

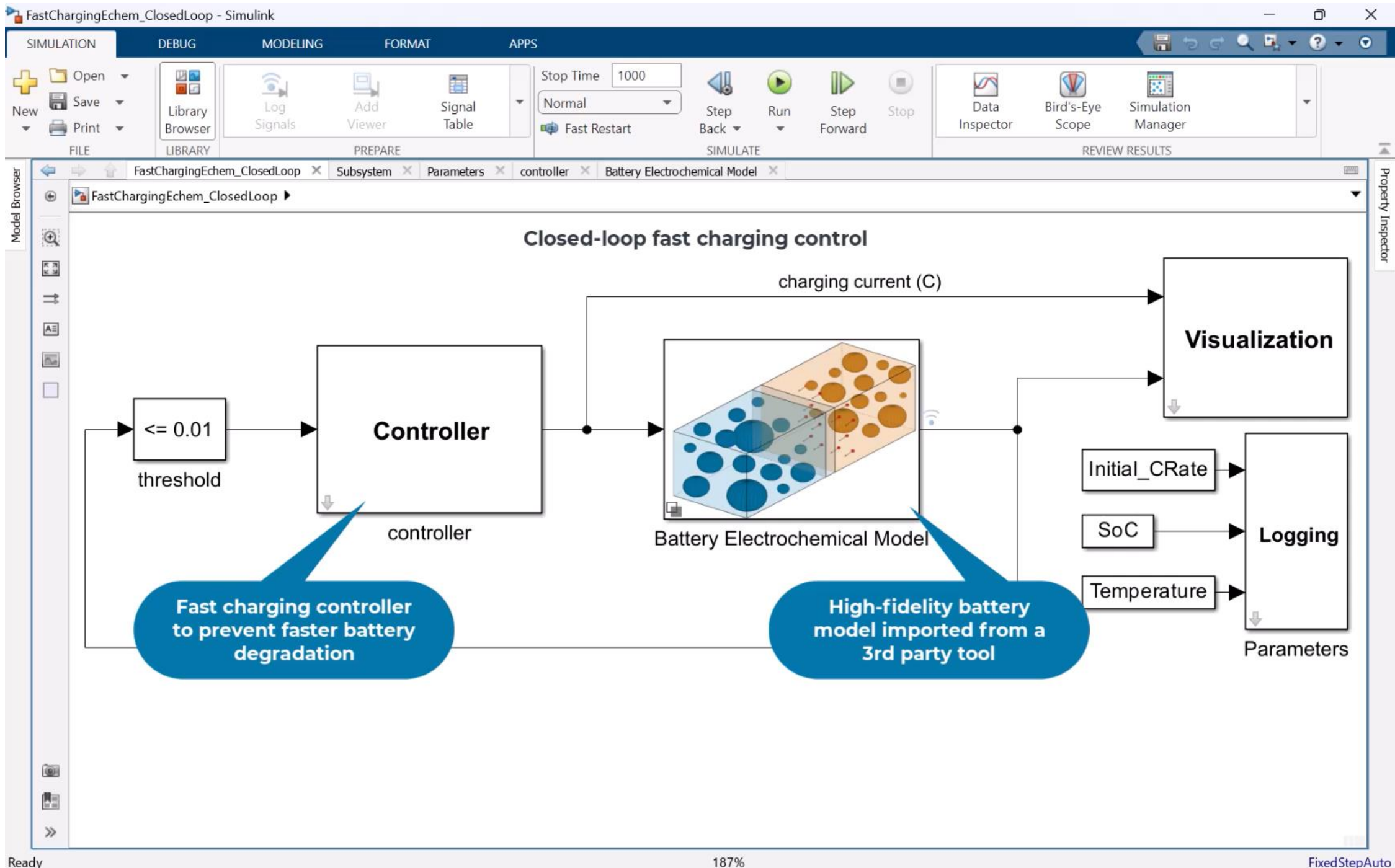
Reduced Order Models for Speeding up Simulation Using AI



Mahaveer Satra

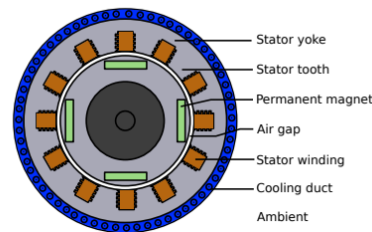
*Industry Application Engineer,
MathWorks
msatra@mathworks.com*





How about Motors?

- Permanent Magnet Synchronous motors are backbone for electrification
- Temperature Excursions in these Motors leads to loss of Torque efficiency and eventual failures
- Need test these devices over possible Thermal Regimes
- Dyno testing is costly and can lead to degraded devices
- Simulation is a must, but faster simulations are essential and Virtual Sensors are bonus



Key takeaways

Enable

Reuse of full-order high-fidelity models for system-level simulations, Hardware-in-the-Loop (HIL) testing, nonlinear control design, and virtual sensor modeling.

Explore

Various ROM techniques in MATLAB to find the best method.

Common challenges

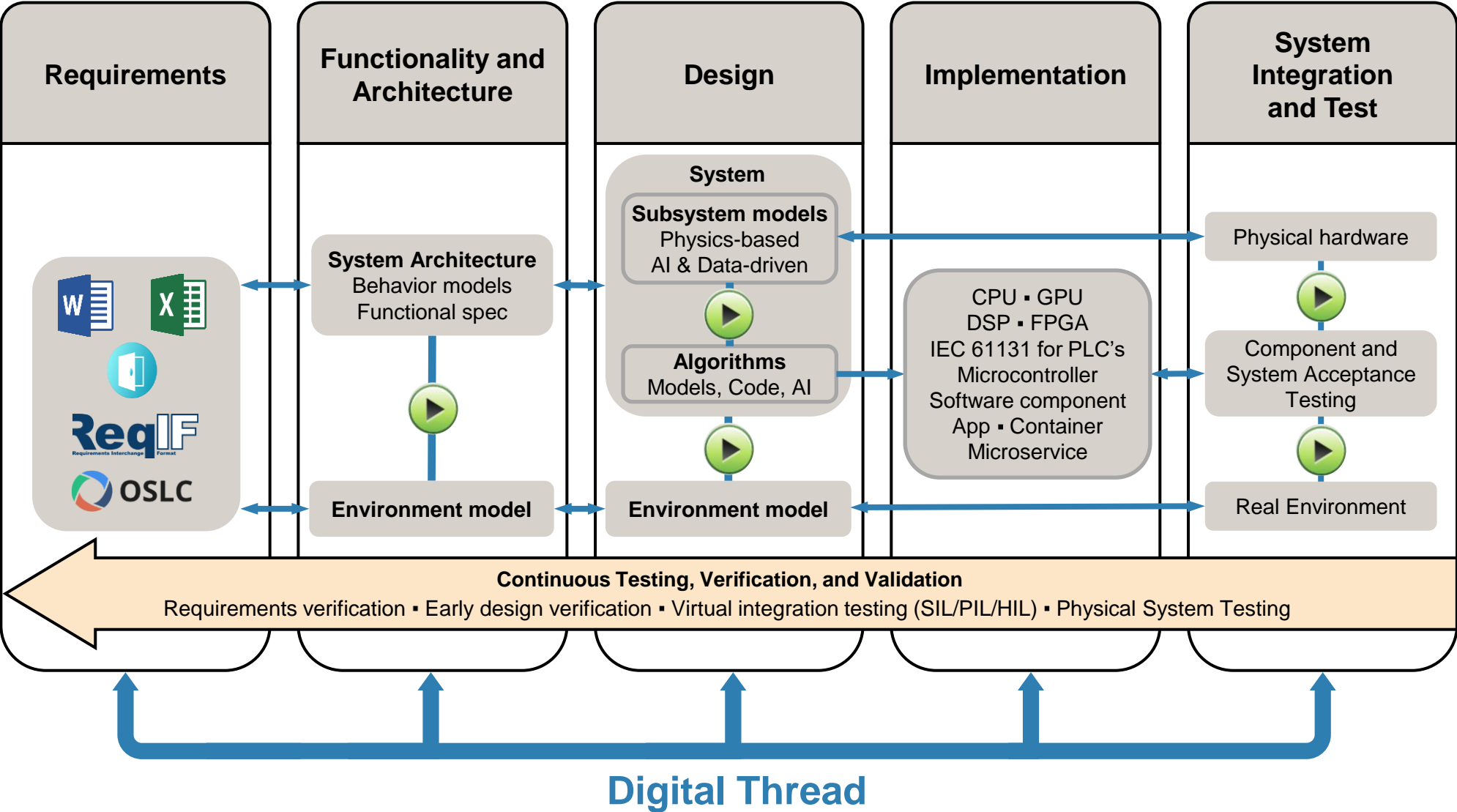


High fidelity models, such as ones from 3rd party FEA/CFD tools, are too slow for system level simulation, control design, and HIL testing.

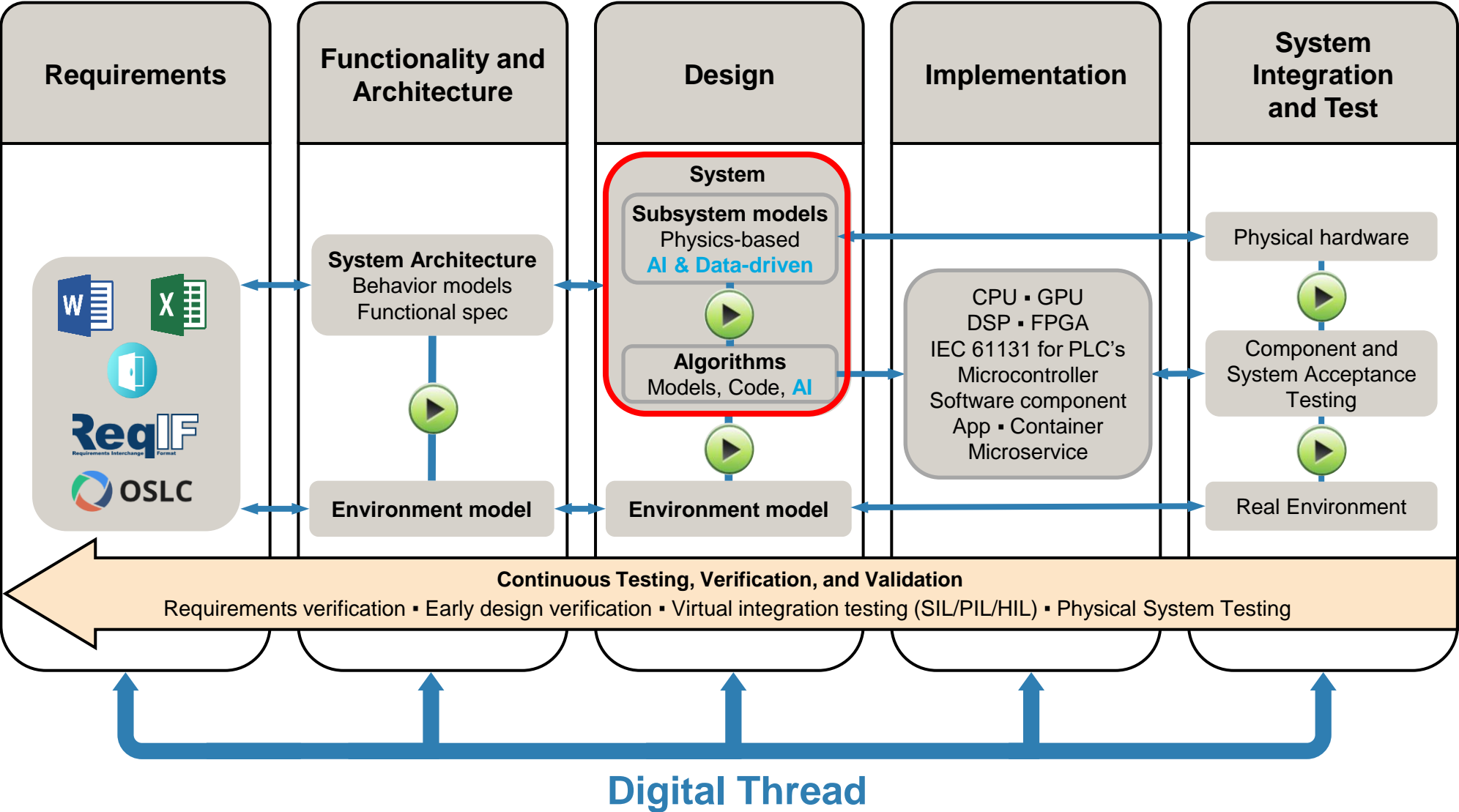


Creating a ROM that produces desired results in terms of speed, accuracy, interpretability, etc.

