전자기 및 열 효과를 고려한 상세한 모터 모델 및 AI 기반 차원 축소 모델 개발

강효석 프로/Ph.D

Application Engineer
MathWorks Korea

Challenges

- Traditionally, workflow of the machine design team and the control team have proceeded separately.
- A linear lumped parameters-based model is good for control design but doesn’t give a complete picture like saturation and losses
- Motor design tools provide good insight about electromagnetic, thermal and mechanical behavior but run very slow
- Detailed Electro- Magnetic and Thermal model is needed to improve efficiency, minimize the torque ripples and validate the controller's performance across the operating range.

Steps to Design a Motor Drive

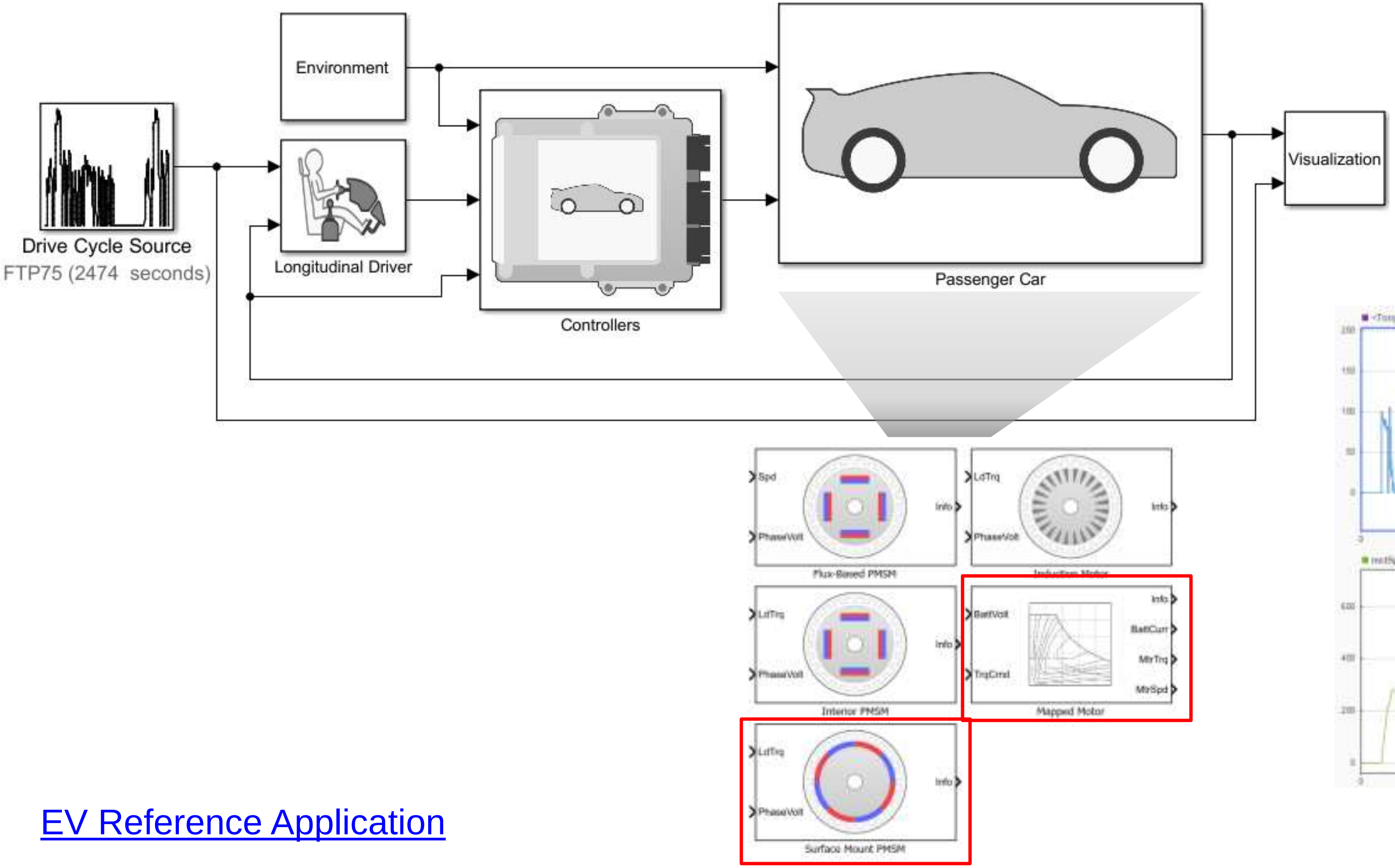
- **Motor Designer**
  - **Configuration:** Sizing, pole/slot design, windings
  - **Performance & Thermal:** Simulate losses and thermal performance
  - **Validate:** Simulate with actual current waveforms for realistic losses

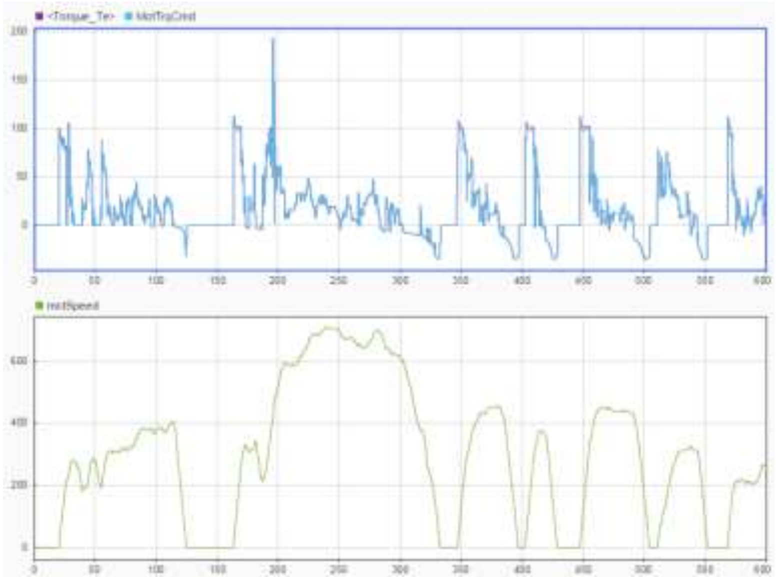
- **Motor Control Engineer**
  - Desing closed -loop Control
  - Operating point management (compute Id, Iq for given torque/speed)
  - Model Drive Electronics
  - And System Simulation to get the efficiency and torque ripples

Let's Put Some Context to the Motor

- Component Selection?
- Component Sizing?
- Trade-off Studies?

- Detailed Component Modeling and Control Design?



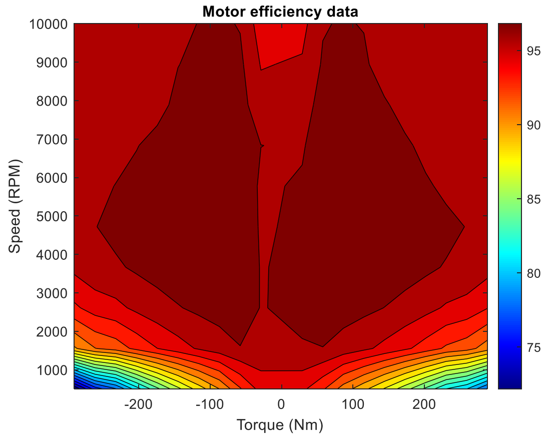


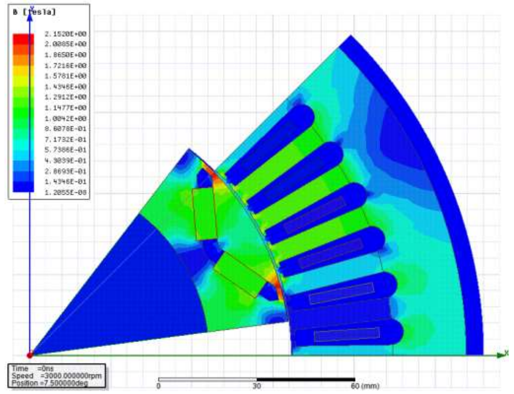
[EV Reference Application]

Workflow for Motor Design Engineers

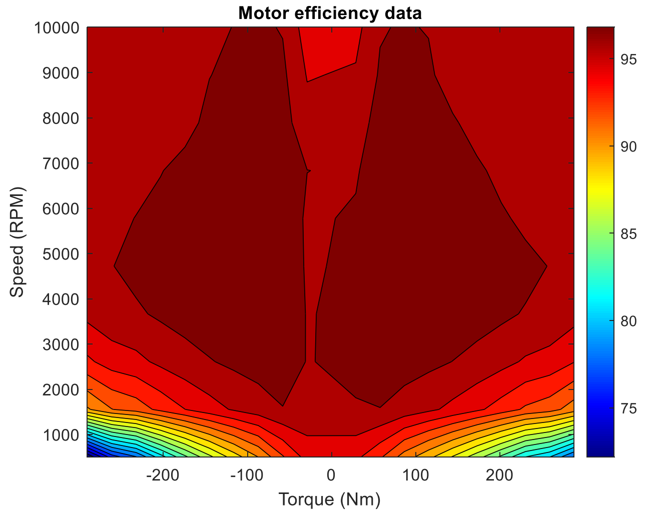
With Motor Design tool, Engineers:

- Explore Design space of the motor and understand the impact of motor geometry, windings on Thermal and Mechanical performance
- Generate Torque Speed curves and efficiency maps for entire operating range
- Do FEA simulations to capture flux linkage for given range of currents and rotor position

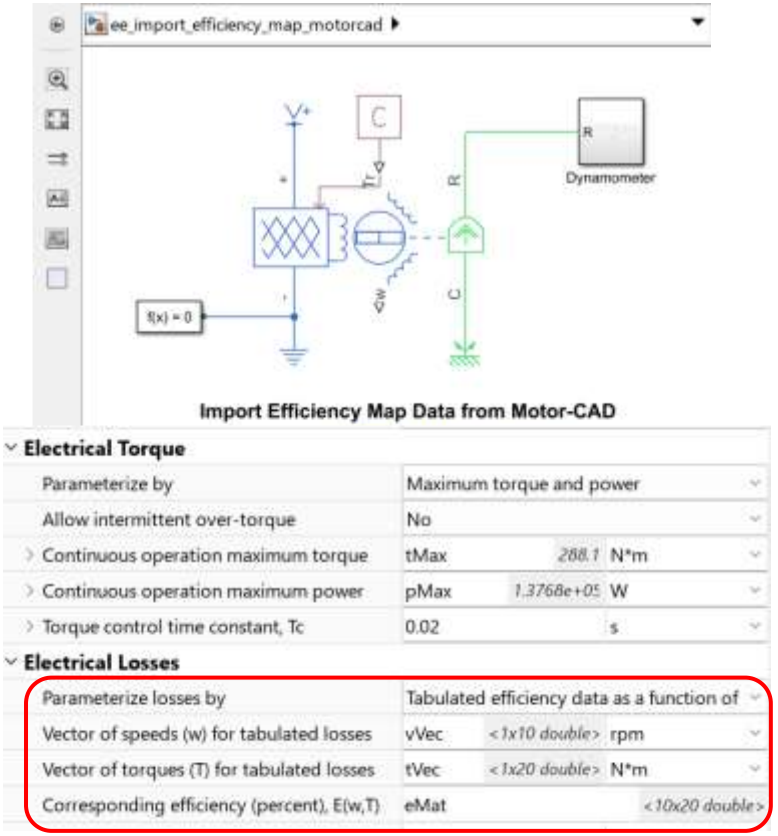


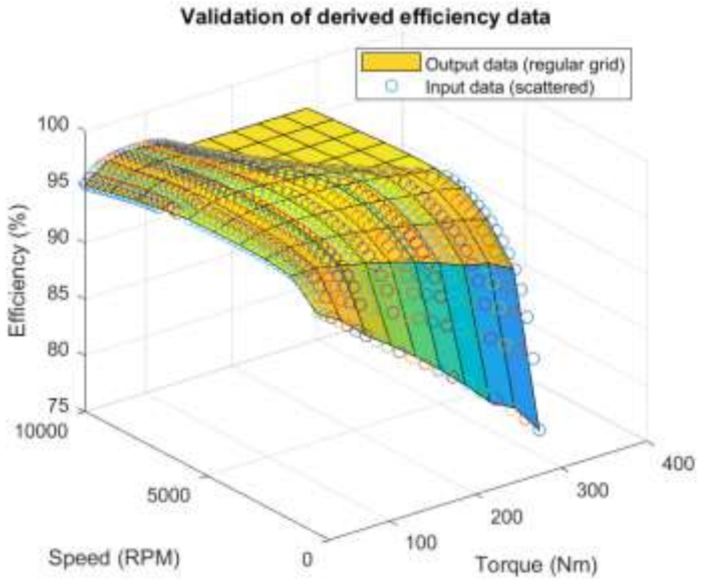


Importing Motor Efficiency Map to Simscape

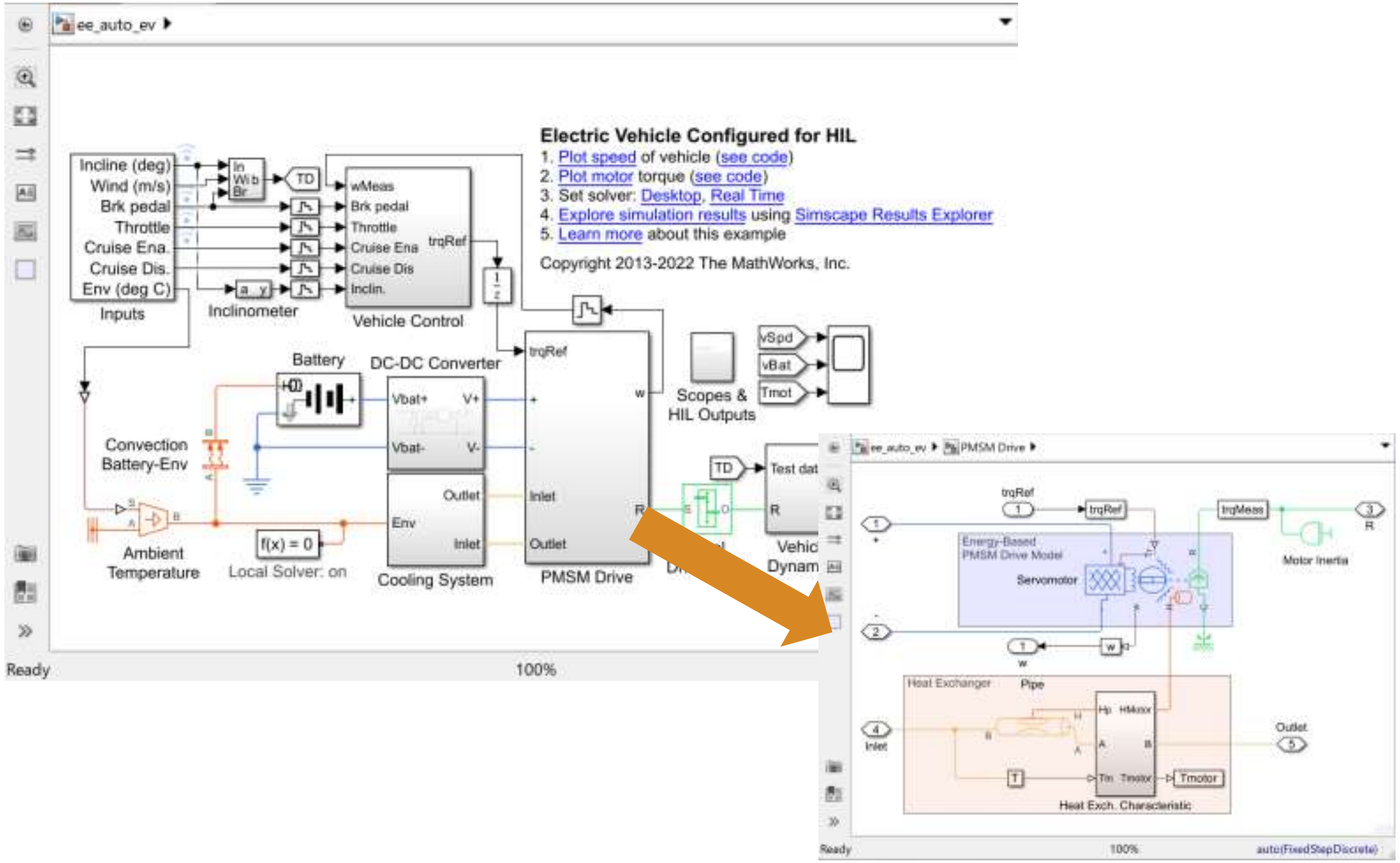


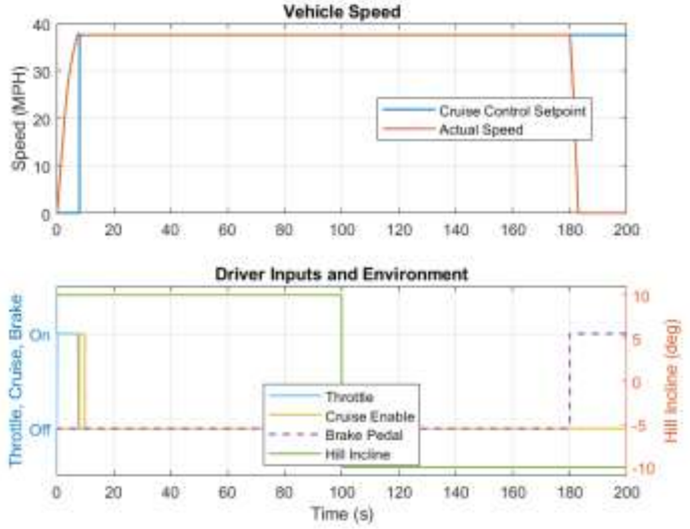
Motor efficiency data from
a Motor Desing tool

