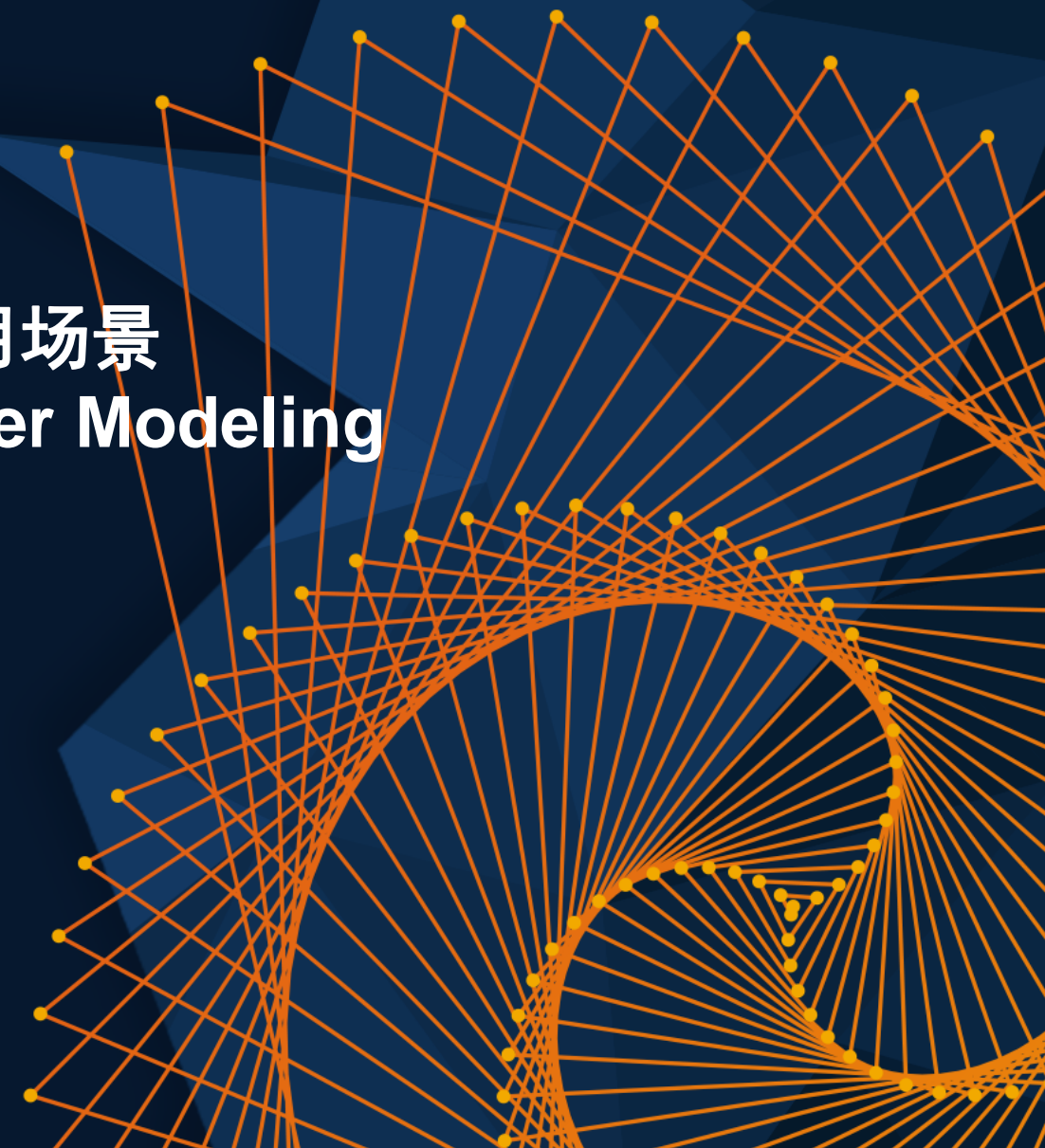


MATLAB EXPO

AI用于Simulink模型的降阶方法和应用场景 AI for Simulink Model Reduced Order Modeling and Application Scenarios



刘海伟, MathWorks



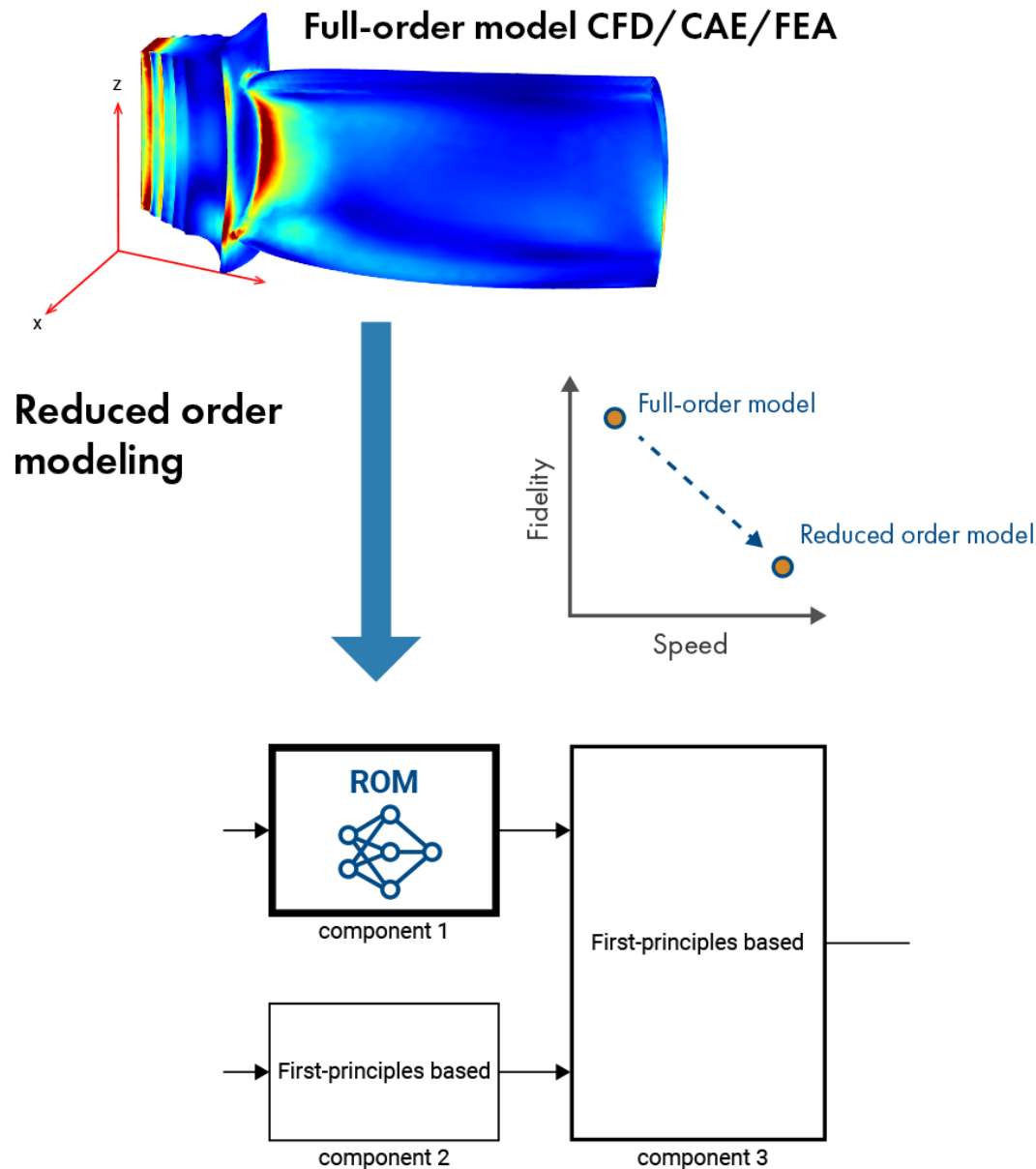
模型降阶

什么是模型降阶？

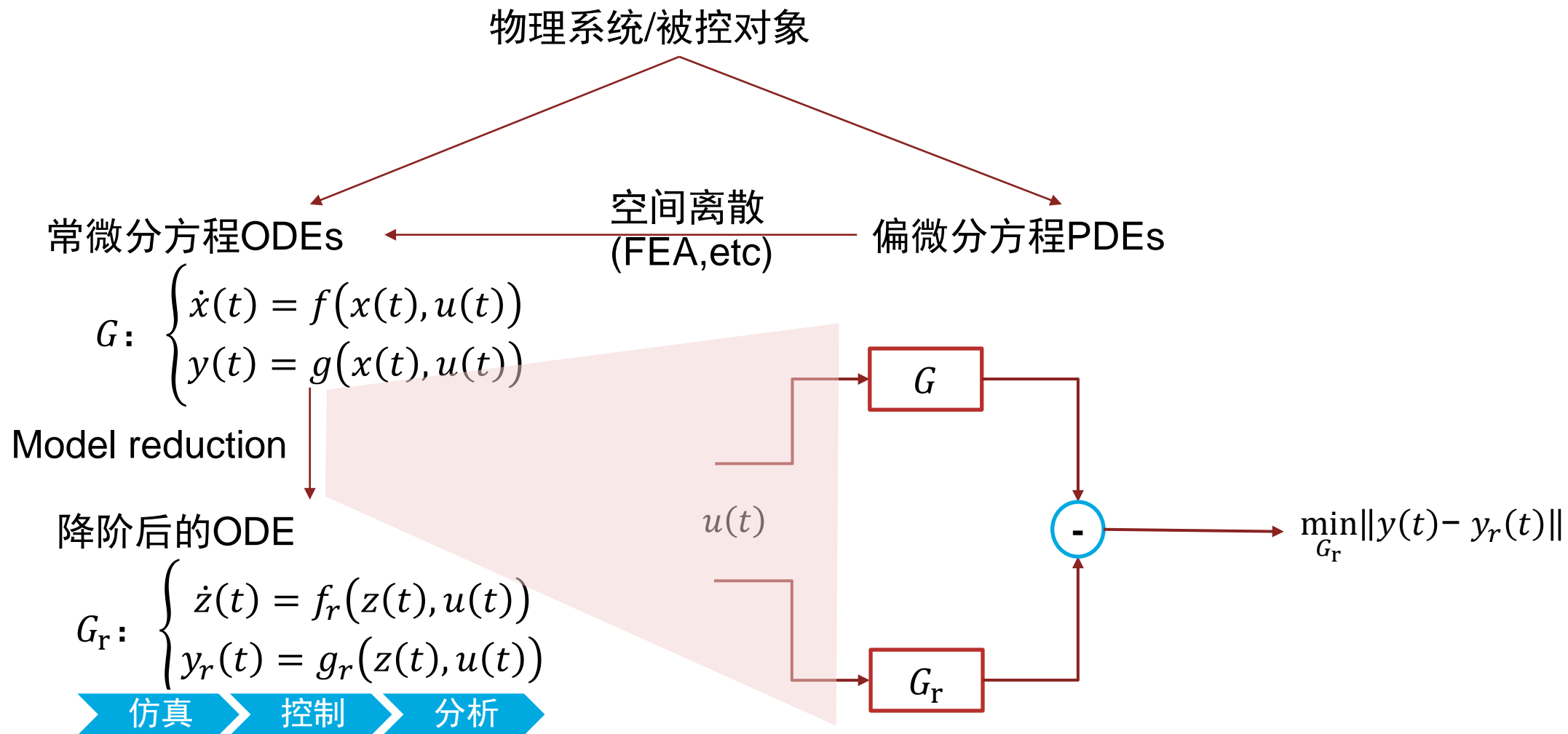
- 减少模型**计算复杂度或存储需求**的一种技术
- 在误差允许的范围内**保持期望的保真度**