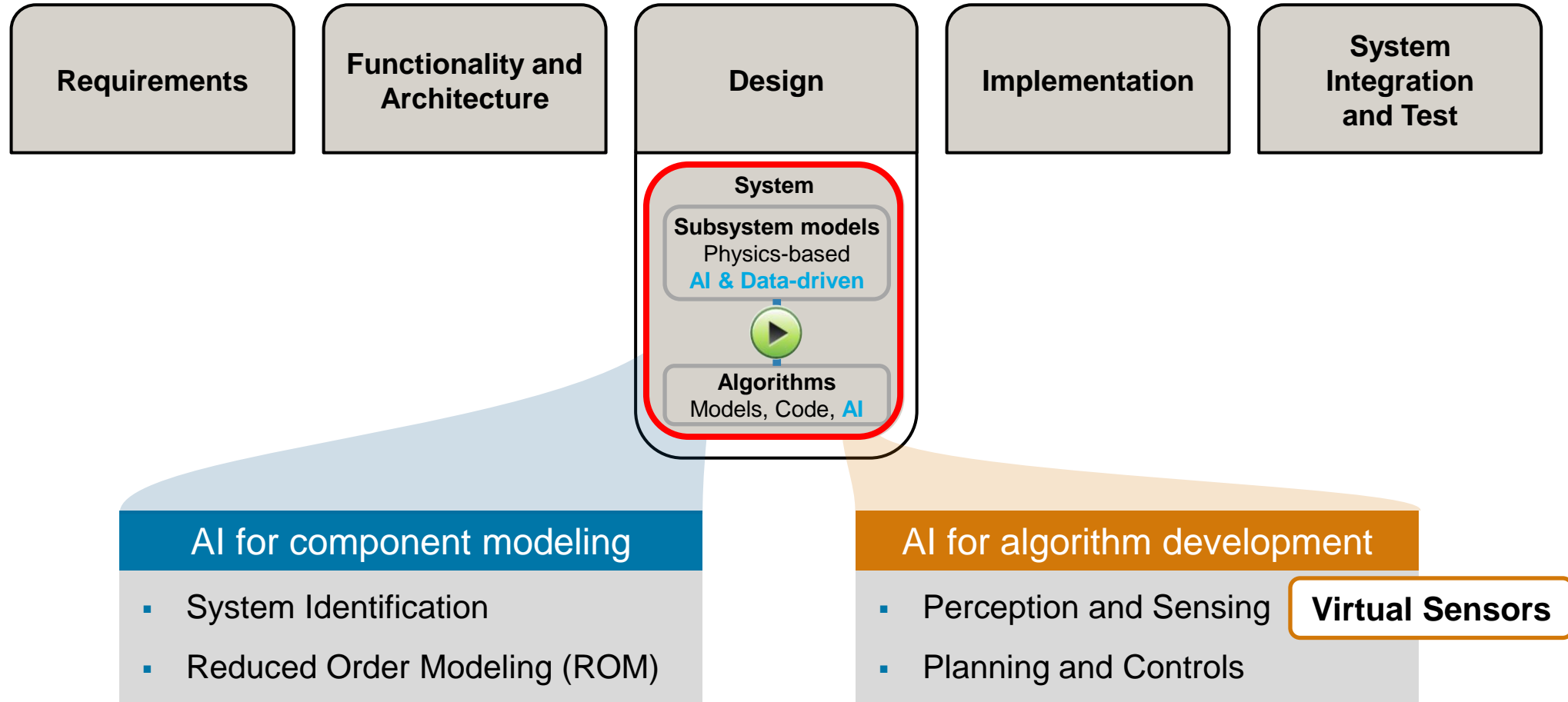
Model-Based Design



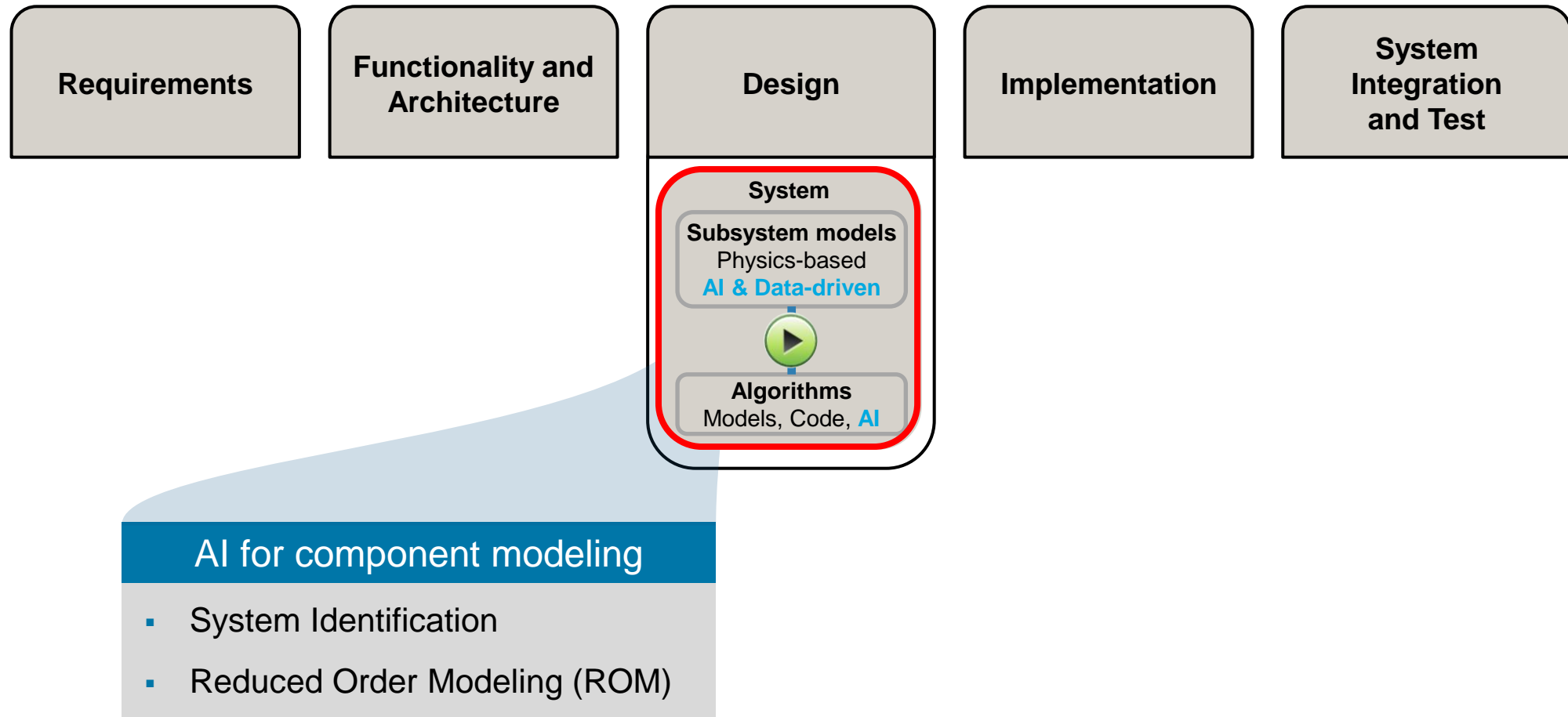
Integrating AI into Model-Based Design



AI for Component Modeling and Algorithm Development



Focus today



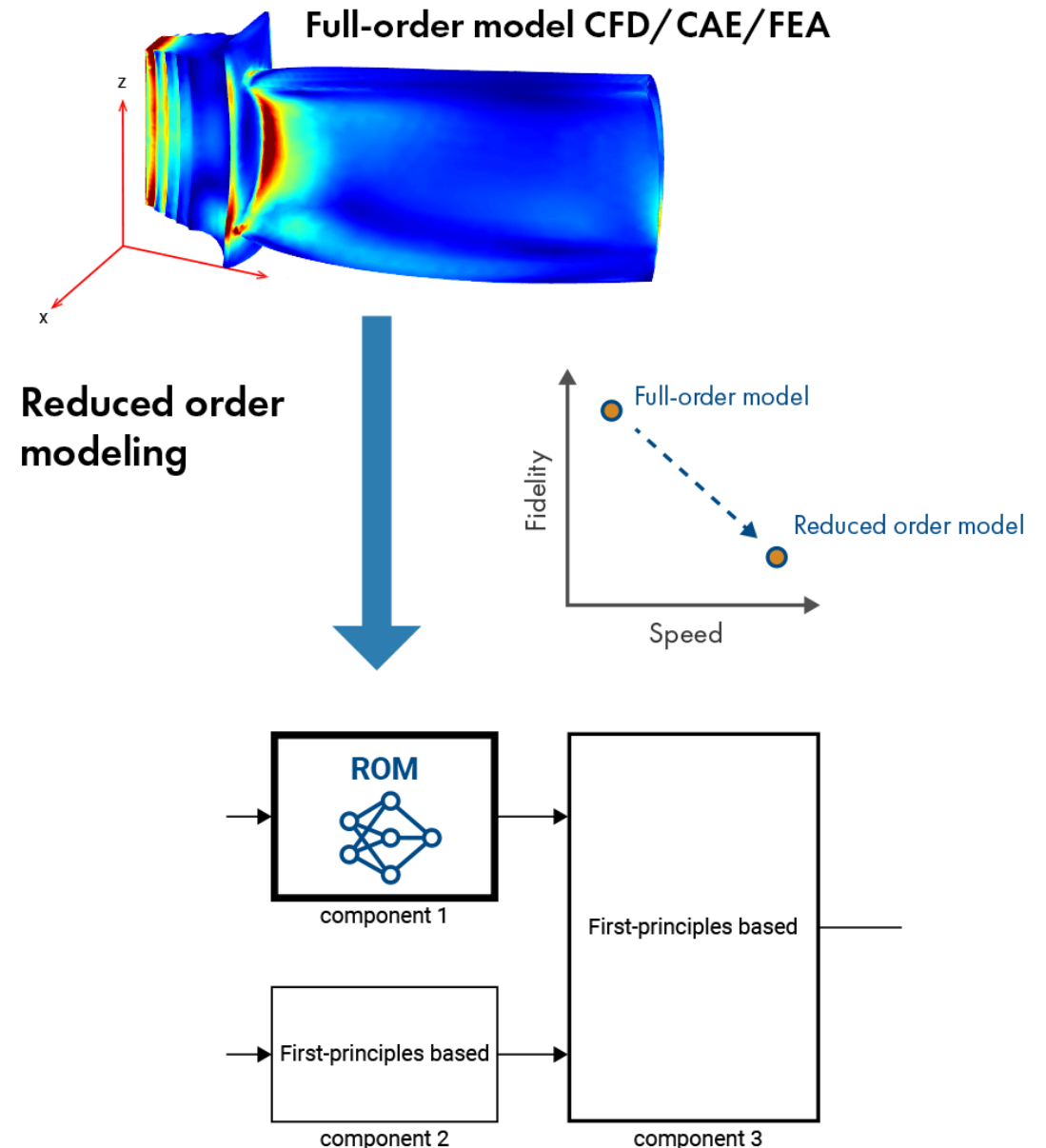
Reduced Order Modeling

What

- Techniques to **reduce the computational complexity** of a computer model
- **Provide reduced, but acceptable fidelity**

Why

- Enable simulation of FEA models in Simulink
- Perform hardware-in-the-loop testing
- Perform control design
- Develop virtual sensors, Digital twins
- Perform design exploration

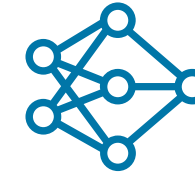


Reduced Order Modeling

How

AI-Based
Data-driven

Inputs
Charging Rate
SoC



AI model

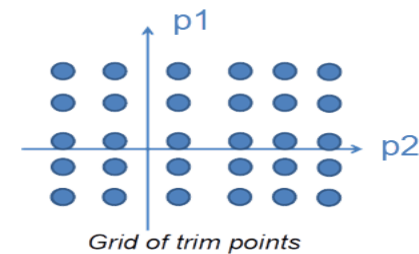
Outputs
Potential
Difference ($V_s - V_e(V)$)

focus today

Reduced order
model

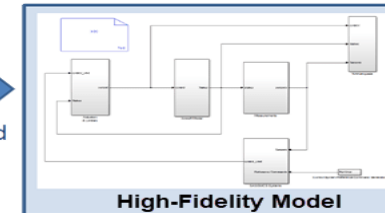
Linearization

Model-based

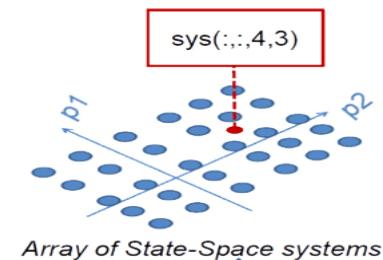


Grid of trim points

Loop through the grid
of trim points



High-Fidelity Model



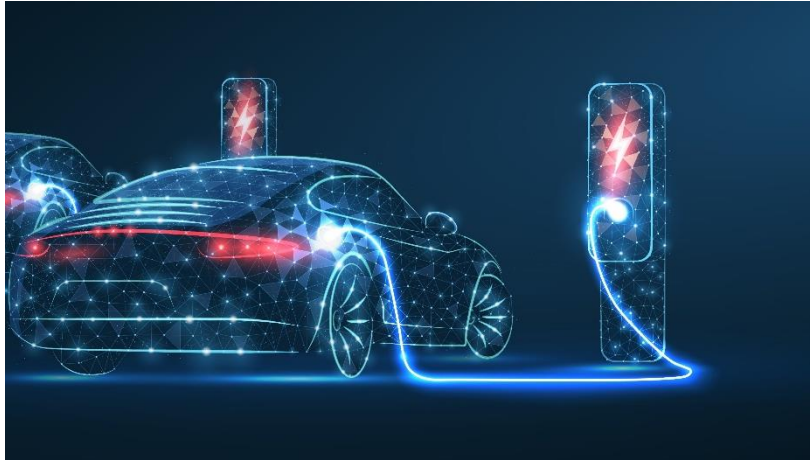
Array of State-Space systems

Identify local model at
each trim point

FEA
Software

Simulink
Simscape Multibody
Control System Toolbox

Case Study: Fast Charging Control of Electric Vehicle Battery System



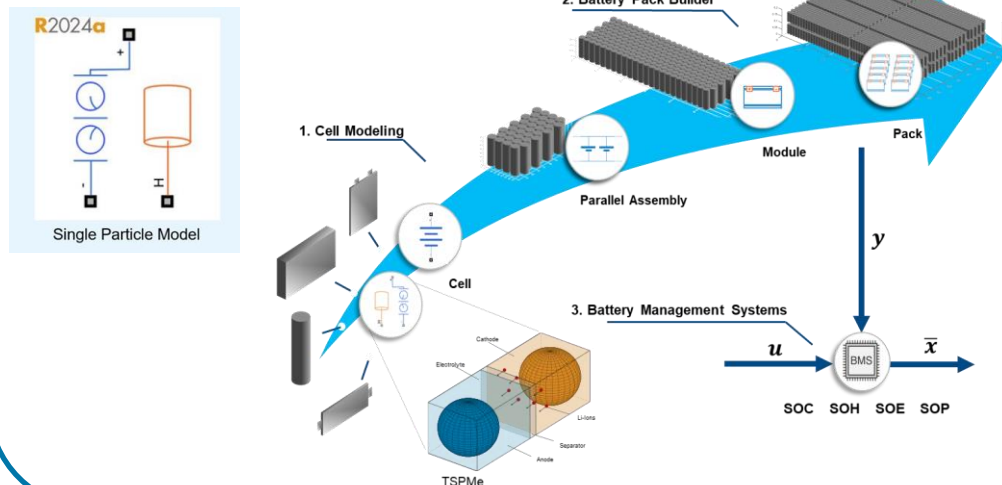
How can I optimize fast charging speed while preventing the battery from faster degradation?

Battery Electrochemical Models

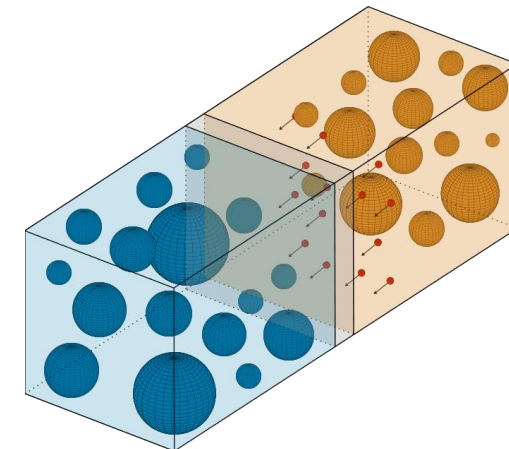
$$J = \frac{i_a}{F} = \frac{i_0}{F} \left\{ \exp \left[\frac{(1-\beta)F}{RT} \eta \right] - \exp \left[-\frac{\beta F}{RT} \eta \right] \right\}$$

$$\nabla \cdot \left[\kappa^{eff} \nabla \bar{V}_2 + g_2 \kappa_D^{eff} \nabla (\ln \bar{c}_2) \right] + J_{V_2} = 0$$

Simscape Battery



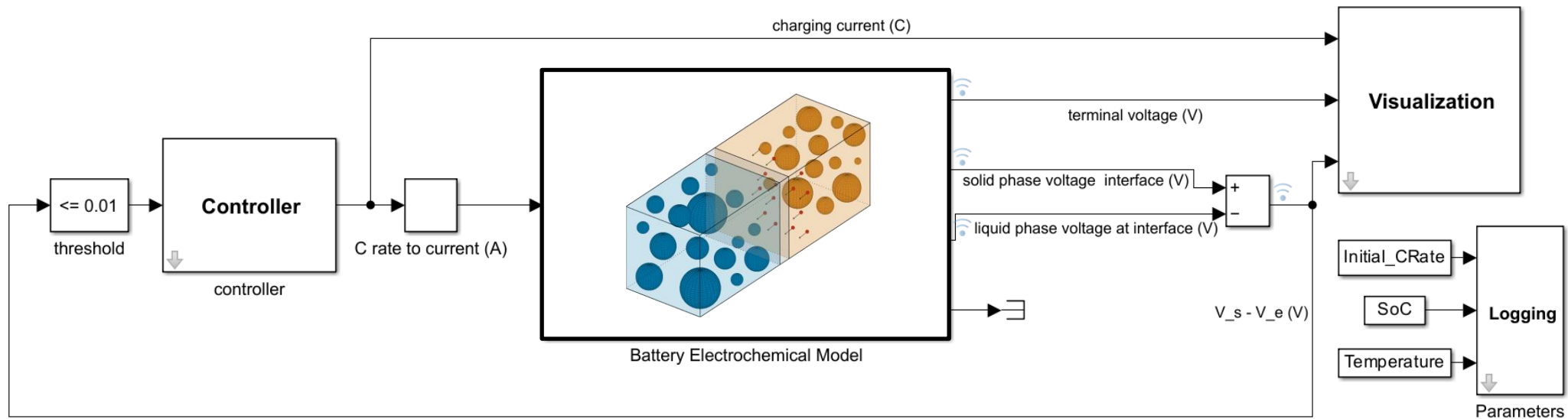
Third-Party Battery Modeling Tools



Example overview

Replacing a high-fidelity electrochemical battery model with an AI-based reduced order model

