

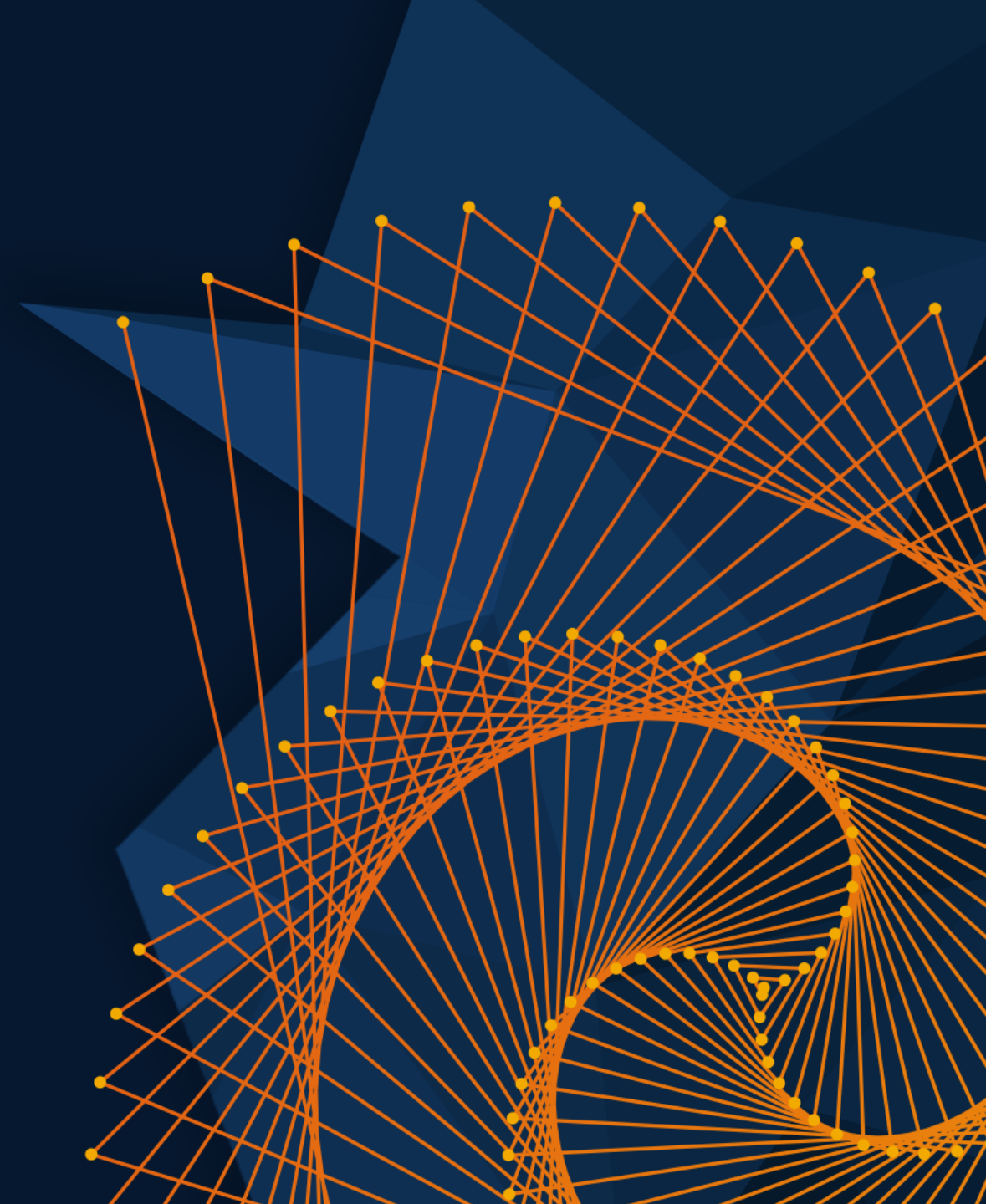
# MATLAB EXPO

2024.06.11 | 그랜드 인터컨티넨탈 서울 파르나스

---

## 차수축소 모델링 활용방안

김종남, 매스웍스 코리아

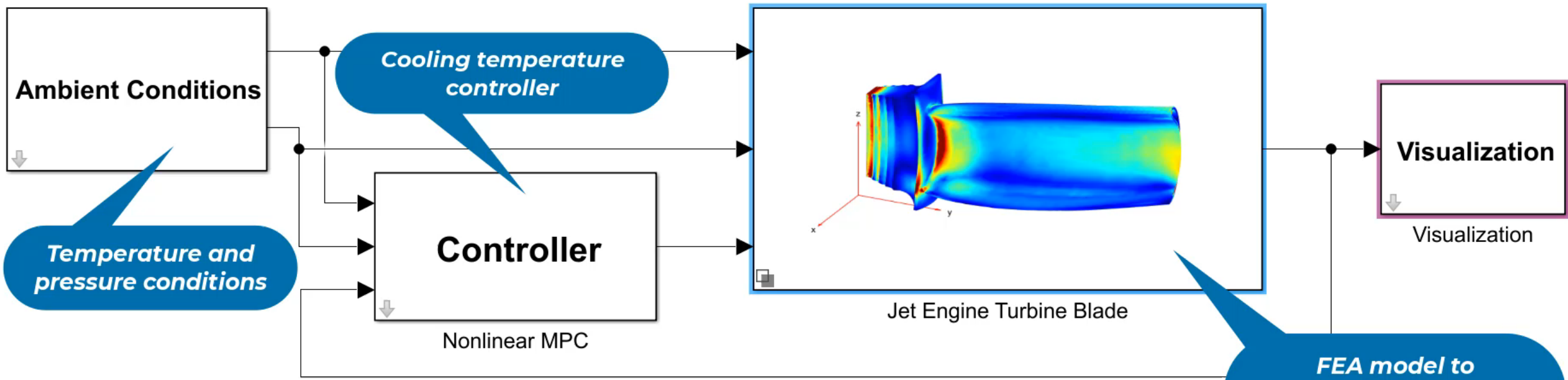


SIMULATION    DEBUG    MODELING    FORMAT    APPS    **VARIANT SUBSYSTEM**

Open Save Print Library Browser Log Signals Add Viewer Signal Table Stop Time 7000 Normal Fast Restart Step Back Run Step Forward Stop Data Inspector Logic Analyzer Simulation Manager

FILE    LIBRARY    PREPARE    SIMULATE    REVIEW RESULTS

SystemLevelSim\_JetEngineBlade x Reduced order model x



*Temperature and pressure conditions*

*Cooling temperature controller*

*FEA model to compute maximum tip displacement*

## Common challenges

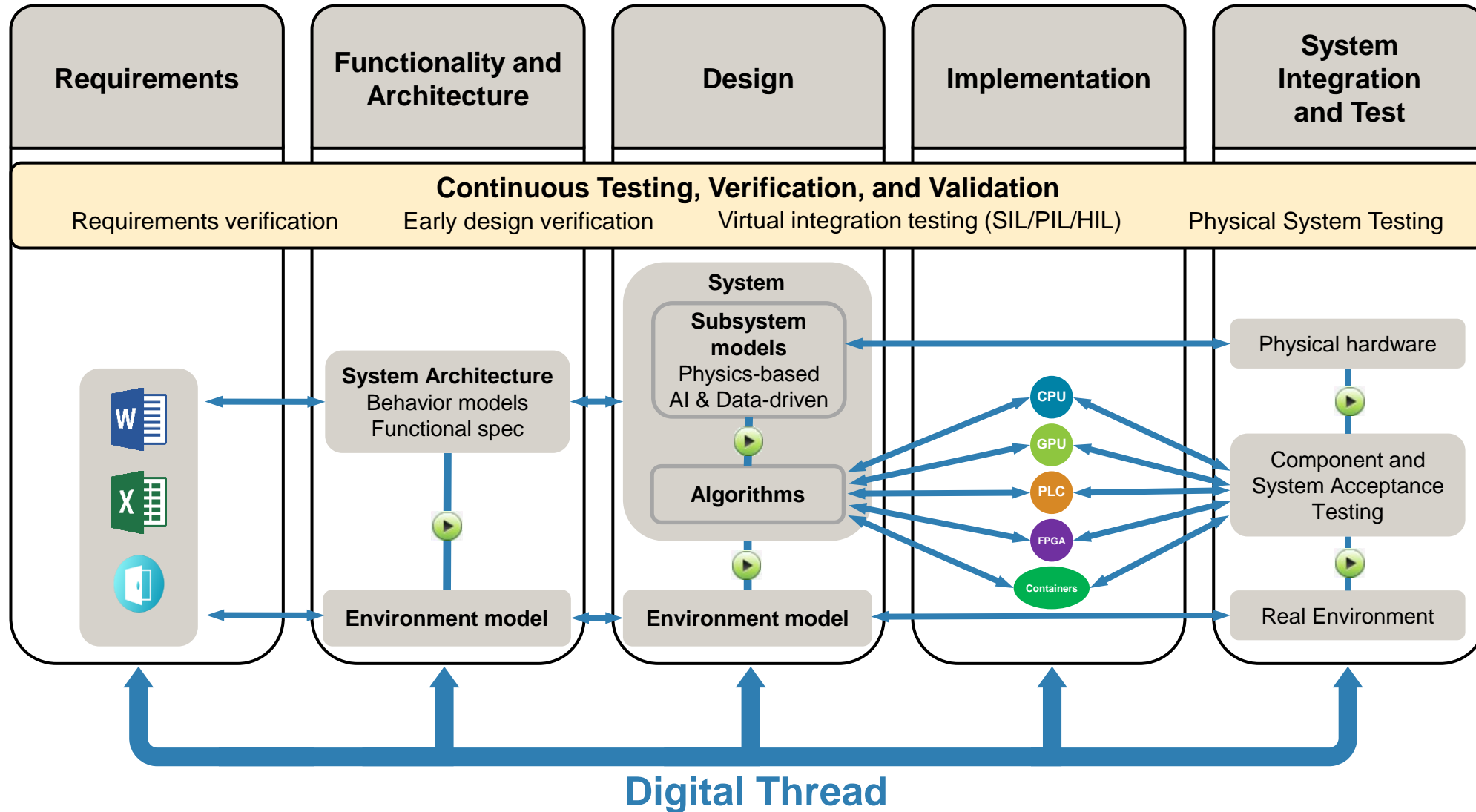


High fidelity models, such as ones from 3<sup>rd</sup> party FEA/CFD tools, are too slow for system level simulation, control design, and HIL testing.

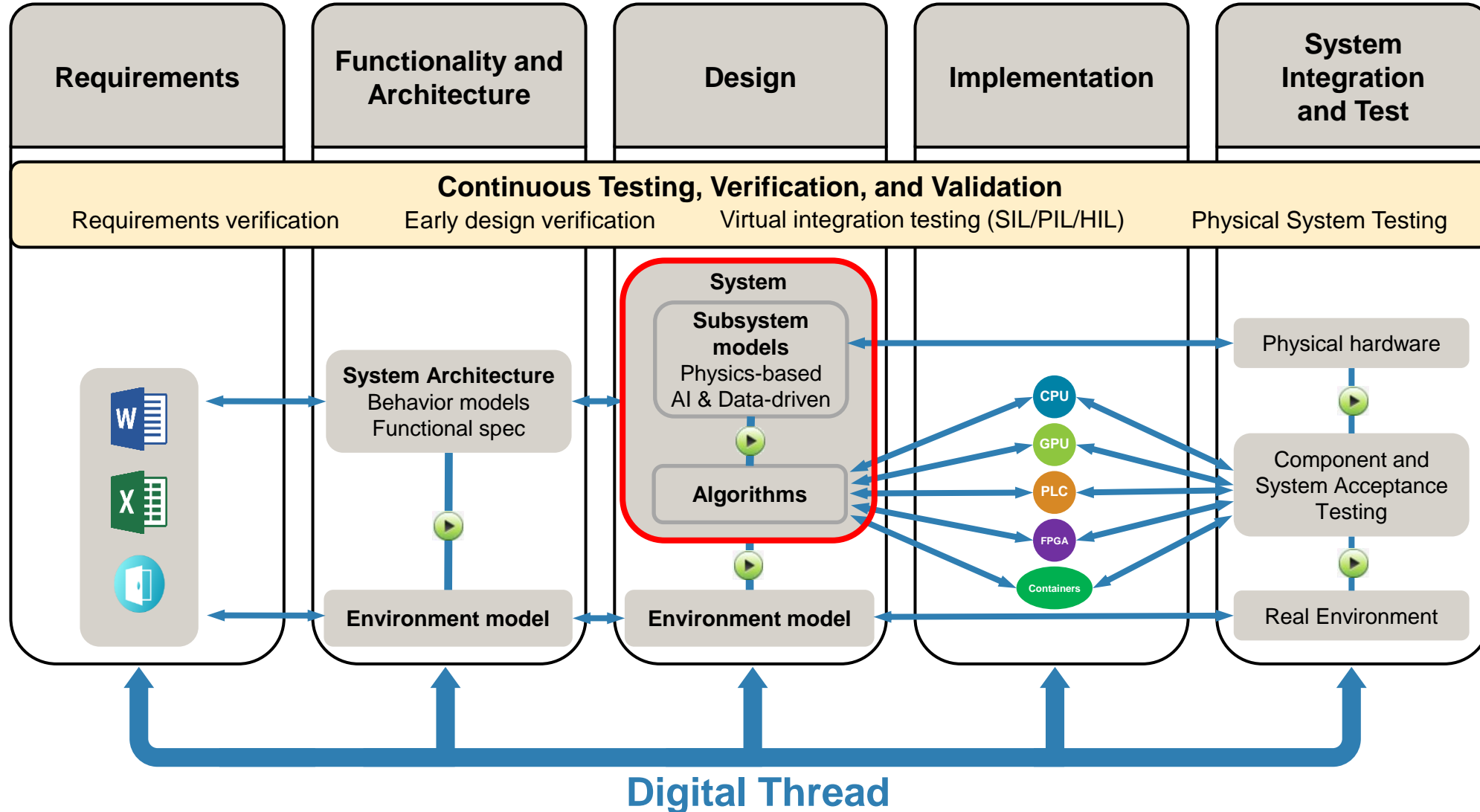


Creating a ROM that produces desired results in terms of speed, accuracy, interpretability, etc.

