



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

강호석 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

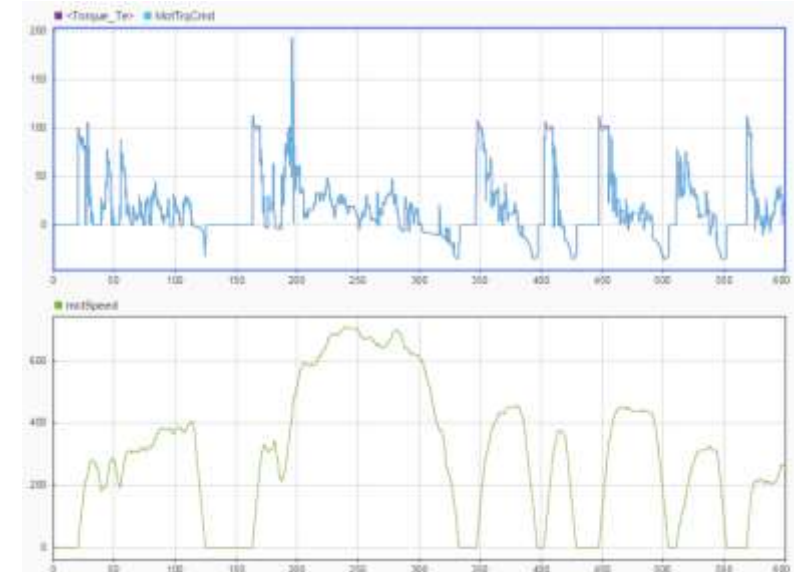
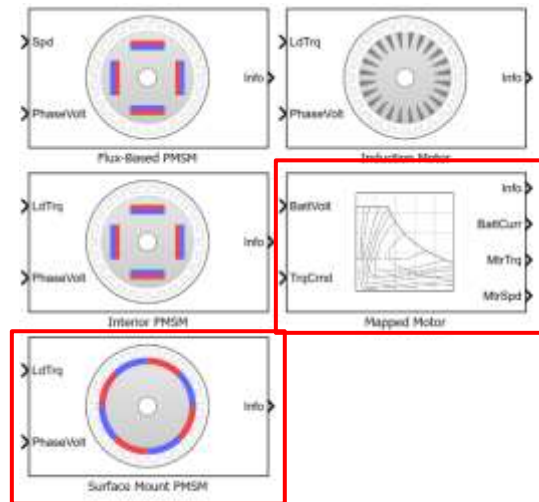
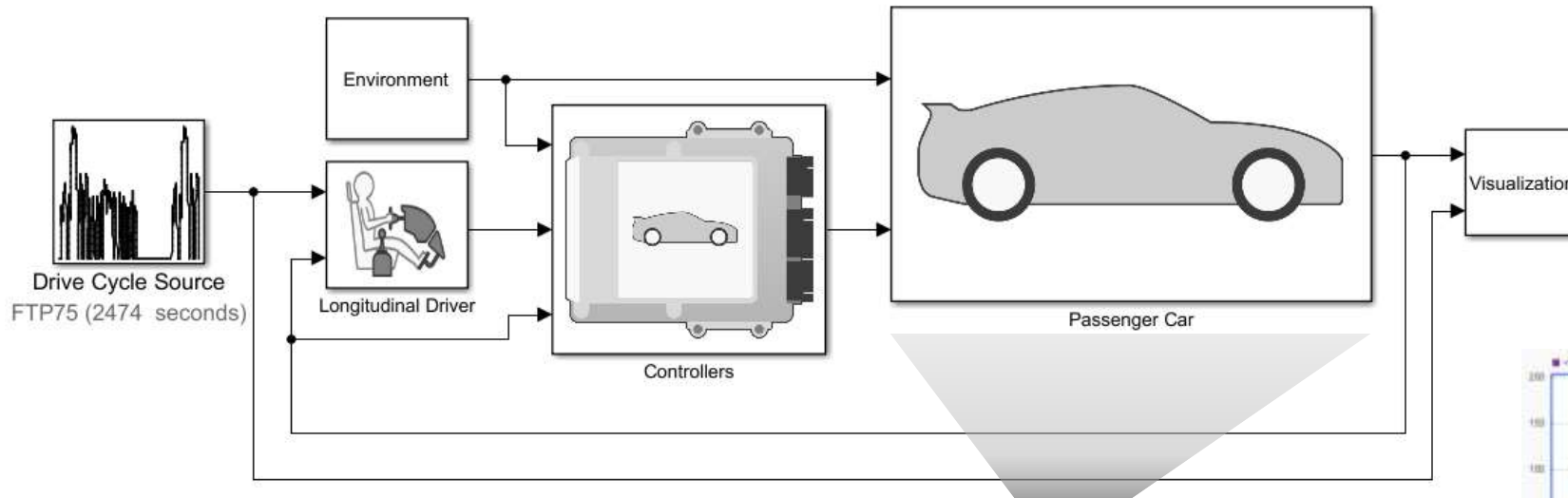
▪ **Motor Control Engineer**

- Designing closed-loop Control
- Operating point management (compute I_d , I_q 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?

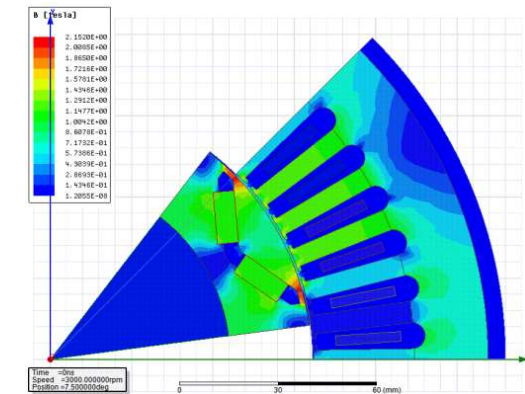
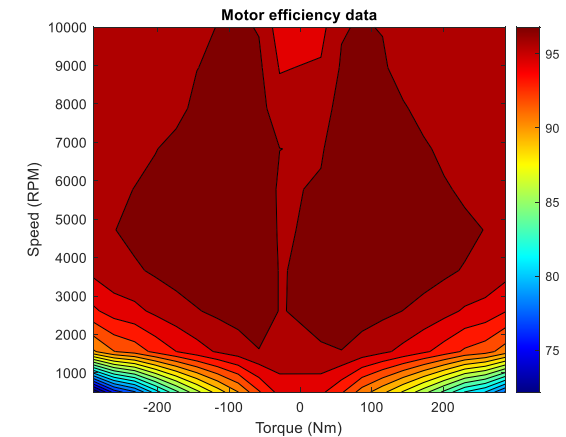
- Detailed Component Modeling and Control Design?



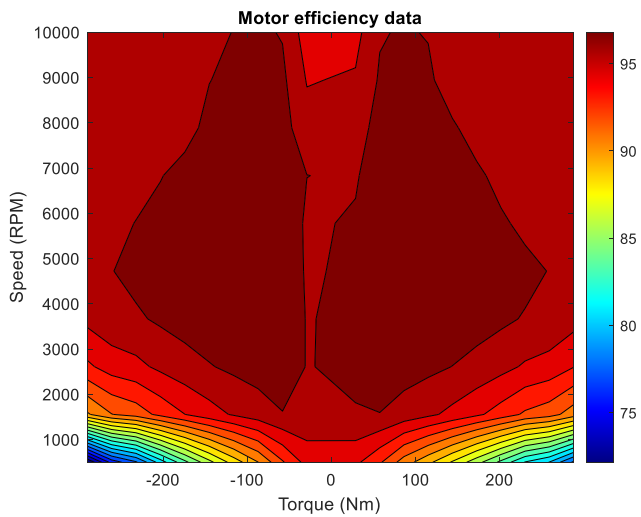
Workflow for Motor Design Engineers

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



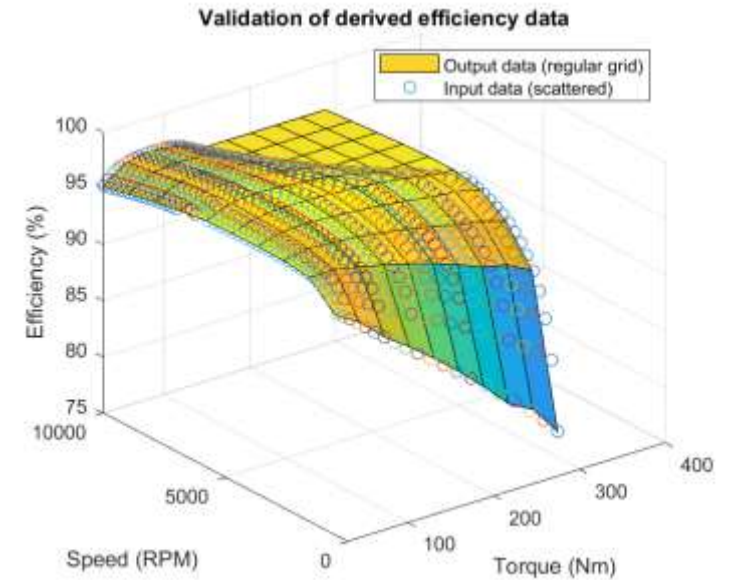
Motor efficiency data from a Motor Desing tool



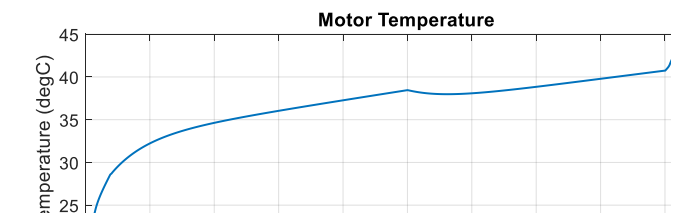
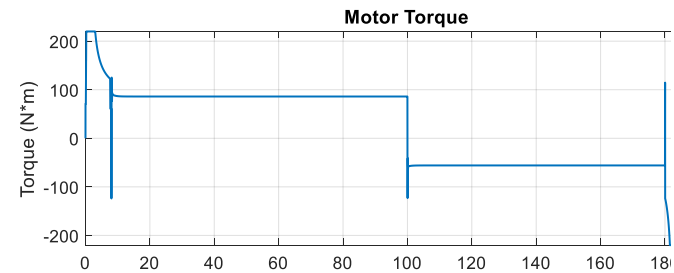
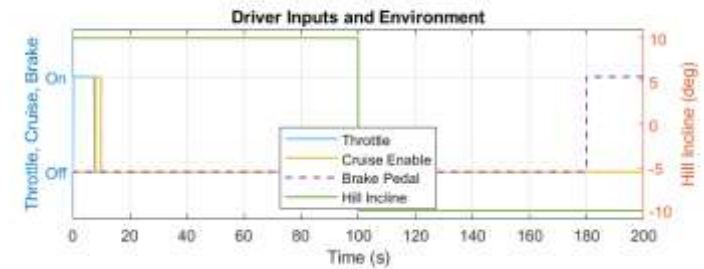
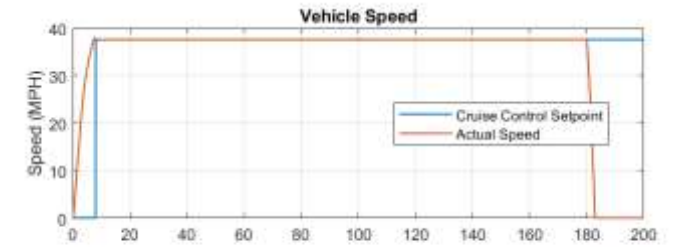
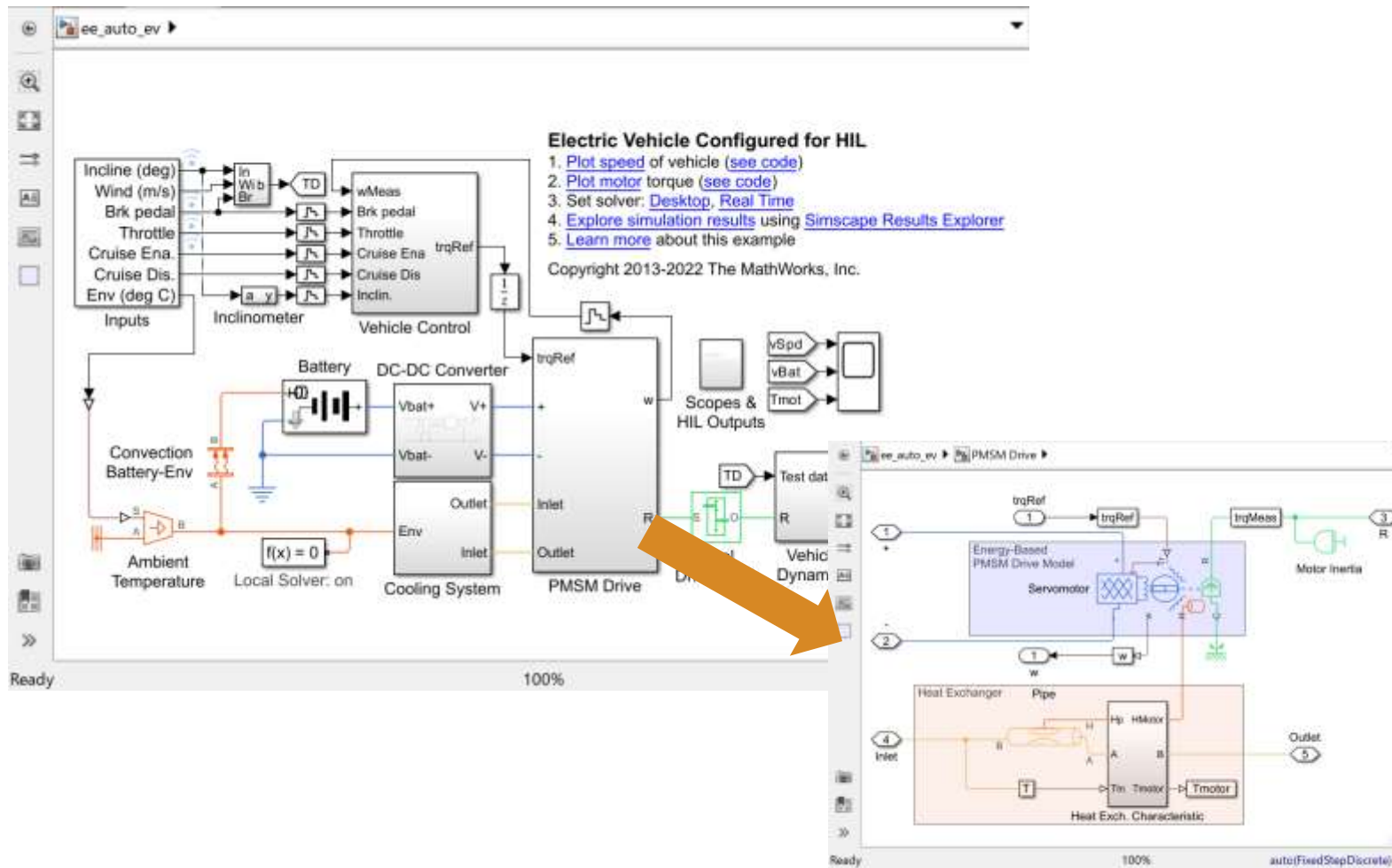
ee_import_efficiency_map_motorcad

Import Efficiency Map Data from Motor-CAD

Electrical Torque	
Parameterize by	Maximum torque and power
Allow intermittent over-torque	No
> Continuous operation maximum torque	tMax 288.1 N*m
> Continuous operation maximum power	pMax 1.3768e+05 W
> Torque control time constant, Tc	0.02 s
Electrical Losses	
Parameterize losses by	Tabulated efficiency data as a function of
Vector of speeds (ω) for tabulated losses	vVec <1x10 double> rpm
Vector of torques (T) for tabulated losses	tVec <1x20 double> N*m
Corresponding efficiency (percent), E(ω ,T)	eMat <10x20 double>