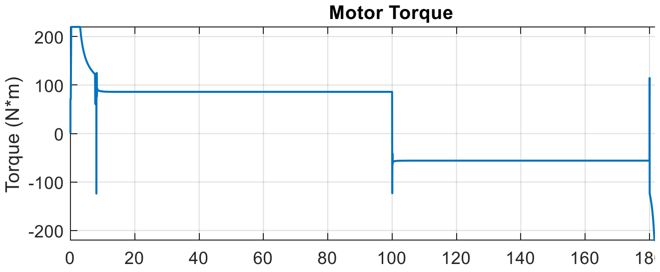


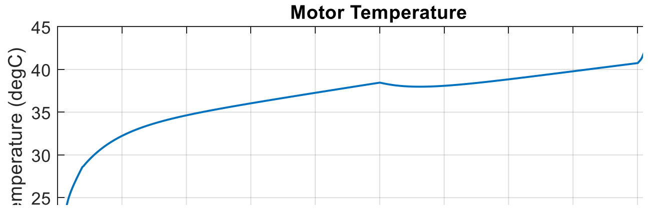


Importing Motor Efficiency Map to Simscape







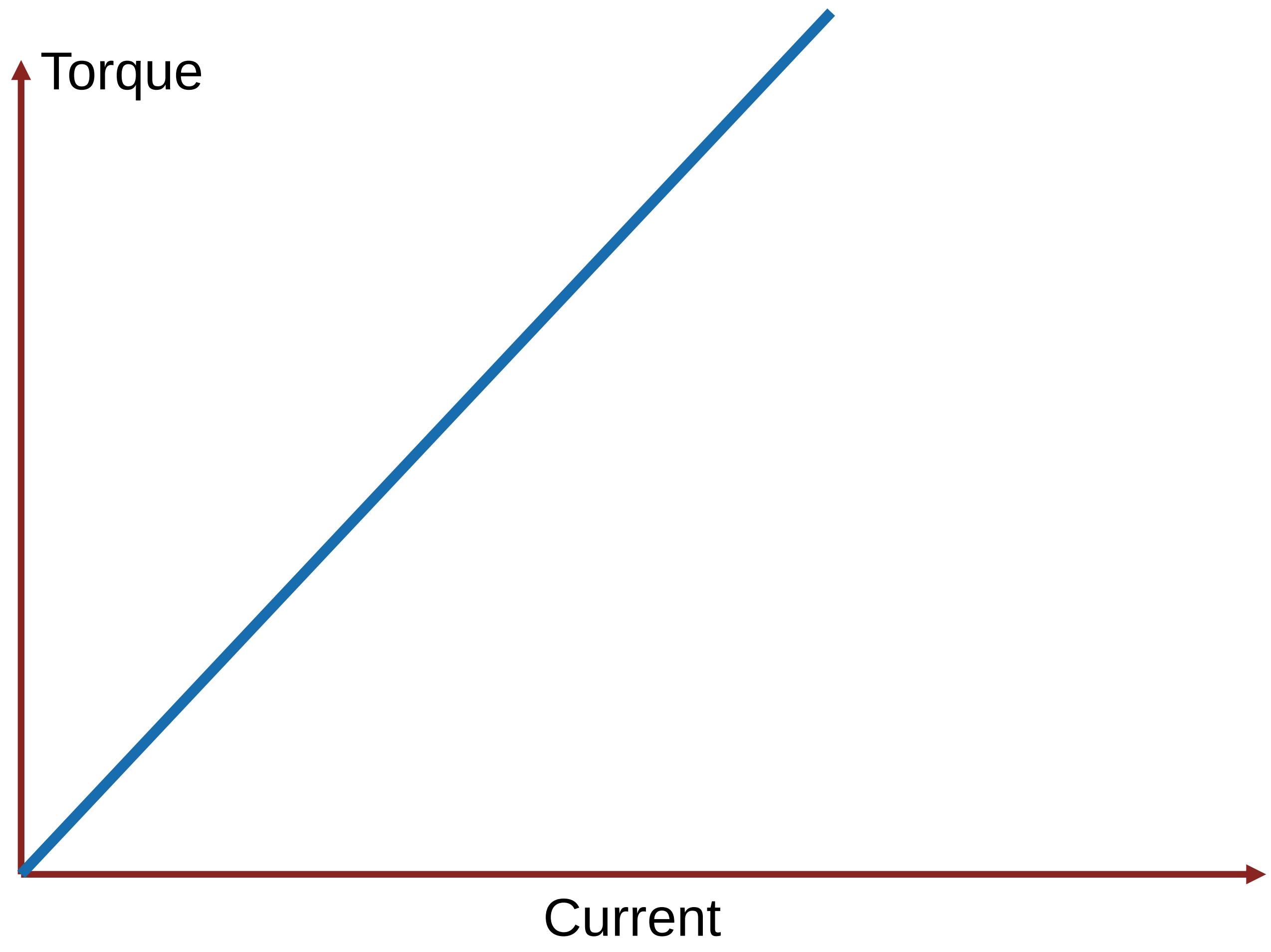


Workflow for Motor Control Engineers

- Develop a plant (Motor + Inverter) model
- Select a Control Architecture
- Tune Control loop gains
- Fine-tune controller parameters to minimize the torque ripples
- Code generation and deployment to the target hardware

How to Deploy Control Algorithm to a Microcontroller

Linear Lumped-Parameter Model



Required Parameters

Electrical Model

$$v_d = R i_d - L_q p \omega_r i_q + L_d \frac{d}{dt} i_d$$

$$v_q = R i_q + p \omega_r (L_d i_d + \lambda) + L_q \frac{d}{dt} i_q$$

$$\omega_e = p \omega_r$$

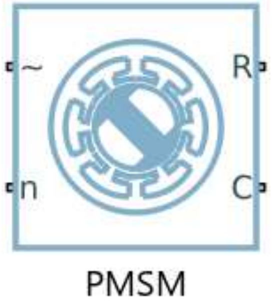
$$T_e = 1.5 p \left[\lambda i_q + (L_d - L_q) i_d i_q \right]$$

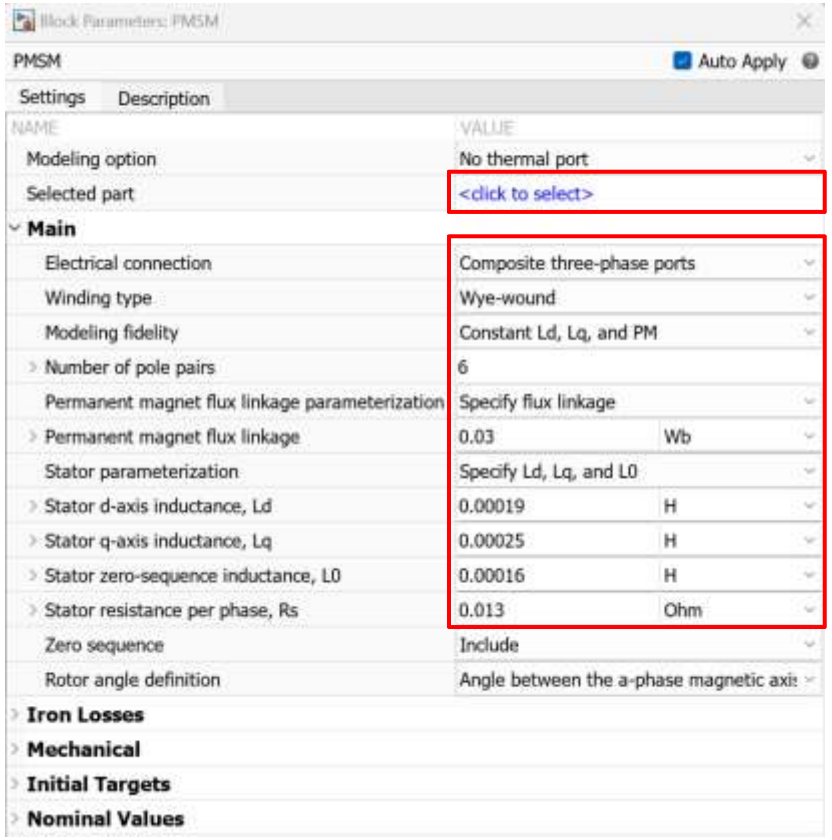
$$T_e = K_t i_q \quad (assumes\ round\ rotor, L_d = L_q)$$

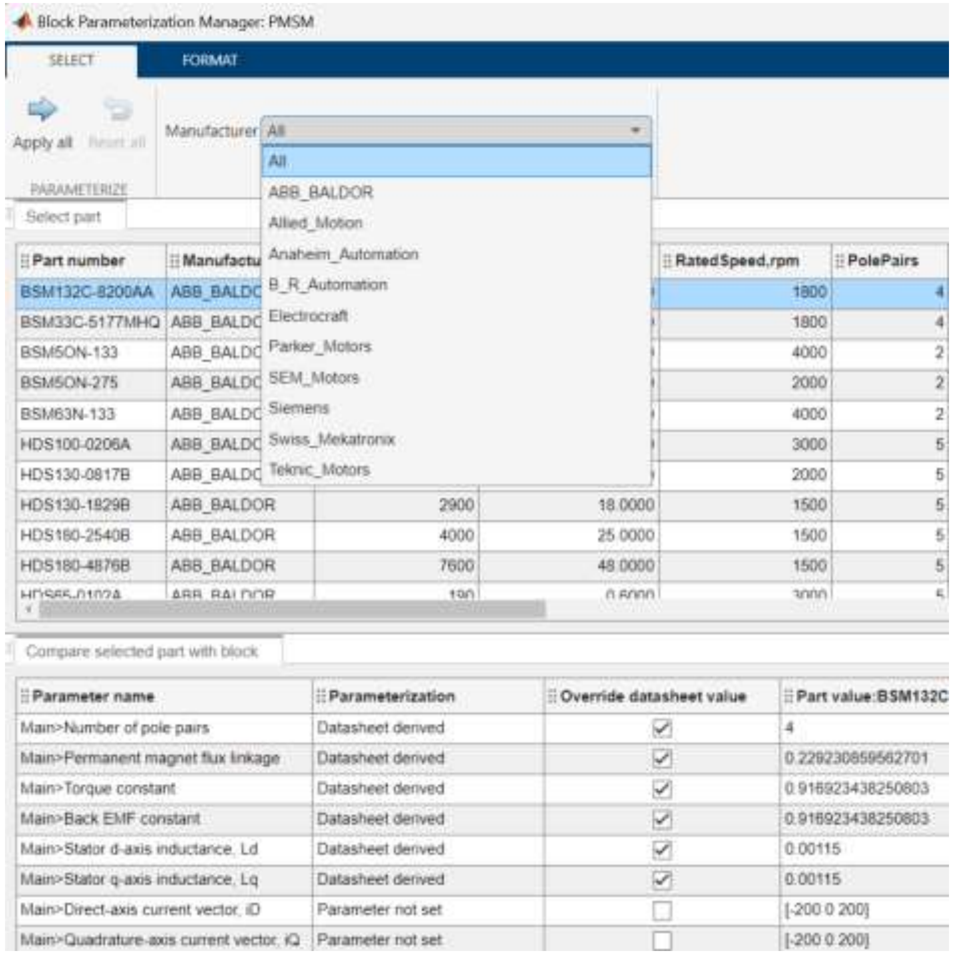
Mechanical Model

$$\frac{d}{dt}\omega_r = \frac{1}{J}(T_e - sgn(\omega_r) J_0 - b\omega_r - T_{load})$$

Motor Parametrization in Simscape







Parameter Estimation by Running Instrumented Tests

- Instrumented tests running on the target
- Sensor-based and Sensor-less modes available
- Supports PMSM and Induction Motor

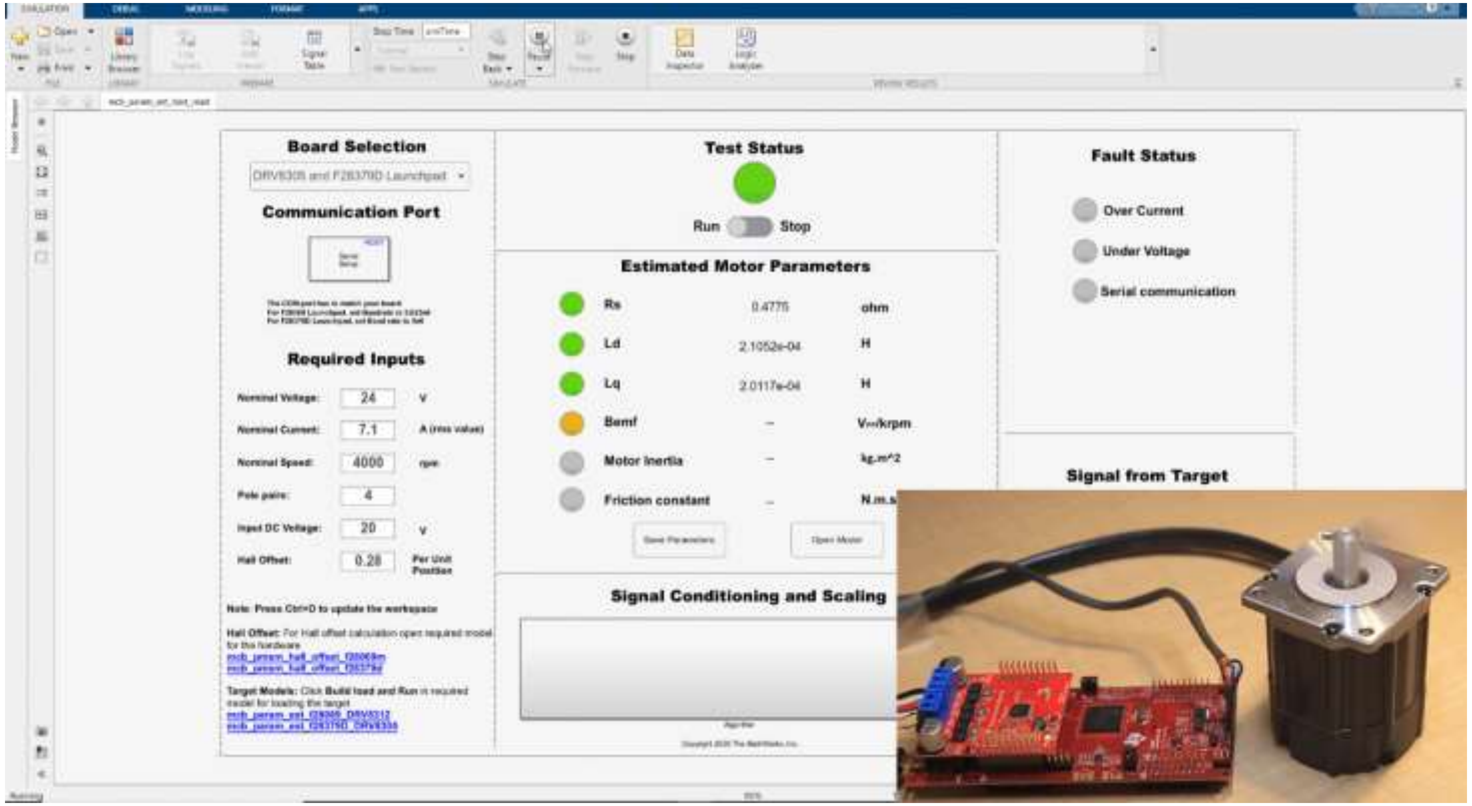
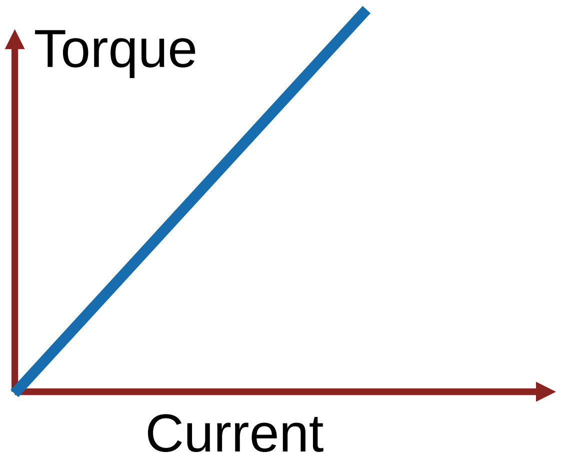
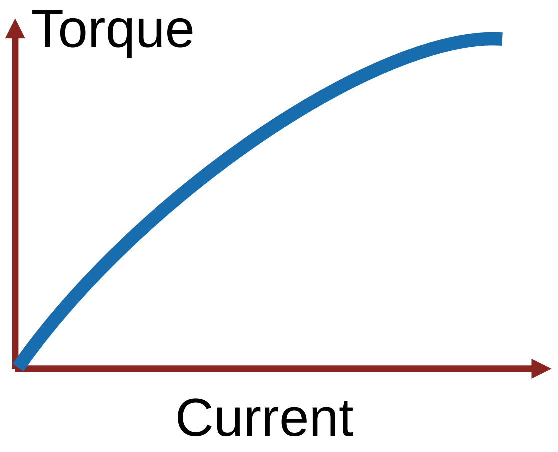


Figure : gif showing parameter estimation capability with TI C2000 hardware

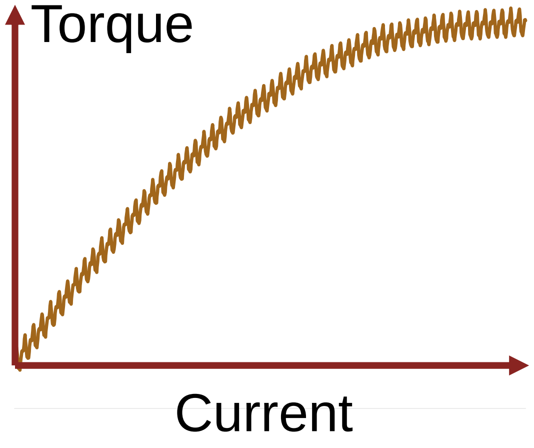
Motor Model Fidelity



Lumped Parameter



Saturation



Saturation +
Spatial Harmonics

Model Non-linear Effects such as Saturation and Spatial Harmonics

Electrical Model

$$v_d = Ri_d - L_q p\omega_r i_q + L_d \frac{d}{dt} i_d$$

$$v_q = Ri_q + p\omega_r (L_d i_d + \lambda) + L_q \frac{d}{dt} i_q$$

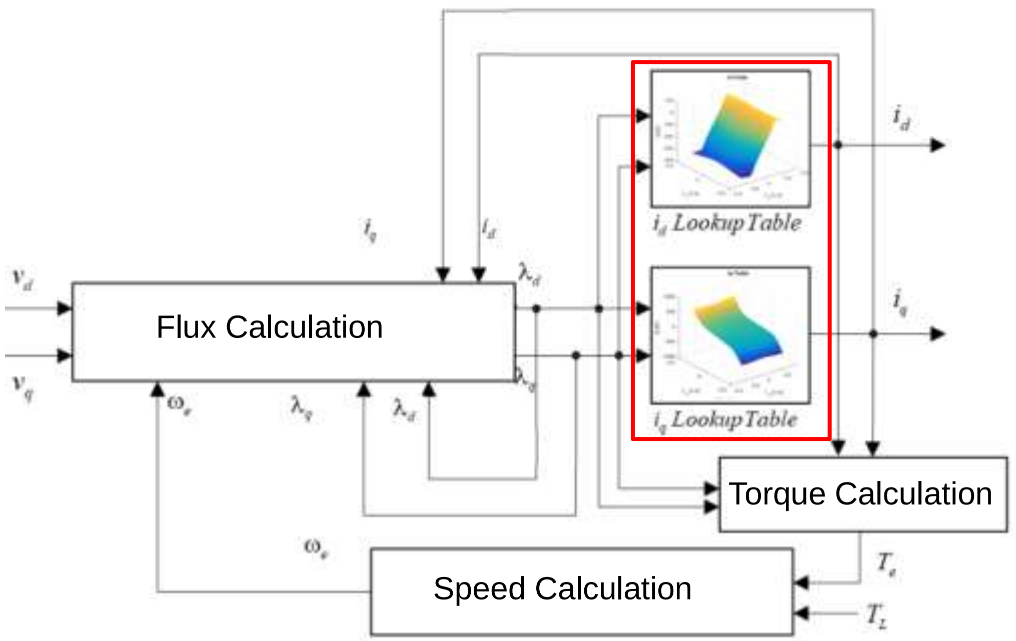
$$\omega_e = p\omega_r$$

$$T_e = 1.5p[\lambda i_q + (L_d - L_q) i_d i_q]$$

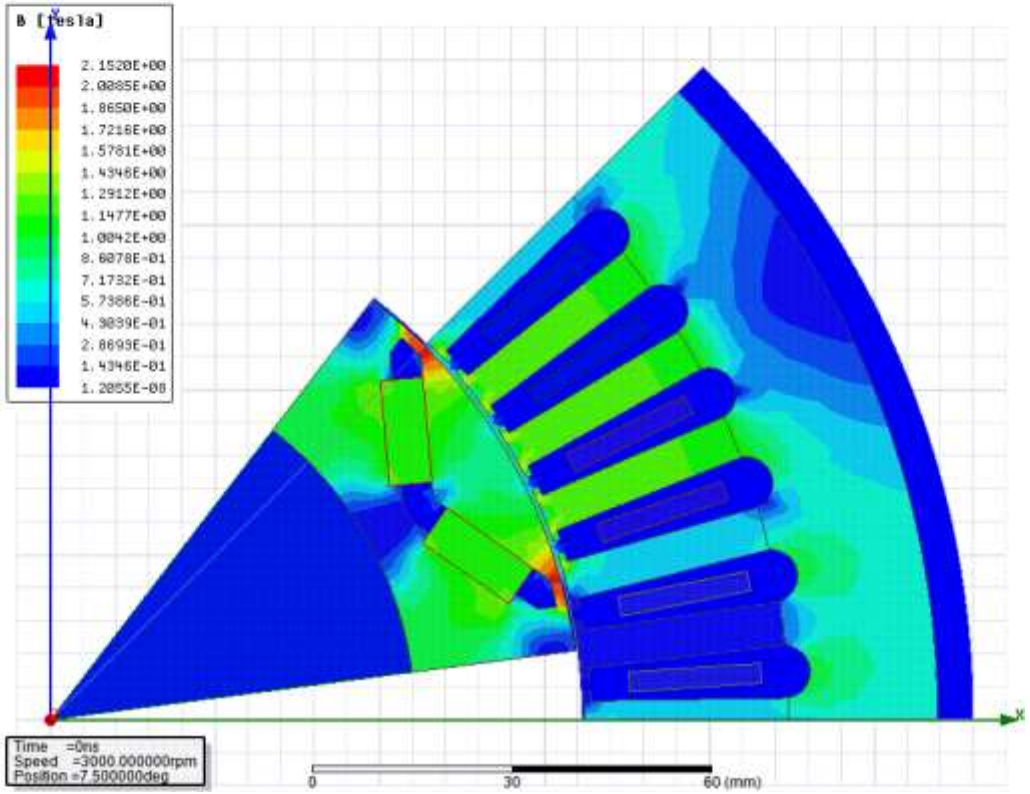
$$T_e = K_t i_q \; (assumes\ round\ rotor, L_d = L_q)$$