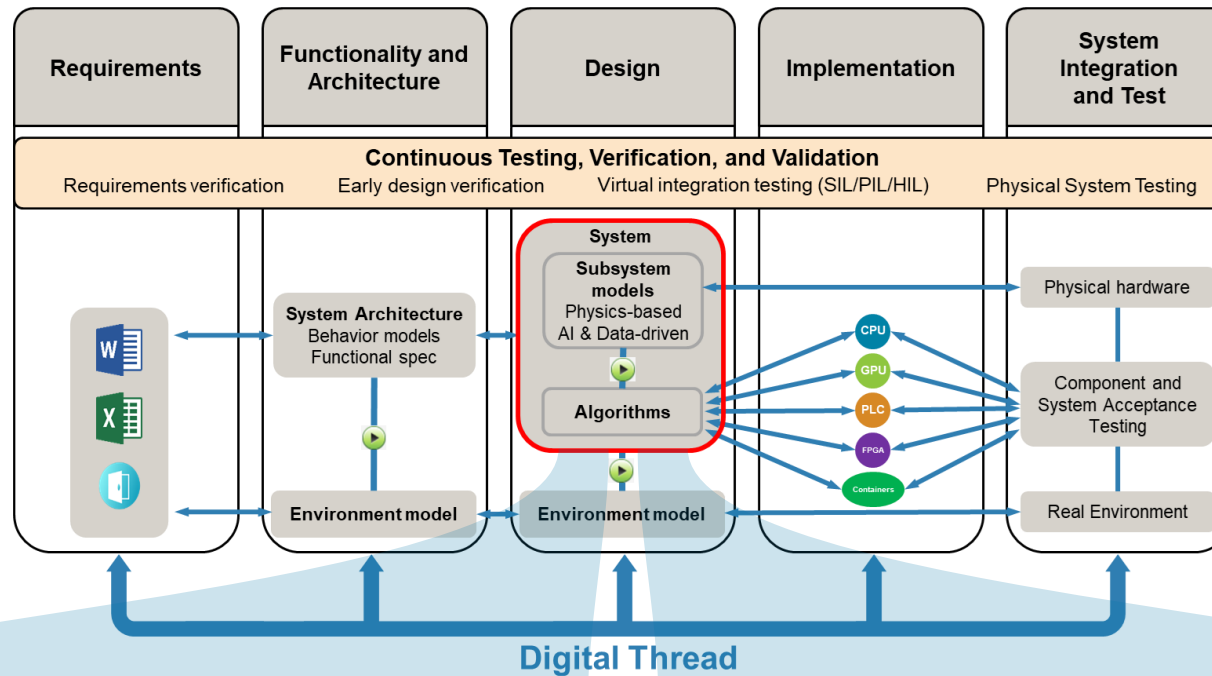
# Model-Based Design



# Integrating AI into Model-Based Design



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



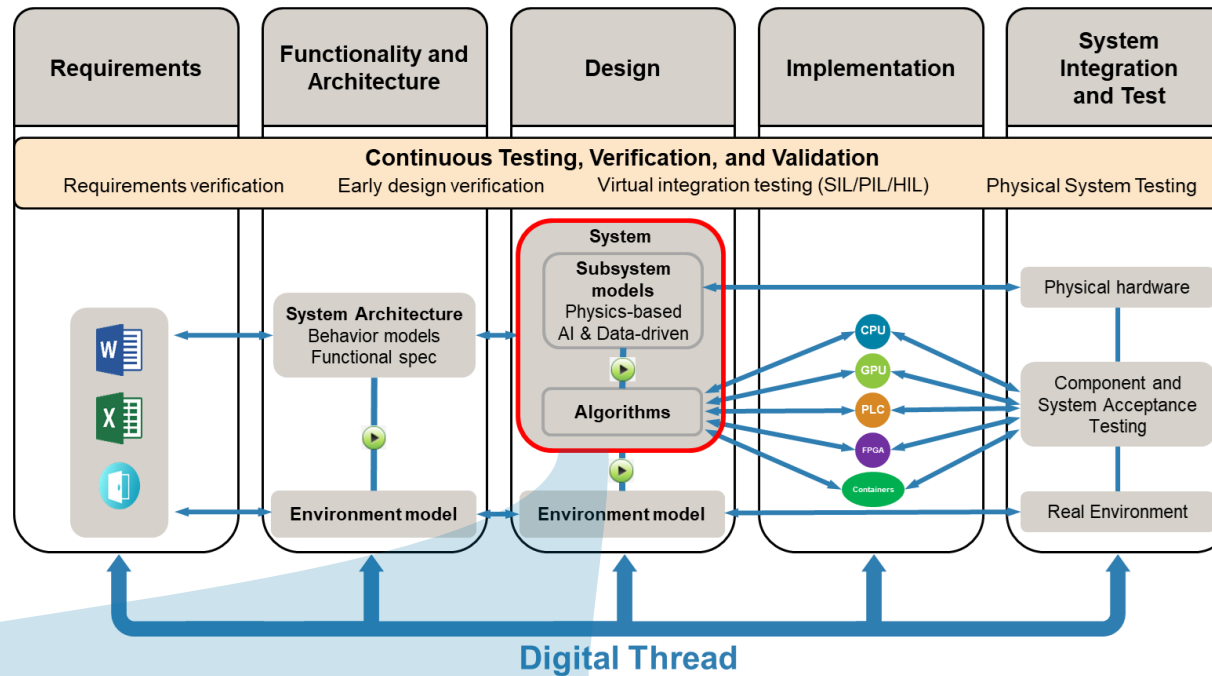
## AI for component modeling

- 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

# Focus today



## 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

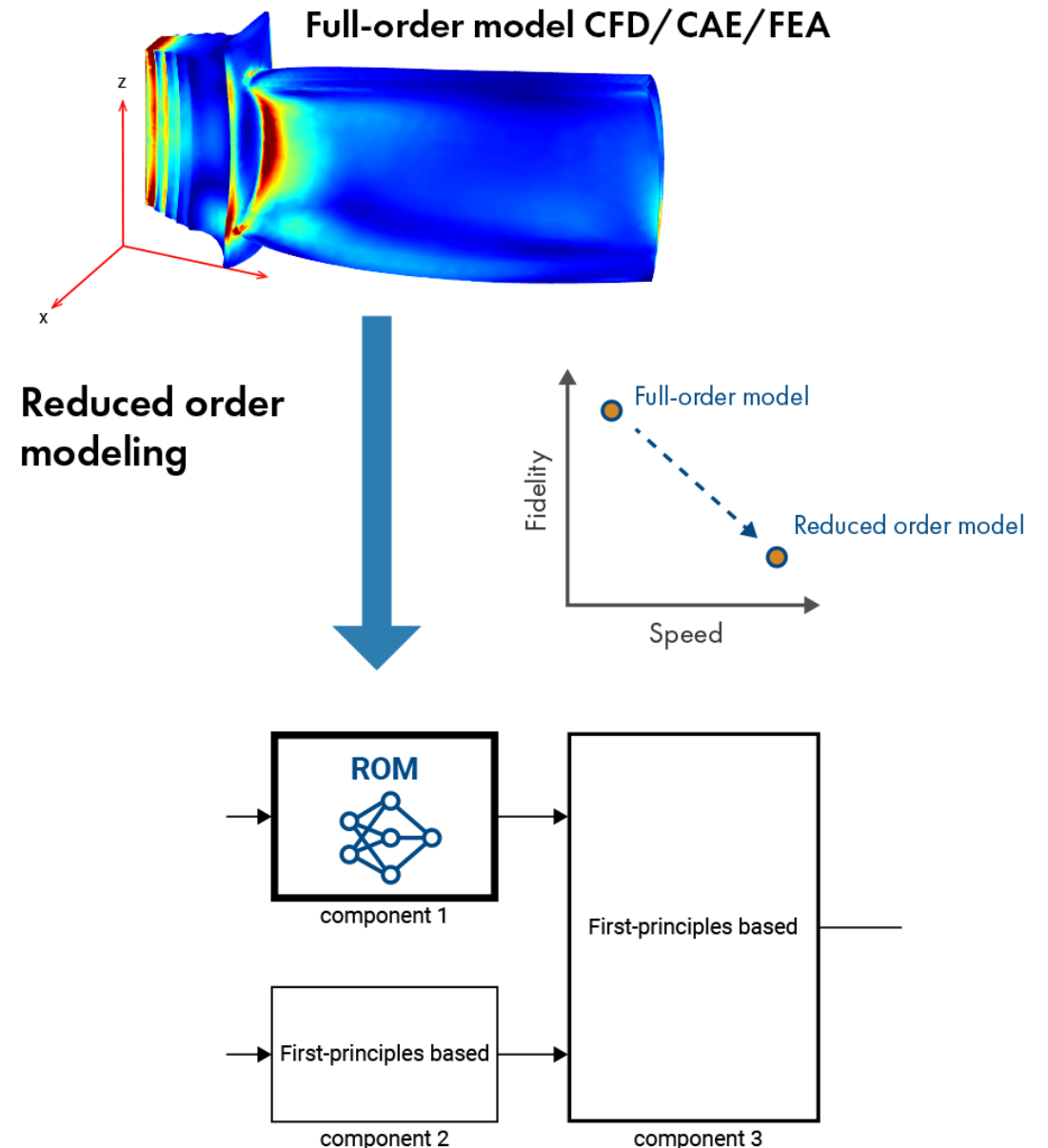
# Reduced Order Modeling

## 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
- Develop virtual sensors, Digital twins
- Enable desktop simulations for orders-of-magnitude longer timescales