Importing Motor Efficiency Map to Simscape

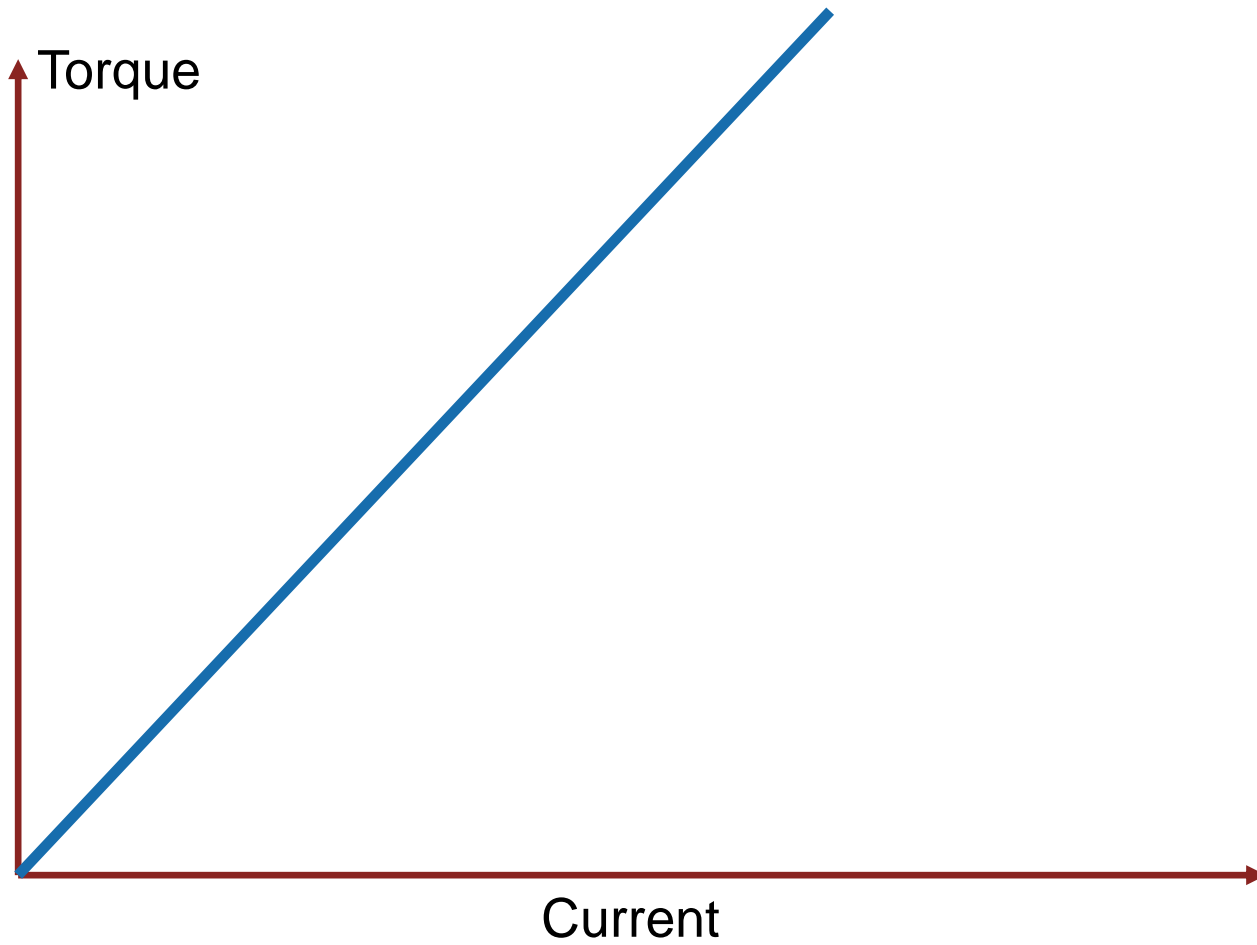


Workflow for Motor Control Engineers

- 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

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

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

$$\omega_e = p \omega_r$$

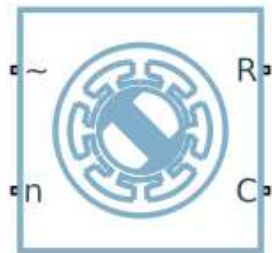
$$T_e = 1.5p[\lambda i_q + (L_d - L_q)i_d i_q]$$

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

Mechanical Model

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

Motor Parametrization in Simscape



PMSM



Block Parameters: PMSM

Settings Description

NAME	VALUE
Modeling option	No thermal port
Selected part	<click to select>
Main	
Electrical connection	Composite three-phase ports
Winding type	Wye-wound
Modeling fidelity	Constant Ld, Lq, and PM
> Number of pole pairs	6
> Permanent magnet flux linkage parameterization	Specify flux linkage
> Permanent magnet flux linkage	0.03 Wb
> Stator parameterization	Specify Ld, Lq, and L0
> Stator d-axis inductance, Ld	0.00019 H
> Stator q-axis inductance, Lq	0.00025 H
> Stator zero-sequence inductance, L0	0.00016 H
> Stator resistance per phase, Rs	0.013 Ohm
Zero sequence	Include
Rotor angle definition	Angle between the a-phase magnetic axis
Iron Losses	
Mechanical	
Initial Targets	
Nominal Values	

Block Parameterization Manager: PMSM

SELECT FORMAT

Apply all Reset all

Manufacturer: All

PARAMETERIZE

Select part

Part number	Manufacturer	RatedSpeed,rpm	PolePairs		
BSM132C-8200AA	ABB_BALDC	1800	4		
BSM33C-5177MHQ	ABB_BALDC	1800	4		
BSM50N-133	ABB_BALDC	4000	2		
BSM50N-275	ABB_BALDC	2000	2		
BSM63N-133	ABB_BALDC	4000	2		
HDS100-0206A	ABB_BALDC	3000	5		
HDS130-0817B	ABB_BALDC	2000	5		
HDS130-1829B	ABB_BALDOR	2900	18 000	1500	5
HDS160-2540B	ABB_BALDOR	4000	25 000	1500	5
HDS180-4876B	ABB_BALDOR	7600	48 000	1500	5
HDS180-1107A	ABB_BALDOR	1000	11 000	1000	4

Compare selected part with block

Parameter name	Parameterization	Override datasheet value	Part value-BSM132C
Main>Number of pole pairs	Datasheet derived	<input checked="" type="checkbox"/>	4
Main>Permanent magnet flux linkage	Datasheet derived	<input checked="" type="checkbox"/>	0.226230859562701
Main>Torque constant	Datasheet derived	<input checked="" type="checkbox"/>	0.916923438250803
Main>Back EMF constant	Datasheet derived	<input checked="" type="checkbox"/>	0.916923438250803
Main>Stator d-axis inductance, Ld	Datasheet derived	<input checked="" type="checkbox"/>	0.00115
Main>Stator q-axis inductance, Lq	Datasheet derived	<input checked="" type="checkbox"/>	0.00115
Main>Direct-axis current vector, iD	Parameter not set	<input type="checkbox"/>	[-200 0 200]
Main>Quadrature-axis current vector, iQ	Parameter not set	<input type="checkbox"/>	[-200 0 200]

Parameter Estimation by Running Instrumented Tests

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

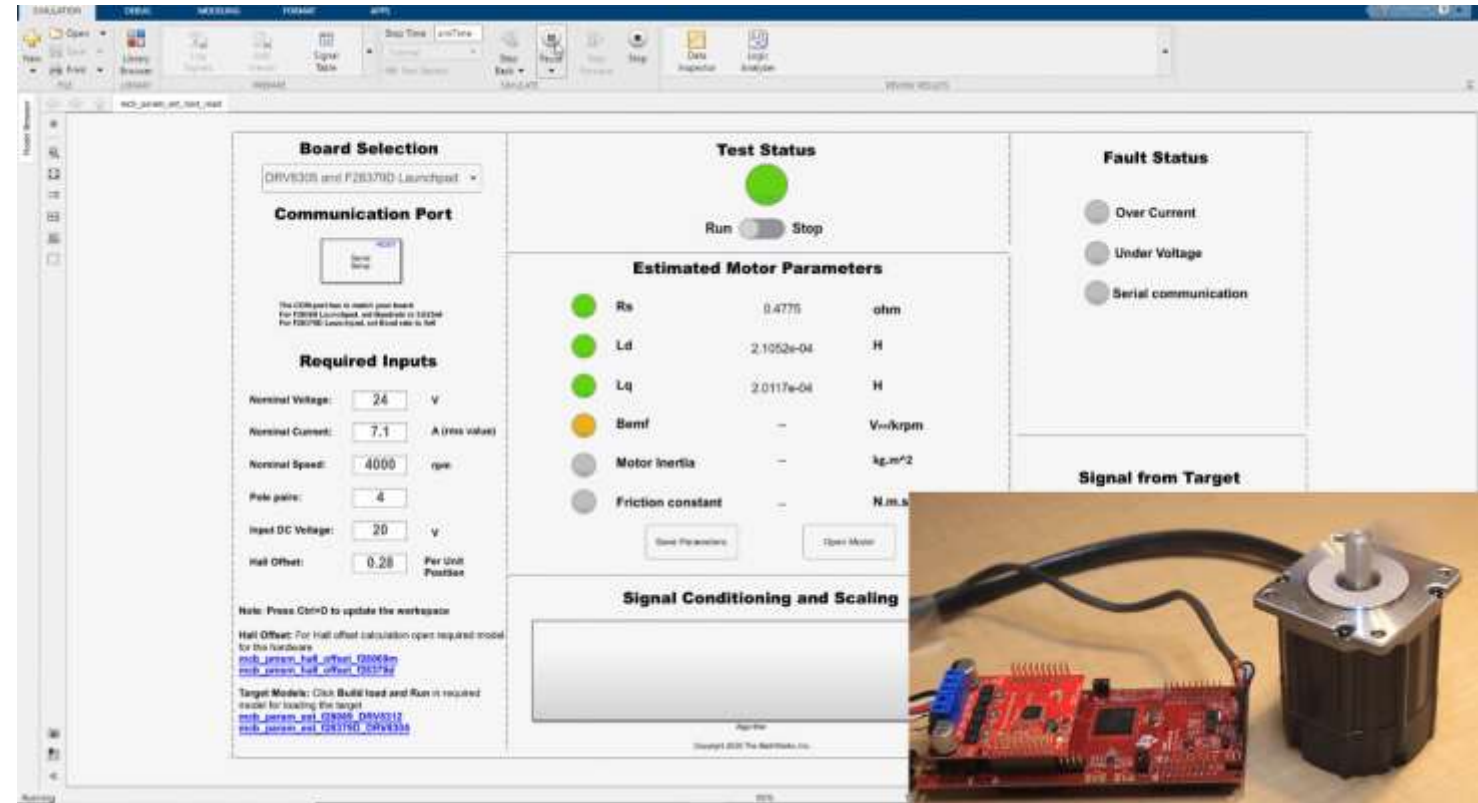
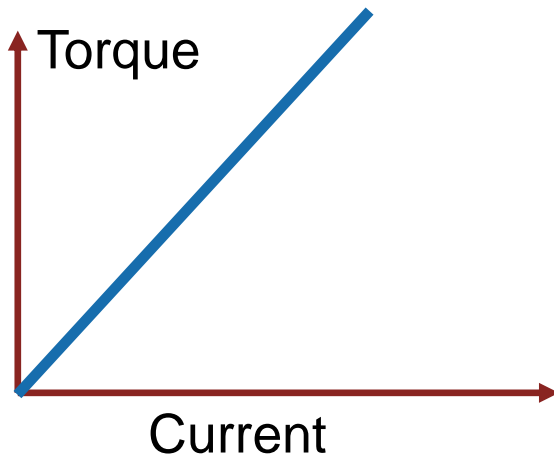
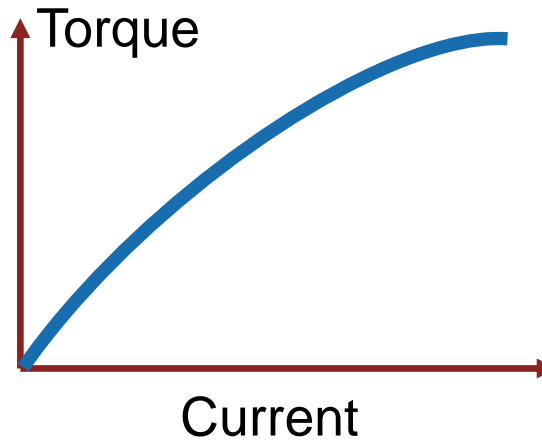


