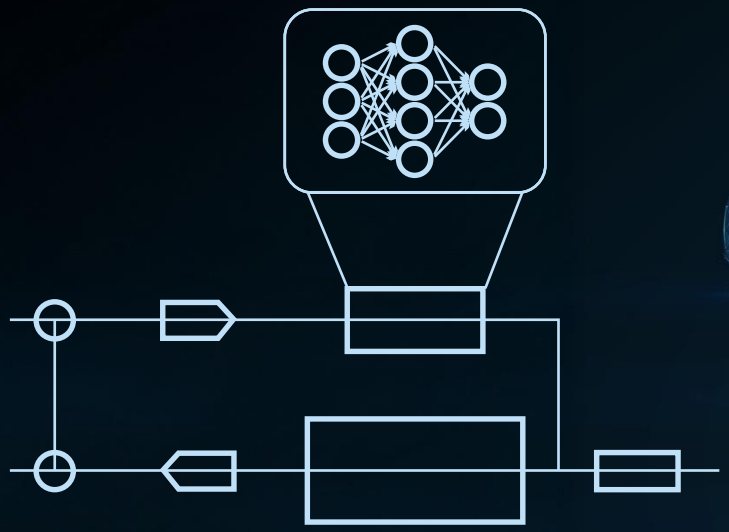


Reduced Order Models for Semiconductor Production Equipment

Mahaveer Satra

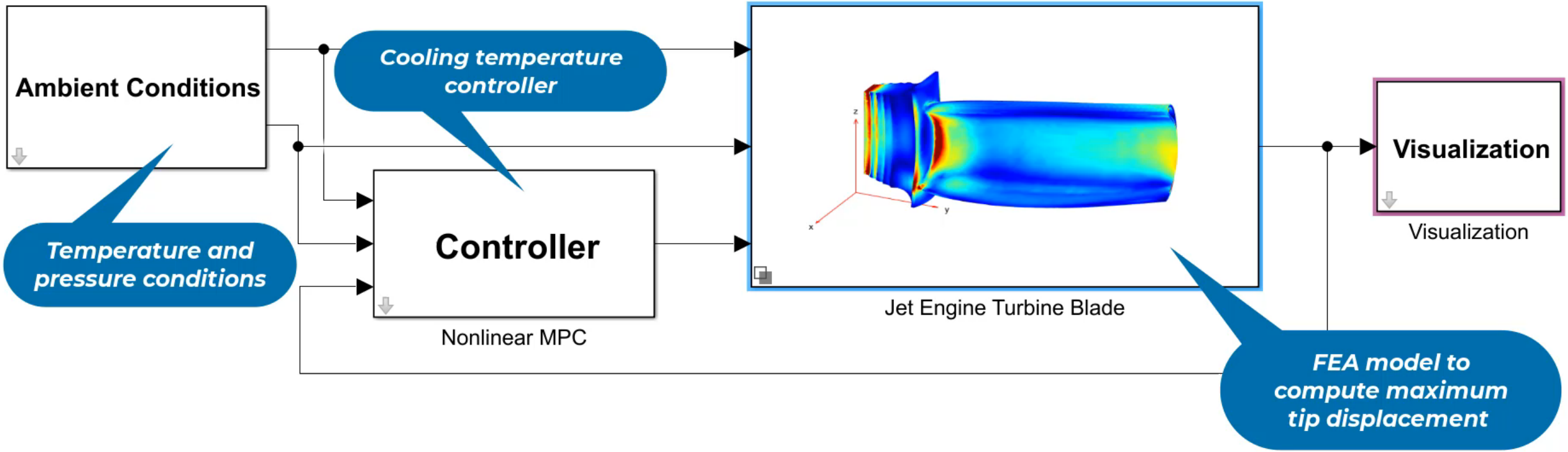
Industry Application Engineer,
MathWorks
msatra@mathworks.com

MathWorks®

SIMULATION DEBUG MODELING FORMAT APPS VARIANT SUBSYSTEM

New Open Save Print — FILE
Library Browser — LIBRARY
Log Signals Add Viewer Signal Table — PREPARE
Stop Time 7000 Normal Fast Restart
Step Back Run Step Forward Stop — SIMULATE
Data Inspector Logic Analyzer Simulation Manager — REVIEW RESULTS

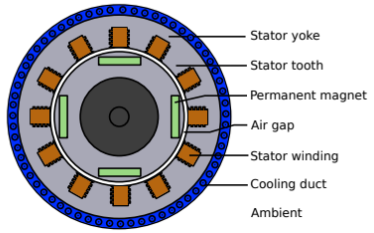
SystemLevelSim_JetEngineBlade | Reduced order model



Ready 169% auto(FixedStepDiscrete)

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

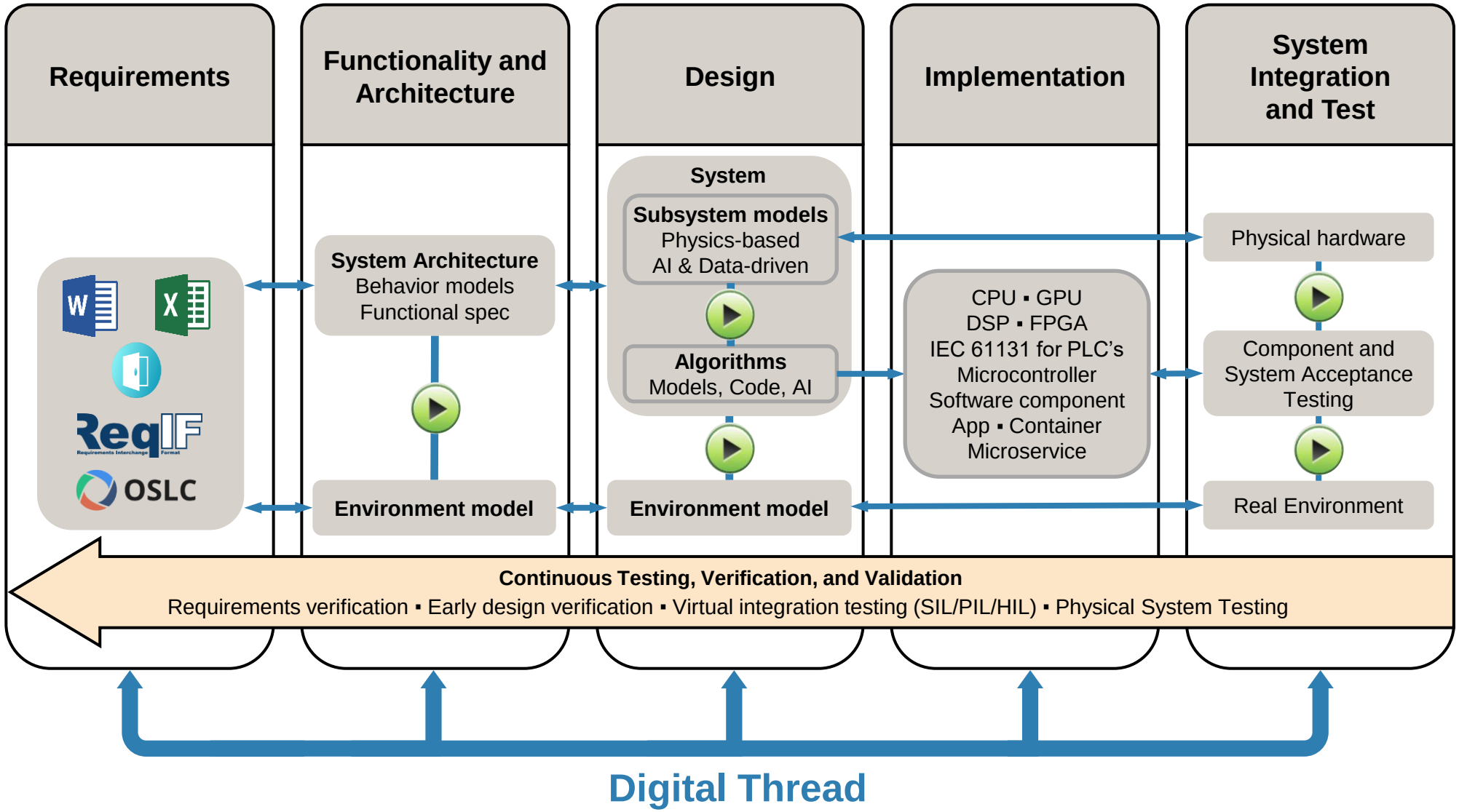
Enable	Reuse of full-order high-fidelity models for system-level simulations, Hardware-in-the-Loop (HIL) testing, nonlinear control design, and virtual sensor modeling.
Explore	Various ROM techniques in MATLAB to find the best method.

Common challenges

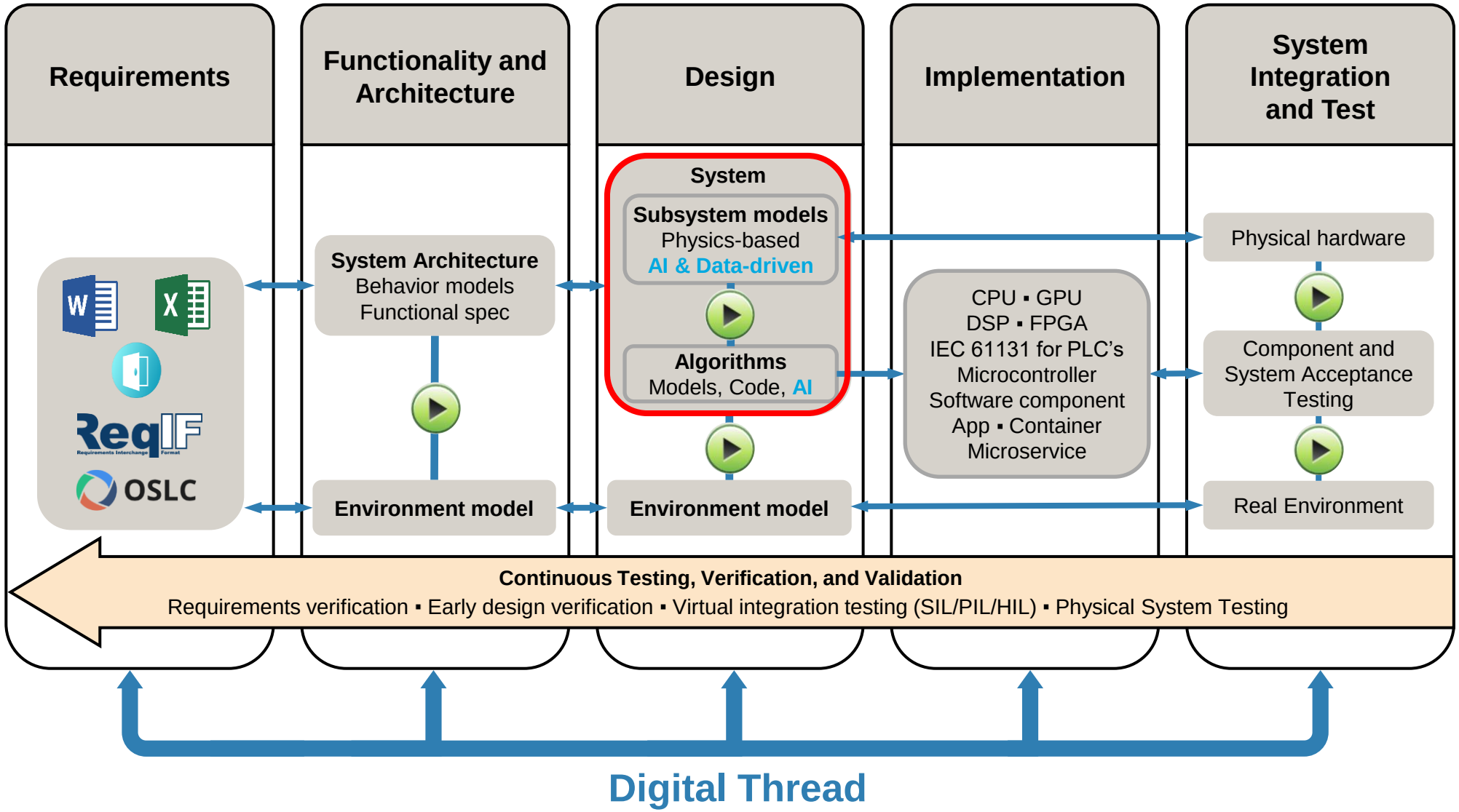
High fidelity models, such as ones from 3rd party FEA/CFD tools, are too slow for system level simulation, control design, and HIL testing.

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

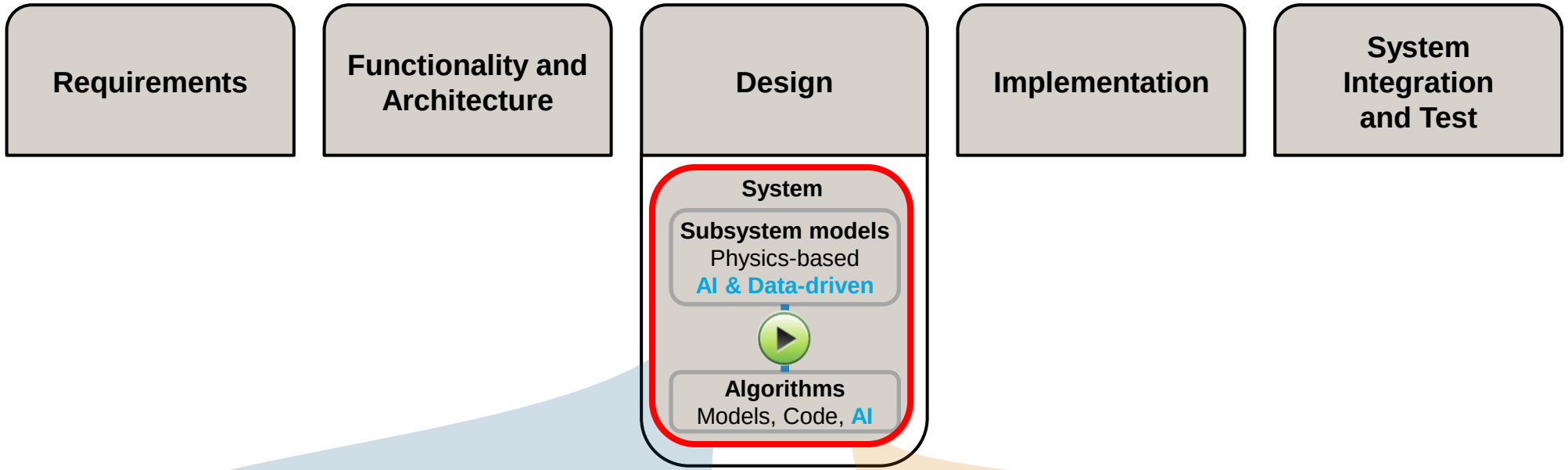
Model-Based Design



Integrating AI into Model-Based Design



AI for component modeling and algorithm development



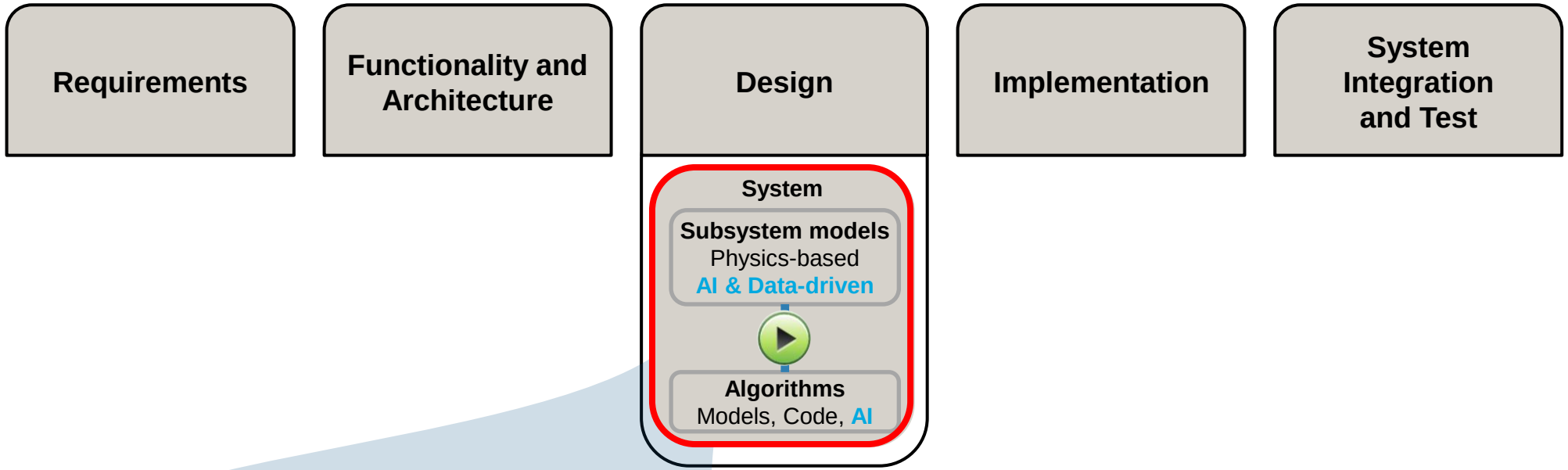
AI for component modeling

- System Identification
- Reduced Order Modeling (ROM)