SIMULINK®

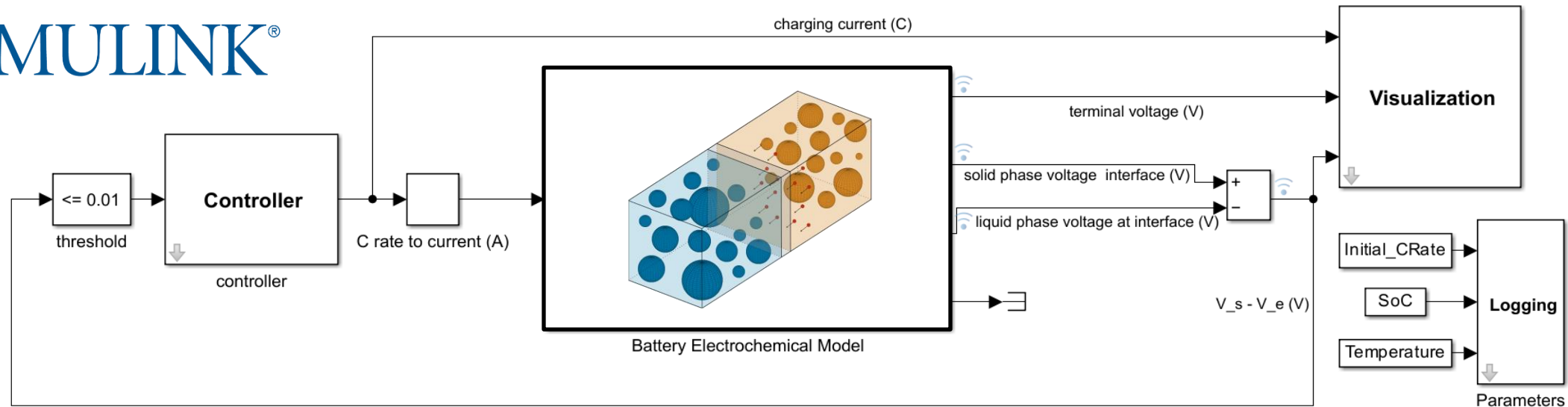


Closed-loop fast charging control

Example overview

Replacing a high-fidelity electrochemical battery model with an AI-based reduced order model

SIMULINK®



~52 minutes for 1 simulation run



Control design

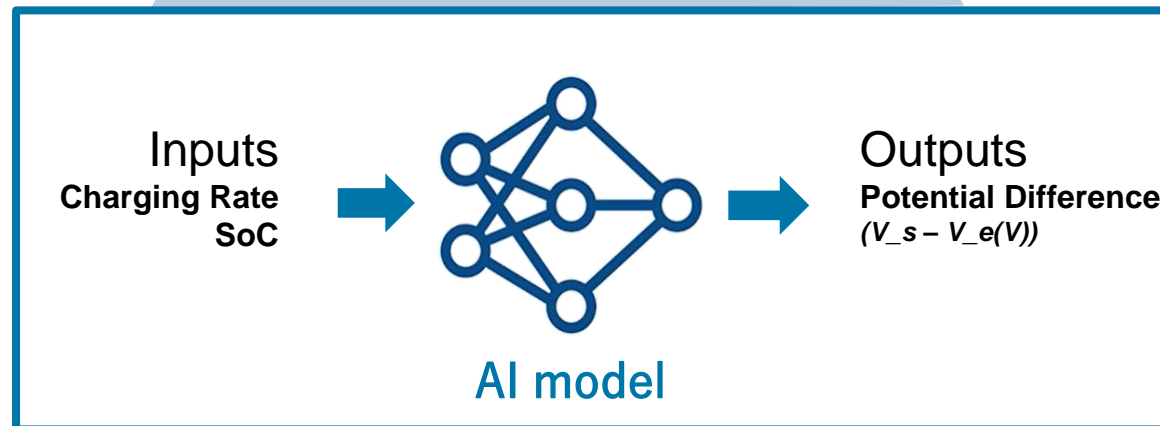
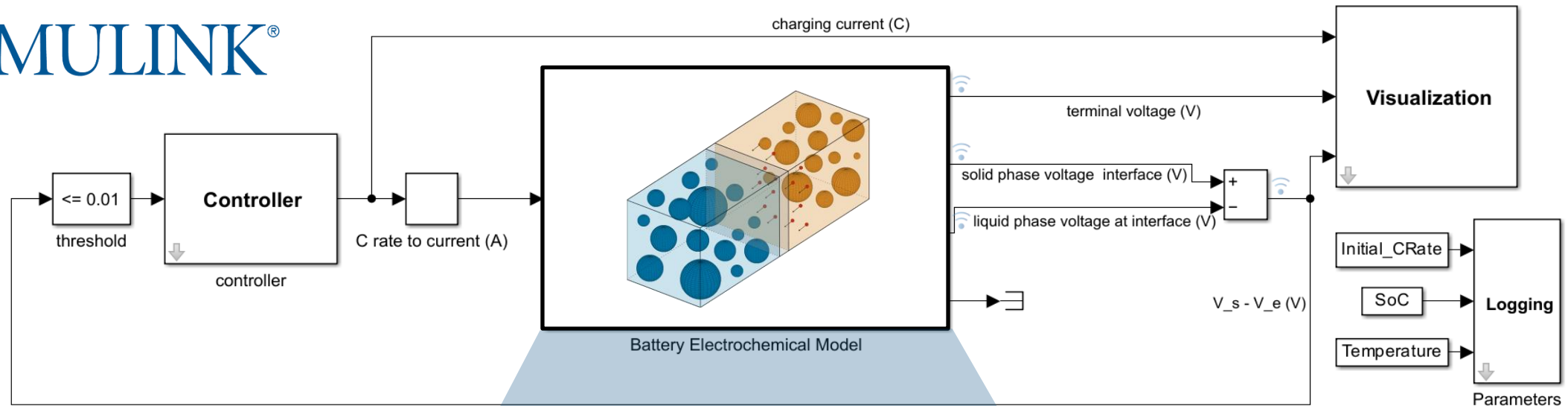
Hardware-in-the-Loop (HIL) testing

Embedded deployment

Example overview

Replacing a high-fidelity electrochemical battery model with an AI-based reduced order model

SIMULINK®



Create AI-Based ROMs using the Reduced Order Modeling Support Package

Set up Design of Experiments (DoE)

Generate input-output data from full-order, high-fidelity subsystems

Train and compare AI-based reduced order models using preconfigured templates

Export trained reduced order models into Simulink or outside of Simulink through FMUs

Products and Services

Reduced Order Modeling with MATLAB and Simulink

Create AI-based reduced order models

[Download add-on \(beta\)](#)

Simulink Add-On for Reduced Order Modeling is modeled in Simulink, including full-order models for system-level desktop simulation, high-fidelity subsystems, and reduced order models.

With Simulink Add-On for Reduced Order Modeling, you can:

- Set up the design of experiments and generate input-output data from full-order, high-fidelity subsystems.
- Train and compare AI-based reduced order models using preconfigured templates.
- Export AI-based surrogate models to MATLAB or Simulink.
- Export reduced order models as Functional Mock-up Units (FMUs) for use in Simulink or outside of Simulink.

Reduced Order Modeler App

The app interface includes a toolbar with icons for New, Open, Save, Edit, Simulation Options, Run, Open Results, and Export. It also features a 'Reduced Order Modeler' tab with sub-tabs for ROM Input, ROM Output, and Simulation Input. The main workspace displays several plots: a bar chart for Ambient, a scatter plot for Cooling, a bar chart for Pressure, and a scatter plot for Pressure. A table on the right shows the pulse amplitude ranges for the signals.

Signal	Min	Max
1 Ambient	800	2000
2 Cooling	90	250
3 Pressure	450000	550000

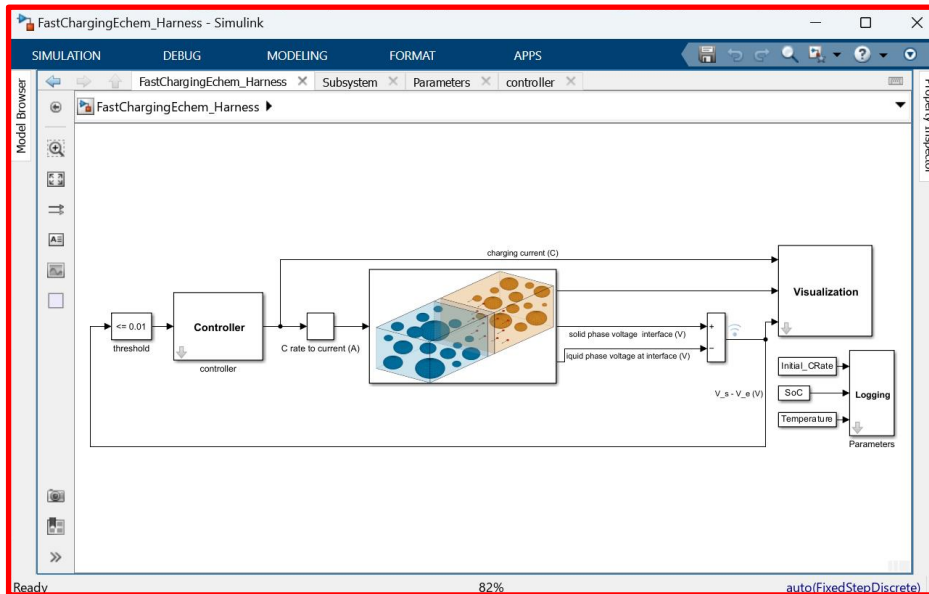
Session opened: Full_model_ROMSession_final.mat

Full-order model CFD/CAE/FEA

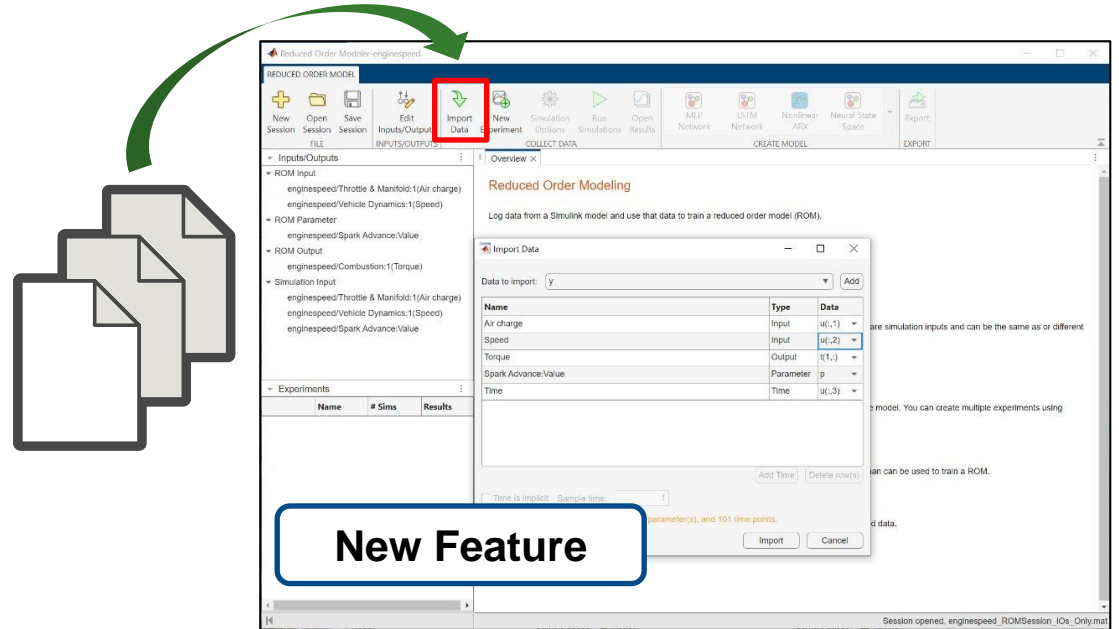
Reduced Order Modeling

The diagram illustrates the workflow: A Full-order model (CFD/CAE/FEA) is used for Reduced Order Modeling. This process involves training a Full-order model and a Reduced order model, comparing their fidelity against speed. The resulting Reduced order model is then used to create component models (component 1, component 2, component 3) which are first-principles based.

Prepare data for training AI models



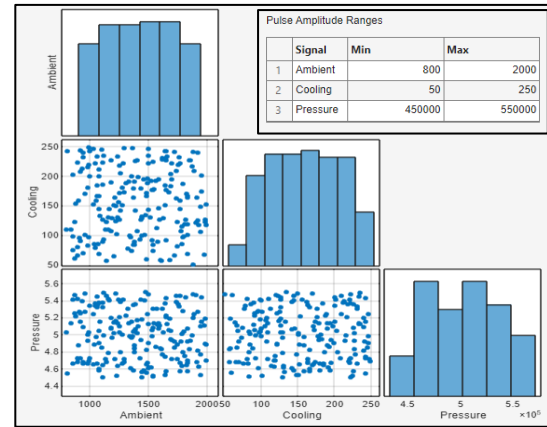
Generate synthetic data from
Simulink/Simscape models



Import pre-collected data from high-fidelity model into the app

Synthetic data generation

Design of Experiments

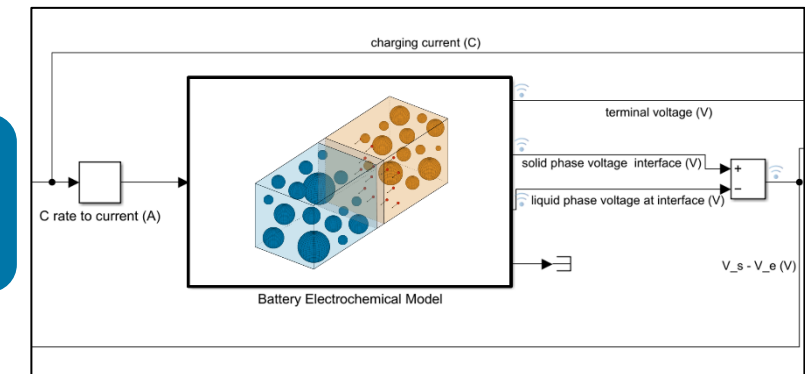
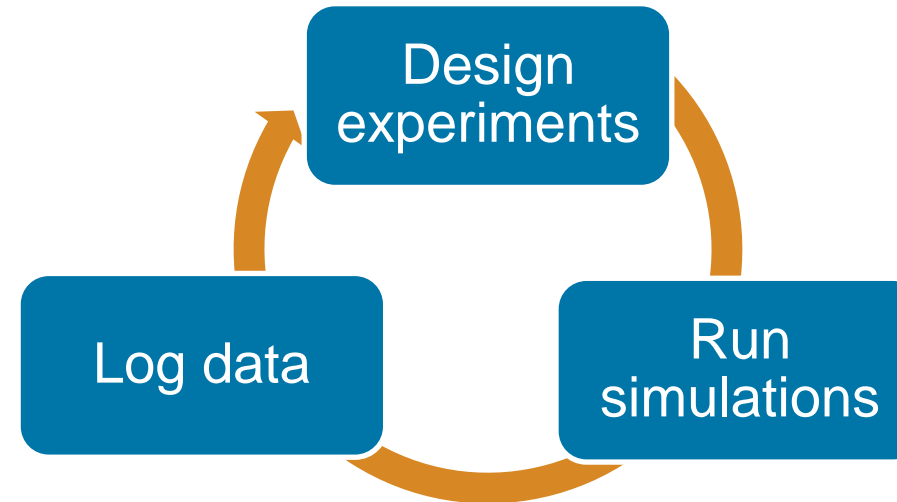