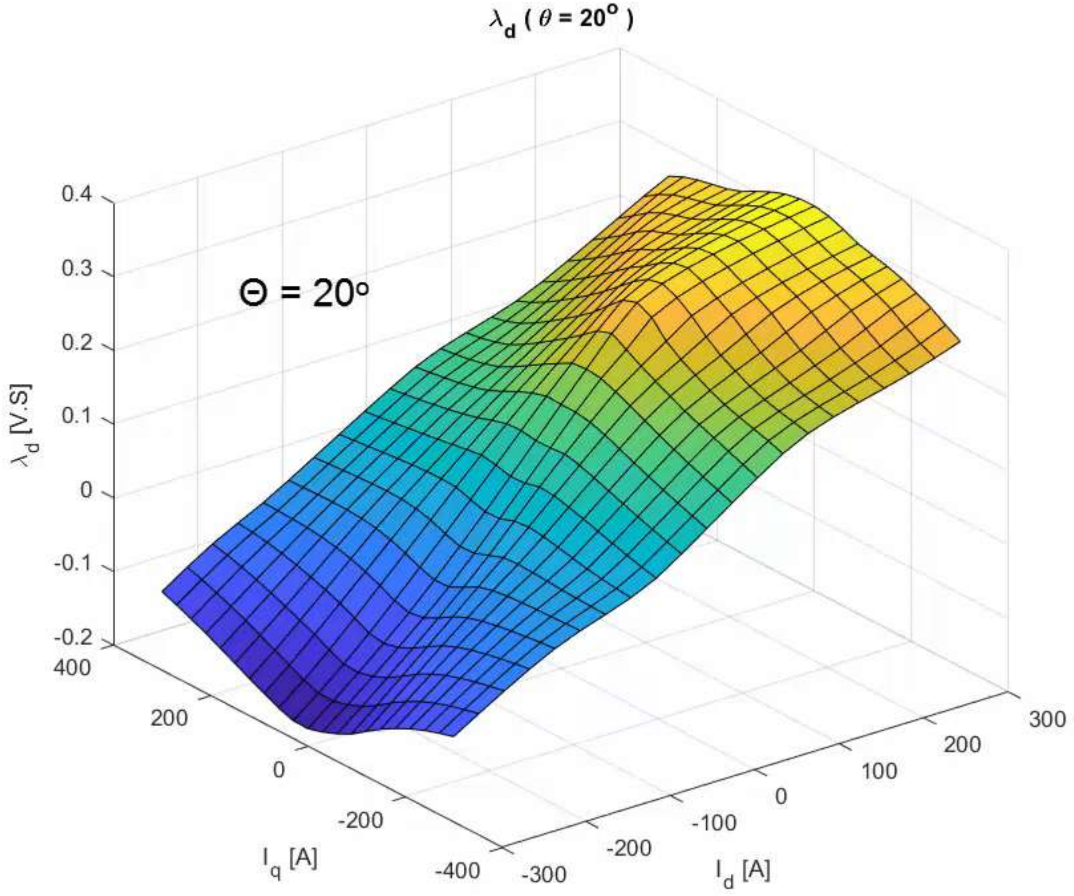
Mechanical Model

$$\frac{d}{dt}\omega_r = \frac{1}{J}(T_e - sgn(\omega_r)J_0 - b\omega_r - T_{load})$$

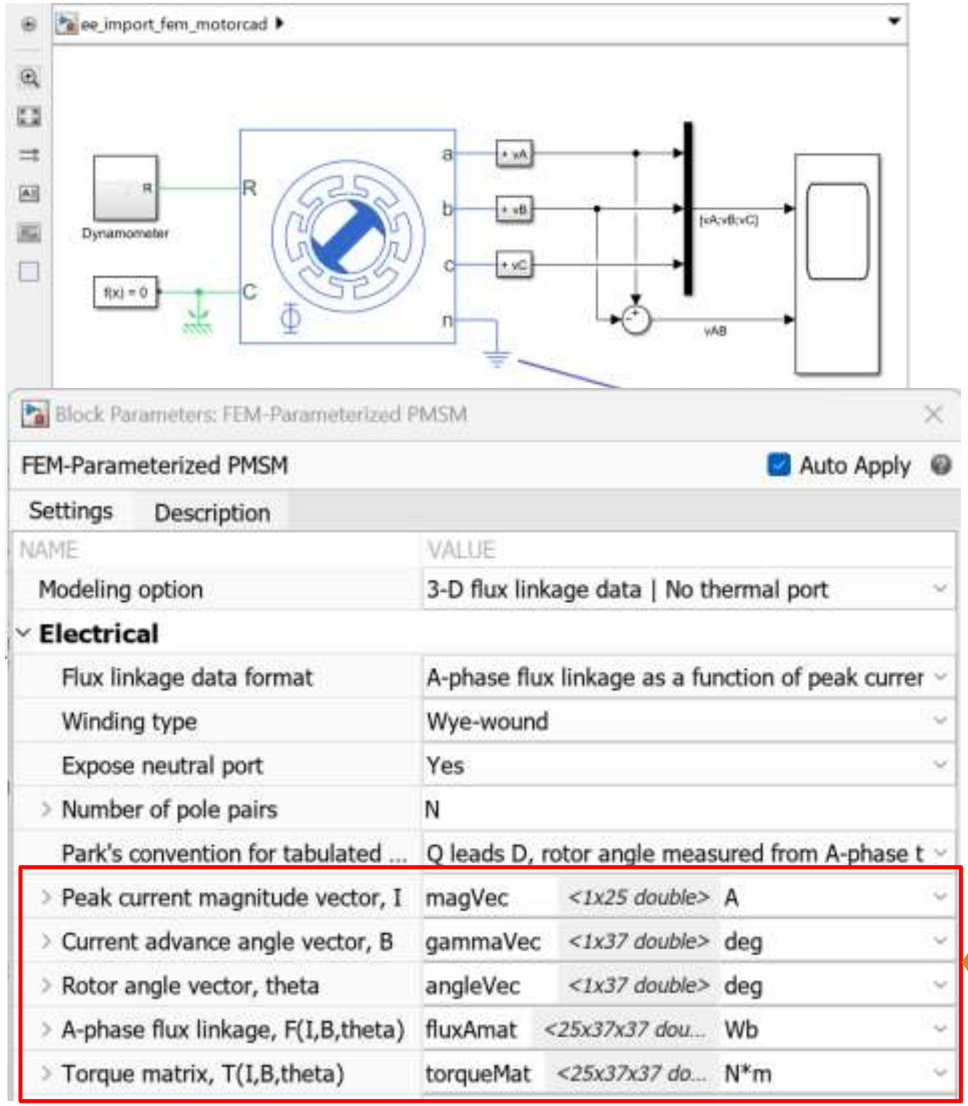


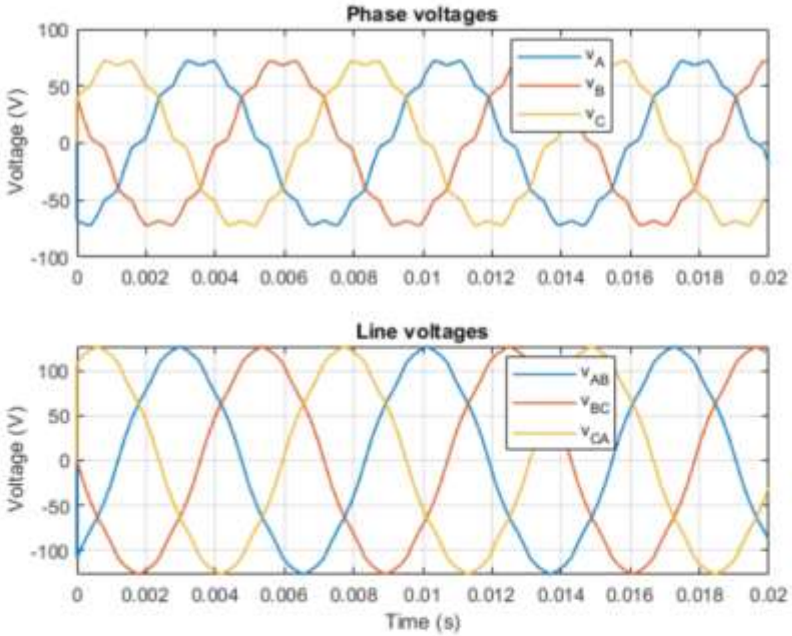
Use Motor Design Tools Flux vs Current and Rotor Position Data





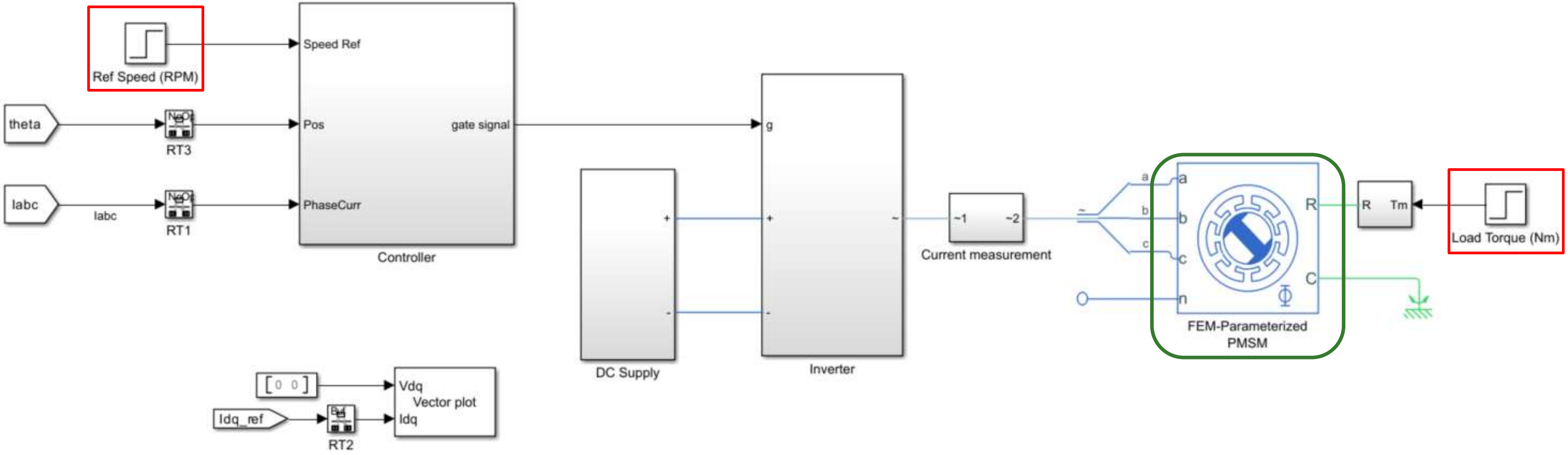
Importing Non-linearities and Loss Map in Simscape





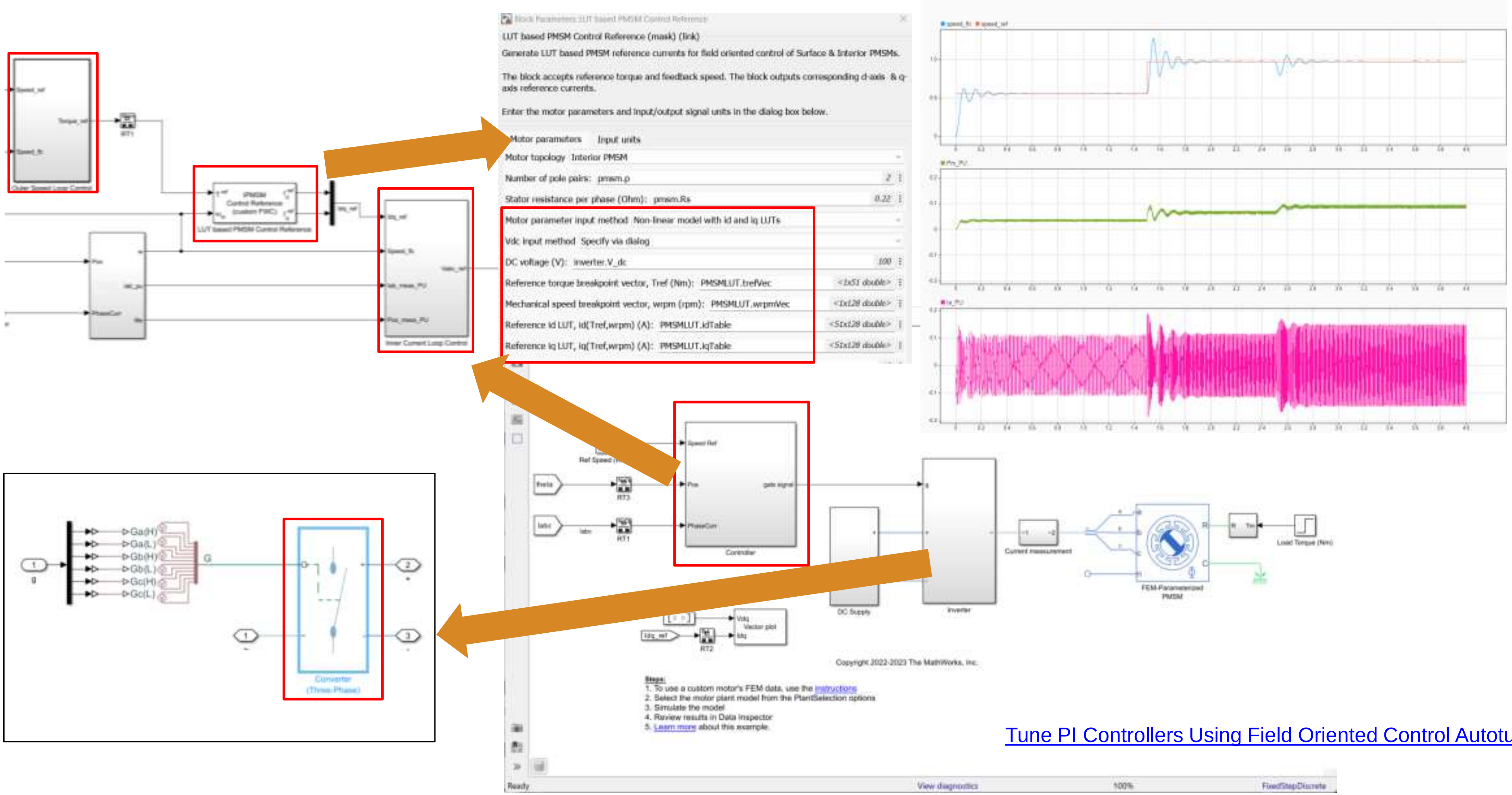

```
N = double(data.PolePairs); % Number of pole pairs
magVec   = data.Stator_Current_Phase_Peak(:,1)'; % Peak current magnitude vector
gammaVec  = data.Phase_Advance(1,:); % Current advance angle vector
angleVec = -data.DriveOffsetAngleLoad/N + ...     % Offset to align A-phase with q-axis
           squeeze(data.Angular_Rotor_Position(1,1,:))'/N; % Rotor angle vector
fluxDmat = data.Angular_Flux_Linkage_D;           % D-axis flux linkage
fluxQmat = data.Angular_Flux_Linkage_Q;           % Q-axis flux linkage
fluxAmat = -data.Angular_Flux_Linkage_Phase_1;    % A-phase flux linkage
torqueMat = data.Angular_Electromagnetic_Torque; % Torque matrix

fLosses = data.Frequency;  % Frequency at which iron losses determined
rotorKhMat = data.Iron_Loss_Rotor_Back_Iron_Hysteresis_Coefficient + ...
             data.Iron_Loss_Rotor_Pole_Hysteresis_Coefficient; % Steinmetz hysteresis loss coefficient matrix
rotorKjMat = data.Iron_Loss_Rotor_Back_Iron_Eddy_Coefficient + ...
             data.Iron_Loss_Rotor_Pole_Eddy_Coefficient; % Steinmetz eddy current loss coefficient matrix
statorKhMat = data.Iron_Loss_Stator_Back_Iron_Hysteresis_Coefficient + ...
             data.Iron_Loss_Stator_Tooth_Hysteresis_Coefficient; % Steinmetz hysteresis loss coefficient matrix
statorKjMat = data.Iron_Loss_Stator_Back_Iron_Eddy_Coefficient + ...
             data.Iron_Loss_Stator_Tooth_Eddy_Coefficient; % Steinmetz eddy current loss coefficient matrix
Rs = data.Phase_Resistance_DC_at_20C; % Stator resistance
```

Look-up Table Based Reference Current Calculation



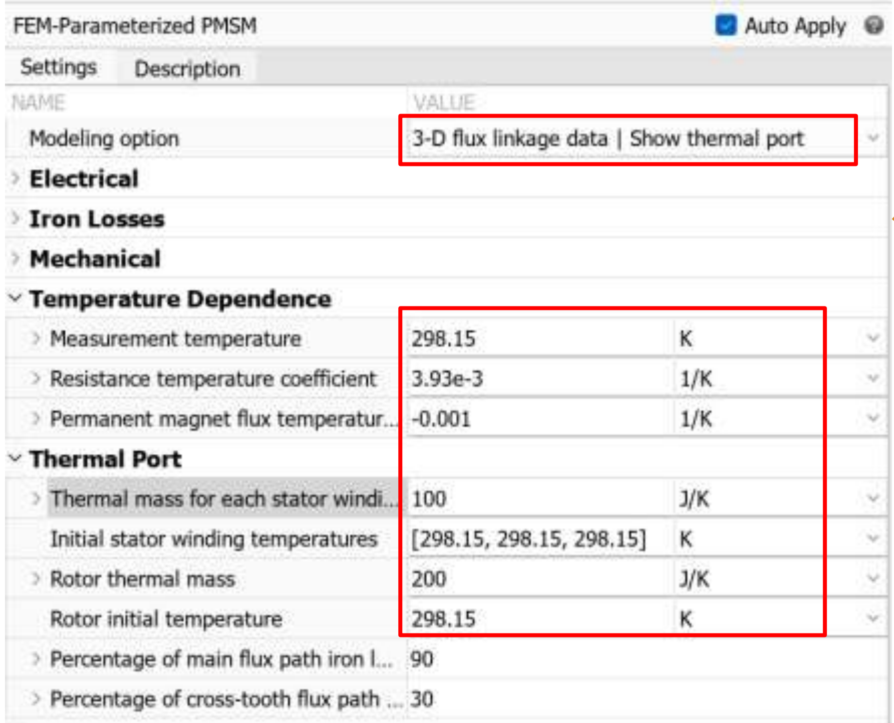
Steps:
1. To use a custom motor's FEM data, use the instructions
2. Select the motor plant model from the PlantSelection options
3. Simulate the model
4. Review results in Data Inspector
5. Learn more about this example.