AI for algorithm development

- Perception and Sensing **Virtual Sensors**
- Planning and Controls

Focus today



AI for component modeling

- System Identification
- Reduced Order Modeling (ROM)

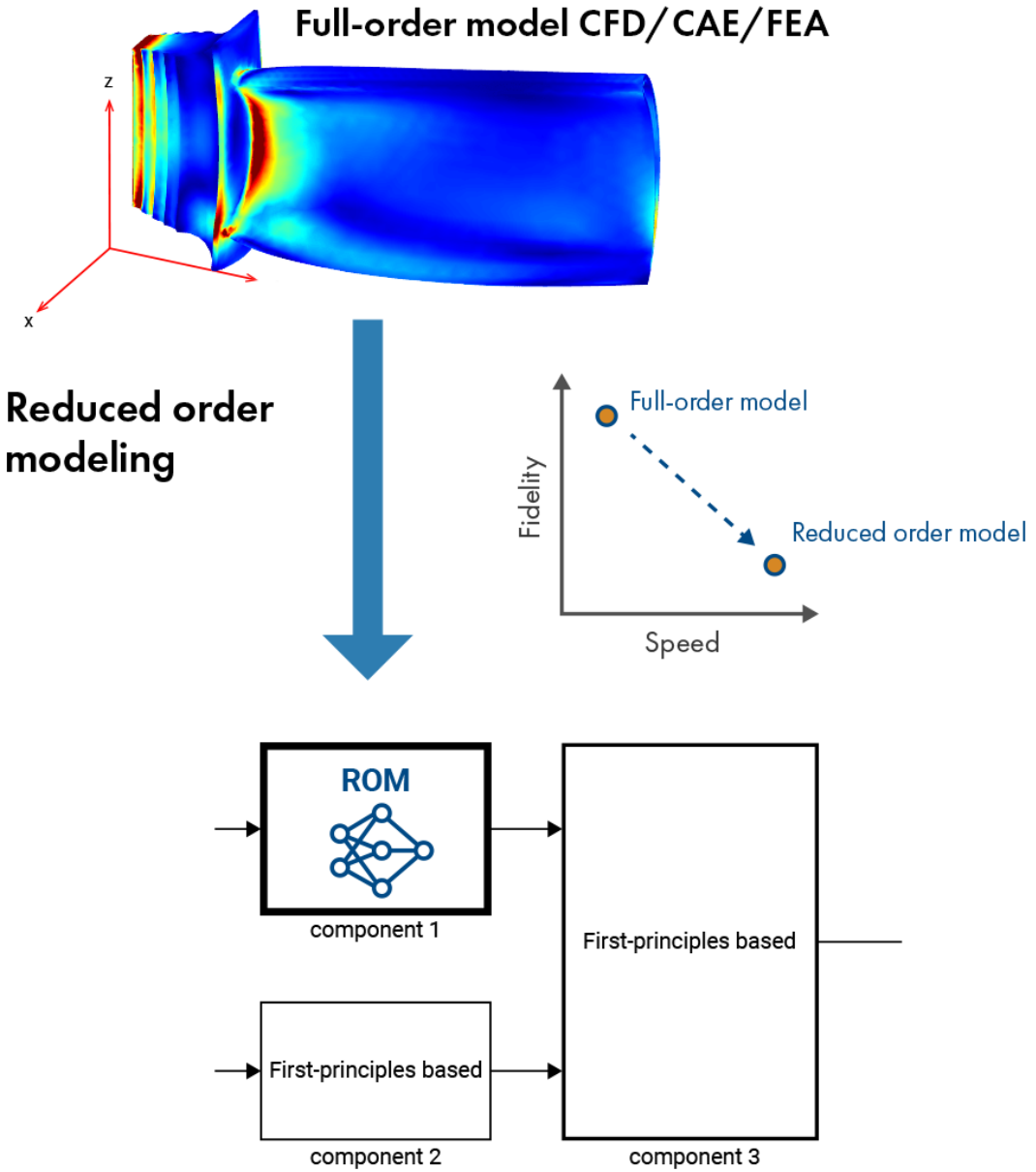
Reduced order modeling

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



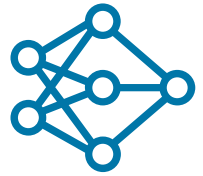
Reduced order modeling

How

Reduced order model

AI-Based Data-driven

Inputs
Charging Rate
SoC

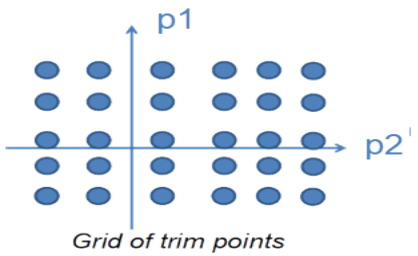


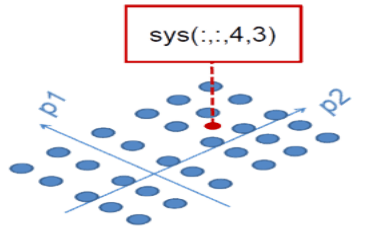
Outputs
Potential Difference ***(V_s – V_e(V))***

AI model

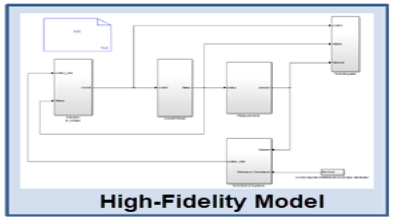
focus today

Linearization





Loop through the grid of trim points



Identify local model at each trim point

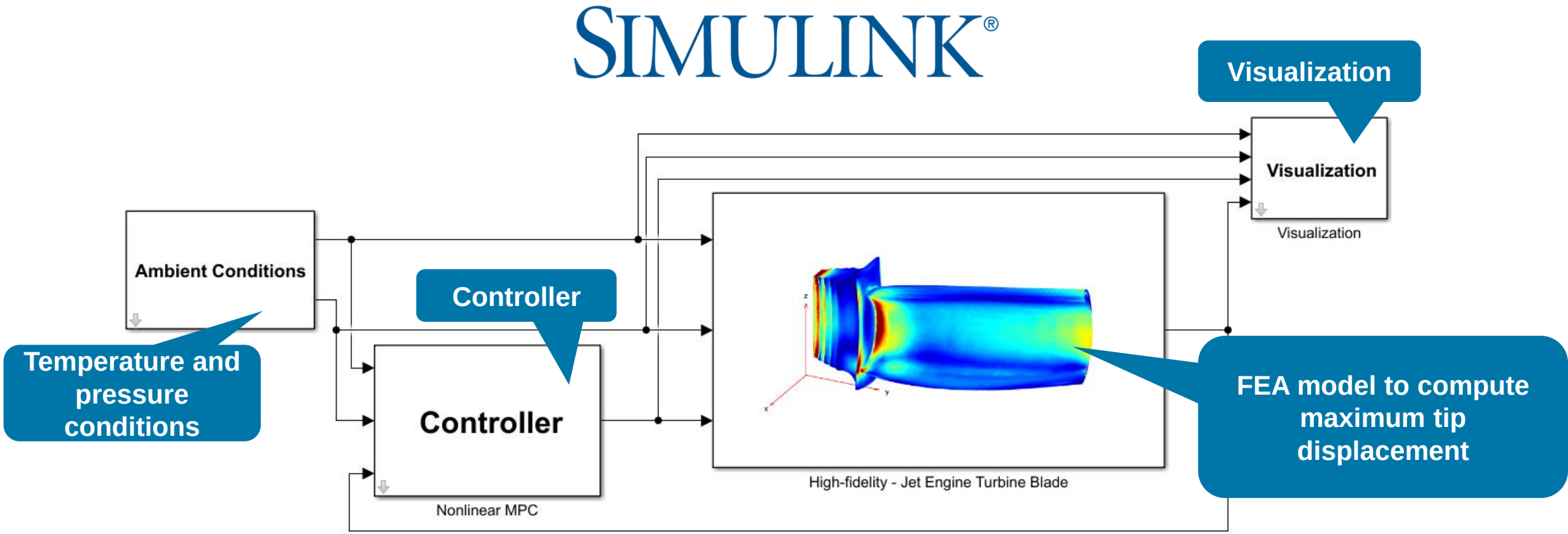
Model-based

FEA Software

Simulink
Simscape Multibody
Control System Toolbox

Example overview

Replacing a high-fidelity jet engine turbine blade model with an AI-based reduced order model

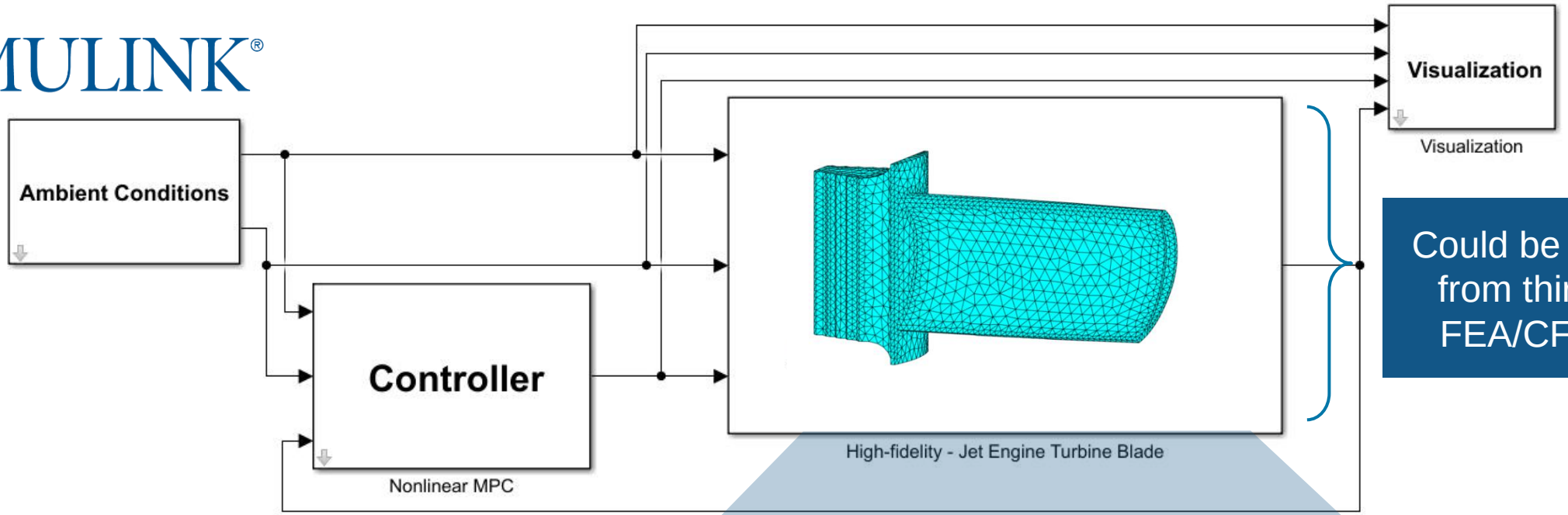


Closed-loop temperature control

Example overview

Replacing a high-fidelity jet engine turbine blade model with an AI-based reduced order model

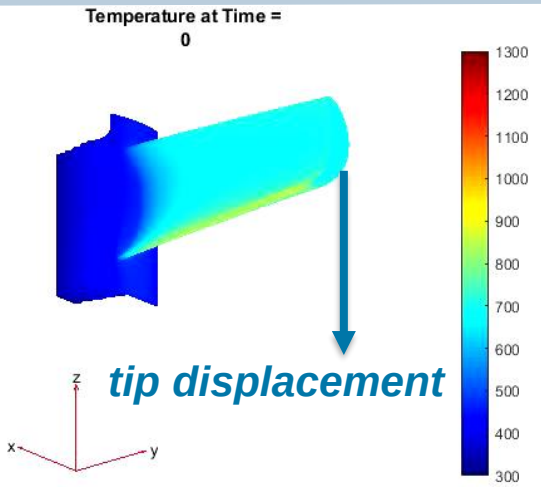
SIMULINK®



Could be imported from third-party FEA/CFD tools

~30 seconds per time step for solving FEA models

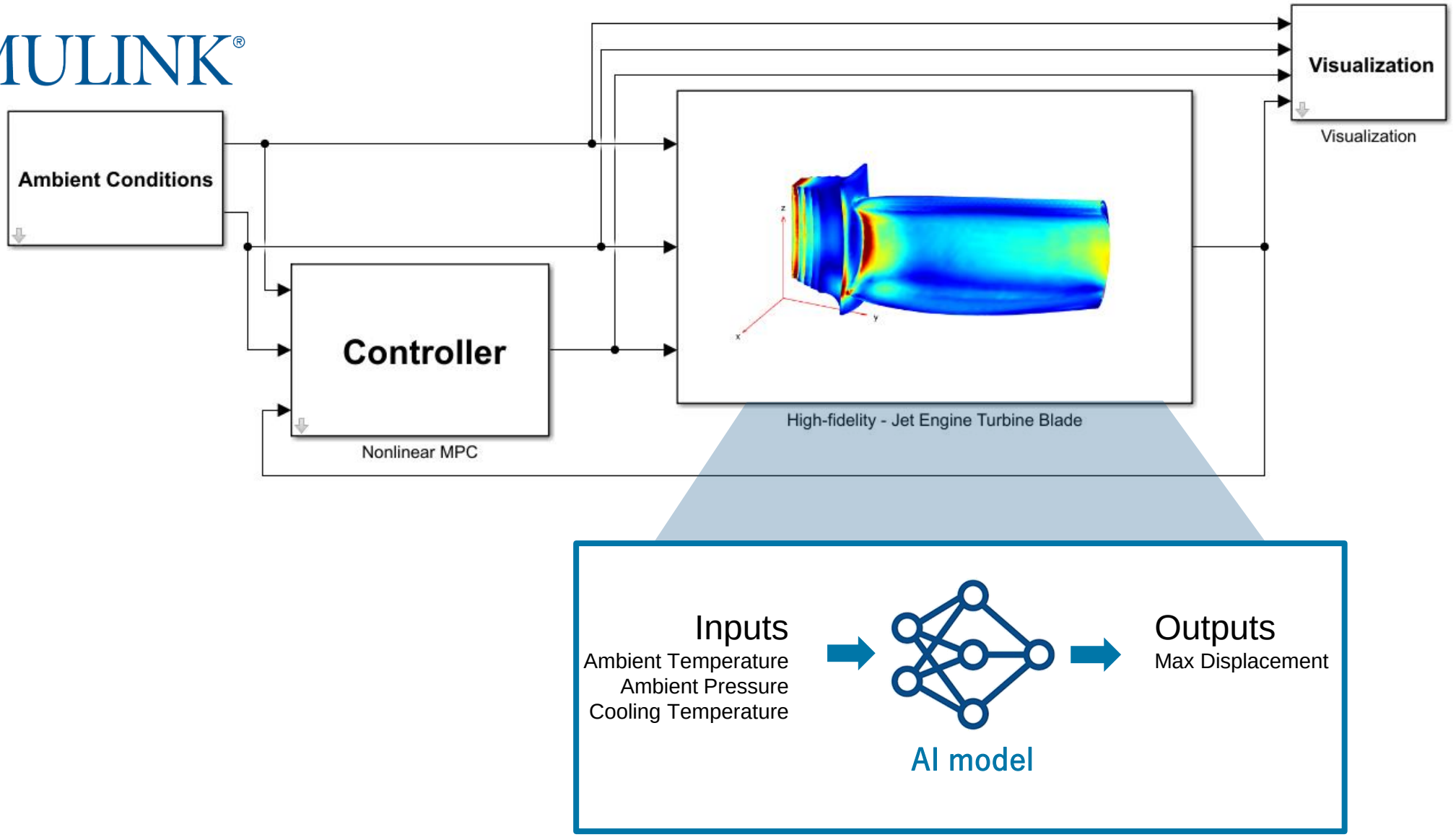
Not suitable for control design and HIL testing