Input features

Charging Current Rate (C)
Initial SoC

Response

Potential Difference



Data Preparation

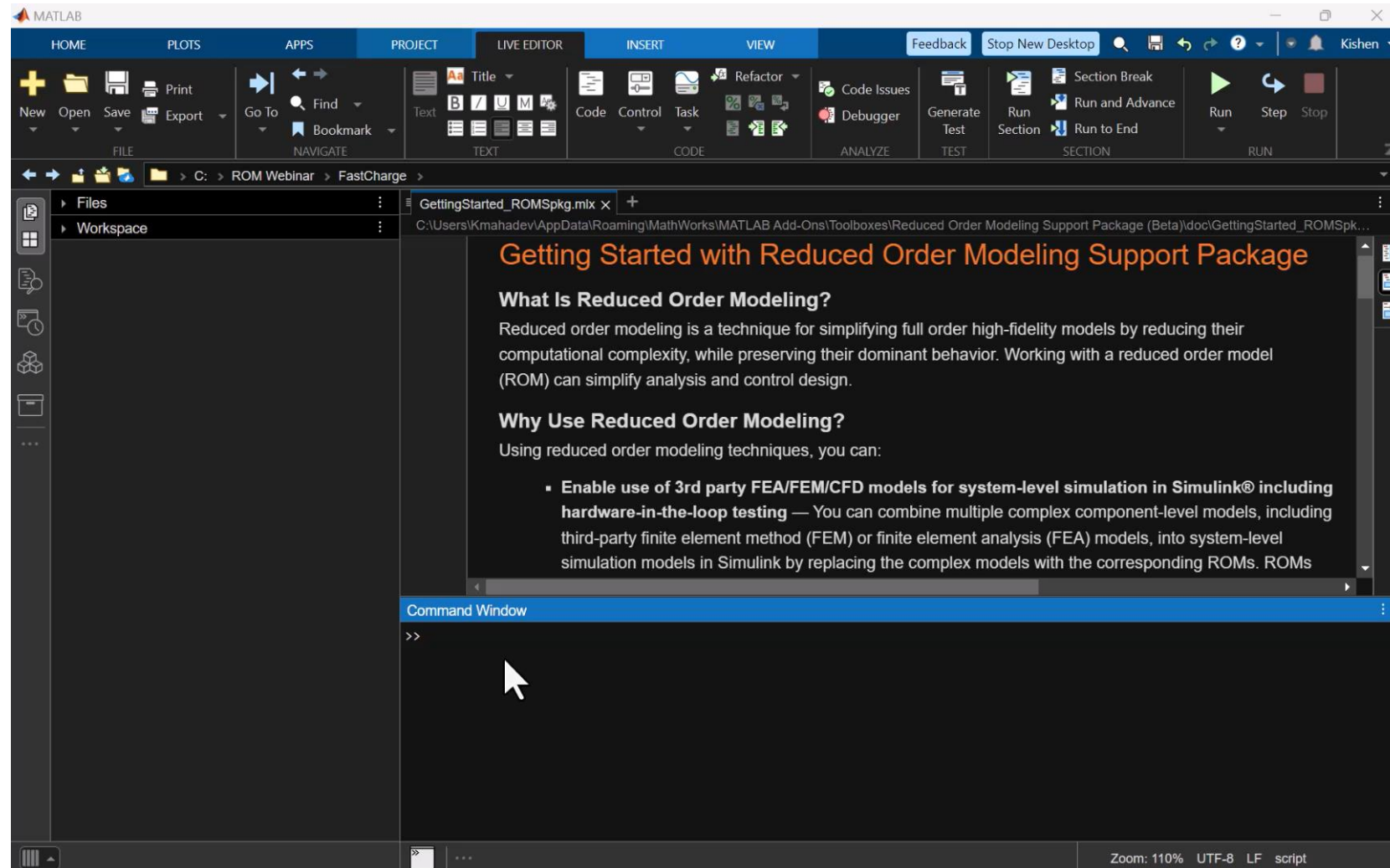
AI Modeling

Simulation & Test

Deployment

Synthetic data generation

Design of Experiments



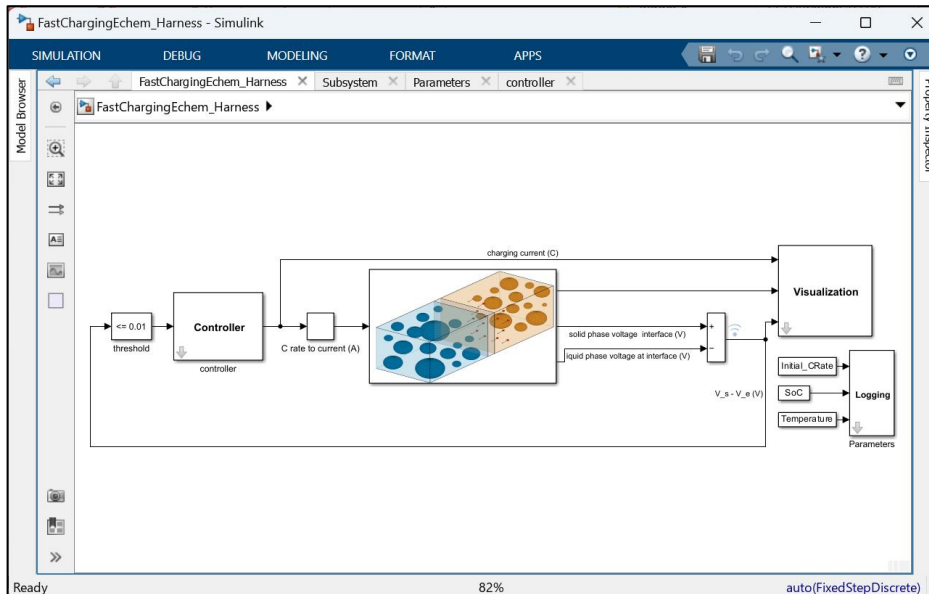
Data Preparation

AI Modeling

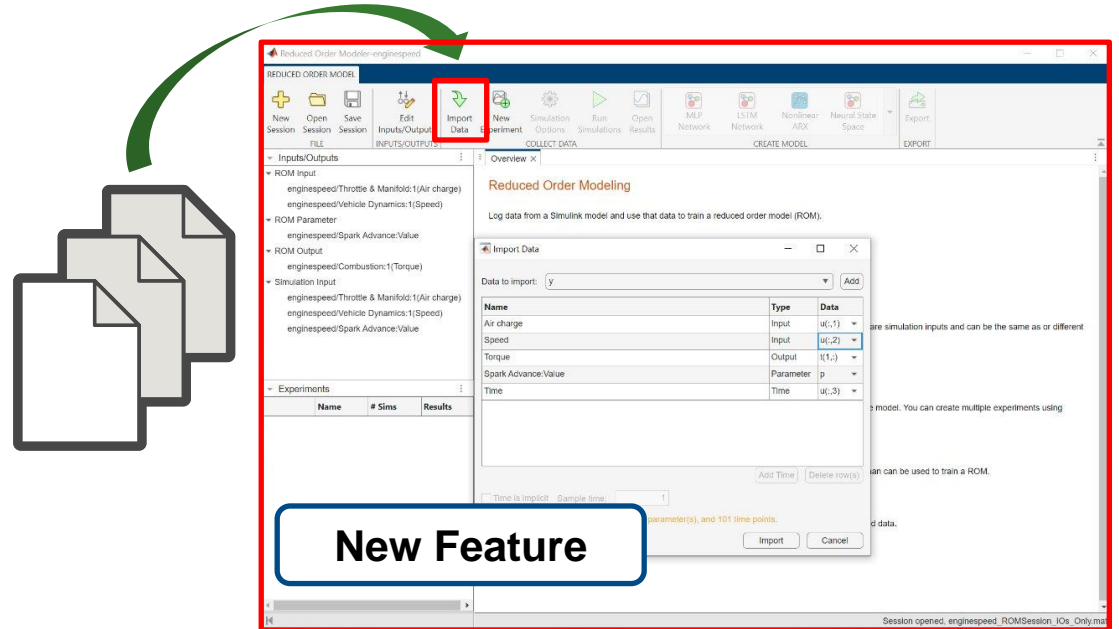
Simulation & Test

Deployment

Prepare data for training AI models



Generate synthetic data from
Simulink/Simscape models



Import pre-collected data from high-
fidelity model into the app

Data Source

Deep Residual Convolutional and Recurrent Neural Networks for Temperature Estimation in Permanent Magnet Synchronous Motors

Wilhelm Kirchgässner
*Department of Power Electronics
and Electrical Drives*
Paderborn University
33095 Paderborn, Germany
kirchgaessner@lea.uni-paderborn.de

Oliver Wallscheid
*Department of Power Electronics
and Electrical Drives*
Paderborn University
33095 Paderborn, Germany
wallscheid@lea.uni-paderborn.de

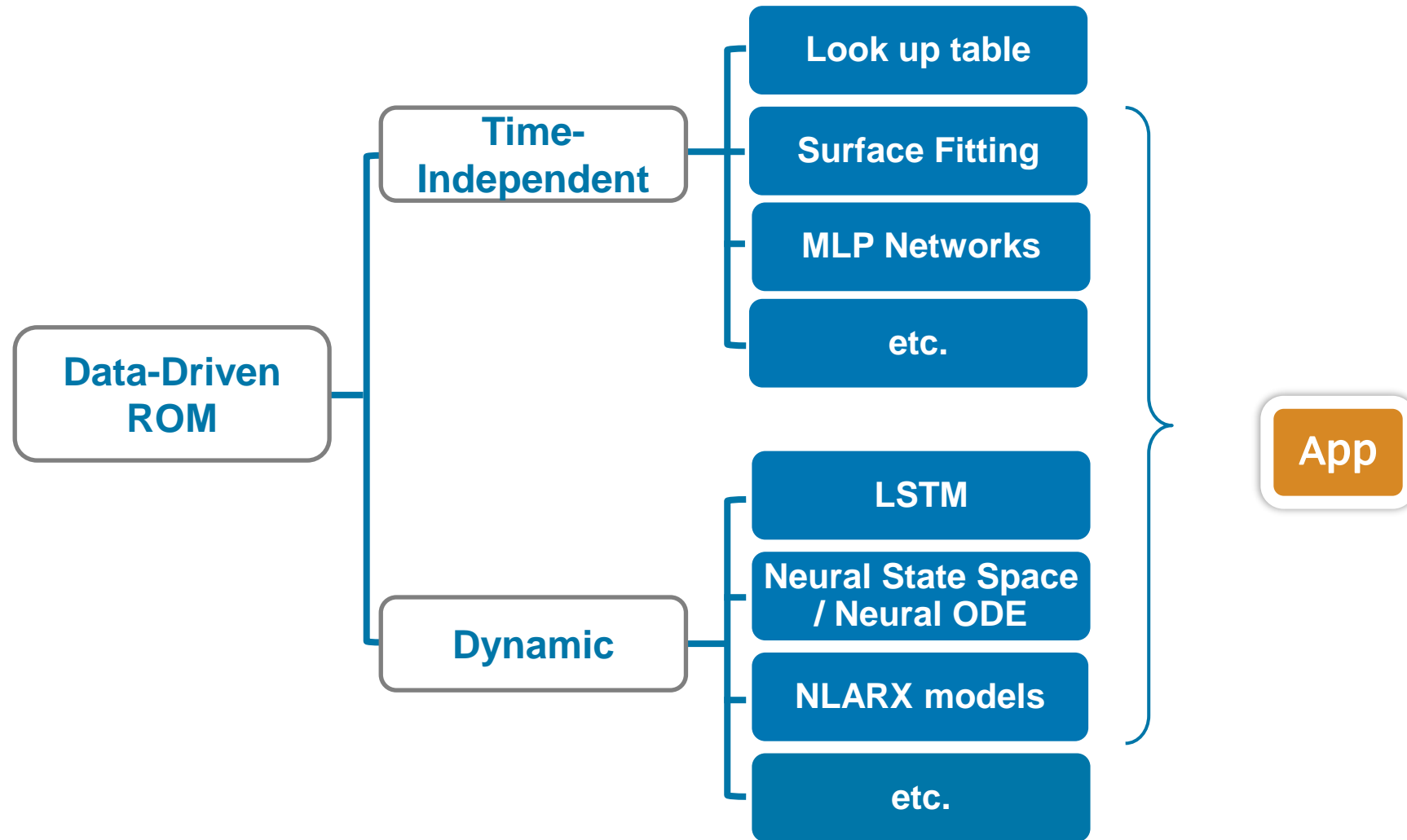
Joachim Böcker
*Department of Power Electronics
and Electrical Drives*
Paderborn University
33095 Paderborn, Germany
boecker@lea.uni-paderborn.de

Abstract—Most traction drive applications using permanent magnet synchronous motors (PMSMs) lack accurate temperature monitoring capabilities so that safe operation is ensured through expensive, oversized materials at the cost of its effective utilization. Classic thermal modeling is conducted with e.g. lumped-parameter thermal networks (LPTNs), which help to estimate internal component temperatures rather precisely but also require expertise in choosing model parameters and lack physical interpretability as soon as their degrees of freedom are curtailed in order to meet the real-time requirement. In this work, deep recurrent and convolutional neural networks with residual connections are empirically evaluated for their feasibility on the sequence learning task of predicting latent high-dynamic temperatures inside PMSMs, which, to the authors' best knowledge, has not been elaborated in previous literature. In a highly utilized PMSM for electric vehicle applications, the temperature profile in the stator teeth, winding, and yoke as well as the rotor's permanent magnets are modeled while their ground

precise thermal state, yet for the rotor part, it is technically and economically infeasible due to an electric motor's sophisticated internal structure and the difficult accessibility of the rotor. Stator temperature monitoring is realized with thermal sensors, but these are usually firmly embedded in the stator so that replacement is not an option, although sensor functionality deteriorates steadily. Since competitive pressure demands perpetual reduction of production costs, there is a commercial interest driving the investigation of sufficiently accurate real-time temperature estimation. In the last decades, various research efforts led to approaches that approximate the heat transfer process e.g. with equivalent circuit diagrams [2] called lumped-parameter thermal networks (LPTNs). This kind of model must forfeit physical interpretability of its structure and parameter values by significantly curtailing degrees of



Data-driven ROM



Reduced Order Modeler-FastChargingEchem_Harness

REDUCED ORDER MODEL SIMULATION RESULT

FILE INPUTS/OUTPUTS COLLECT DATA CREATE MODEL EXPORT

New Session Open Session Save Session Edit Inputs/Outputs Import Data New Experiment Simulation Options Run Simulations Open Results MLP Network LSTM Network Nonlinear ARX **Neural State Space** Export

Inputs/Outputs

- ROM Input
 - FastChargingEchem_Harness/Unit Delay:1(charging c
- ROM Parameter
 - FastChargingEchem_Harness:SoC
- ROM Output
 - FastChargingEchem_Harness/Subtract:1(V_s - V_e (V
- Simulation Input
 - FastChargingEchem_Harness:SoC

Experiments

	Name	# Sims	Results
<input checked="" type="checkbox"/>	SoC_Values	5	Data
<input checked="" type="checkbox"/>	SoC_Distributions	25	Data

Overview X Result: SoC_Values X Result: SoC_Distributions X

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

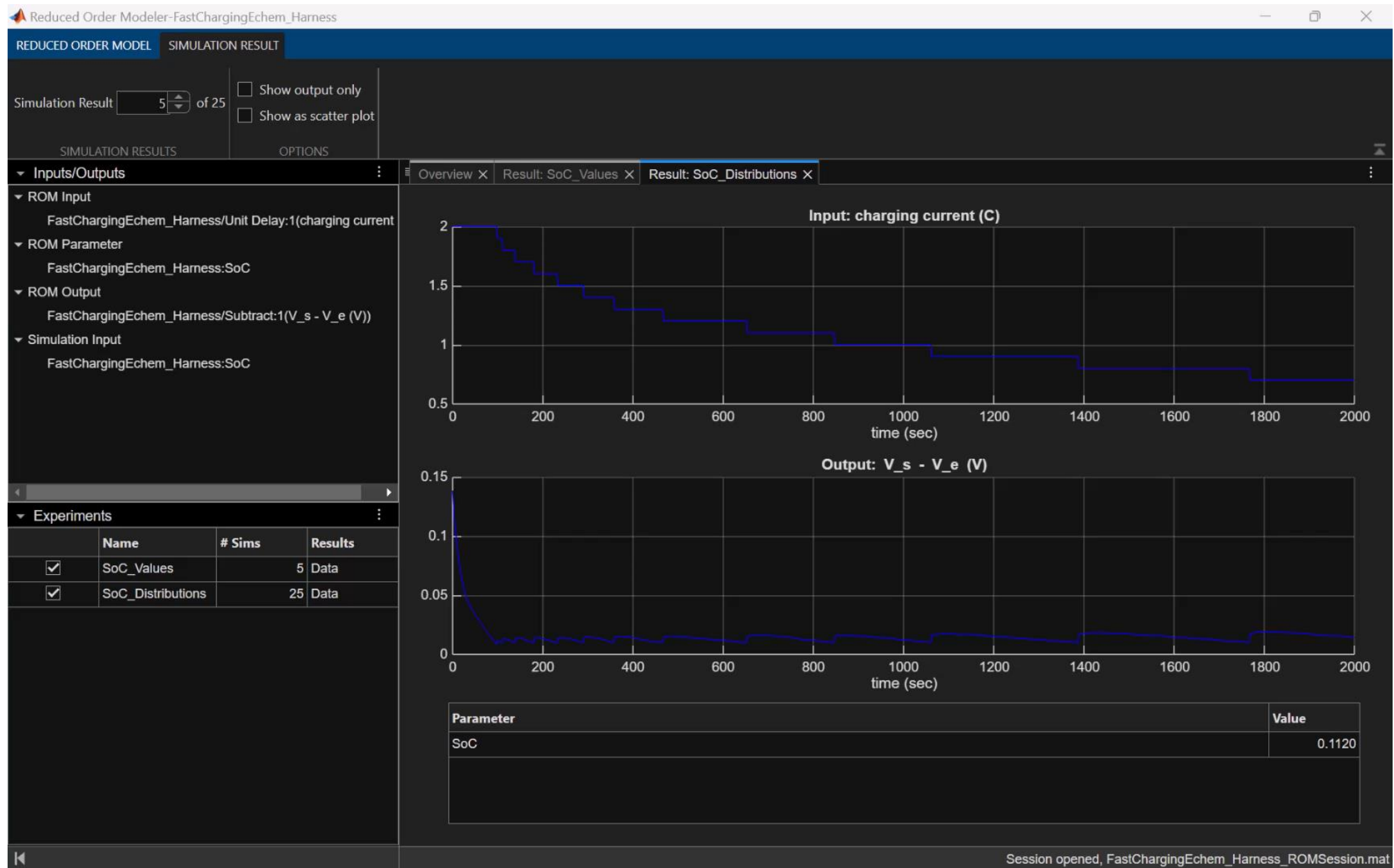
State Network (f)