为什么进行模型降阶

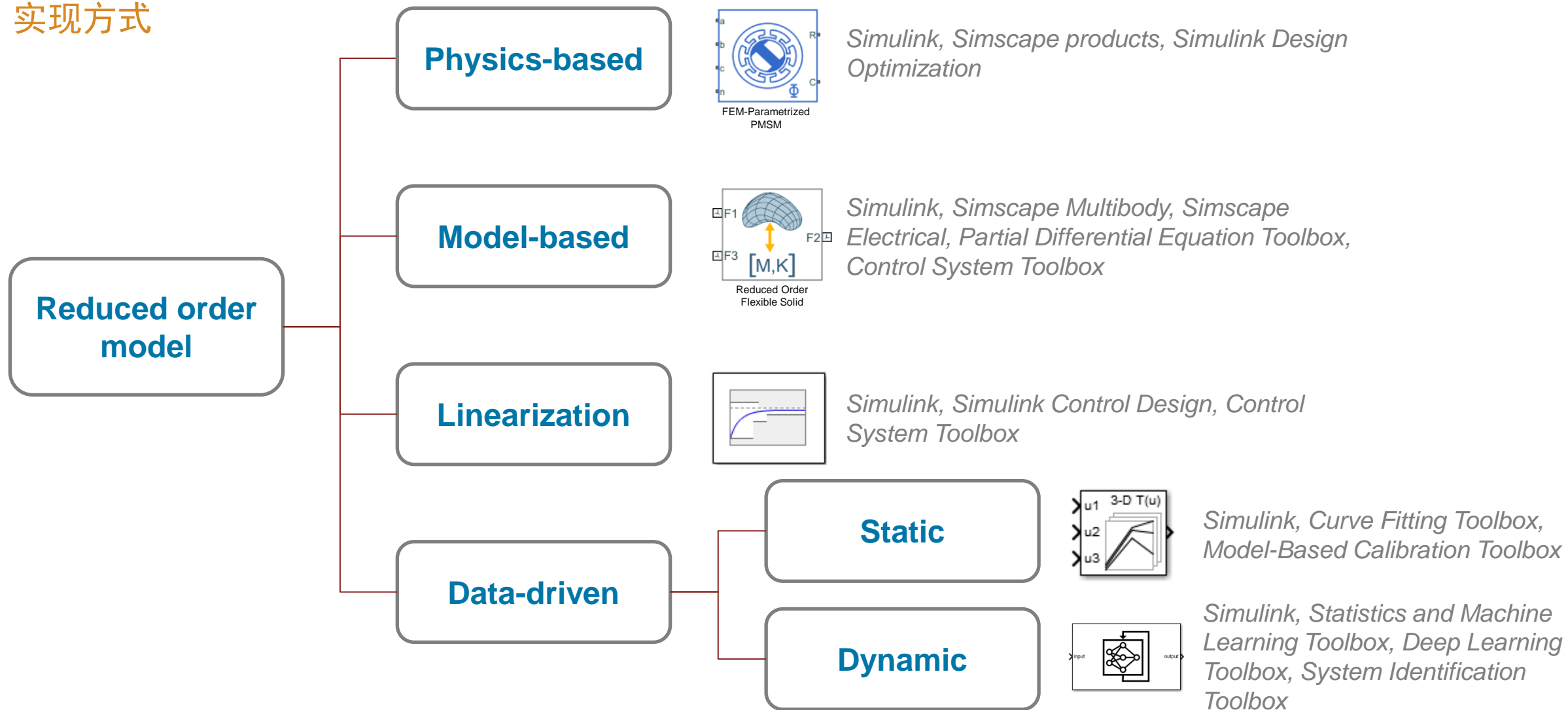
- 加速系统级桌面仿真
- Hardware-in-the-loop testing
- Enable system-level simulation
- 开发虚拟传感器，数字孪生
- 控制设计（例如 模型预测控制）



模型降阶：Overview



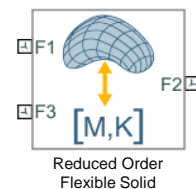
模型降阶 实现方式



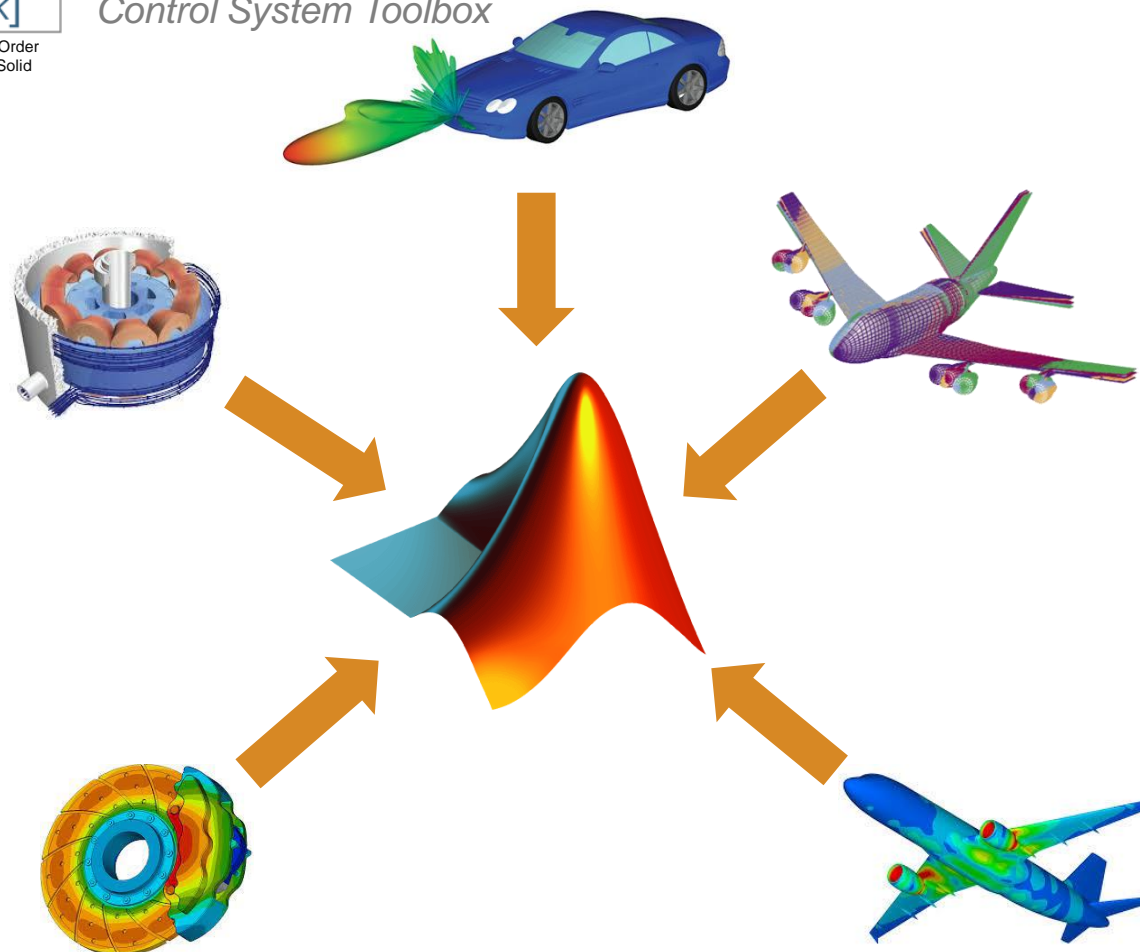
模型降阶

将FEA 和 Simulink/Simscape结合

Model-based



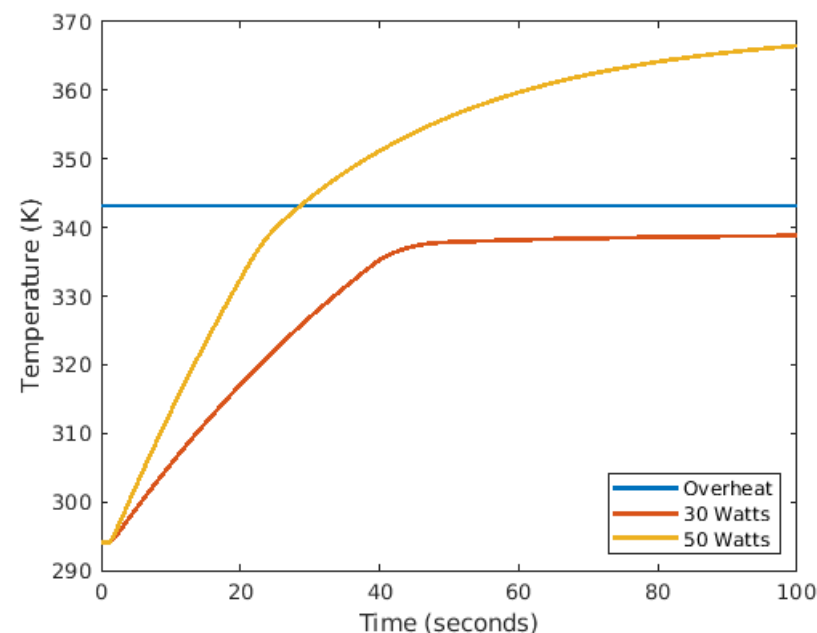
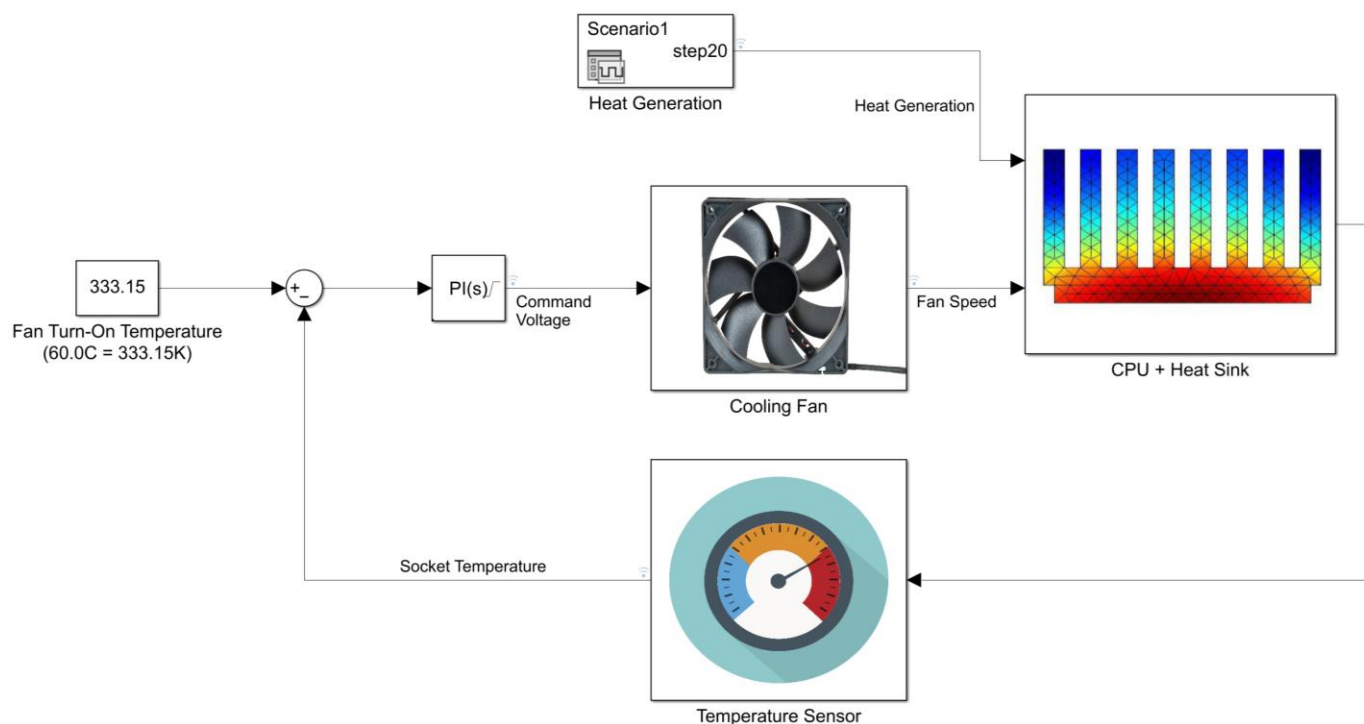
Simulink, Simscape Multibody, Simscape Electrical, Partial Differential Equation Toolbox, Control System Toolbox



模型降阶

使用瞬态温度模型用于系统设计

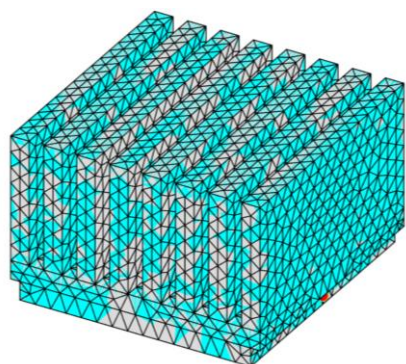
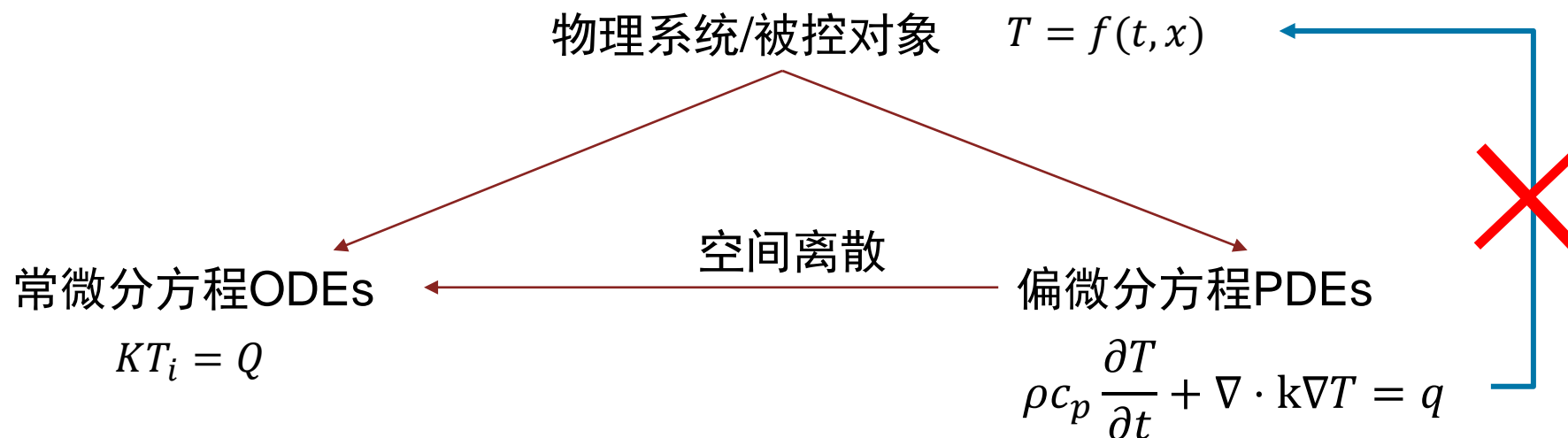
给定一个固定的风扇和散热器几何形状 确定CPU的最大安全运行功率



- 100种不同的运行功率，每次仿真100秒，一个温度场的FEA几个小时

[Model available on MATLAB Central](#)