Example overview

Replacing a high-fidelity jet engine turbine blade model with an AI-based reduced order model

SIMULINK®



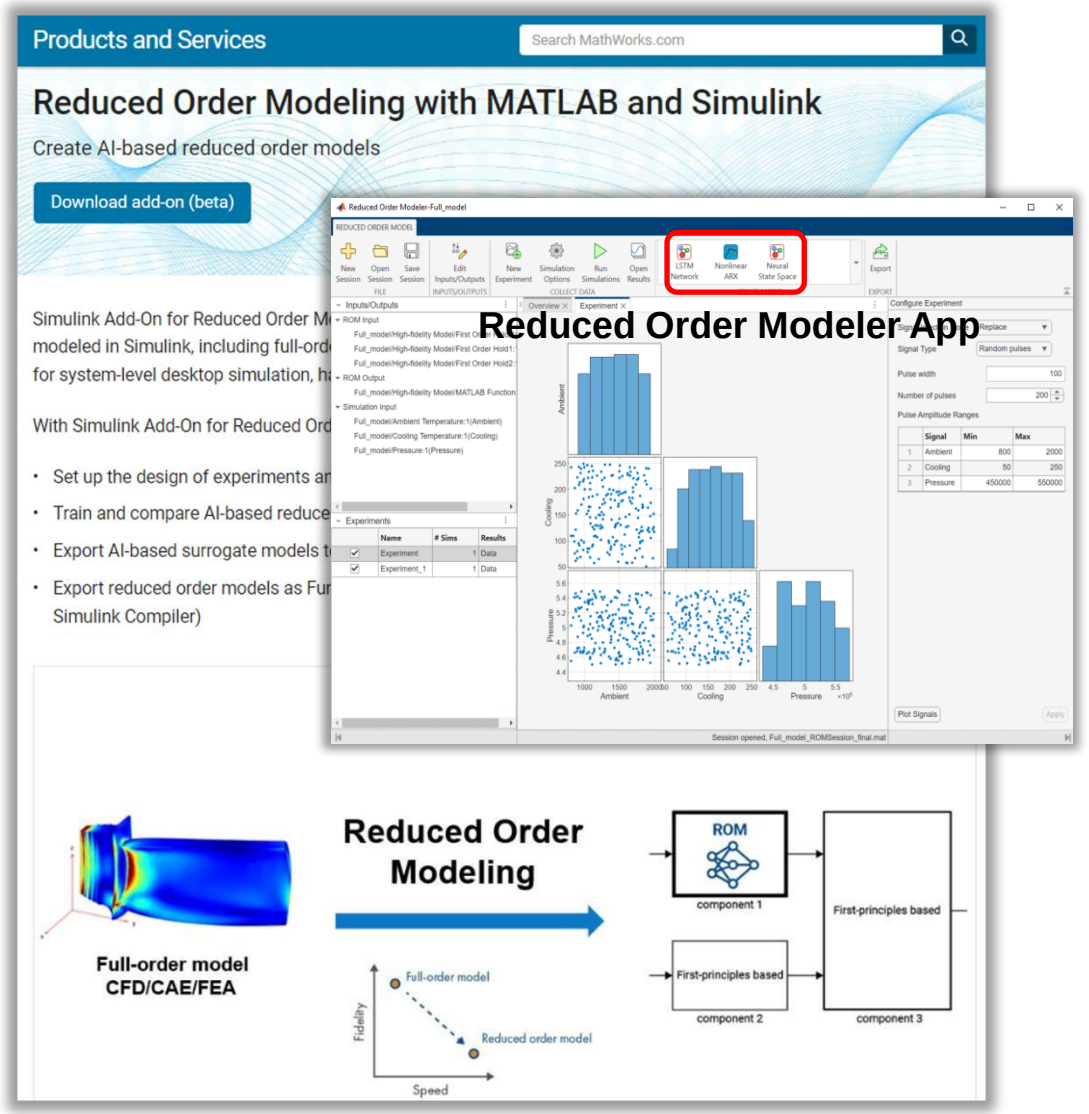
Create AI-based ROMs using the reduced order modeling support package

Set up Design of Experiments (DoE)

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

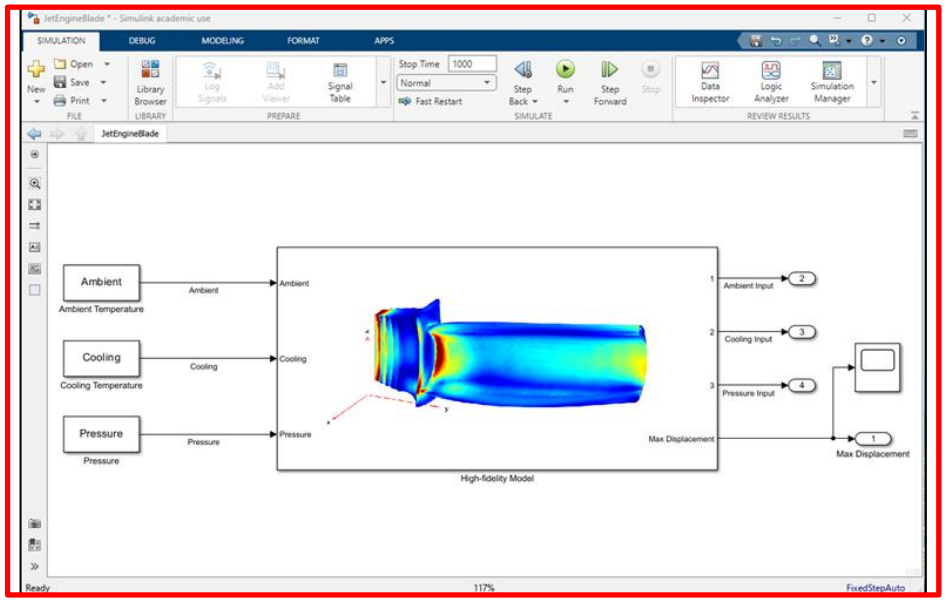
Train and compare AI-based reduced order models using preconfigured templates

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

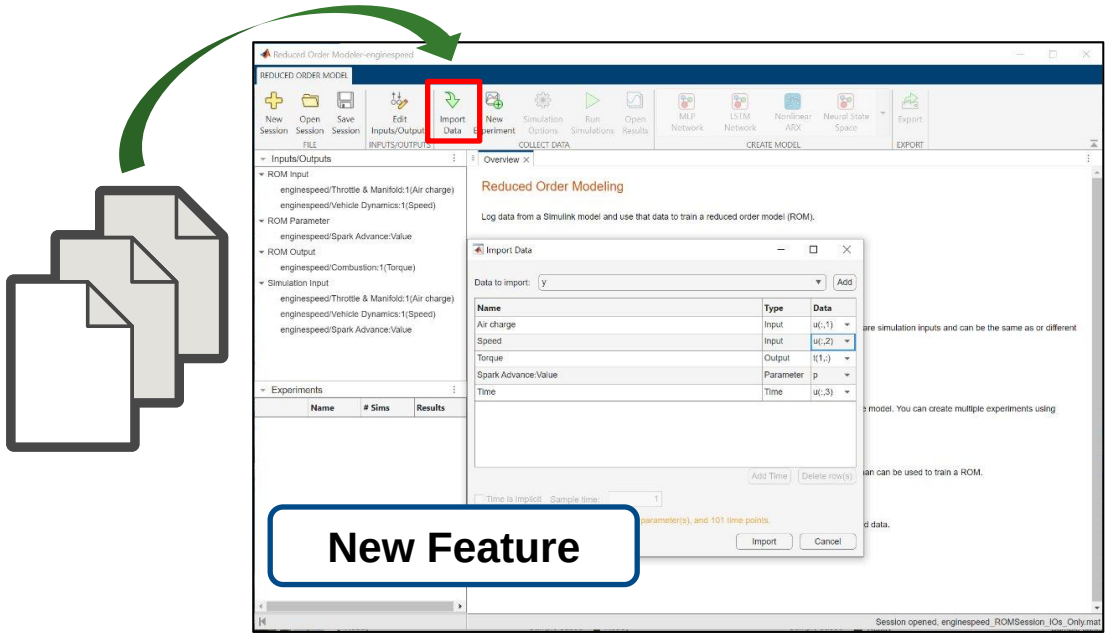


[Reduced Order Modeling with MATLAB and Simulink]

Prepare data for training AI models



Generate synthetic data from Simulink/Simscape models

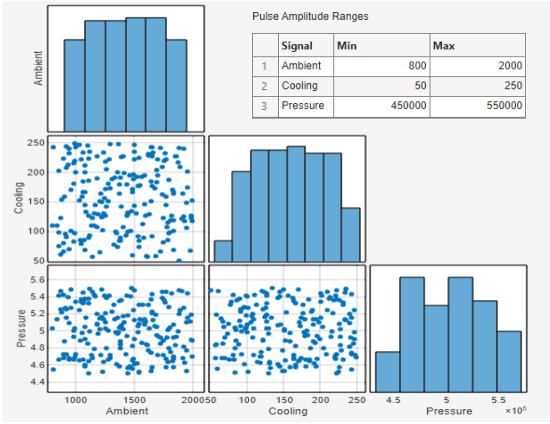


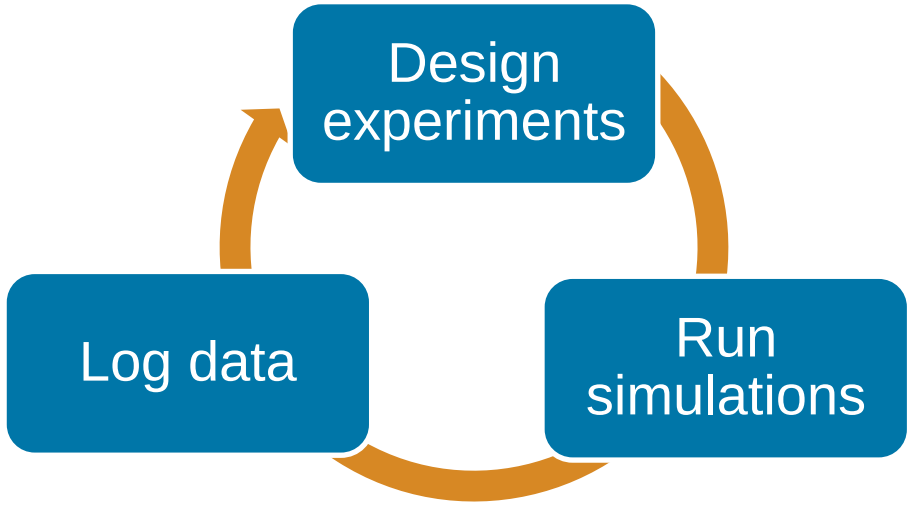
Import pre-collected data from high-fidelity model into the app