Output Network (g)

AI-based ROM using Neural State Space (also known as Neural ODE)

Create Deep Learning-based nonlinear state-space models

Session opened, FastChargingEchem_Harness_ROMSession.mat



Reduced Order Modeler-FastChargingEchem_Harness

REDUCED ORDER MODEL SIMULATION RESULT

FILE INPUTS/OUTPUTS COLLECT DATA MODEL EXPORT

New Session Open Session Save Session Edit Inputs/Outputs Import Data New Experiment Simulation Options Run Simulations Open Results MLP Network **LSTM Network** Nonlinear ARX Neural State Space Export

Inputs/Outputs Overview X Result: SoC_Values X Result: SoC_D

ROM Input FastChargingEchem_Harness/Unit Delay:1(c

ROM Parameter FastChargingEchem_Harness:SoC

ROM Output FastChargingEchem_Harness/Subtract:1(V

Simulation Input FastChargingEchem_Harness:SoC

Experiments

	Name	# Sims	Re
<input checked="" type="checkbox"/>	SoC_Values	5	Da
<input checked="" type="checkbox"/>	SoC_Distributions	25	Da

Inputs

Sequenceinput sequenceInput...

lstm lstmLayer

fc fullyConnected...

fc2 fullyConnected...

regressionout... regressionLayer

Outputs

Diagram illustrating the LSTM architecture:

Inputs: $x_n, \dots, x_2, x_1, x_0$

Output: h_0

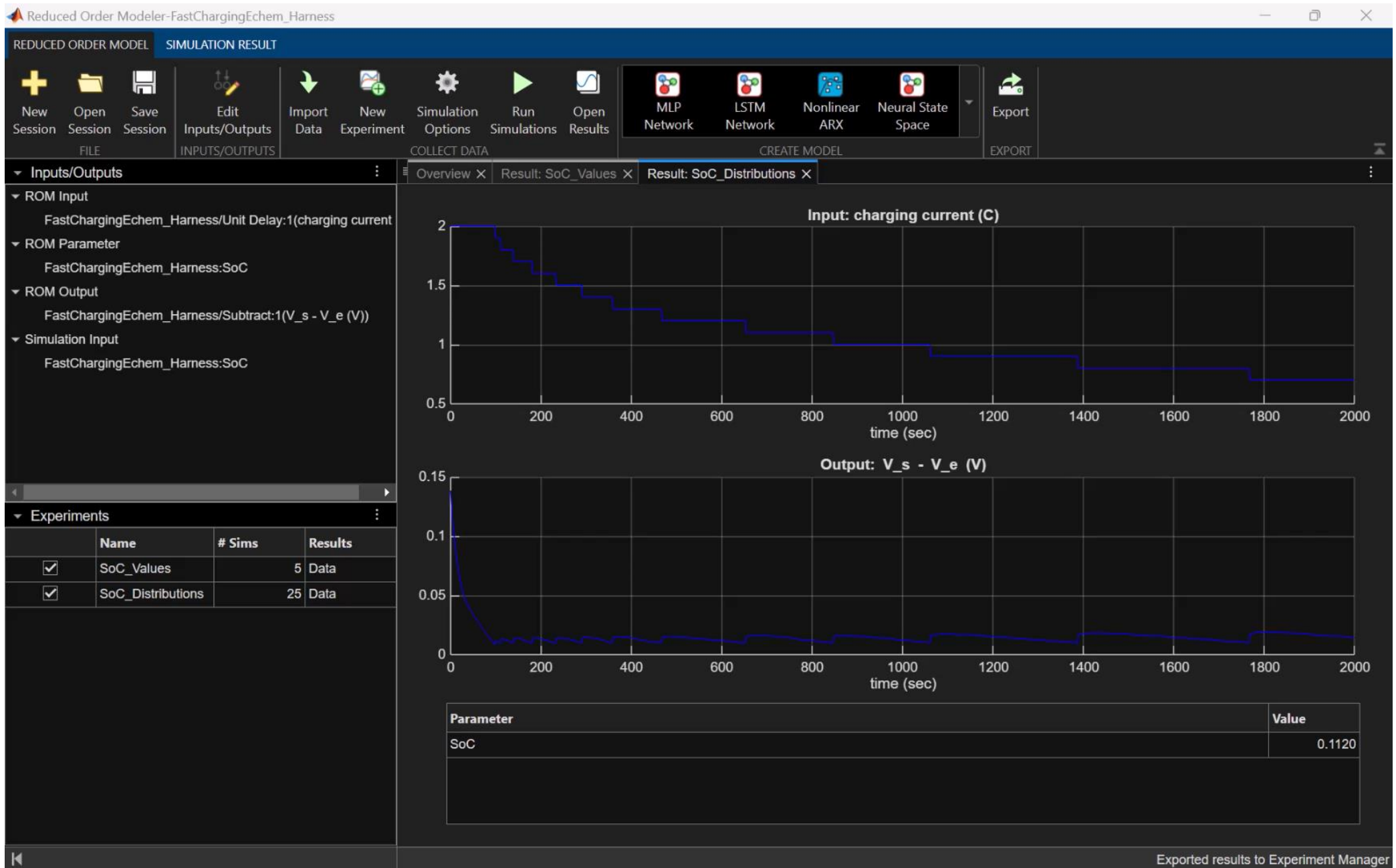
Internal state: State

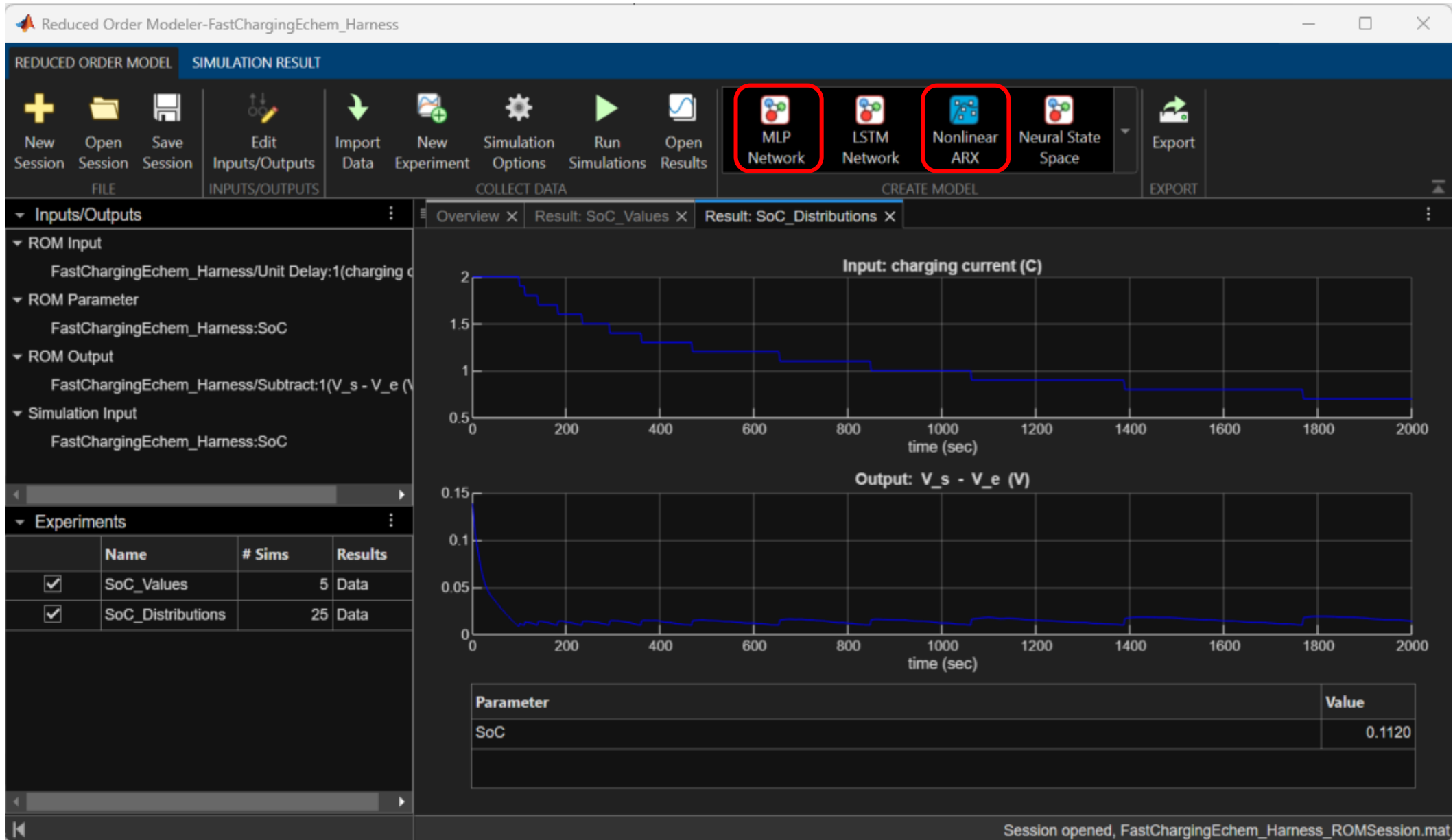
Output is used with next time step

AI-based ROM using LSTMs

Capture time dependencies in time-series data

Session opened, FastChargingEchem_Harness_ROMSession.mat





Data Preparation

AI Modeling

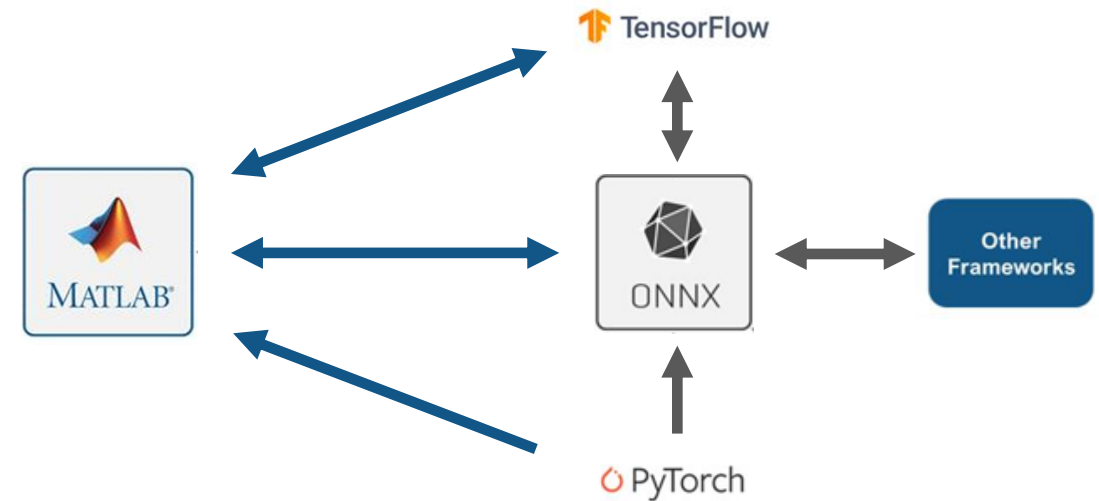
Simulation & Test

Deployment

MATLAB interoperates with other frameworks

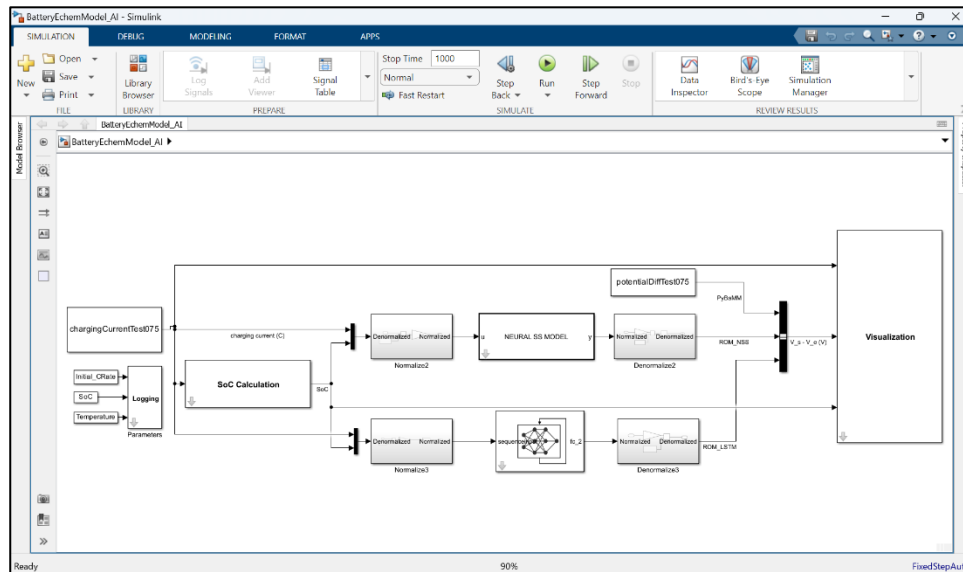
Framework interoperability bridges the gap between data science, engineering and production

TensorFlow-Keras Import	R2017b
ONNX Converter (Import & Export)	R2018a
TensorFlow Converter (Import)	R2021a
TensorFlow Converter (Export)	R2022b
PyTorch Converter (Import)	R2022b

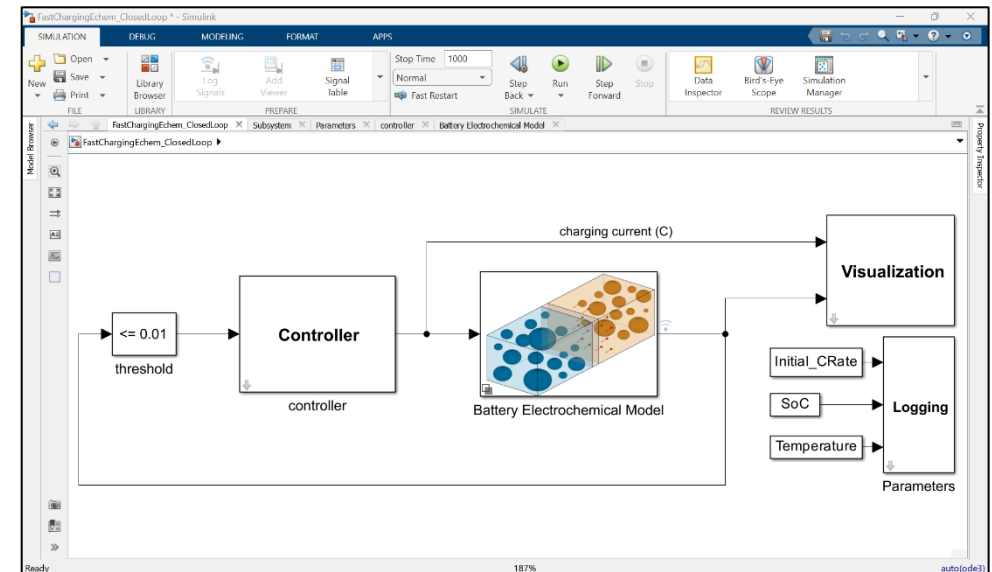


Integrate your AI model for system-level simulation and test

Integration of trained AI model into Simulink

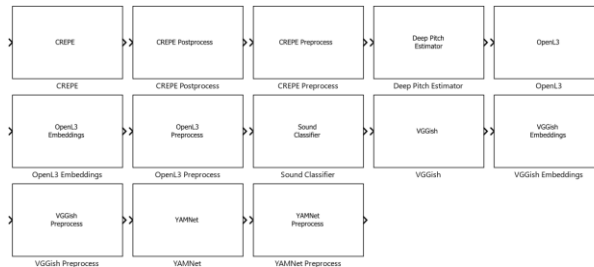


System-level simulation

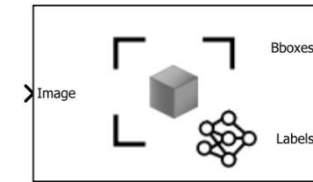


AI libraries in Simulink are expanding to include more AI blocks for more applications