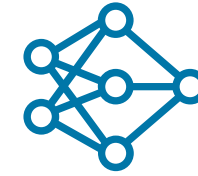


# Reduced Order Modeling

How

AI-Based  
Data-driven

Inputs  
Ambient Temperature  
Ambient Pressure  
Cooling Temperature



AI model



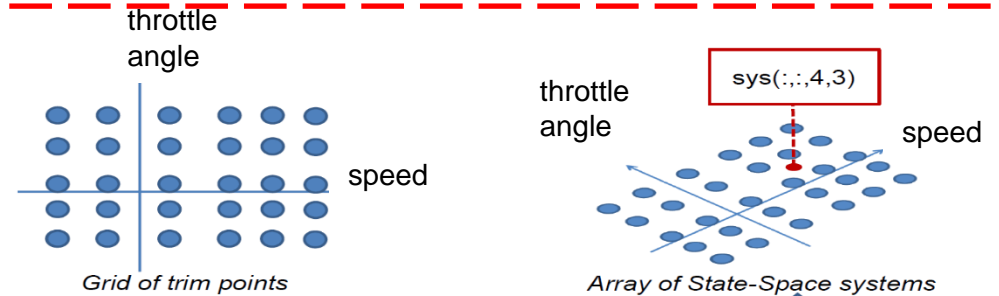
Outputs  
Max Displacement

*focus today*

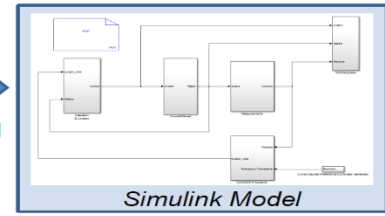
Reduced order  
model

Linearization

Model-based



Loop through the grid  
of trim points



Identify local model at  
each trim point

FEA  
Software

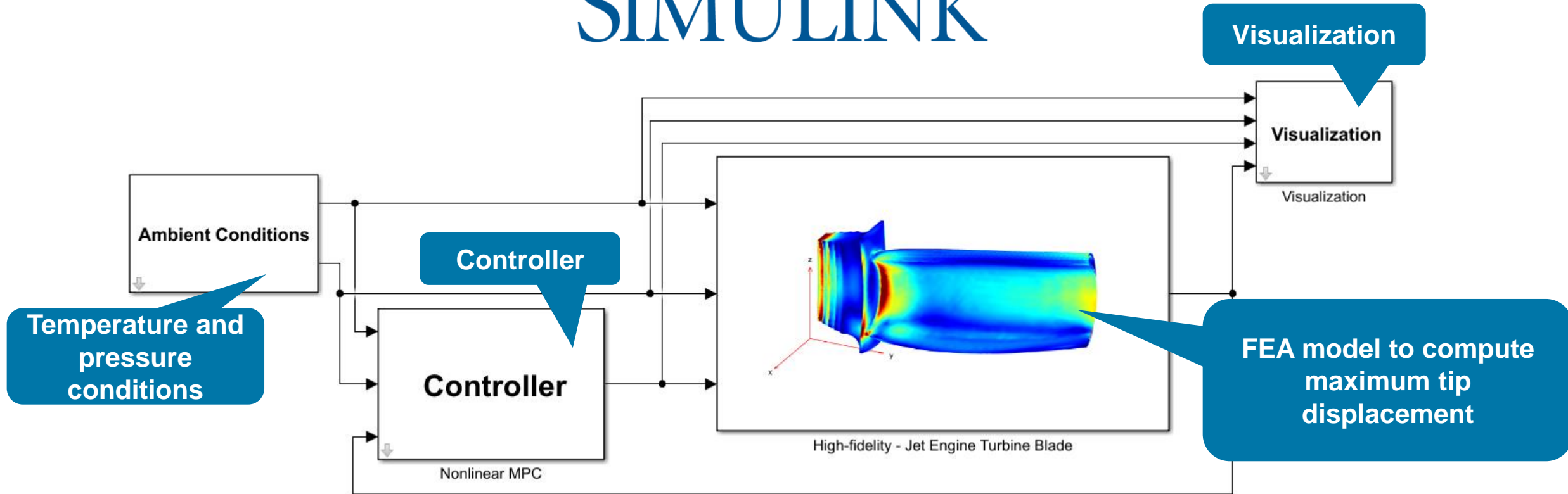


Simulink  
Simscape Multibody  
Control System Toolbox

# Example overview

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

# SIMULINK®

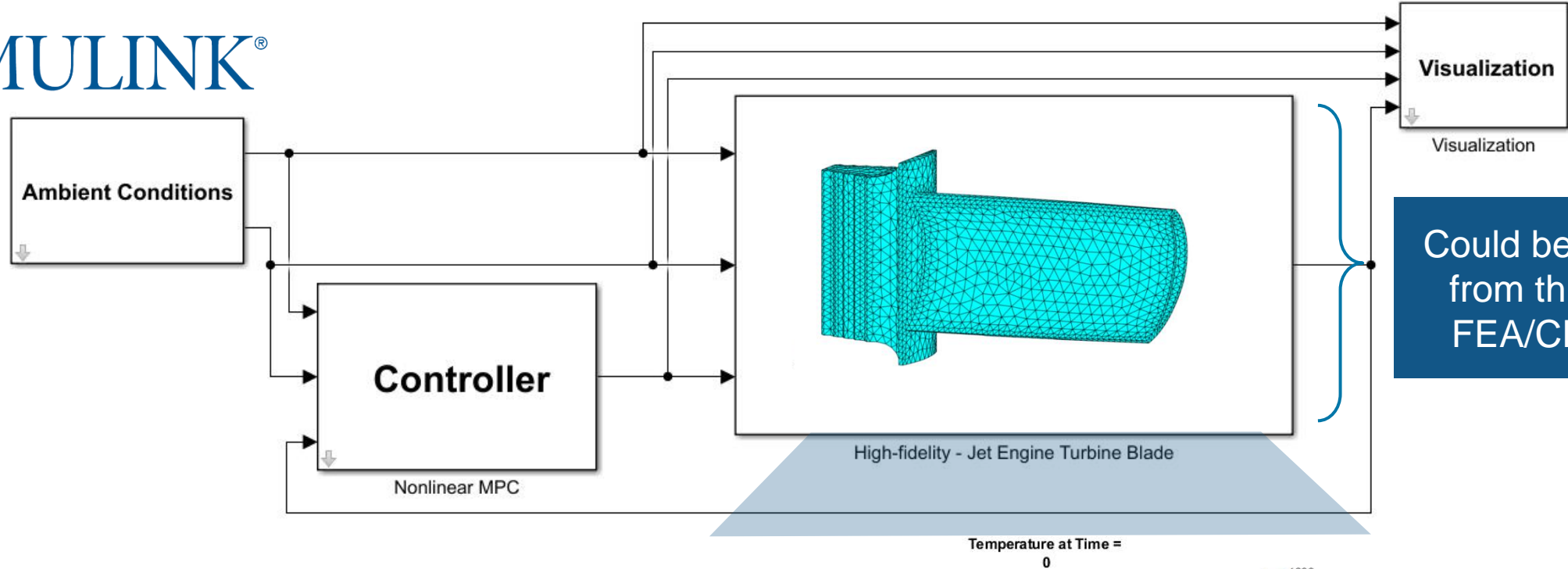


## Closed-loop temperature control

# Example overview

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

SIMULINK®

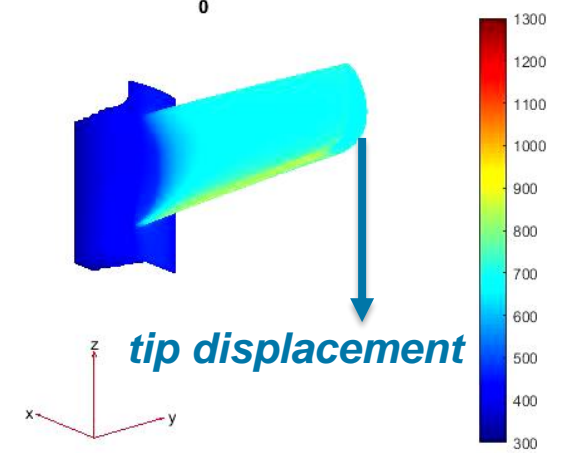


Could be imported from third-party FEA/CFD tools

~30 seconds per time step for solving FEA models



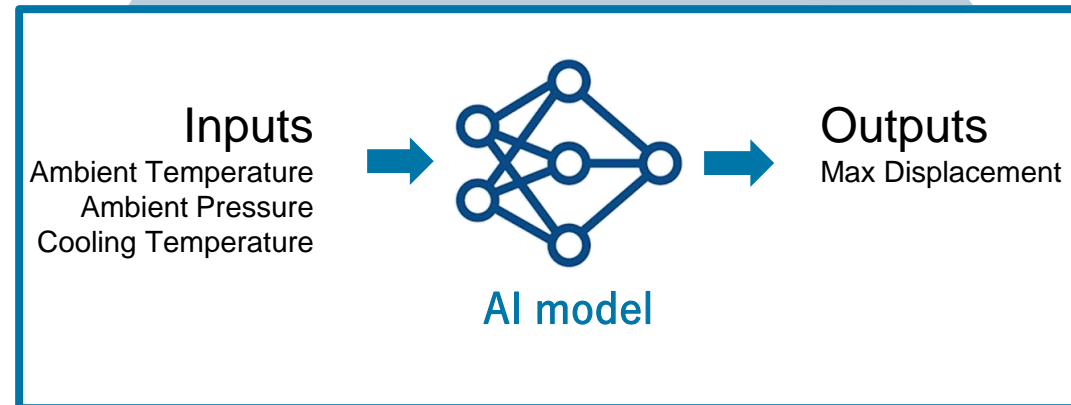
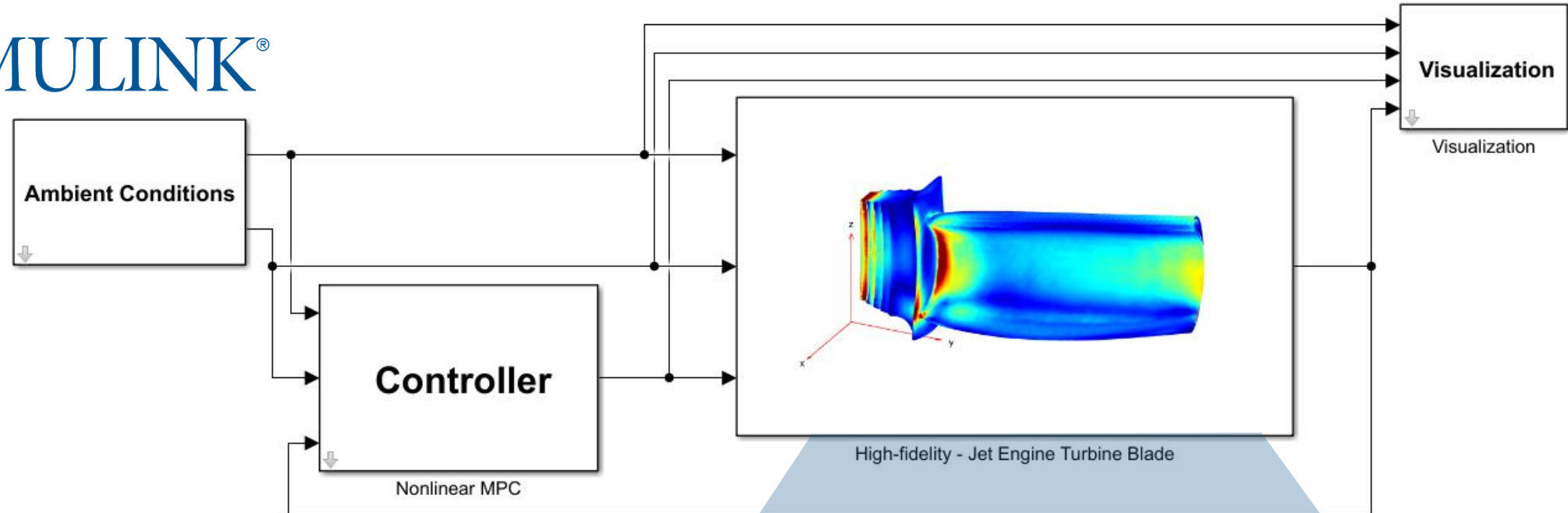
Not suitable for control design and HIL testing



# 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

The image shows a screenshot of the MathWorks website for 'Reduced Order Modeling with MATLAB and Simulink' and a screenshot of the 'Reduced Order Modeler' software interface. The software interface includes a toolbar with 'LSTM network', 'Nonlinear ARX', and 'Neural State Space' options highlighted in a red box. It also features a 'Design of Experiments' (DoE) section with a table of experiments and a 'Plot Signals' section with a table of signal ranges.

| Signal     | Min    | Max    |
|------------|--------|--------|
| 1 Ambient  | 800    | 2000   |
| 2 Cooling  | 50     | 250    |
| 3 Pressure | 450000 | 550000 |

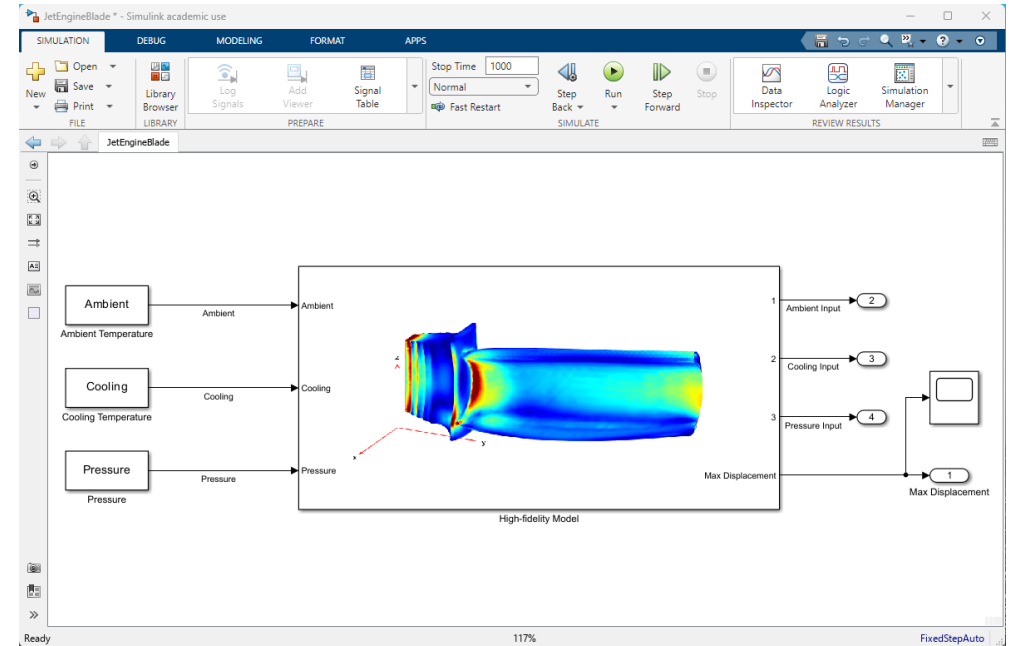
  

The diagram illustrates the 'Reduced Order Modeling' process. It starts with a 'Full-order model CFD/CAE/FEA' (represented by a 3D model of a turbine). This leads to 'Reduced Order Modeling', which produces a 'ROM component 1' (represented by a neural network icon). This ROM component is then integrated with 'First-principles based component 2' and 'First-principles based component 3' to form a complete system. A graph shows 'Fidelity' on the y-axis and 'Speed' on the x-axis, with a dashed arrow pointing from a high-fidelity 'Full-order model' to a lower-fidelity 'Reduced order model'.