Data Preparation | AI Modeling | Simulation & Test | Deployment

Synthetic Data Generation

Design of Experiments



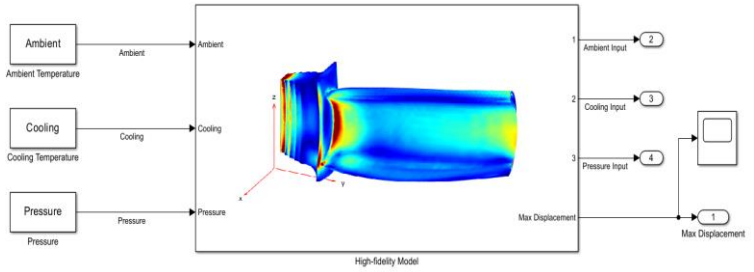


Input features

Ambient Temperature
Ambient Pressure
Cooling Temperature

Response

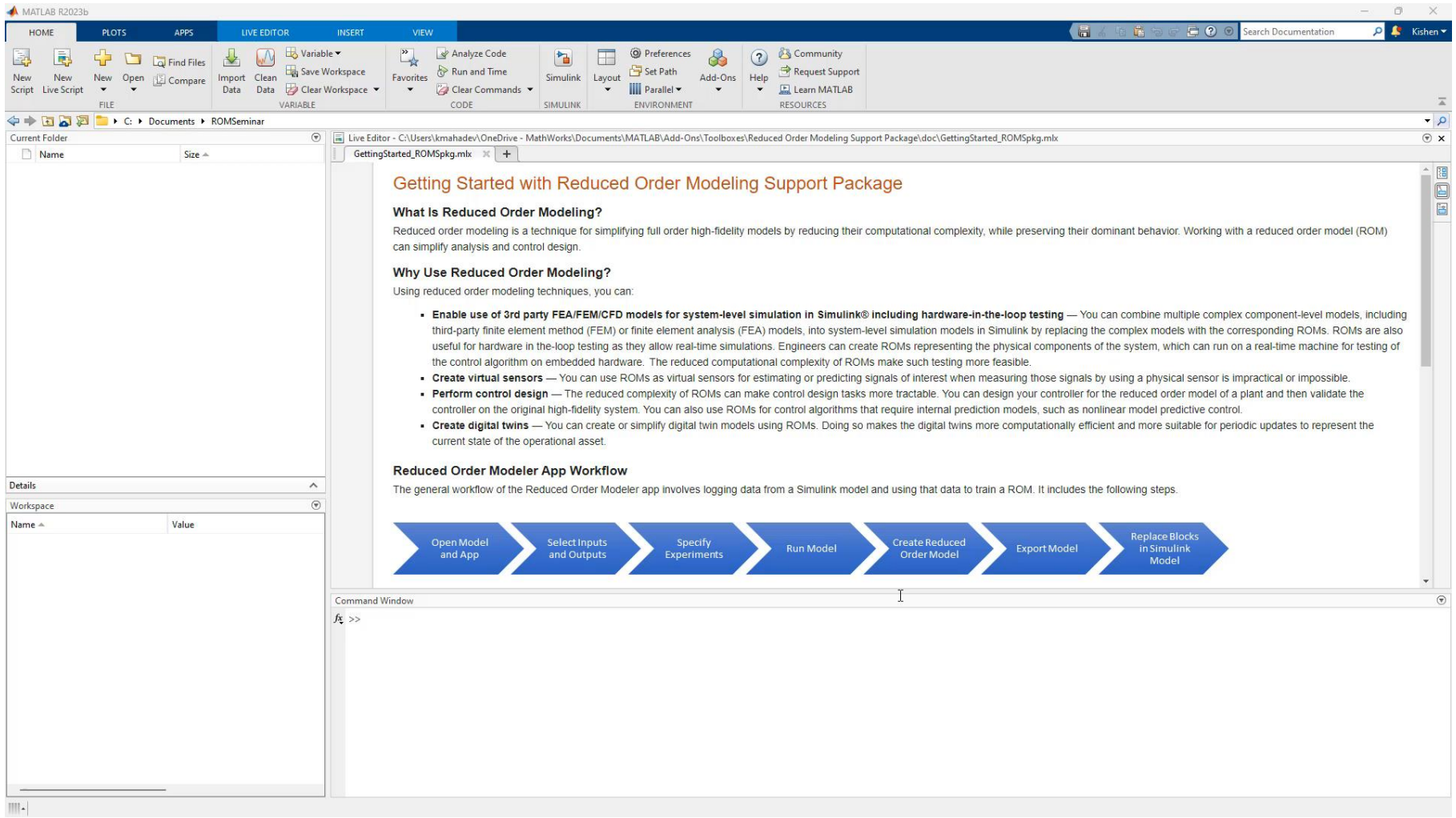
Max Displacement



Data Preparation | AI Modeling | Simulation & Test | Deployment

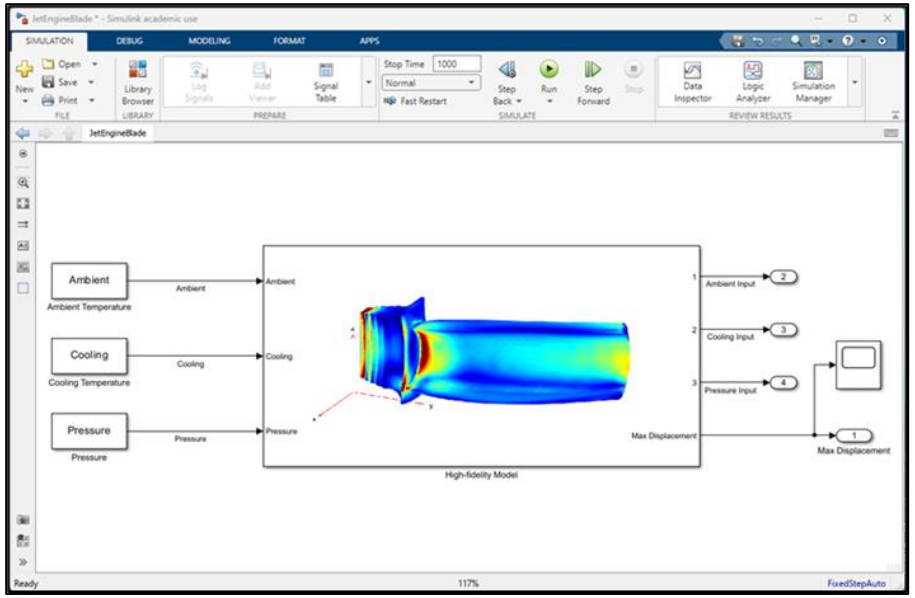
Synthetic Data Generation

Design of Experiments

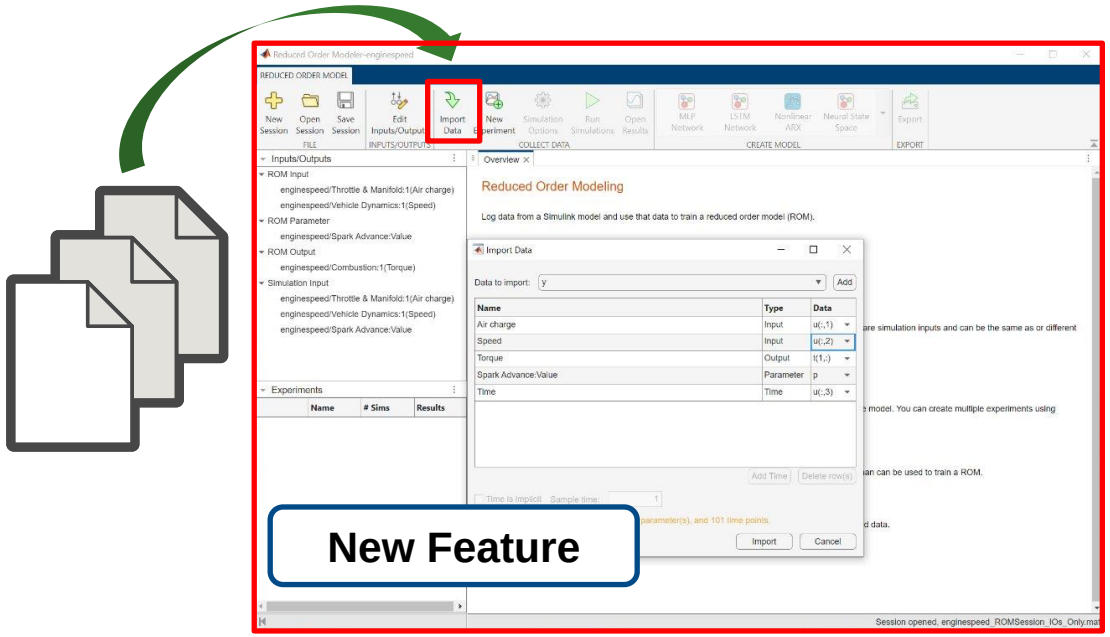


Data Preparation | AI Modeling | Simulation & Test | Deployment

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 Preparation | AI Modeling | Simulation & Test | Deployment

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

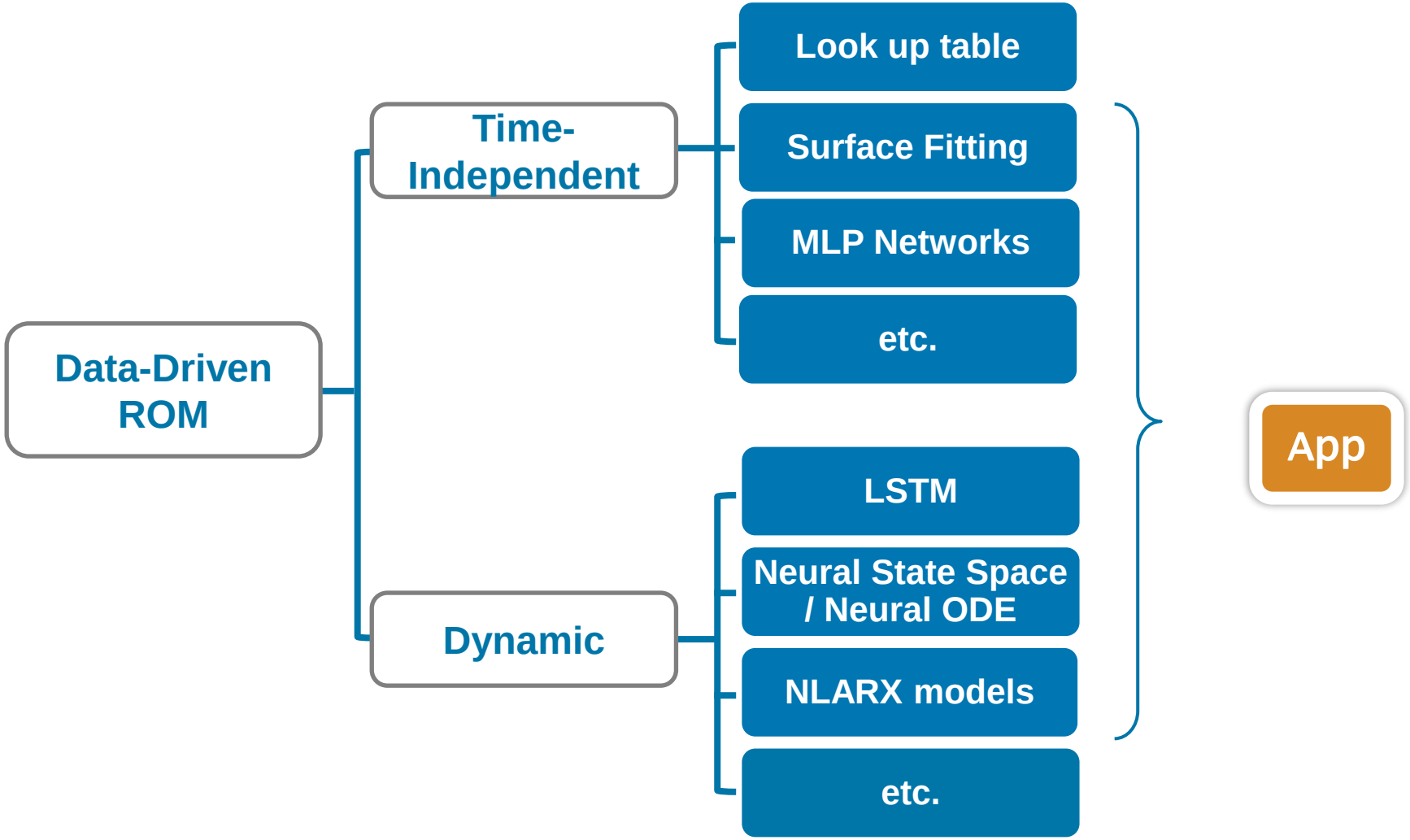
Oliver Wallscheid
Department of Power Electronics and Electrical Drives
Paderborn University
33095 Paderborn, Germany
wallscheid@lea.uni-paderborn.de

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



Data Preparation | AI Modeling | Simulation & Test | Deployment

Reduced Order Modeler-JetEngineBlade

REDUCED ORDER MODEL | SIMULATION RESULT

New Session | Open Session | Save Session — FILE
Edit Inputs/Outputs — INPUTS/OUTPUTS
New Experiment | Simulation Options | Run Simulations | Open Results — COLLECT DATA
LSTM Network | Nonlinear ARX | Neural State Space — CREATE MODEL
Export — EXPORT

Inputs/Outputs

- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure_input)
- ROM Output
  - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)
- Simulation Input
  - JetEngineBlade/Ambient Temperature:1(Ambient)
  - JetEngineBlade/Cooling Temperature:1(Cooling)
  - JetEngineBlade/Pressure:1(Pressure)

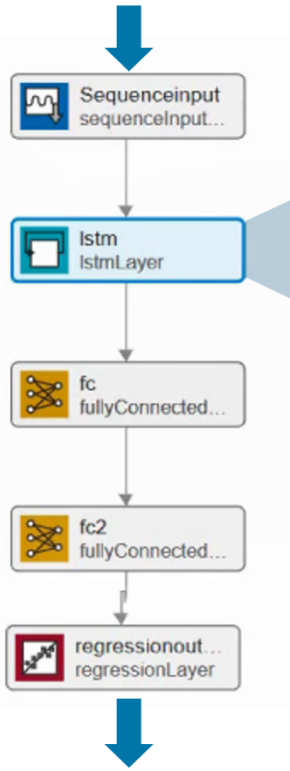
Experiments

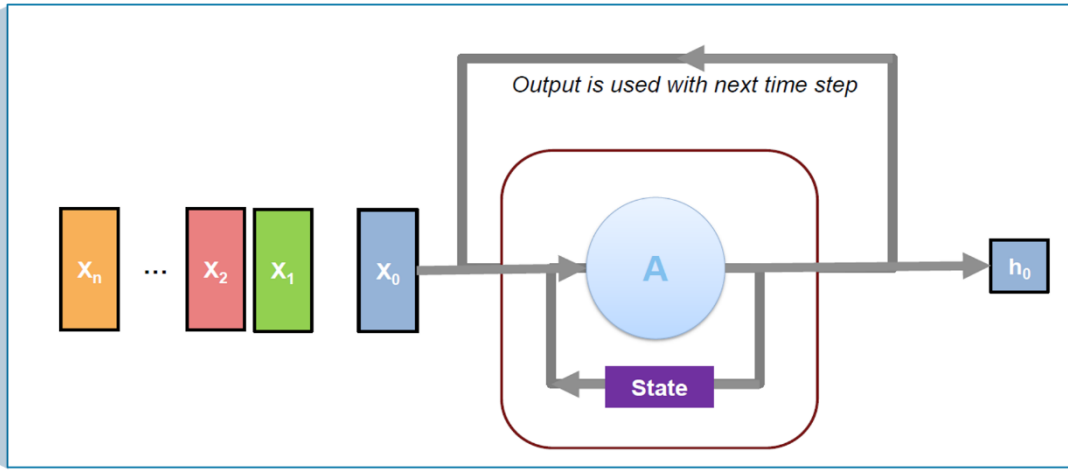
	Name	# Sims	Results
☑	Experiment	1	Data
☑	Experiment_1	1	Data

Result: Experiment_1

Inputs

- Ambient Temperature
- Cooling Temperature
- Pressure





Outputs

- Max Displacement

AI-based ROM using LSTMs

Capture time dependencies in time-series data

Data Preparation | AI Modeling | Simulation & Test | Deployment

Session opened, engineBlade_ROMsession_results.mat

Reduced Order Modeler-JetEngineBlade

REDUCED ORDER MODEL | SIMULATION RESULT

Simulation Result 1 of 1

Show output only

Show as scatter plot

SIMULATION RESULTS | OPTIONS

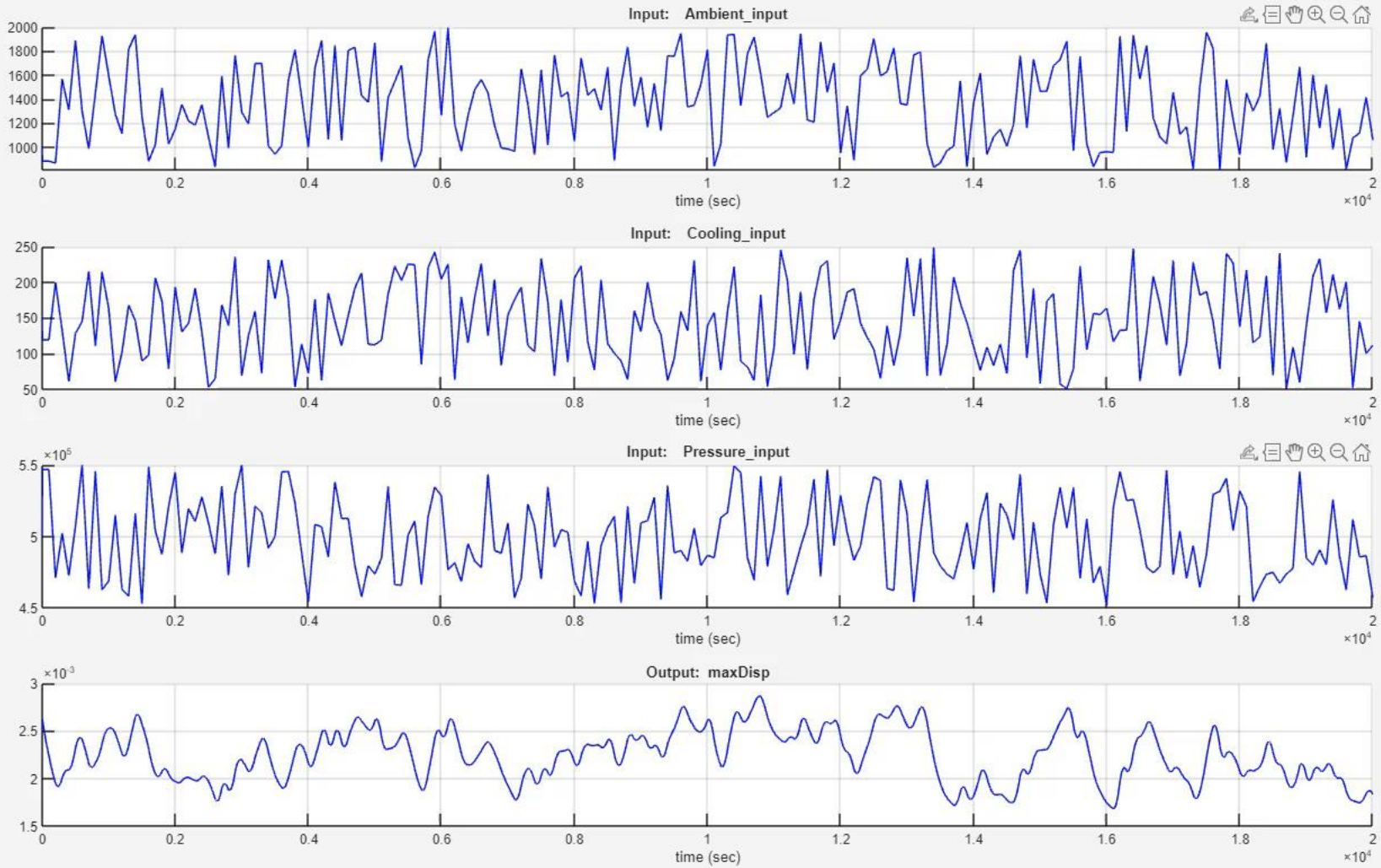
Inputs/Outputs

- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure_input)
- ROM Output
  - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)
- Simulation Input
  - JetEngineBlade/Ambient Temperature:1(Ambient)
  - JetEngineBlade/Cooling Temperature:1(Cooling)
  - JetEngineBlade/Pressure:1(Pressure)

