MATLAB EXPO

June 4, 2024 | Warsaw

AI with Model-Based Design: Reduced Order Modeling

Sebastian Bomberg, MathWorks

MathWorks

SIMULATION DEBUG MODELING FORMAT APPS VARIANT SUBSYSTEM

New Open Save Print FILE

Library Browser LIBRARY

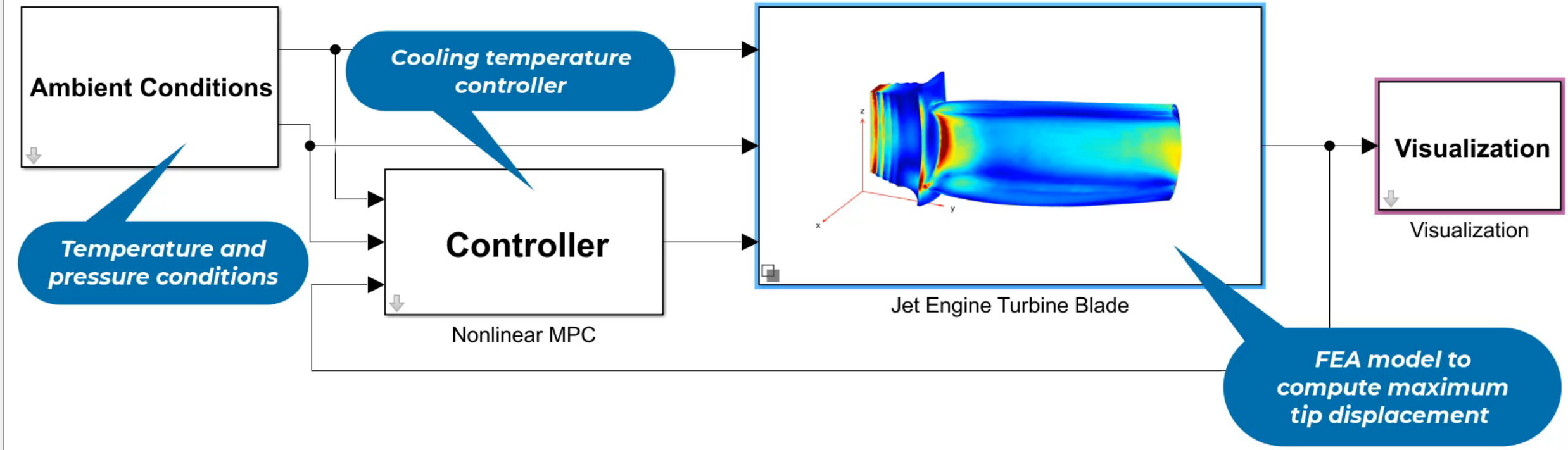
Log Signals Add Viewer Signal Table PREPARE

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)

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.

Key takeaways

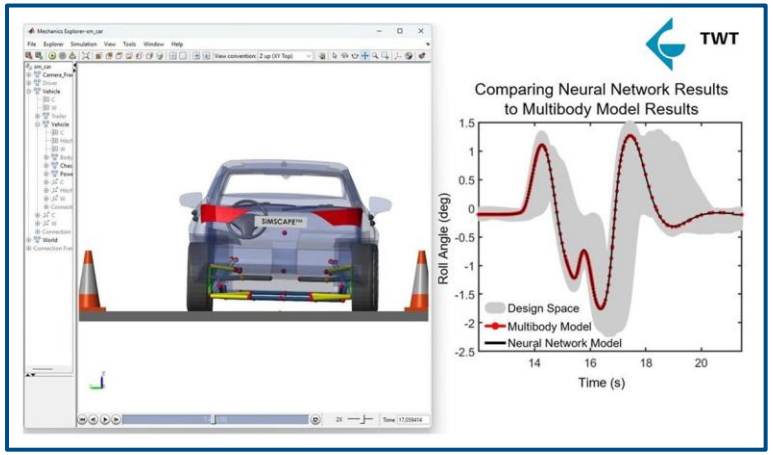
Enable	Reuse of full-order high-fidelity models for system-level simulations, Hardware-in-the-Loop (HIL) testing, nonlinear control design, and virtual sensor modeling.
Explore	Various ROM techniques in MATLAB to find the best method.

TWT GmbH Develops New Workflow for Tuning Automotive Suspension Designs Using Deep Learning and Multibody Simulation

Using MATLAB, Simulink, and Deep Learning Toolbox, TWT GmbH has accelerated the optimization of automotive suspension designs via a deep learning network trained using data from simulations of a Simscape multibody model.

Key Outcomes/Advantages:

- New workflow with MATLAB and Simulink improved suspension design, helping reduce roll angle by up to 50%
- Using Global Optimization Toolbox reduced optimization time for suspension simulations from 16 days to 5 minutes
- Single-environment workflow established for multibody simulation and deep learning



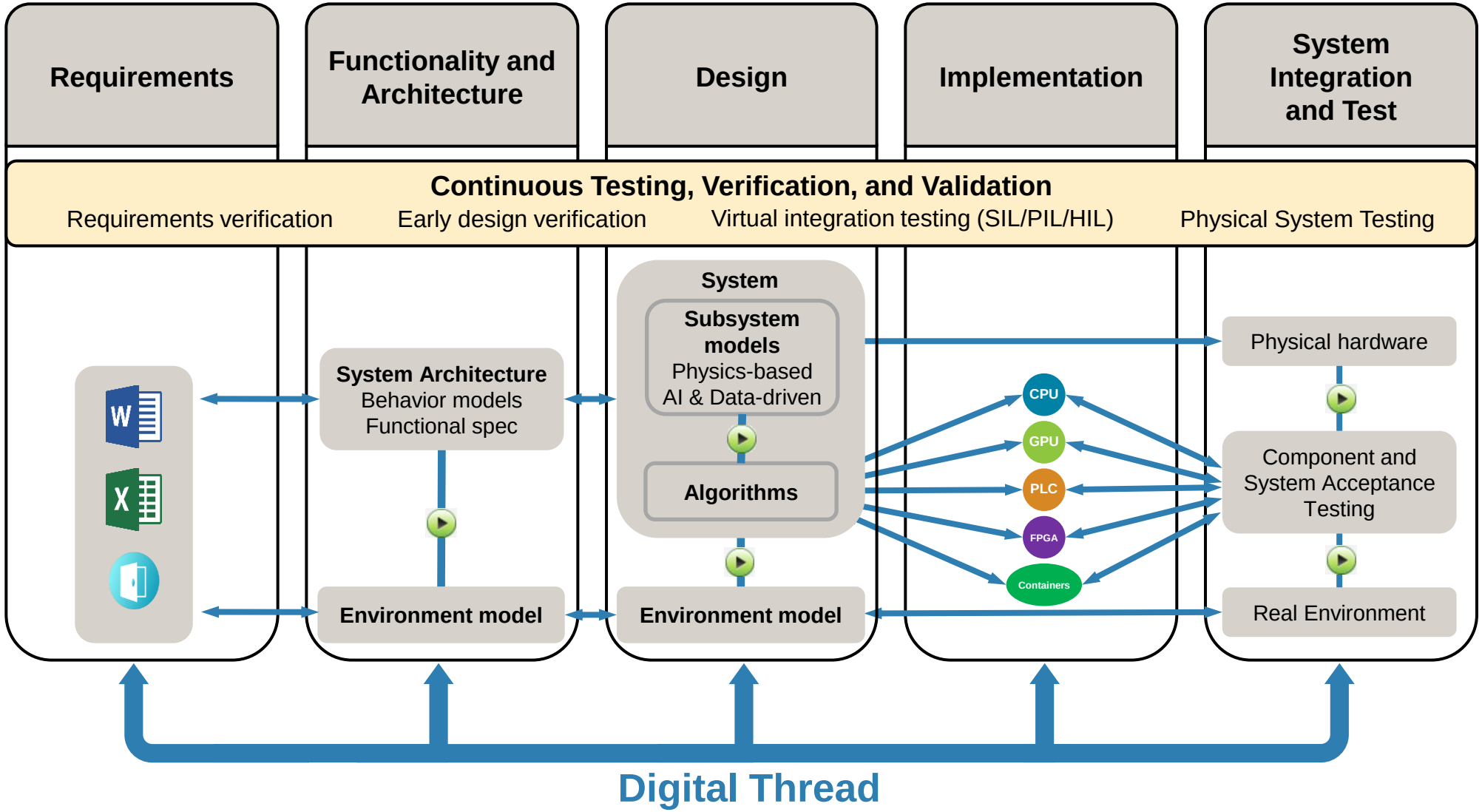
Comparison of simulation results between a Simscape vehicle model with a multibody suspension and a neural network trained using synthetic data from the Simscape model.

"As expected, performance was satisfactory, proving that all three networks could generalize well on new data. While the Keras neural network managed to fit the first peak almost perfectly, it was not as accurate with the reaming plot. The hybrid neural network was an improvement and generally fit the real curve. Finally, as expected, the MATLAB neural network had the best performance, and its prediction was extremely accurate."

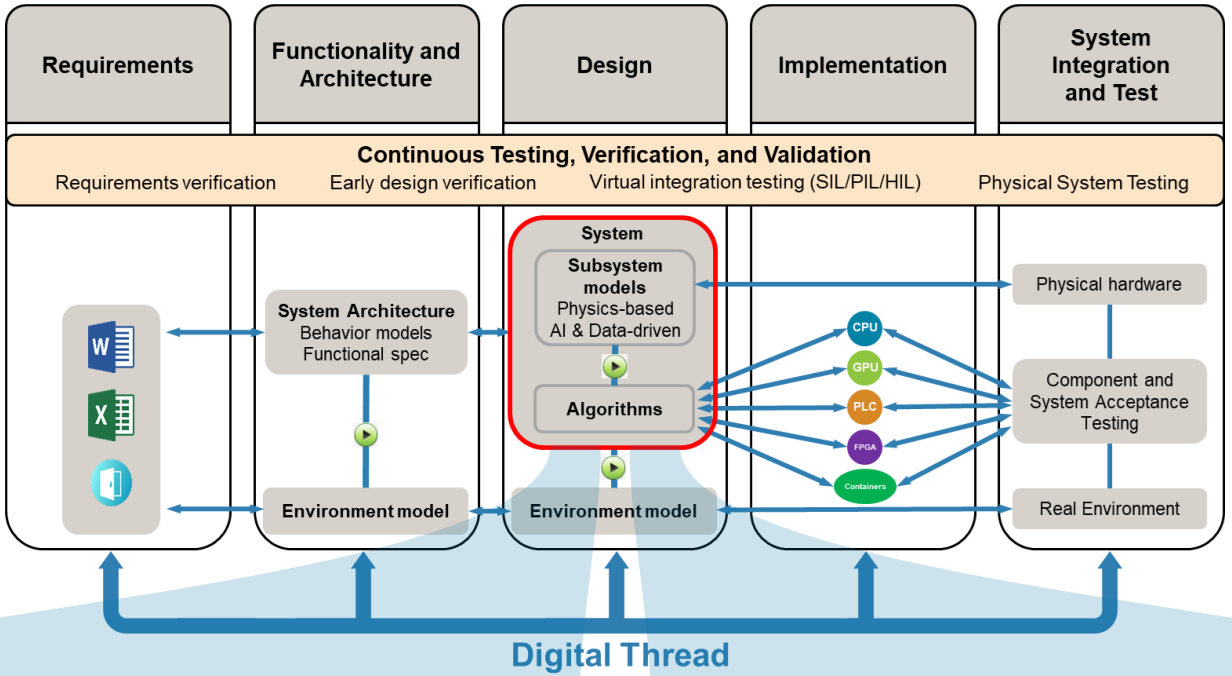
- Georgios Papageorgiou, TWT GmbH Science & Innovation