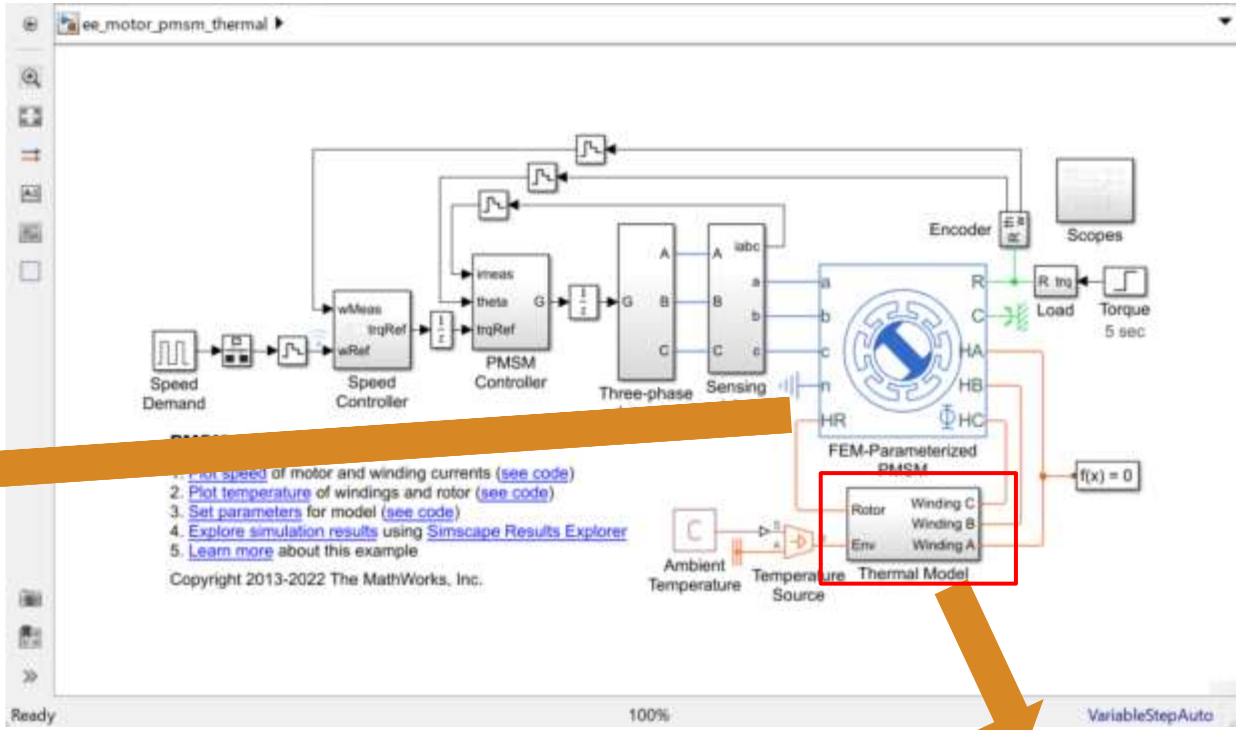
Look-up Table Based Field Oriented Control

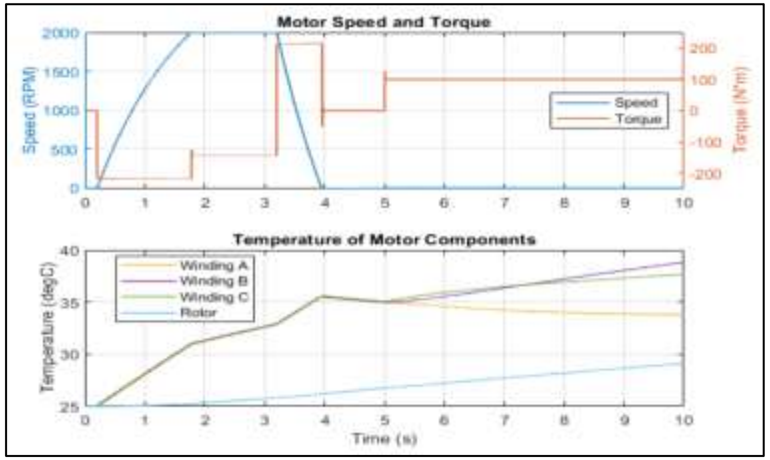


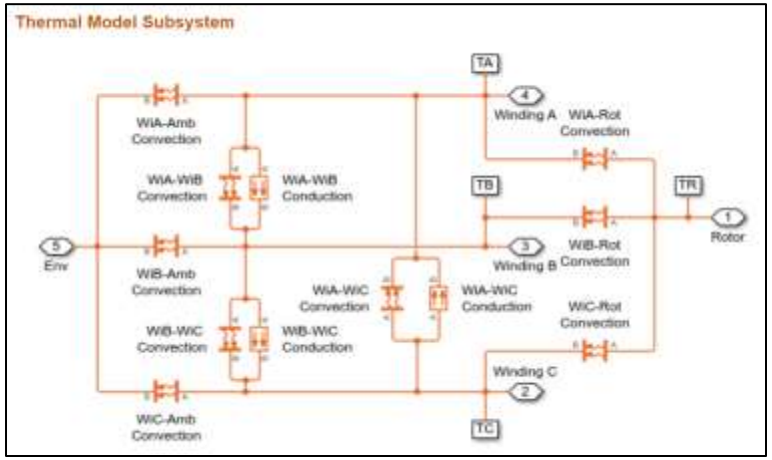
[Tune PI Controllers Using Field Oriented Control Autotuner]

Motor with Thermal Model









Challenges and Solutions

High fidelity models, such as ones from 3rd party FEA tools, are too slow for system level simulation and HIL testing.

Balancing a ROM that ensure desired results in terms of speed, accuracy, interpretability, etc.

Reduced Order Models (ROM) Speed Up System Analysis, Design and Facilitate Real-time Deployment

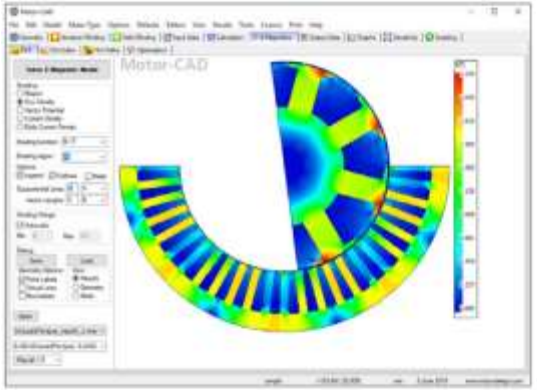
What

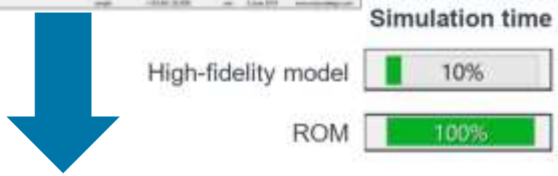
- Techniques to reduce the computational complexity of a computer model
- Preserved acceptable fidelity with-in controlled error

Why

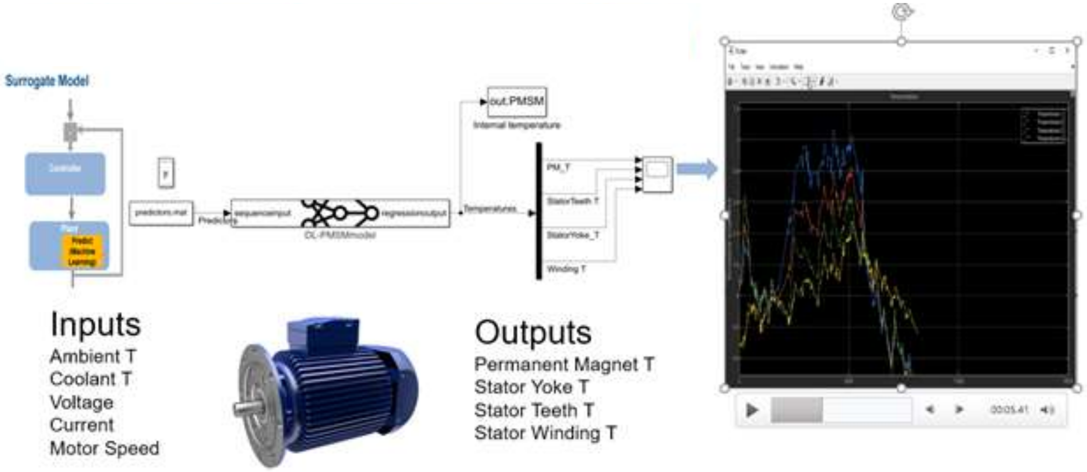
- Enable faster simulation of high-fidelity FEA models in Simulink
- Perform hardware-in-the-loop testing
- Develop virtual sensors, Digital twins
- Enable desktop simulations for orders-of-magnitude longer timescales

High-fidelity model





Reduced-Order Model (ROM)



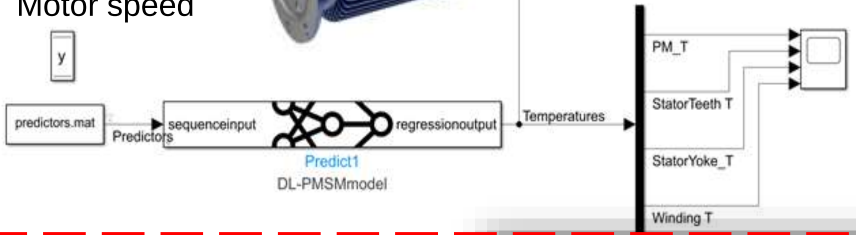
Different ROM Techniques Are Suitable for Different Applications

Reduced order model

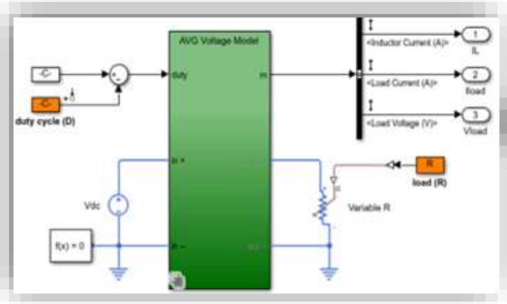
AI-Based Data-driven

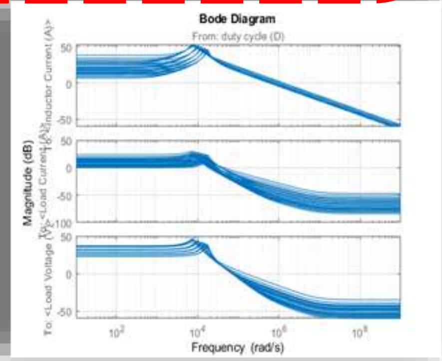
Input
Ambient T
Coolant T
Voltage
Current
Motor speed

Output
Permanent Magnet T
Stator Yoke T
Stator Teeth T
Stator winding T



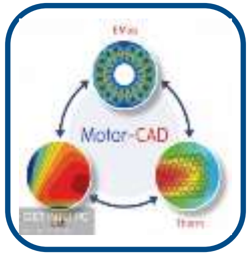
Linearization



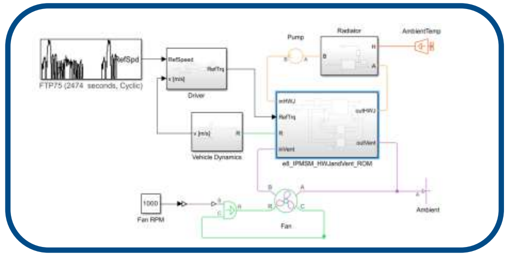


Model-based

Motor-CAD



Integrating with detailed Simscape model

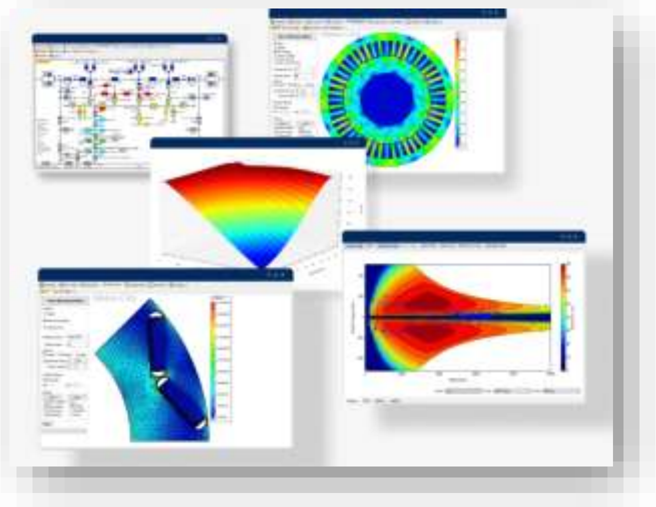


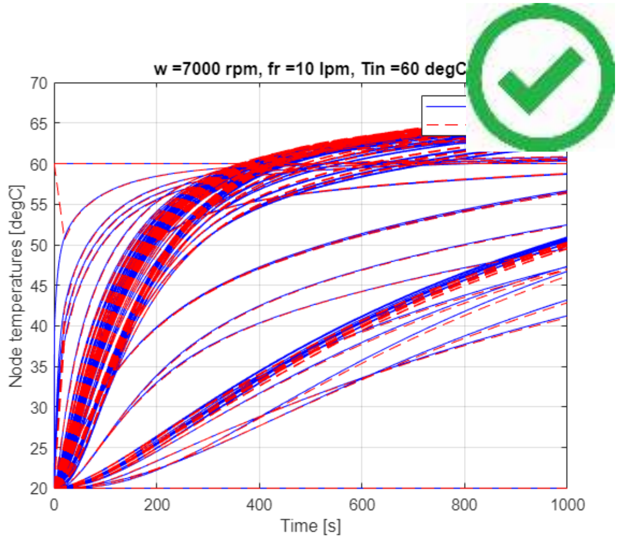
Workflow for Simulink Reduced Order Thermal Modeling (SROTM)

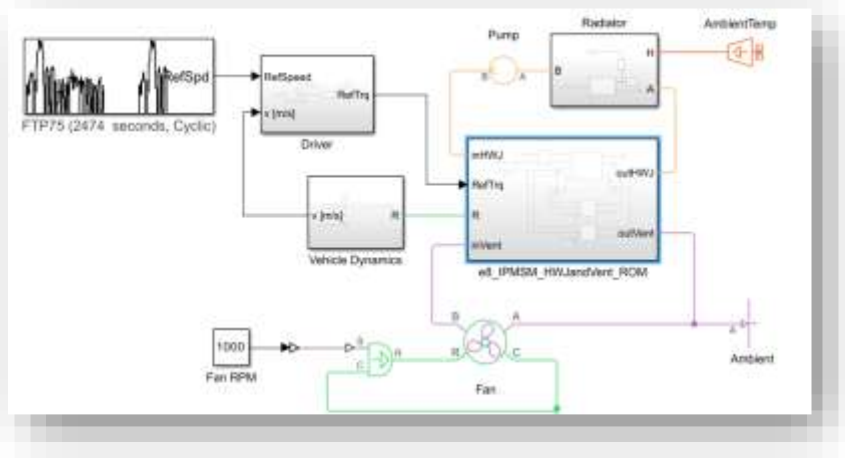
1 Generate a SROTM through Motor design tools

2 Validate the SROTM

3 Integrate the SROTM with vehicle-level dynamics

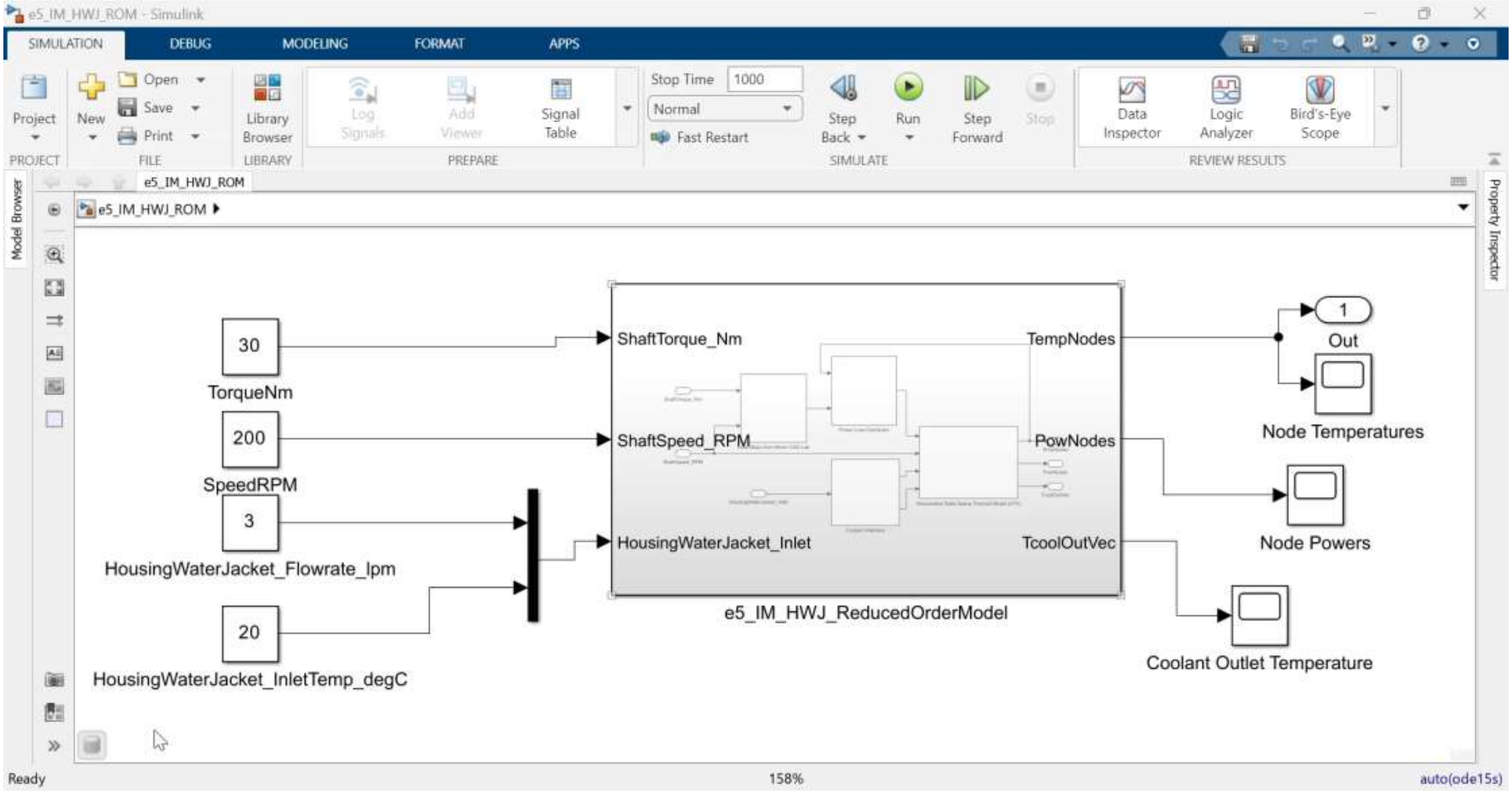






Speed-up Multi-physics analysis across the full torque-speed operating range.