模型降阶

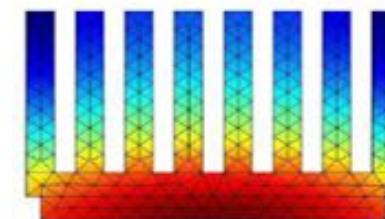


$$M\dot{T} + KT = Q$$

数值离散解

$$T_h(x) = \sum_i T_i \psi_i(x)$$

$$\sum_i T_i \int_{\Omega} k \nabla \psi_i(x) \cdot \nabla \psi_j(x) dV + \sum_i \int_{\partial\Omega} (-k T_i \nabla \psi_i(x)) \cdot \mathbf{n} \psi_j(x) dS = \int_{\Omega} q (\sum_i T_i \psi_i(x)) \psi_j(x) dV$$



稳态场景

$$\nabla \cdot k \nabla T = q$$

$$\int_{\Omega} k \nabla T \cdot \nabla \varphi dV + \int_{\partial\Omega} (-k \nabla T) \cdot \mathbf{n} \varphi dS = \int_{\Omega} q \varphi dV$$

引入试函数 φ 得到 弱形式方程

模型降阶

Solve Finite Element Analysis Workflows with PDE Toolbox

$$\rho c_p \frac{\partial T}{\partial t} + \nabla \cdot k \nabla T = q \xrightarrow{\text{FEM}} \mathbf{MT} + \mathbf{KT} = \mathbf{Q}$$

Geometry

Preprocess

Solve

Postprocess

```
heatsink = createpde('thermal','transient');
```

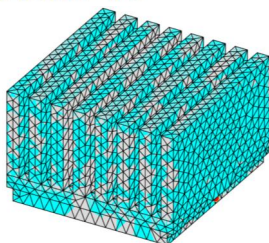
```
geom = importGeometry("heatsink.stl");
```

```
generateMesh(heatsink)
```

```
ans =
```

```
FEMesh with properties:
```

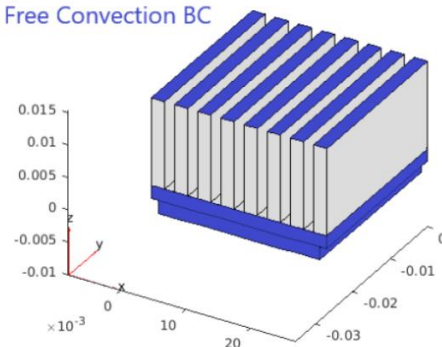
```
Nodes: [3x24410 double]
Elements: [10x12840 double]
MaxElementSize: 0.0017
MinElementSize: 8.4006e-04
MeshGradation: 1
GeometricOrder: '1'
```



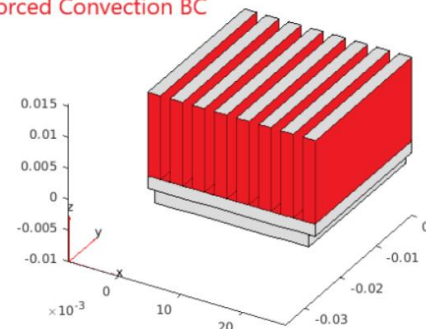
```
% aluminum alloy 6060 T6
mtl = thermalProperties(heatsink,...
    'ThermalConductivity',210, ...
    'MassDensity',2710, ...
    'SpecificHeat',900);
```

```
thermalIC(model,Tambient);
R = solve(model,0:60:7200);
```

Free Convection BC



Forced Convection BC



```
sysMats = assembleFEMatrices(heatsink)
```

```
sysMats = struct with fields:
```

```
K: [24410x24410 double]
A: [24410x24410 double]
F: [24410x1 double]
Q: [24410x24410 double]
G: [24410x1 double]
H: [1x24410 double]
R: [24410x1 double]
M: [24410x24410 double]
```

```
freeBC = thermalBC(heatsink,'Face',freeBCfaces,'HeatFlux',1);
forcedBC = thermalBC(heatsink,'Face',forcedBCfaces,'HeatFlux',1);
```

Statistics & ML Toolbox
Deep Learning Toolbox
Model Based Calibration Toolbox

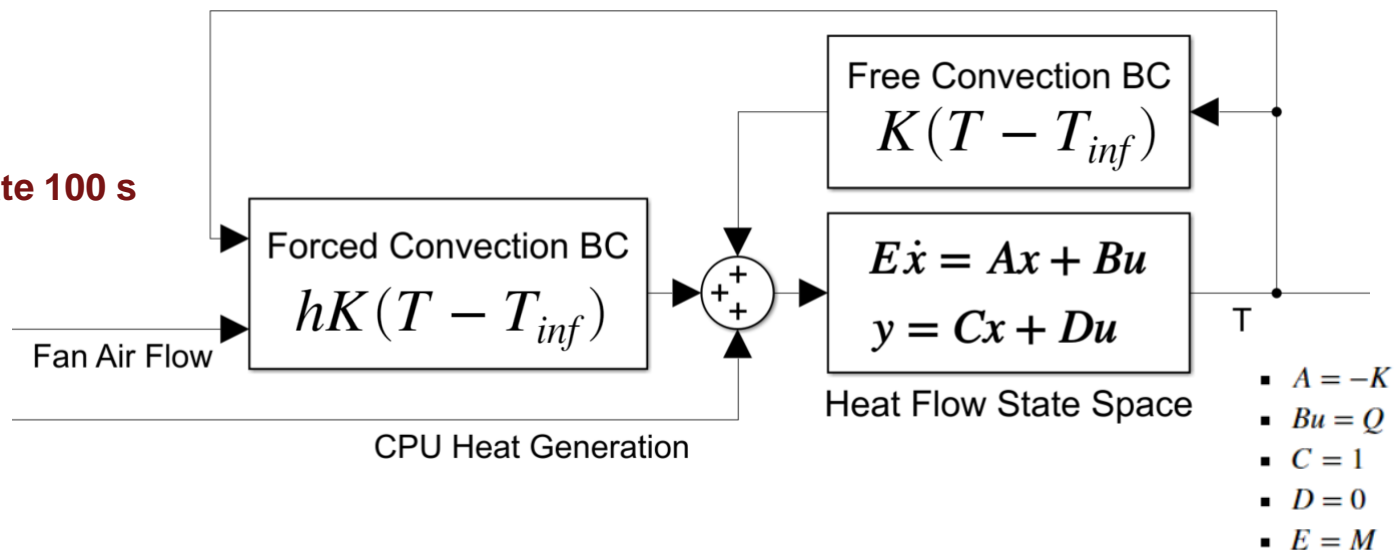
模型降阶

- **descriptor state-space block** 可以实现在Simulink中使用ODE矩阵
 - 结合其他block来约束 BCs, sources, sinks, etc.

```
sysMats = assembleFEMatrices(heatsink)
```

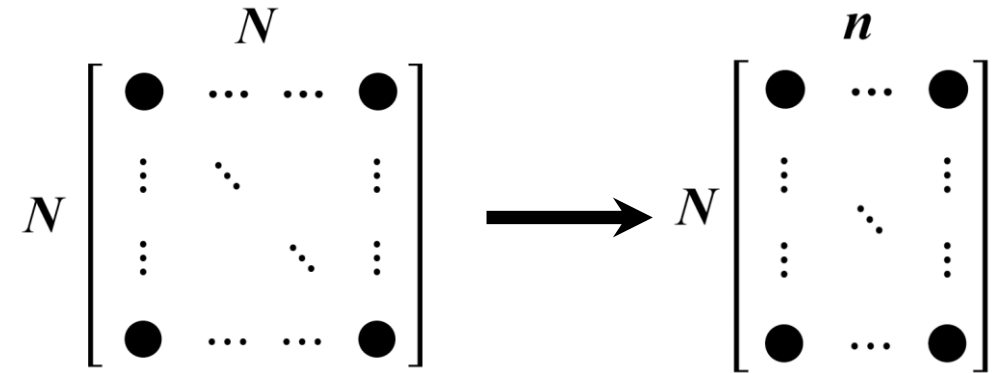
sysMats = struct with fields:

K: [24410×24410 double] **~1 day to simulate 100 s**
 A: [24410×24410 double]
 F: [24410×1 double]
 Q: [24410×24410 double]
 G: [24410×1 double]
 H: [1×24410 double]
 R: [24410×1 double]
 M: [24410×24410 double]



模型降阶

- 大矩阵变小，保持模型动态特性的同时减少状态数
 - 当前温度传导示例，高频模态会快速衰退
- 找到正交模态空间，保留低频模态
 - 对角化ODE方程 $M\dot{T} = Q - KT$ ，得到变换矩阵 V
 - 忽略掉对应高频模态（大特征值）的矩阵列向量



$$A_r = V^T A V \quad B_r = V^T B$$

$$C_r = C V \quad D_r = D \quad E_r = V^T E V$$

$$K, M \in \mathbf{R}^{N \times N}$$



$$K_r, M_r \in \mathbf{R}^{n \times n}$$



$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

eigs()
reduce()
 freqsep()

Model Reduction 方法

Selection

去掉目标频域范围外的特征

`freqsep()`

Approximation

找到并移除对目标输出影响小的特征

`balred()`

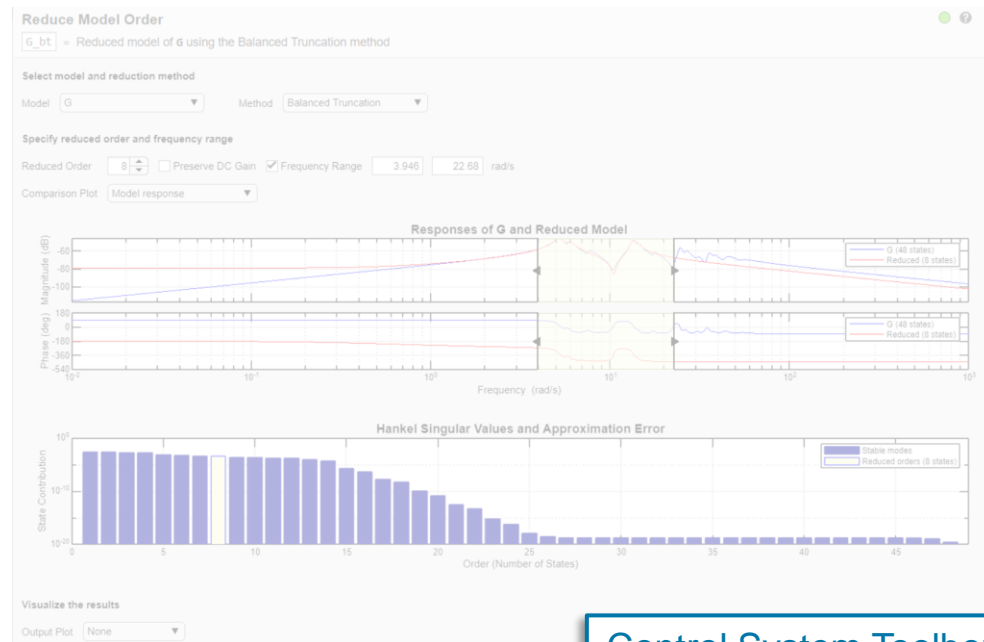
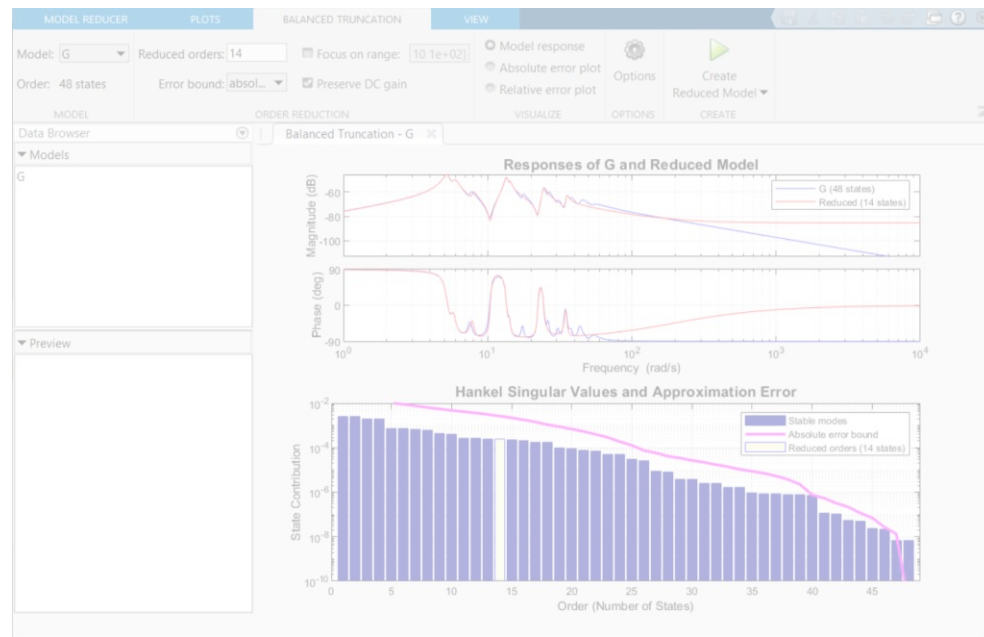
Simplification