Experiments

	Name	# Sims	Results
☑	Experiment	1	Data
☑	Experiment_1	1	Data

Overview | Result: Experiment | Result: Experiment_1 | Experiment_1



Session opened, engineBlade_ROMsession_results.mat

Reduced Order Modeler-JetEngineBlade

REDUCED ORDER MODEL | SIMULATION RESULT

New Session | Open Session | Save Session — FILE
Edit Inputs/Outputs — INPUTS/OUTPUTS
New Experiment | Simulation Options | Run Simulations | Open Results — COLLECT DATA
LSTM Network | Nonlinear ARX | Neural State Space
Export — EXPORT

Inputs/Outputs

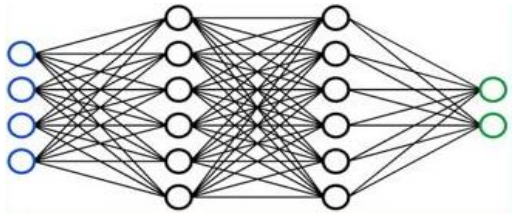
- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure_input)
- ROM Output
  - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)
- Simulation Input
  - JetEngineBlade/Ambient Temperature:1(Ambient)
  - JetEngineBlade/Cooling Temperature:1(Cooling)
  - JetEngineBlade/Pressure:1(Pressure)

Experiments

	Name	# Sims	Results
☑	Experiment	1	Data
☑	Experiment_1	1	Data

Overview | Result: Expe | nt_1

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



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

Data Preparation | AI Modeling | Simulation & Test | Deployment

Session opened, engineBlade_ROMsession_results.mat

Reduced Order Modeler-JetEngineBlade

REDUCED ORDER MODEL | SIMULATION RESULT

New Session | Open Session | Save Session — FILE
Edit Inputs/Outputs — INPUTS/OUTPUTS
New Experiment | Simulation Options | Run Simulations | Open Results — COLLECT DATA
LSTM Network | Nonlinear ARX | Neural State Space — CREATE MODEL
Export — EXPORT

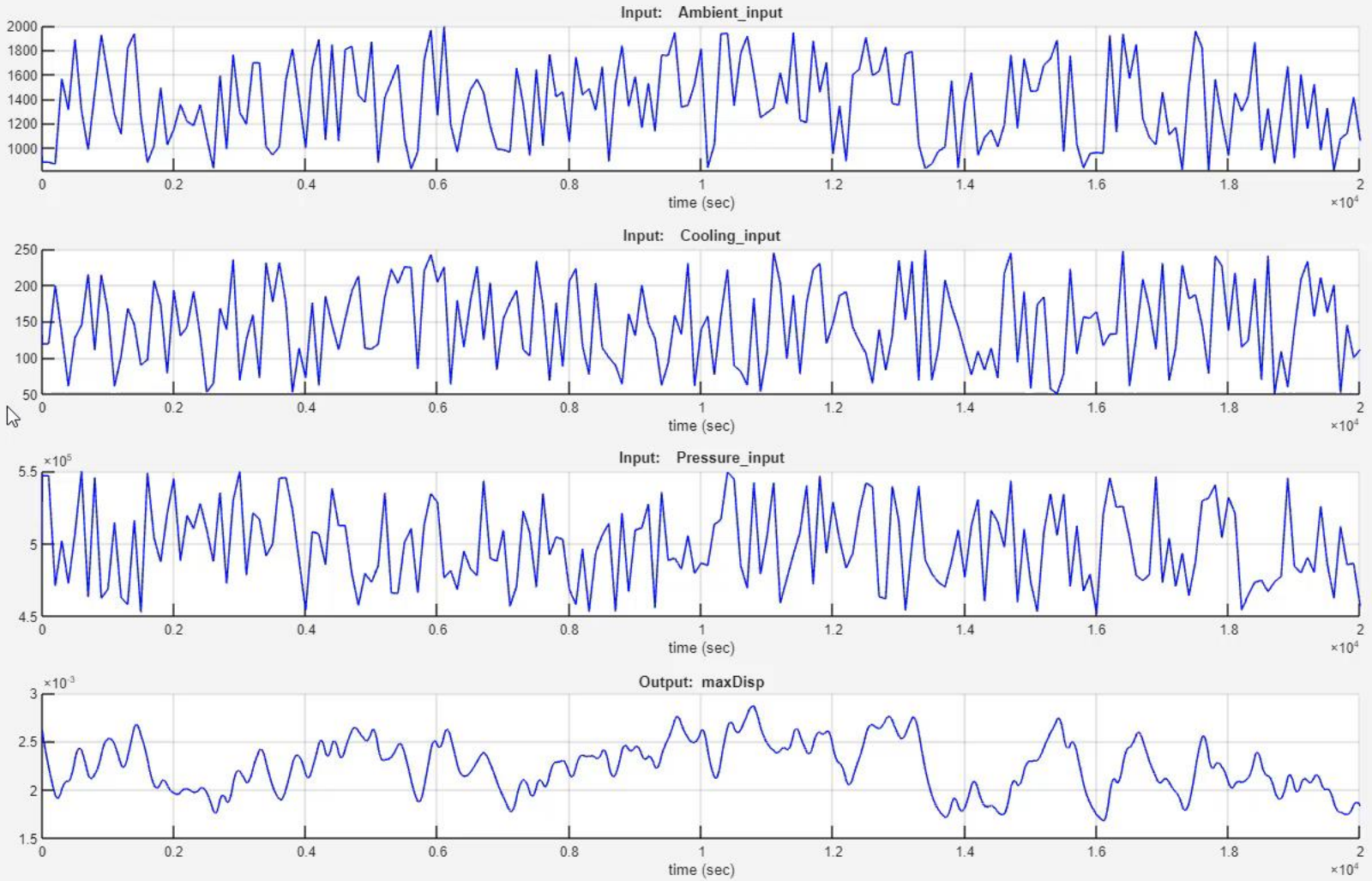
Inputs/Outputs

- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure_input)
- ROM Output
  - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)
- Simulation Input
  - JetEngineBlade/Ambient Temperature:1(Ambient)
  - JetEngineBlade/Cooling Temperature:1(Cooling)
  - JetEngineBlade/Pressure:1(Pressure)

Experiments

	Name	# Sims	Results
☑	Experiment	1	Data
☑	Experiment_1	1	Data

Overview | Result: Experiment | Result: Experiment_1 | Experiment_1



Session opened, engineBlade_ROMsession_results.mat

Reduced Order Modeler-JetEngineBlade

REDUCED ORDER MODEL | SIMULATION RESULT

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

FILE | INPUTS/OUTPUTS | COLLECT DATA | EXPORT

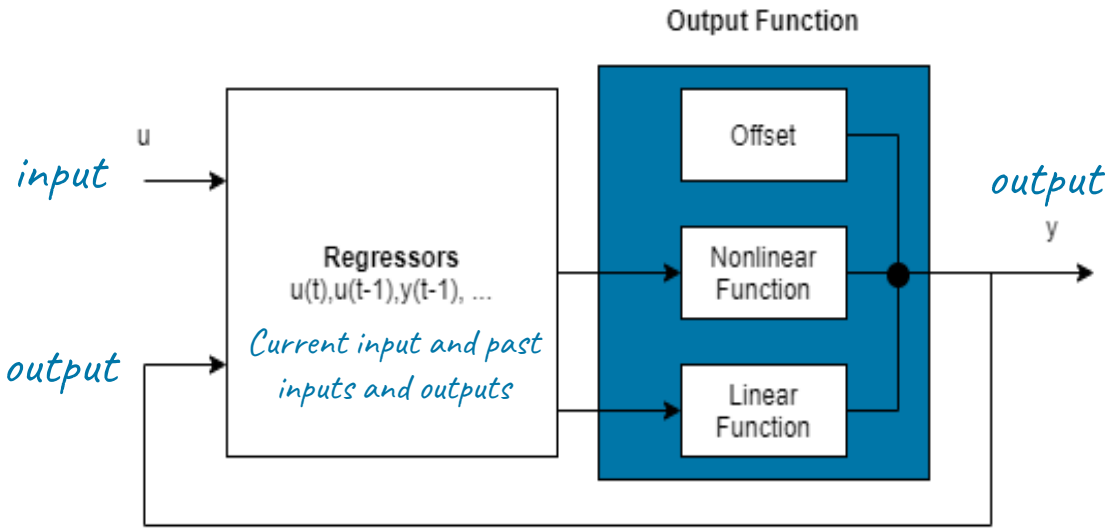
Inputs/Outputs

- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure_input)
- ROM Output
  - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)
- Simulation Input
  - JetEngineBlade/Ambient Temperature:1(Ambient)
  - JetEngineBlade/Cooling Temperature:1(Cooling)
  - JetEngineBlade/Pressure:1(Pressure)

Experiments

	Name	# Sims	Results
☑	Experiment	1	Data
☑	Experiment_1	1	Data

Overview | Result: Experiment_1



AI-based ROM using Nonlinear ARX

Extend linear models and model nonlinear behavior using flexible nonlinear functions

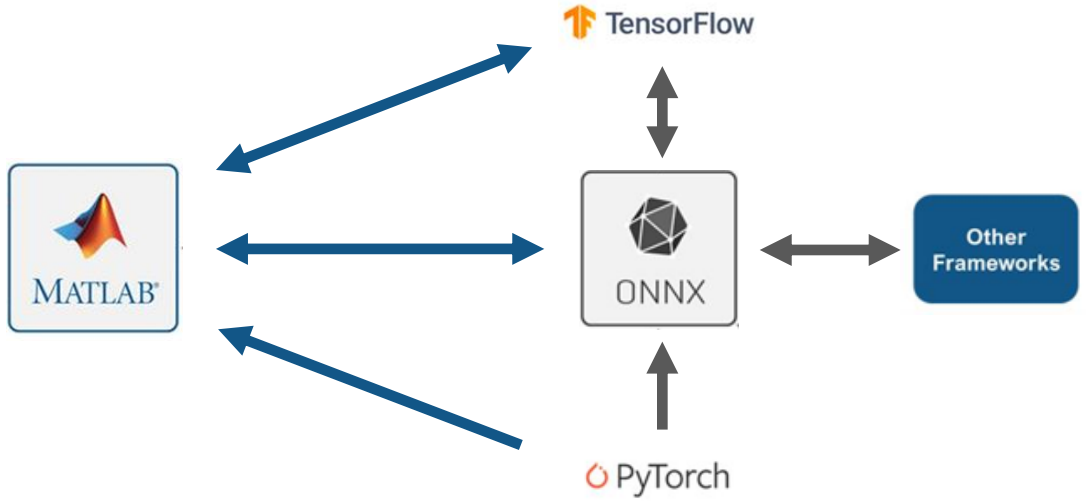
Data Preparation | **AI Modeling** | Simulation & Test | Deployment

Session opened, engineBlade_ROMsession_results.mat

MATLAB interoperates with other frameworks

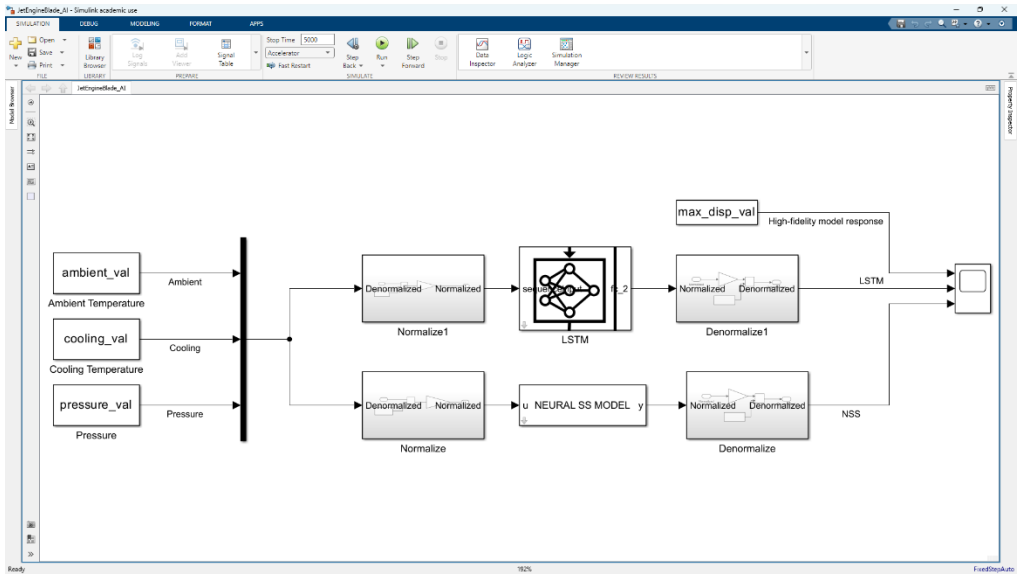
Framework interoperability bridges the gap between data science, engineering and production

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

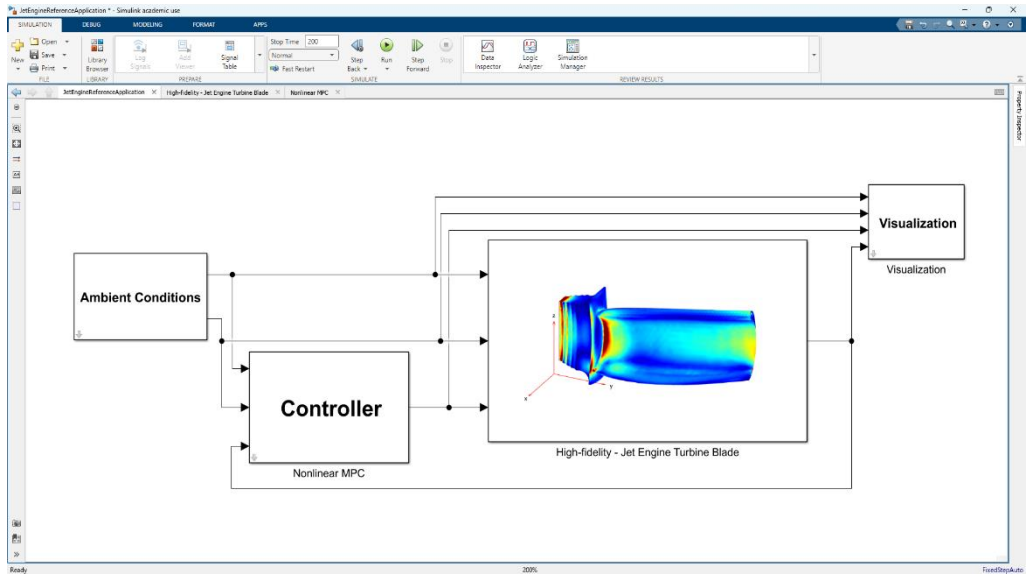


Integrate your AI model for system-level simulation and test

Integration of trained AI model into Simulink



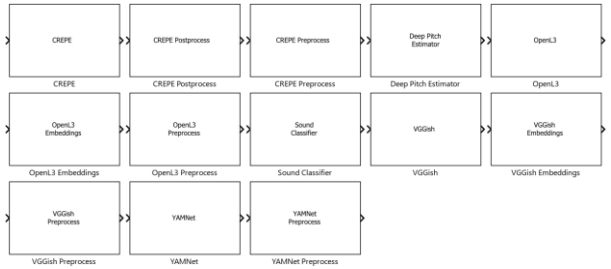
System-level simulation



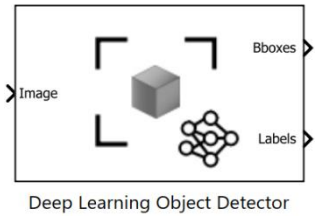
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