Reduced Order Thermal Modeling using Motor Design Tools



Interpolated State-Space Thermal Model

Interpolated State Space Thermal Model implements the state space model as:

$$\frac{dx}{dt} = A\,x + B\,p$$

where:

x is the vector of node temperatures (the states), *Tnodes*

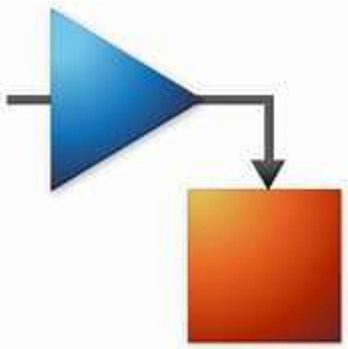
p is the vector of node losses, *nodeloss*

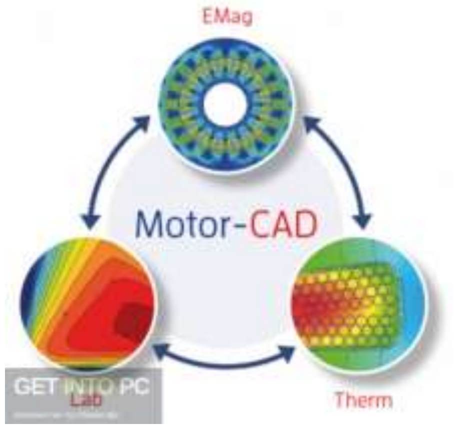
A and ***B*** are interpolated from a set of state-space matrices at different operating points (**speed, coolant flow rates, and coolant inlet temperatures**):

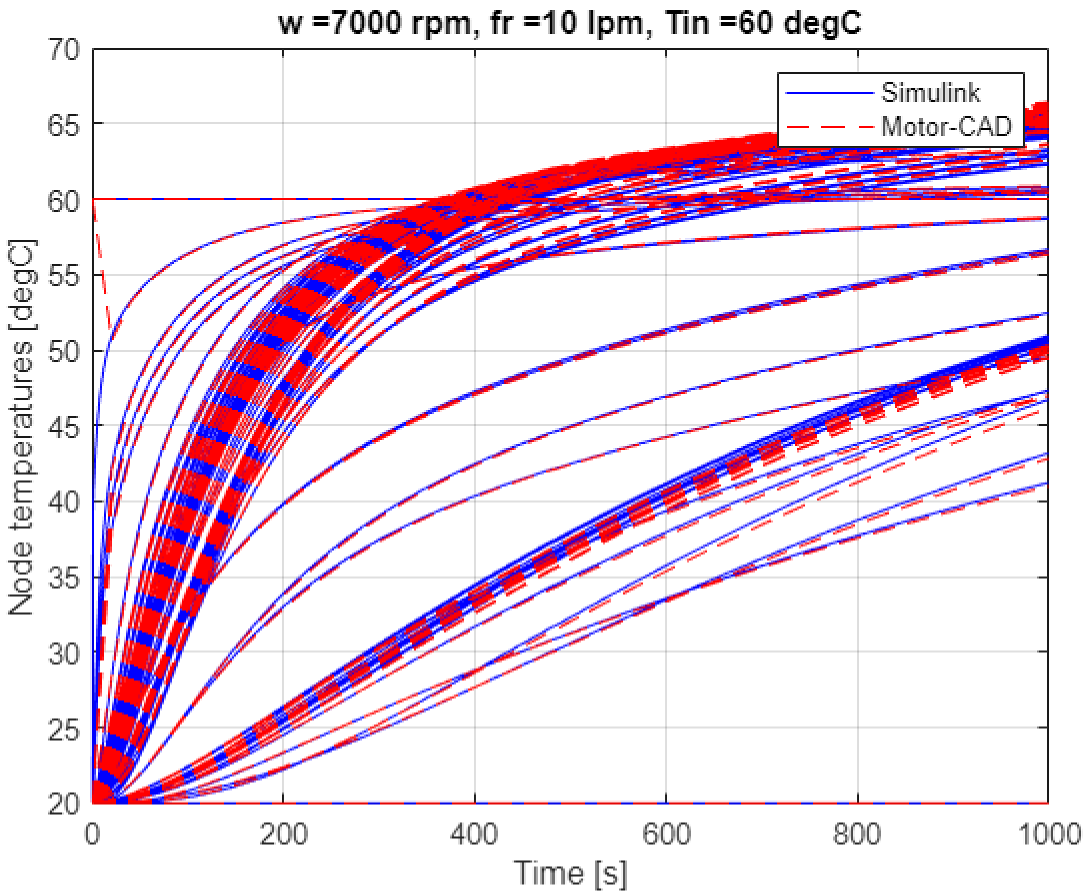
A = Afun(w, fr, Tin)
B = Bfun(w, fr, Tin)

This type of model is commonly known as *Linear Parameter Varying model* (**LPV model**)

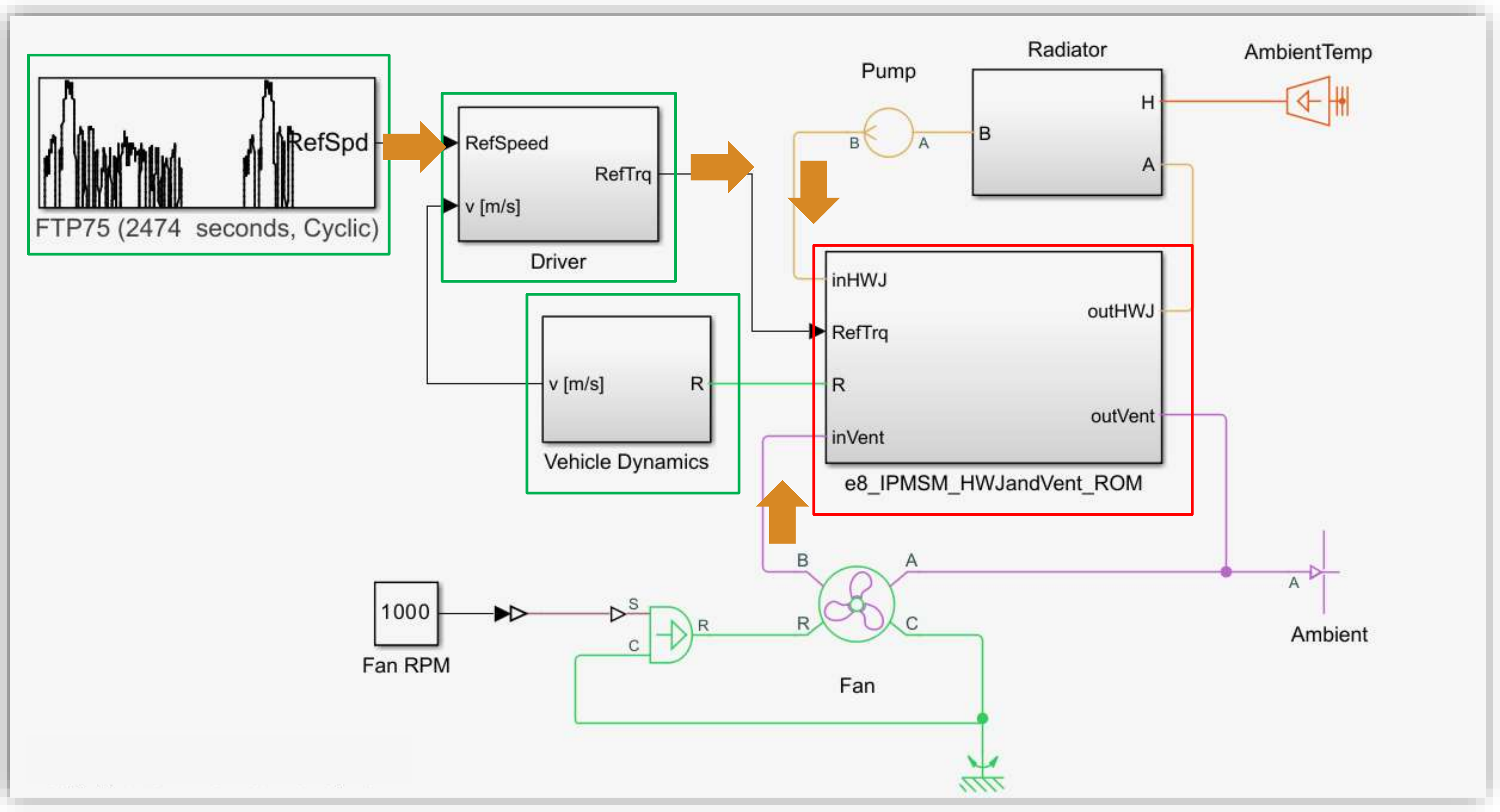
Comparison between Simulink and Motor-Design Tool







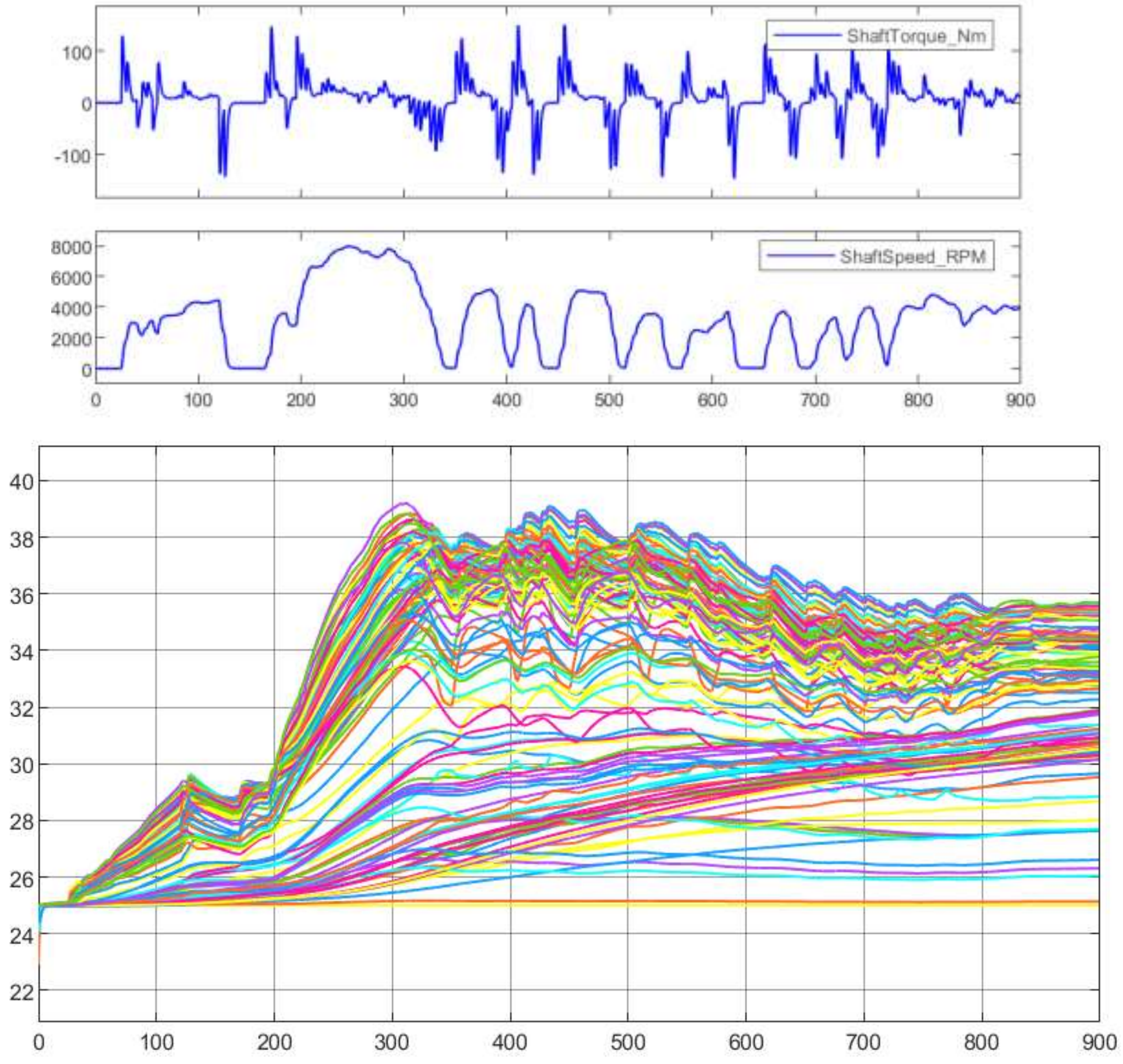
Integrate SROTM with Simscape System-level Vehicle Model



Faster System Level Simulation

Torque &
Speed

Nodes
Temperature



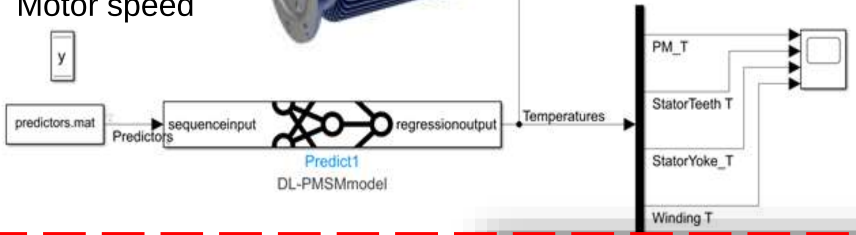
Different ROM Techniques Are Suitable for Different Applications

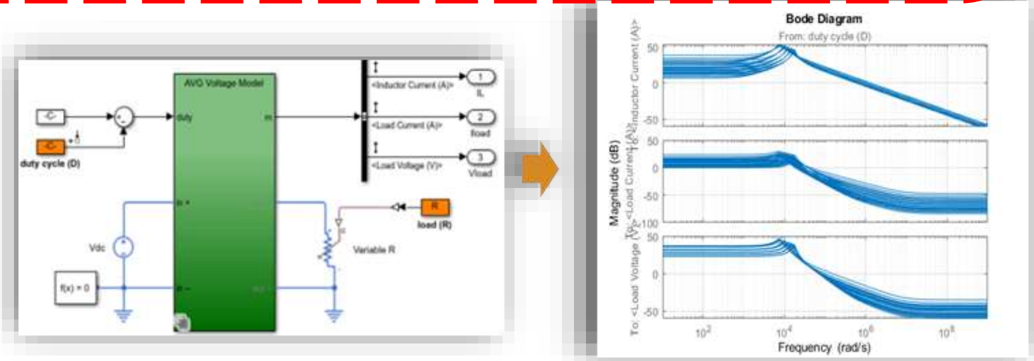
Reduced order model

- **AI-Based Data-driven**
- **Linearization**
- **Model-based**

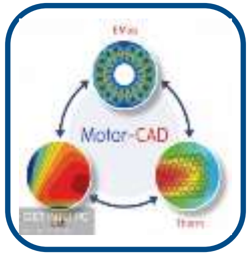
Input
Ambient T
Coolant T
Voltage
Current
Motor speed

Output
Permanent Magnet T
Stator Yoke T
Stator Teeth T
Stator winding T

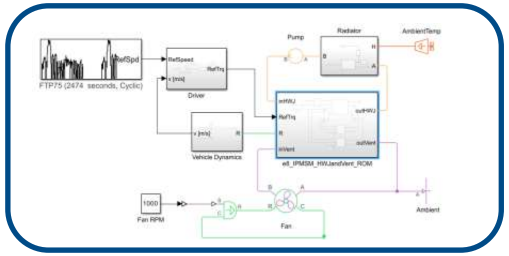




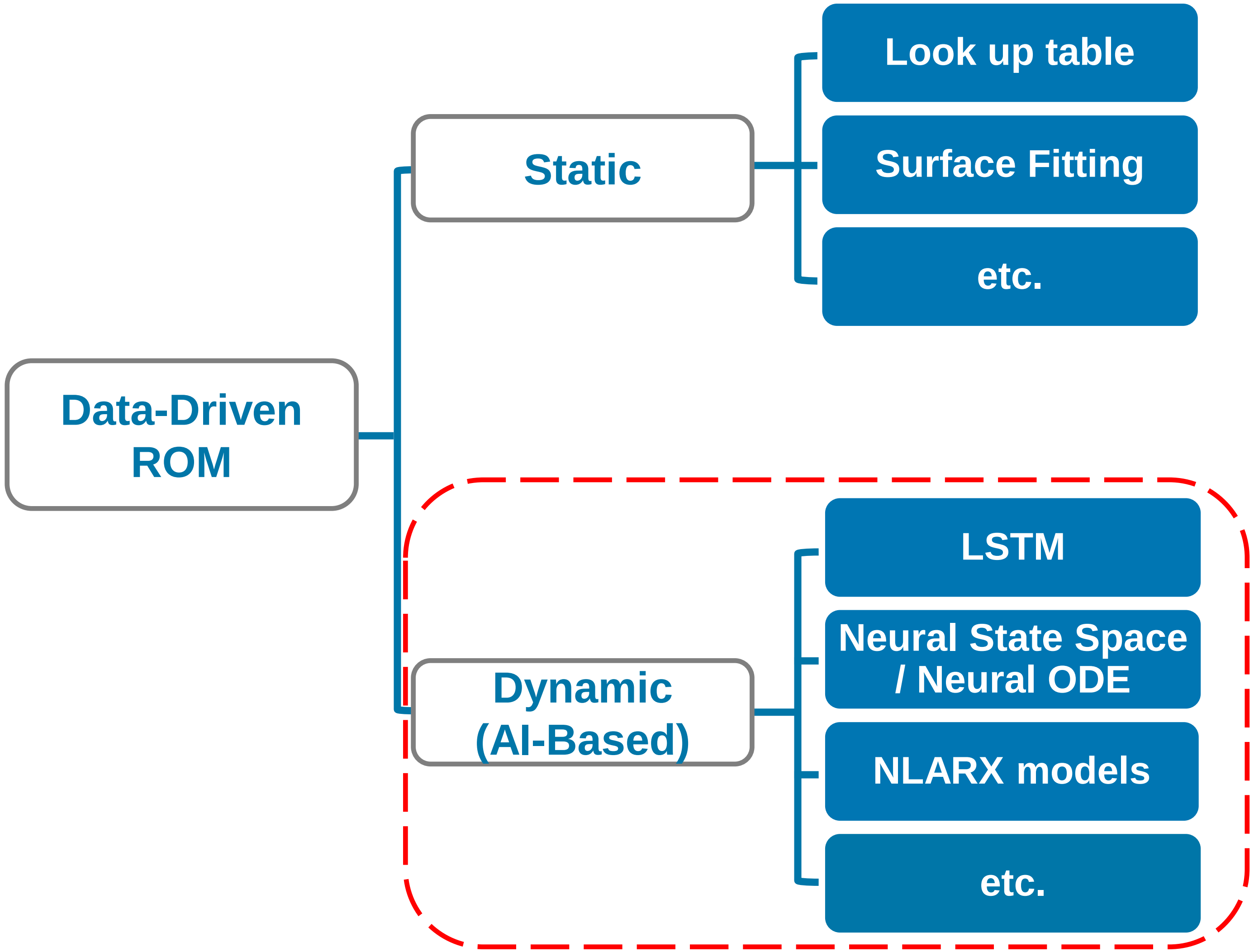
Motor-CAD



Integrating with detailed Simscape model



Data-driven ROM



Data Preparation | **AI Modeling** | Simulation & Test | Deployment

AI-driven System Design

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

Simulation & Test

- Integration with complex systems
- System simulation
- System verification and validation