# Generate data for training



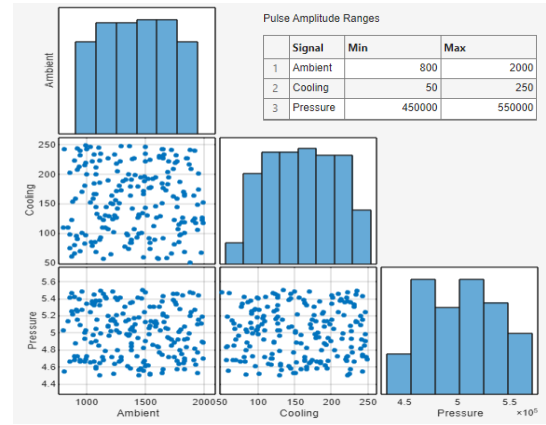
Physical system



Simulink/Simscape

# Synthetic Data Generation

## Design of Experiments

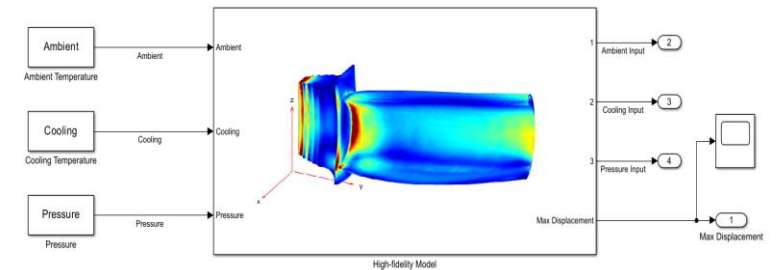
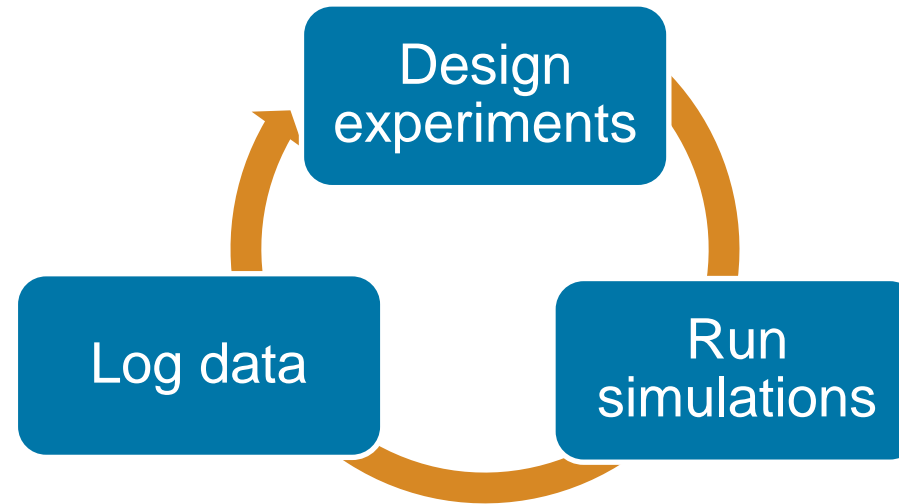


### Input features

Ambient Temperature  
Ambient Pressure  
Cooling Temperature

### Response

Max Displacement



Data Preparation

AI Modeling

Simulation & Test

Deployment

# Synthetic Data Generation

## Design of Experiments

**Getting Started with Reduced Order Modeling Support Package**

**What Is Reduced Order Modeling?**  
 Reduced order modeling is a technique for simplifying full order high-fidelity models by reducing their computational complexity, while preserving their dominant behavior. Working with a reduced order model (ROM) can simplify analysis and control design.

**Why Use Reduced Order Modeling?**  
 Using reduced order modeling techniques, you can:

- **Enable use of 3rd party FEA/FEM/CFD models for system-level simulation in Simulink® including hardware-in-the-loop testing** — You can combine multiple complex component-level models, including third-party finite element method (FEM) or finite element analysis (FEA) models, into system-level simulation models in Simulink by replacing the complex models with the corresponding ROMs. ROMs are also useful for hardware in-the-loop testing as they allow real-time simulations. Engineers can create ROMs representing the physical components of the system, which can run on a real-time machine for testing of the control algorithm on embedded hardware. The reduced computational complexity of ROMs make such testing more feasible.
- **Create virtual sensors** — You can use ROMs as virtual sensors for estimating or predicting signals of interest when measuring those signals by using a physical sensor is impractical or impossible.
- **Perform control design** — The reduced complexity of ROMs can make control design tasks more tractable. You can design your controller for the reduced order model of a plant and then validate the controller on the original high-fidelity system. You can also use ROMs for control algorithms that require internal prediction models, such as nonlinear model predictive control.
- **Create digital twins** — You can create or simplify digital twin models using ROMs. Doing so makes the digital twins more computationally efficient and more suitable for periodic updates to represent the current state of the operational asset.

**Reduced Order Modeler App Workflow**  
 The general workflow of the Reduced Order Modeler app involves logging data from a Simulink model and using that data to train a ROM. It includes the following steps.

```

  graph LR
    A[Open Model and App] --> B[Select Inputs and Outputs]
    B --> C[Specify Experiments]
    C --> D[Run Model]
    D --> E[Create Reduced Order Model]
    E --> F[Export Model]
    F --> G[Replace Blocks in Simulink Model]
  