Reduce without approximation

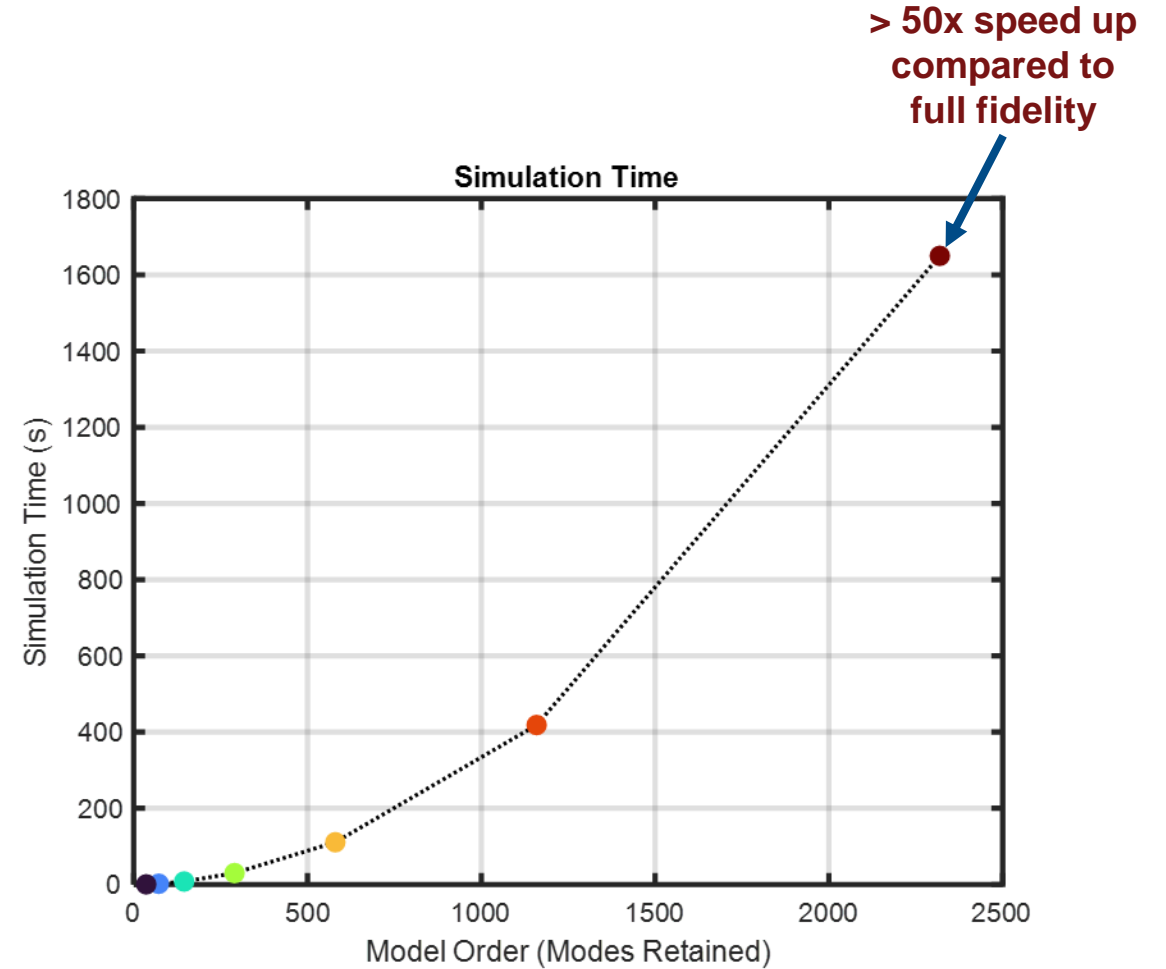
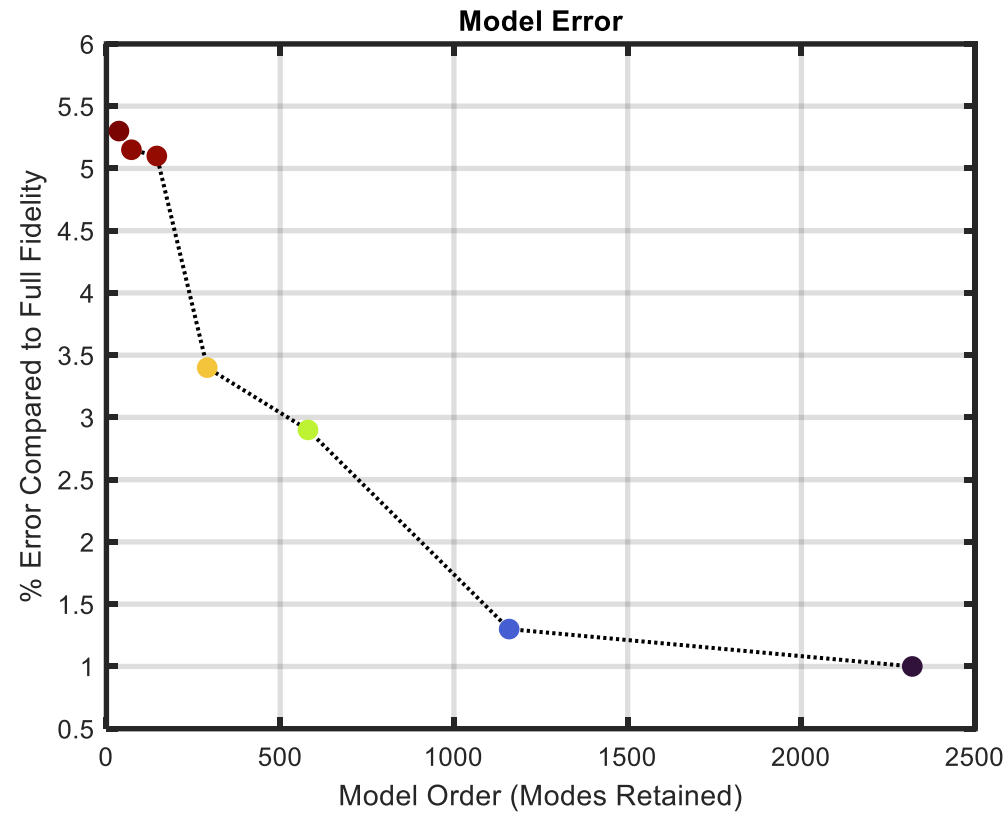
`minreal()`

检测并移除对目标输出没有影响的部分

`simnreal()`



降阶模型的性能



$$K, M \in \mathbf{R}^{24,410 \times 24,410}$$

1天



$$K_r, M_r \in \mathbf{R}^{64 \times 64}$$

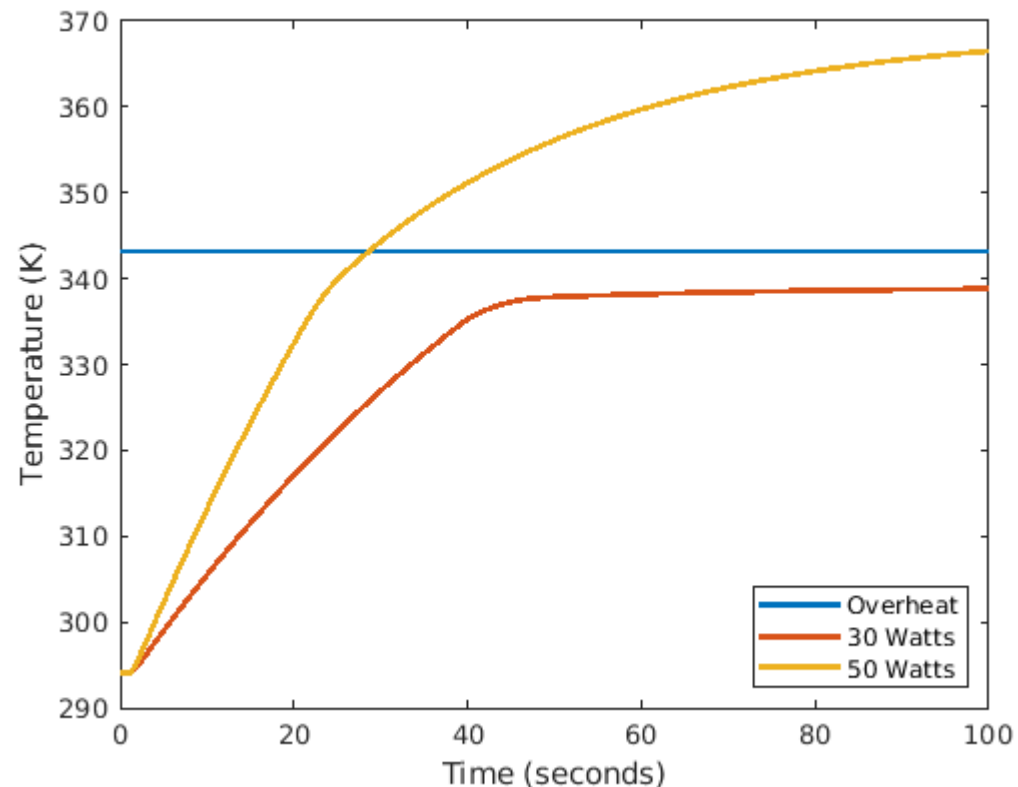
<1 分钟

使用降阶后的模型进行设计优化

```
nSims = 100;
in(1:nSims) = Simulink.SimulationInput mdl;
for i = 1:nSims
    in(i) = setBlockParameter(in(i), [mdl '/Heat Generation'], ...
        'ActiveScenario', ['Scenario' num2str(i)]);
end
out = parsim(in, 'TransferBaseWorkspaceVariables', 'on');
openSimulationManager(in, out)
```

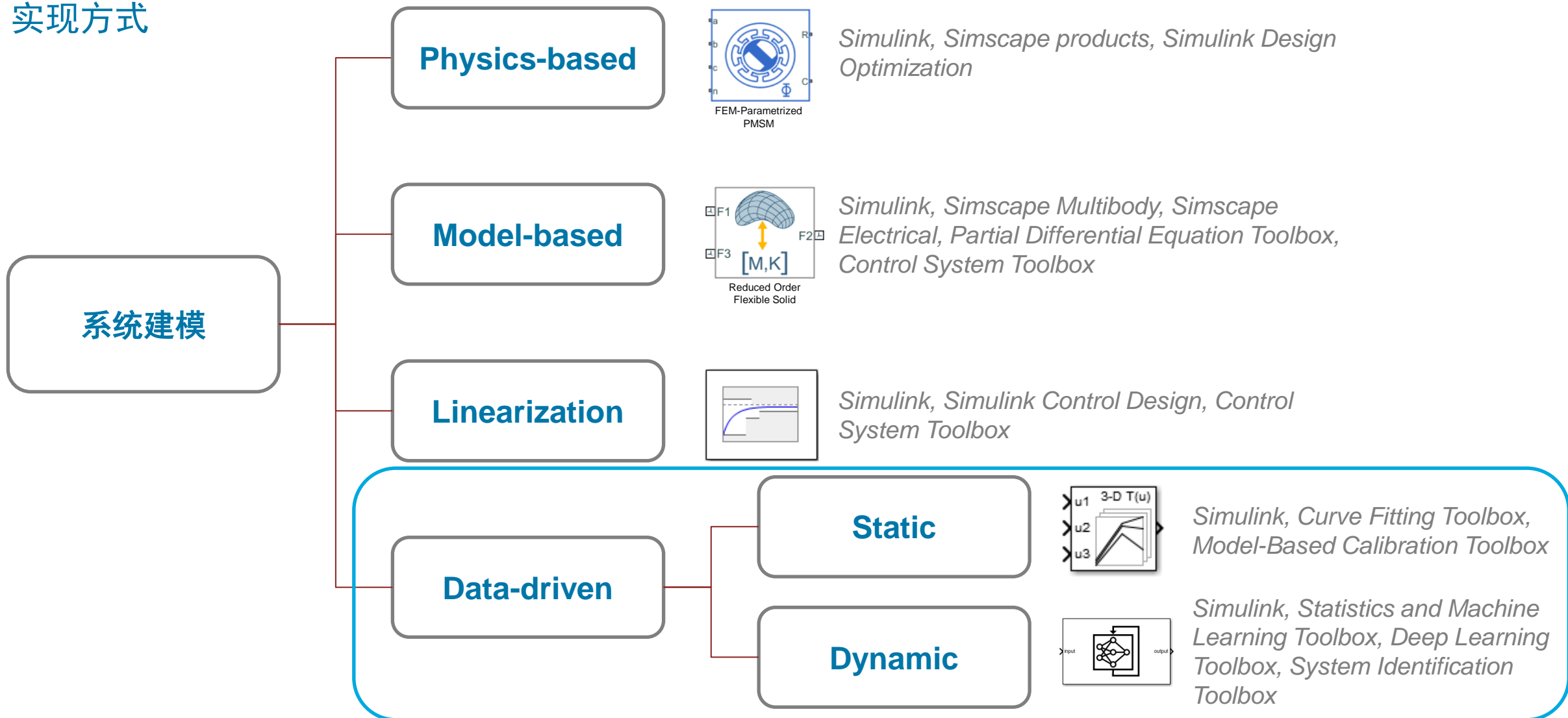
```
[08-Jun-2021 16:24:58] Starting Simulink on parallel workers...
[08-Jun-2021 16:25:30] Loading project on parallel workers...
[08-Jun-2021 16:25:30] Configuring simulation cache folder on parallel workers.
[08-Jun-2021 16:25:39] Loading model on parallel workers...
[08-Jun-2021 16:26:16] Transferring base workspace variables used in the model
[08-Jun-2021 16:26:21] Running simulations...
[08-Jun-2021 16:26:56] Completed 1 of 100 simulation runs
[08-Jun-2021 16:26:57] Completed 2 of 100 simulation runs
[08-Jun-2021 16:26:57] Completed 3 of 100 simulation runs
```

```
[08-Jun-2021 16:32:12] Completed 97 of 100 simulation runs
[08-Jun-2021 16:32:15] Completed 98 of 100 simulation runs
[08-Jun-2021 16:32:16] Completed 99 of 100 simulation runs
[08-Jun-2021 16:32:19] Completed 100 of 100 simulation runs
[08-Jun-2021 16:32:19] Cleaning up parallel workers...
```