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

Specialized

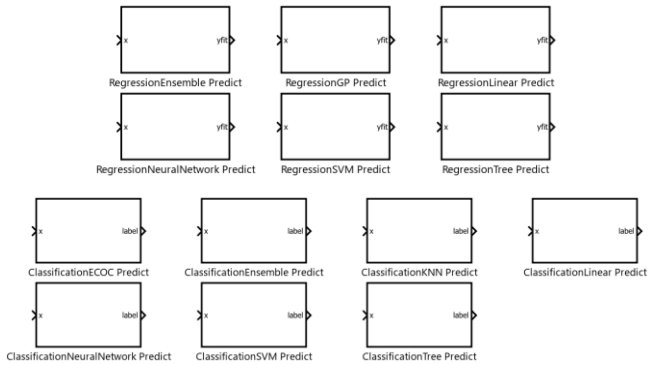


Audio Toolbox

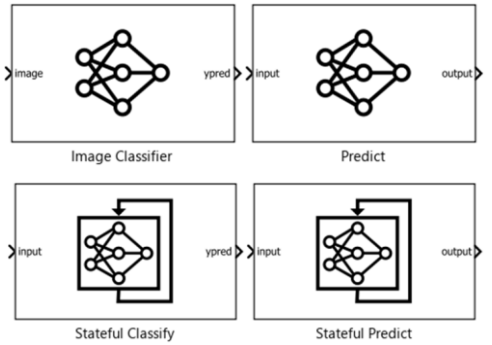


Computer Vision Toolbox

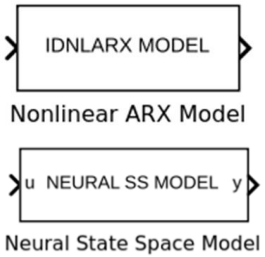
AI Core



Statistics and Machine Learning Toolbox



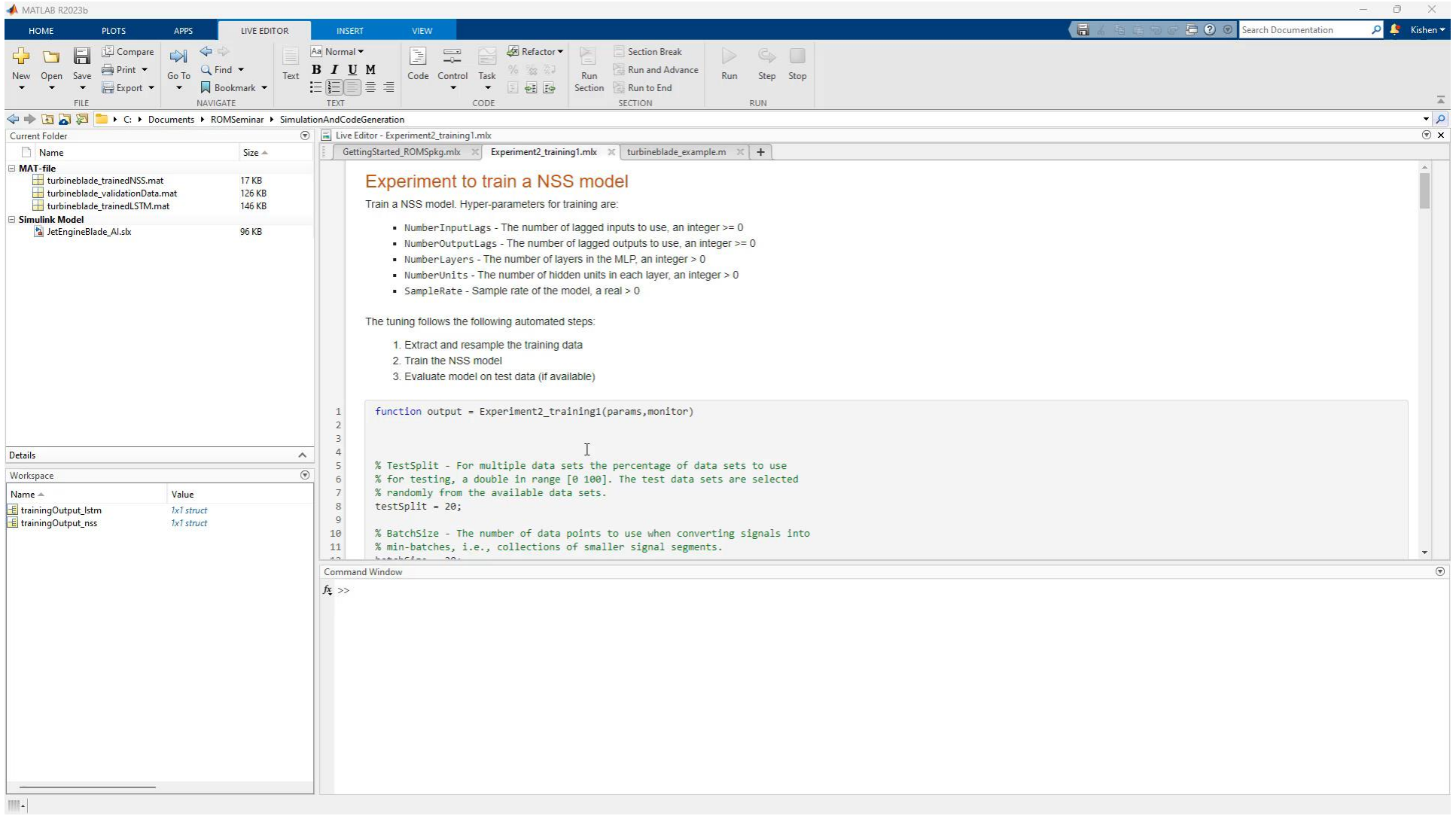
Deep Learning Toolbox



System Identification 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

Simulink Profiler

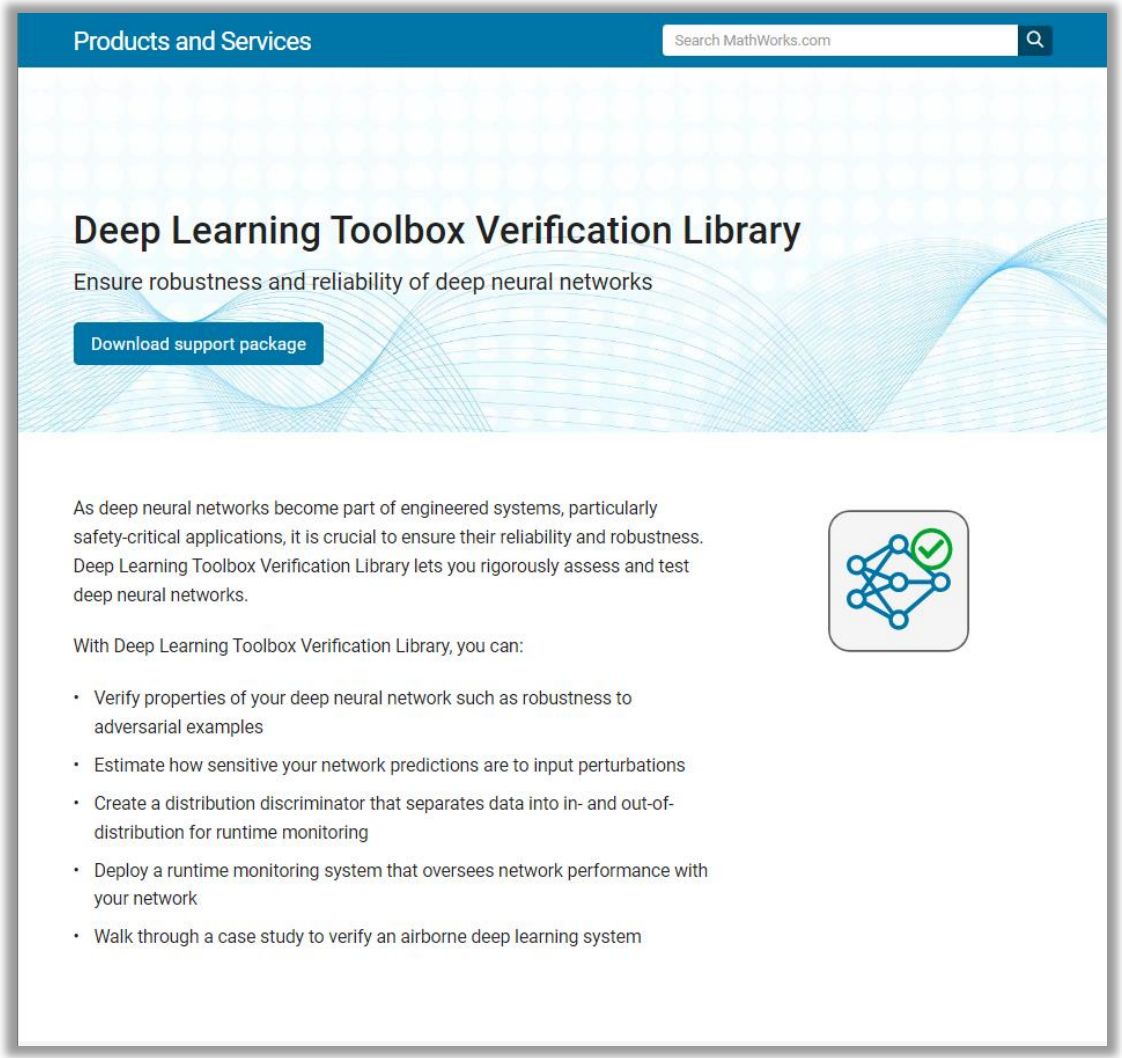
Path	Time Plot (Dark Band = Self Time)	Total Time (s)	Self Time (s)	Number of Calls
JetEngineBlade_AI		17.207	1.807	2014
LSTM		11.465	0.000	0
Scope1		3.895	3.895	1004
Neural State Space Model		0.028	0.000	0
From Workspace1		0.008	0.008	1003
Ambient Temperature		0.002	0.002	1003
Cooling Temperature		0.001	0.001	1003
Pressure		0.001	0.001	1003
Normalize1		0.000	0.000	0
Denormalize1		0.000	0.000	0
Denormalize		0.000	0.000	0
Normalize		0.000	0.000	0

Neural state-space model is approximately 1e6x faster than the FEA model

Deep Learning Toolbox verification library

Ensure robustness and reliability of deep neural networks

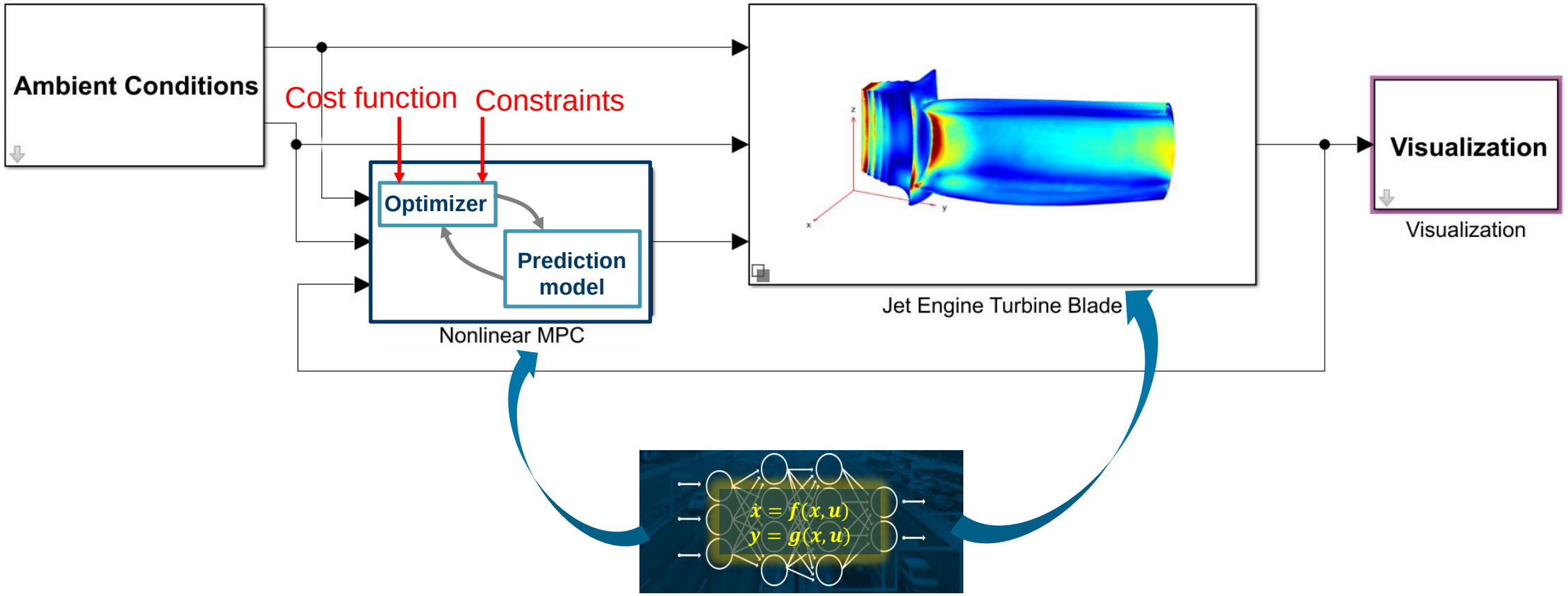
- **Verify Deep Neural Network Robustness for Classification**
- **Estimate Deep Neural Network Output Bounds for Regression**
- **Build Safe Deep Learning Systems with Runtime Monitoring**
- **Case Study: Verifying an Airborne Deep Learning System**



Deep Learning Toolbox Verification Library

Control Design with Model Predictive Controller

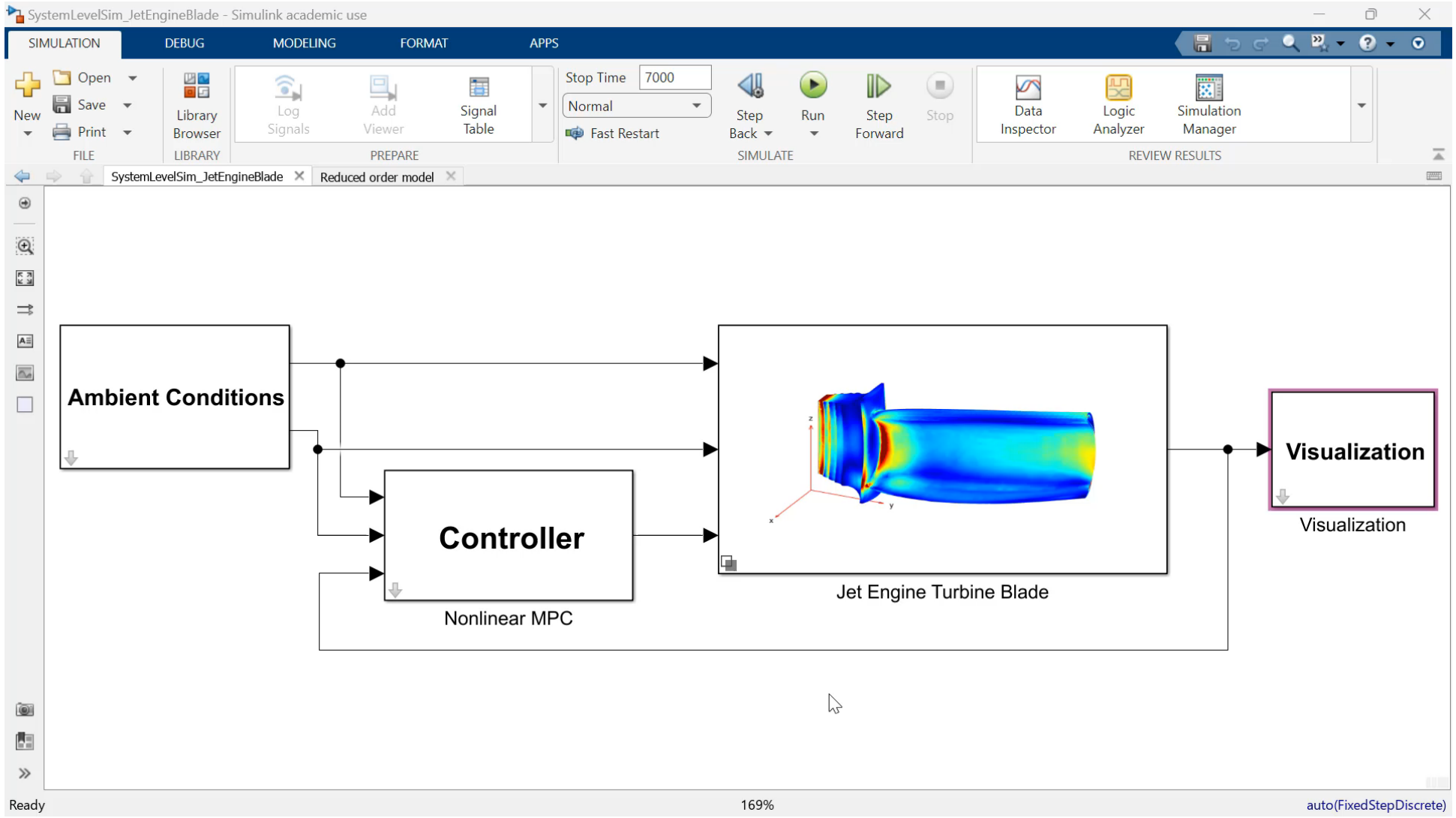
SIMULINK®



[Nonlinear MPC with data-driven prediction model]