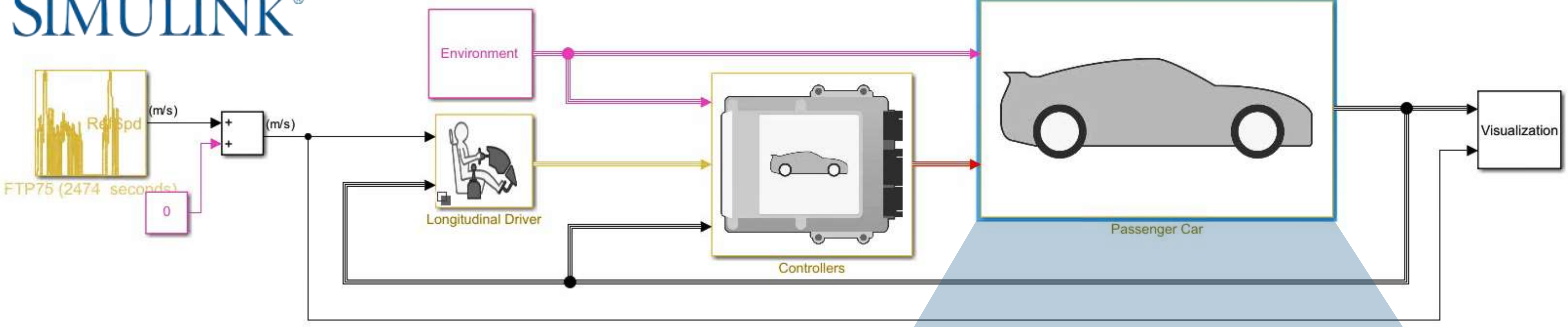
Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

Example Overview

Replacing a first-principles motor model with an AI-based Reduced Order Model

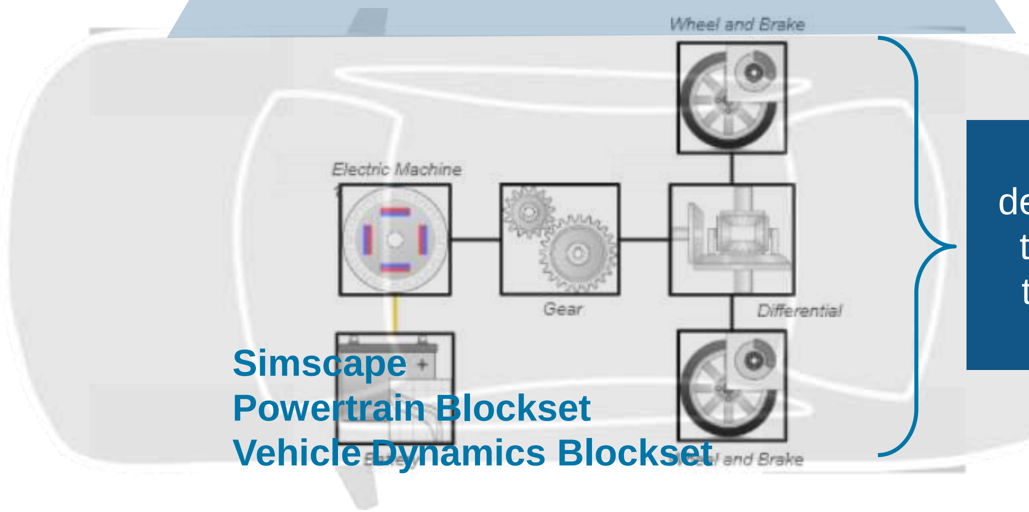
SIMULINK



High fidelity

Complex model

Slow simulation

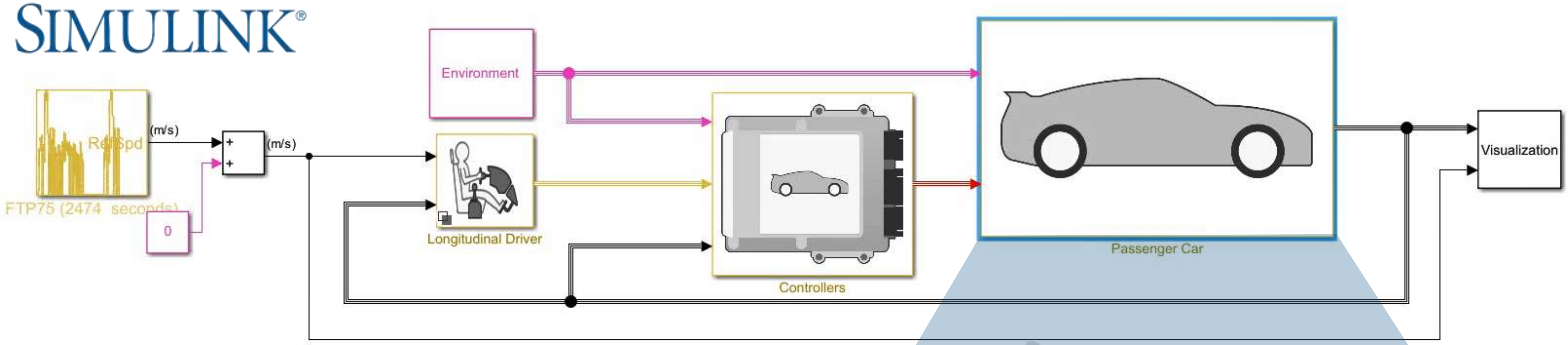


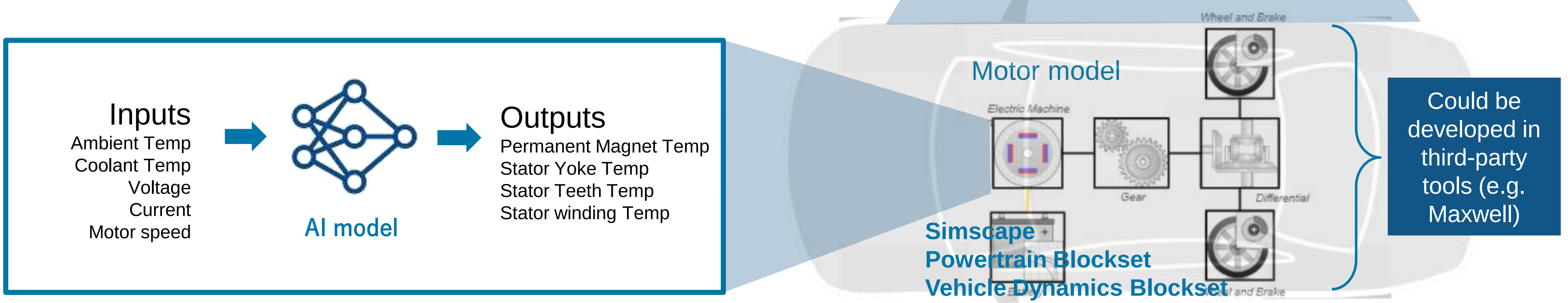
Simscape
Powertrain Blockset
Vehicle Dynamics Blockset

Could be developed in third-party tools (e.g. Maxwell)

Example Overview

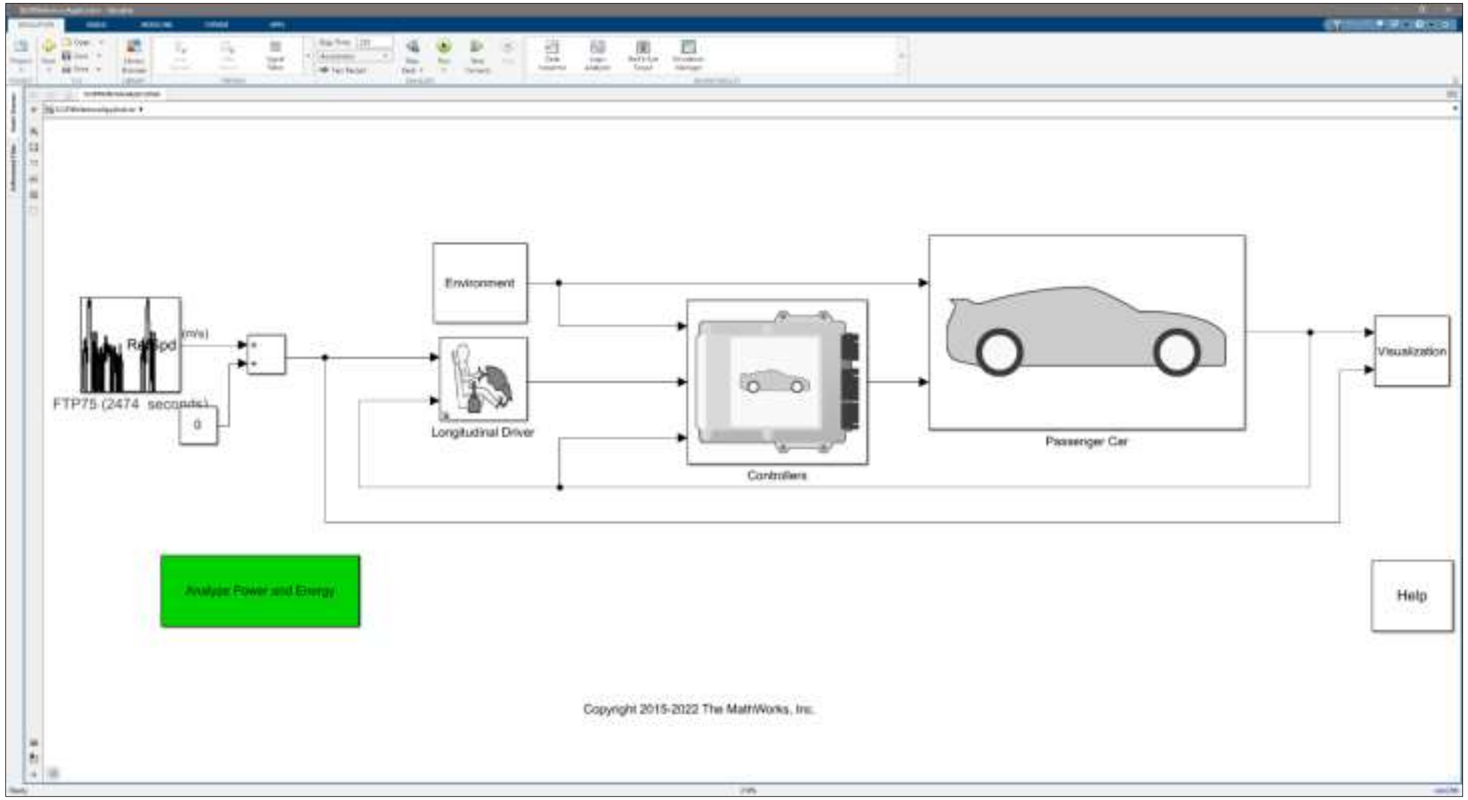
Replacing a first-principles motor model with an AI-based Reduced Order Model





BLACK BOX — GREY BOX — WHITE BOX

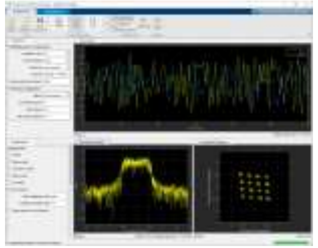
Generate Synthetic Data for Training



Simulink/Simscape

Other techniques:

GANs



Wireless Waveform Generator

Unreal Engine®

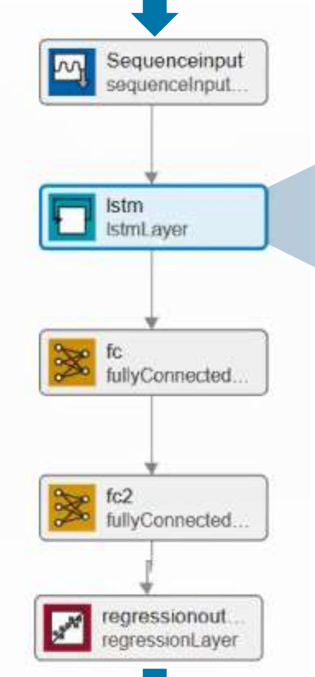
Data Preparation | **AI Modeling** | **Simulation & Test** | **Deployment**

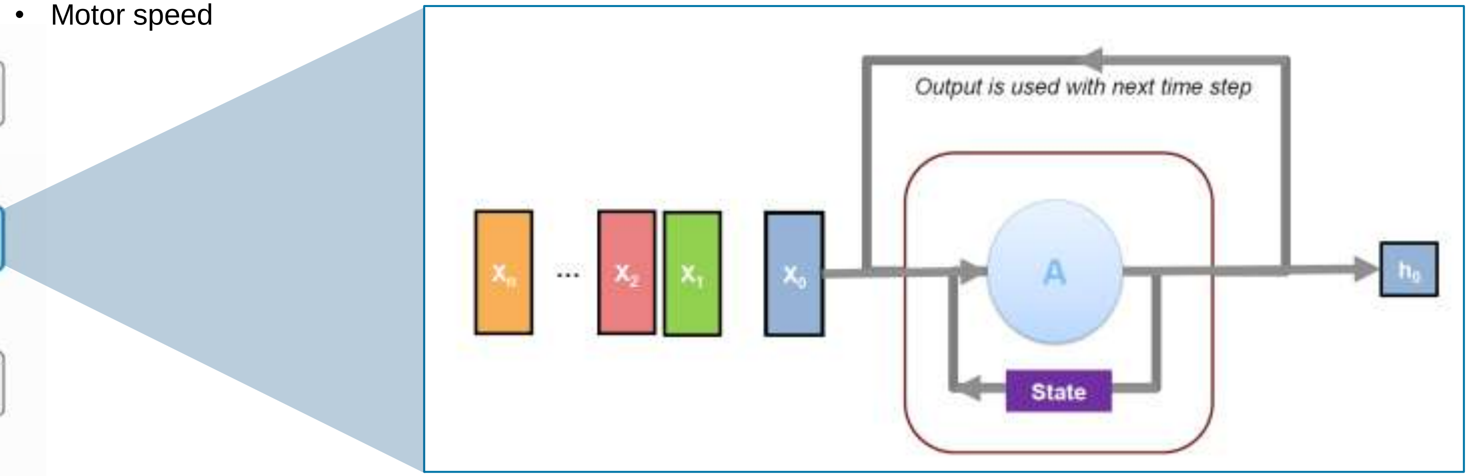
AI-based ROM using LSTMs

Capture time dependencies in time-series data

Inputs

- Ambient Temp
- Coolant Temp
- Voltage
- Current
- Motor speed





LSTMs carry a memory cell (state) throughout

Outputs

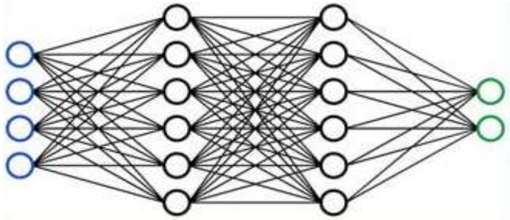
- Permanent Magnet Temp
- Stator Yoke Temp
- Stator Teeth Temp
- Stator winding Temp

Data Preparation | **AI Modeling** | Simulation & Test | Deployment

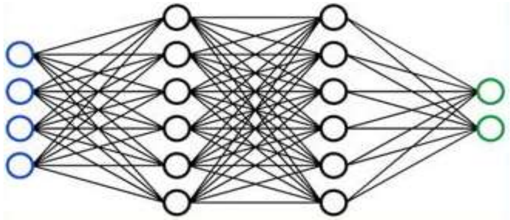
AI-based ROM using Neural State Space

Create DL-based nonlinear state-space models without having to be a deep learning expert

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \end{cases}$$



State Network (*f*)



Output Network (*g*)

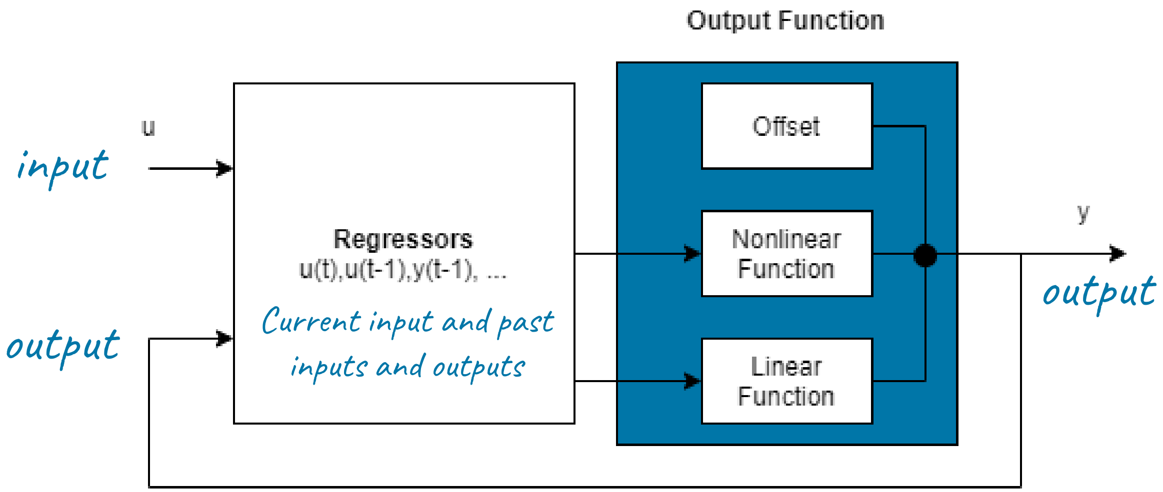
- The nonlinear state function f and nonlinear output function g are feedforward neural networks that learn from data
- Popularly known as Neural ODE in deep learning community

AI-based ROM using Nonlinear ARX

Surrogate Modeling using Sigmoid-based Nonlinear ARX Model

Nonlinear ARX (NLARX)