```

Command Window  
 >>

Data Preparation

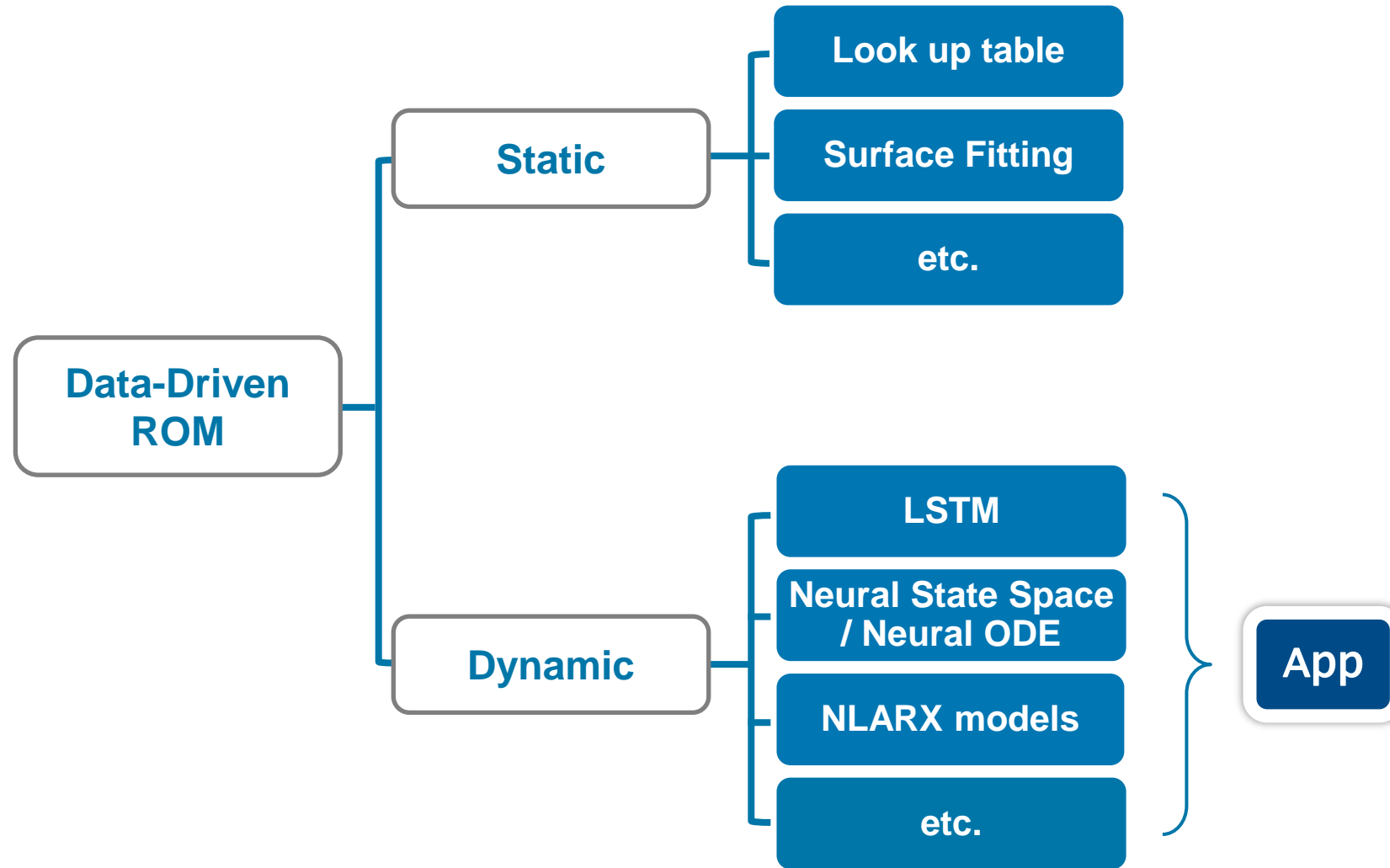
AI Modeling

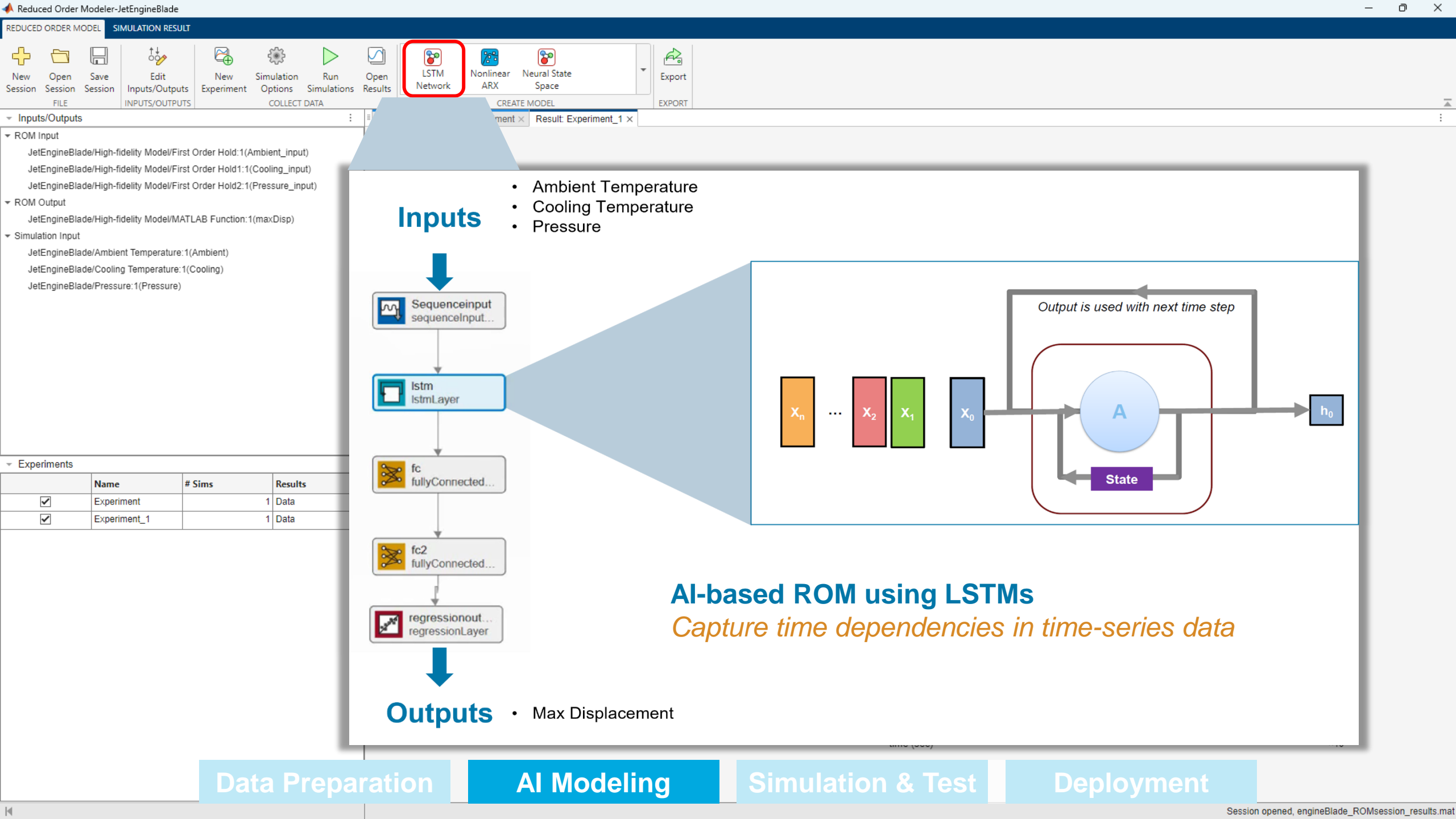
Simulation & Test

Deployment



# Data-driven ROM





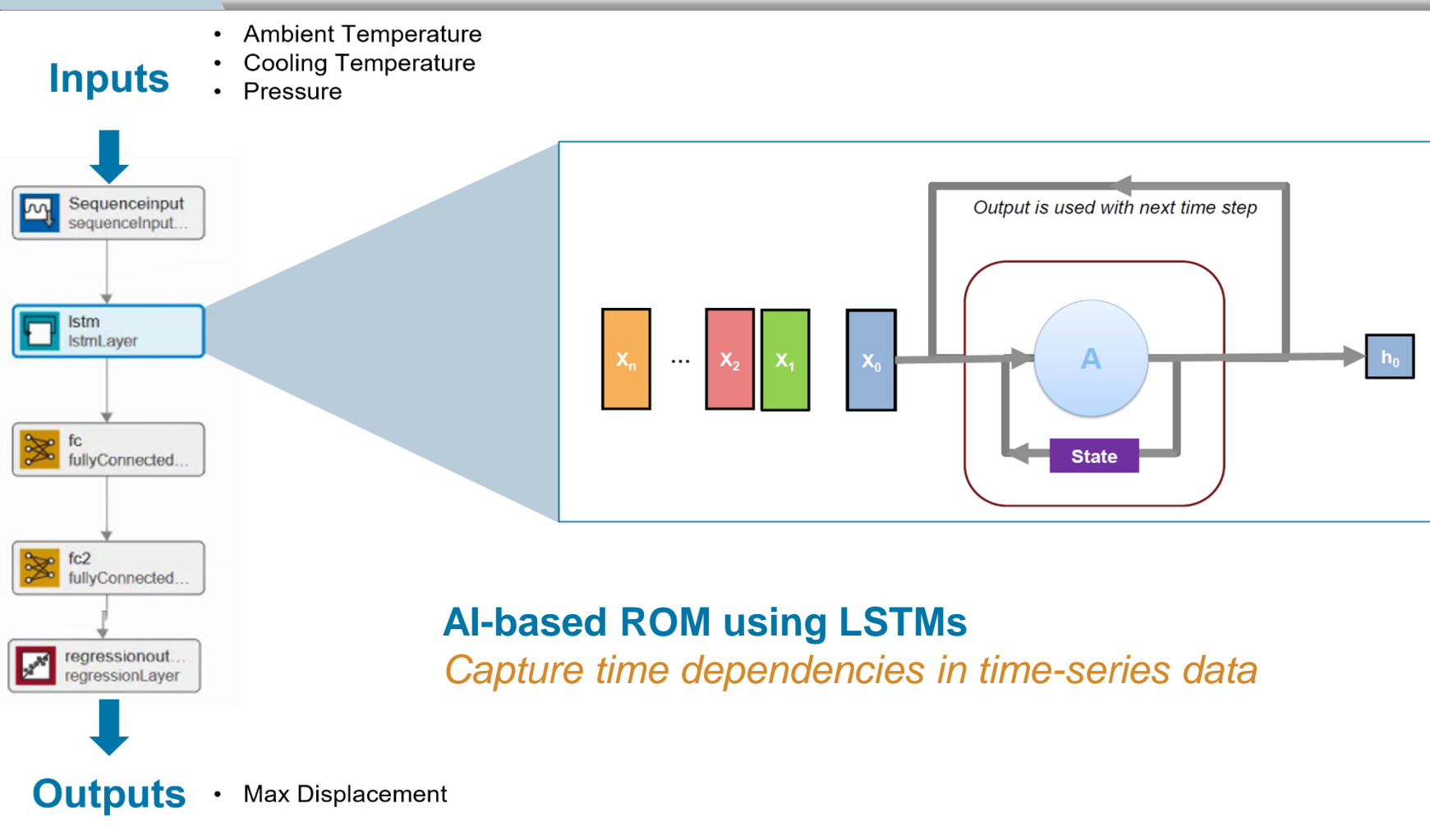
Toolbar icons: New Session, Open Session, Save Session, Edit Inputs/Outputs, New Experiment, Simulation Options, Run Simulations, Open Results, LSTM Network, Nonlinear ARX, Neural State Space, Export.

Inputs/Outputs

- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(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 |
|-------------------------------------|--------------|--------|---------|
| <input checked="" type="checkbox"/> | Experiment   | 1      | Data    |
| <input checked="" type="checkbox"/> | Experiment_1 | 1      | Data    |



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

Simulation Result  of 1

Show output only

Show as scatter plot

SIMULATION RESULTS OPTIONS

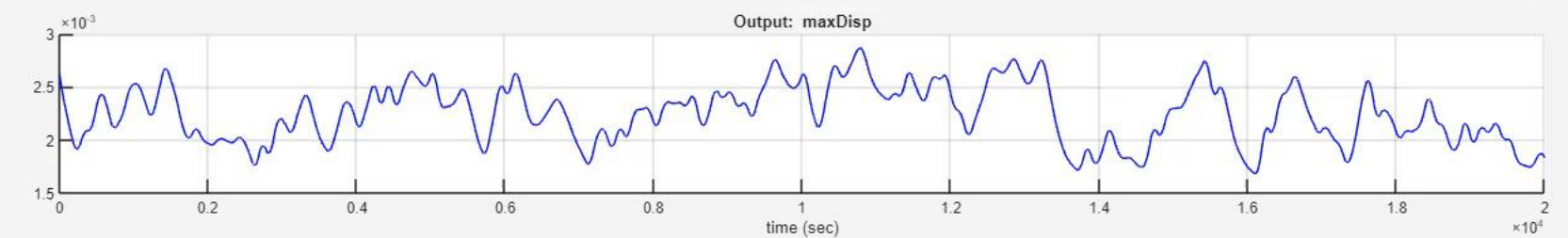
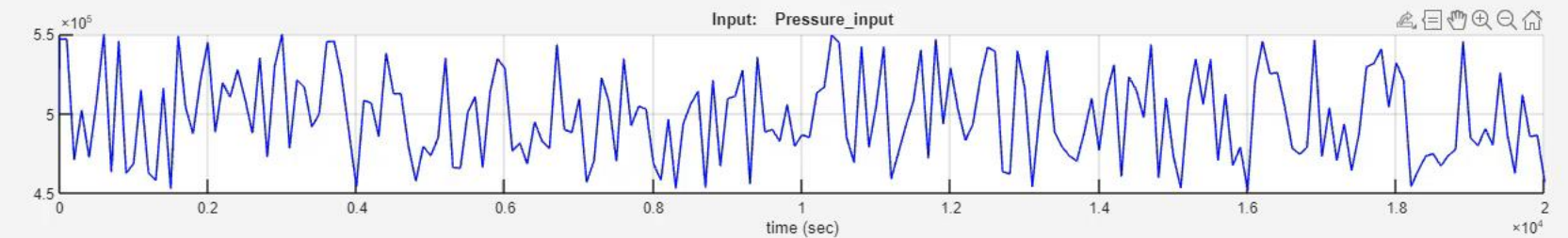
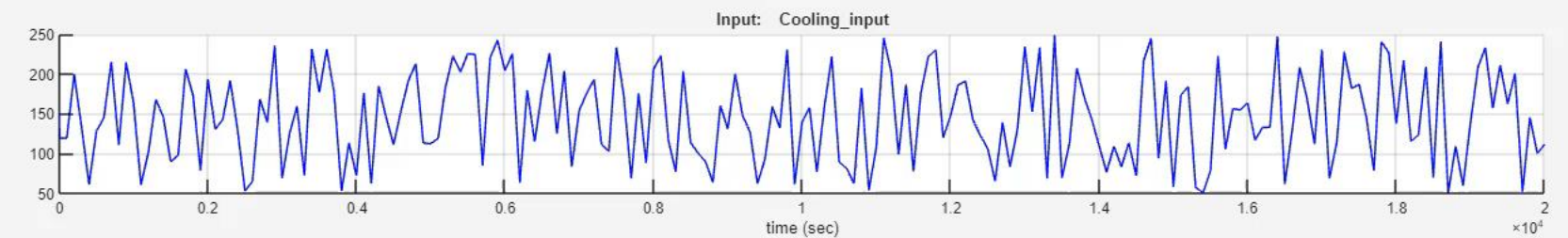
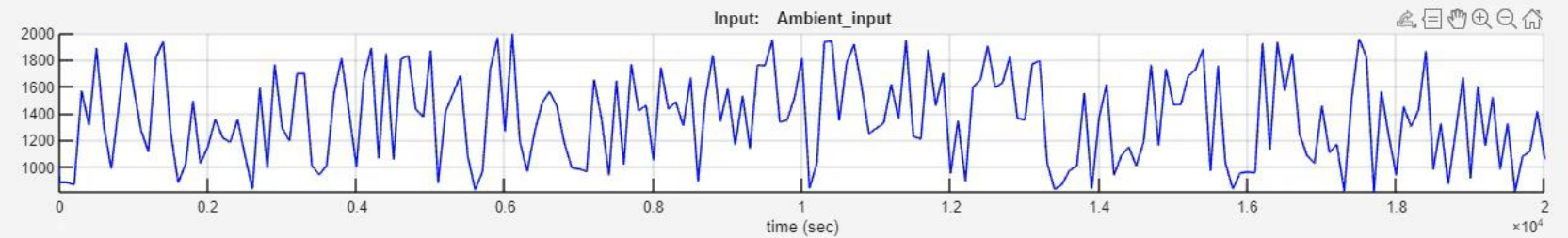
Inputs/Outputs

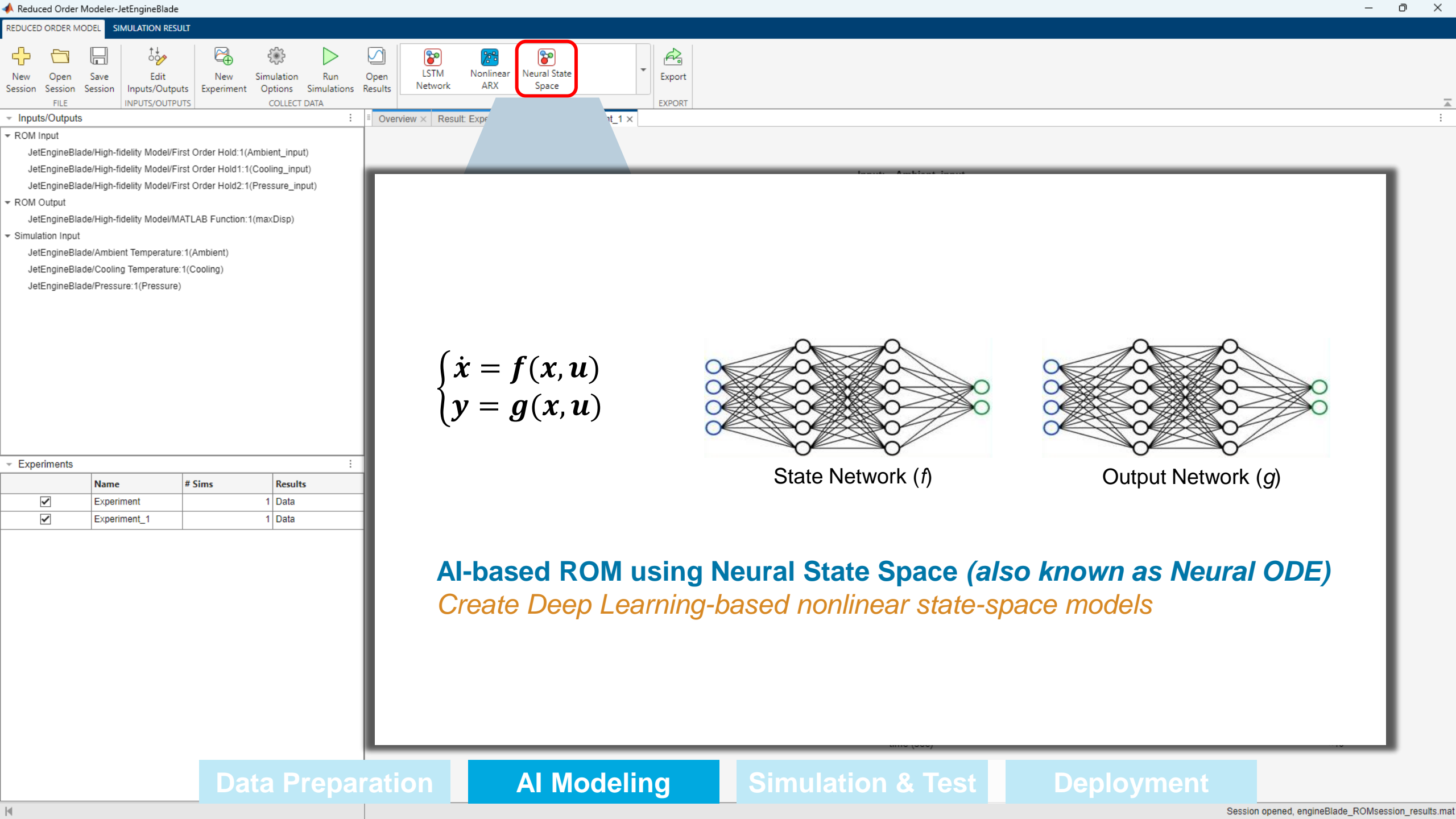
- ROM Input
  - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient\_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling\_input)
  - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure\_input)
- ROM Output
  - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)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

Experiments

|                                     | Name         | # Sims | Results |
|-------------------------------------|--------------|--------|---------|
| <input checked="" type="checkbox"/> | Experiment   | 1      | Data    |
| <input checked="" type="checkbox"/> | Experiment_1 | 1      | Data    |

Overview × Result: Experiment × Result: Experiment\_1 × Experiment\_1 ×





Toolbar with icons for: 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.

