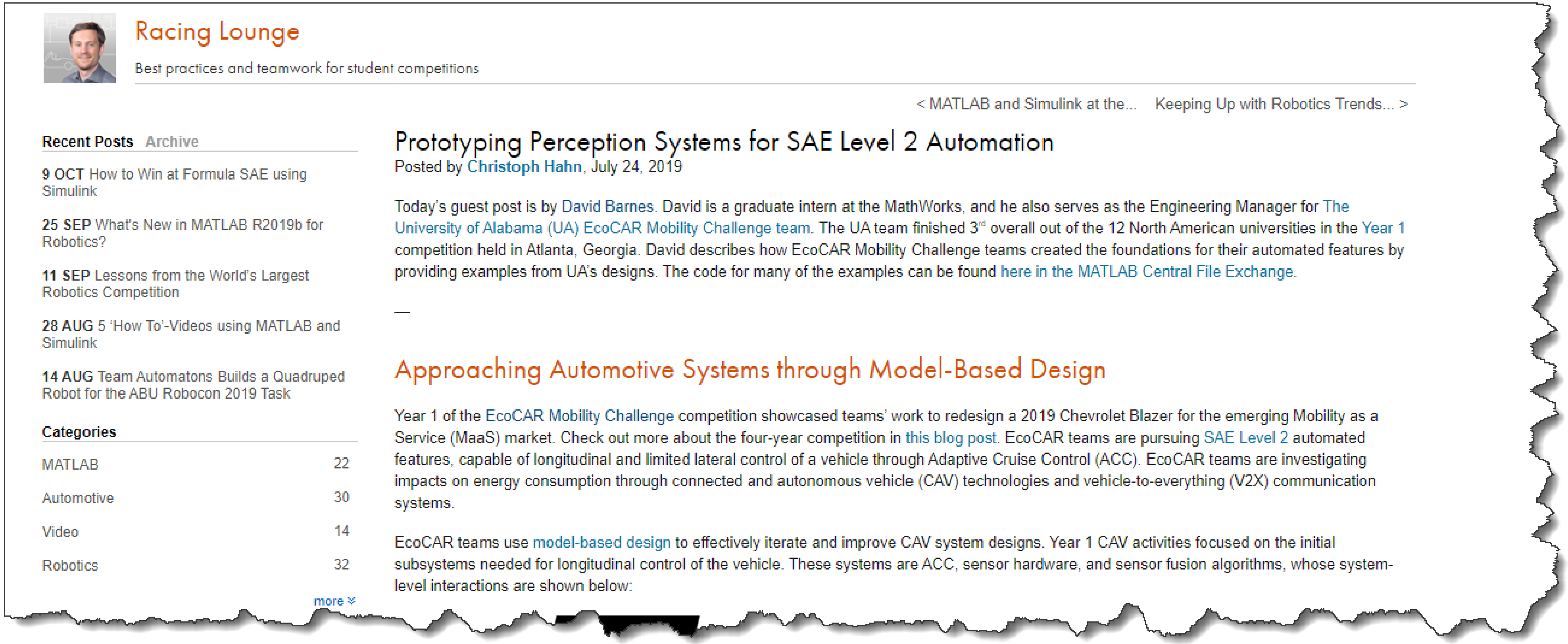


Driver-In-Loop Functionality



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



Racing Lounge

Best practices and teamwork for student competitions

< MATLAB and Simulink at the... Keeping Up with Robotics Trends... >

Recent Posts Archive

9 OCT How to Win at Formula SAE using Simulink

25 SEP What's New in MATLAB R2019b for Robotics?

11 SEP Lessons from the World's Largest Robotics Competition

28 AUG 5 'How To'-Videos using MATLAB and Simulink

14 AUG Team Automatons Builds a Quadruped Robot for the ABU Robocon 2019 Task

Categories

MATLAB	22
Automotive	30
Video	14
Robotics	32

more

Prototyping Perception Systems for SAE Level 2 Automation

Posted by Christoph Hahn, July 24, 2019

Today's guest post is by David Barnes. David is a graduate intern at the MathWorks, and he also serves as the Engineering Manager for The University of Alabama (UA) EcoCAR Mobility Challenge team. The UA team finished 3rd overall out of the 12 North American universities in the Year 1 competition held in Atlanta, Georgia. David describes how EcoCAR Mobility Challenge teams created the foundations for their automated features by providing examples from UA's designs. The code for many of the examples can be found here in the MATLAB Central File Exchange.