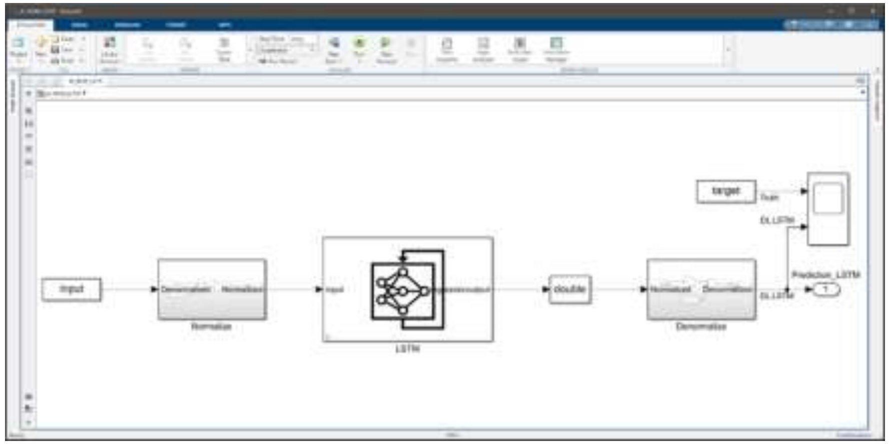
- Extends linear ARX to the nonlinear case
- Flexible choice of nonlinear functions
- May be more interpretable than Deep Learning models
- Potentially faster training and simulation



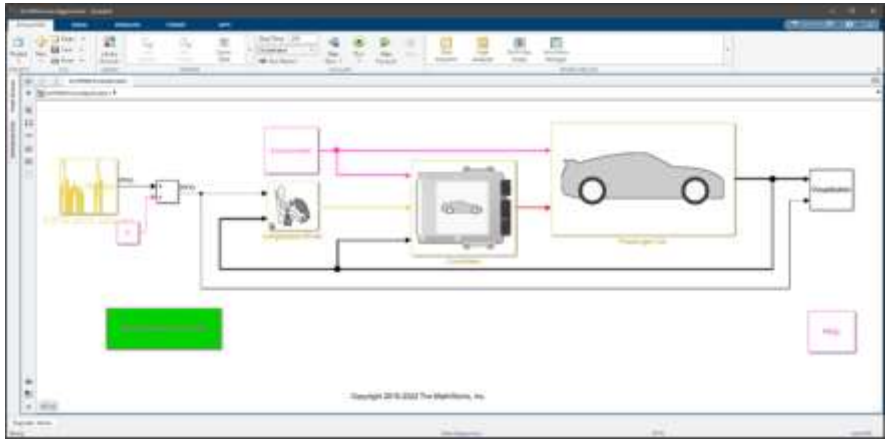
Data Preparation | **AI Modeling** | Simulation & Test | Deployment

Integrate Your AI Model for System-level Simulation and Test

Integration of trained AI model into Simulink

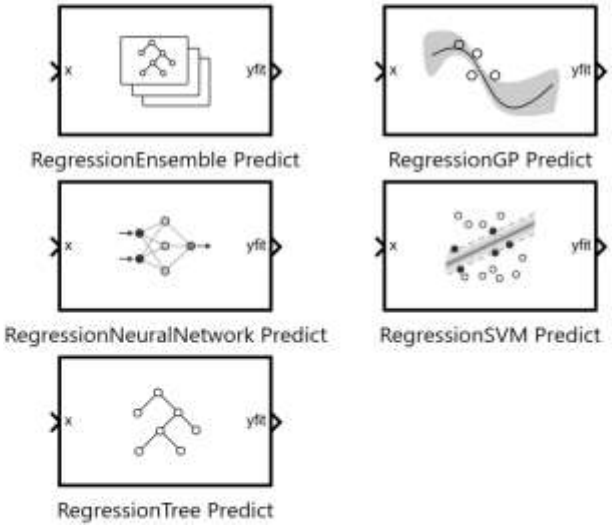


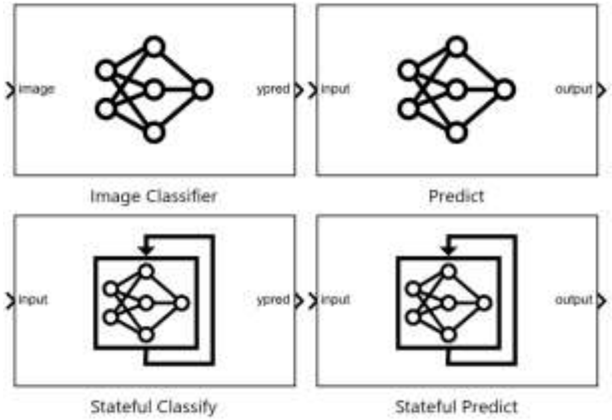
System-level simulation



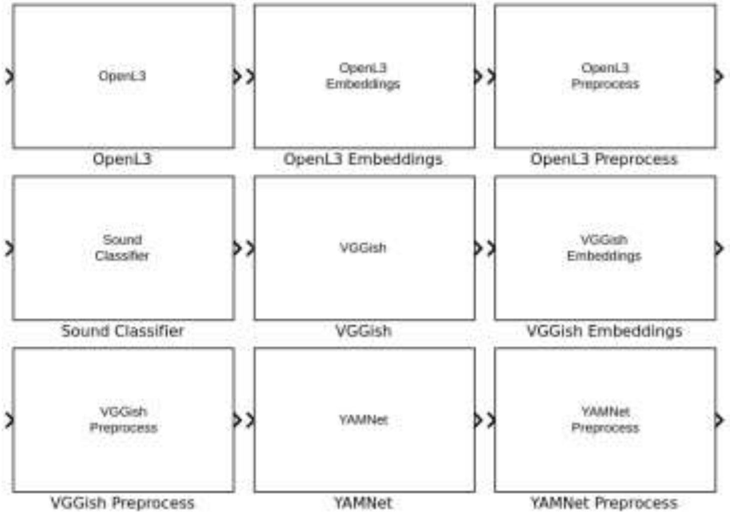
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

AI Libraries in Simulink Are Expanding to Include More AI Blocks for More Applications

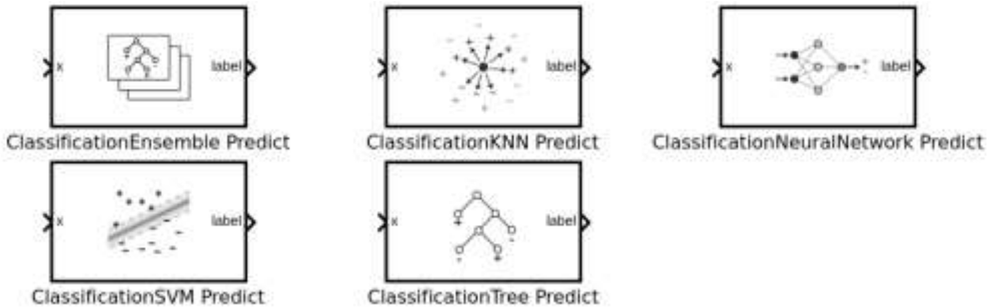


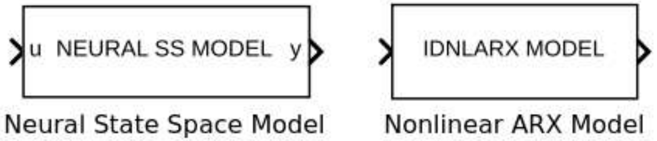


Deep Learning Toolbox

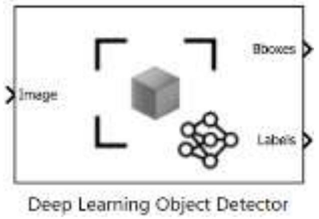


Audio Toolbox





System Identification Toolbox

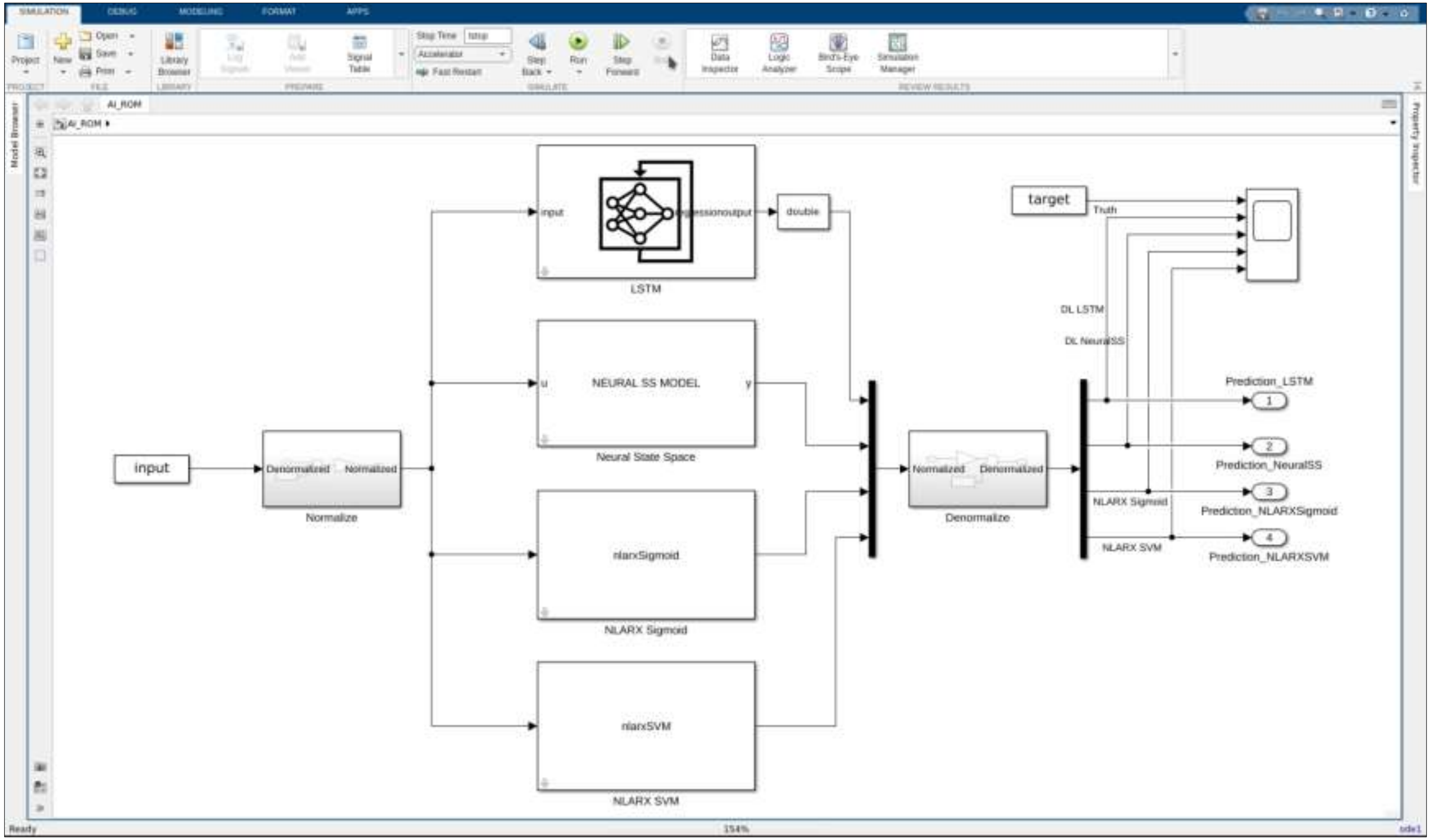


Computer Vision Toolbox

Statistics and Machine Learning Toolbox

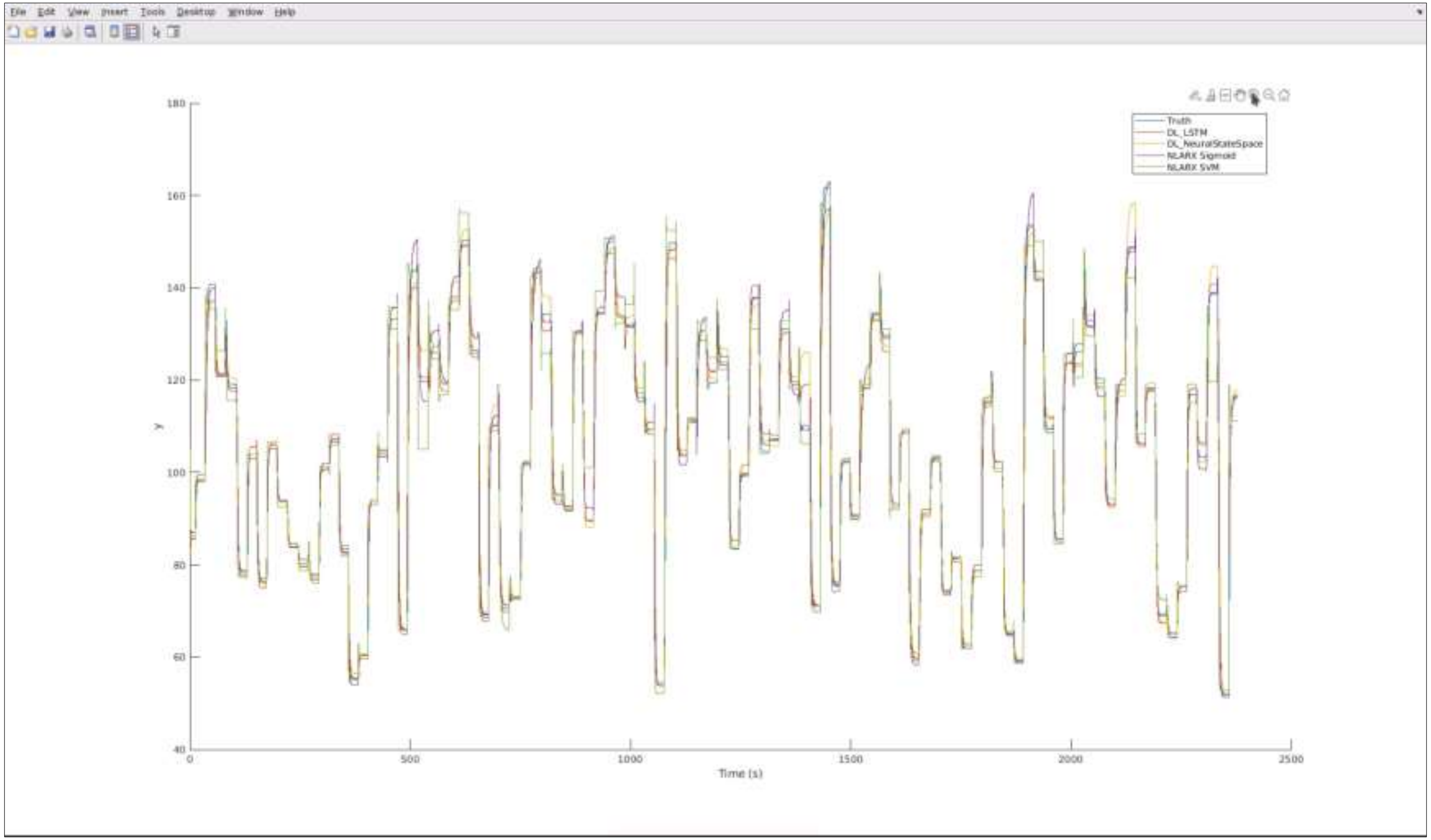
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Integration of trained AI models into Simulink



Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Integration of trained AI models into Simulink



Data Preparation | AI Modeling | **Simulation & Test** | Deployment

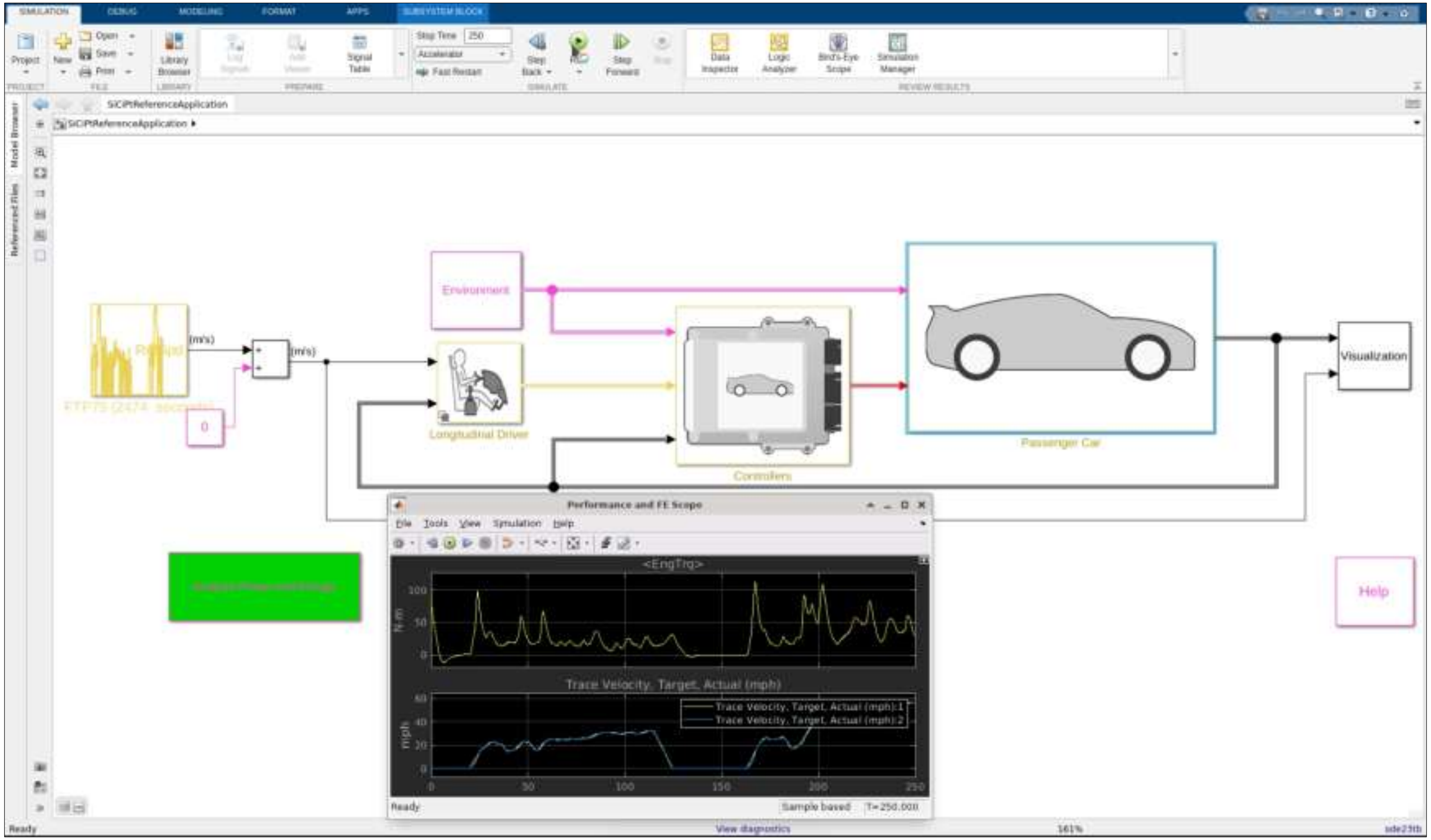
Integration of trained AI models into Simulink