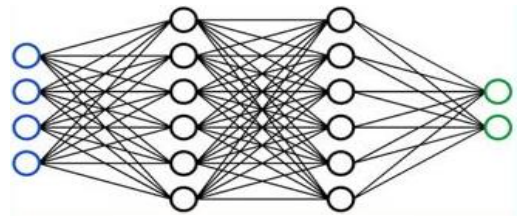
Inputs/Outputs

- 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)

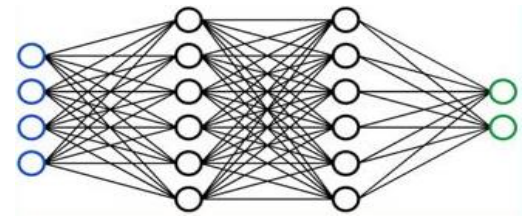
Experiments

|                                     | Name         | # Sims | Results |
|-------------------------------------|--------------|--------|---------|
| <input checked="" type="checkbox"/> | Experiment   |        | 1 Data  |
| <input checked="" type="checkbox"/> | Experiment_1 |        | 1 Data  |

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



State Network (f)



Output Network (g)

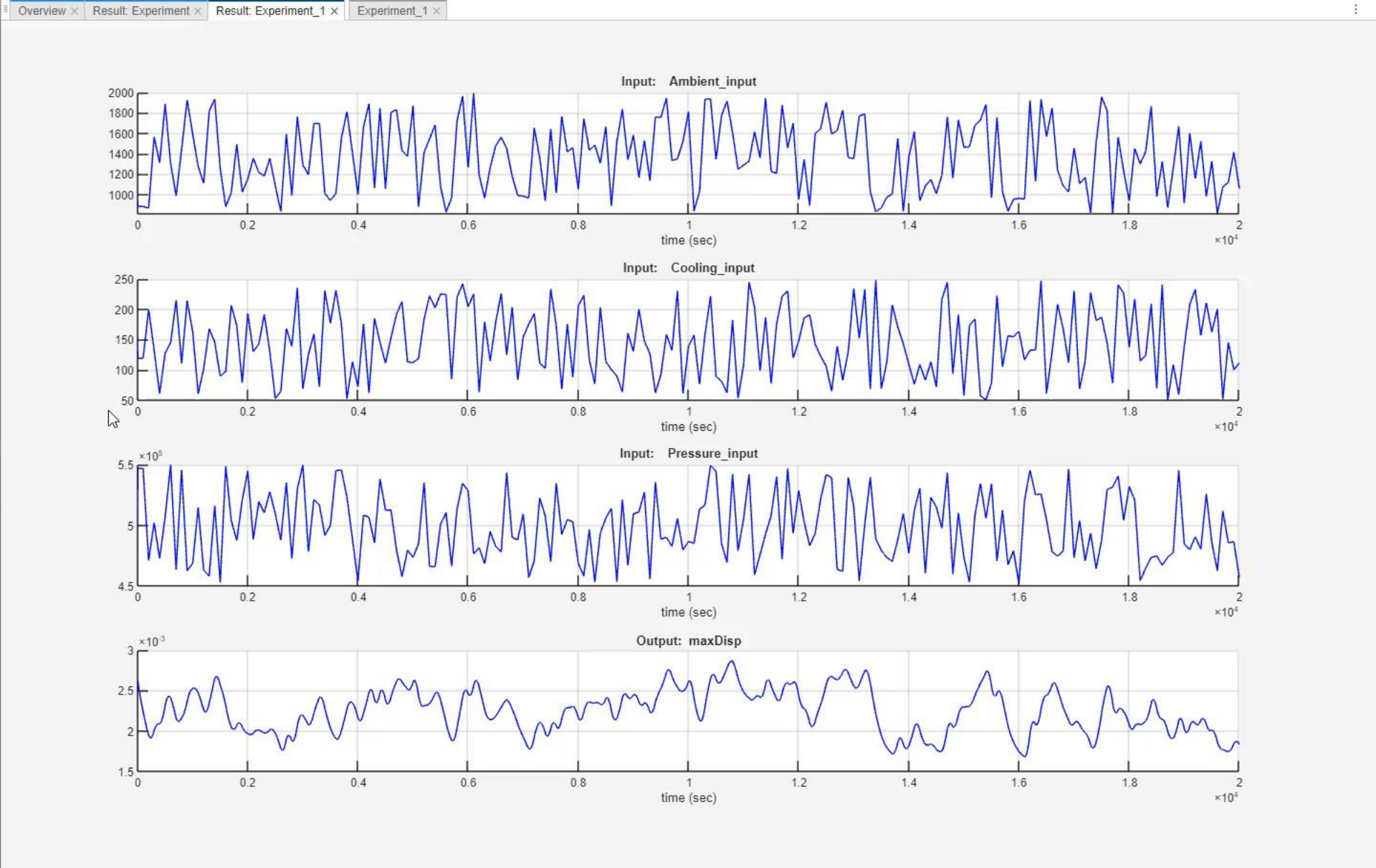
**AI-based ROM using Neural State Space (also known as Neural ODE)**  
*Create Deep Learning-based nonlinear state-space models*

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 |
|-------------------------------------|--------------|--------|---------|
| <input checked="" type="checkbox"/> | Experiment   |        | 1 Data  |
| <input checked="" type="checkbox"/> | Experiment_1 |        | 1 Data  |



# 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**

# Integration of trained AI models into Simulink

The screenshot displays the MATLAB R2023b Live Editor interface. On the left, the 'Current Folder' pane shows a Simulink model named 'JetEngineBlade\_AI.slx'. The 'Workspace' pane shows two variables: 'trainingOutput\_lstm' and 'trainingOutput\_nss', both of type '1x1 struct'. The main editor area is titled 'Experiment to train a NSS model' and contains the following text:

Train a NSS model. Hyper-parameters for training are:

- NumberInputLags - The number of lagged inputs to use, an integer  $\geq 0$
- NumberOutputLags - The number of lagged outputs to use, an integer  $\geq 0$
- NumberLayers - The number of layers in the MLP, an integer  $> 0$
- NumberUnits - The number of hidden units in each layer, an integer  $> 0$
- SampleRate - Sample rate of the model, a real  $> 0$

The tuning follows the following automated steps:

1. Extract and resample the training data
2. Train the NSS model
3. Evaluate model on test data (if available)

Below the text is a code editor showing the start of a function:

```

1 function output = Experiment2_training1(params,monitor)
2
3
4
5
6 % TestSplit - For multiple data sets the percentage of data sets to use
7 % for testing, a double in range [0 100]. The test data sets are selected
8 % randomly from the available data sets.
9 testSplit = 20;
10
11 % BatchSize - The number of data points to use when converting signals into
12 % min-batches, i.e., collections of smaller signal segments.
13 batchSize = 20;

```

The Command Window at the bottom is empty, showing the prompt 'fu >>'.