—

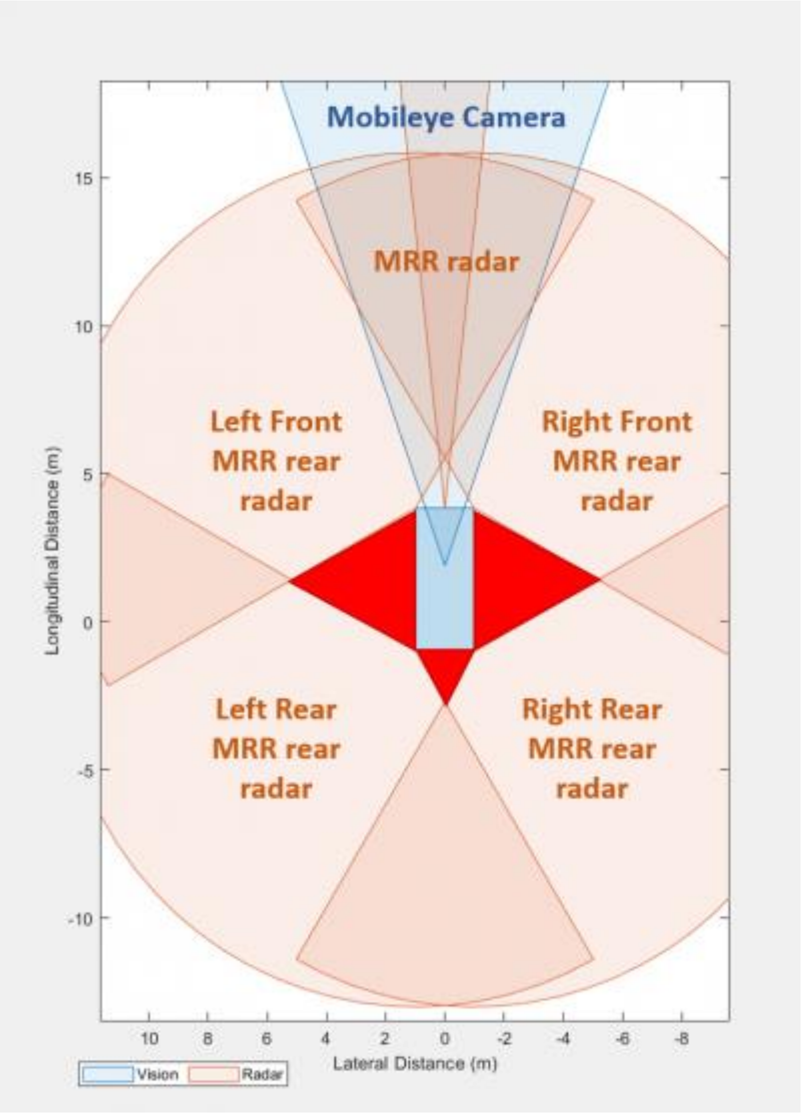
Approaching Automotive Systems through Model-Based Design

Year 1 of the EcoCAR Mobility Challenge competition showcased teams' work to redesign a 2019 Chevrolet Blazer for the emerging Mobility as a Service (MaaS) market. Check out more about the four-year competition in this blog post. EcoCAR teams are pursuing SAE Level 2 automated features, capable of longitudinal and limited lateral control of a vehicle through Adaptive Cruise Control (ACC). EcoCAR teams are investigating impacts on energy consumption through connected and autonomous vehicle (CAV) technologies and vehicle-to-everything (V2X) communication systems.