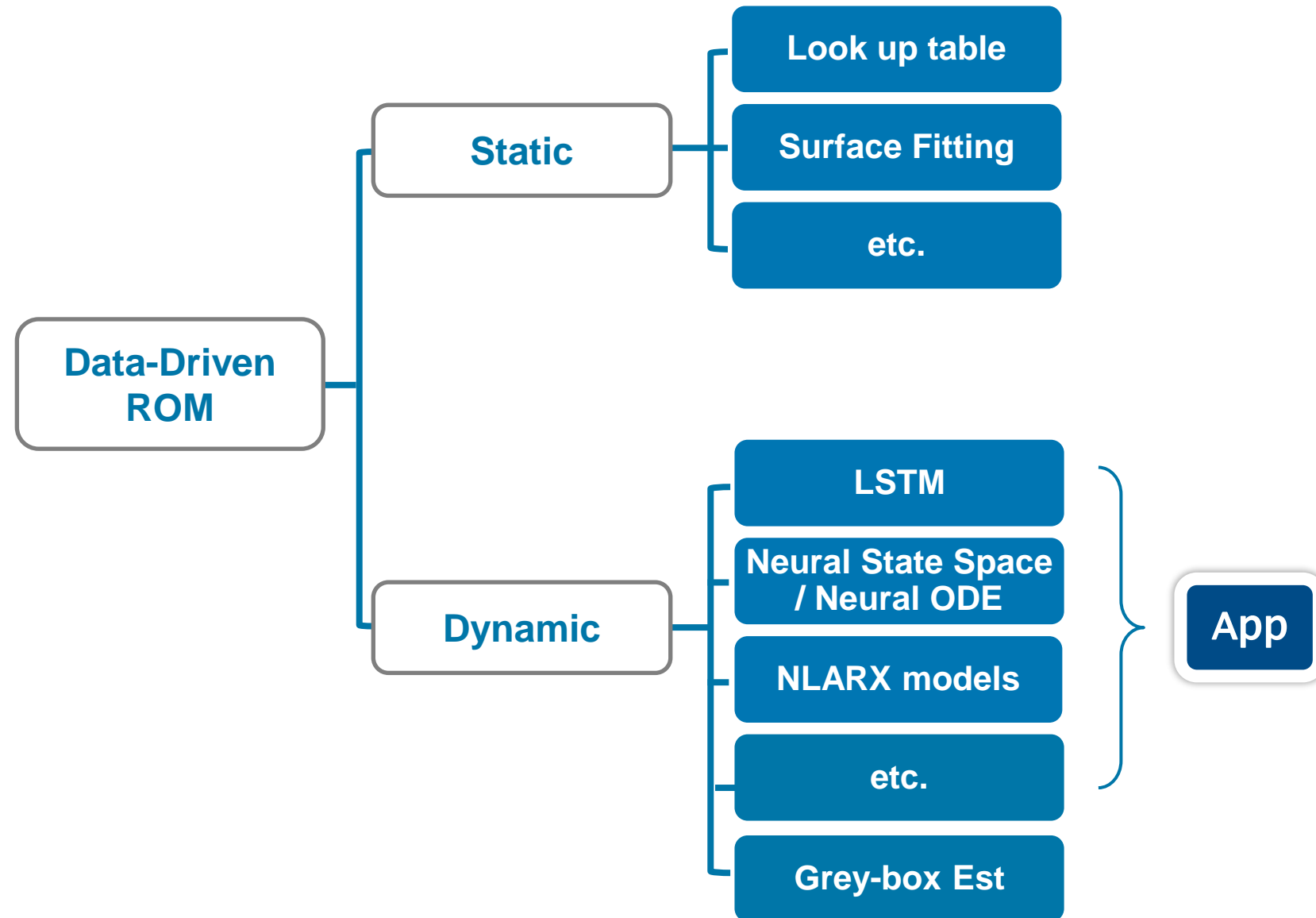


- 设计参数扫描从全保真3个月到降阶后7分钟
- ~5% 错误率的折衷 (c. 1% error换10 小时仿真时间)

模型降阶 实现方式



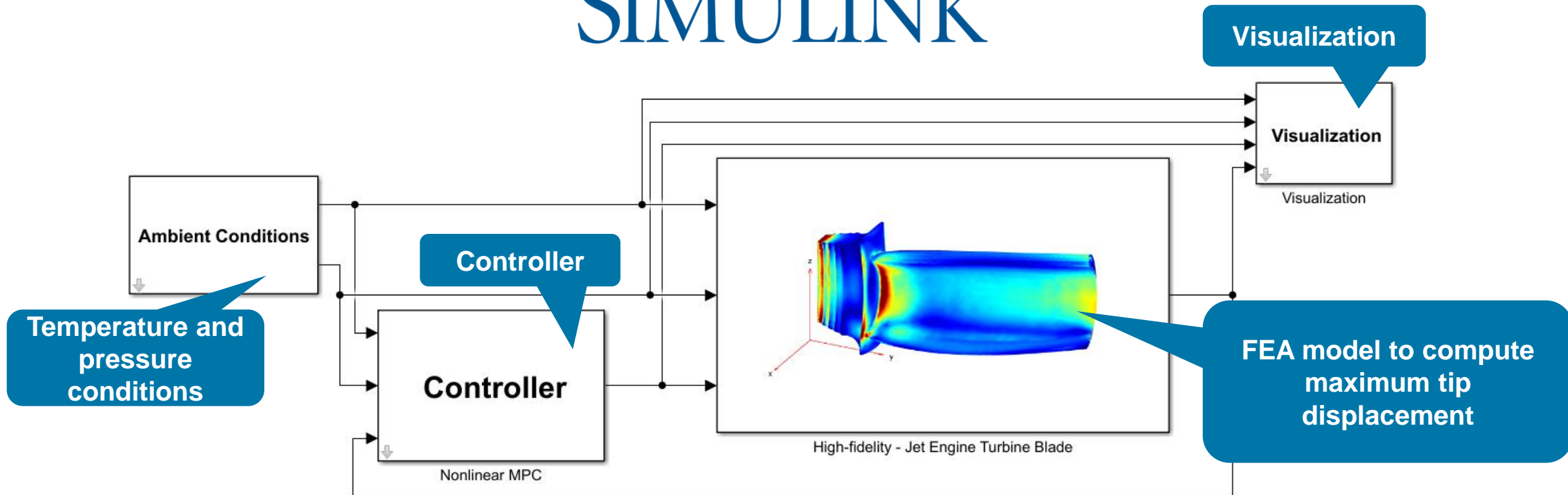
Data-driven ROM



示例概览

用基于AI的降阶模型替代高保真喷气发动机涡轮叶片模型

SIMULINK®

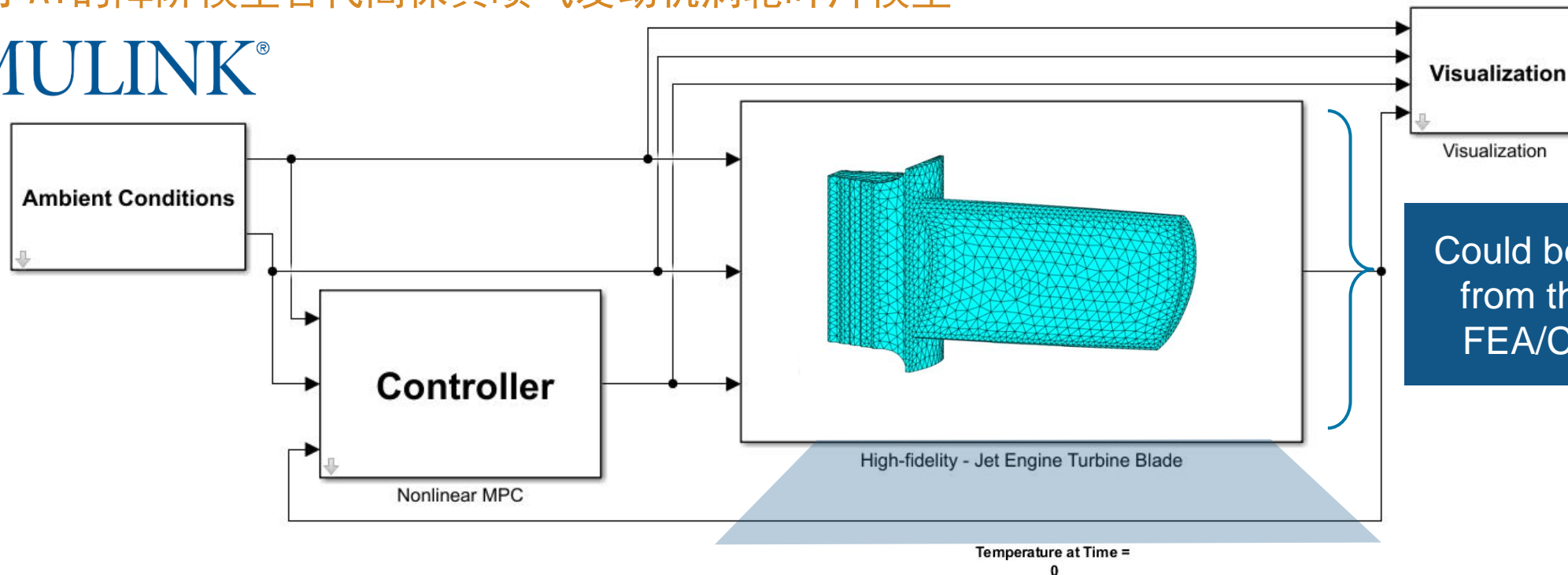


Closed-loop temperature control

示例概览

用基于AI的降阶模型替代高保真喷气发动机涡轮叶片模型

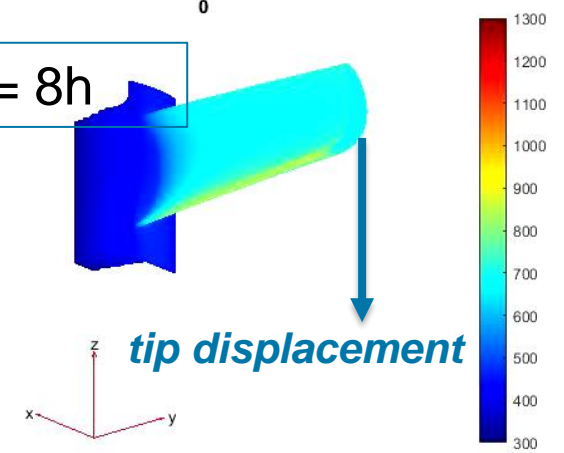
SIMULINK®



~30 seconds per time step for solving FEA models * 5000 = 8h



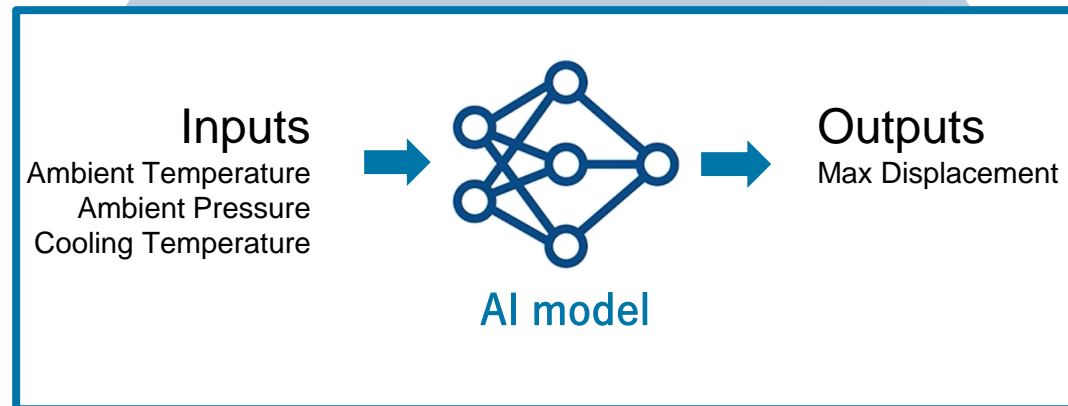
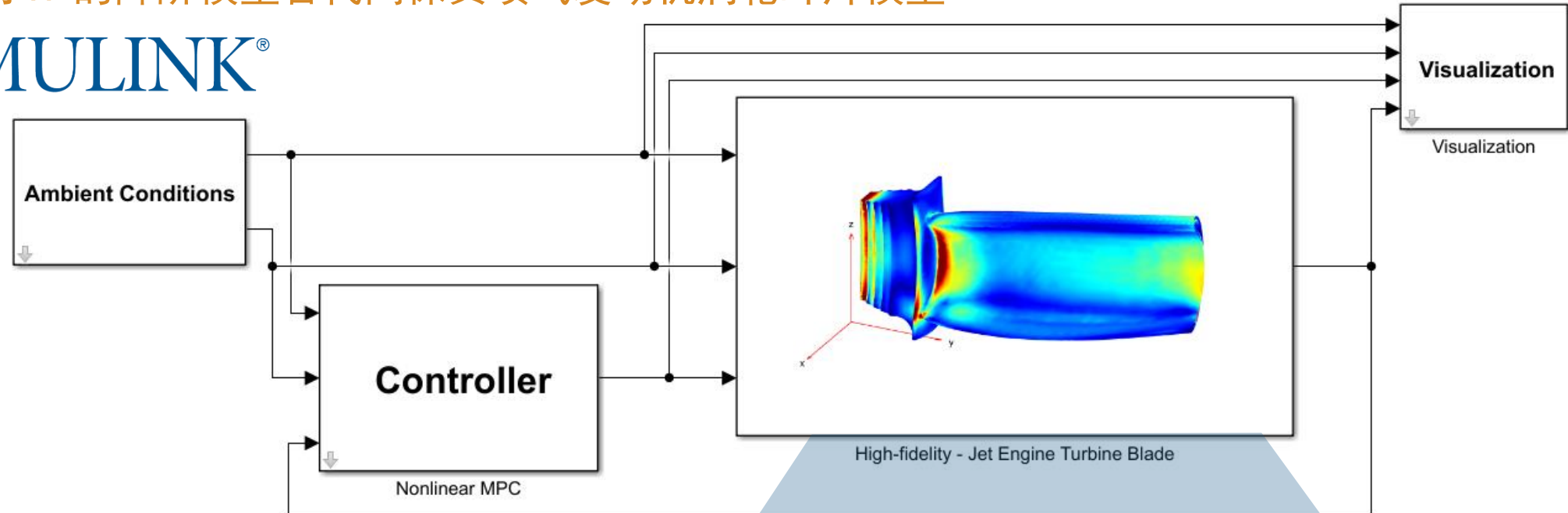
Not suitable for control design and HIL testing