Data Preparation

AI Modeling

Simulation & Test

Deployment

# Integration of trained AI models into Simulink

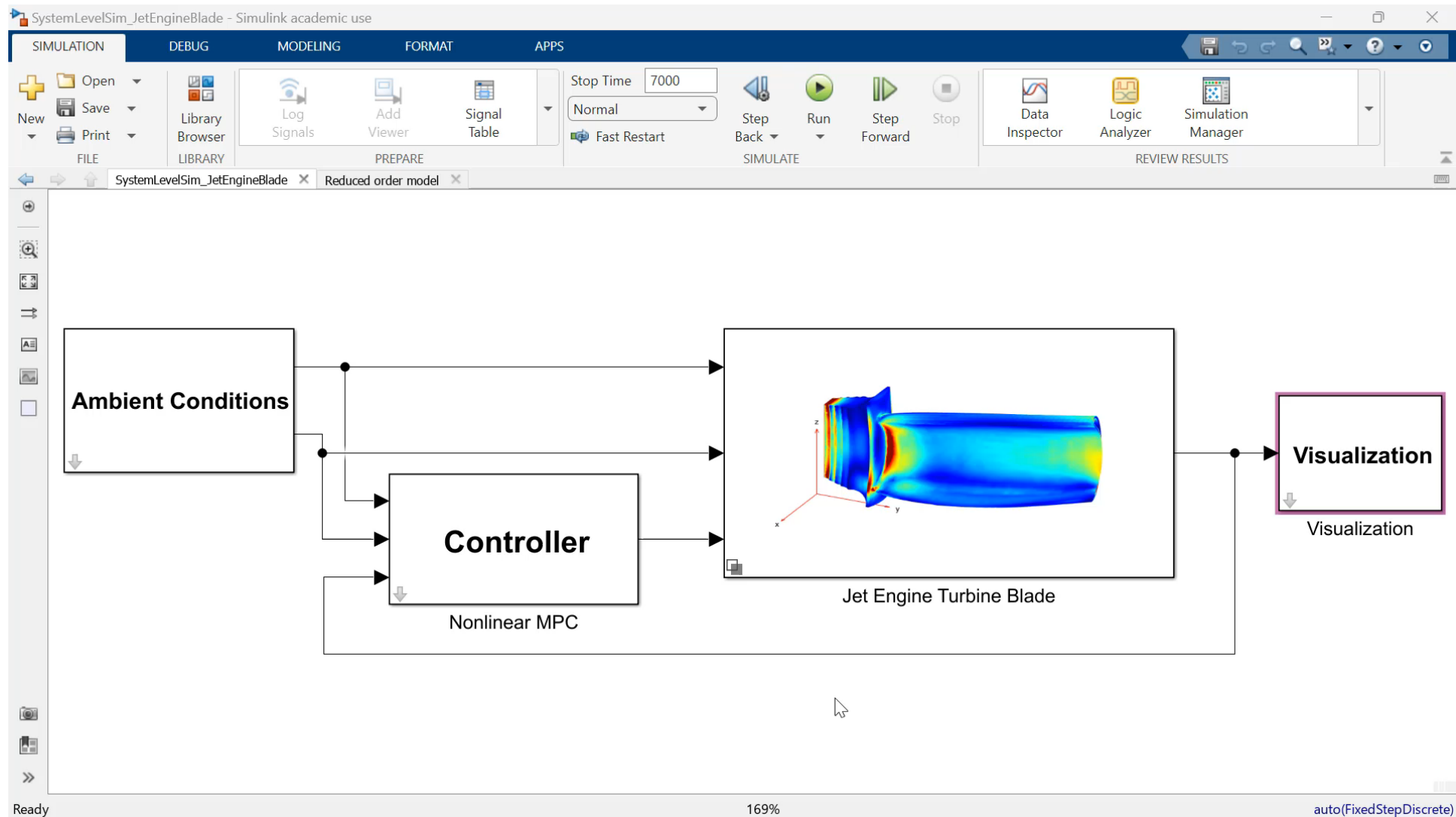
## Simulink Profiler

| 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*



# System-level simulation



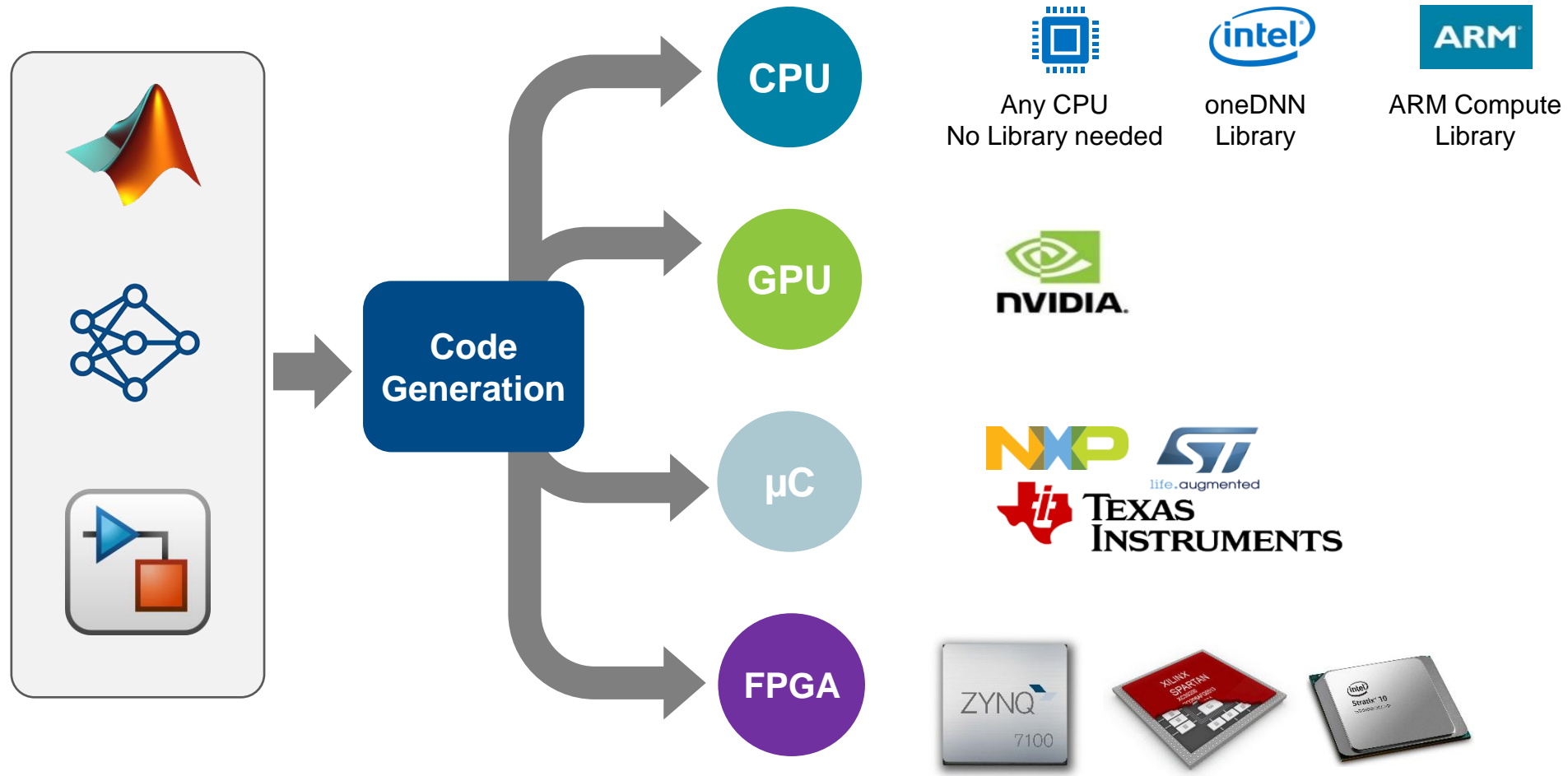
Data Preparation

AI Modeling

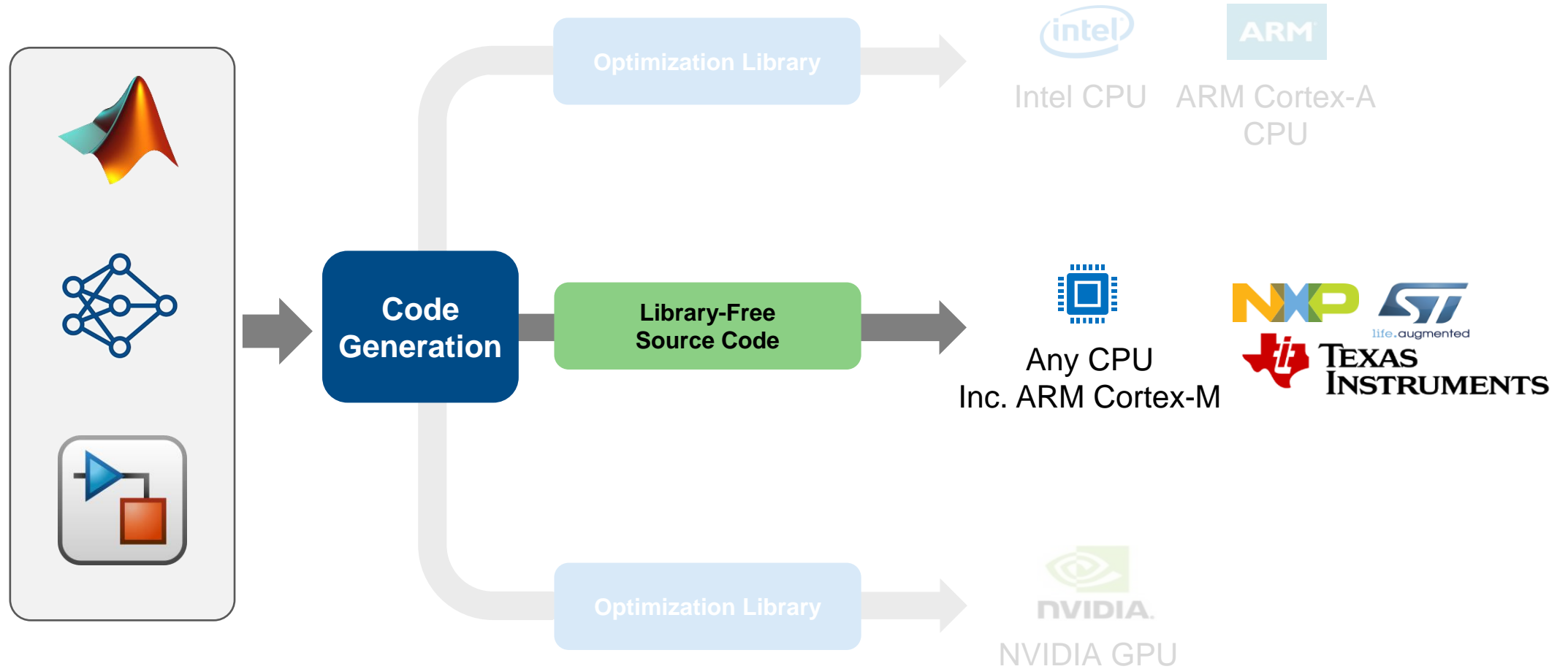
Simulation & Test

Deployment

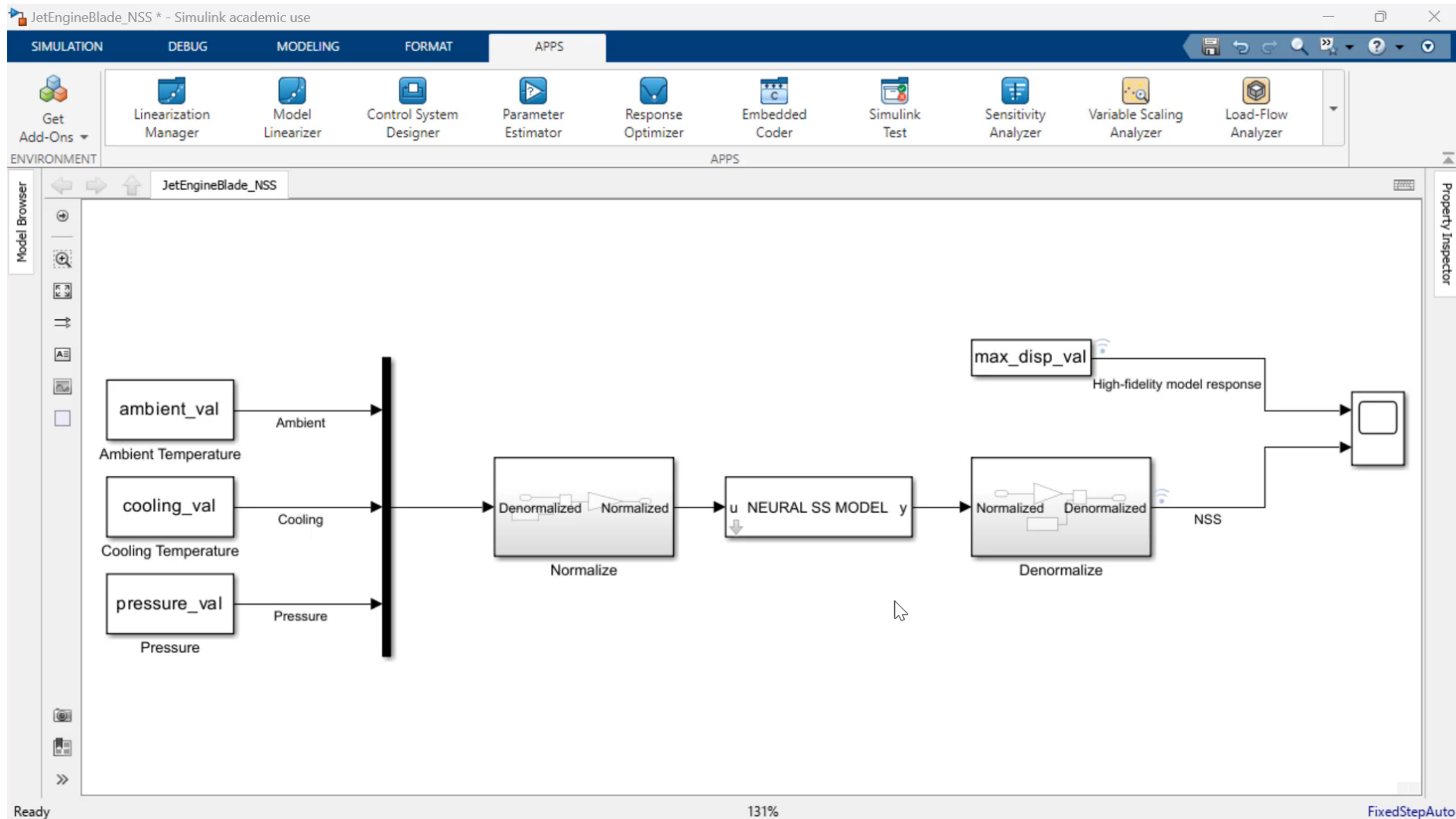
# Deploy to target with zero coding errors