Link to user story

Model-Based Design



Integrate AI models into MBD for system-level simulation and code generation



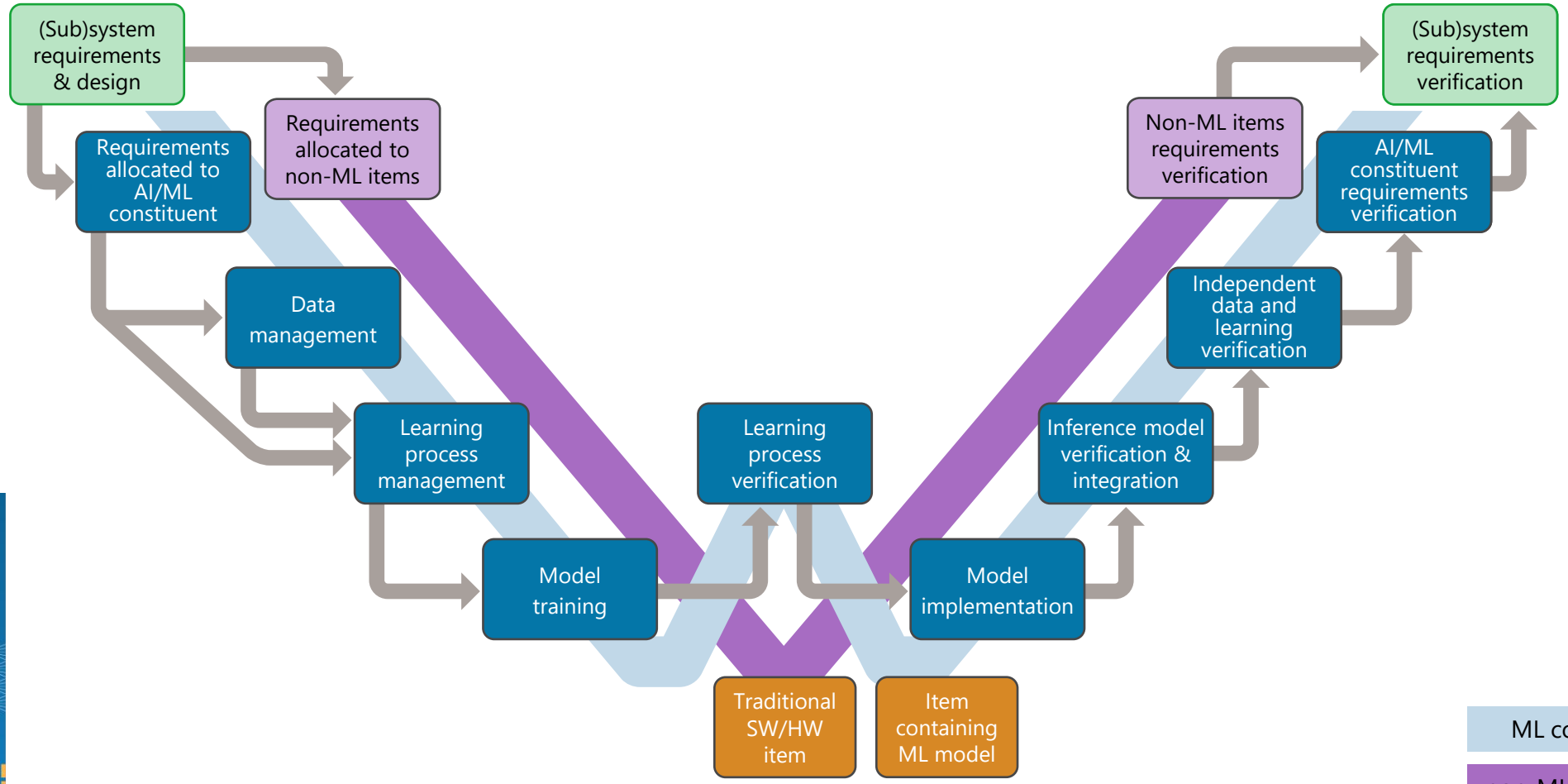
AI for component modeling

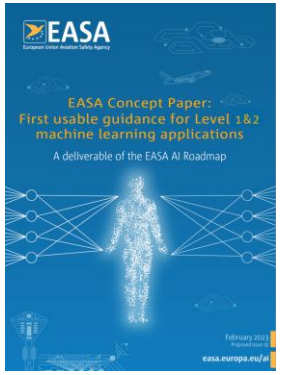
- HIL testing and system-level simulation for high-fidelity models
- Modeling component dynamics from data when first-principles models cannot be obtained

AI for algorithm development

- Virtual sensor modeling
- Sensor fusion
- Object detection

W-shaped development process adapting the classical V-shaped cycle to ML applications





Source: EASA Concept Paper Proposed ISSUE 02 'First usable guidance for Level 1&2 machine learning applications'
Credit: EASA, Daedalean

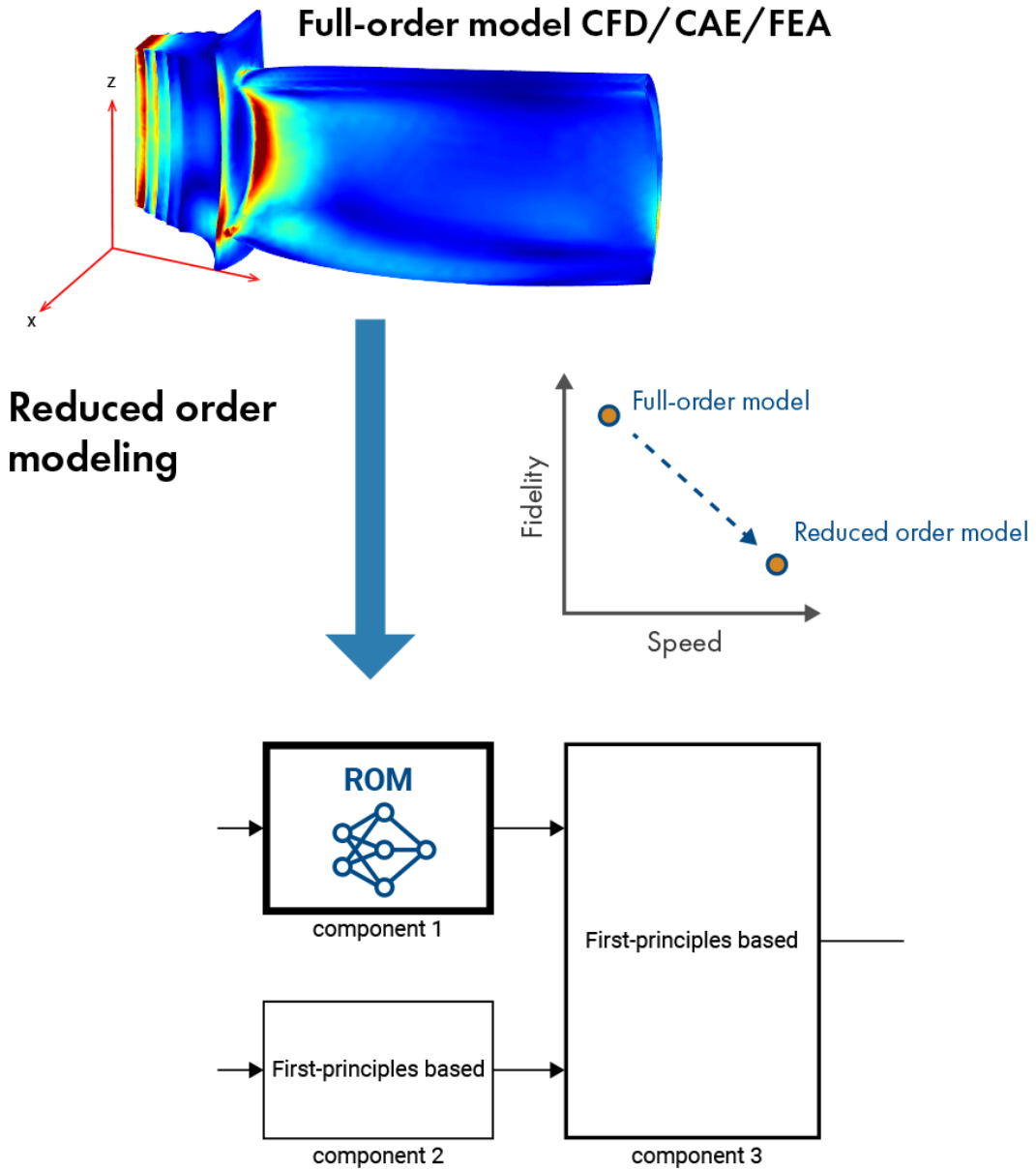
Reduced Order Modeling

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
- Enable desktop simulations for orders-of-magnitude longer timescales