示例概览

用基于AI的降阶模型替代高保真喷气发动机涡轮叶片模型

SIMULINK®



用于模型降阶的Simulink Add-On

创建基于AI的降阶模型(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 Modeler App

Simulink Add-On for Reduced Order Modeling is modeled in Simulink, including full-order models for system-level desktop simulation, high-fidelity subsystems, and surrogate 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 Simulink or outside of Simulink through FMUs
- Export reduced order models as Full-Order Models (FOMs) to Simulink Compiler

Reduced Order Modeling

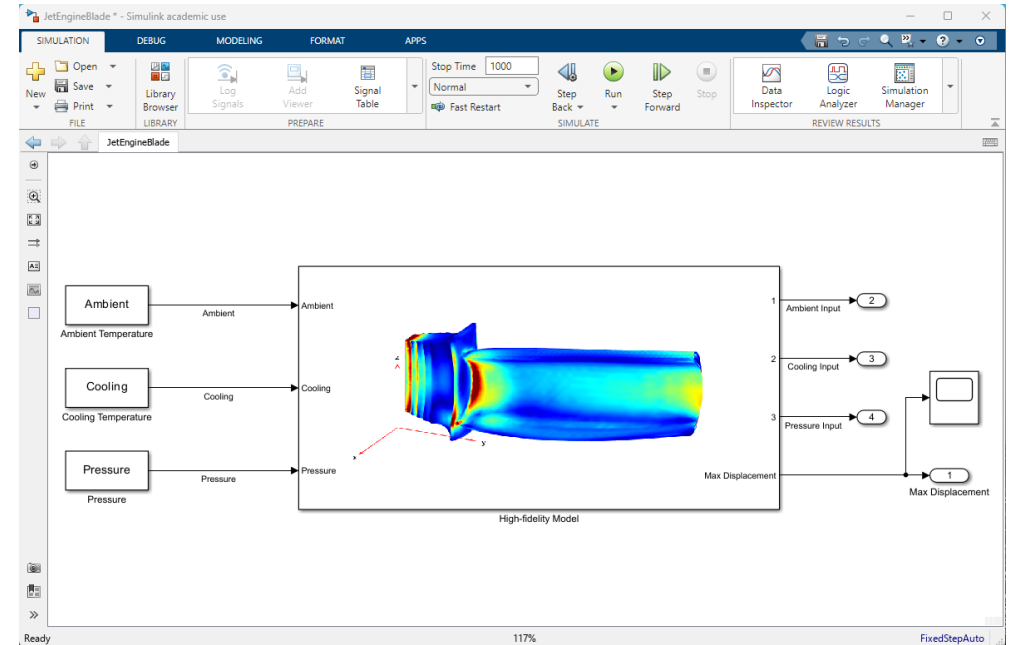
Full-order model CFD/CAE/FEA → ROM component 1 + First-principles based component 2 → First-principles based component 3

Fidelity vs Speed graph showing the trade-off between model accuracy and computational speed.

生成训练数据



Physical system



Simulink/Simscape

Data Preparation

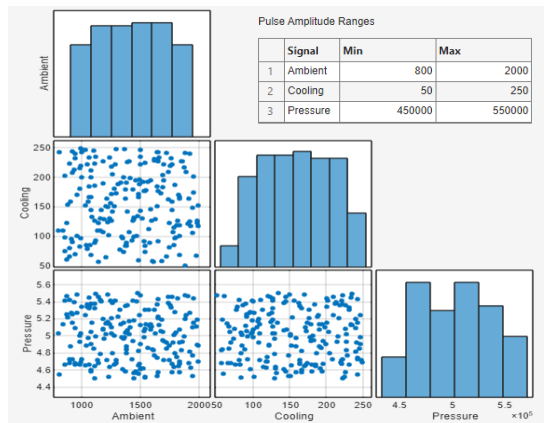
AI Modeling

Simulation & Test

Deployment

合成数据生成

Design of Experiments (实验设计)

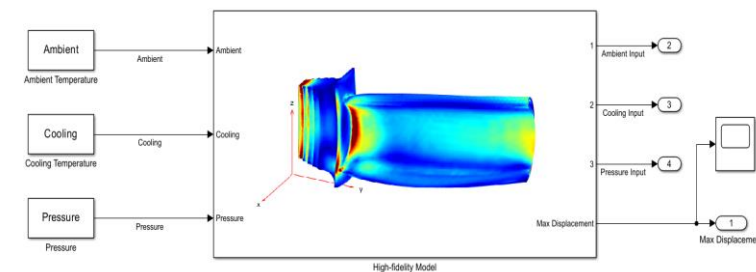
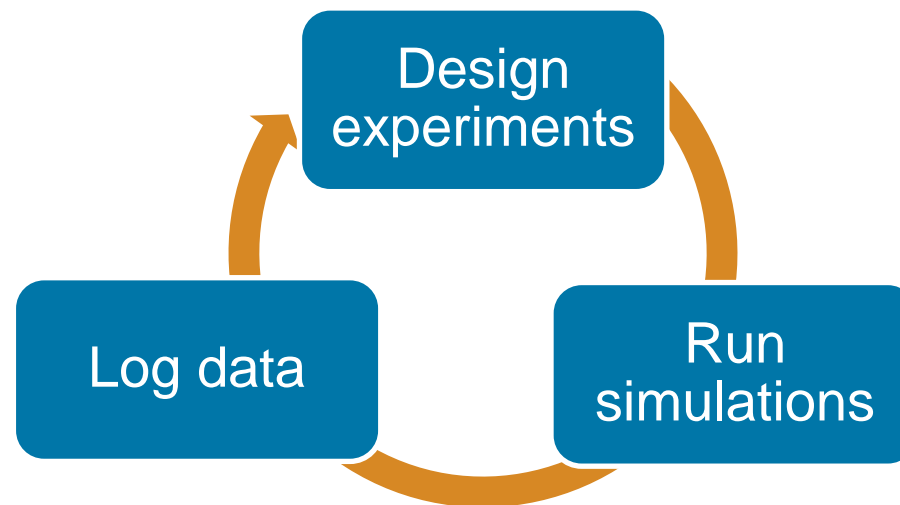


Input features

Ambient Temperature
Ambient Pressure
Cooling Temperature

Response

Max Displacement



Data Preparation

AI Modeling

Simulation & Test

Deployment

合成数据生成

Design of Experiments (实验设计)