Figure : gif showing parameter estimation capability with TI C2000 hardware

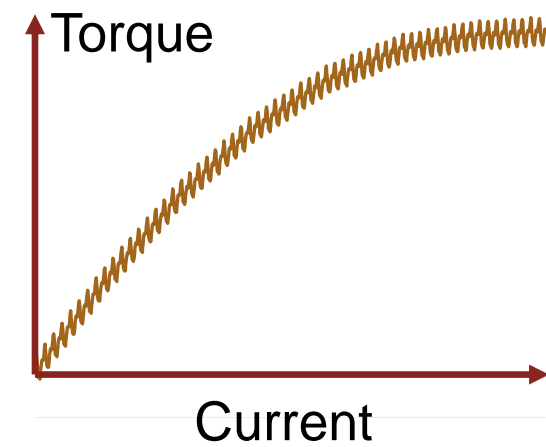
Motor Model Fidelity



Lumped Parameter



Saturation



Saturation +
Spatial Harmonics

Model Non-linear Effects such as Saturation and Spatial Harmonics

Electrical Model

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

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

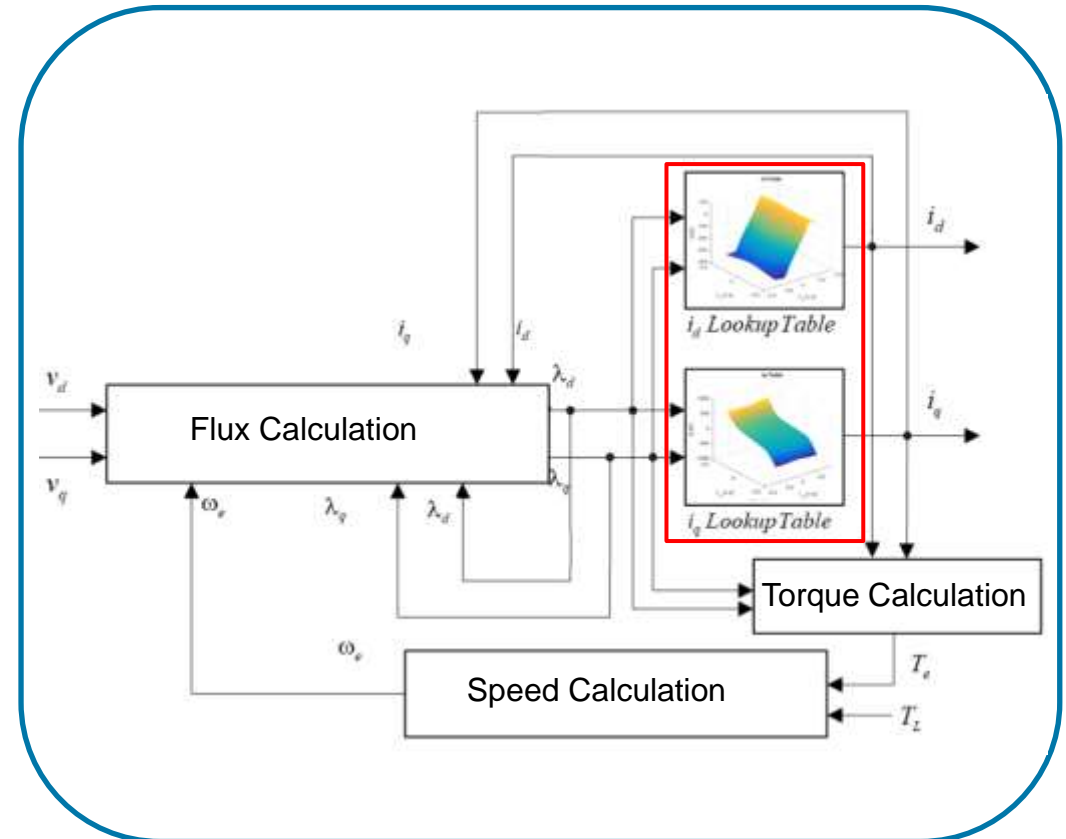
$$\omega_e = p \omega_r$$

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

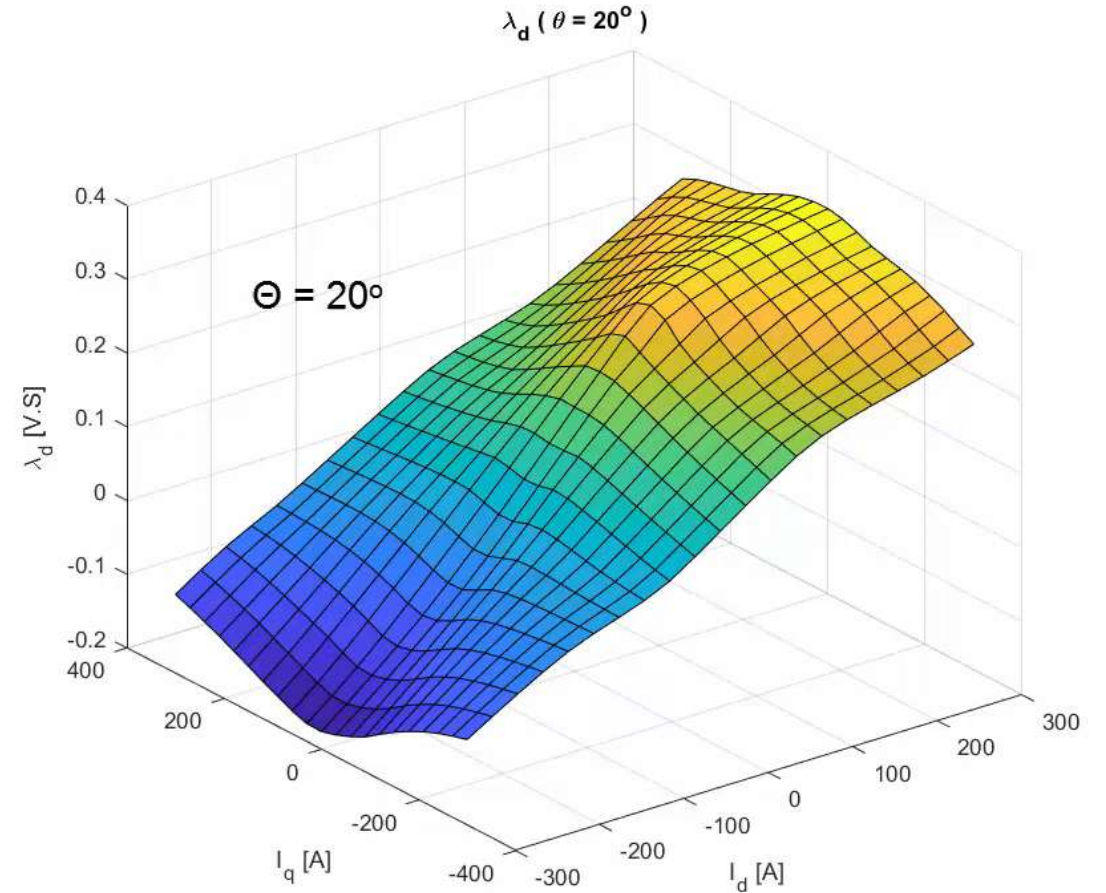
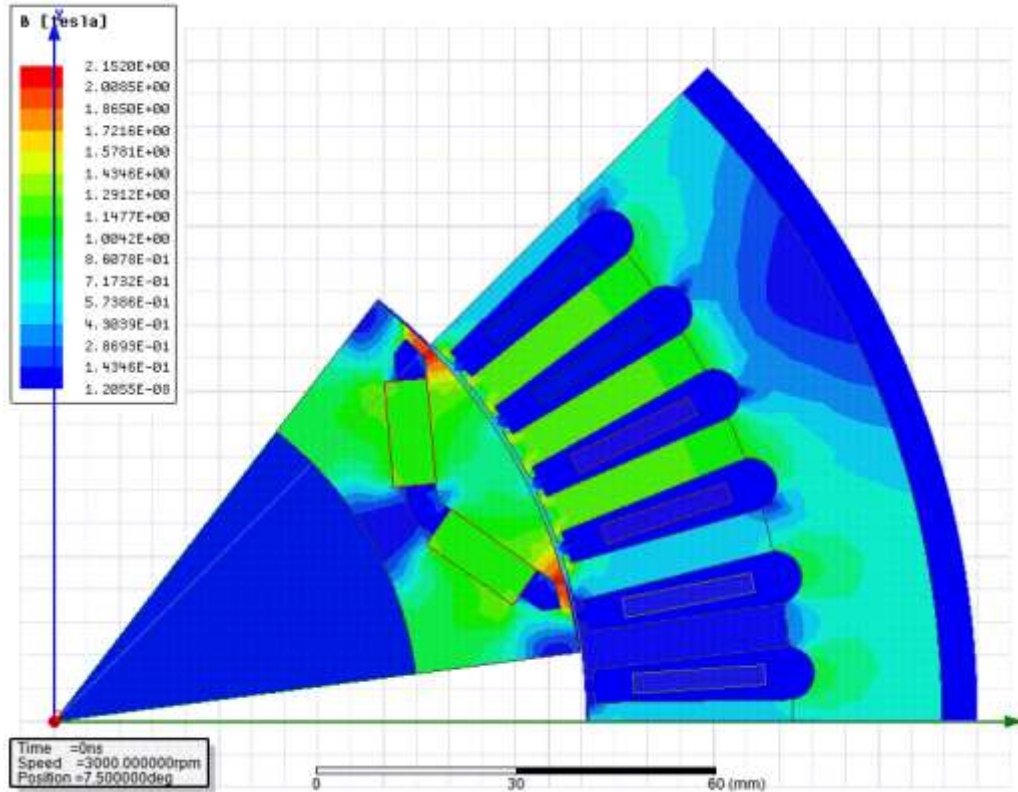
$$T_e = K_t i_q \quad (\text{assumes round rotor, } L_d = L_q)$$

Mechanical Model

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



Use Motor Design Tools Flux vs Current and Rotor Position Data

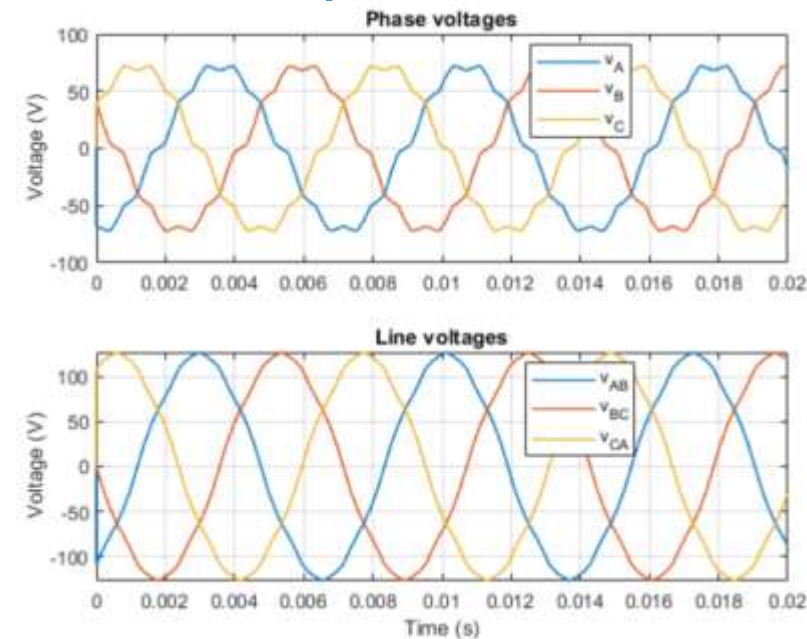


Importing Non-linearities and Loss Map in Simscape

Block Parameters: FEM-Parameterized PMSM

FEM-Parameterized PMSM Auto Apply

Settings	Description
NAME	VALUE
Modeling option	3-D flux linkage data No thermal port
Electrical	
Flux linkage data format	A-phase flux linkage as a function of peak curren
Winding type	Wye-wound
Expose neutral port	Yes
> Number of pole pairs	N
Park's convention for tabulated ...	Q leads D, rotor angle measured from A-phase t
> Peak current magnitude vector, I	magVec <1x25 double> A
> Current advance angle vector, B	gammaVec <1x37 double> deg
> Rotor angle vector, theta	angleVec <1x37 double> deg
> A-phase flux linkage, F(I,B,theta)	fluxAmat <25x37x37 dou...> Wb
> Torque matrix, T(I,B,theta)	torqueMat <25x37x37 do...> N*m