EcoCAR teams use model-based design to effectively iterate and improve CAV system designs. Year 1 CAV activities focused on the initial subsystems needed for longitudinal control of the vehicle. These systems are ACC, sensor hardware, and sensor fusion algorithms, whose system-level interactions are shown below:

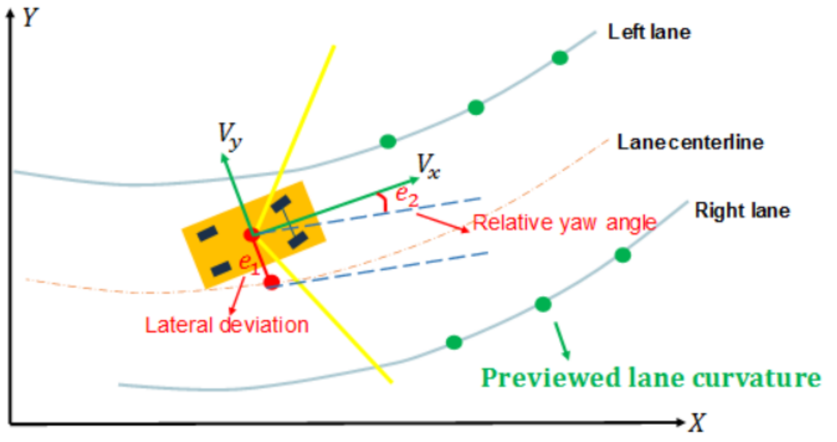
Vehicle Dynamics for ADAS Applications

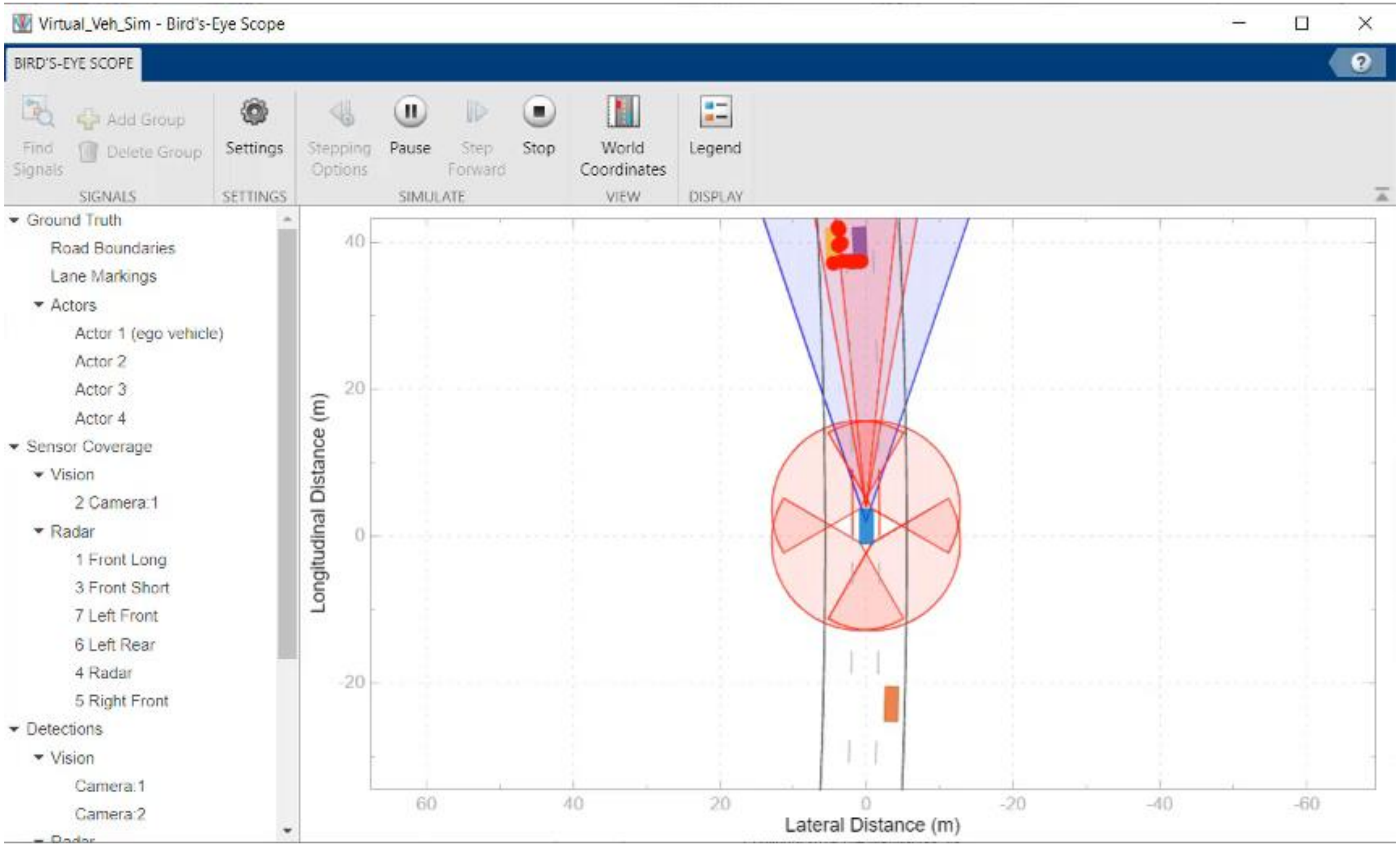
- Perception System for SAE Level 2
  – Utilize Automated Driving Toolbox Driving Scenario Designer app
  – Configure vision and mid-range radar (MRR) sensors on ego vehicle
  – Red areas are blind spots
  – Integrated sensors into virtual vehicle model