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]
  
```

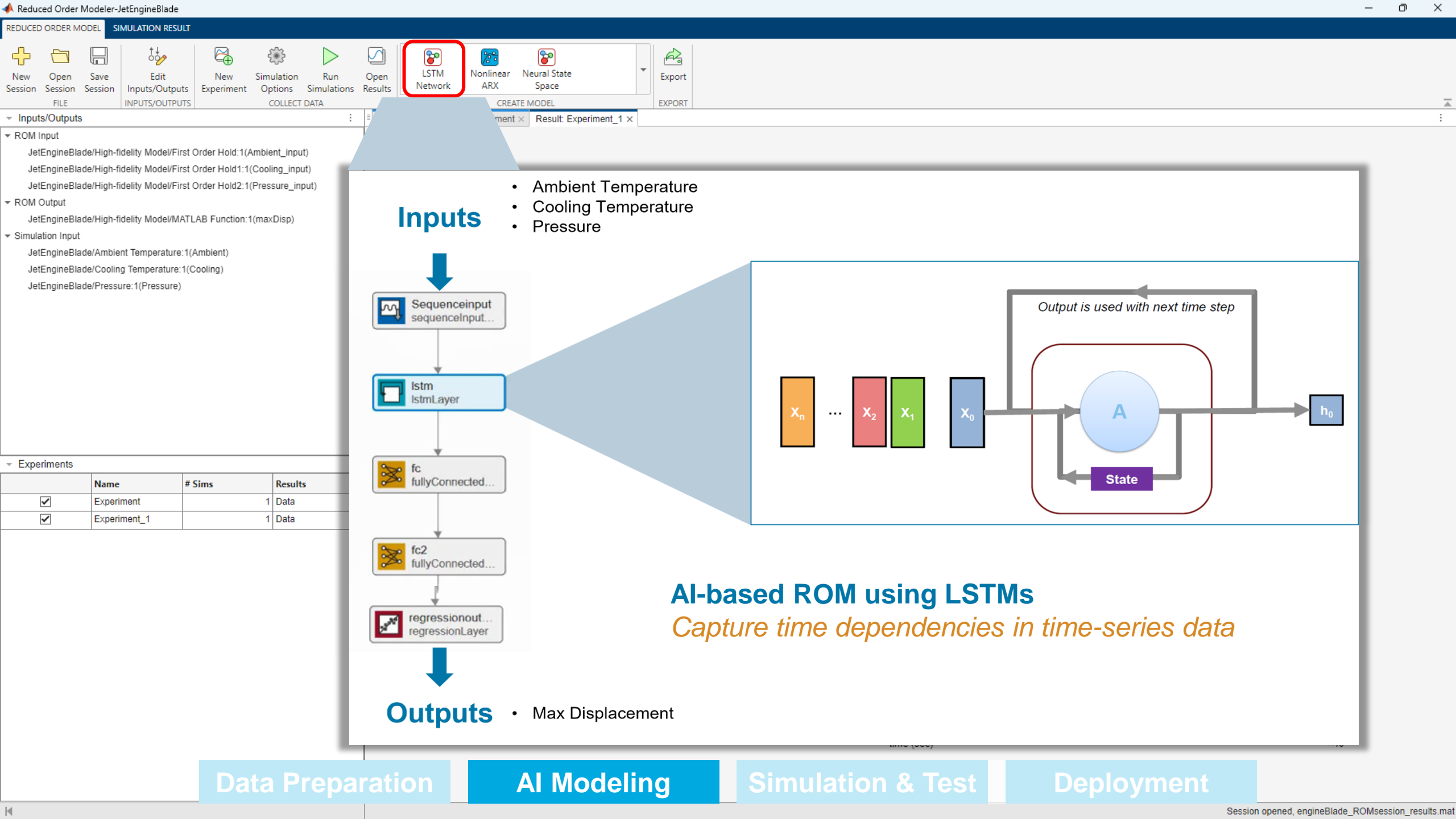
Command Window
fx >>

Data Preparation

AI Modeling

Simulation & Test

Deployment



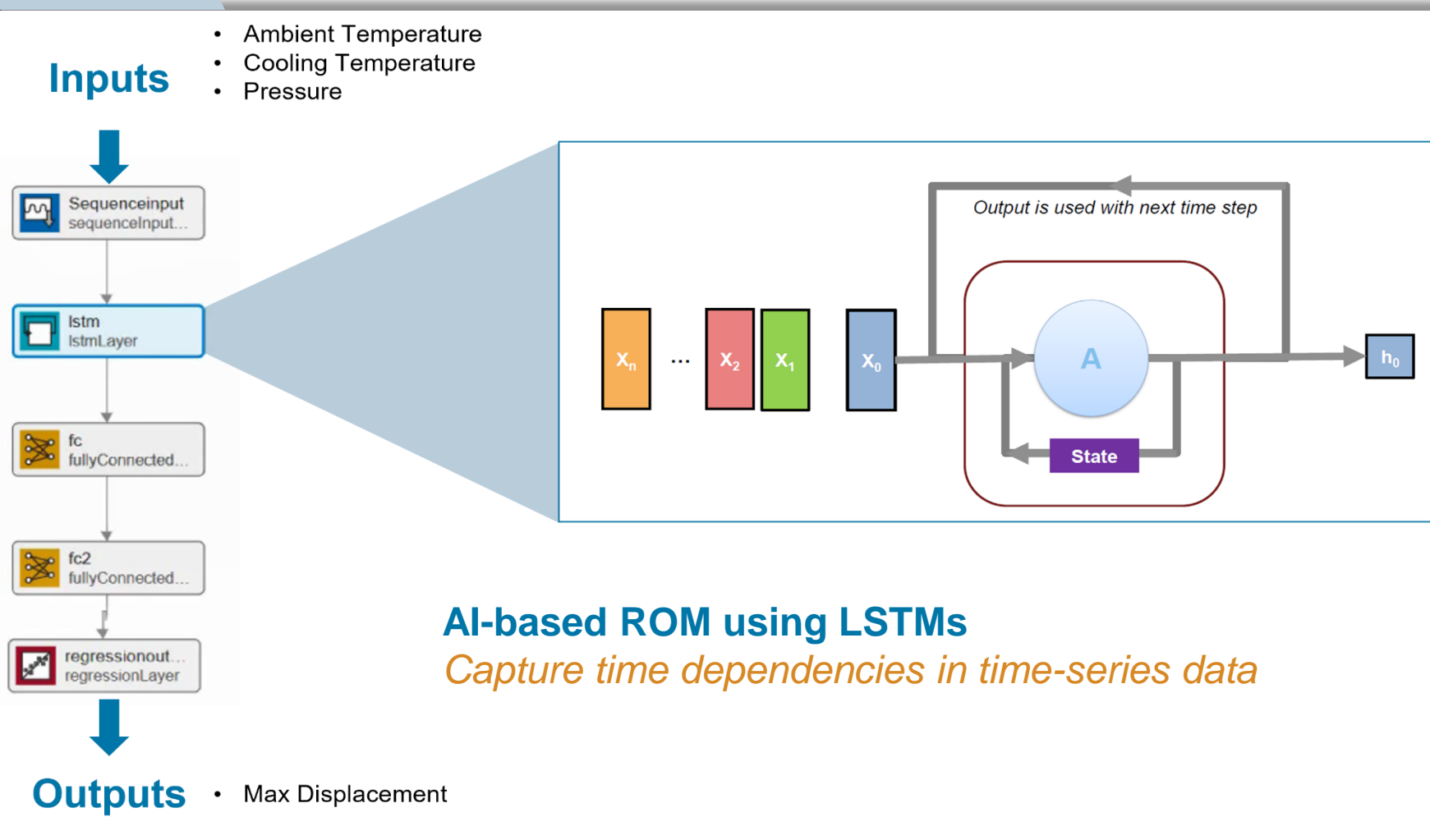
Toolbar icons: New Session, Open Session, Save Session, Edit Inputs/Outputs, New Experiment, Simulation Options, Run Simulations, Open Results, LSTM Network (highlighted), Nonlinear ARX, Neural State Space, Export.

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



Simulation Result 1 of 1

Show output only

Show as scatter plot

SIMULATION RESULTS OPTIONS

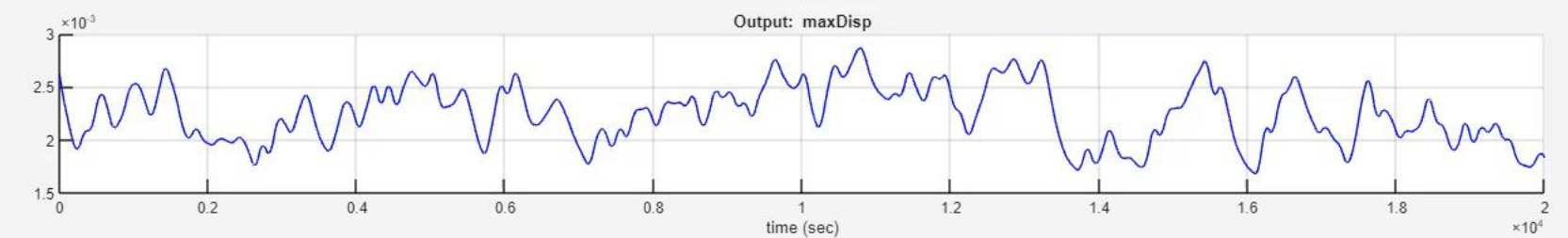
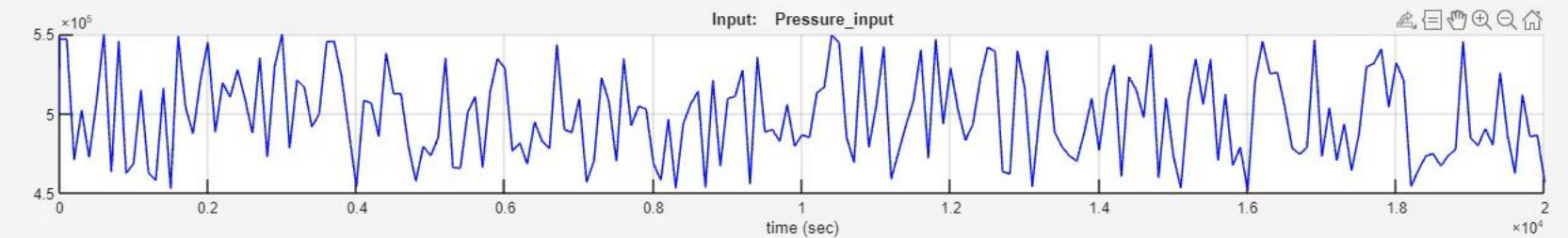
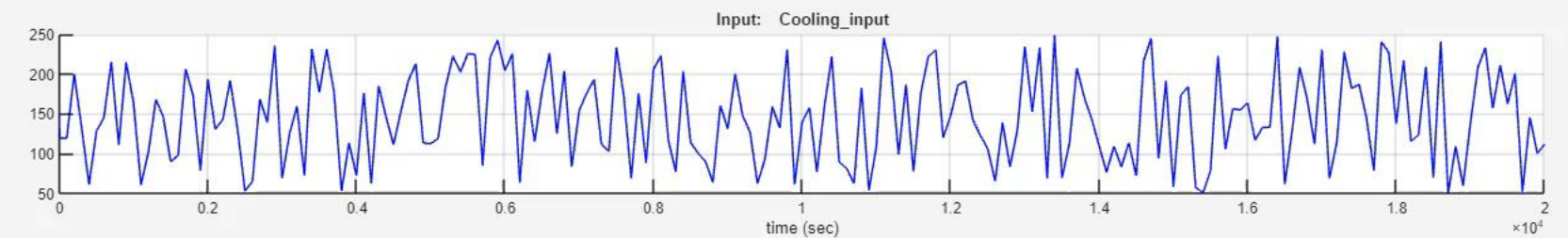
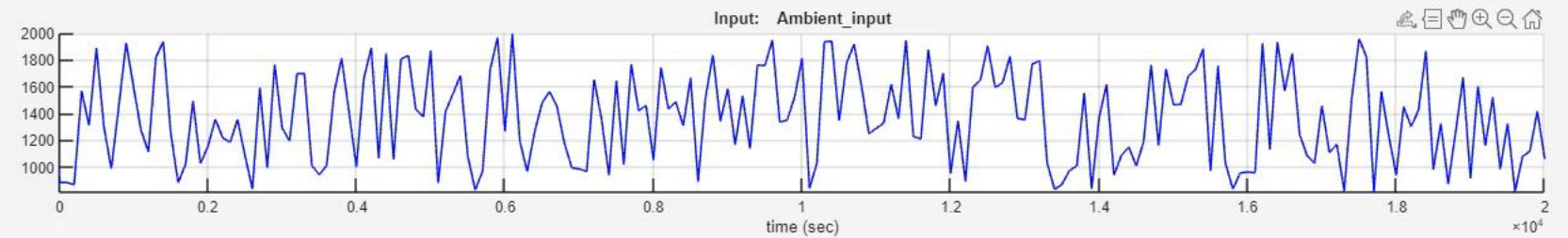
Inputs/Outputs

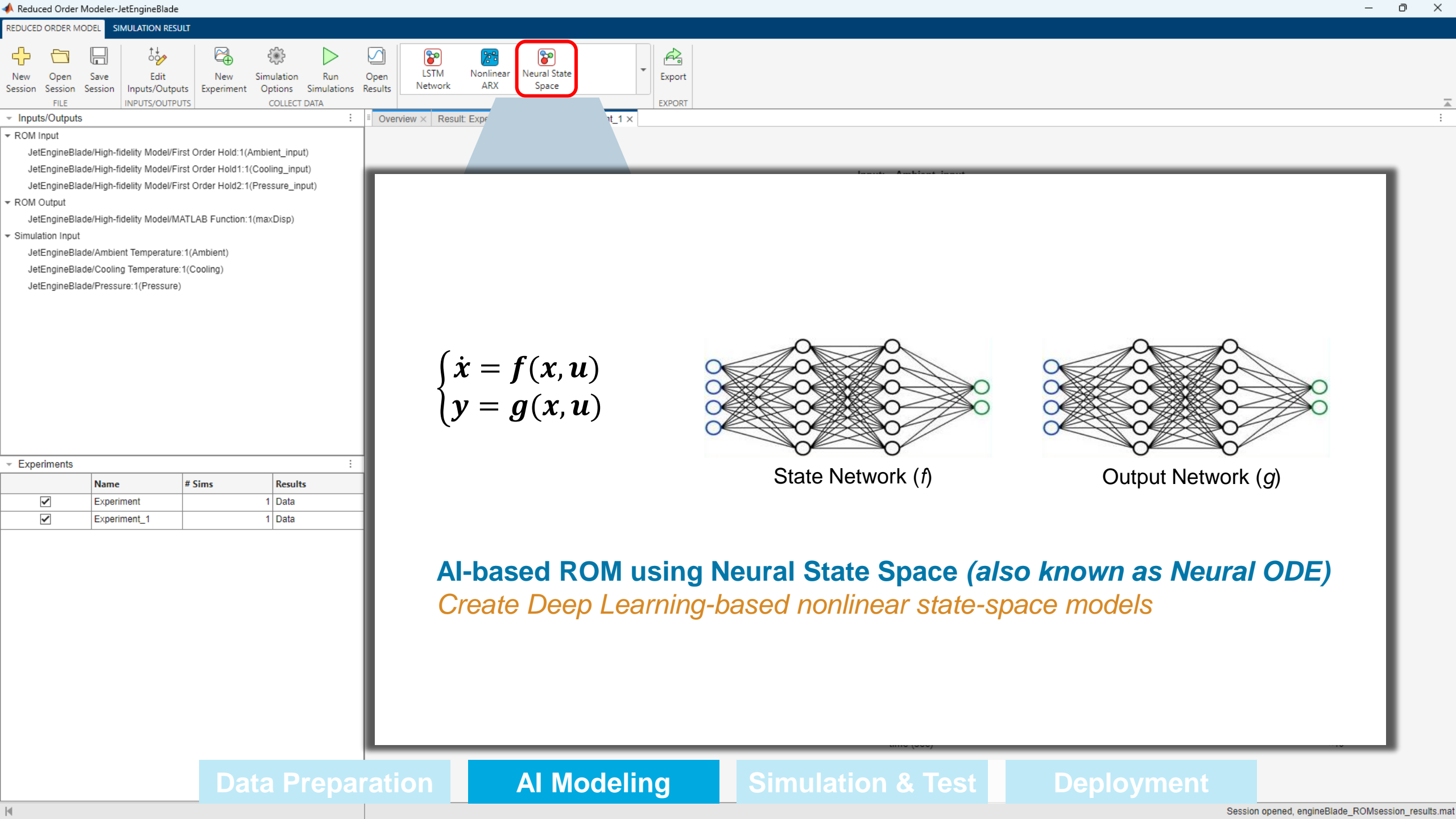
- ROM Input
 - JetEngineBlade/High-fidelity Model/First Order Hold:1(Ambient_input)
 - JetEngineBlade/High-fidelity Model/First Order Hold1:1(Cooling_input)
 - JetEngineBlade/High-fidelity Model/First Order Hold2:1(Pressure_input)
- ROM Output
 - JetEngineBlade/High-fidelity Model/MATLAB Function:1(maxDisp)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

Experiments

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

Overview × Result: Experiment × Result: Experiment_1 × Experiment_1 ×





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

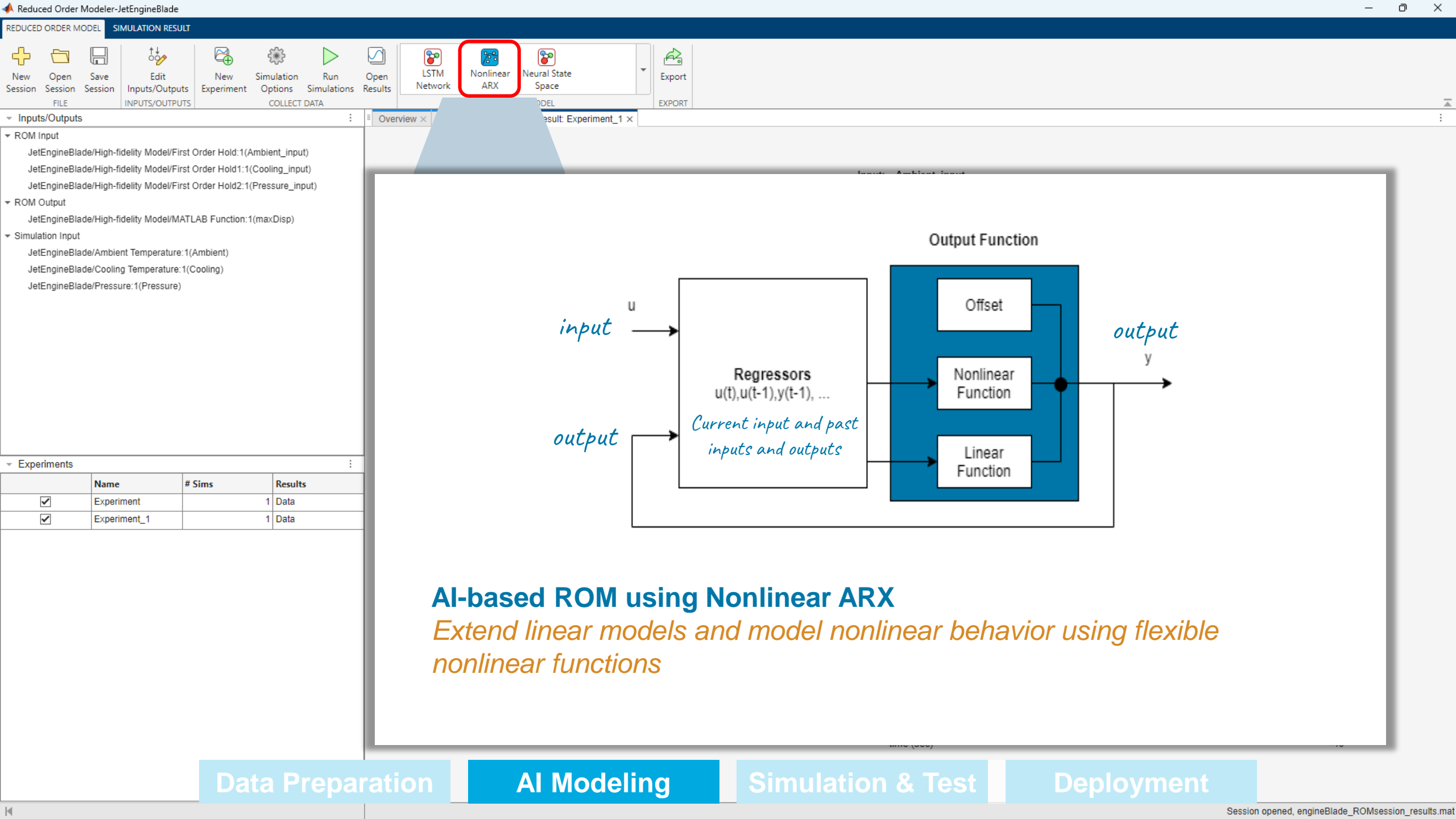
	Name	# Sims	Results
<input checked="" type="checkbox"/>	Experiment		1 Data
<input checked="" type="checkbox"/>	Experiment_1		1 Data

Data Preparation

AI Modeling

Simulation & Test

Deployment



AI-based ROM using Nonlinear ARX

Extend linear models and model nonlinear behavior using flexible nonlinear functions

Data Preparation

AI Modeling

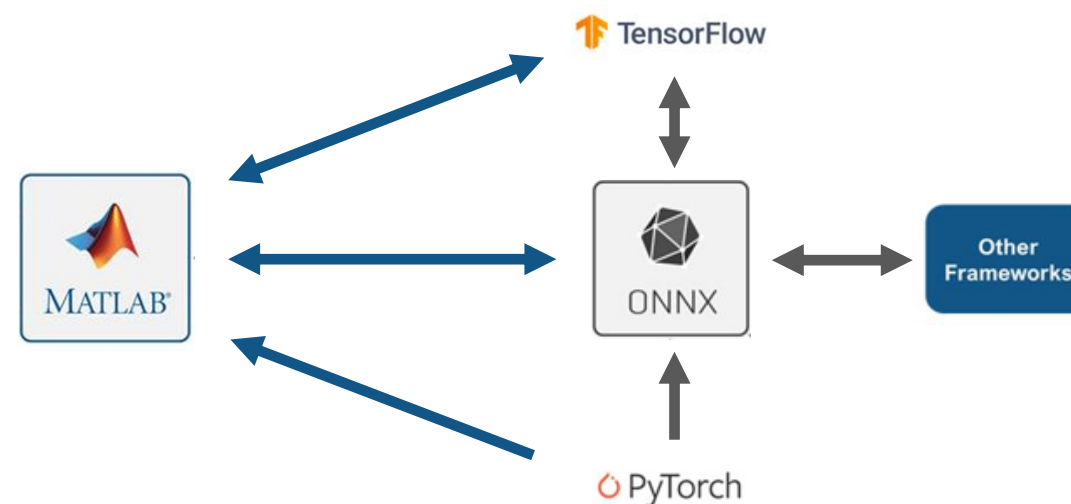
Simulation & Test

Deployment

MATLAB 和其他AI框架互操作

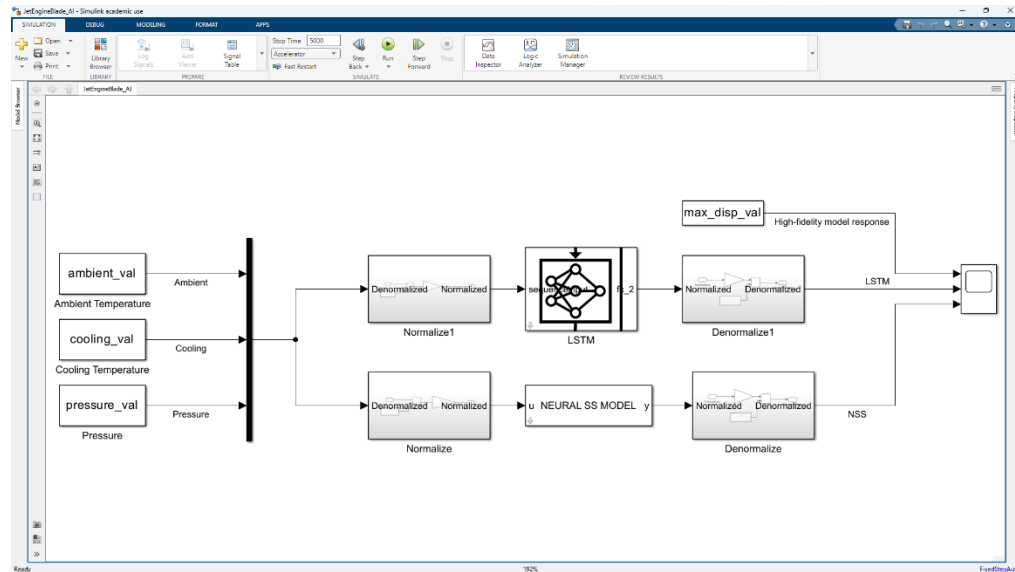
框架互操作性构建了数据科学、工程和生产之间的桥梁

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

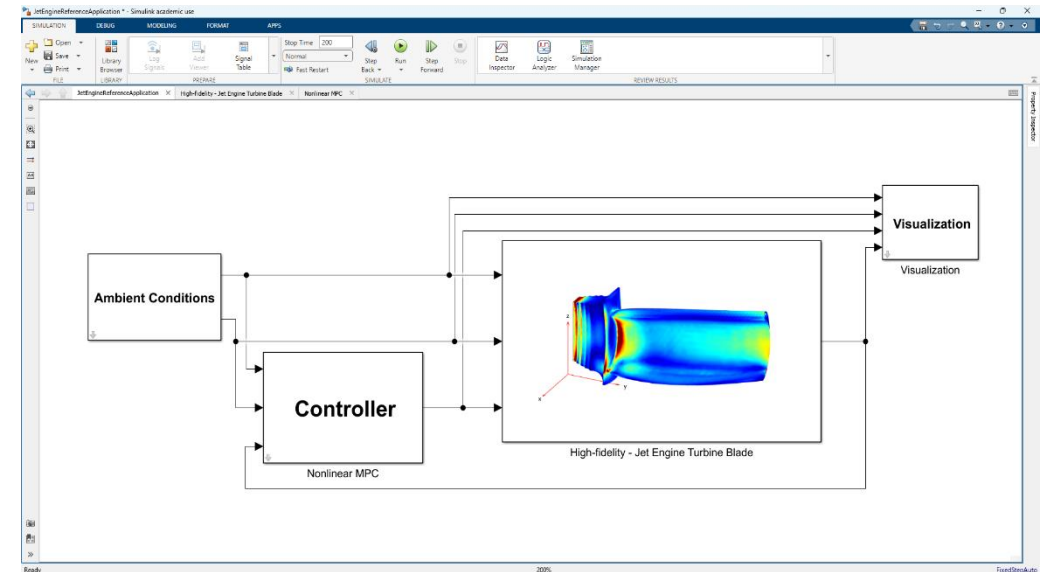


集成AI模型进行系统级仿真和测试

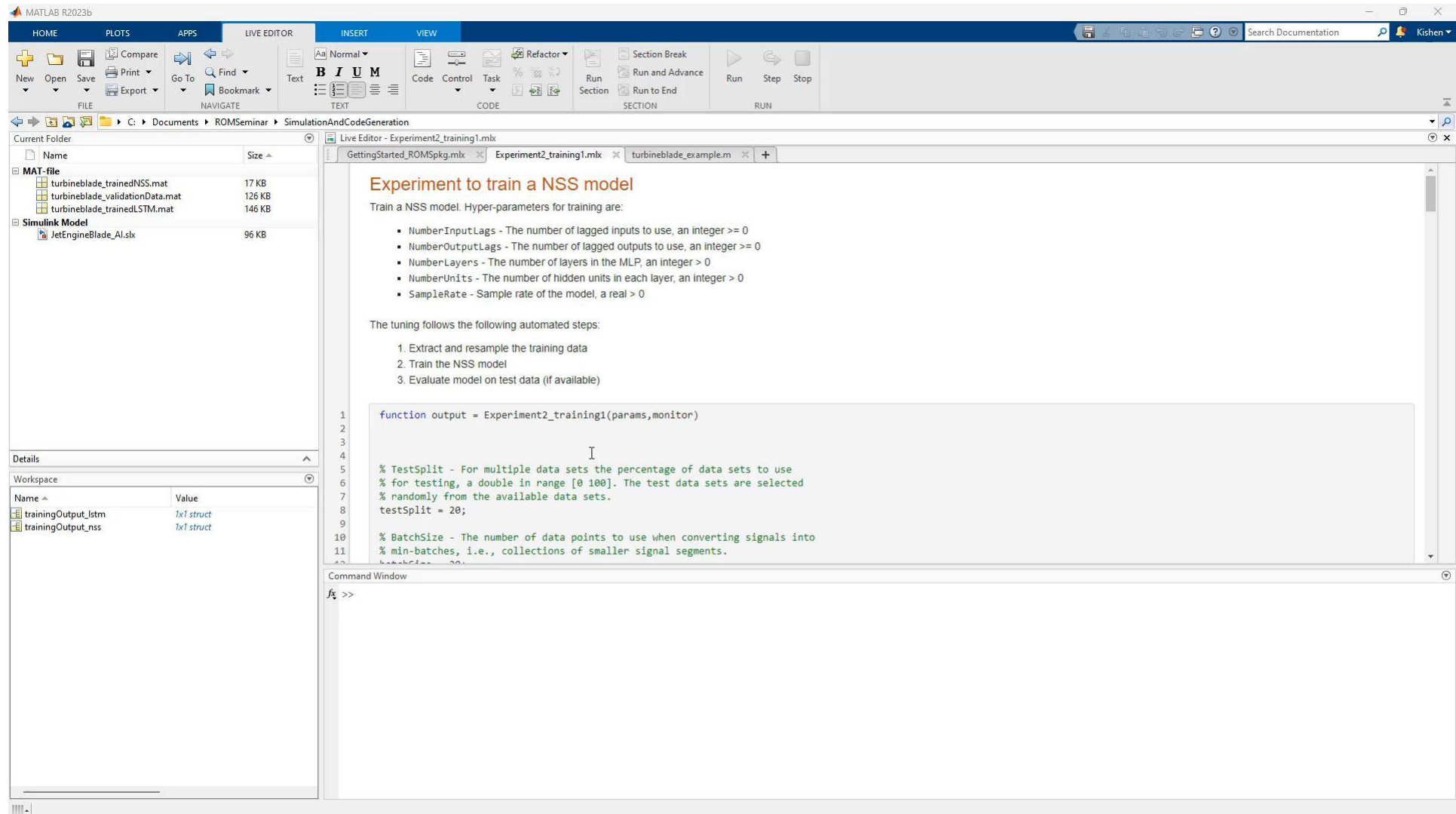