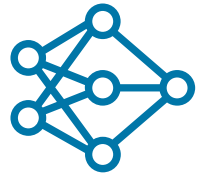
Reduced Order Modeling

How

Reduced order model

AI-Based Data-driven

Inputs
Ambient Temperature
Ambient Pressure
Cooling Temperature

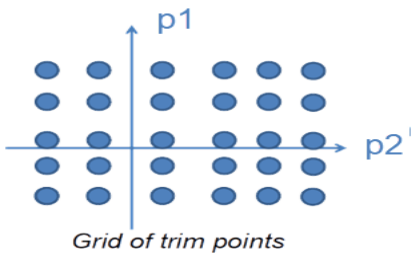


AI model

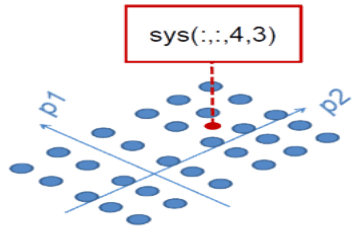
Outputs
Max Displacement

focus today

Linearization

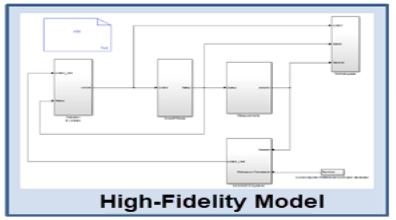


Grid of trim points



Array of State-Space systems

Loop through the grid of trim points



Identify local model at each trim point

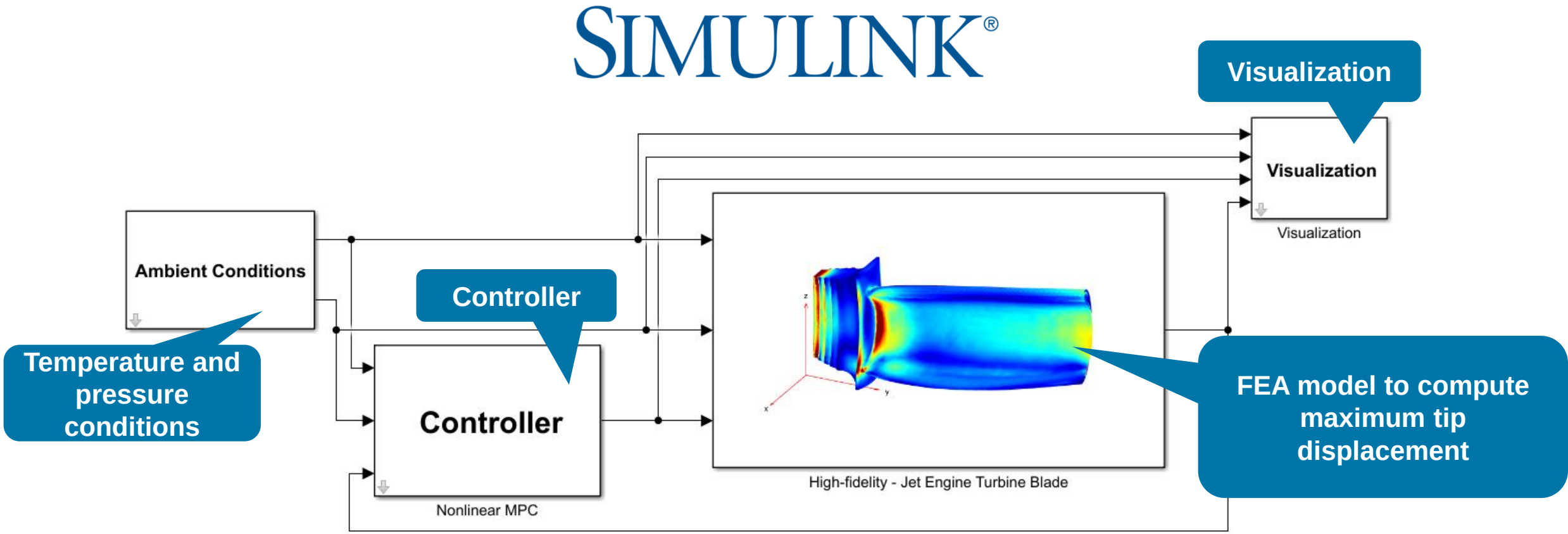
Model-based

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

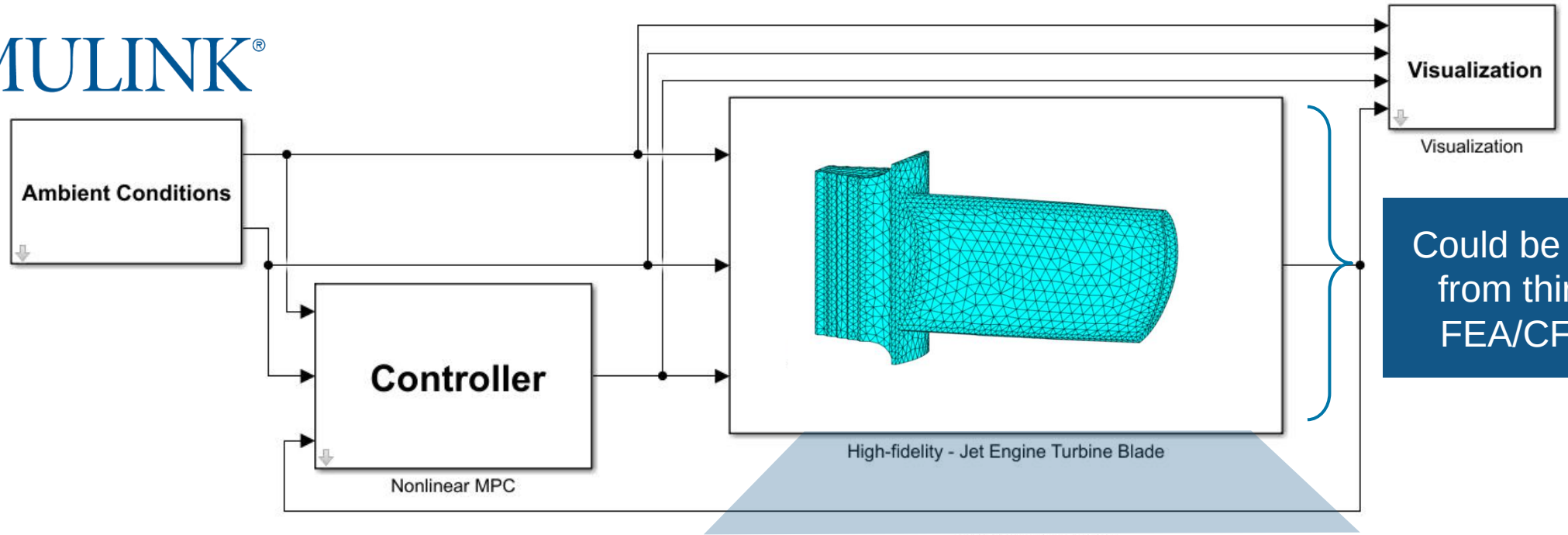


Closed-loop temperature control

Example overview

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

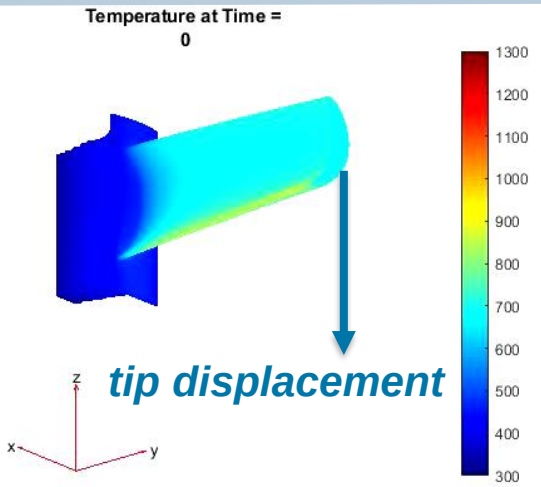
SIMULINK®



Could be imported from third-party FEA/CFD tools

~30 seconds per time step for solving FEA models

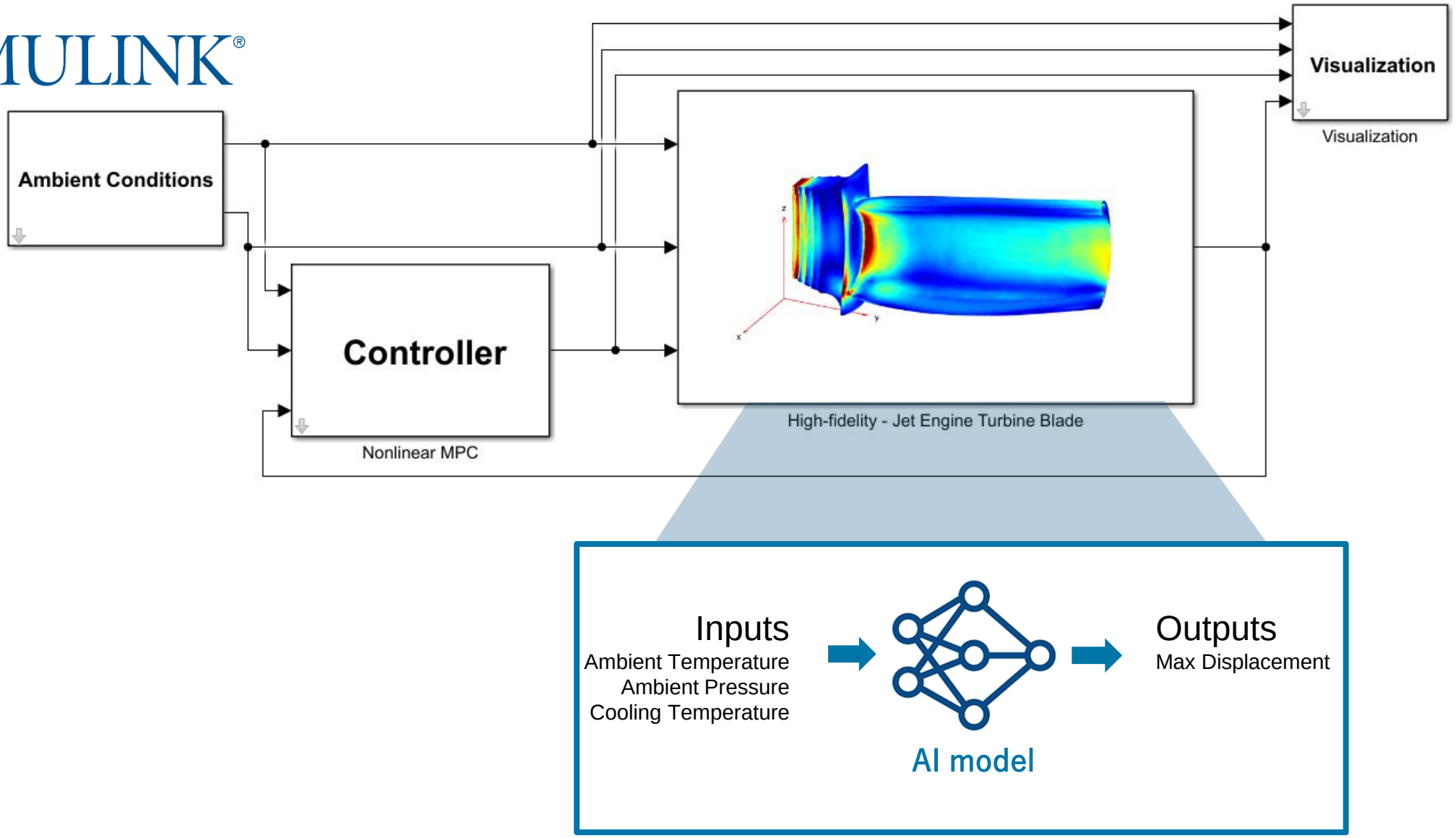
Not suitable for control design and HIL testing



Example overview

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

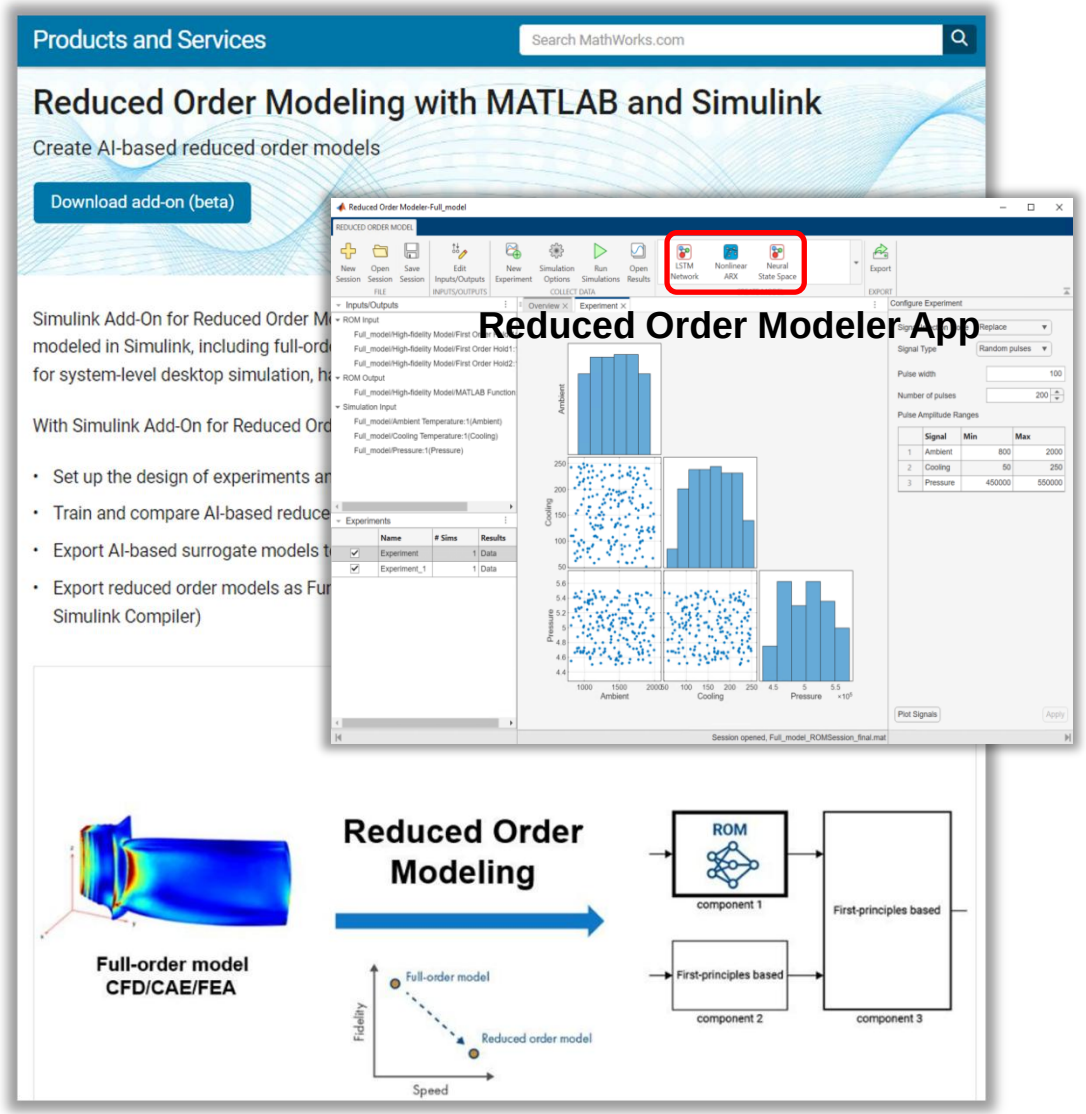
SIMULINK®



Introducing Simulink Add-On for Reduced Order Modeling

Create AI-based reduced order models (ROM)

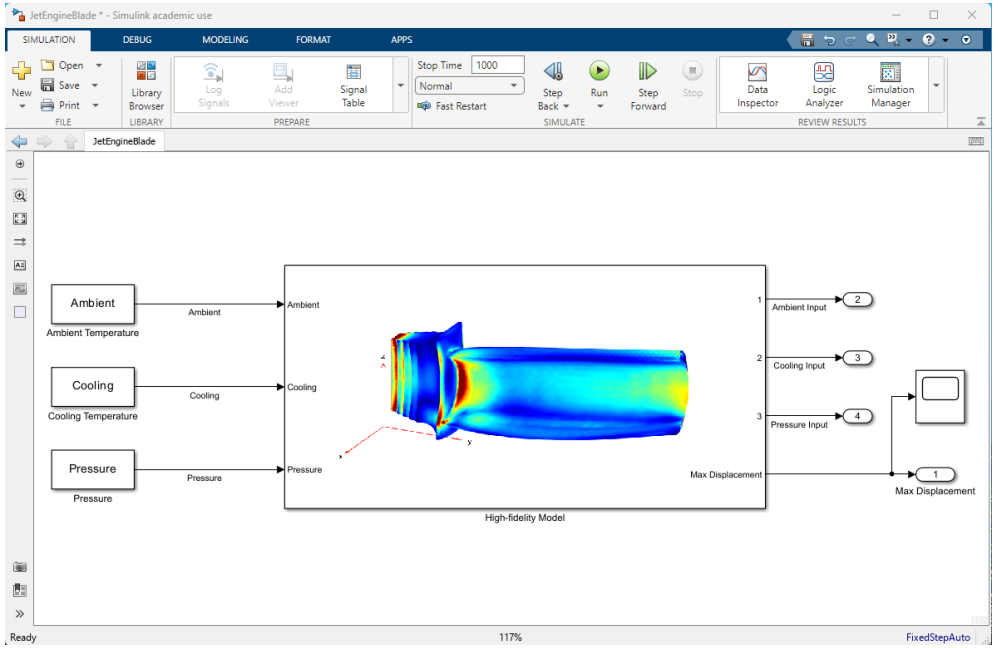
- Set up Design of Experiments (DoE)
- Generate input-output data from full-order, high-fidelity subsystems
- Train and compare AI-based reduced order models using preconfigured templates
- Export trained reduced order models into Simulink or outside of Simulink through FMUs