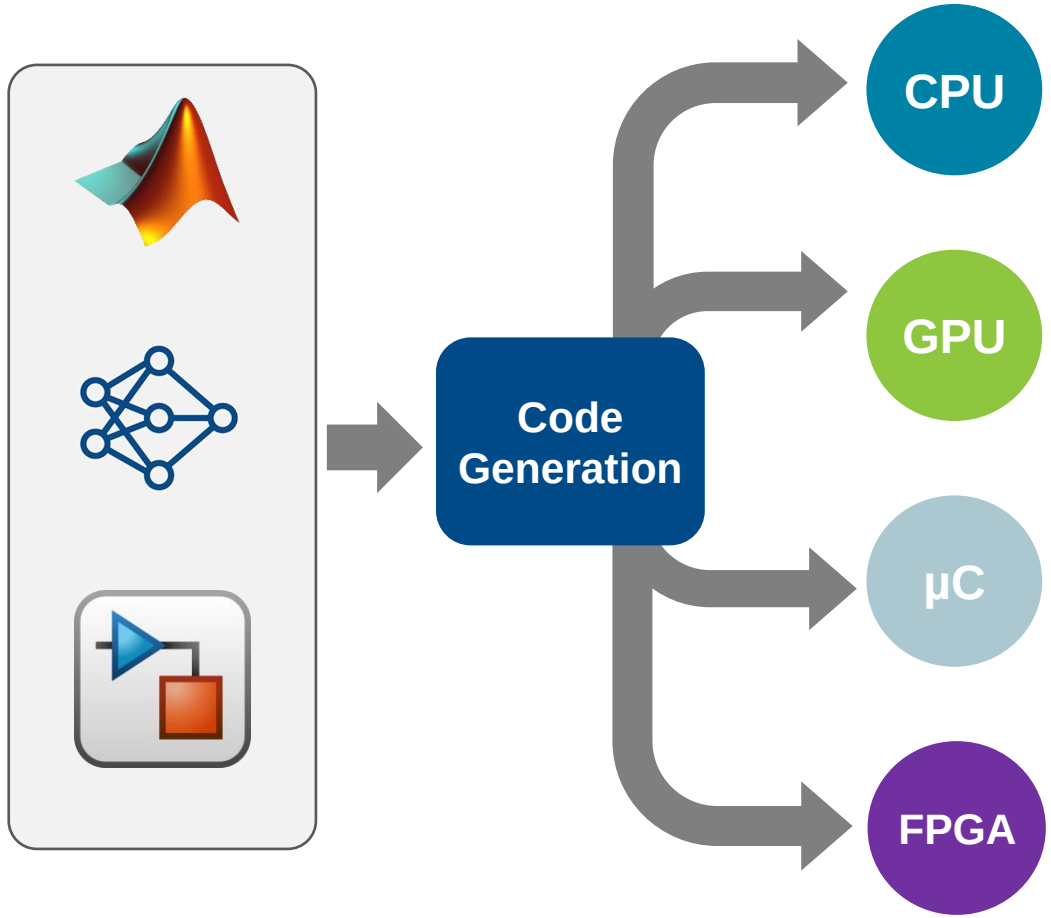
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

System-level simulation



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

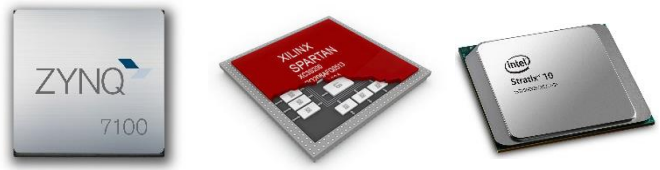
Deploy to target with zero coding errors



Any CPU
No Library needed

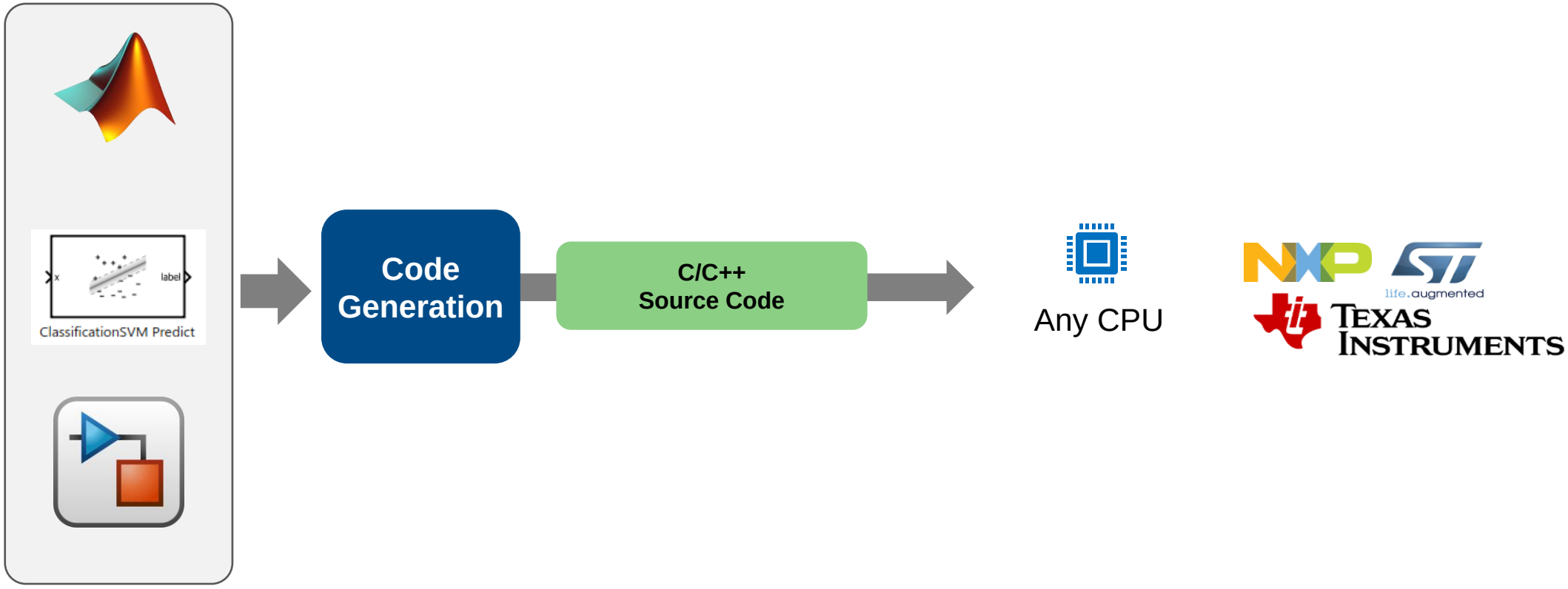
oneDNN
Library

ARM Compute
Library



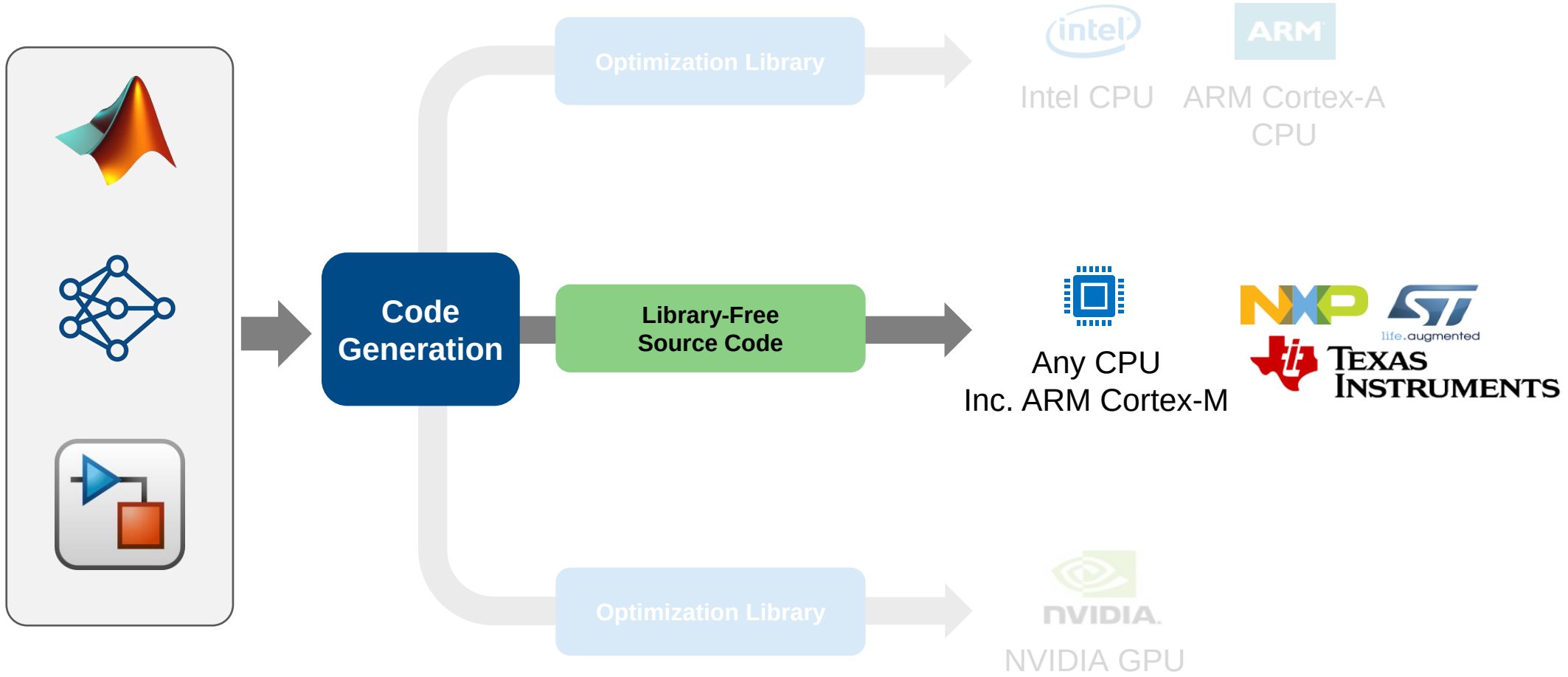
Data Preparation | AI Modeling | Simulation & Test | Deployment

Use Embedded Coder to generate code for machine learning



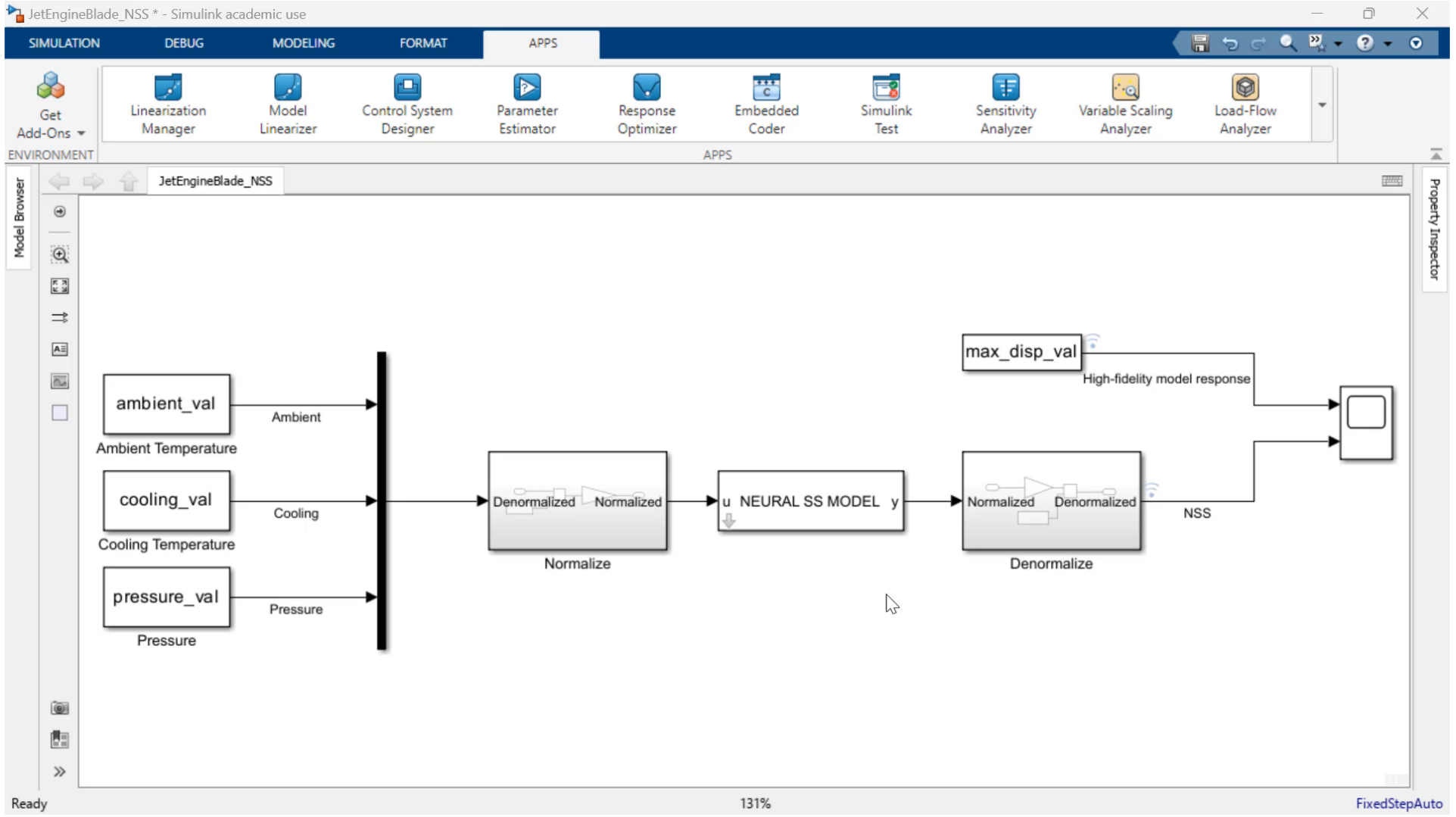
Data Preparation | AI Modeling | Simulation & Test | **Deployment**

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



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

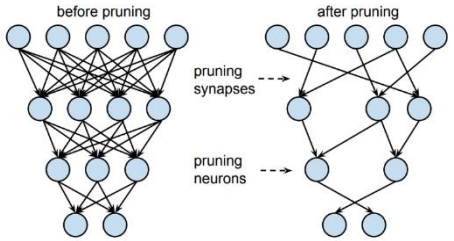
Generate Library-Free C Code for Deep Learning Networks



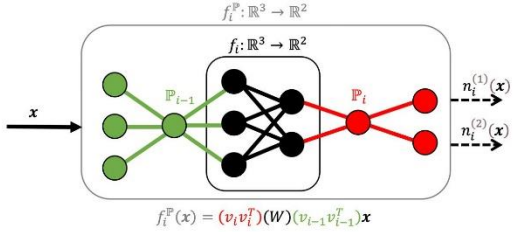
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Model compression reduces Model for Deployment (reduces learnable from 2M to 850 K)