Simulink Profiler

Path	Time Plot (Dark Band = Self Time)	Total Time (s)	Self Time (s)	Number of Calls
AI_ROM		49.440	45.732	142760
LSTM		2.643	0.000	0
NLARX Sigmoid		0.284	0.000	0
Neural State Space		0.195	0.000	0
Scope		0.188	0.188	23795
From Workspace2		0.161	0.161	23794
Demux		0.128	0.128	95184
From Workspace1		0.054	0.054	23794
Prediction_LSTM		0.040	0.040	23794
Prediction_NeuralSS		0.006	0.006	23794
Prediction_NLARXSigmoid		0.005	0.005	23794
Prediction_NLARXSVM		0.004	0.004	23794
NLARX SVM		0.001	0.000	0
Normalize		0.000	0.000	0
Cast To Double		0.000	0.000	3
Denormalize		0.000	0.000	0

Data Preparation | AI Modeling | **Simulation & Test** | Deployment

System-level Simulation



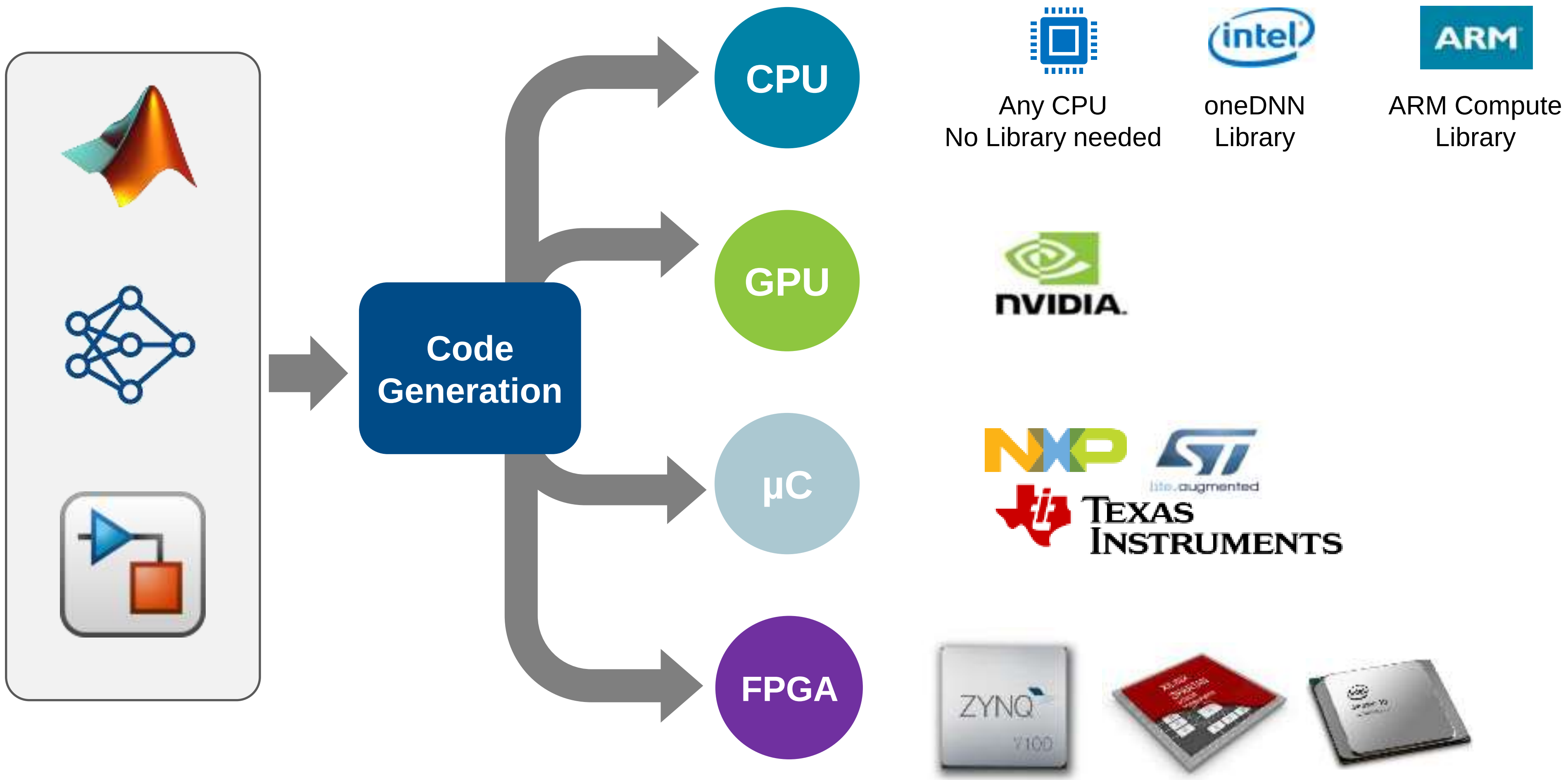
Data Preparation

AI Modeling

Simulation & Test

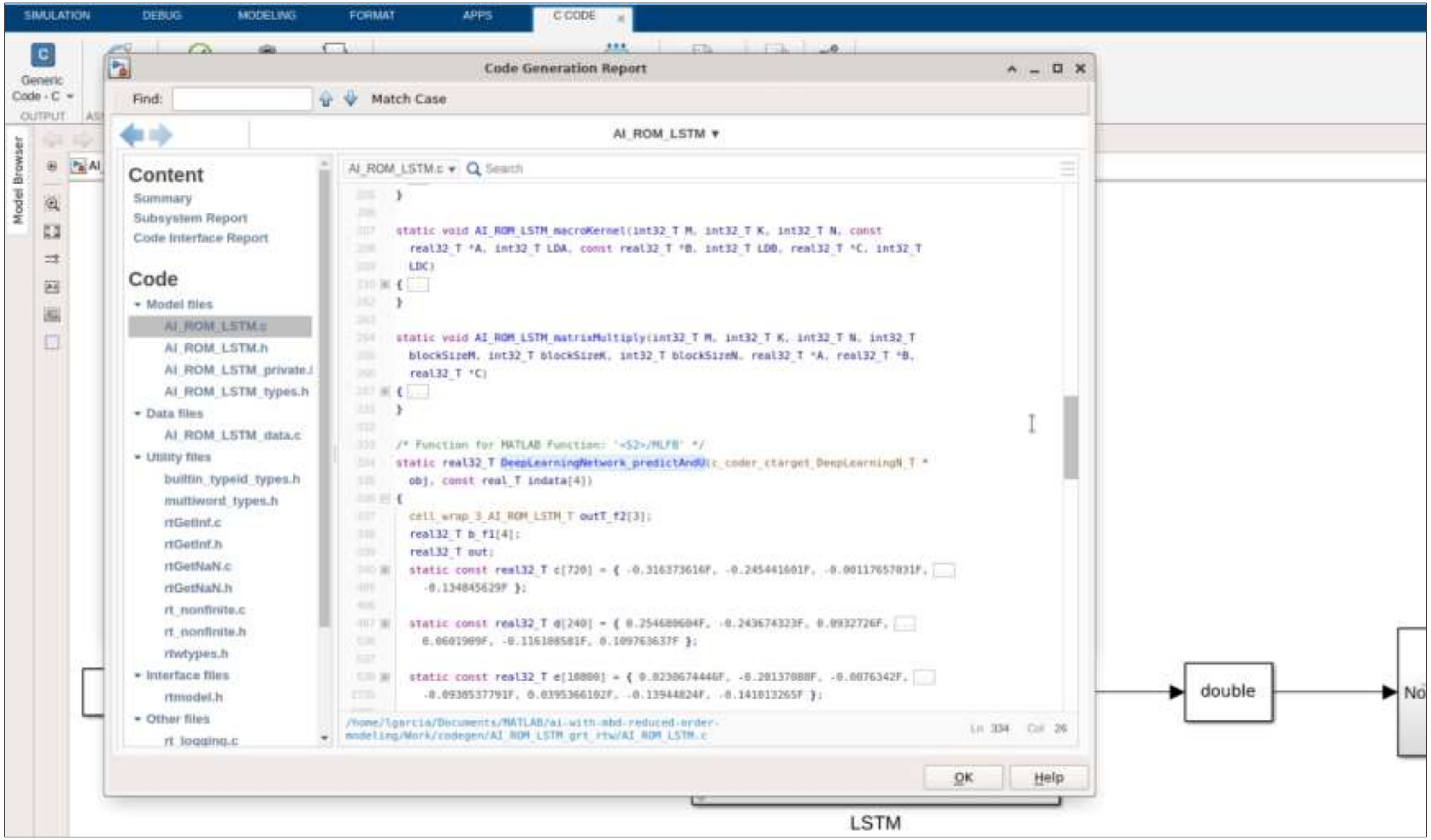
Deployment

Deploy to Target with Zero Coding Errors



Data Preparation | AI Modeling | Simulation & Test | **Deployment**

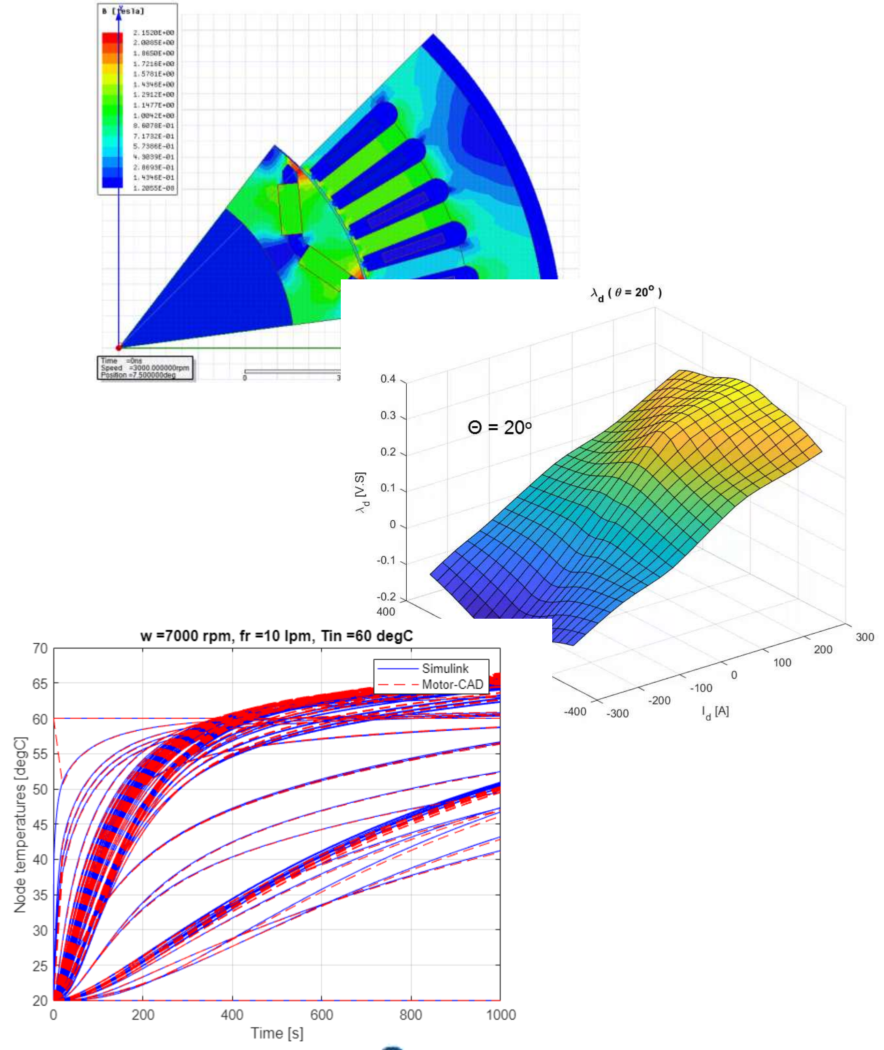
Generate C Code for Deep Learning Networks

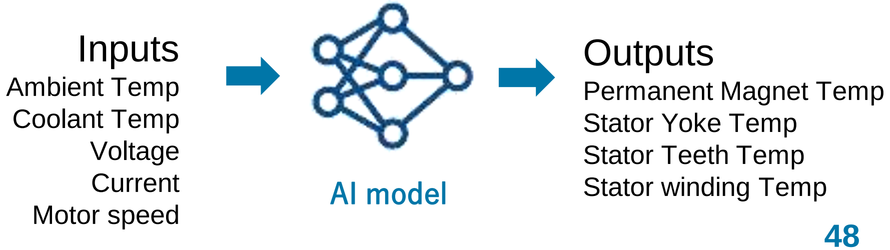


Data Preparation | AI Modeling | Simulation & Test | **Deployment**

Key Takeaways

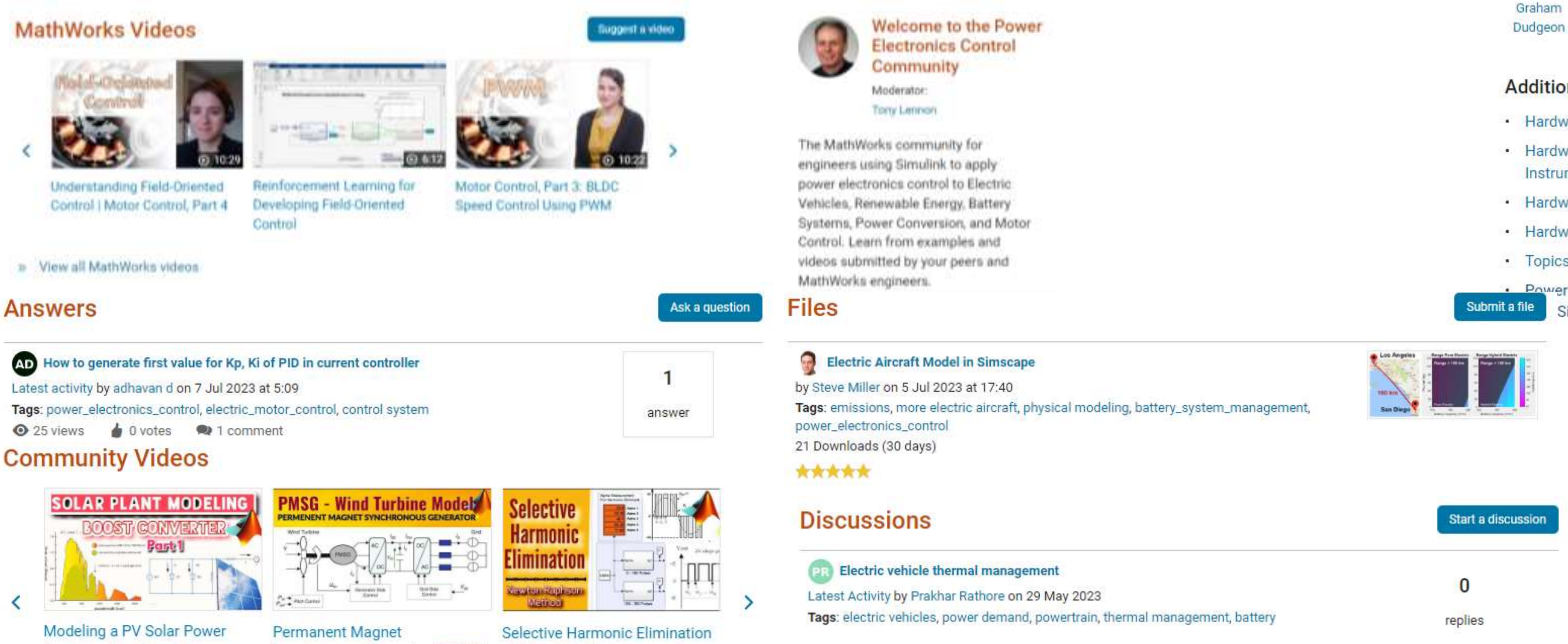
- Collaboration between Motor Design, and Motor Controller Design teams
- Use motor design data to understand system behavior, and design efficient control algorithms
- Reduce Order Model for faster simulation and to analyze the thermal losses
- Integrating the ROM with vehicle dynamics to speed up simulation
- Within MathWorks tools, AI-Based ROM enables data generation, AI modeling, simulation & testing, and deployment all in a single environment.

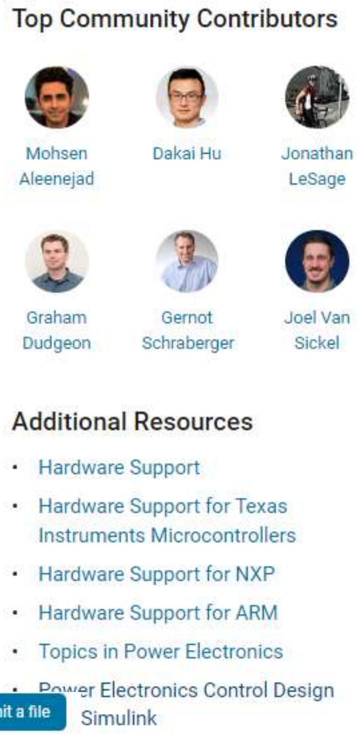




Visit the MathWorks Electrification Solutions page, MATLAB Central to find Models, Answers, and How-to Videos

https://www.mathworks.com/solutions/electrification/motor-drives-traction-motors.html

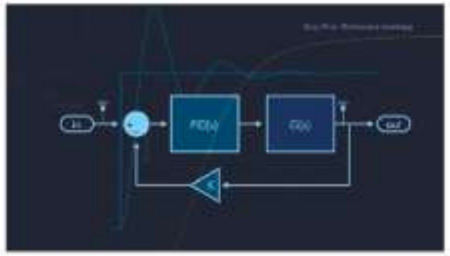




Enable Your Team on Motor Control

Power Electronics Control Design with Simulink and Simscape

Learn to model power electronic systems in the Simulink environment using Simscape Electrical™ and to design control with Simulink Control Design.

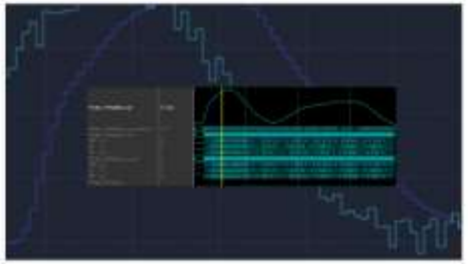


Control System Design with MATLAB and Simulink

Learn to design and model control systems with Simulink. Topics include system identification, parameter estimation, control system analysis, and response optimization.

Embedded Coder for Production Code Generation

Develop Simulink models for deployment in embedded systems. Topics include code structure and execution, code generation options and optimizations, and deploying code to target hardware.



Generating HDL Code from Simulink

Learn to prepare Simulink models for HDL code generation, generate HDL code and testbench for a compatible Simulink model, and perform speed and area optimizations.

Power Electronics Simulation Onramp

5 modules | 1 hour | Languages

Learn the basics of simulating power electronics converters in Simscape.

Circuit Simulation Onramp

7 modules | 2 hours | Languages

Learn the basics of simulating electrical circuits in Simscape.

Learn More

- Calibrating Optimal PMSM Torque Control with Field-Weakening Using Model-Based Calibration
- Field-Oriented Control of PMSMs with Simulink
- Simulate, Design, and Test Field-Weakening Control Design with Simulink
- MathWorks solution for Motor Drives and Traction Motors
- Import a Motor-CAD Thermal Model into Simulink and Simscape
- AI with Model-Based Design: Reduced Order Modeling