Reduced Order Modeling with MATLAB and Simulink

Generate data for training

Physical system

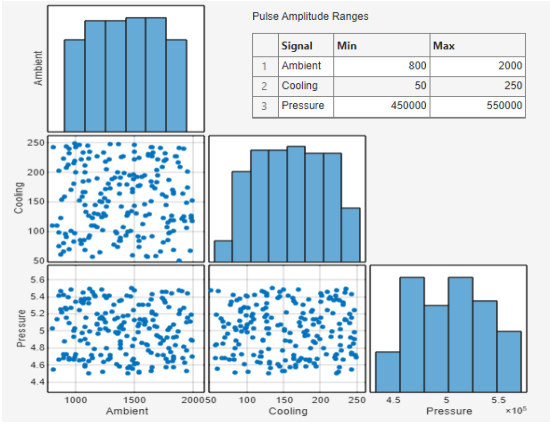


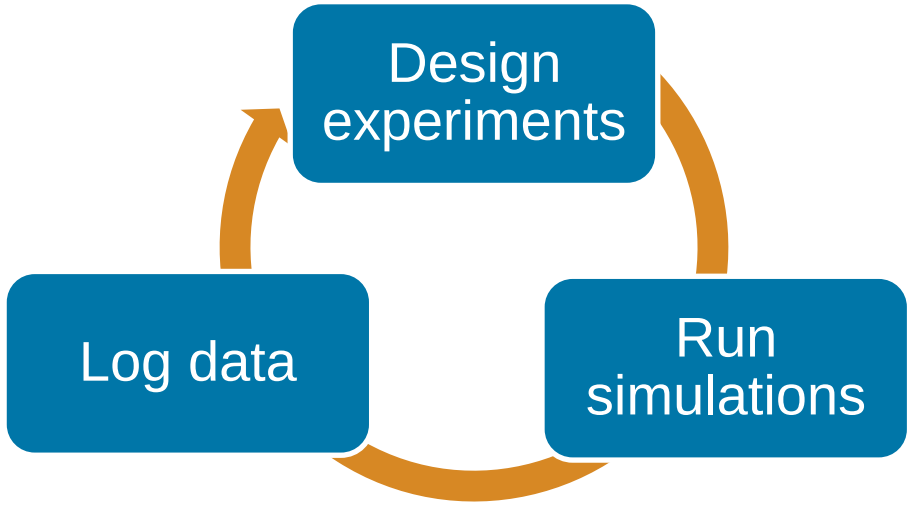
Simulink/Simscape

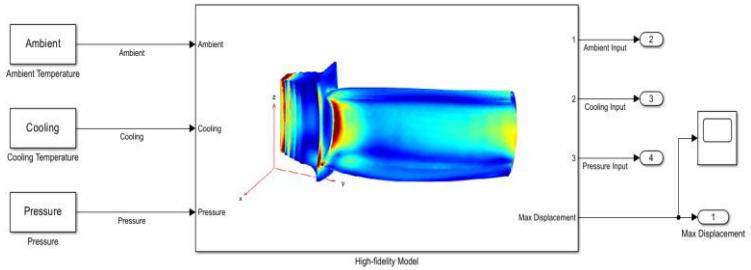
Data Preparation | AI Modeling | Simulation & Test | Deployment

Synthetic Data Generation

Design of Experiments







Input features

Ambient Temperature
Ambient Pressure
Cooling Temperature

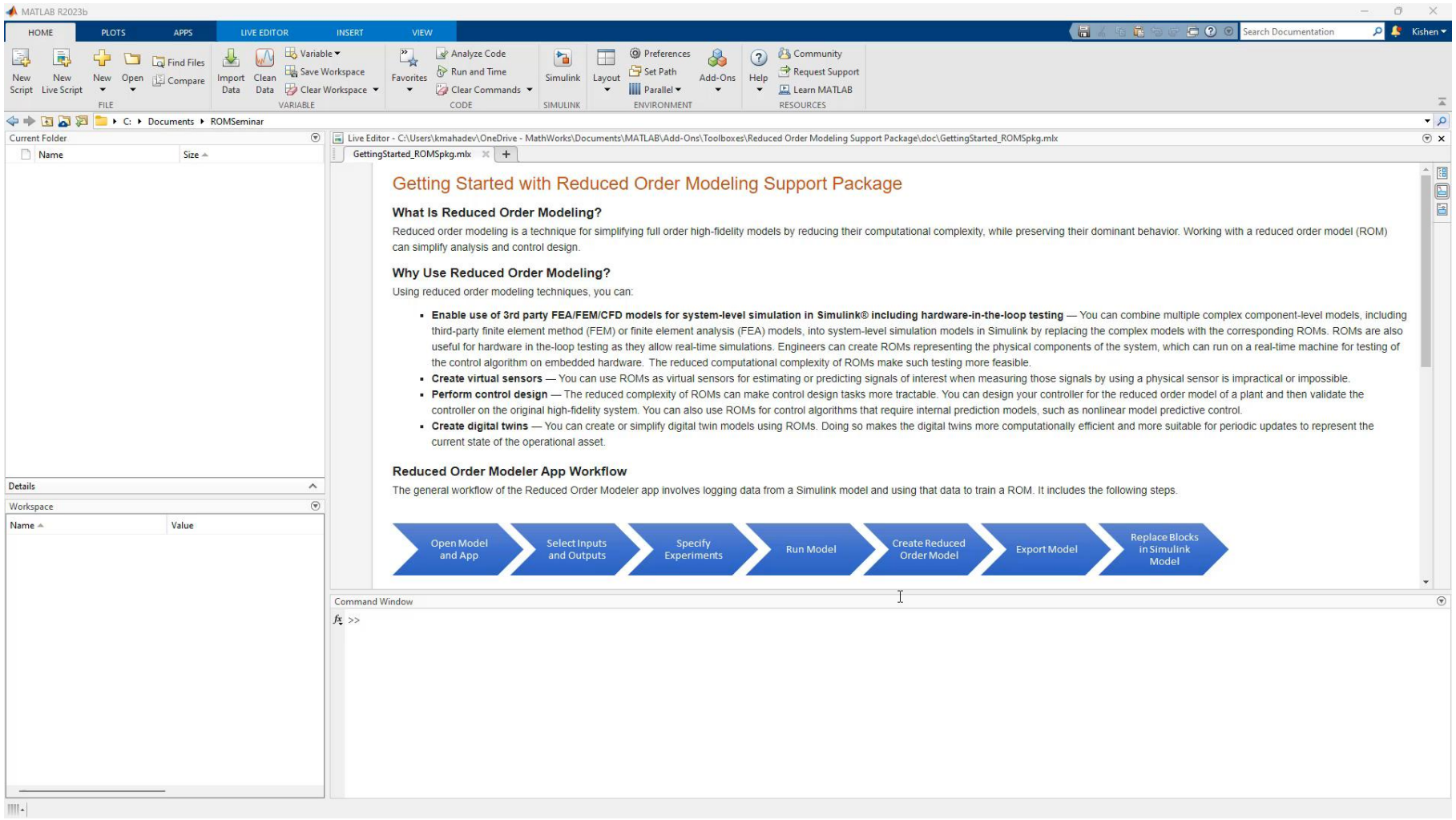
Response

Max Displacement

Data Preparation | AI Modeling | Simulation & Test | Deployment

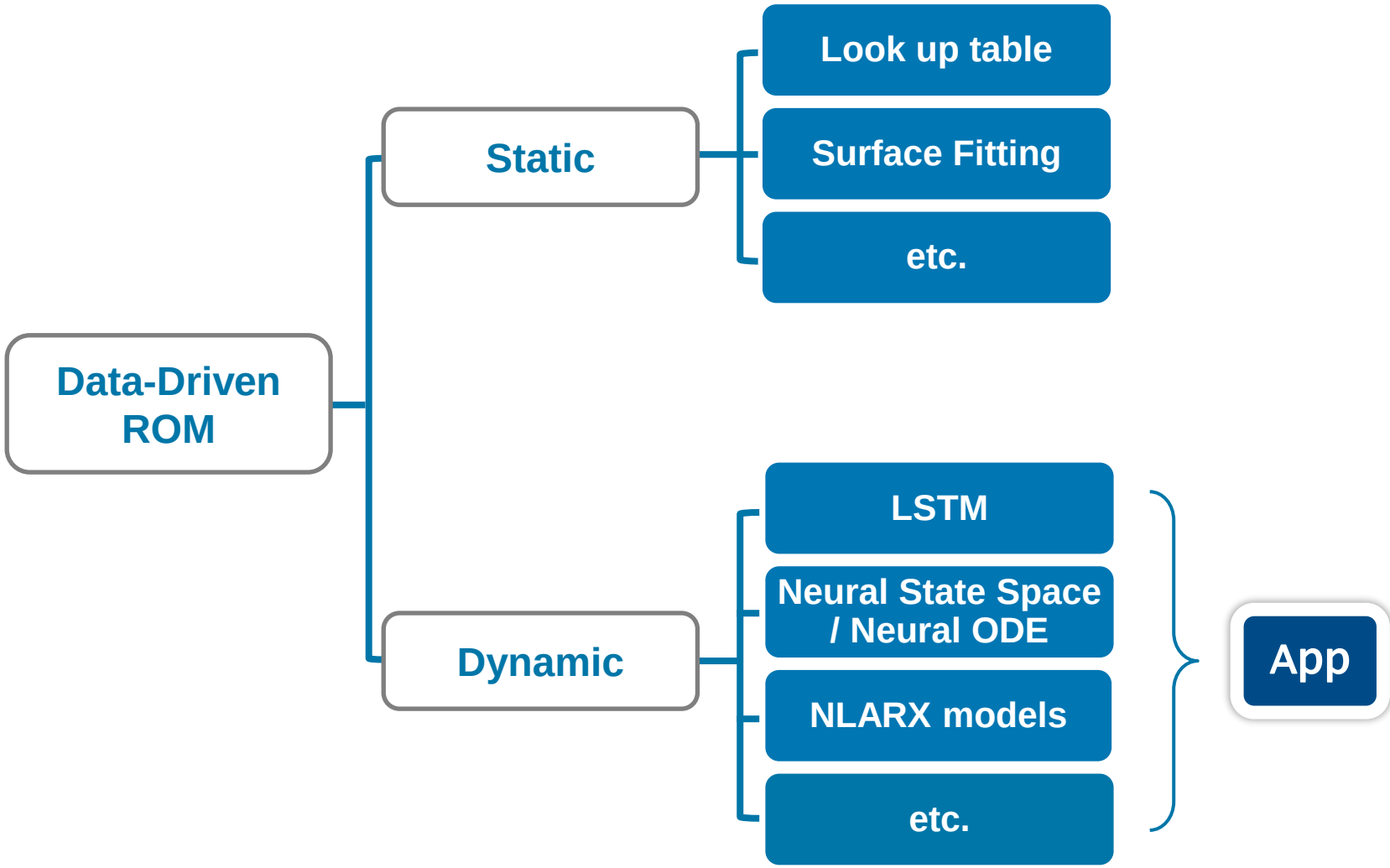
Synthetic Data Generation

Design of Experiments



Data Preparation | AI Modeling | Simulation & Test | Deployment

Data-driven ROM



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

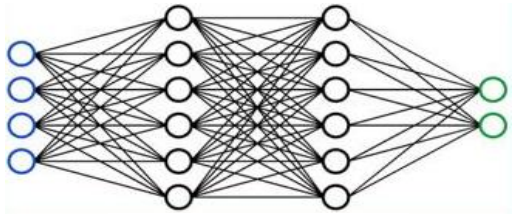
- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure_input)
- ROM Output
  - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)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

Experiments

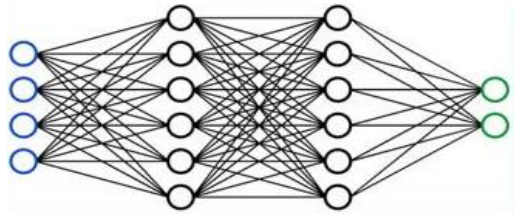
	Name	# Sims	Results
☑	Experiment	1	Data
☑	Experiment_1	1	Data