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



# Generate Library-Free C Code for Deep Learning Networks



Data Preparation

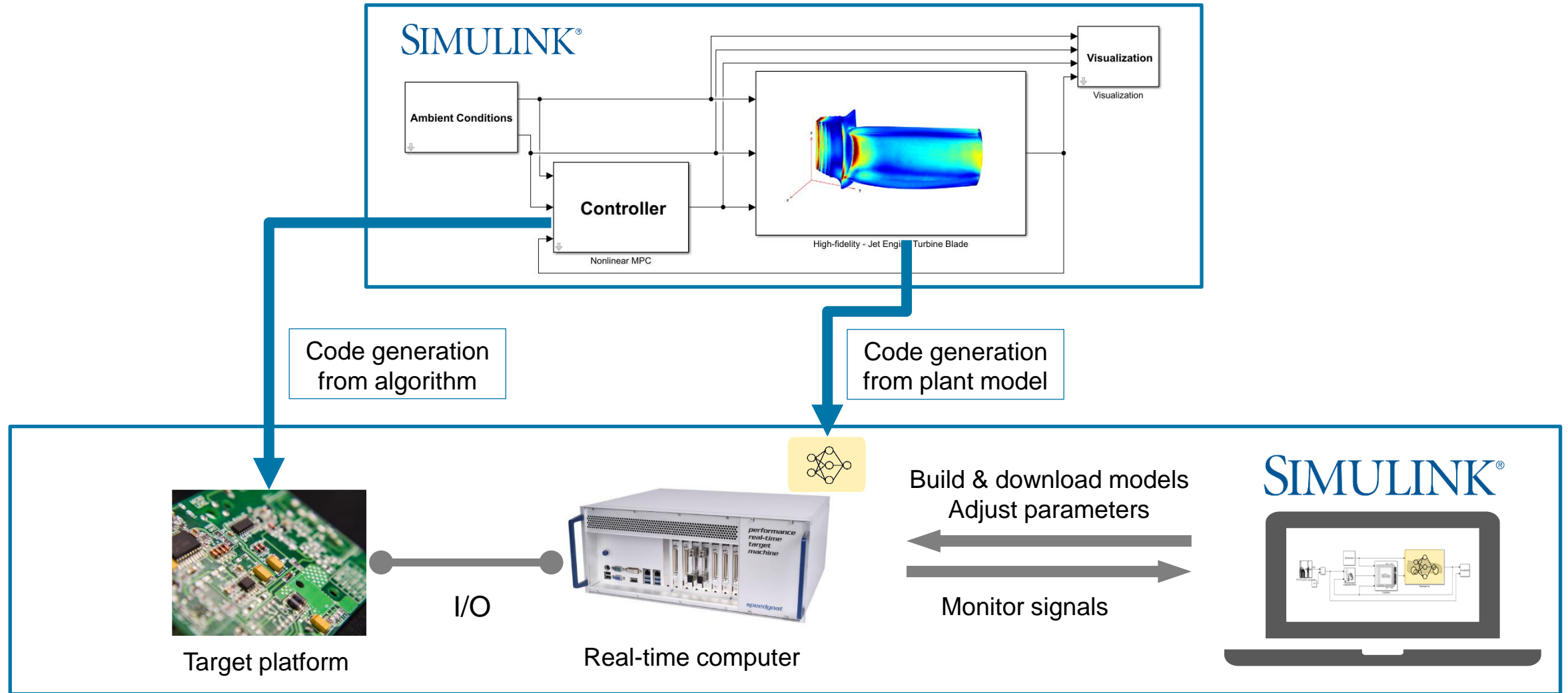
AI Modeling

Simulation & Test

Deployment

# Hardware-in-the-loop simulation

*System-level integration and test*



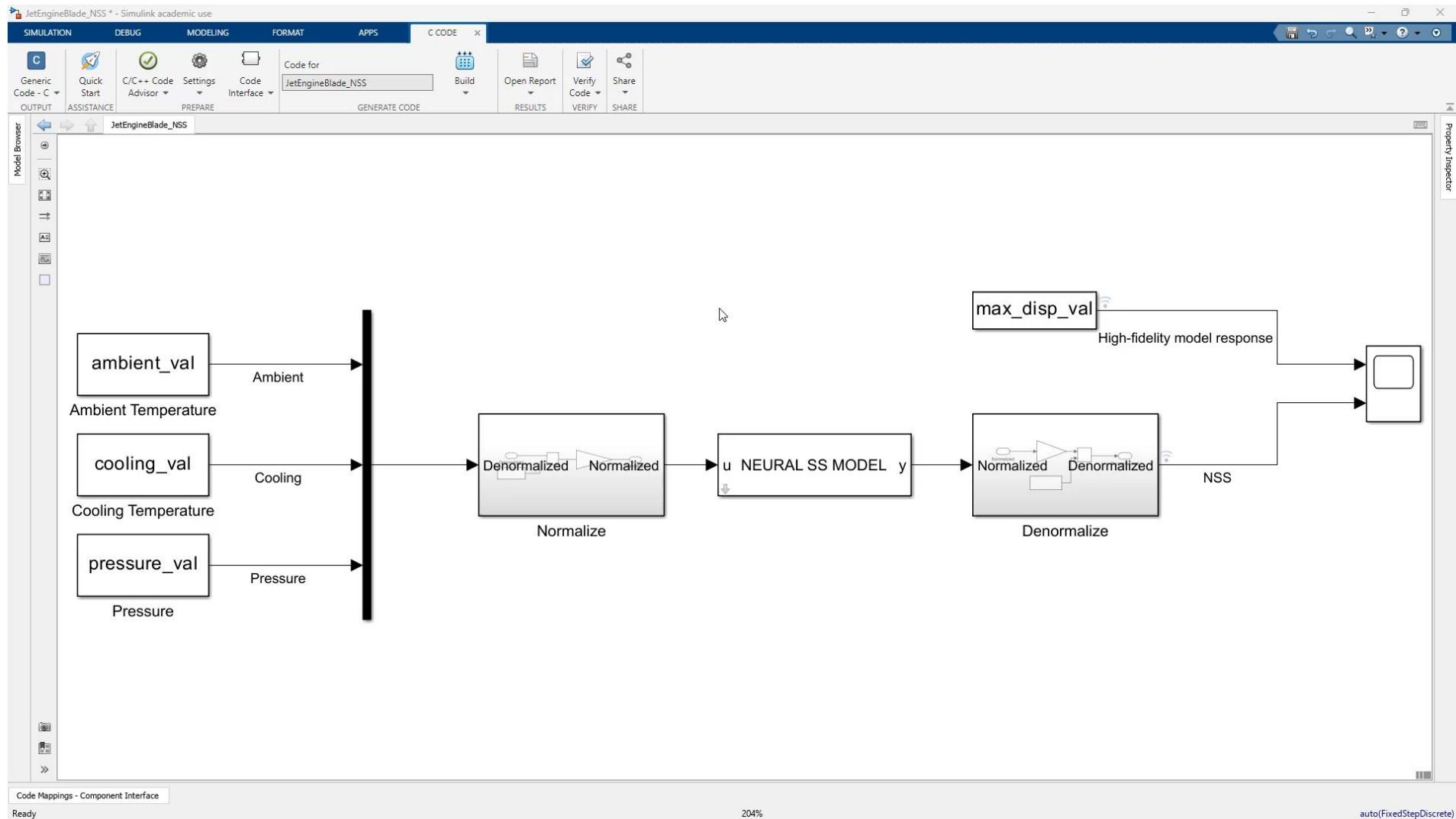
Data Preparation

AI Modeling

Simulation & Test

Deployment

# Hardware-in-the-loop simulation



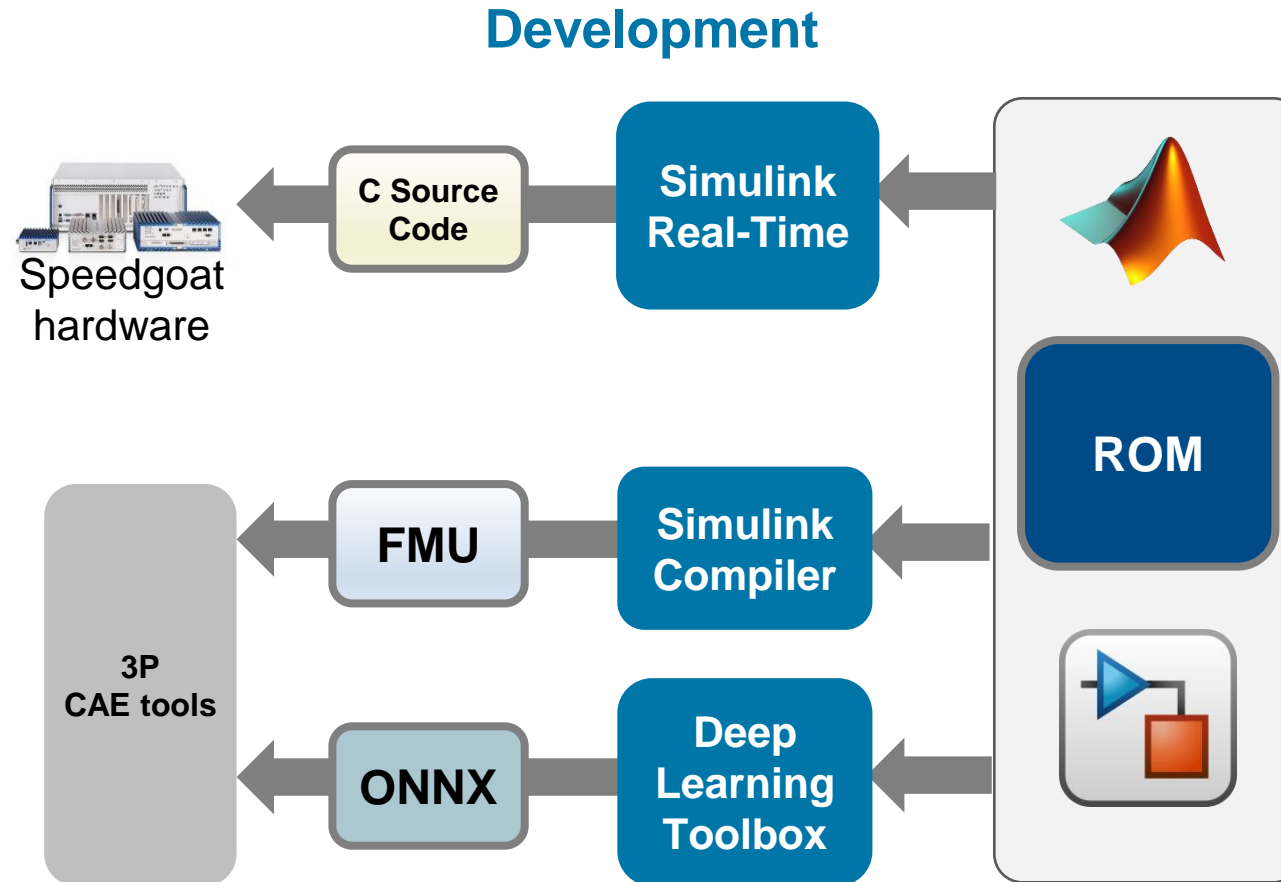
Data Preparation

AI Modeling

Simulation & Test

Deployment

# Use ROMs outside of Simulink for development



# Manage AI tradeoffs for your system

|                  | <b>LSTM</b><br>Long Short-Term Memory<br>Network | <b>Neural State<br/>                     Space</b><br>(Neural ODE) |
|------------------|--|--|
| Training Speed   | ●  | ●  |
| Interpretability | ●  | ●  |
| Inference Speed  | ●  | ●  |
| Model Size       | ●  | ●  |
| Accuracy (RSME)  | ●  | ●  |

*Results are specific to Jet Engine Blade Example*



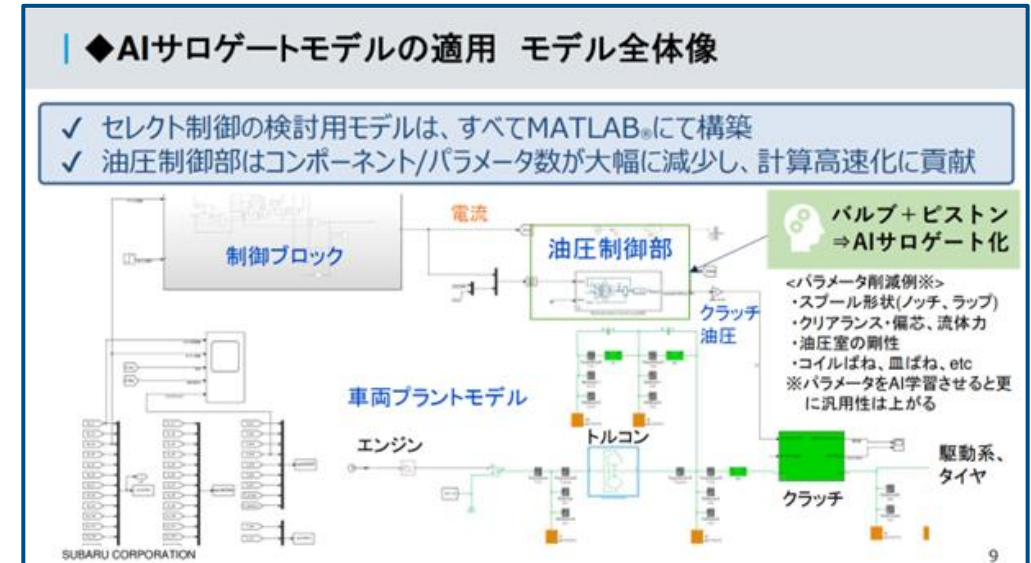


# SUBARU Uses AI Surrogate Model to Reduce Transmission Control System Analysis Time

Using MATLAB, engineers at 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.

## Key Outcomes/Advantages:

- Achieved a 99% reduction in calculation time compared to the original 1D model
- Constructed AI surrogate model in MATLAB that can reproduce waveforms with arbitrary current, oil temperature, and source pressure readings
- Accurately reproduced waveforms, even in oil temperature ranges where the model has not been trained



The AI surrogate model for studying selective control was built completely in MATLAB.

*The AI model can now reproduce waveforms at any source pressure, oil temperature, and current. The calculation time can be significantly reduced while ensuring the accuracy of hydraulic waveforms.*

# Key Takeaways

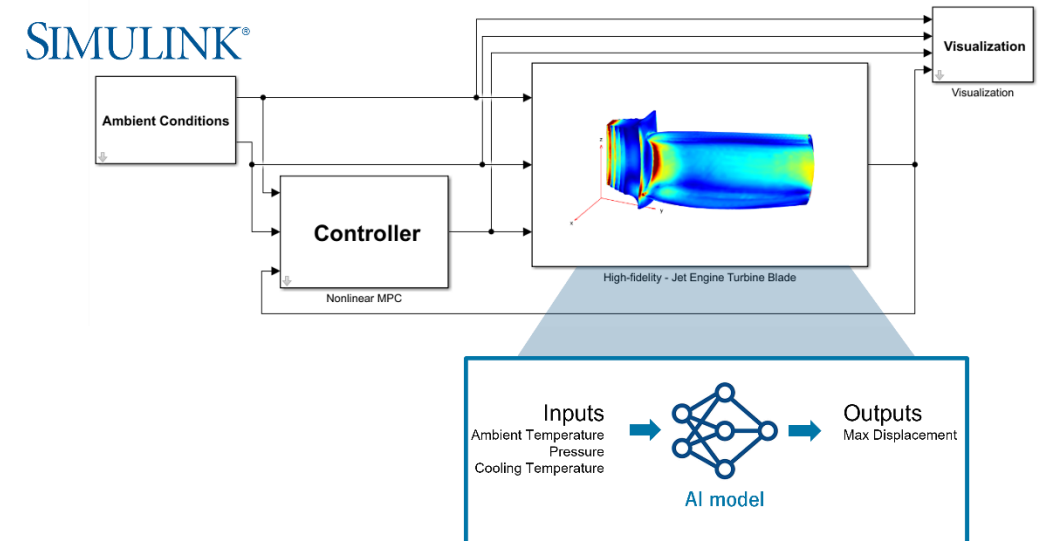
## Enable

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



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