Specialized

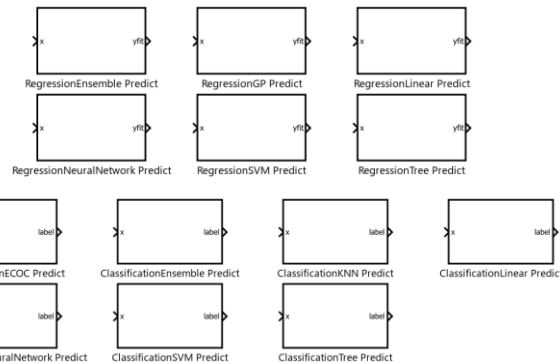


Audio Toolbox



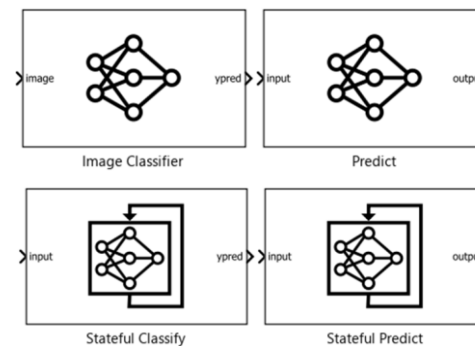
Deep Learning Object Detector

Computer Vision Toolbox

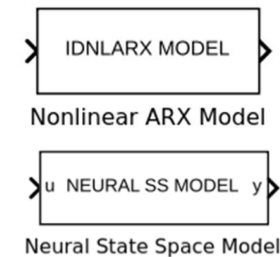


Statistics and Machine Learning Toolbox

AI Core

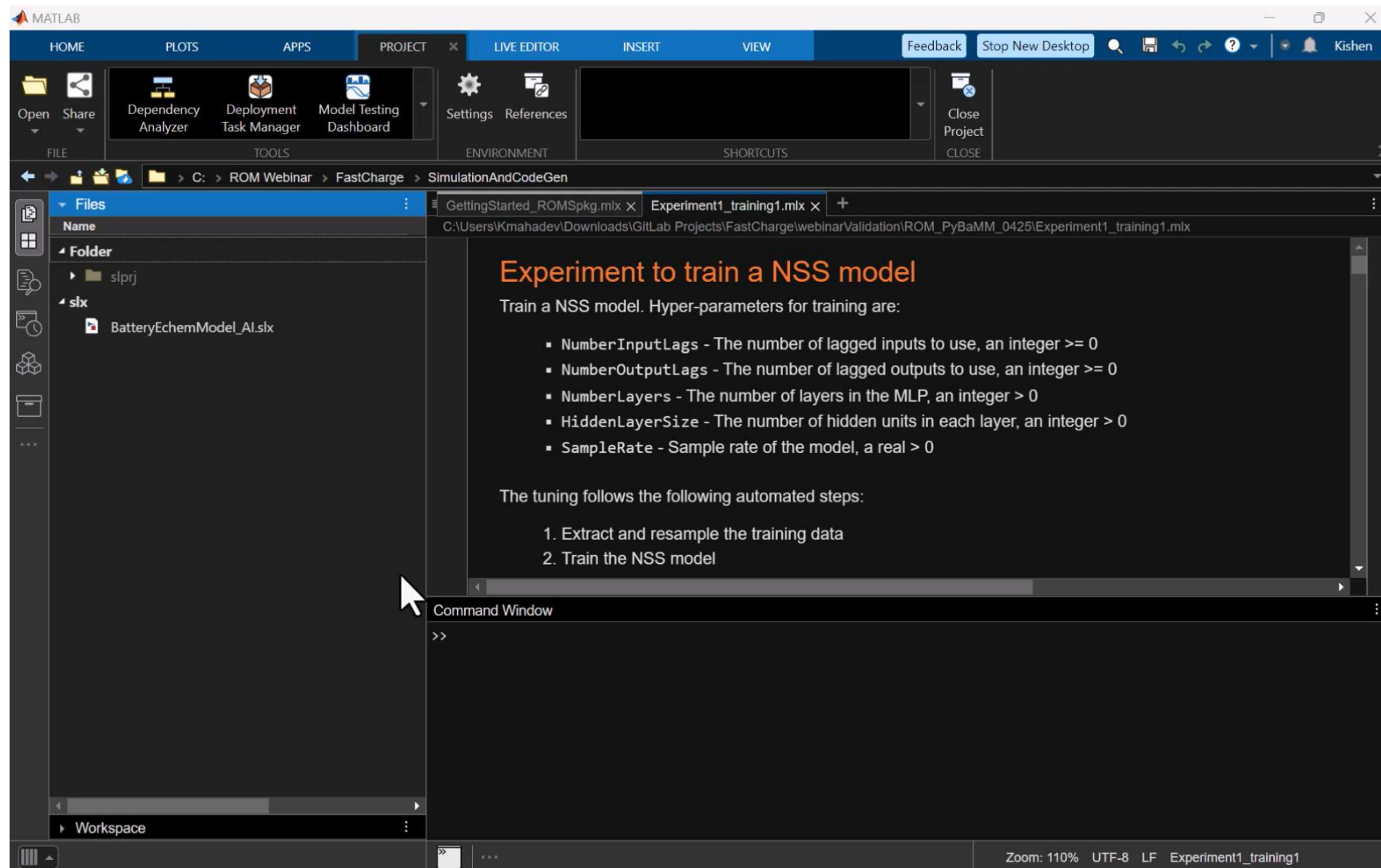


Deep Learning Toolbox







System Identification Toolbox

Integration of trained AI models into Simulink



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
▼ BatteryROM		8.926	1.101	3012
> Stateful Predict		5.429	0.000	0
Scope		2.029	2.029	337
> Neural State Space Model1		0.319	0.008	1338
> Denormalize3		0.014	0.000	0
> Denormalize2		0.012	0.000	0
From Workspace		0.009	0.009	1002
Integrator Limited1		0.004	0.004	2336
> Normalize2		0.002	0.000	0
> Normalize3		0.002	0.000	0
From Workspace1		0.002	0.002	1002
Initial SOC1		0.001	0.001	1002
Gain1		0.001	0.001	1002

Neural state-space model took
~0.3 seconds when compared to
~52 minutes from high-fidelity
 battery model

Deep Learning Toolbox Verification Library

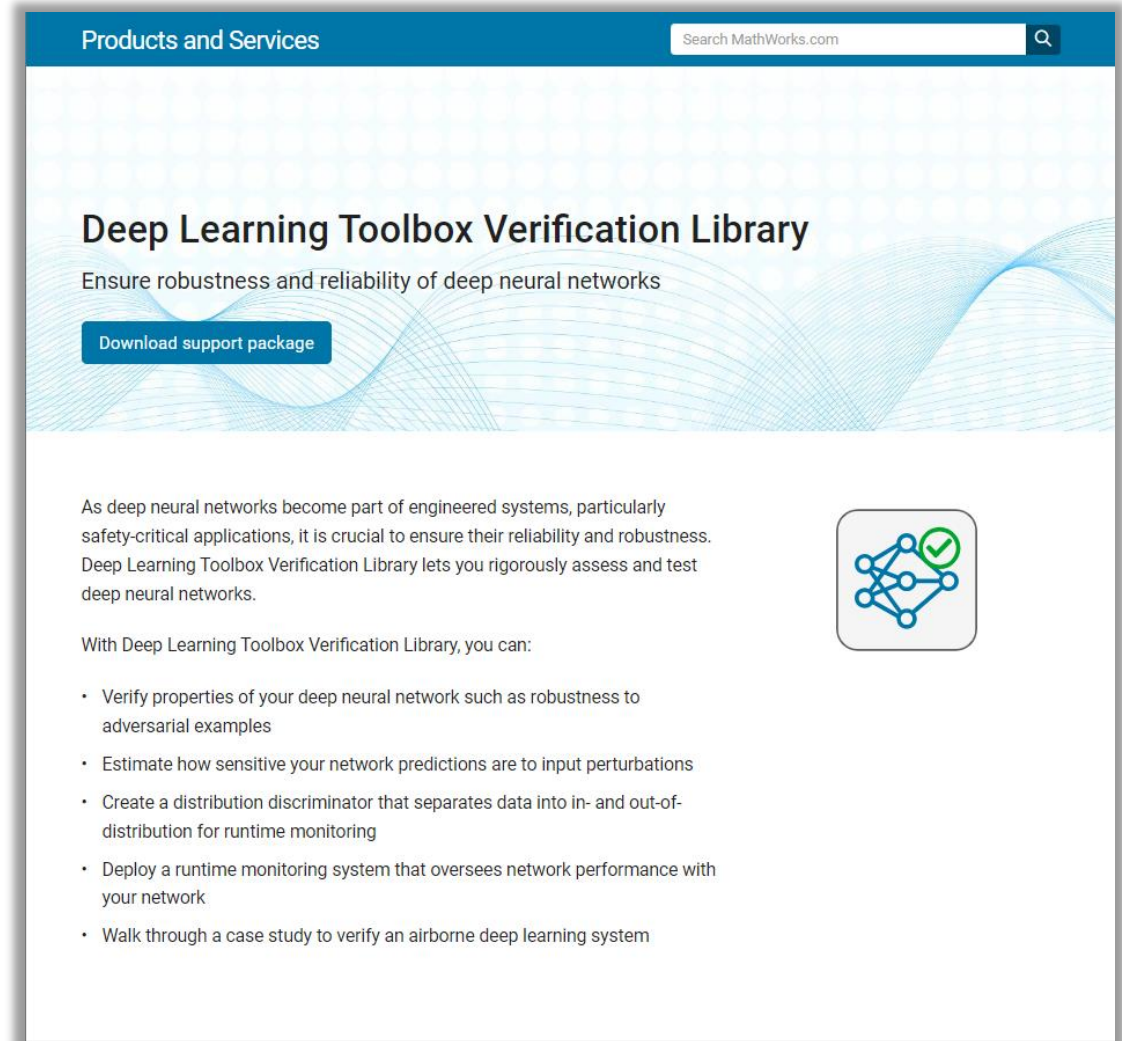
Ensure robustness and reliability of deep neural networks

**Verify Deep Neural Network
Robustness for Classification**

**Estimate Deep Neural Network
Output Bounds for Regression**

**Build Safe Deep Learning Systems
with Runtime Monitoring**

**Case Study: Verifying an
Airborne Deep Learning System**



The screenshot shows the MathWorks website for the Deep Learning Toolbox Verification Library. The header includes 'Products and Services' and a search bar. The main heading is 'Deep Learning Toolbox Verification Library' with the subtext 'Ensure robustness and reliability of deep neural networks'. A 'Download support package' button is visible. The content area explains the importance of verifying deep neural networks in safety-critical applications and lists capabilities: verifying robustness to adversarial examples, estimating output bounds for perturbations, creating distribution discriminators, deploying runtime monitoring, and a case study on an airborne system. A neural network icon with a checkmark is on the right.

Products and Services

Deep Learning Toolbox Verification Library

Ensure robustness and reliability of deep neural networks

[Download support package](#)

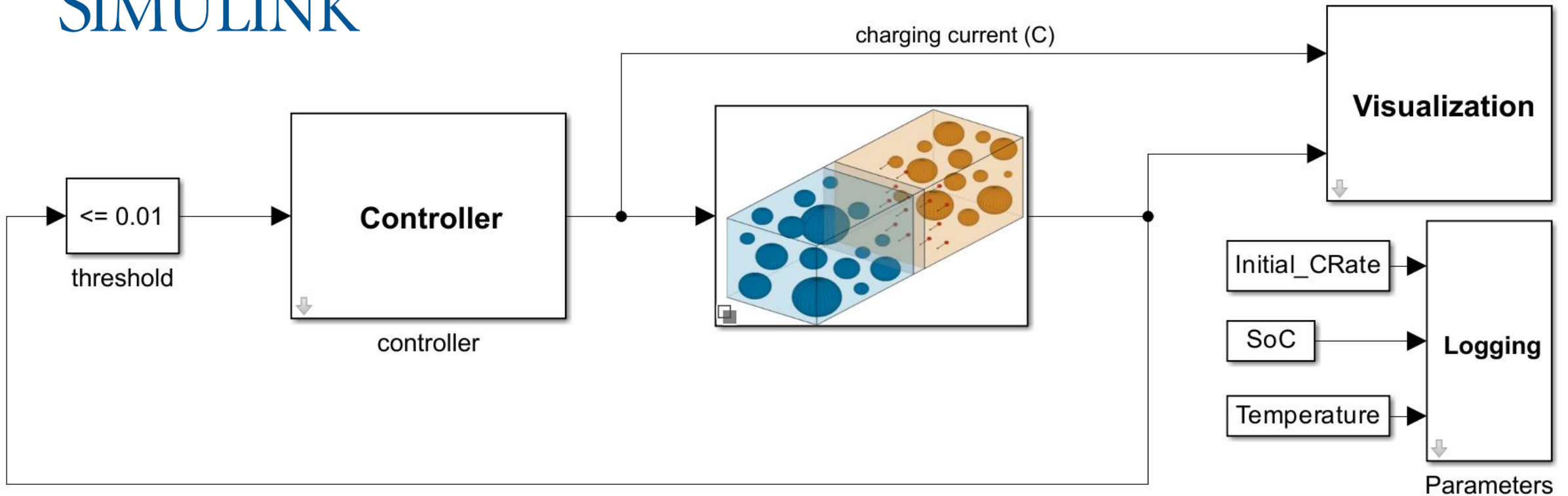
As deep neural networks become part of engineered systems, particularly safety-critical applications, it is crucial to ensure their reliability and robustness. Deep Learning Toolbox Verification Library lets you rigorously assess and test deep neural networks.