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

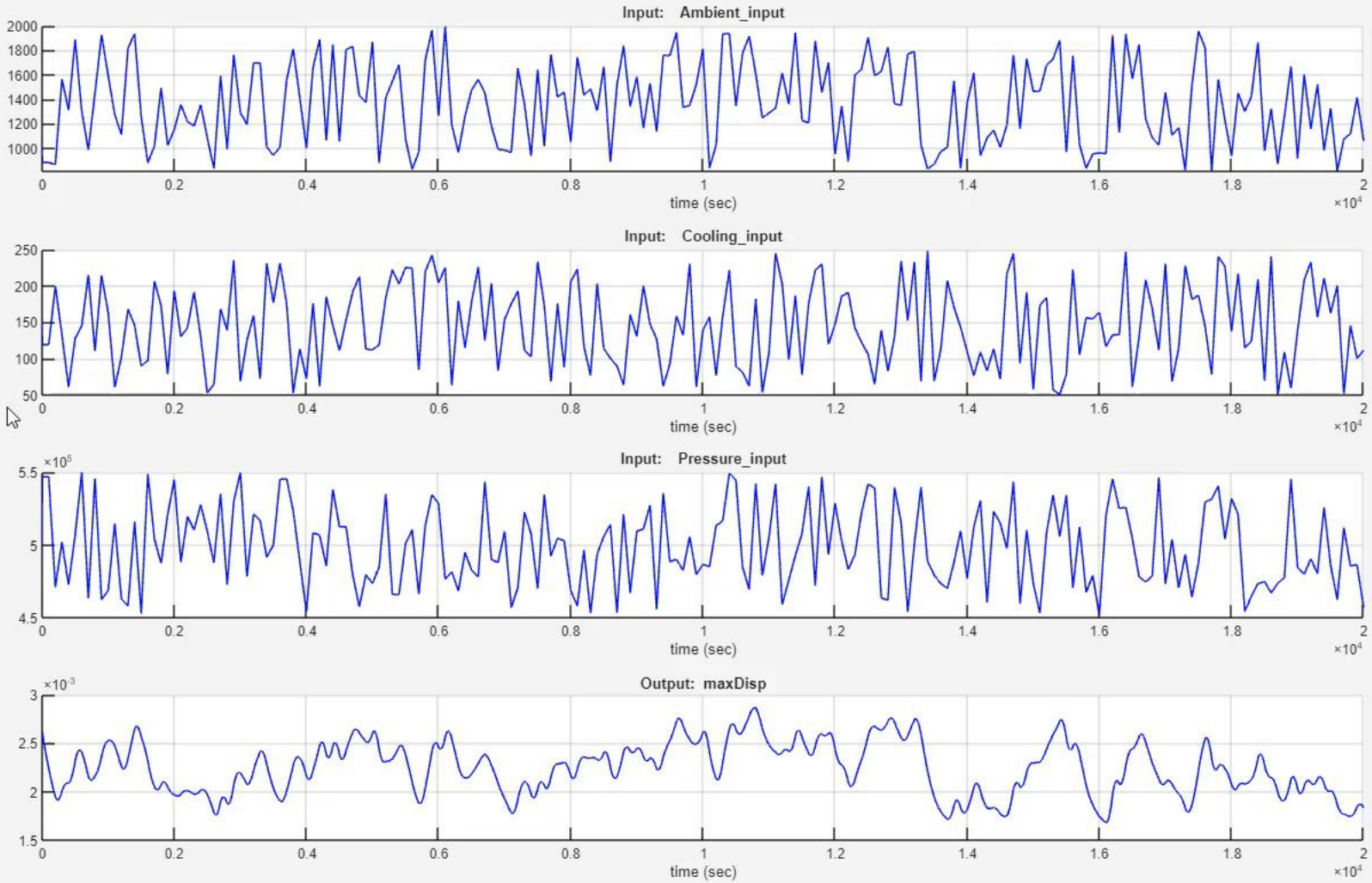
Inputs/Outputs

- 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

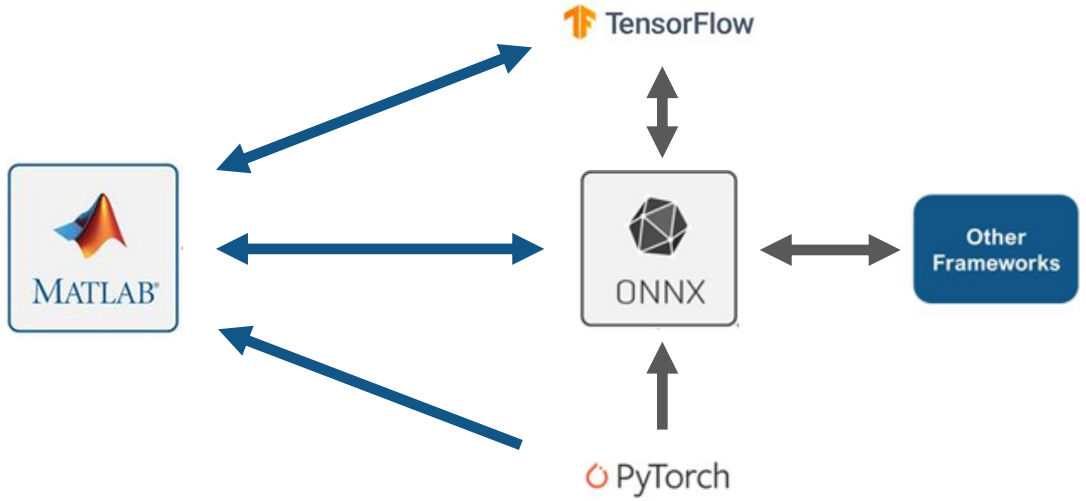


Session opened, engineBlade_ROMsession_results.mat

MATLAB interoperates with other frameworks

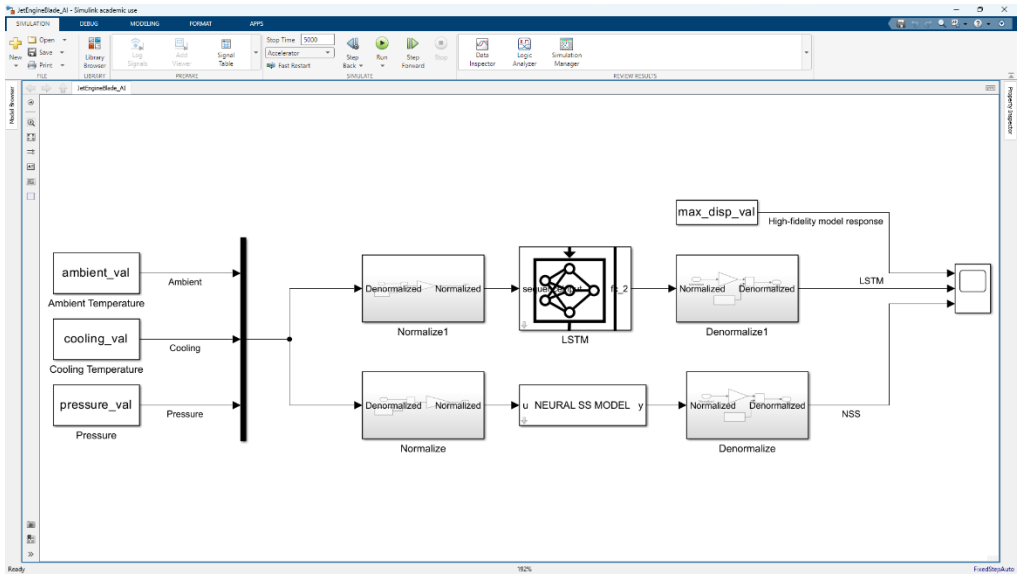
Framework interoperability bridges the gap between data science, engineering and production

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

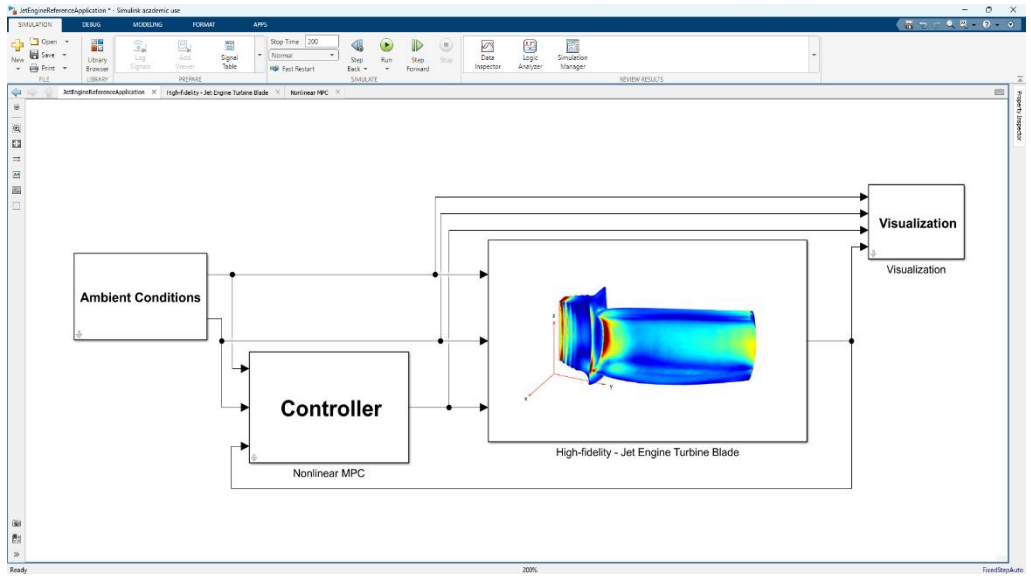


Integrate your AI model for system-level simulation and test

Integration of trained AI model into Simulink



System-level simulation

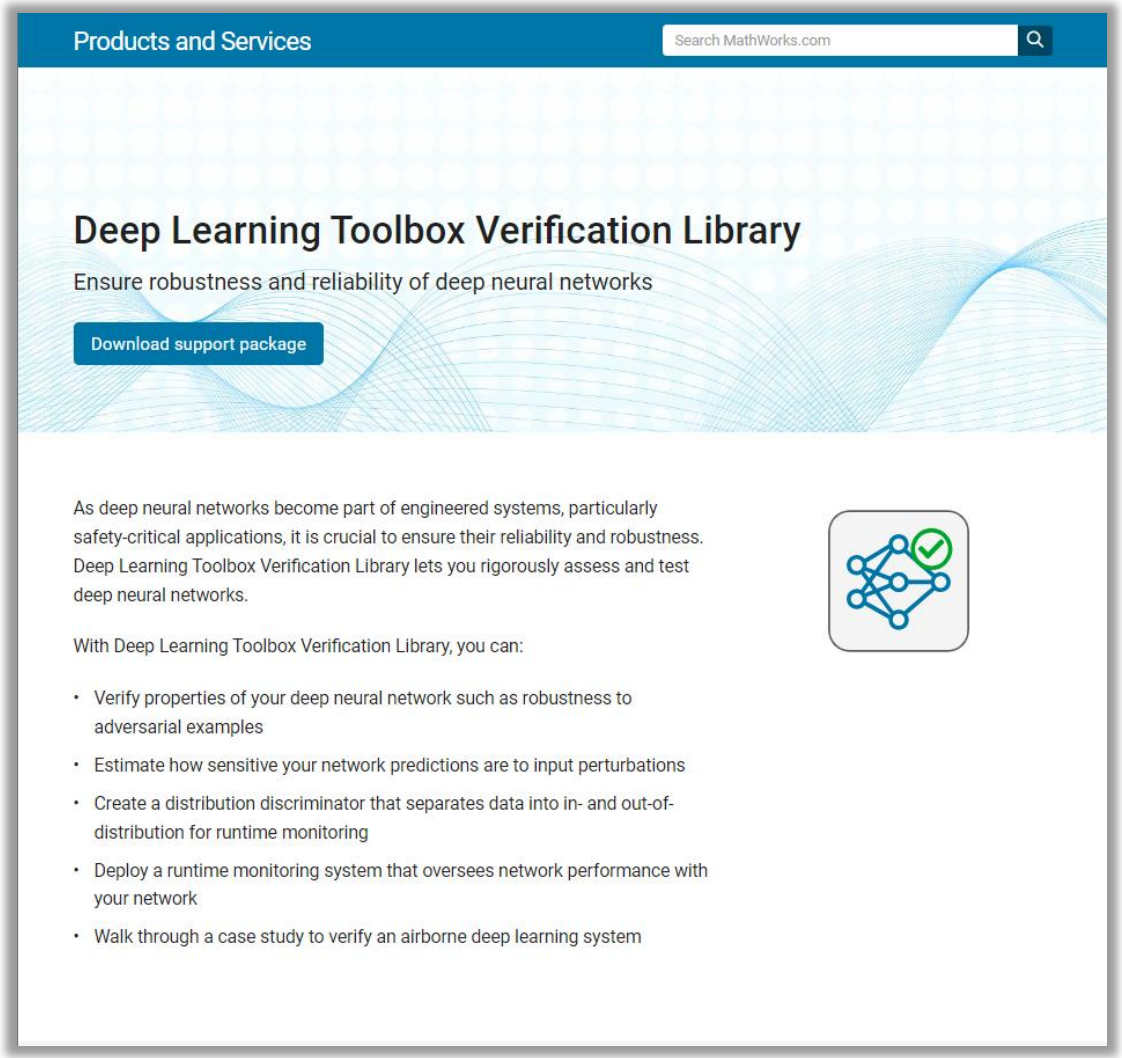


Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Deep Learning Toolbox Verification Library

Ensure robustness and reliability of deep neural networks

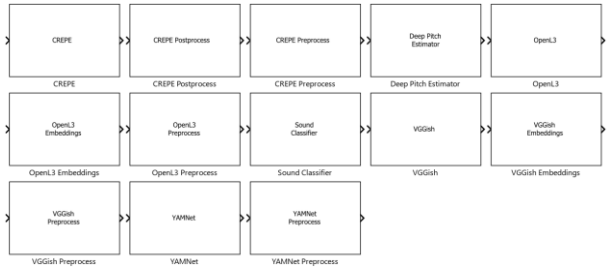
- **Verify Deep Neural Network Robustness for Classification**
- **Estimate Deep Neural Network Output Bounds for Regression**
- **Build Safe Deep Learning Systems with Runtime Monitoring**
- **Case Study: Verifying an Airborne Deep Learning System**



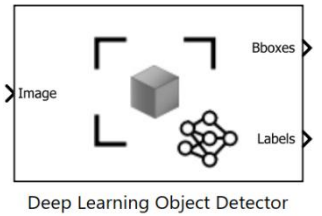
Deep Learning Toolbox Verification Library