Structural Compression

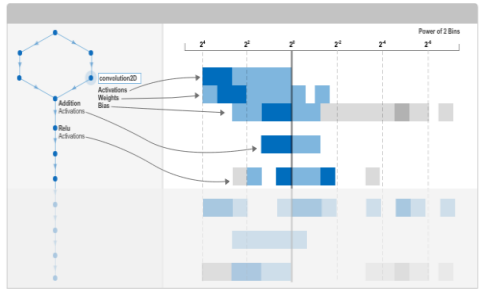


Pruning convolutional neural networks



[Projection](#) of deep neural networks

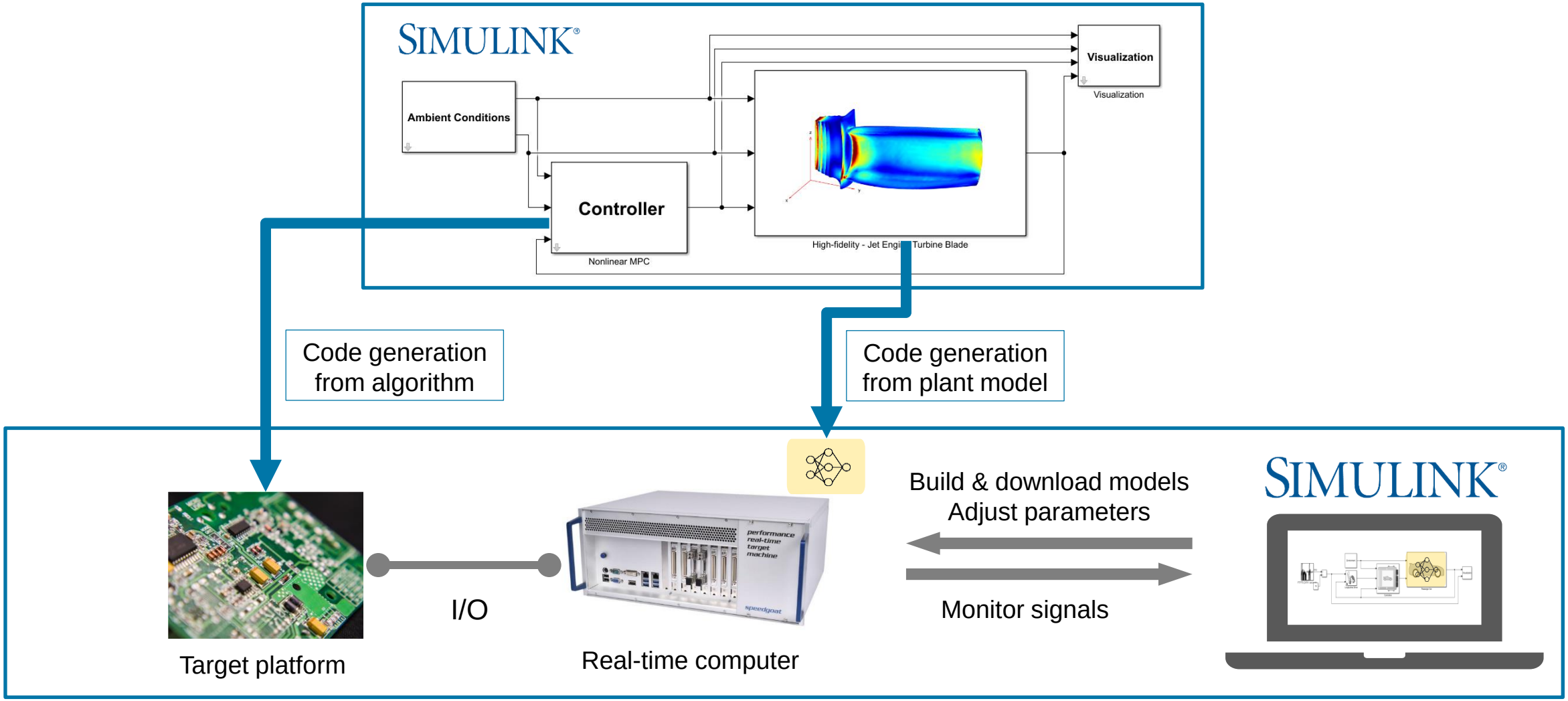
Datatype Compression



Quantization of network weights to lower precision datatypes (bfloat16, int8)

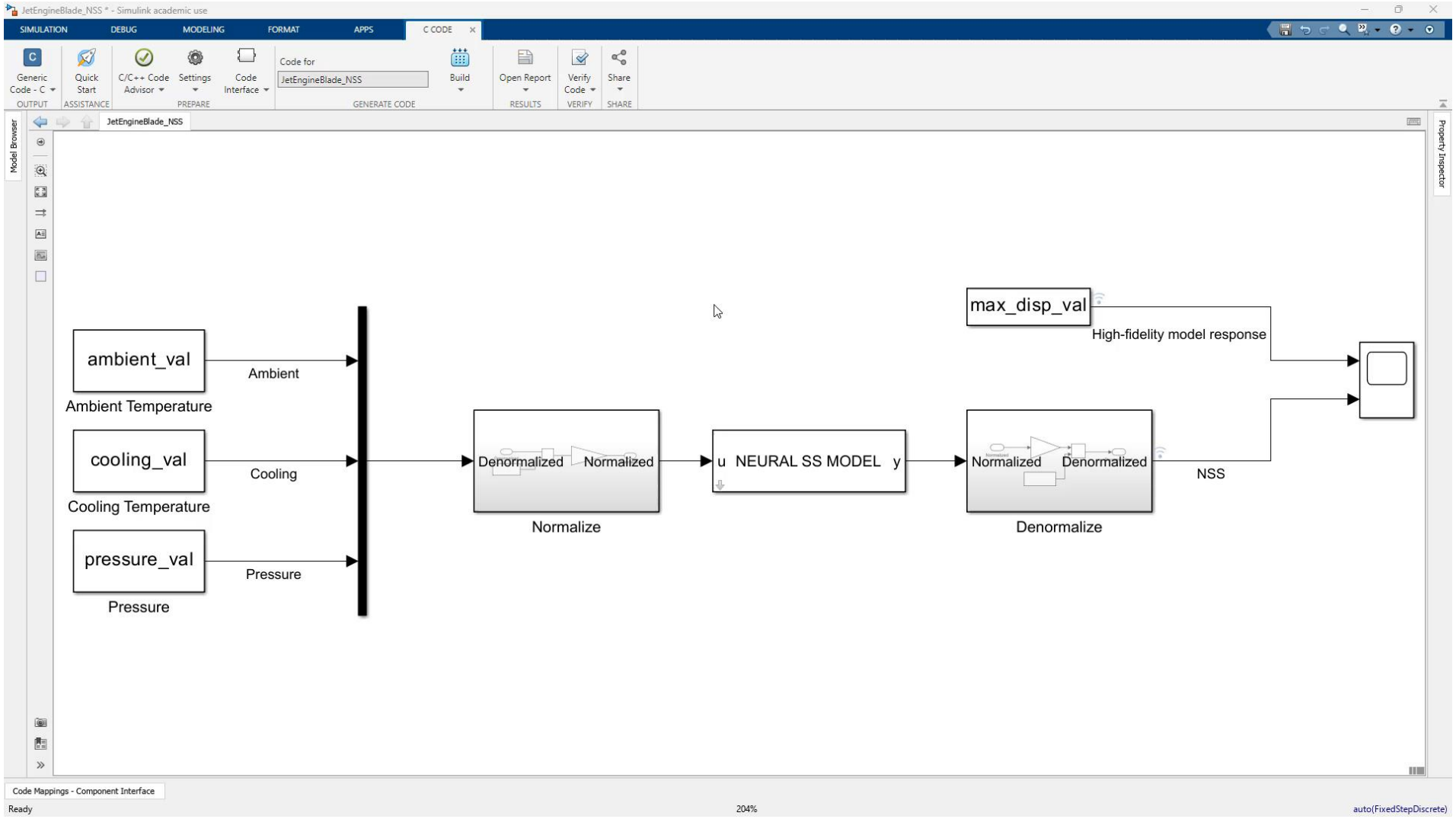
Hardware-in-the-loop simulation

System-level integration and test



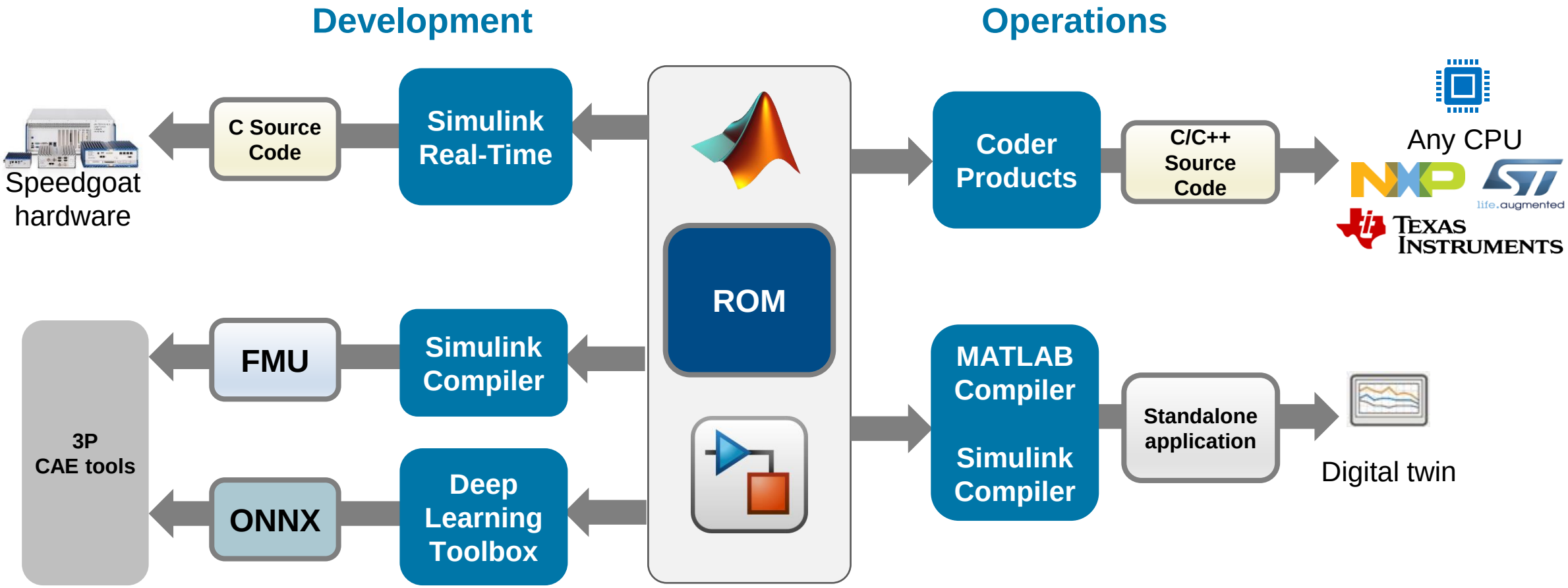
Data Preparation | AI Modeling | Simulation & Test | **Deployment**

Hardware-in-the-loop simulation



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

Use ROMs outside of Simulink, for development and operation stages



Data Preparation | AI Modeling | Simulation & Test | Deployment

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 Jet Engine Blade 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

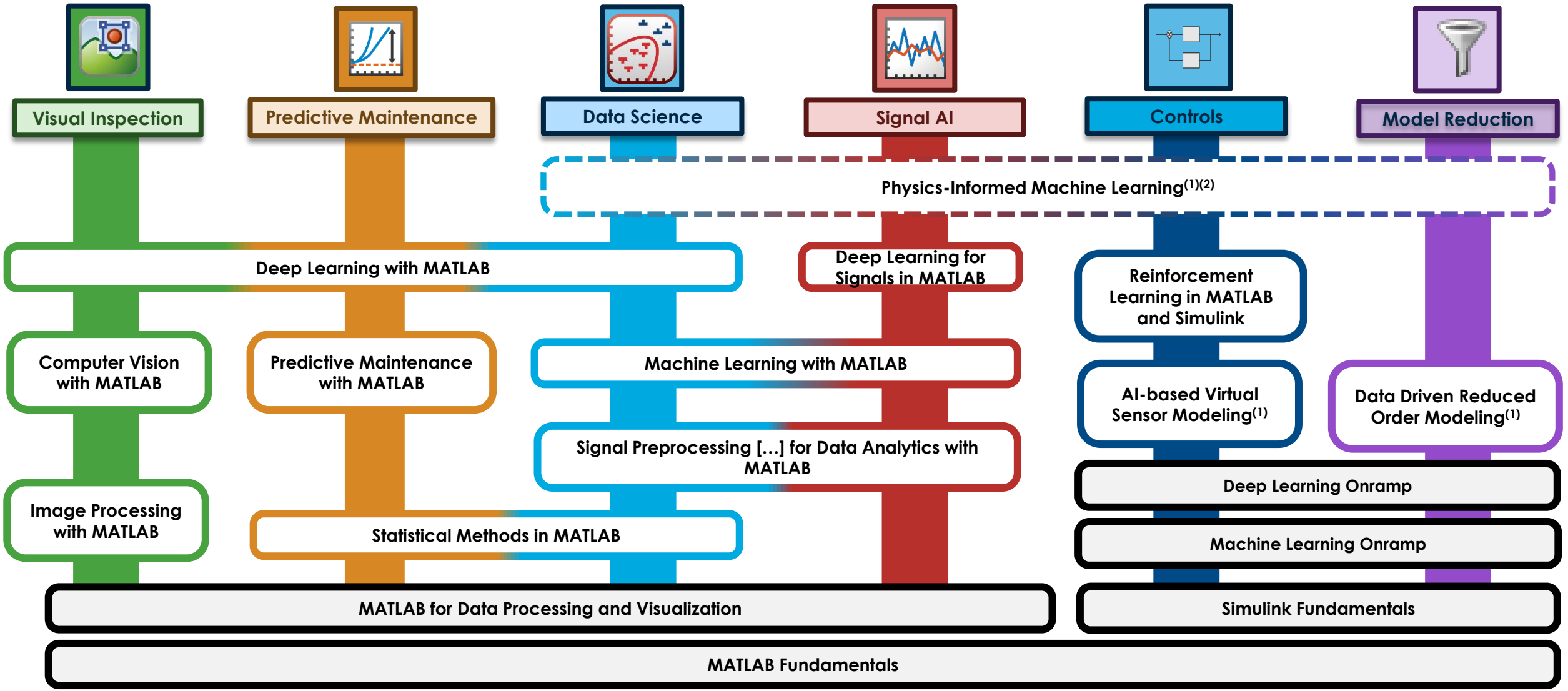
Consulting Engineering

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

Complimentary → Funded

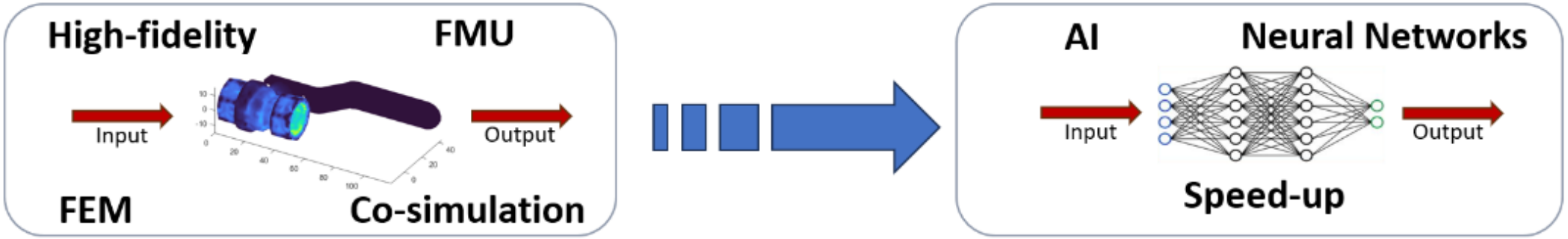
AI training pathways

(1) available as private-only
(2) module, not a full training



Click on the boxes for a full agenda, or see the overview of AI-courses on our website

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**