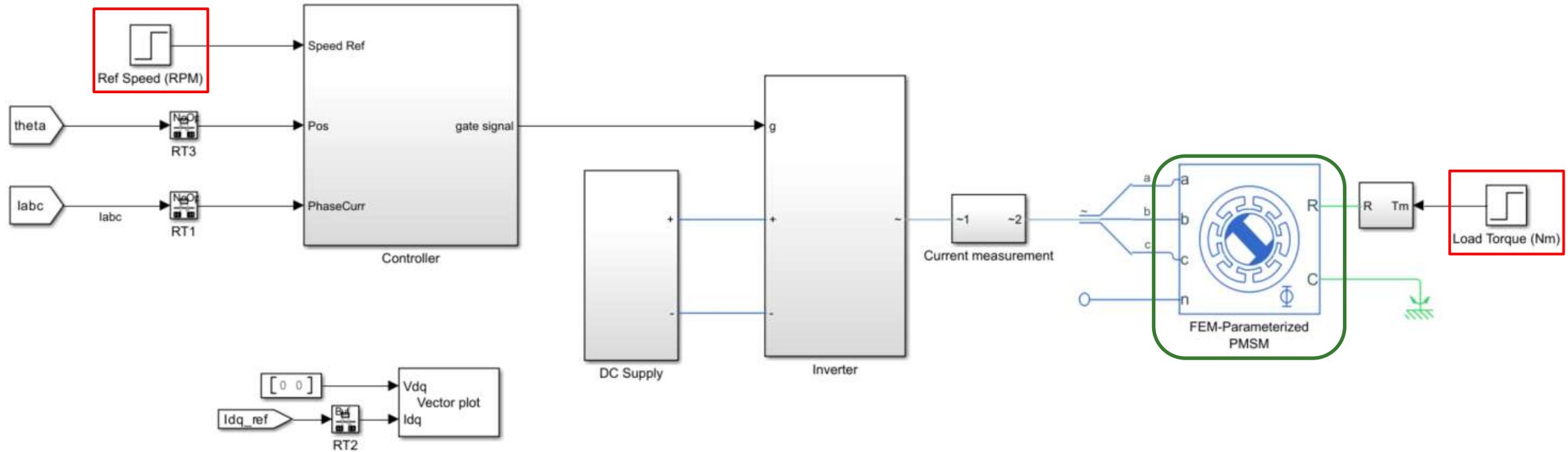


```

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



Copyright 2022-2023 The MathWorks, Inc.

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

Block Parameters: LUT based PMSM Control Reference

LUT based PMSM Control Reference (mask) (link)

Generate LUT based PMSM reference currents for field oriented control of Surface & Interior PMSMs.

The block accepts reference torque and feedback speed. The block outputs corresponding d-axis & q-axis reference currents.

Enter the motor parameters and input/output signal units in the dialog box below.

Motor parameters Input units

Motor topology Interior PMSM

Number of pole pairs: `pmsm_p` 2

Stator resistance per phase (Ohm): `pmsm_Rs` 0.22

Motor parameter input method: Non-linear model with id and iq LUTs

Vdc input method: Specify via dialog

DC voltage (V): `inverter_V_dc` 600

Reference torque breakpoint vector, `Tref` (Nm): `PMSMLUT.trefVec` <1x51 double>

Mechanical speed breakpoint vector, `wrpm` (rpm): `PMSMLUT.wrpmVec` <1x128 double>

Reference id LUT, `id`(`Tref`,`wrpm`) (A): `PMSMLUT.idTable` <51x128 double>

Reference iq LUT, `iq`(`Tref`,`wrpm`) (A): `PMSMLUT.iqTable` <51x128 double>

Steps:

1. To use a custom motor's FEM data, use the [introduction](#)
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.

Copyright 2022-2023 The MathWorks, Inc.

Steps:

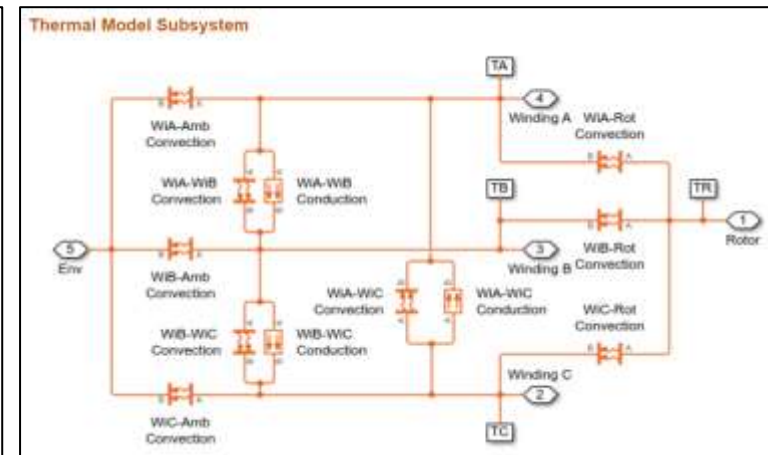
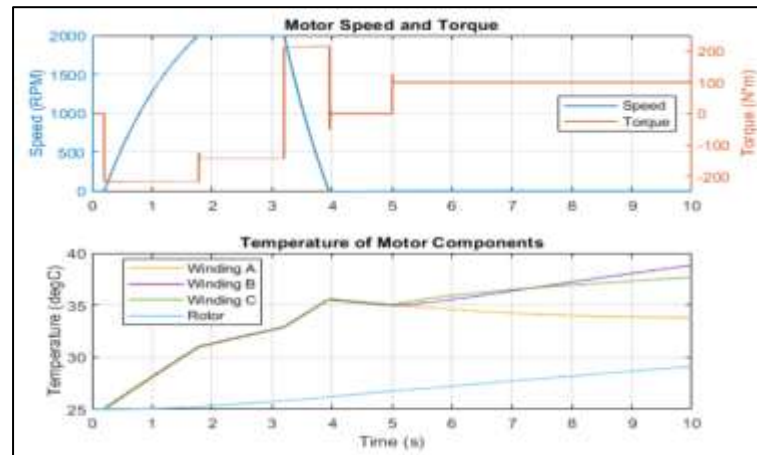
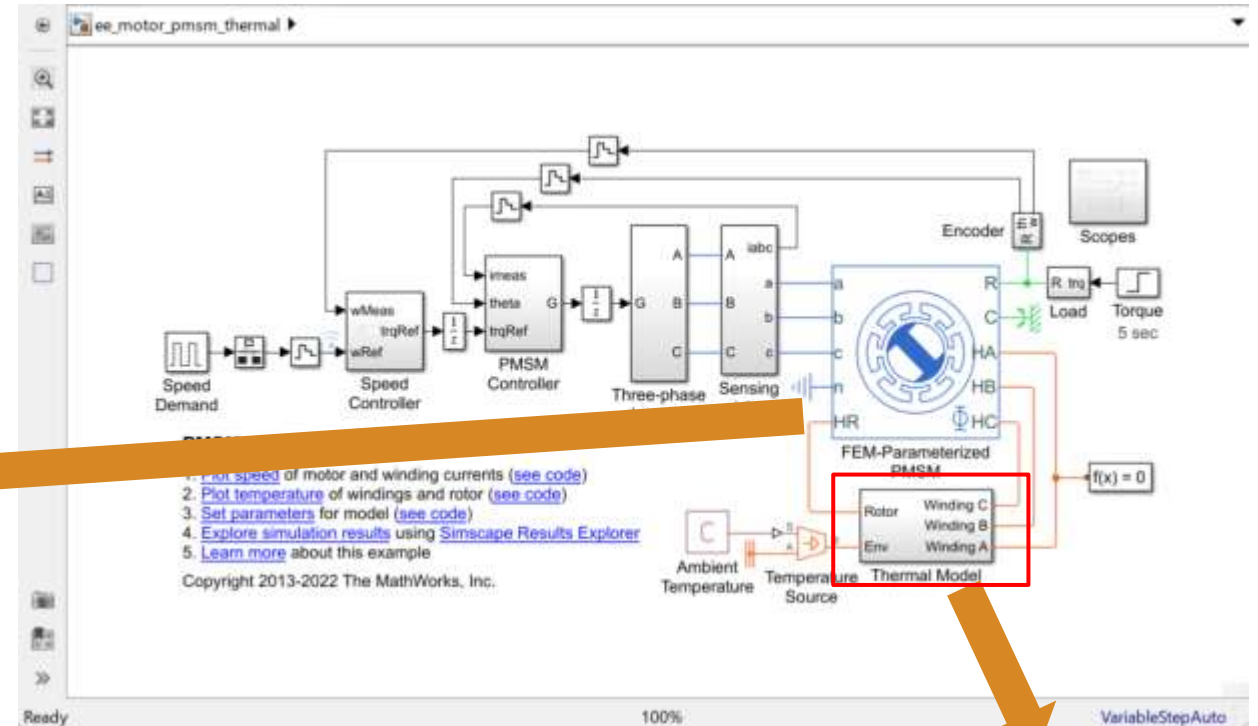
1. To use a custom motor's FEM data, use the [introduction](#)
2. Select the motor plant model from the PlantSelection options
3. Simulate the model
4. Review results in Data Inspector
5. [Learn more](#) about this example.

[Tune PI Controllers Using Field Oriented Control Autotuner](#)

Motor with Thermal Model

FEM-Parameterized PMSM Auto Apply

Settings	Description	VALUE
Modeling option	3-D flux linkage data Show thermal port	
> Electrical		
> Iron Losses		
> Mechanical		
> Temperature Dependence		
Measurement temperature	298.15	K