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

Specialized

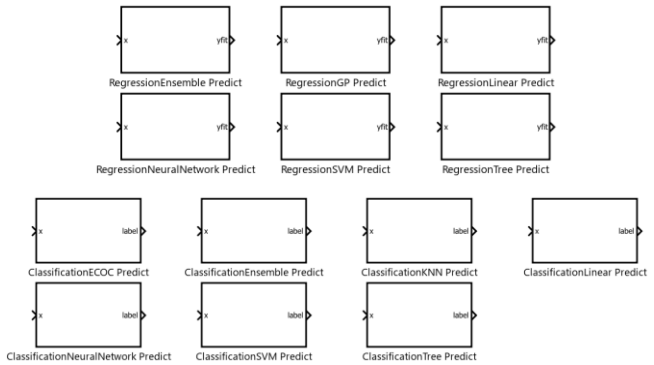


Audio Toolbox

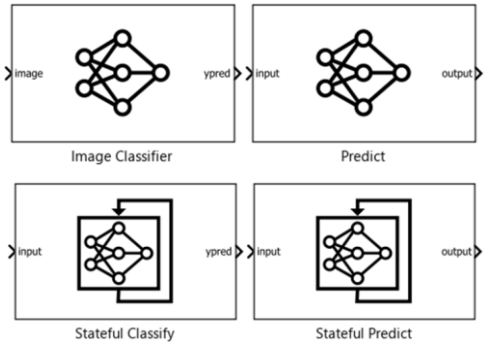


Computer Vision Toolbox

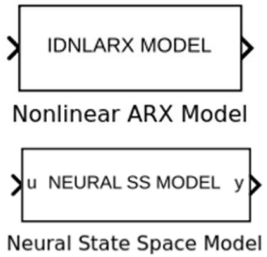
AI Core



Statistics and Machine Learning Toolbox



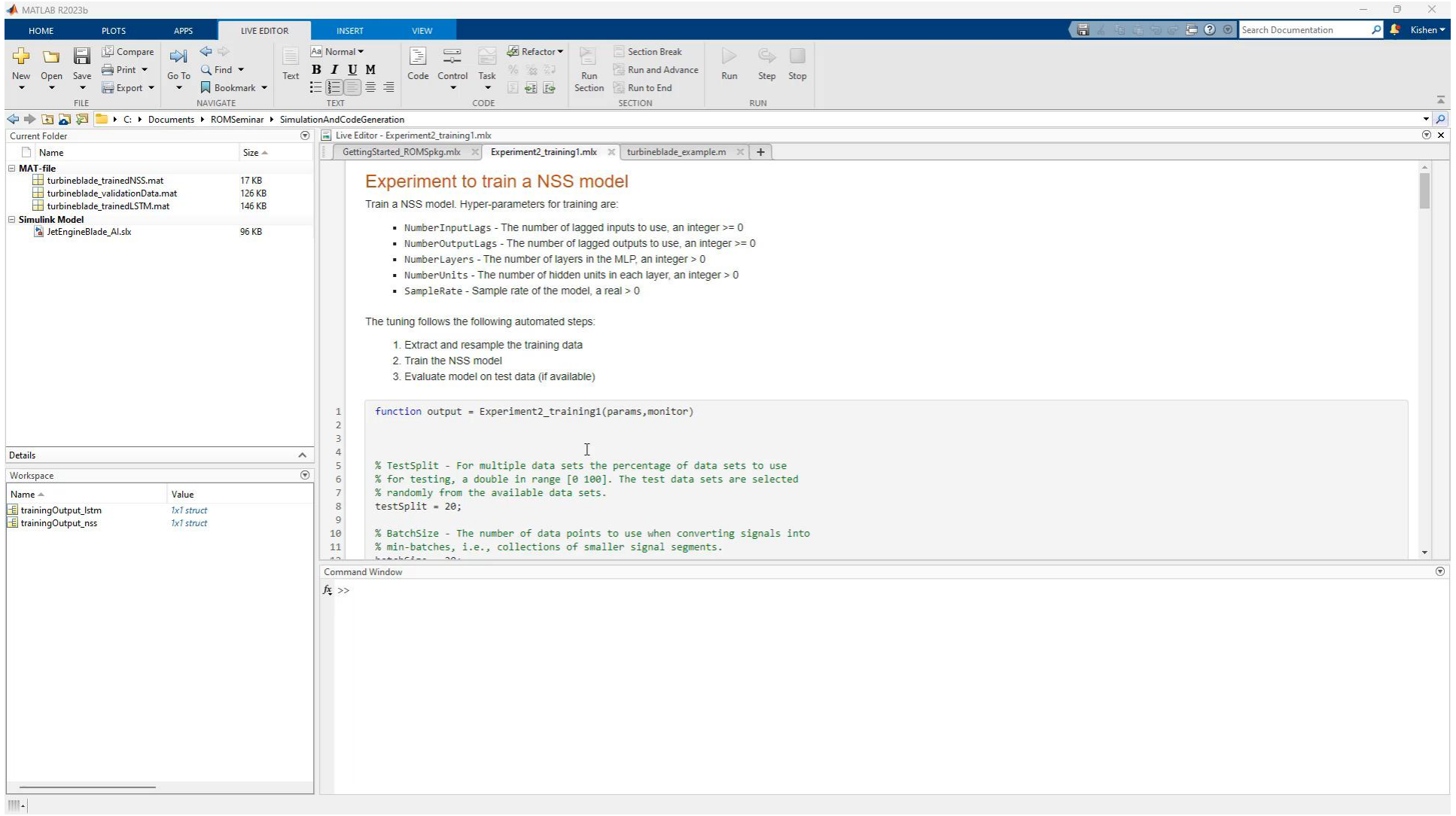
Deep Learning Toolbox



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

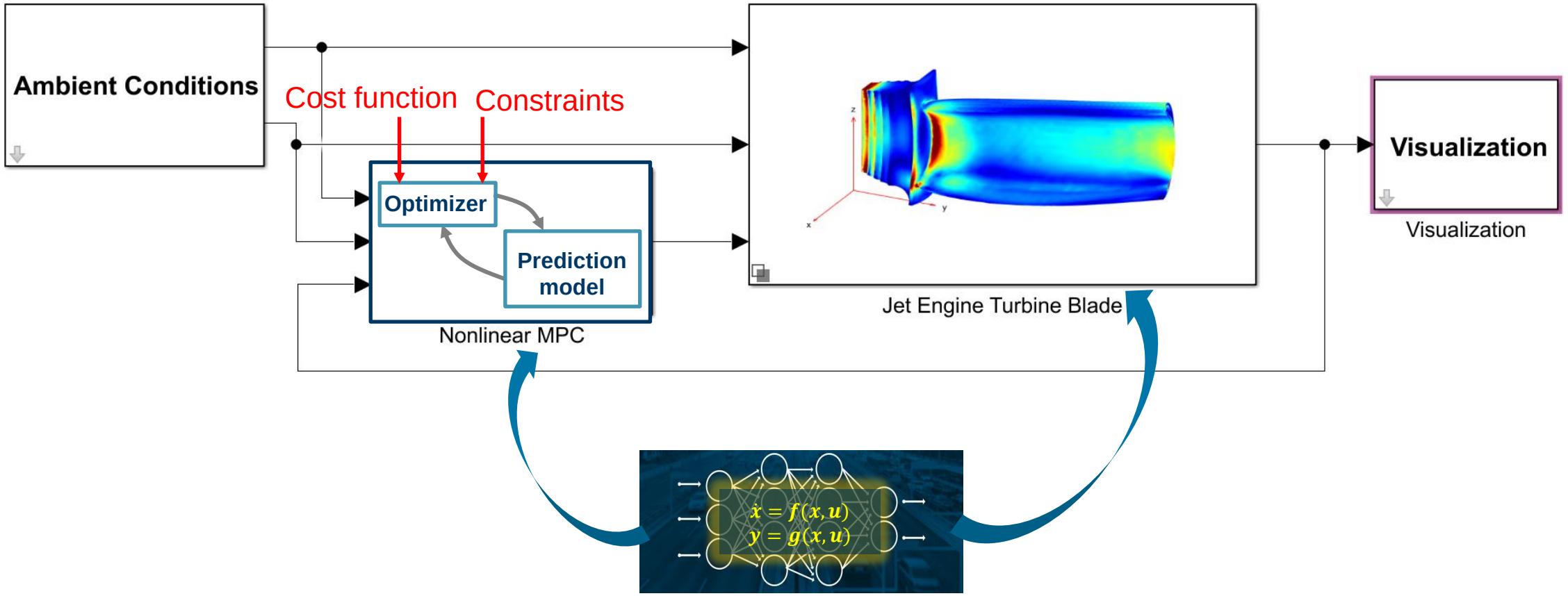
Simulink Profiler

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

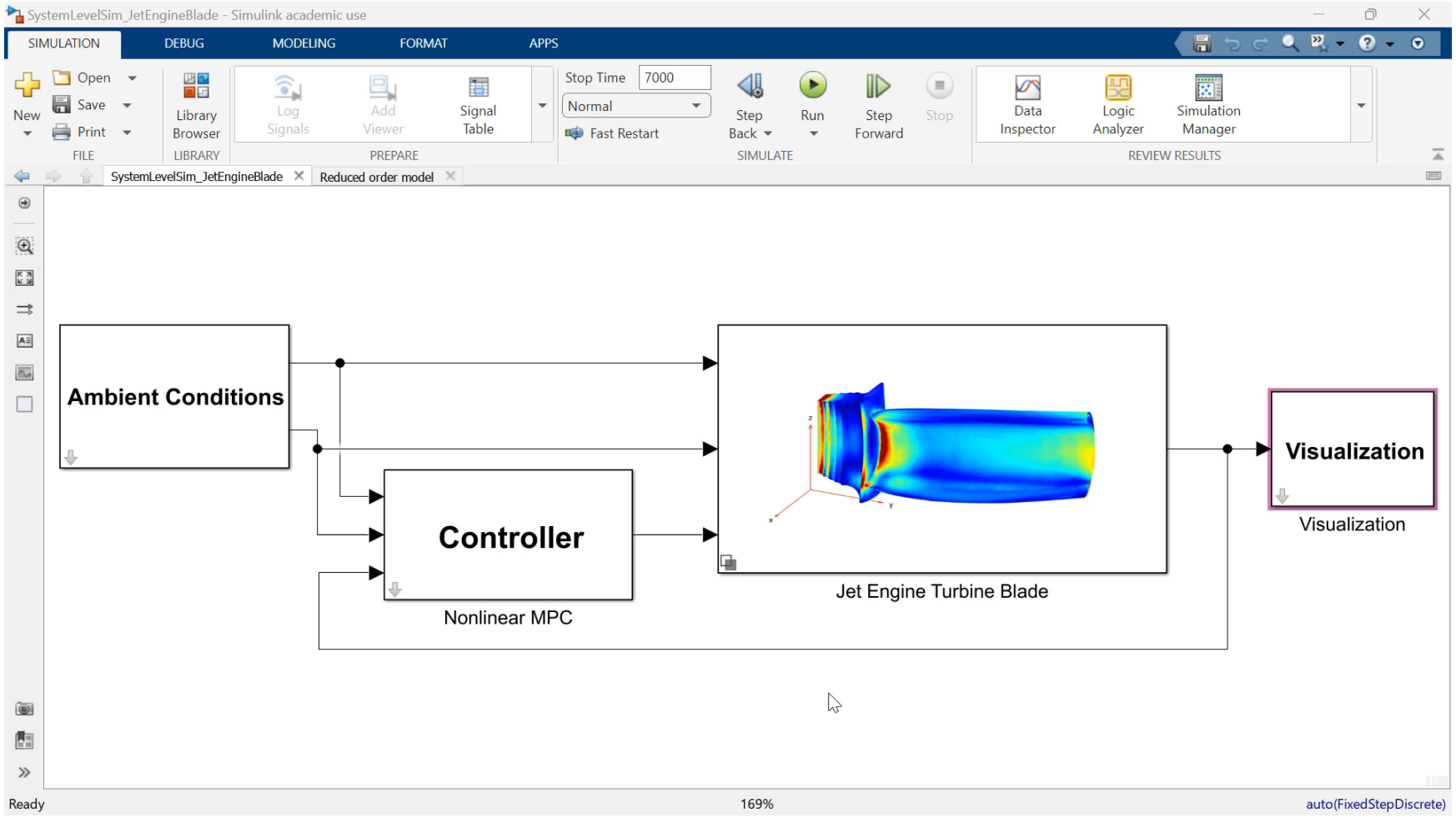
Neural state-space model is approximately 1e6x faster than the FEA model