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:

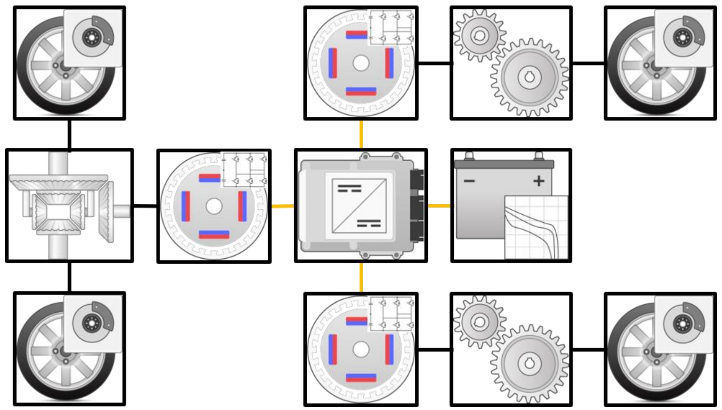
- Perception: Design LiDAR, vision, radar, and sensor fusion algorithms with recorded and live data
- Planning: Visualize street maps, design path planners, and generate C/C++ code
- Controls: Design a model-predictive controller for traffic jam assist, test with synthetic scenes and sensors, and generate 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 hand code

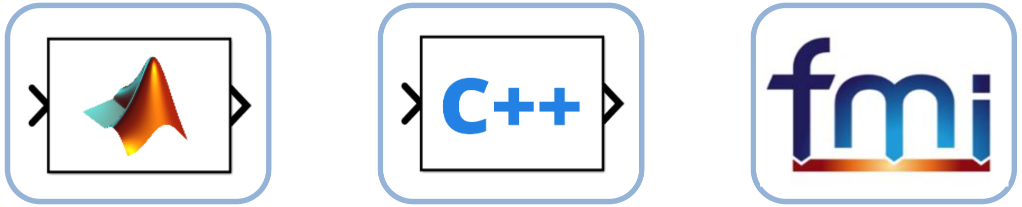
Shusen Zhang, MathWorks

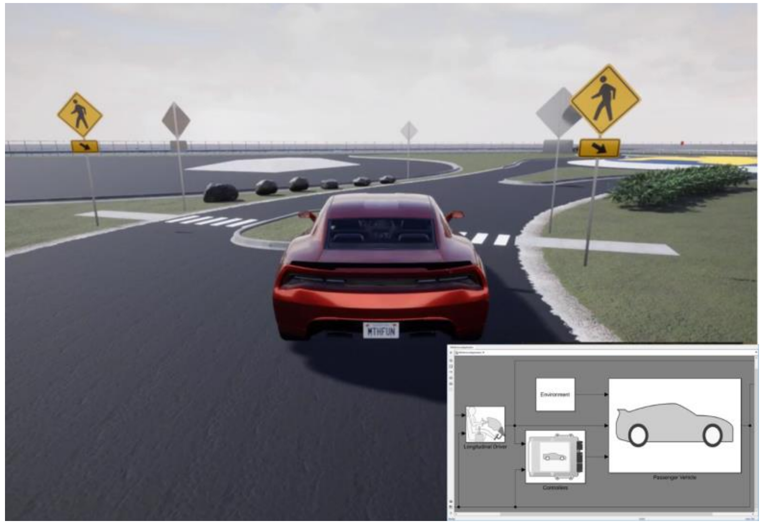
Model-Based Design == Model Reuse

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