Resistance temperature coefficient	3.93e-3	1/K
Permanent magnet flux temperatur...	-0.001	1/K
> Thermal Port		
Thermal mass for each stator windi...	100	J/K
Initial stator winding temperatures	[298.15, 298.15, 298.15]	K
Rotor thermal mass	200	J/K
Rotor initial temperature	298.15	K
Percentage of main flux path iron l...	90	
Percentage of cross-tooth flux path ...	30	



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

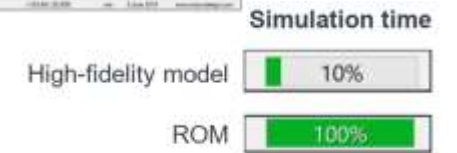
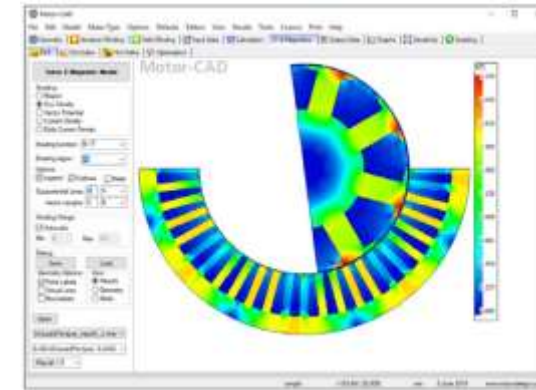
What

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

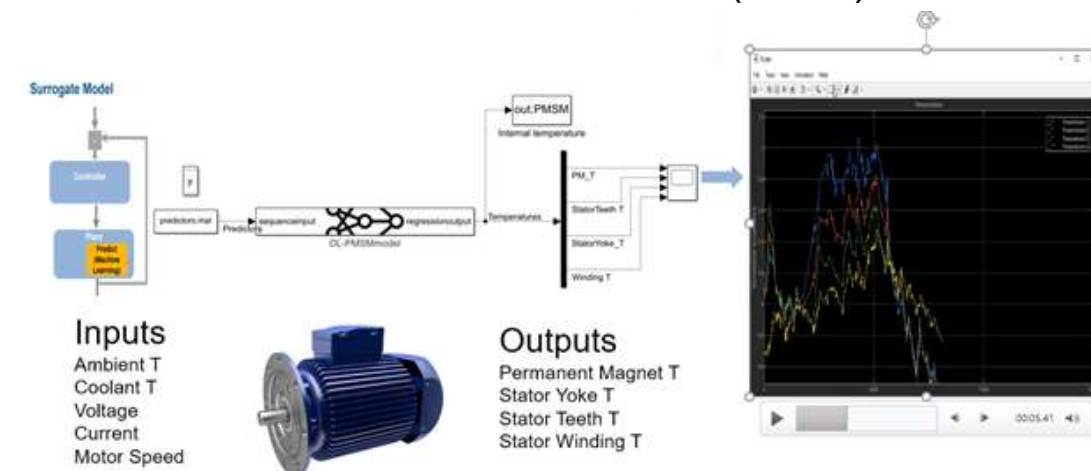
Why

- 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

**AI-Based
Data-driven**

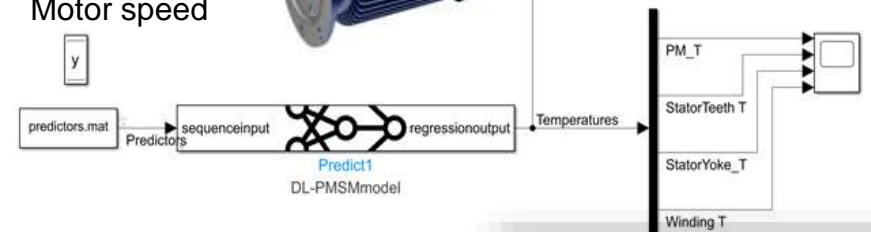
Input

Ambient T
Coolant T
Voltage
Current
Motor speed



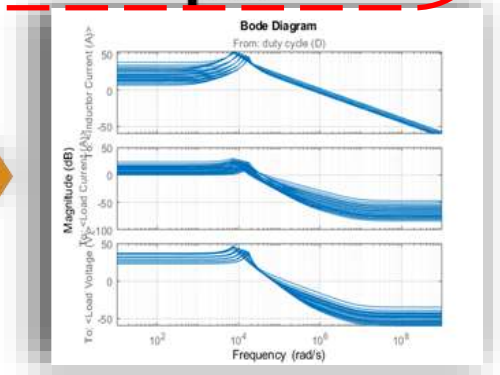
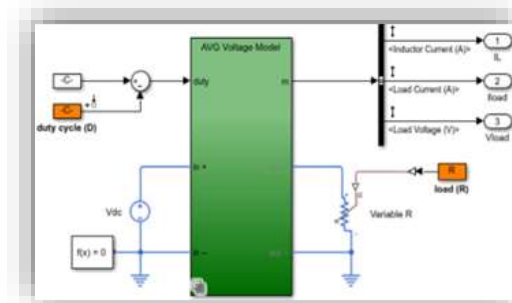
Output

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



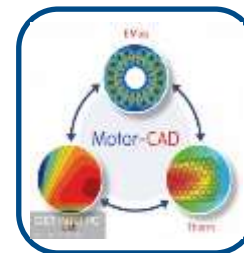
**Reduced order
model**

Linearization

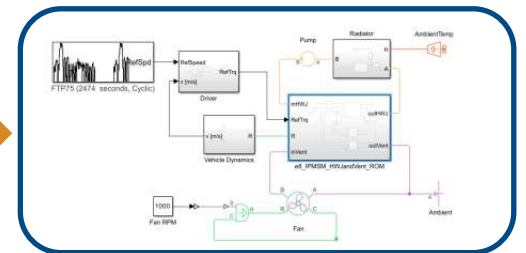


Model-based

Motor-CAD



**Integrating with detailed
Simscape model**



Workflow for Simulink Reduced Order Thermal Modeling (SROTM)

1

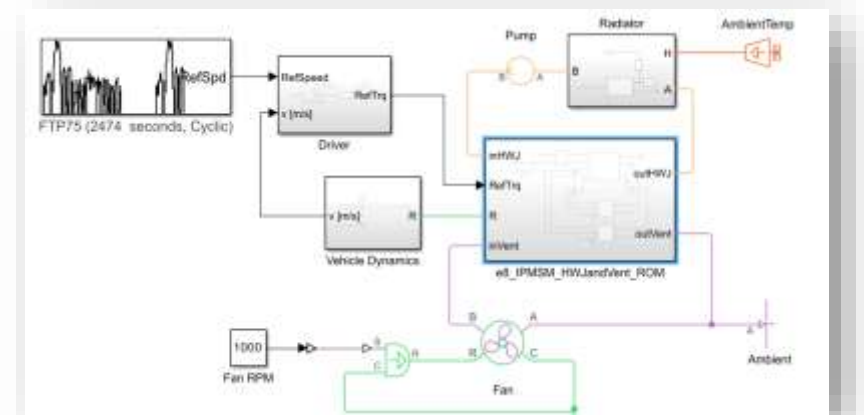
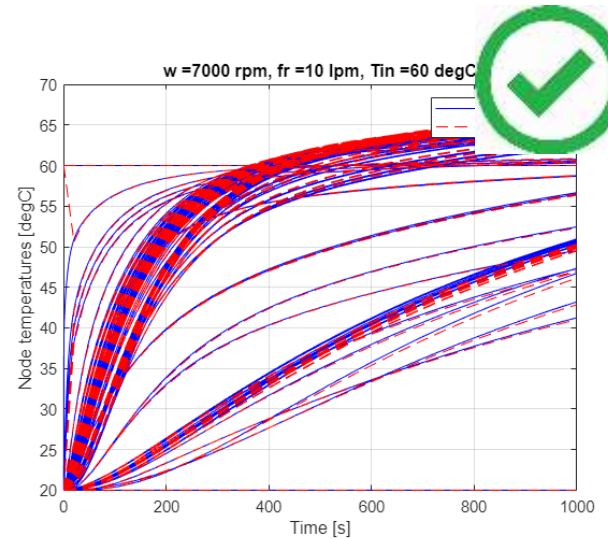
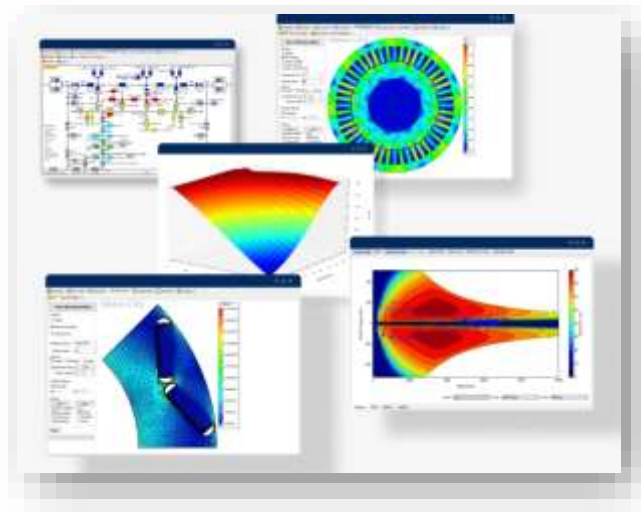
Generate a SROTM through Motor design tools

2

Validate the SROTM

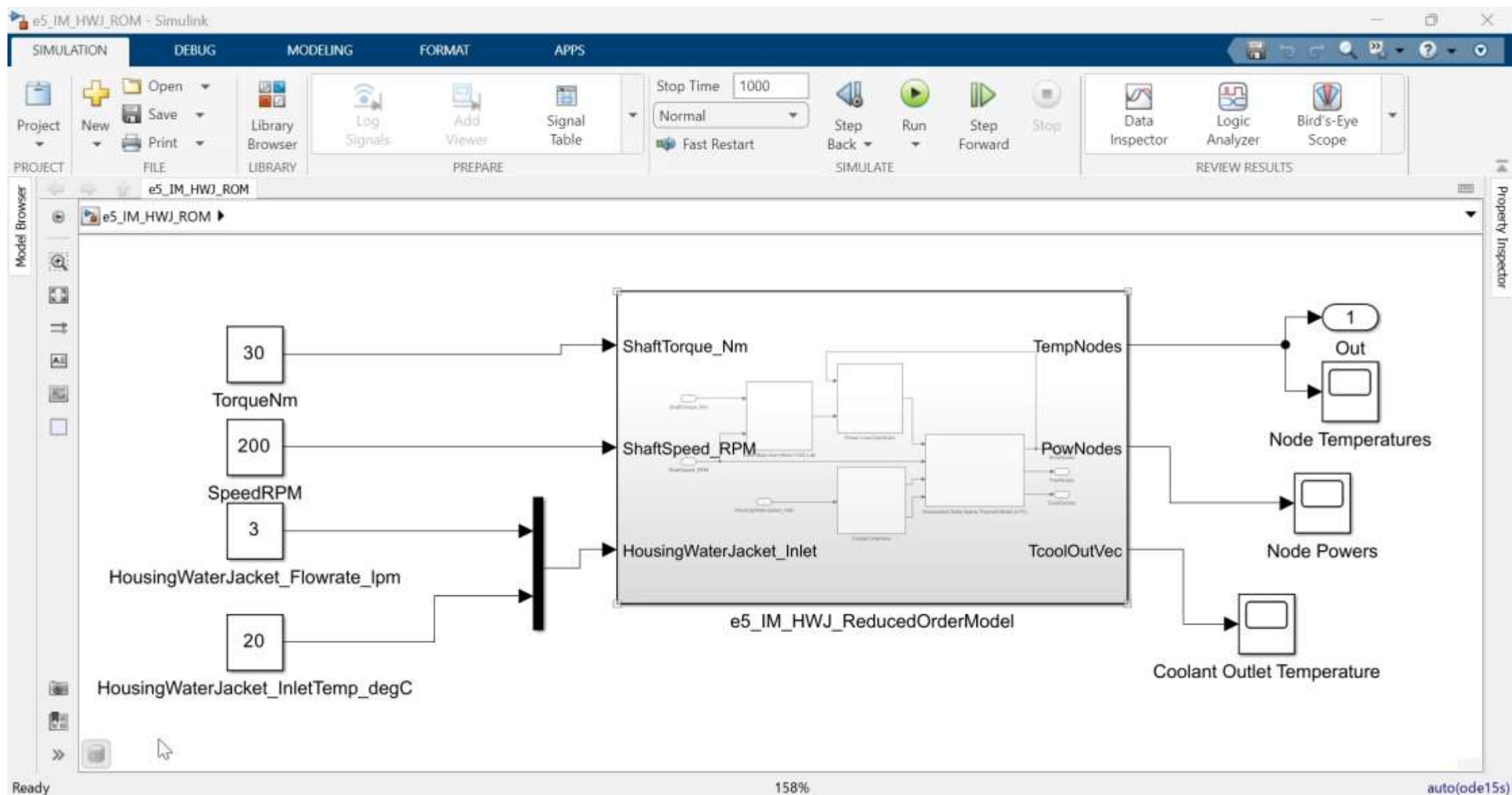
3

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:

\mathbf{x} is the vector of node temperatures (the states), T_{nodes}

\mathbf{p} is the vector of node losses, $node_{loss}$

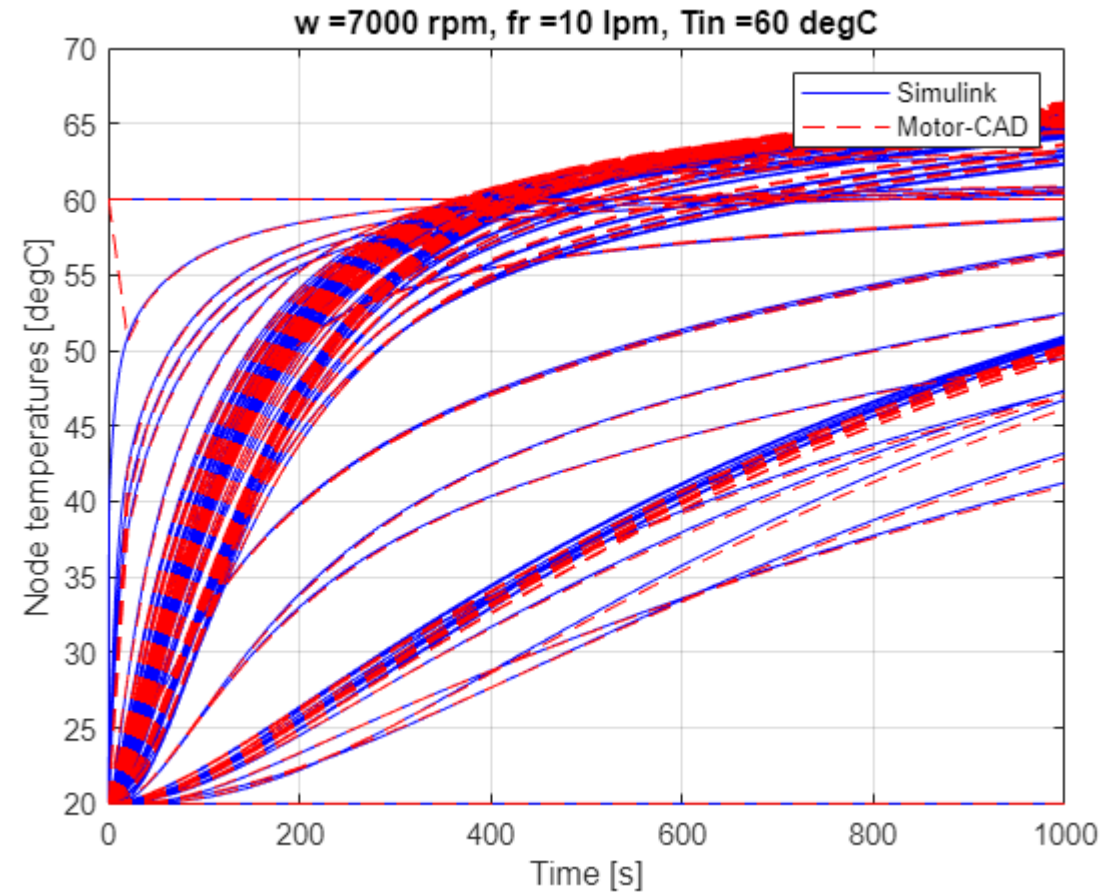
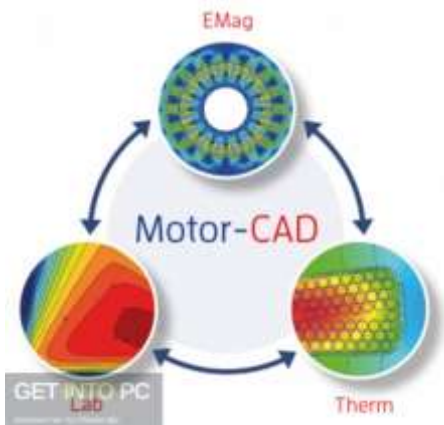
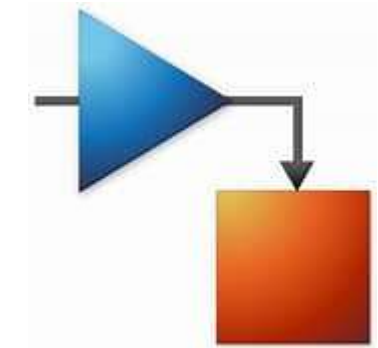
\mathbf{A} and \mathbf{B} are interpolated from a set of state-space matrices at different operating points (**speed, coolant flow rates, and coolant inlet temperatures**):

$$A = A_{fun}(w, fr, Tin)$$

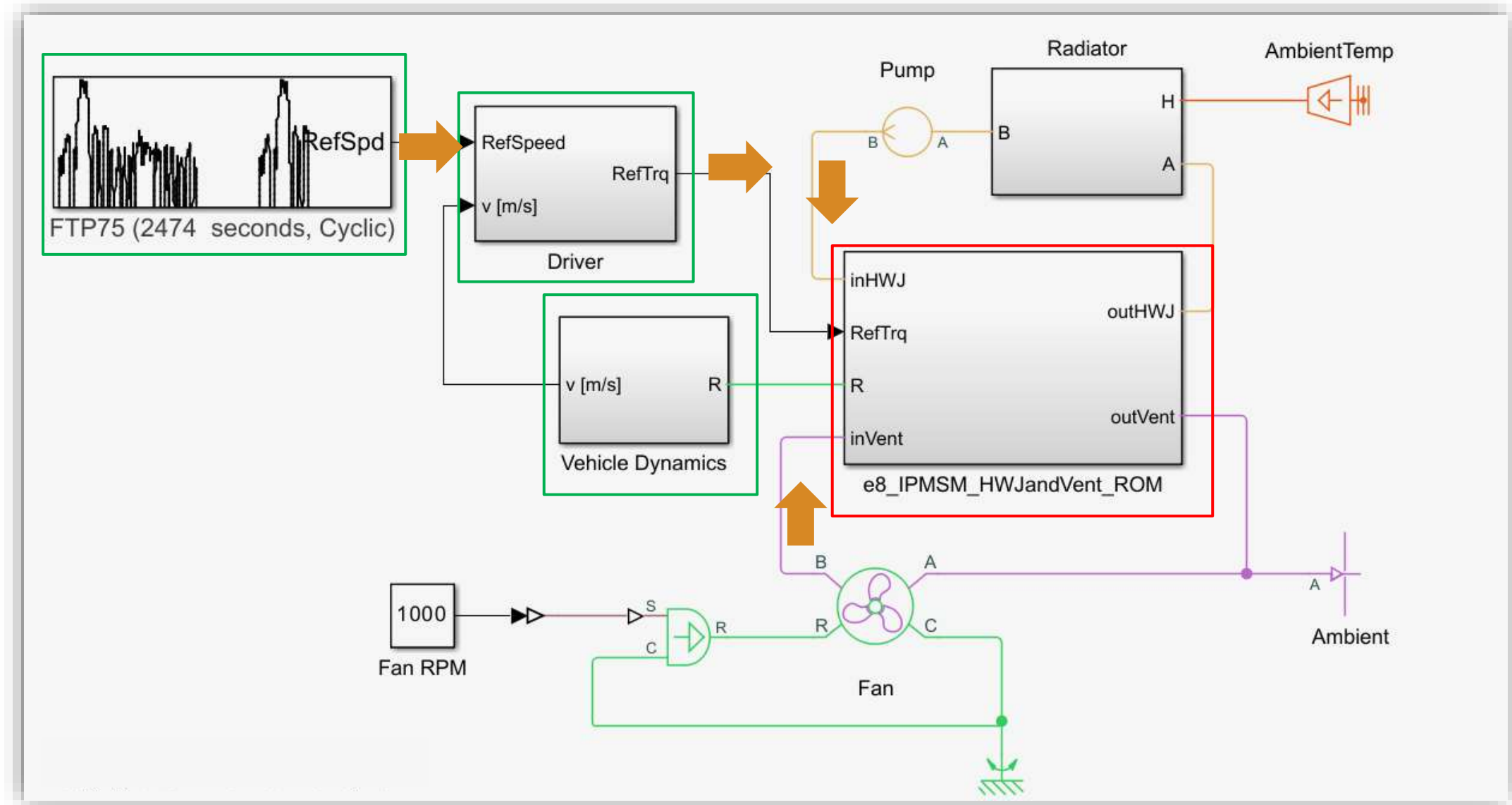
$$B = B_{fun}(w, fr, Tin)$$

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