Control Design with Model Predictive Controller

SIMULINK®

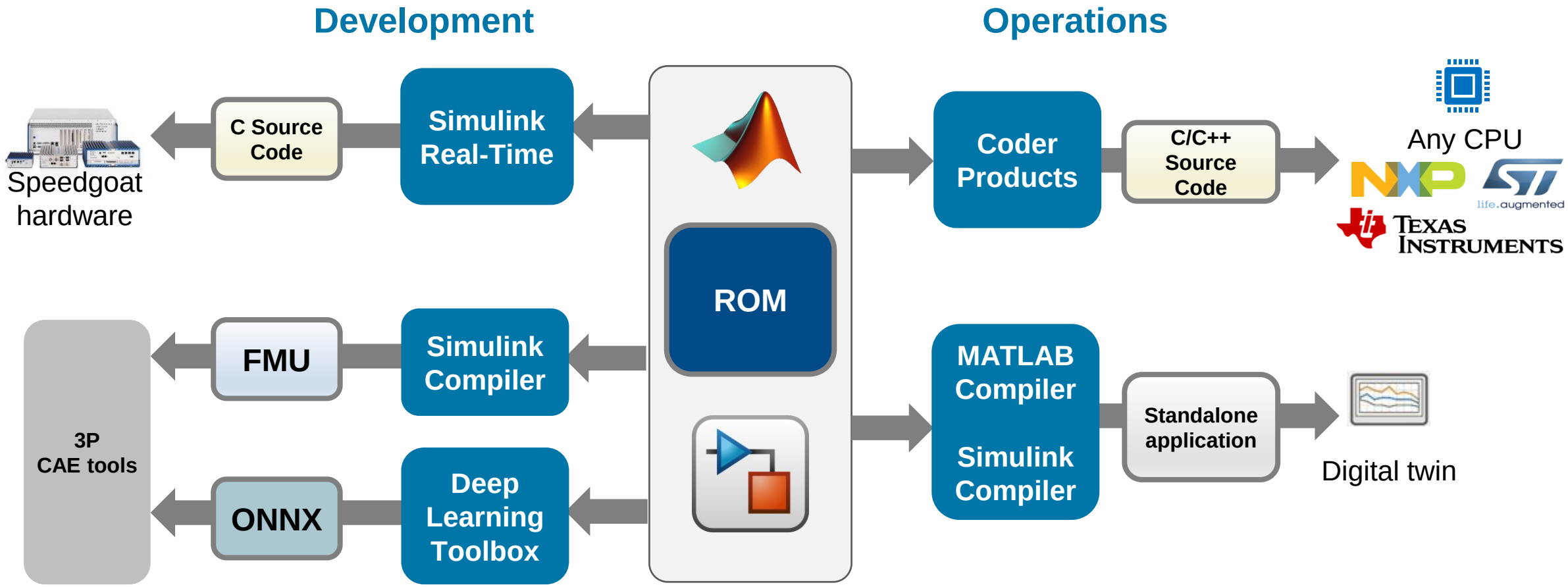


System-level simulation



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

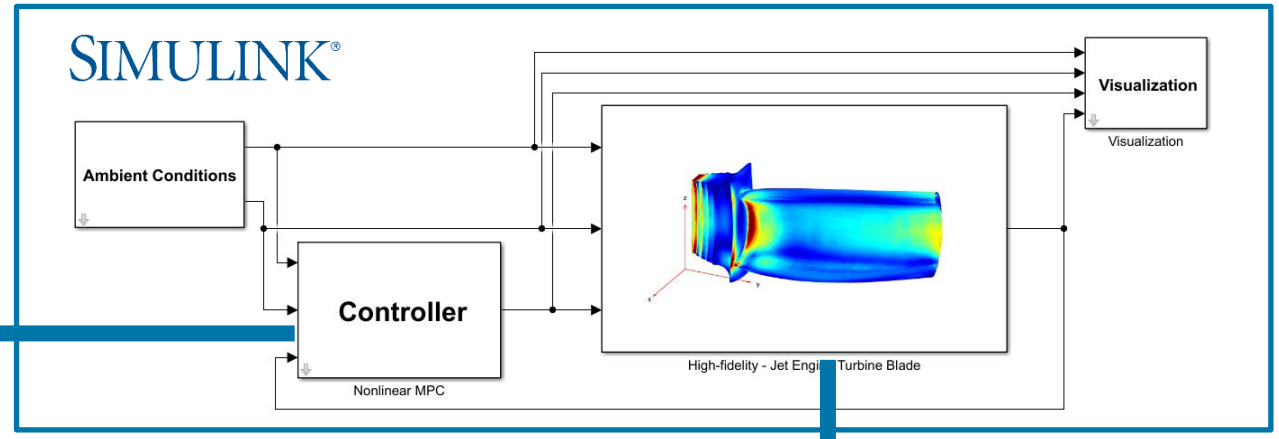
Use ROMs outside of Simulink, for development and operation stages



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

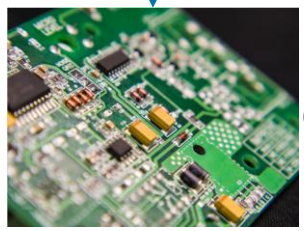
Hardware-in-the-loop simulation

System-level integration and test



Code generation from algorithm

Code generation from plant model



Target platform

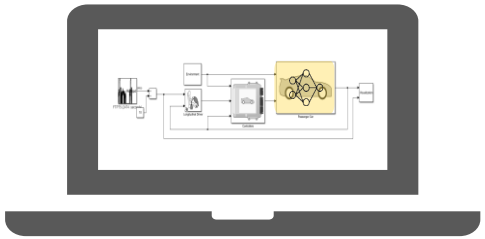
I/O

Real-time computer

Build & download models
Adjust parameters

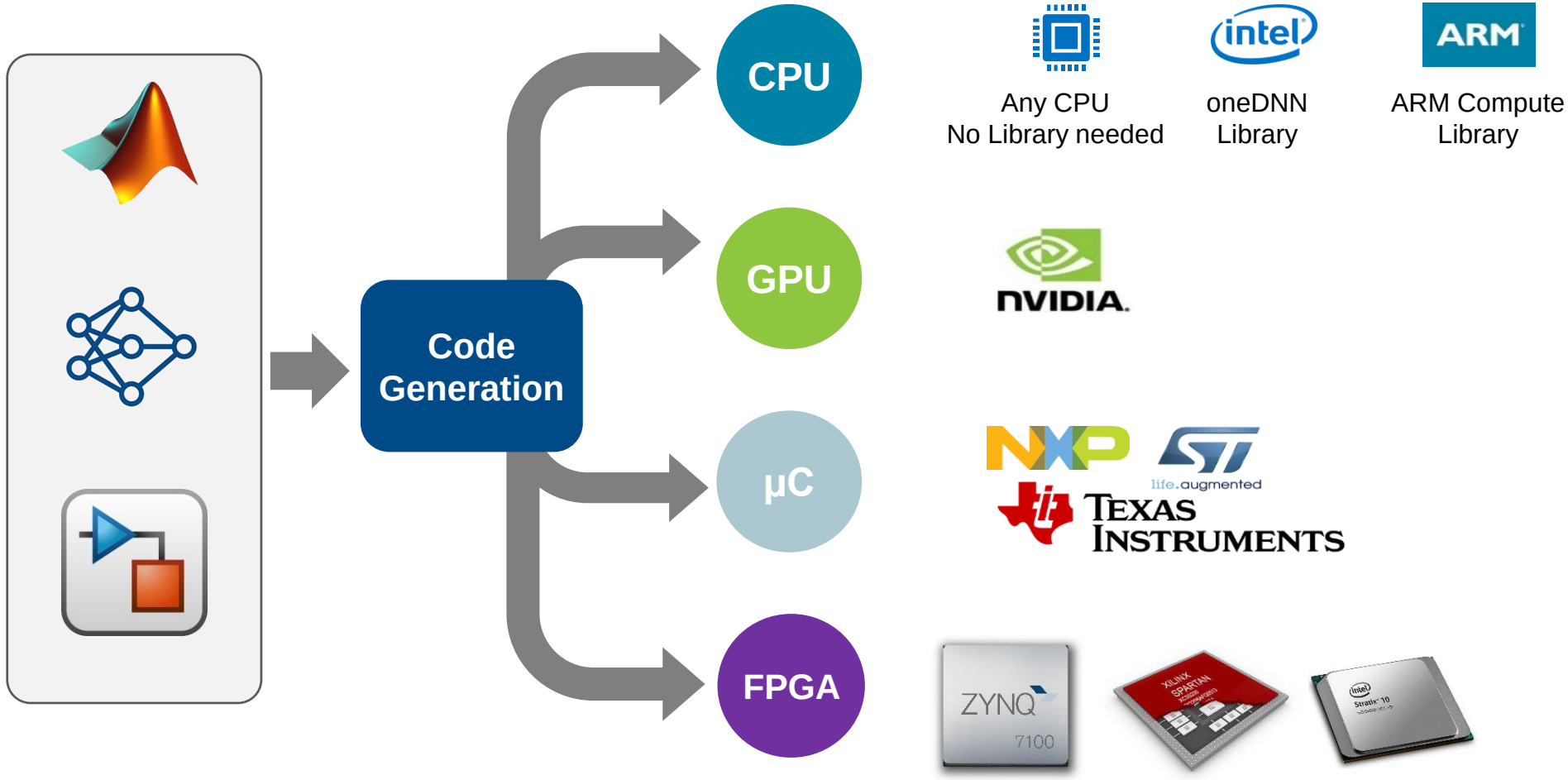
Monitor signals

SIMULINK



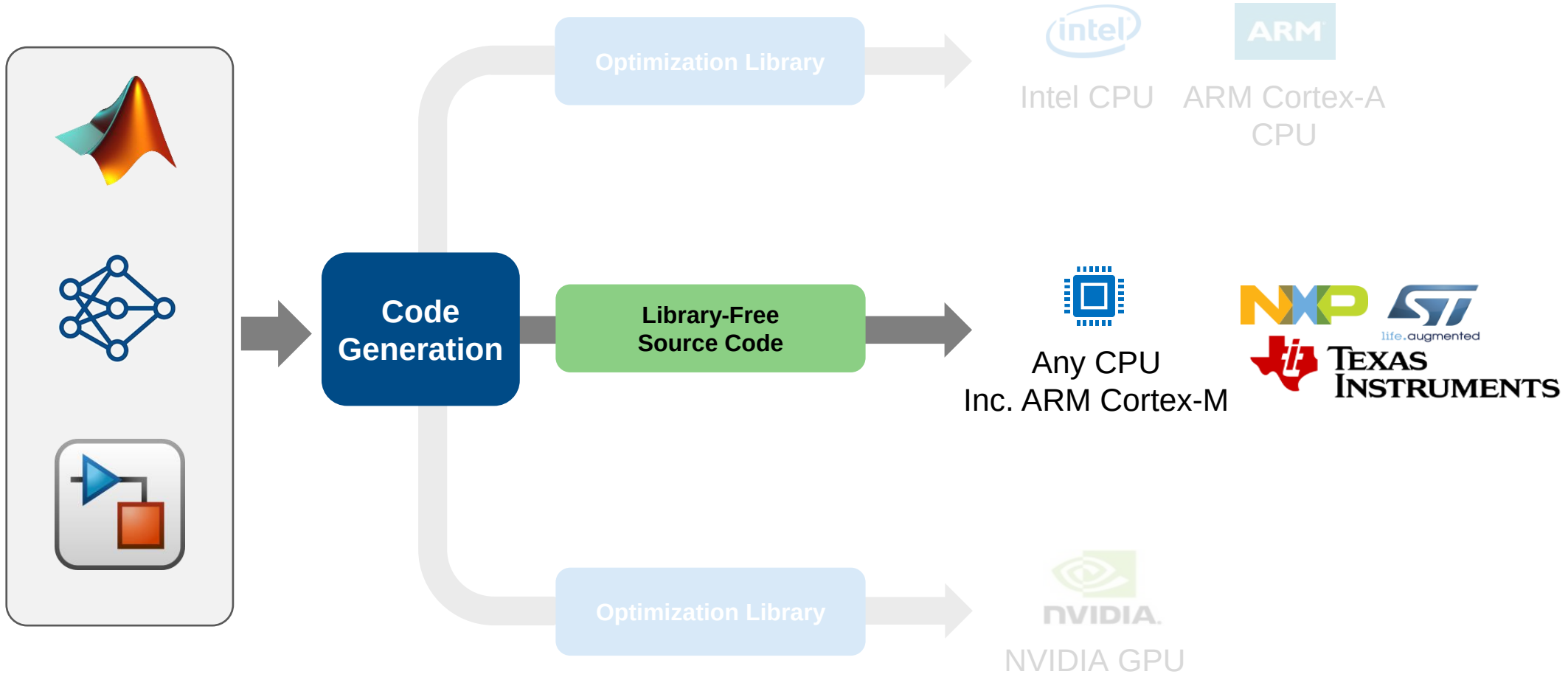
Data Preparation | AI Modeling | Simulation & Test | Deployment

Deploy to target with zero coding errors



Data Preparation | AI Modeling | Simulation & Test | Deployment

Generate Library-Free C/C++ Code for Deep Learning Networks



Data Preparation | AI Modeling | Simulation & Test | Deployment

AI workflow for engineered systems

Data Preparation

- Data pre-processing
- Feature engineering
- Simulation-generated data

AI Modeling

- Model design and tuning
- Python interoperability
- Explain models and predictions