With Deep Learning Toolbox Verification Library, you can:

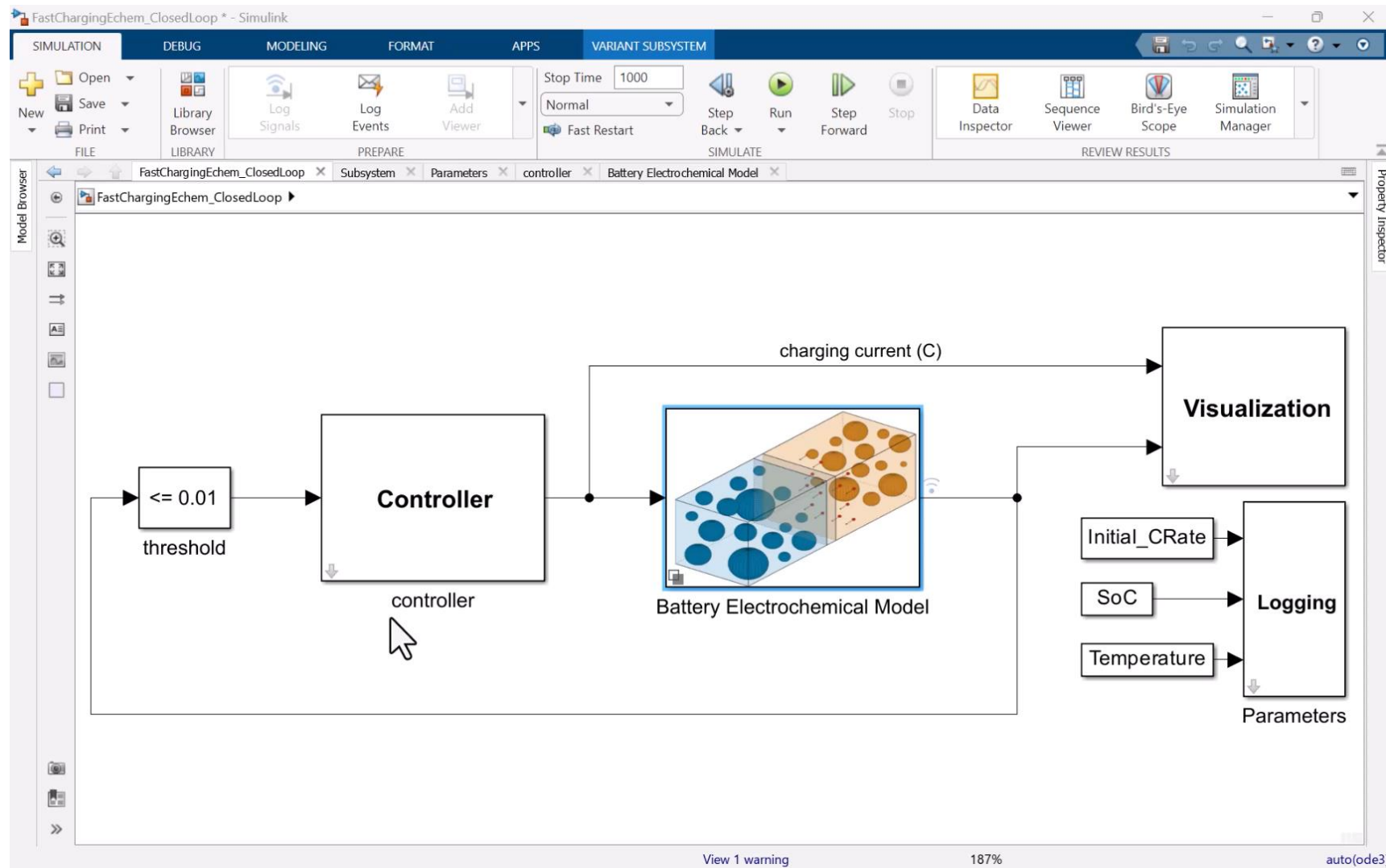
- Verify properties of your deep neural network such as robustness to adversarial examples
- Estimate how sensitive your network predictions are to input perturbations
- Create a distribution discriminator that separates data into in- and out-of-distribution for runtime monitoring
- Deploy a runtime monitoring system that oversees network performance with your network
- Walk through a case study to verify an airborne deep learning system

Closed-Loop fast charging control

SIMULINK®

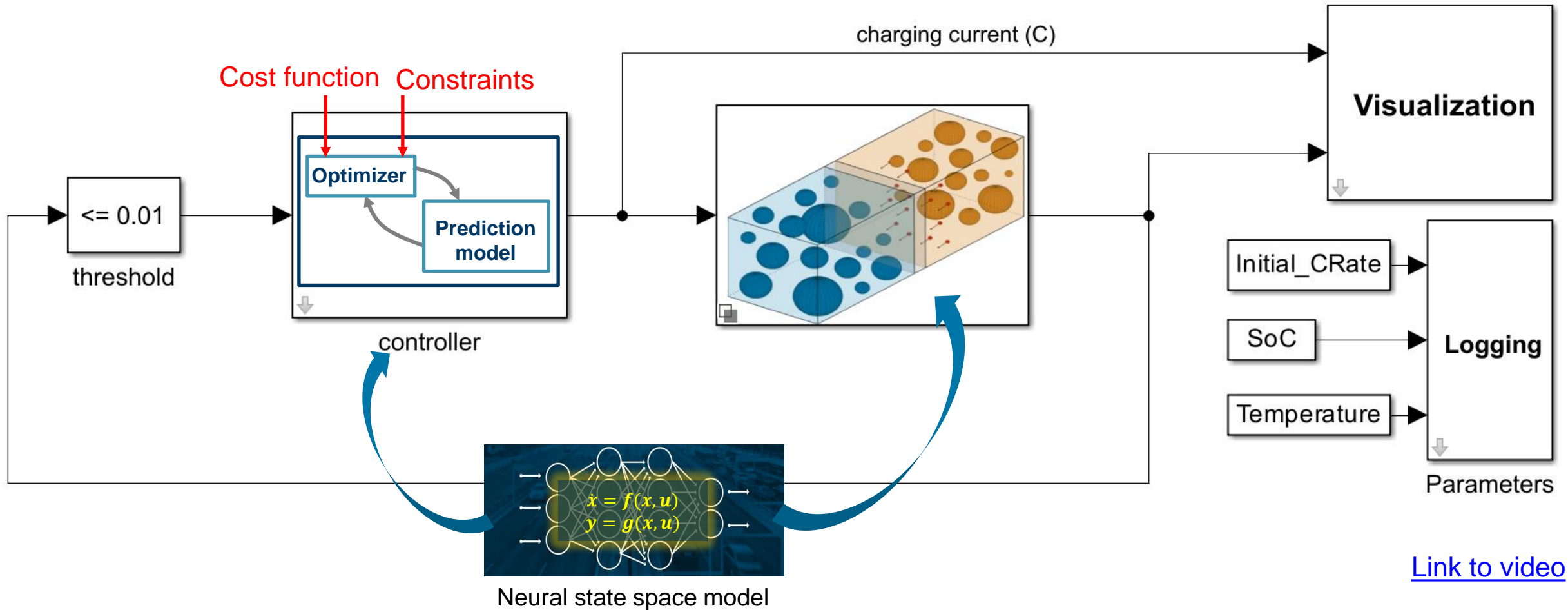


System-level simulation



Closed-Loop fast charging control with Model Predictive Controller

SIMULINK®



[Link to video](#)