Comparison between Simulink and Motor-Design Tool

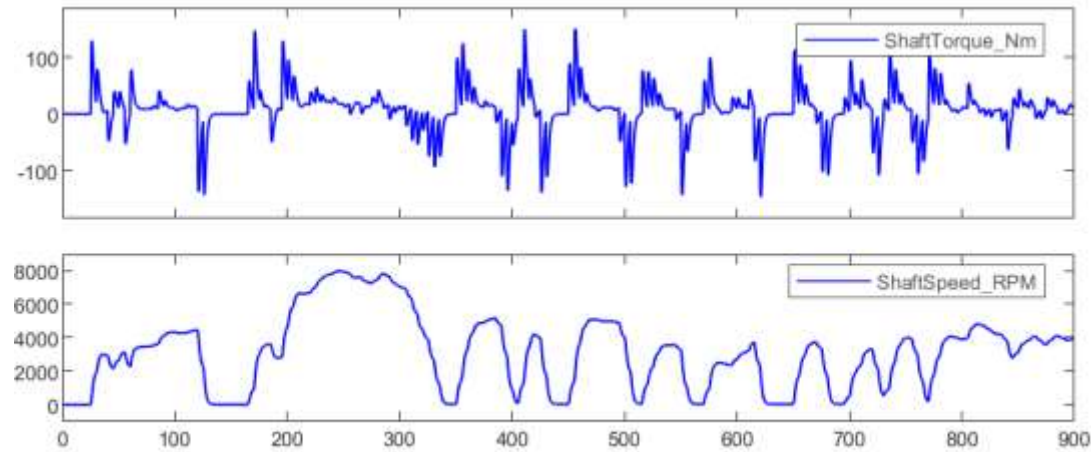


Integrate SROTM with Simscape System-level Vehicle Model

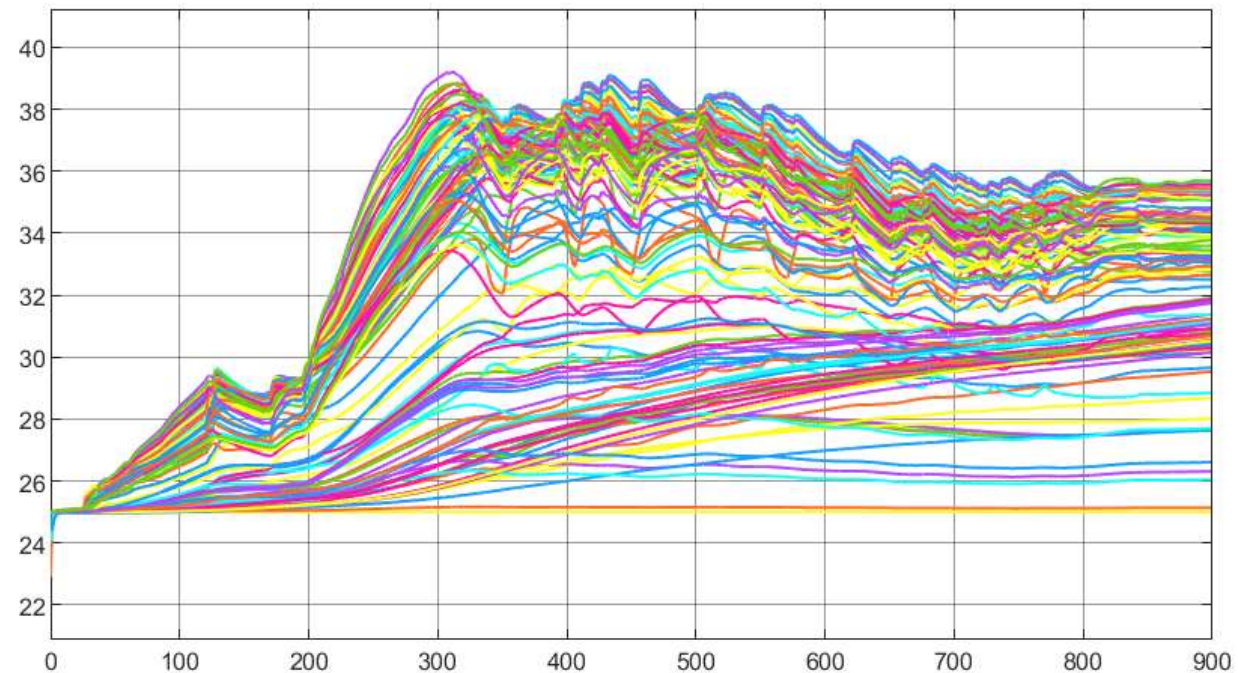


Faster System Level Simulation

Torque &
Speed



Nodes
Temperature



Different ROM Techniques Are Suitable for Different Applications

**AI-Based
Data-driven**

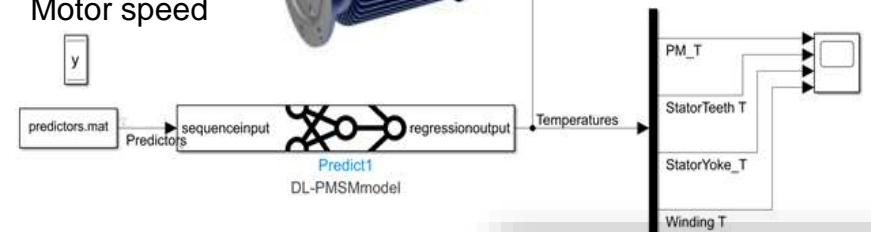
Input

Ambient T
Coolant T
Voltage
Current
Motor speed



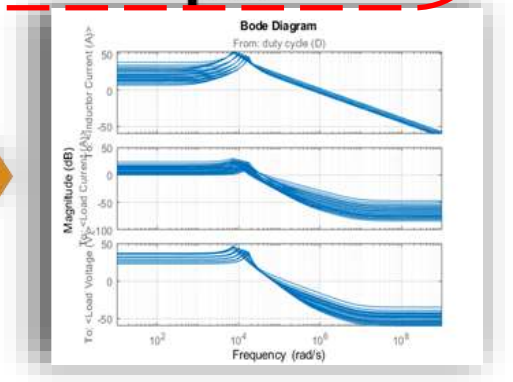
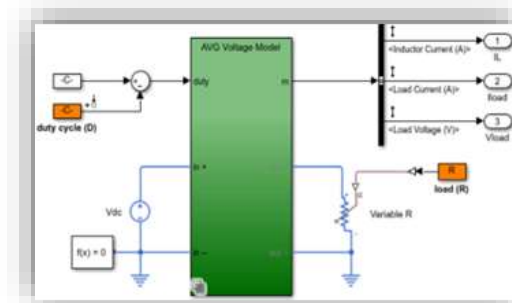
Output

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



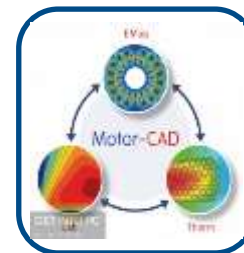
**Reduced order
model**

Linearization

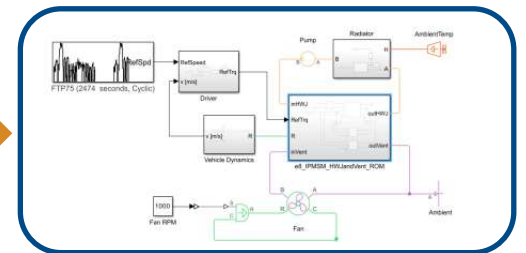


Model-based

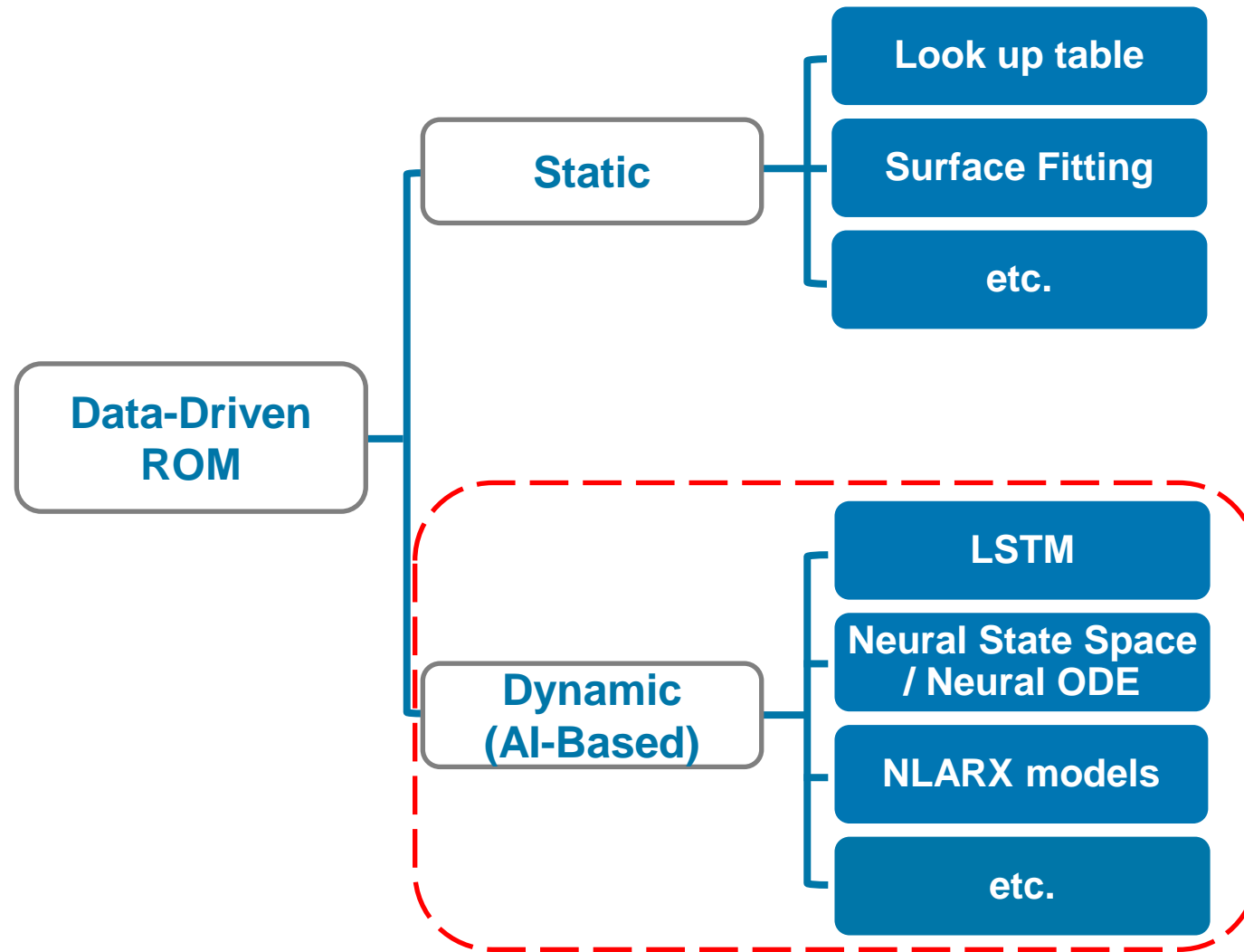
Motor-CAD



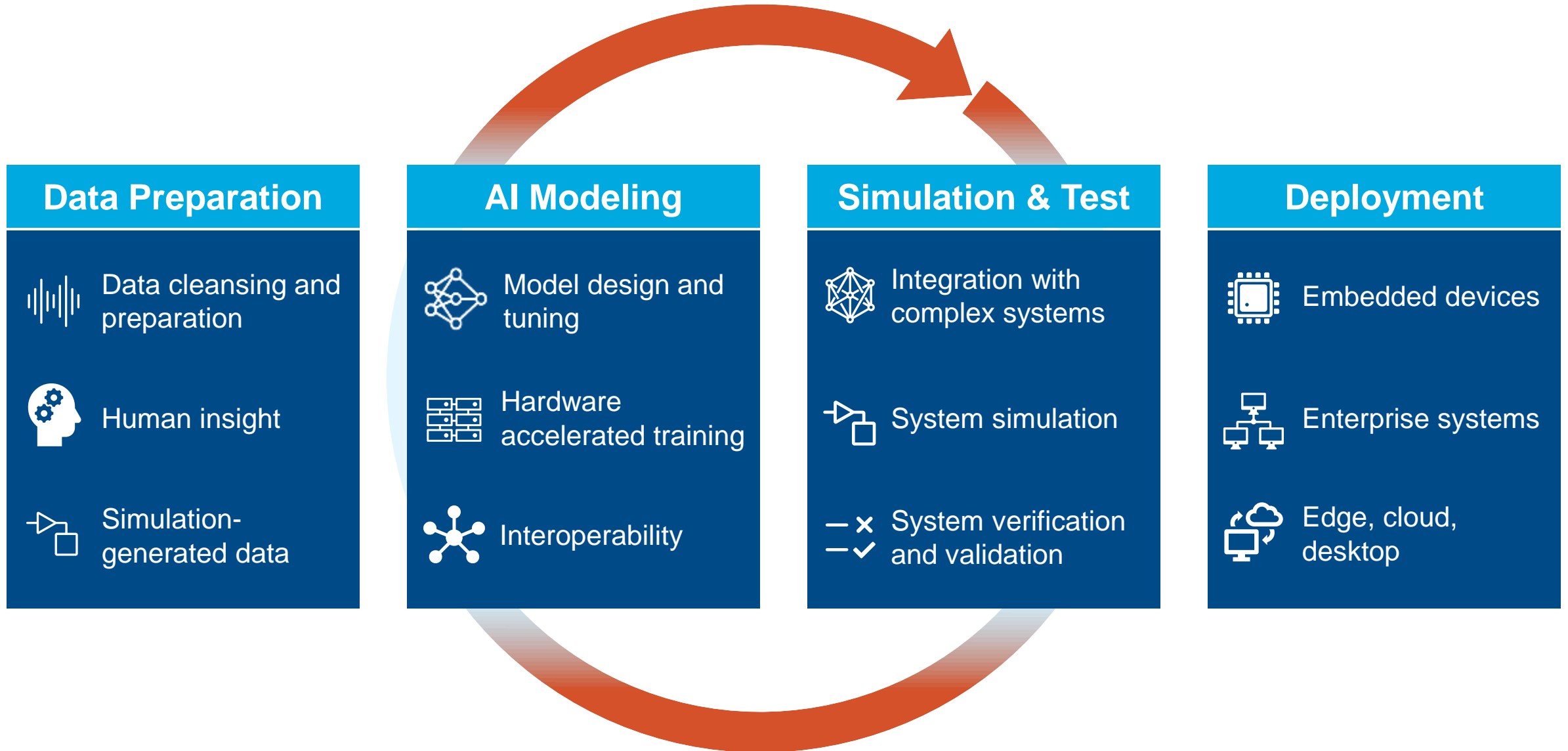
**Integrating with detailed
Simscape model**



Data-driven ROM

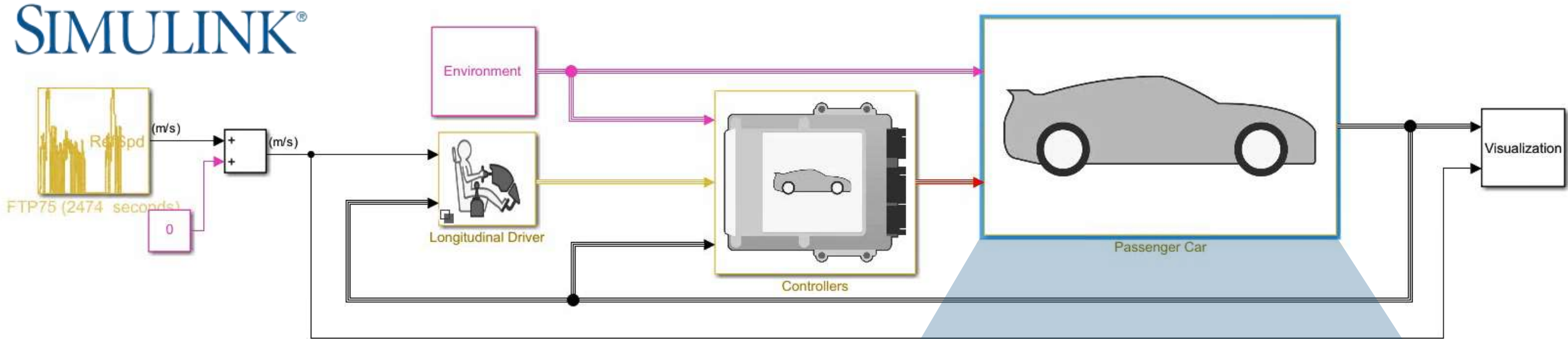


AI-driven System Design



Example Overview

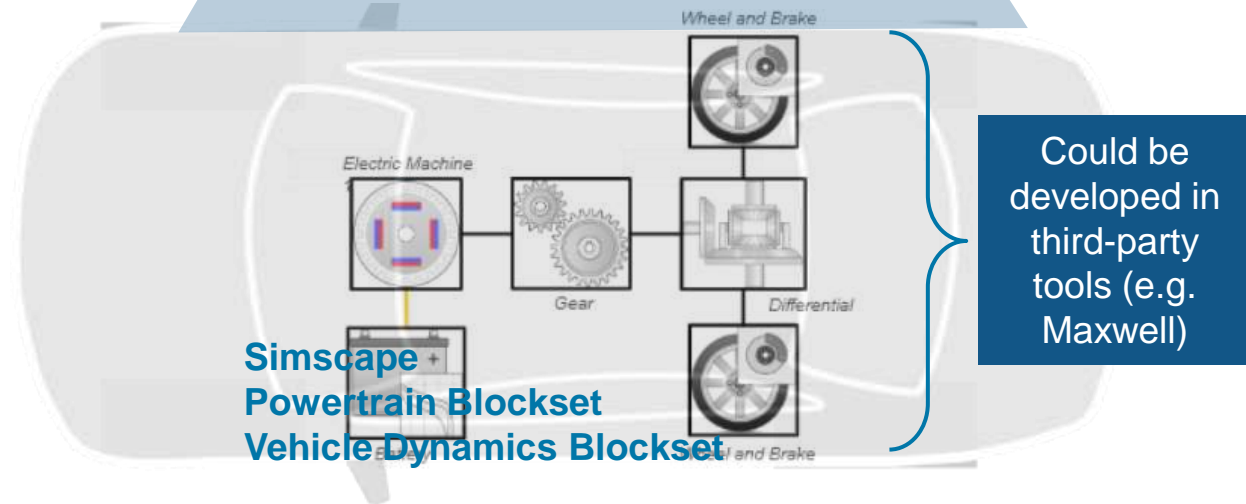
Replacing a first-principles motor model with an AI-based Reduced Order Model



High fidelity

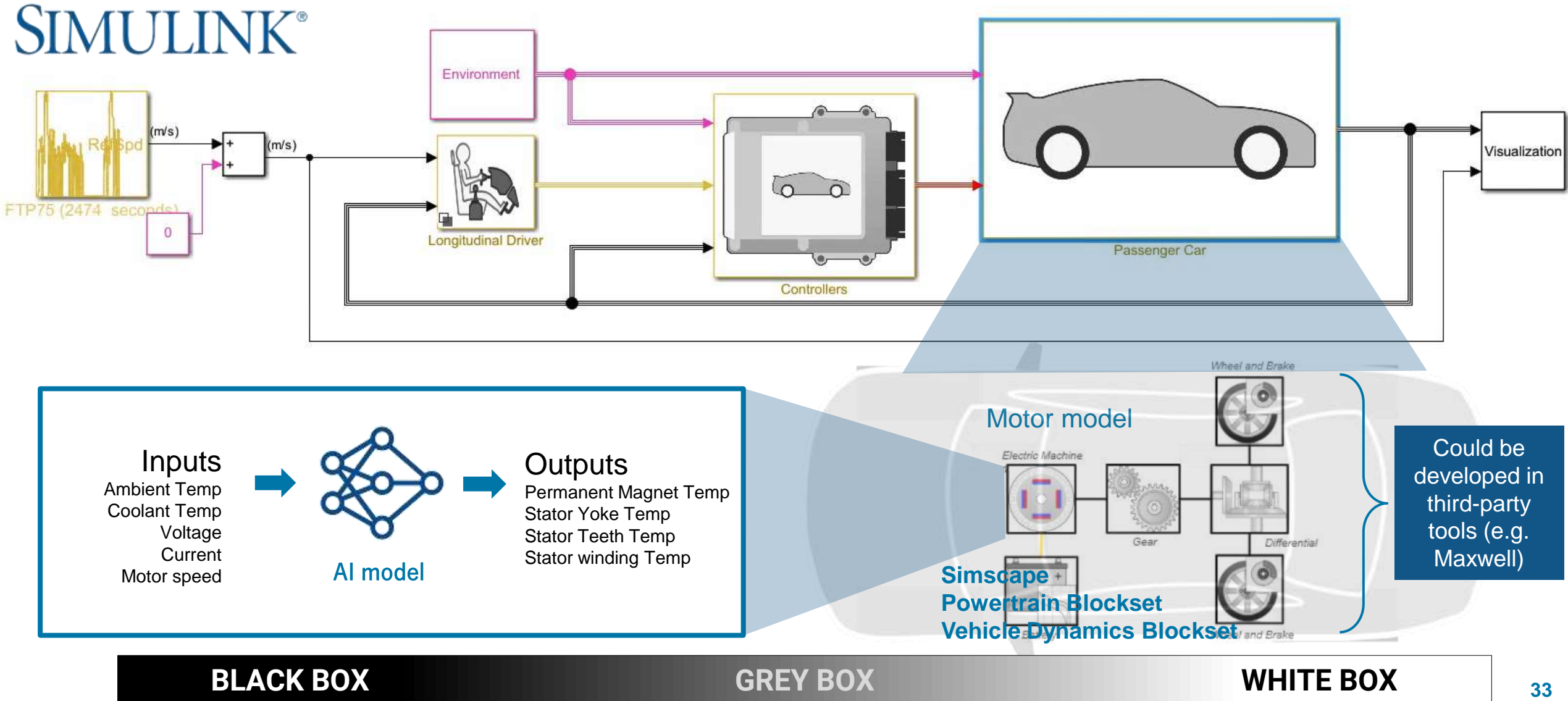
Complex model

Slow simulation



Example Overview

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

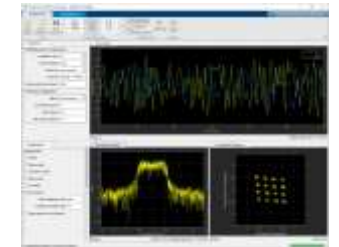


Generate Synthetic Data for Training

Other techniques:



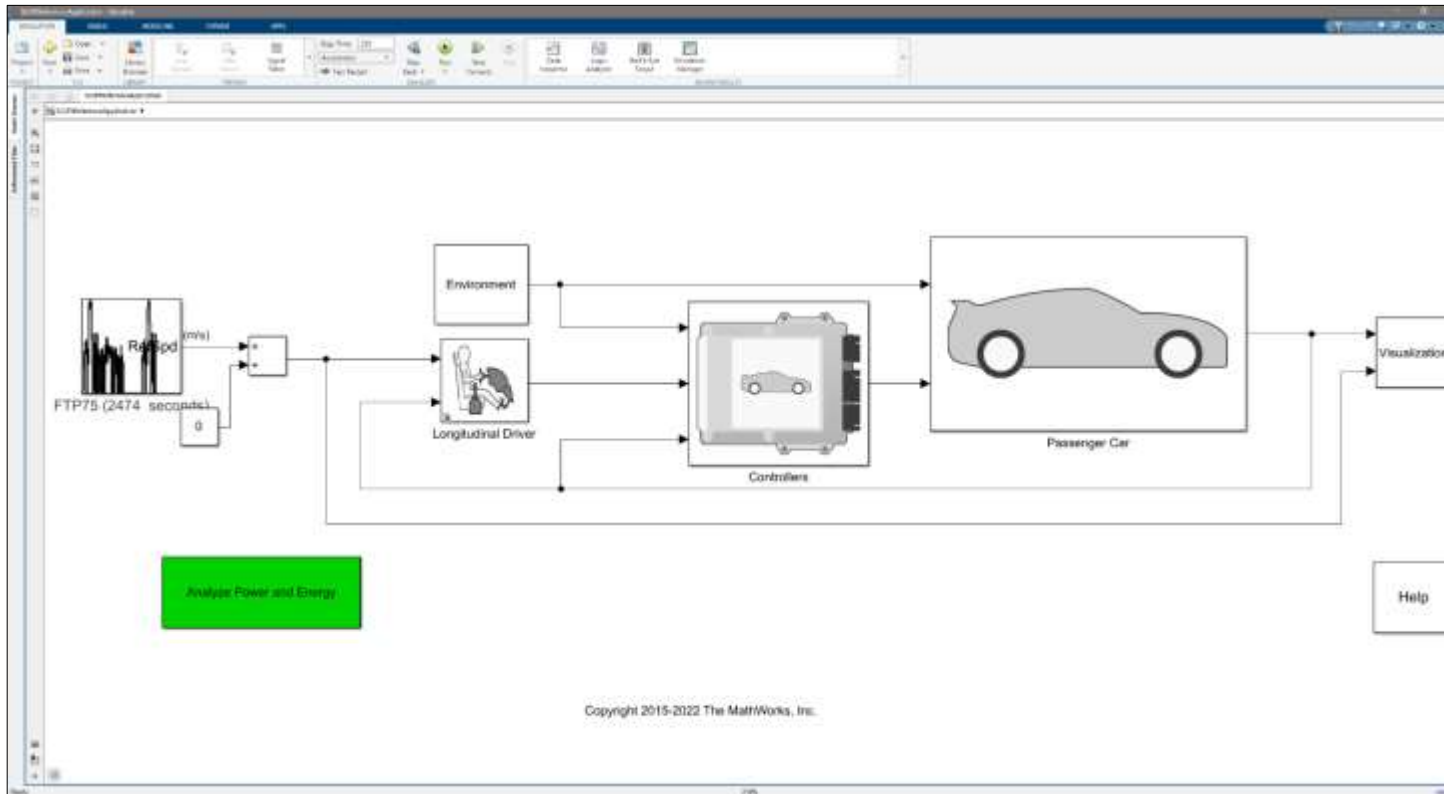
GANs



Wireless Waveform Generator



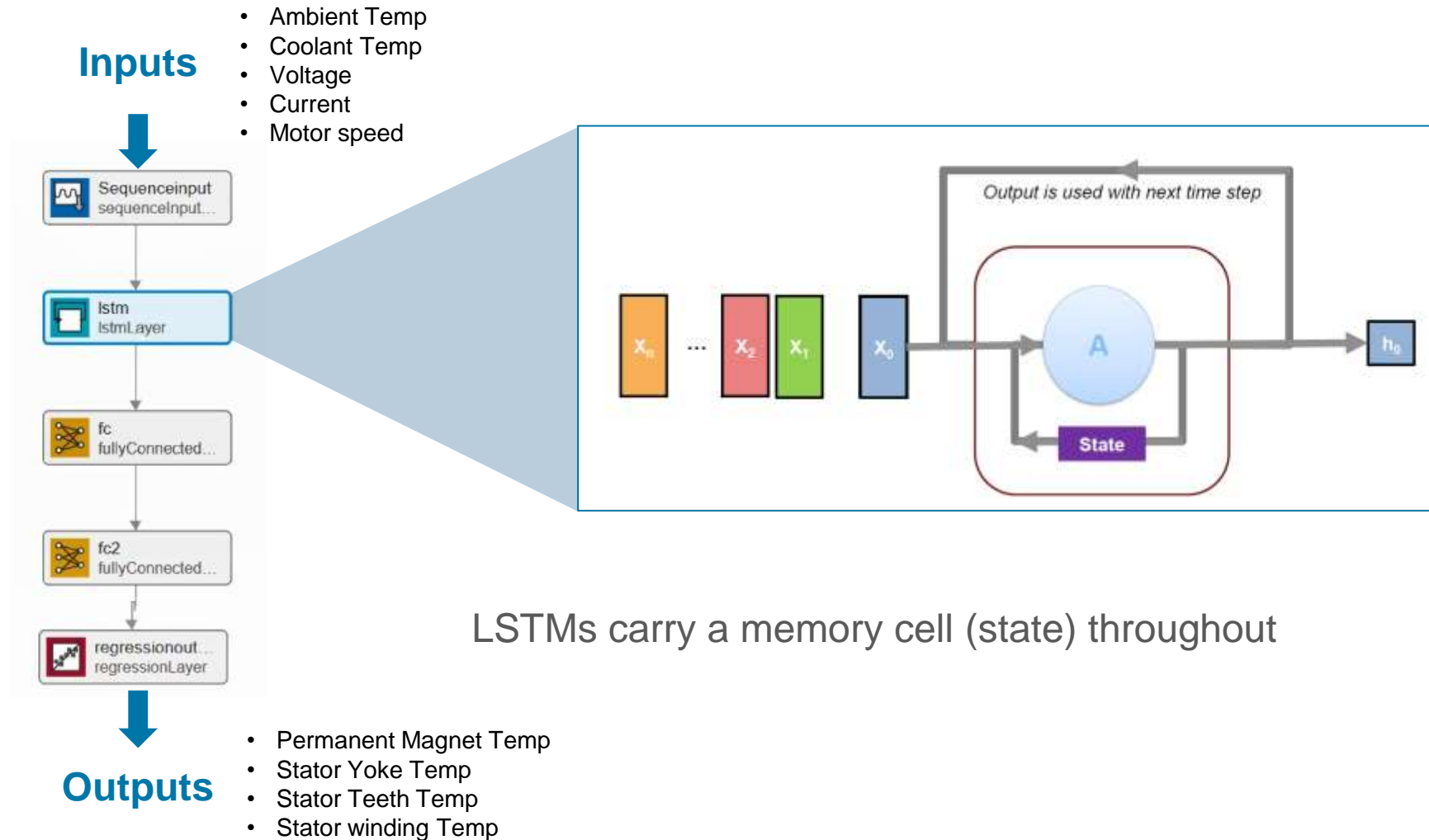
Unreal Engine®



Simulink/Simscape

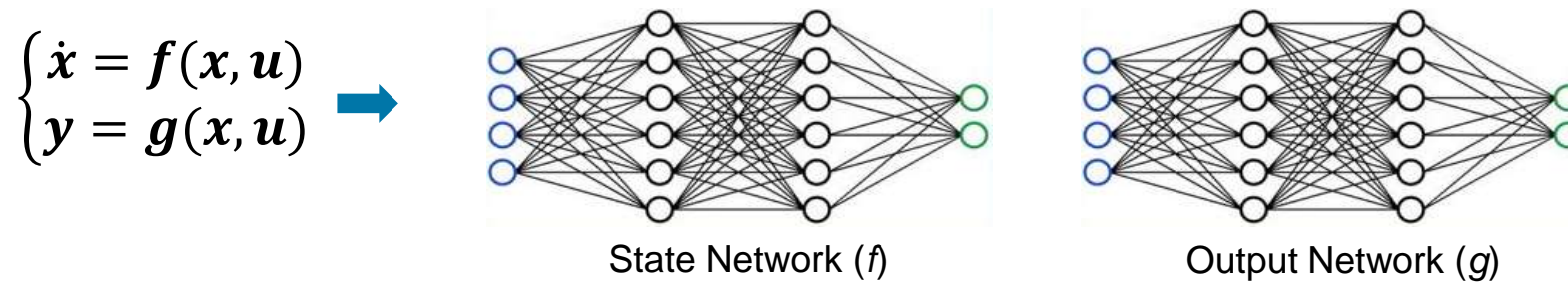
AI-based ROM using LSTMs

Capture time dependencies in time-series data



AI-based ROM using Neural State Space

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



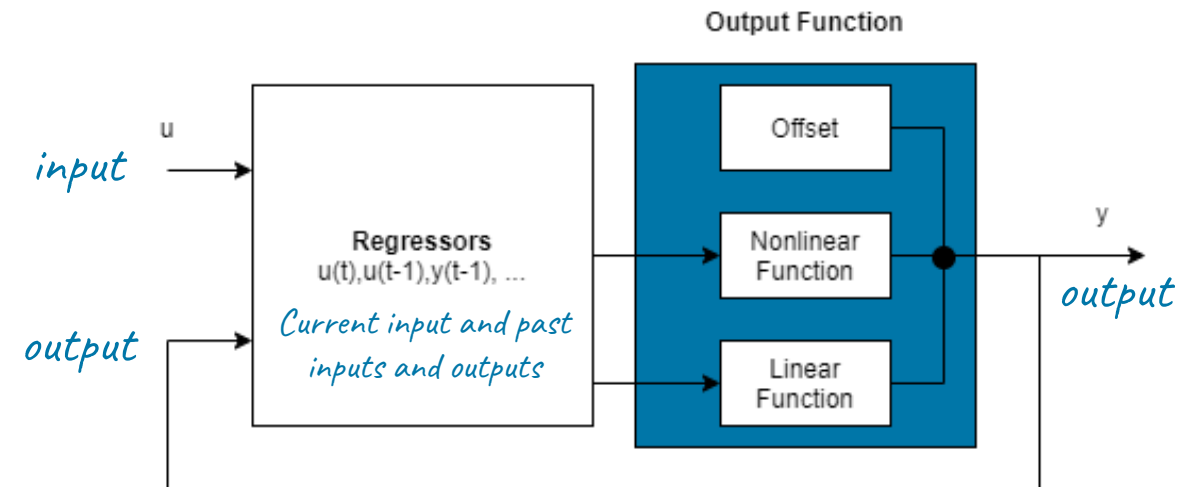
- The nonlinear state function f and nonlinear output function g are feedforward neural networks that learn from data
- Popularly known as Neural ODE in deep learning community

AI-based ROM using Nonlinear ARX

Surrogate Modeling using Sigmoid-based Nonlinear ARX Model