Simulation & Test

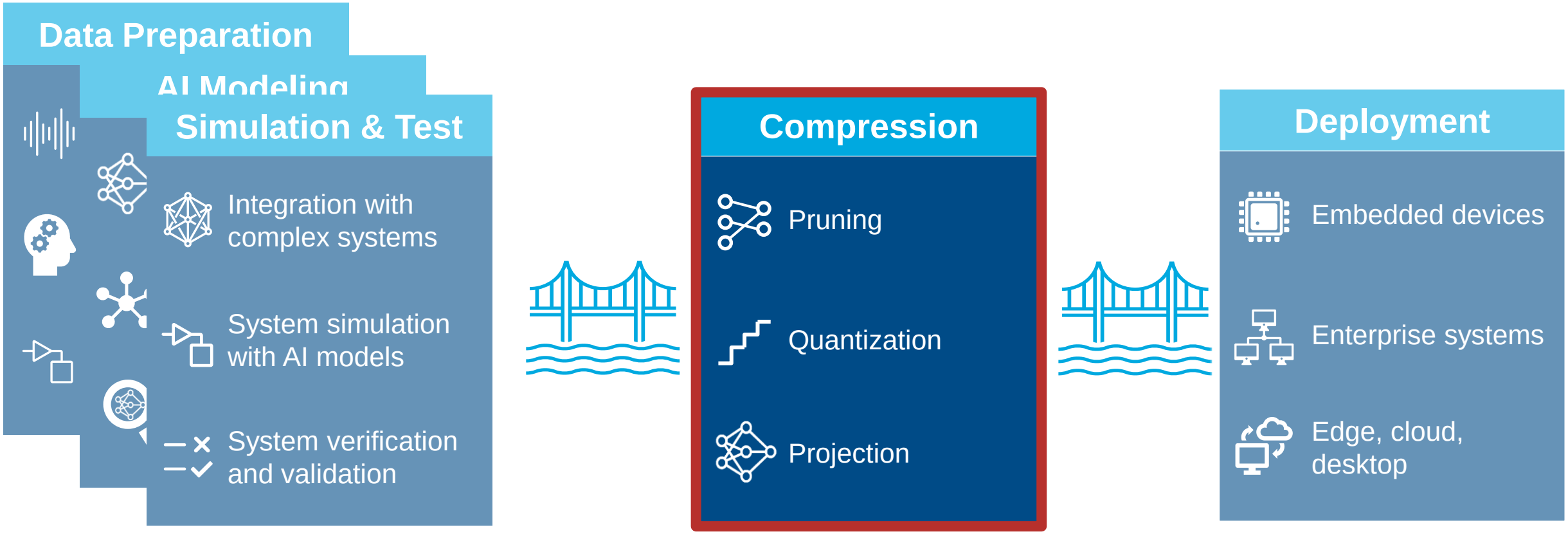
- Integration with complex systems
- System simulation with AI models
- System verification and validation

Deployment

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

32

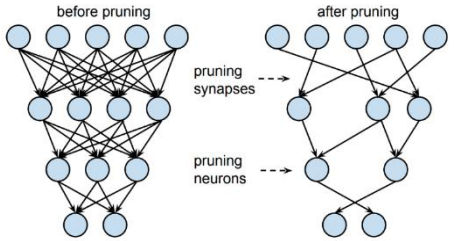
Model compression bridges the gap between design & deployment



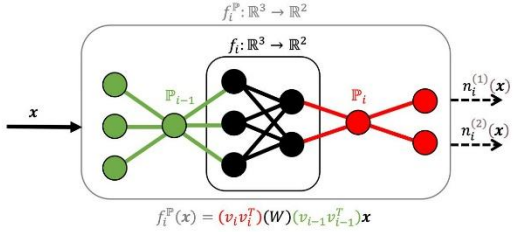
Data Preparation AI Modeling Simulatio Compression yment

Model Compression

Structural Compression

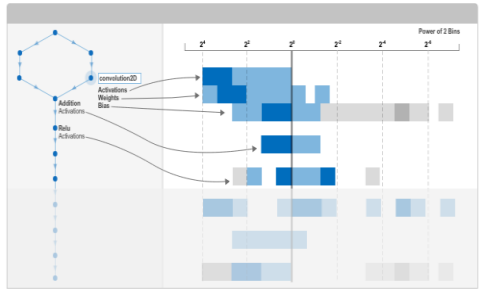


Pruning convolutional neural networks



Projection of deep neural networks

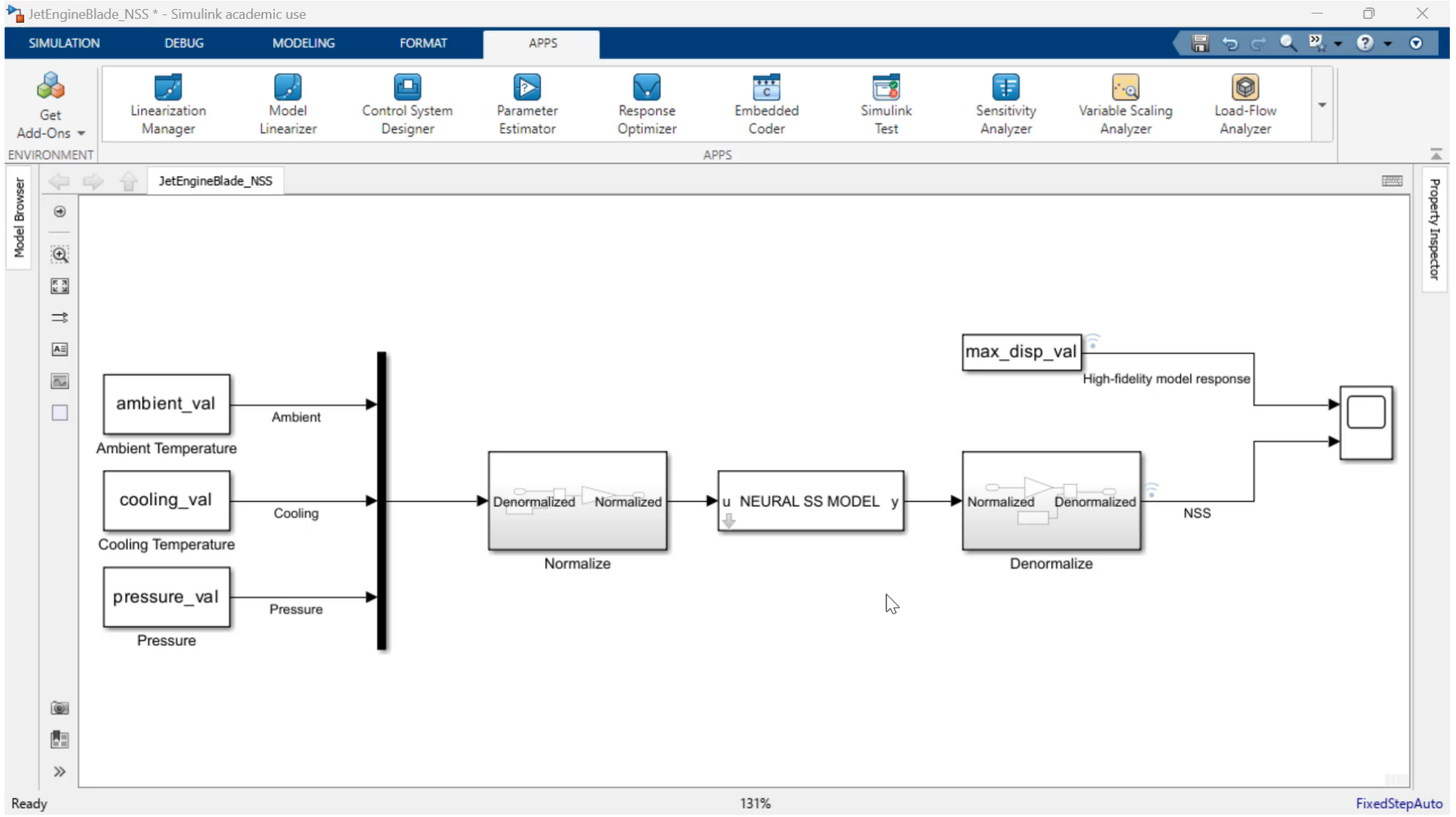
Datatype Compression



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

[Quantization, Projection, and Pruning]

Generate Library-Free C Code for Deep Learning Networks



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

Manage AI tradeoffs for your system

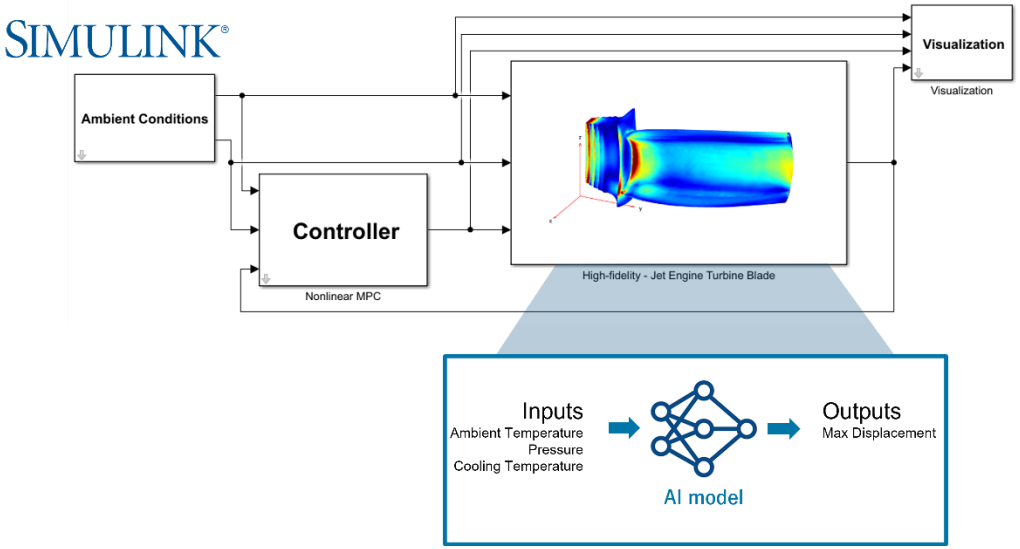
	LSTM Long Short-Term Memory Network	**Neural State Space** (Neural ODE)
Training Speed		
Interpretability		
Inference Speed		
Model Size		
Accuracy (RSME)		

Results are specific to Jet Engine Blade Example

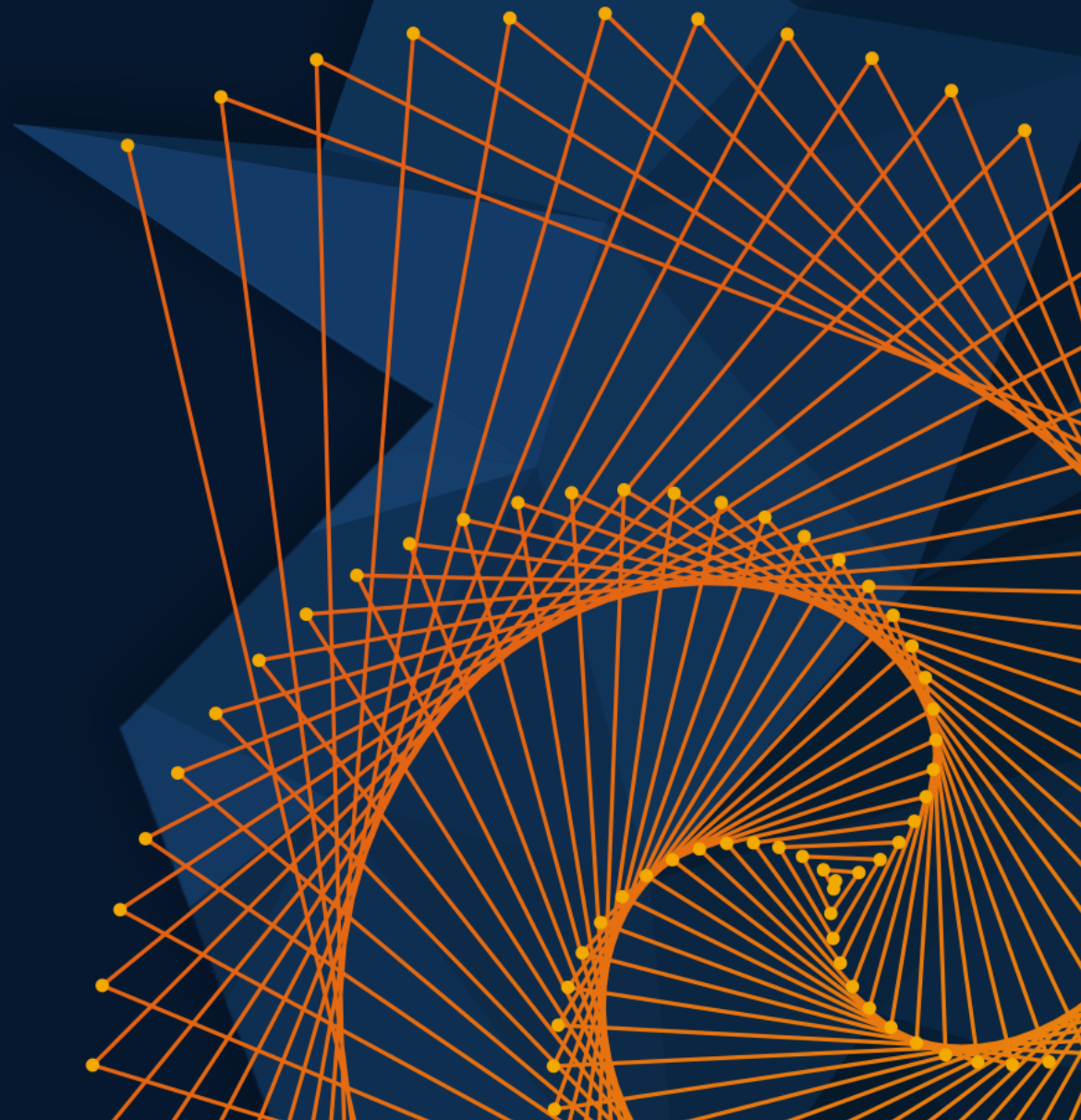
Key Takeaways

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 FEA model that computes tip displacement** of a jet engine blade
- **Integrate trained AI model into Simulink** for control design and system-level simulation
- **Generate C code and perform HIL tests**



MATLAB EXPO



MathWorks®