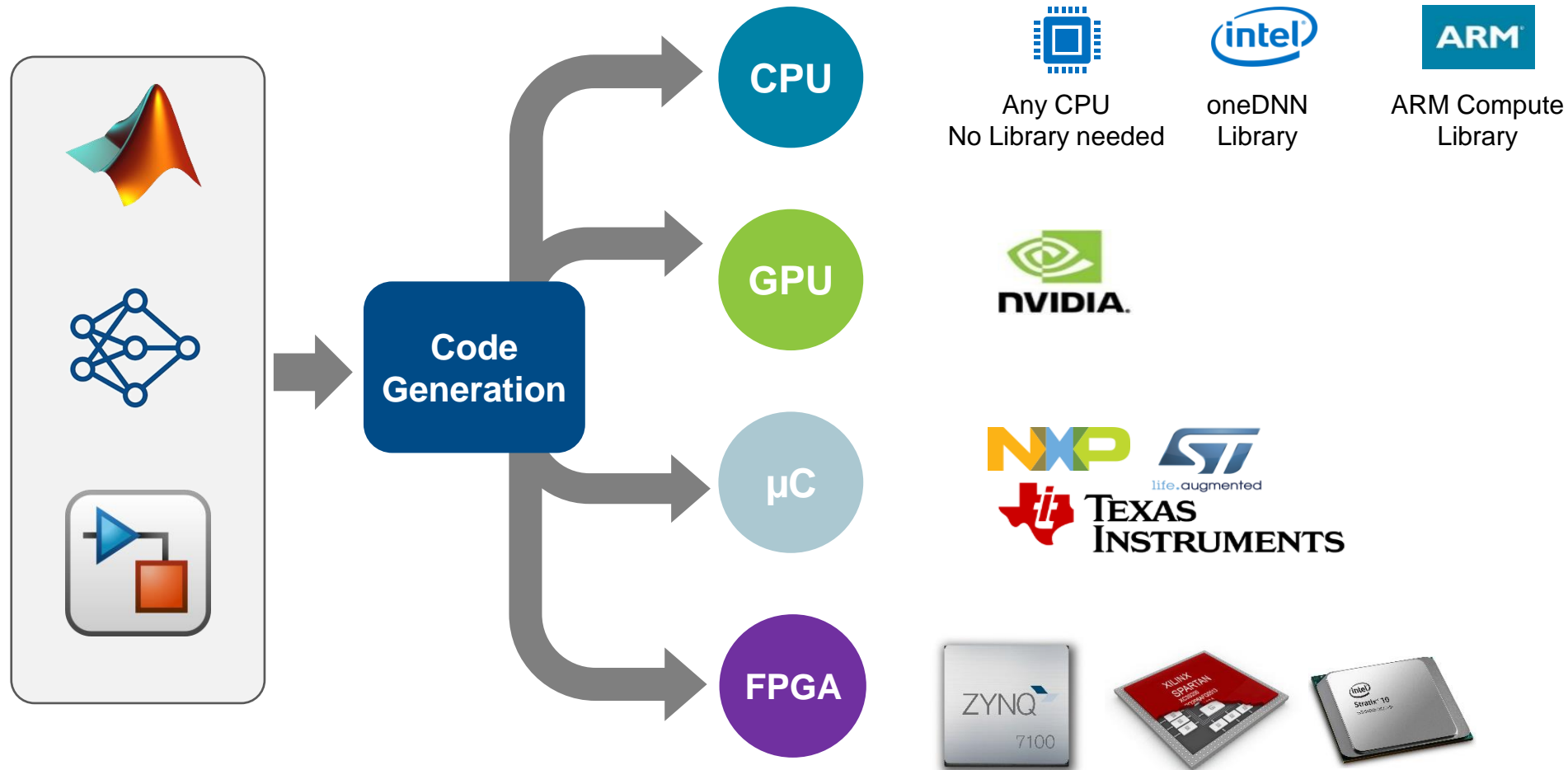
Data Preparation

AI Modeling

Simulation & Test

Deployment

Deploy to target with zero coding errors



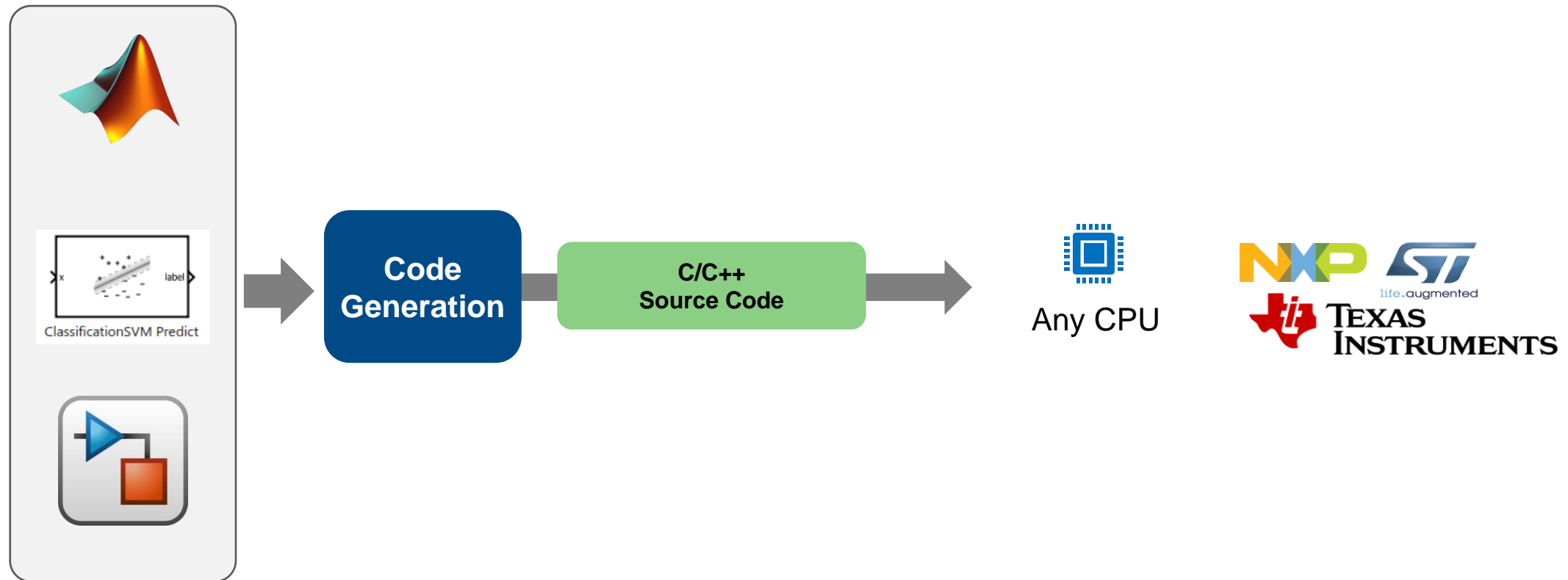
Data Preparation

AI Modeling

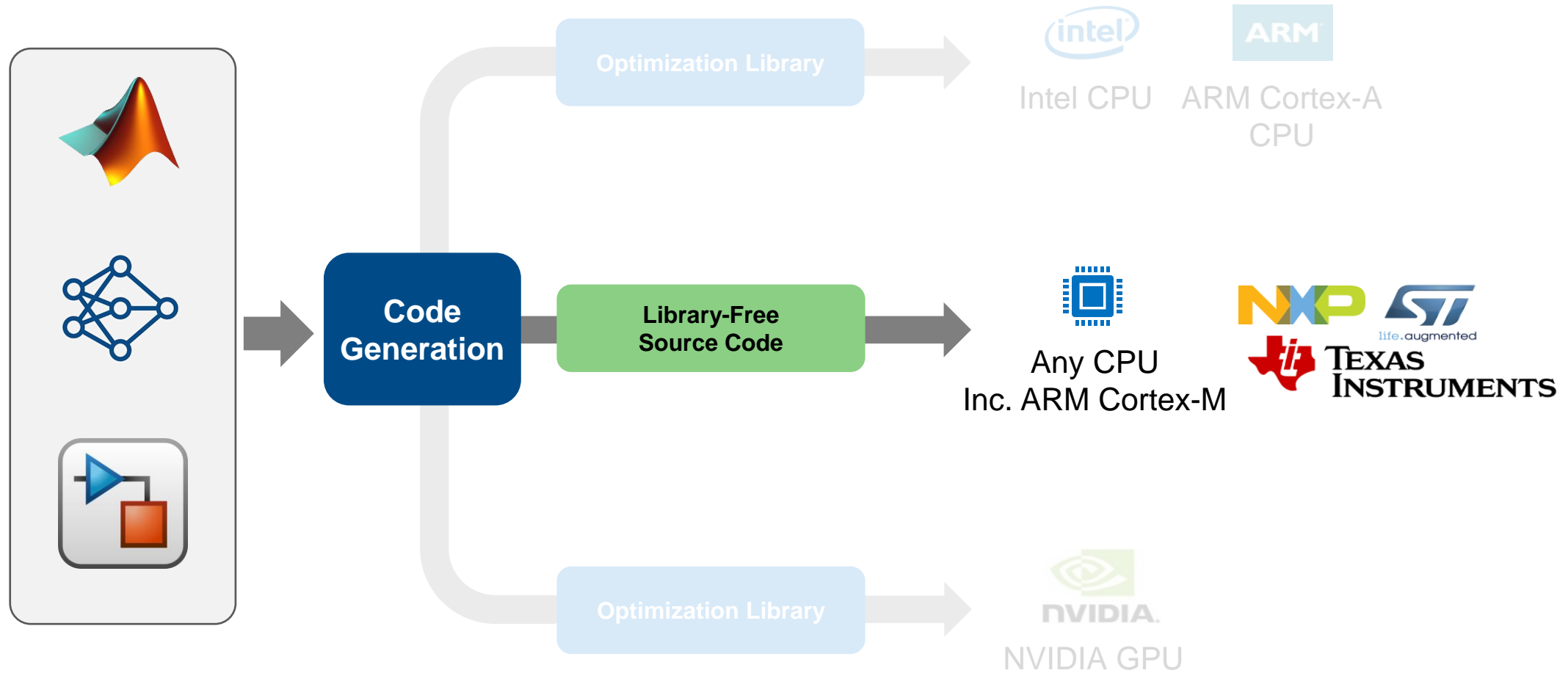
Simulation & Test

Deployment

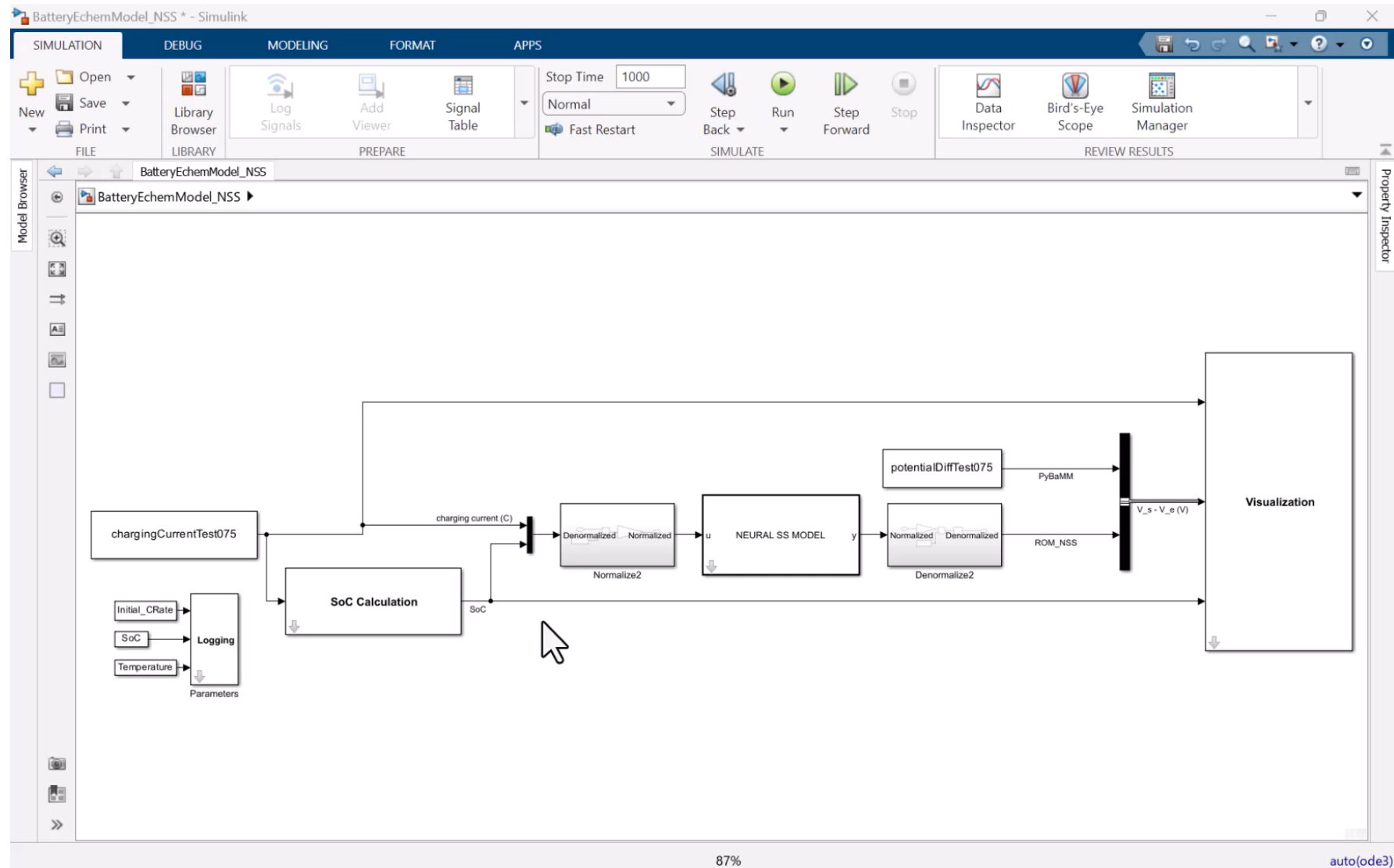
Use Embedded Coder to generate code for machine learning



Generate library-free C/C++ code for deep learning networks



Generate library-free C/C++ code for deep learning networks



Data Preparation

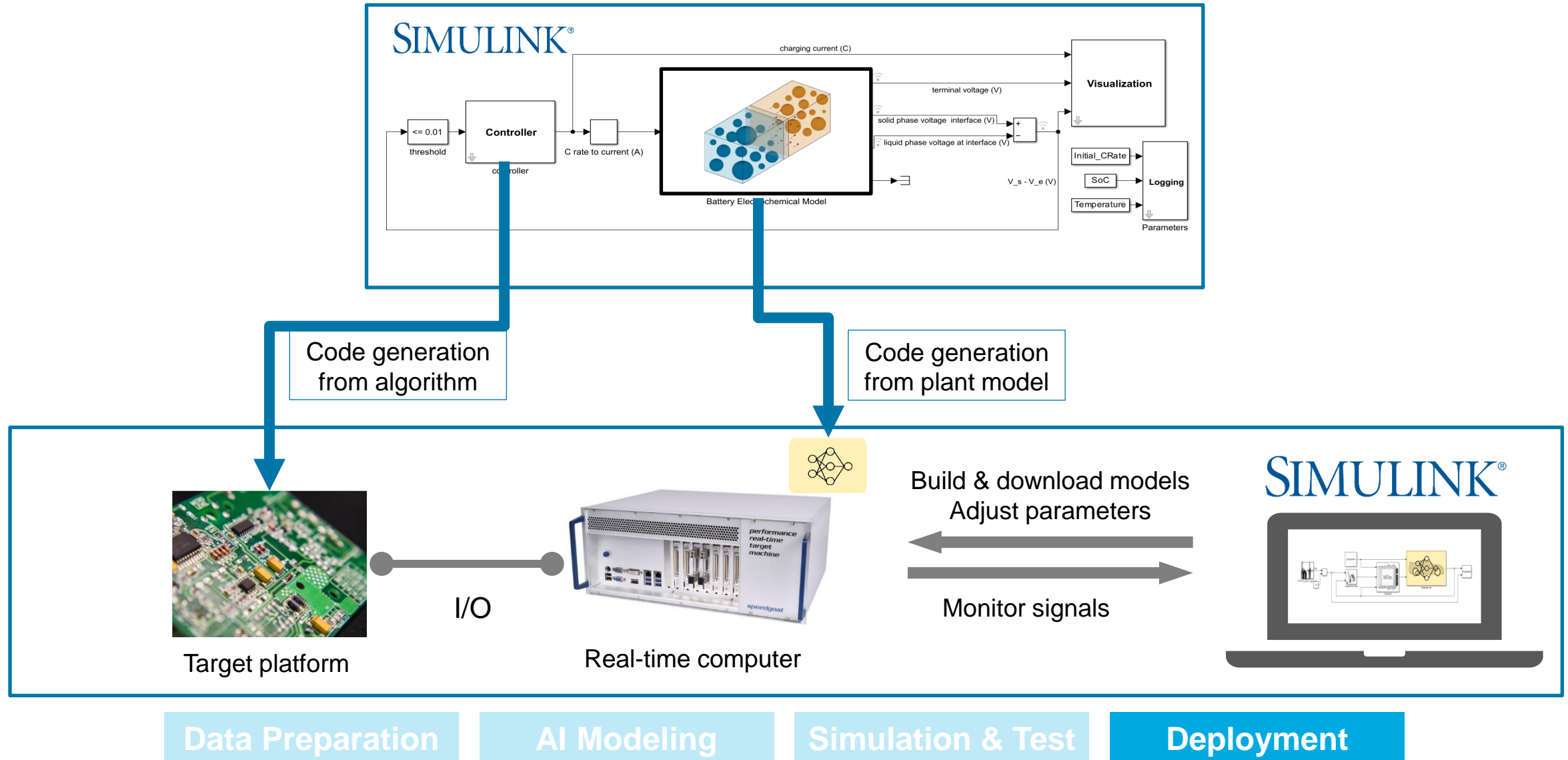
AI Modeling

Simulation & Test

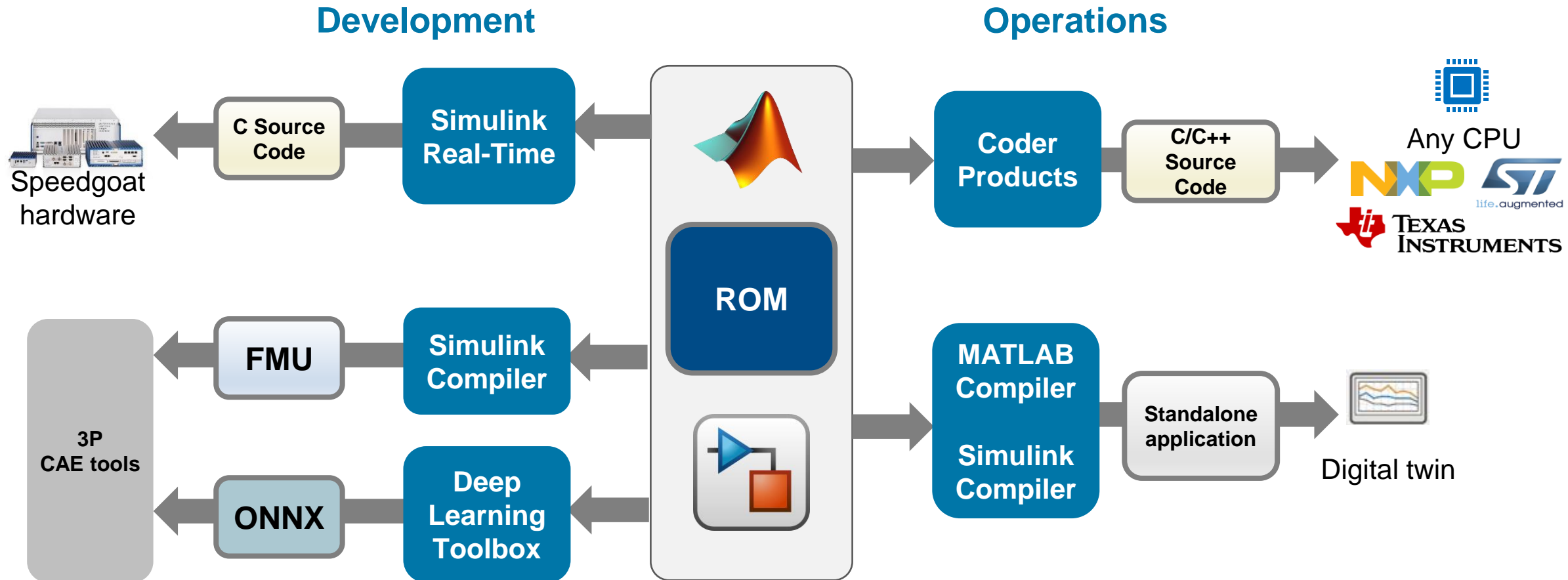
Deployment

Hardware-in-the-loop simulation

System-level integration and test



Use ROMs outside of Simulink, for development and operation stages



Manage AI tradeoffs for your system

	LSTM Long Short-Term Memory Network	Neural State Space (Neural ODE)
Training Speed	●	●
Interpretability	●	●
Inference Speed	●	●
Model Size	●	●
Accuracy (RSME)	●	●

Results are specific to Battery Electrochemical Model Example

Better



Okay



Worse



Reduced Order Modeling user stories



Subaru developed a **surrogate AI model to optimize transmission hydraulic systems**, achieving a **99% reduction in calculation times** compared to the original third-party 1D model

[Link to user story](#)



Cummins implemented a **deep learning neural network** to improve the **speed of engine cycle simulations for performance predictions**

[Link to user story](#)

MathWorks Service and Support Mechanisms

MathWorks has a team of over 700 **customer-facing engineers** – we welcome the opportunity to discuss how you can get the most out of your software investments and achieve your goals.



Technical Support

- Product questions
- General support
- 508-647-7000



AE (Application Engineering) Support

- Product/Capability demonstrations
- Workshops, Webinars, etc.
- Evaluation support



Extended AE Support

- Guided support for adoption of new tools/processes
- Deep Engagements
- Proof of Concept



Professional Courses

- Paid training on specific tools and/or processes
- On-site, web-based instructor lead, & self-paced online



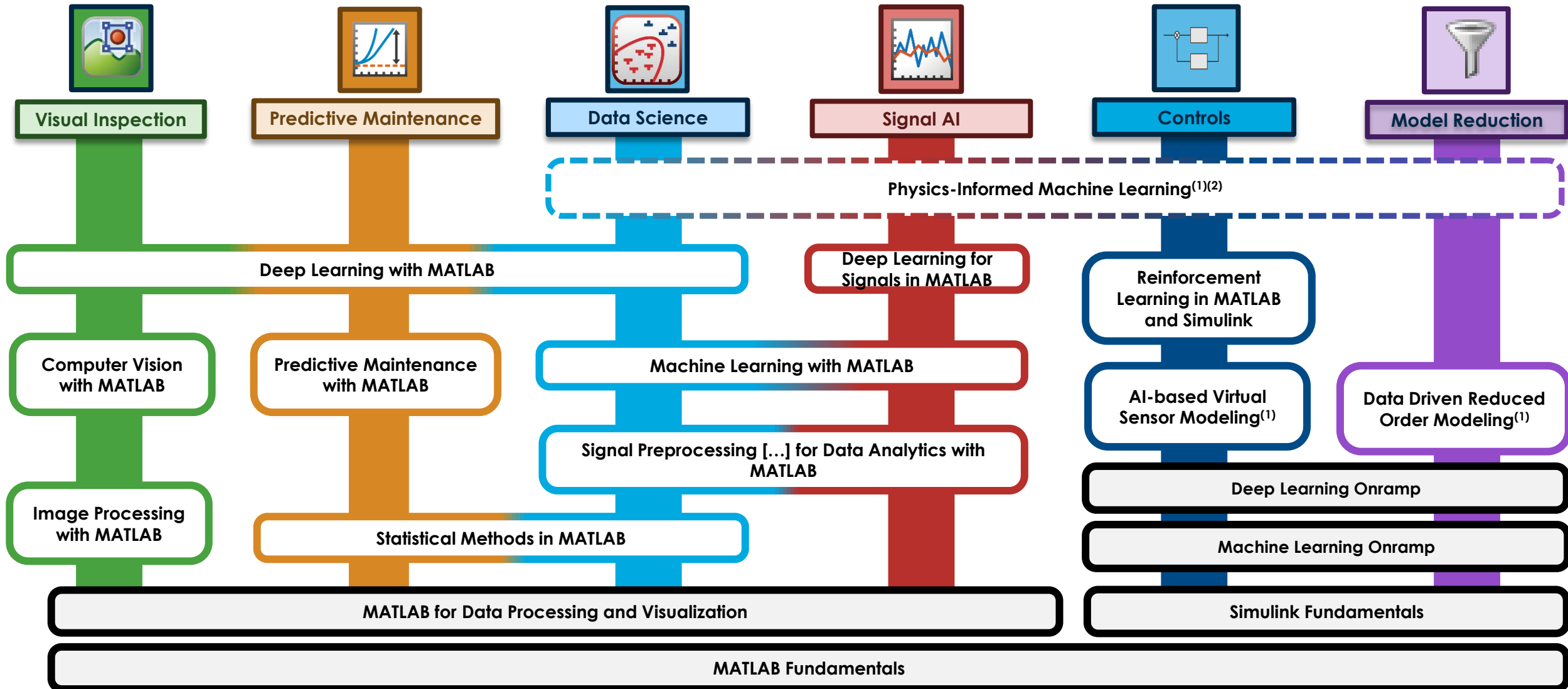
Consulting Engineering

- Paid engagements (custom targets, tool customization, advisory services)

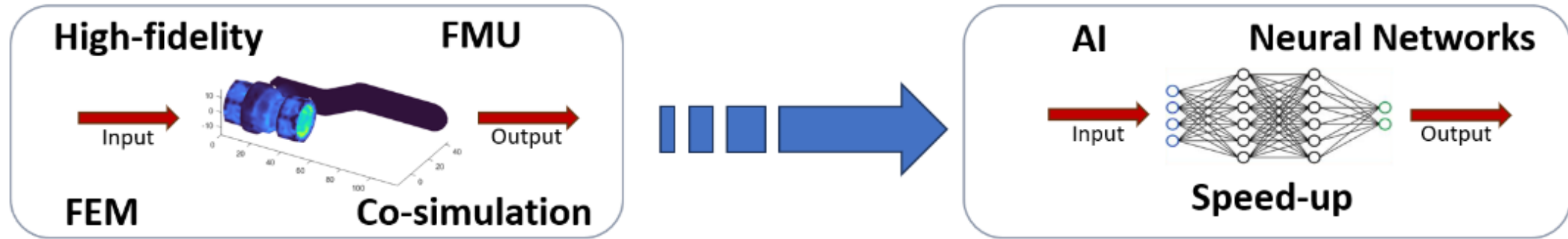
Complimentary

Funded

AI Training Pathways



2 Day Custom Training Available For Data Driven Reduced Order Modeling



- Generating data for reduced order modeling
- Data-driven AI-ROM models as surrogates for high-fidelity components
- Validation and simulation with reduced order models
- Recurrent neural networks, neural state space models, and system identification techniques

[Full Course Outline](#)

*Available upon request as private training only

Key takeaways

Enable

Reuse of full-order high-fidelity models for system-level simulations, Hardware-in-the-Loop (HIL) testing, nonlinear control design, and virtual sensor modeling

Explore

Various ROM techniques in MATLAB to find the best method.

- **Generate synthetic data** from Simulink
- **Train AI Models** to **replace high-fidelity battery electrochemical and PMSM model**
- **Integrate trained AI model into Simulink** for control design and system-level simulation
- **Generate C code and perform HIL tests**