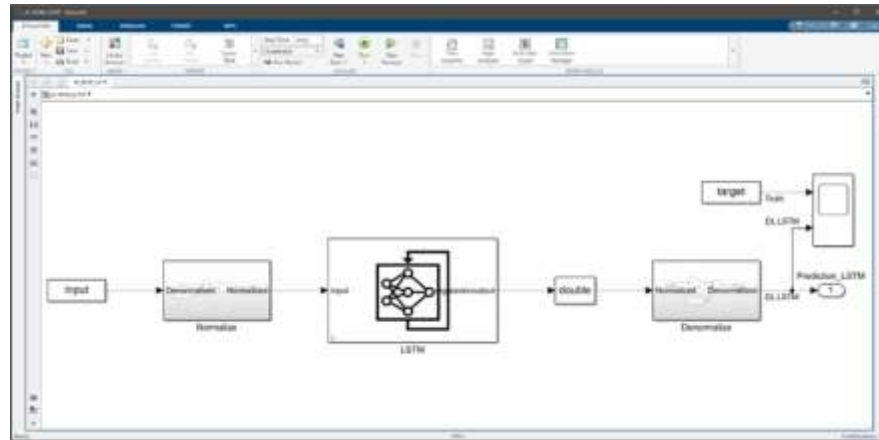
Nonlinear ARX (NLARX)

- 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

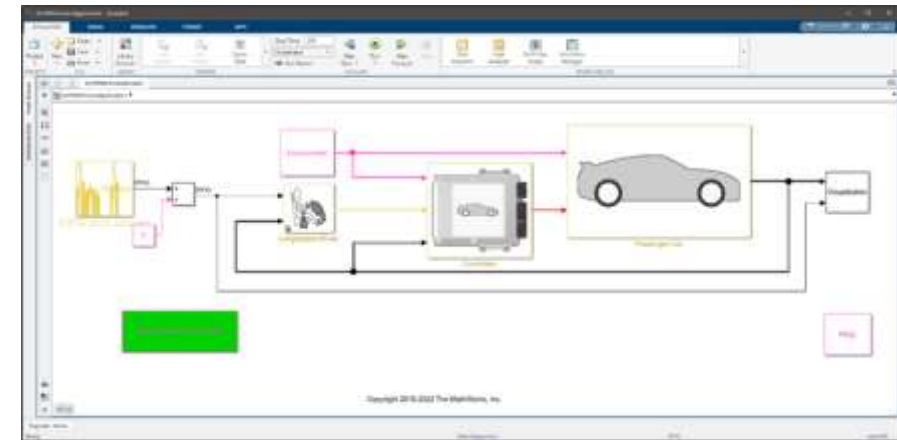


Integrate Your AI Model for System-level Simulation and Test

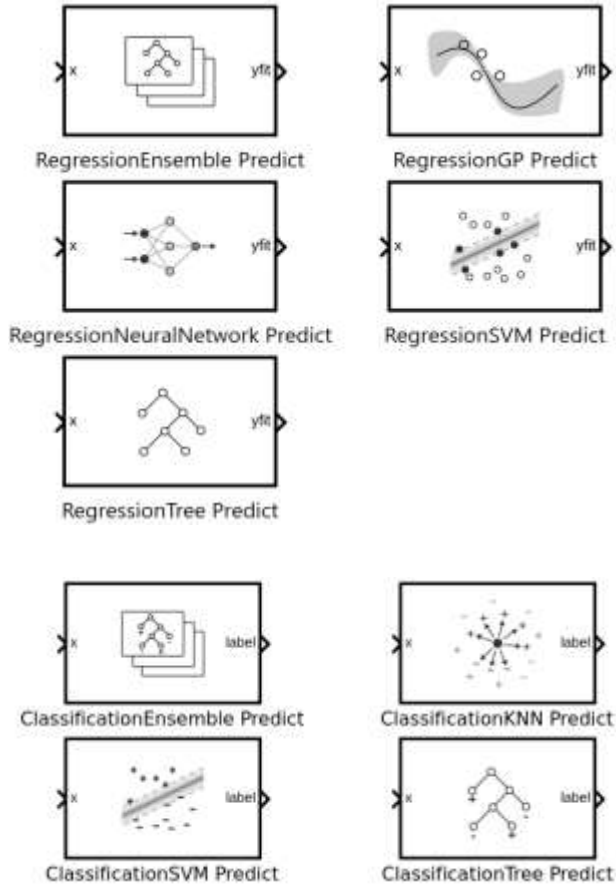
Integration of trained AI model into Simulink



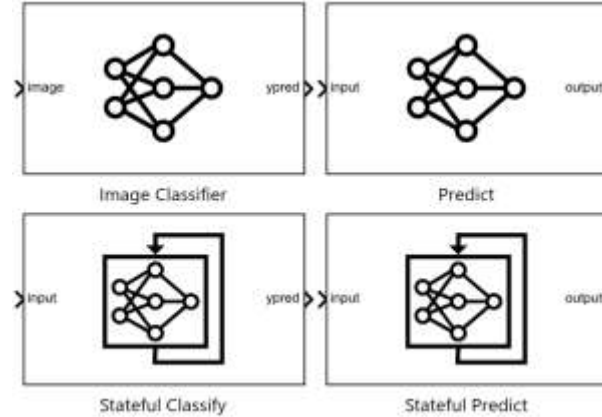
System-level simulation



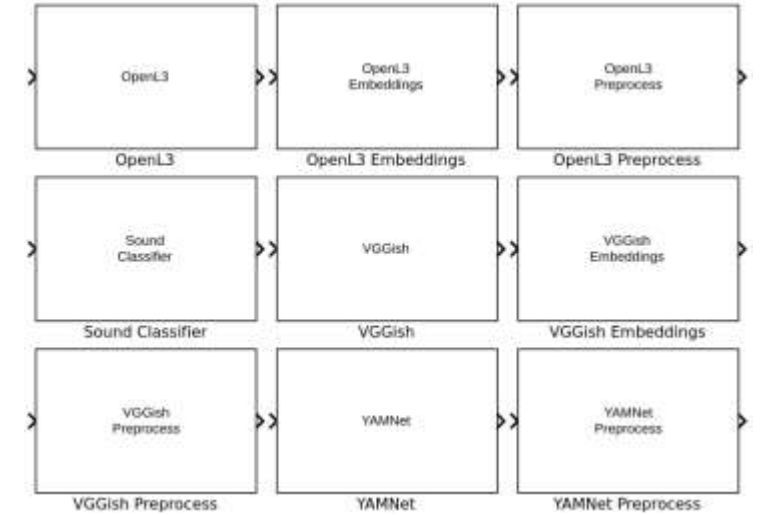
AI Libraries in Simulink Are Expanding to Include More AI Blocks for More Applications



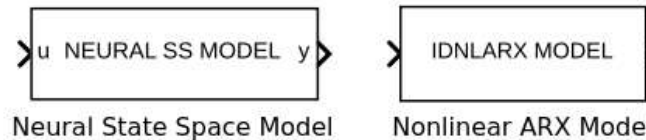
Statistics and Machine Learning Toolbox



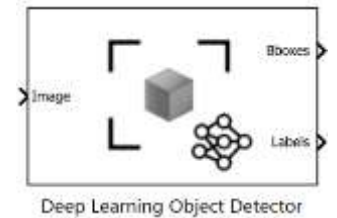
Deep Learning Toolbox



Audio Toolbox

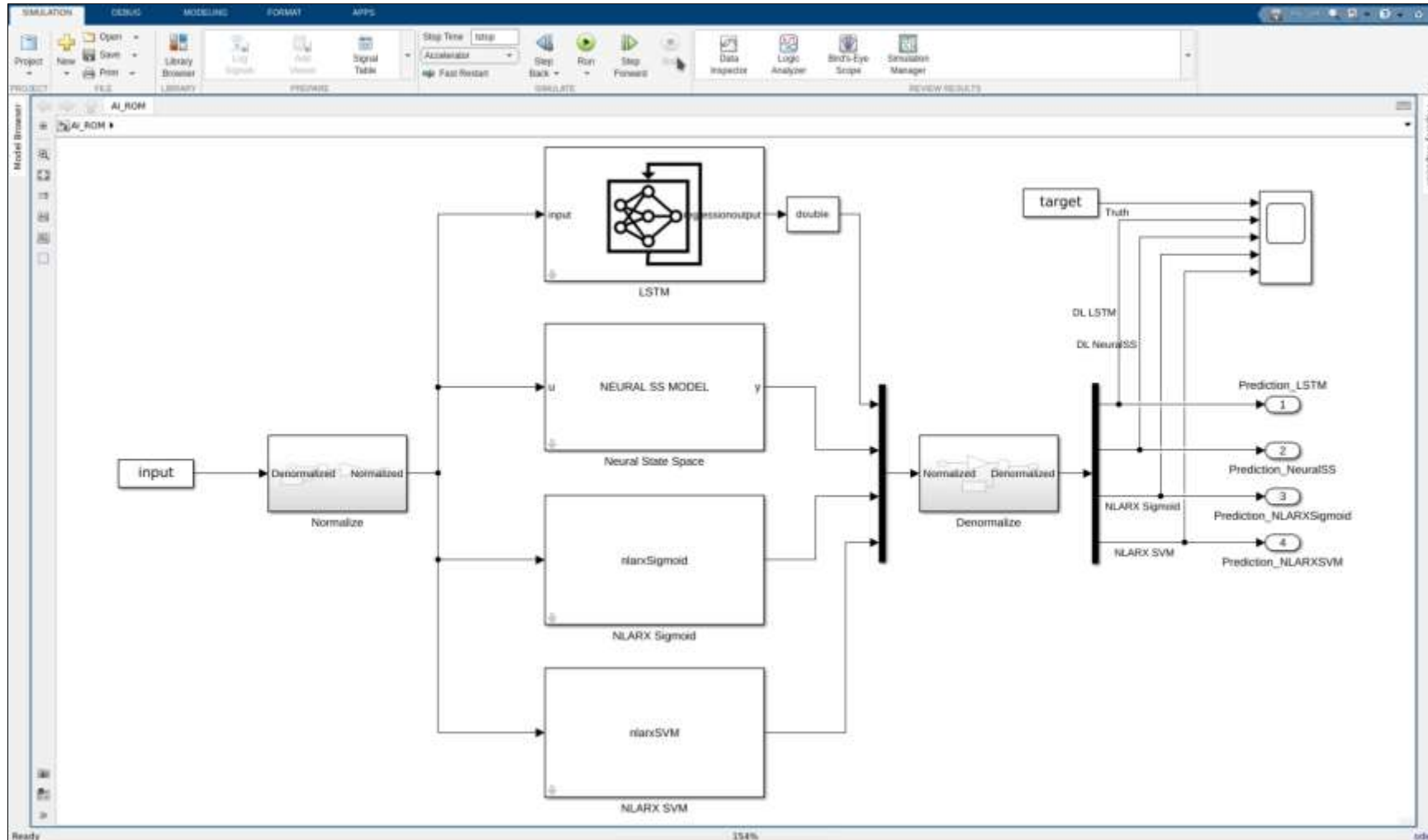


System Identification Toolbox



Computer Vision Toolbox

Integration of trained AI models into Simulink



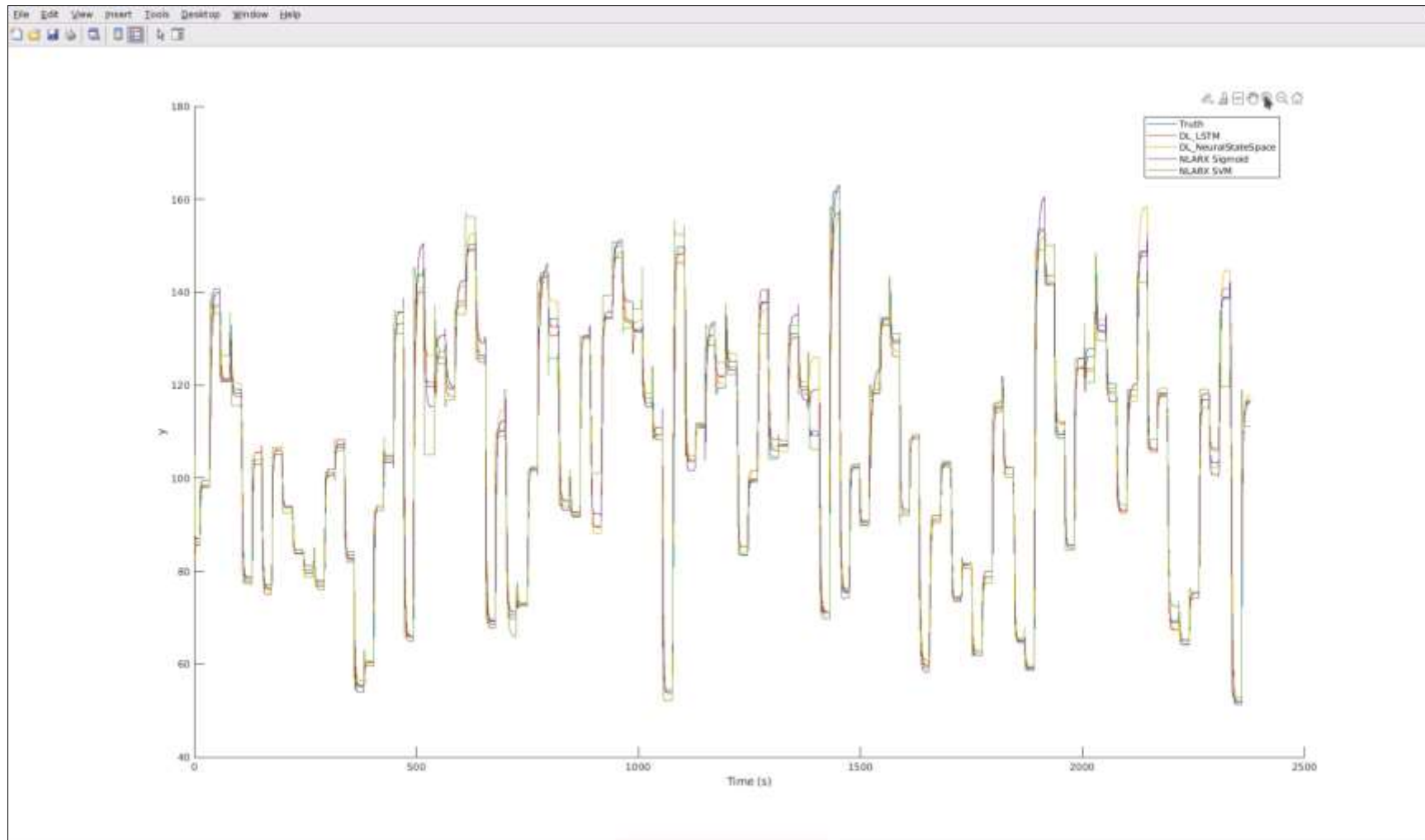
Data Preparation

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

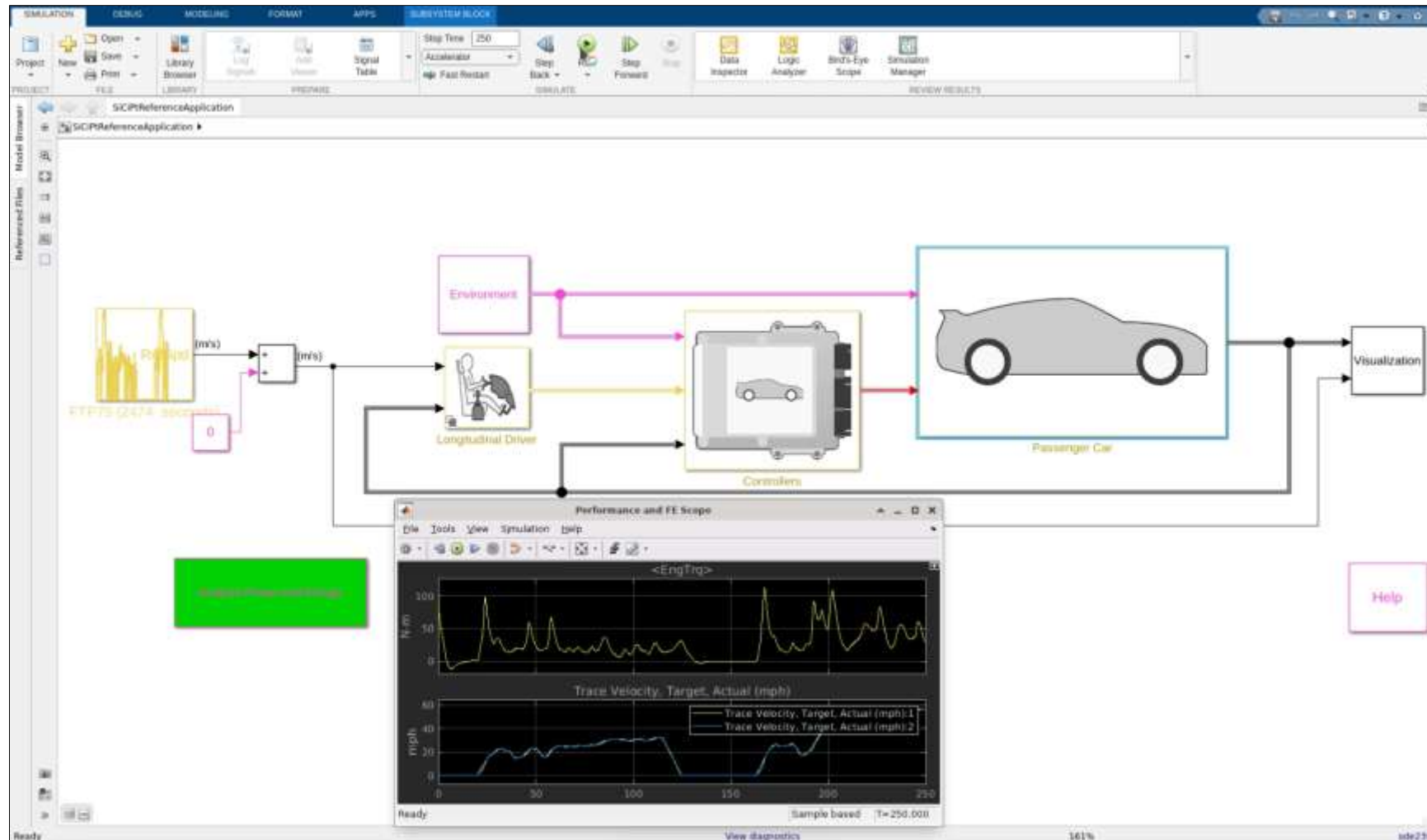
Data Preparation

AI Modeling

Simulation & Test

Deployment

System-level Simulation



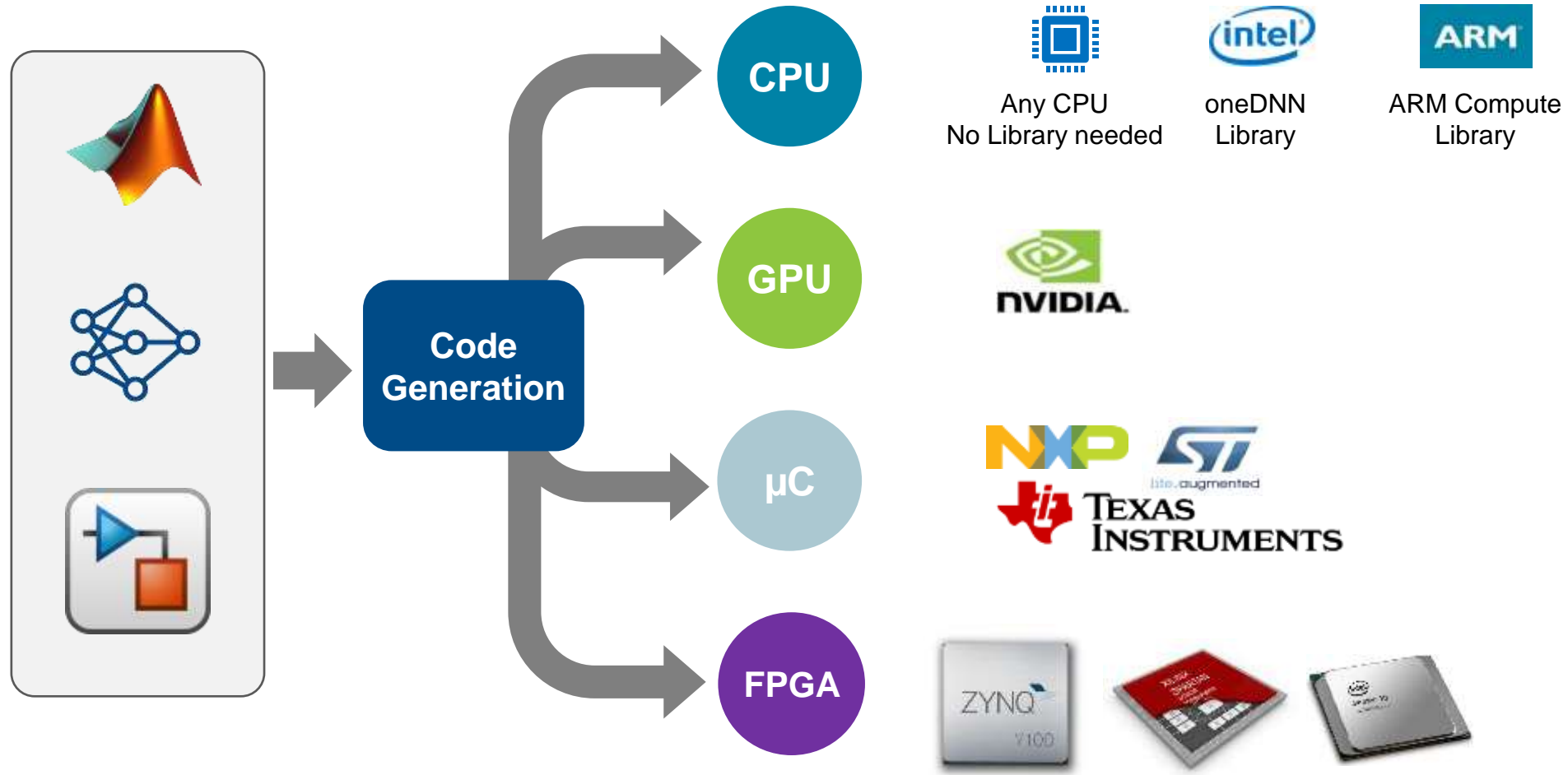
Data Preparation

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

The screenshot displays the MATLAB C Code Generation environment. The main window shows the 'Code Generation Report' for 'AI_ROM_LSTM'. The 'Code' section is expanded, showing the source file 'AI_ROM_LSTM.c'. The code includes function definitions for 'AI_ROM_LSTM_macroKernel' and 'AI_ROM_LSTM_matrixMultiply', and a main function 'DeepLearningNetwork_predictAndJ' that calls 'cell_wrap_3_AI_ROM_LSTM_T_outT_f2(3)'. The code is annotated with line numbers and comments. Below the code editor, a block diagram shows a 'double' block connected to a 'No' block, indicating the data type and deployment target.

Data Preparation

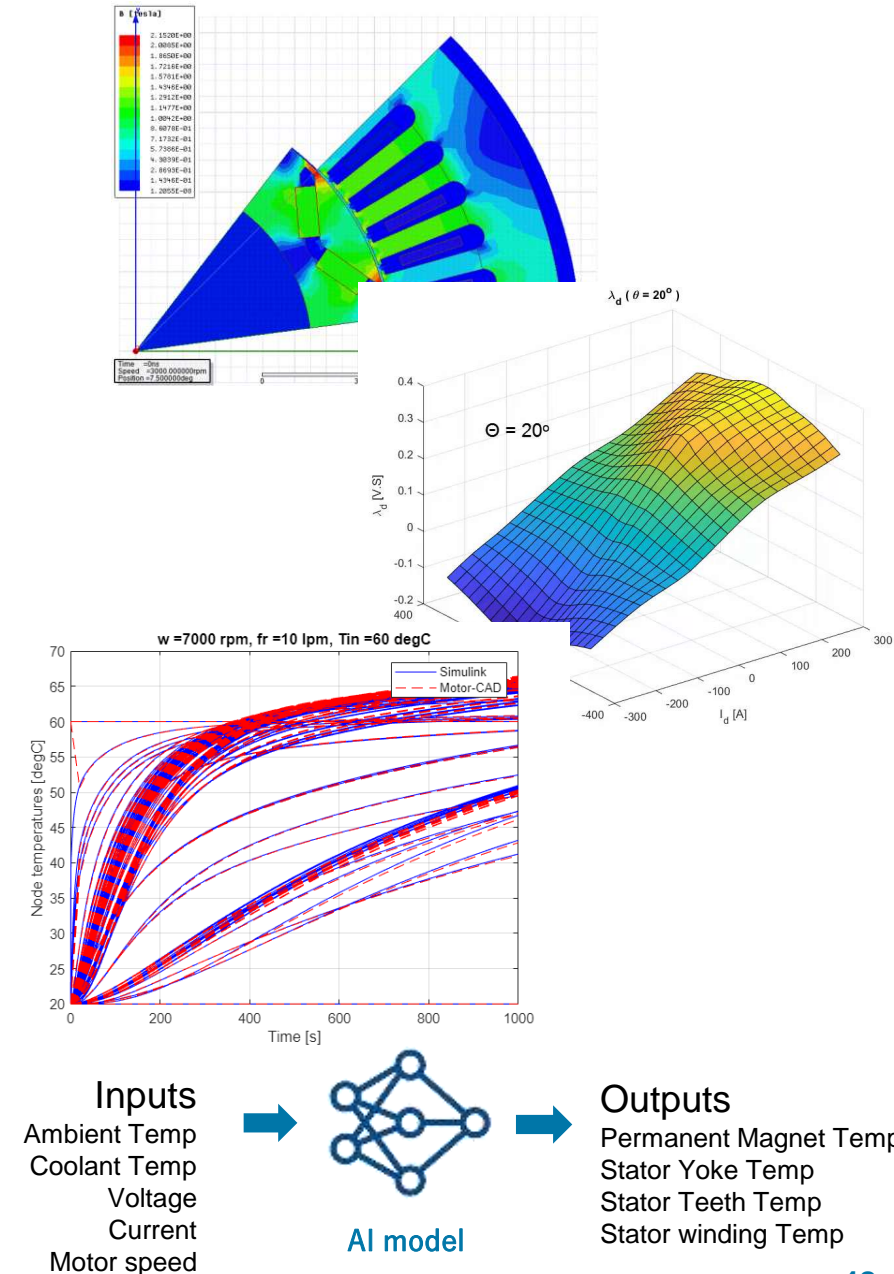
AI Modeling

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>



MathWorks Videos

[Suggest a video](#)



Understanding Field-Oriented Control | Motor Control, Part 4



Reinforcement Learning for Developing Field-Oriented Control



Motor Control, Part 3: BLDC Speed Control Using PWM

[View all MathWorks videos](#)



Welcome to the Power Electronics Control Community
Moderator: Tony Lannon

The MathWorks community for engineers using Simulink to apply power electronics control to Electric Vehicles, Renewable Energy, Battery Systems, Power Conversion, and Motor Control. Learn from examples and videos submitted by your peers and MathWorks engineers.

Answers

[Ask a question](#)

AD How to generate first value for Kp, Ki of PID in current controller

Latest activity by adhavan d on 7 Jul 2023 at 5:09

Tags: power_electronics_control, electric_motor_control, control system

25 views 0 votes 1 comment

1
answer

Files

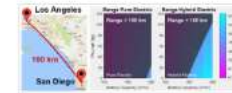
Electric Aircraft Model in Simscape

by Steve Miller on 5 Jul 2023 at 17:40

Tags: emissions, more electric aircraft, physical modeling, battery_system_management, power_electronics_control

21 Downloads (30 days)

★★★★★



[Submit a file](#)

Simulink

Discussions

[Start a discussion](#)

PR Electric vehicle thermal management

Latest Activity by Prakhar Rathore on 29 May 2023

Tags: electric vehicles, power demand, powertrain, thermal management, battery

0
replies

Top Community Contributors



Mohsen Aleenejad



Dakai Hu



Jonathan LeSage



Graham Dudgeon



Gernot Schrabberger

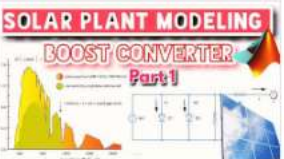


Joel Van Sickle

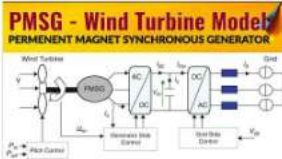
Additional Resources

- [Hardware Support](#)
- [Hardware Support for Texas Instruments Microcontrollers](#)
- [Hardware Support for NXP](#)
- [Hardware Support for ARM](#)
- [Topics in Power Electronics](#)
- [Power Electronics Control Design](#)


Community Videos



Modeling a PV Solar Power



Permanent Magnet



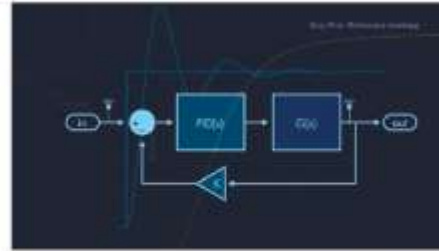
Selective Harmonic Elimination

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.



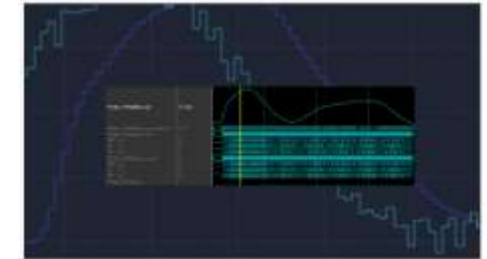
Control System Design with MATLAB and Simulink

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](#)