Integration of trained AI model into Simulink



System-level simulation



将训练好的AI模型集成到Simulink中



Current Folder: C:\Documents \ ROMSeminar \ SimulationAndCodeGeneration

Live Editor - Experiment2_training1.mlx

GettingStarted_ROMSpkg.mlx | Experiment2_training1.mlx | turbineblade_example.m | Experiment2_training1.mlx

Experiment to train a NSS model

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

- NumberInputLags - The number of lagged inputs to use, an integer ≥ 0
- NumberOutputLags - The number of lagged outputs to use, an integer ≥ 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)

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

Workspace

Name	Value
trainingOutput_lstm	1x1 struct
trainingOutput_nss	1x1 struct

Command Window

```
f >>
```

Data Preparation

AI Modeling

Simulation & Test

Deployment

将训练好的AI模型集成到Simulink中

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

Simulink中的AI blocks正在扩展，以包含更多的AI blocks，用于更多的应用程序

Specialized

Audio Toolbox

Computer Vision Toolbox

AI Core

Statistics and Machine Learning Toolbox

Deep Learning Toolbox

System Identification Toolbox

Deep Learning Toolbox Verification Library

保证深度学习网络的鲁棒和可信

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

Data Preparation

AI Modeling

Simulation & Test [Deep Learning Toolbox Verification Library](#)

The screenshot shows the product page for the Deep Learning Toolbox Verification Library. The page has a blue header with 'Products and Services' and a search bar. The main content area features the product title 'Deep Learning Toolbox Verification Library' and a sub-headline 'Ensure robustness and reliability of deep neural networks'. A 'Download support package' button is visible. Below this, there is a paragraph of text explaining the library's purpose and a list of capabilities. A small icon of a neural network with a checkmark is also present.

Products and Services

Deep Learning Toolbox Verification Library

Ensure robustness and reliability of deep neural networks

[Download support package](#)

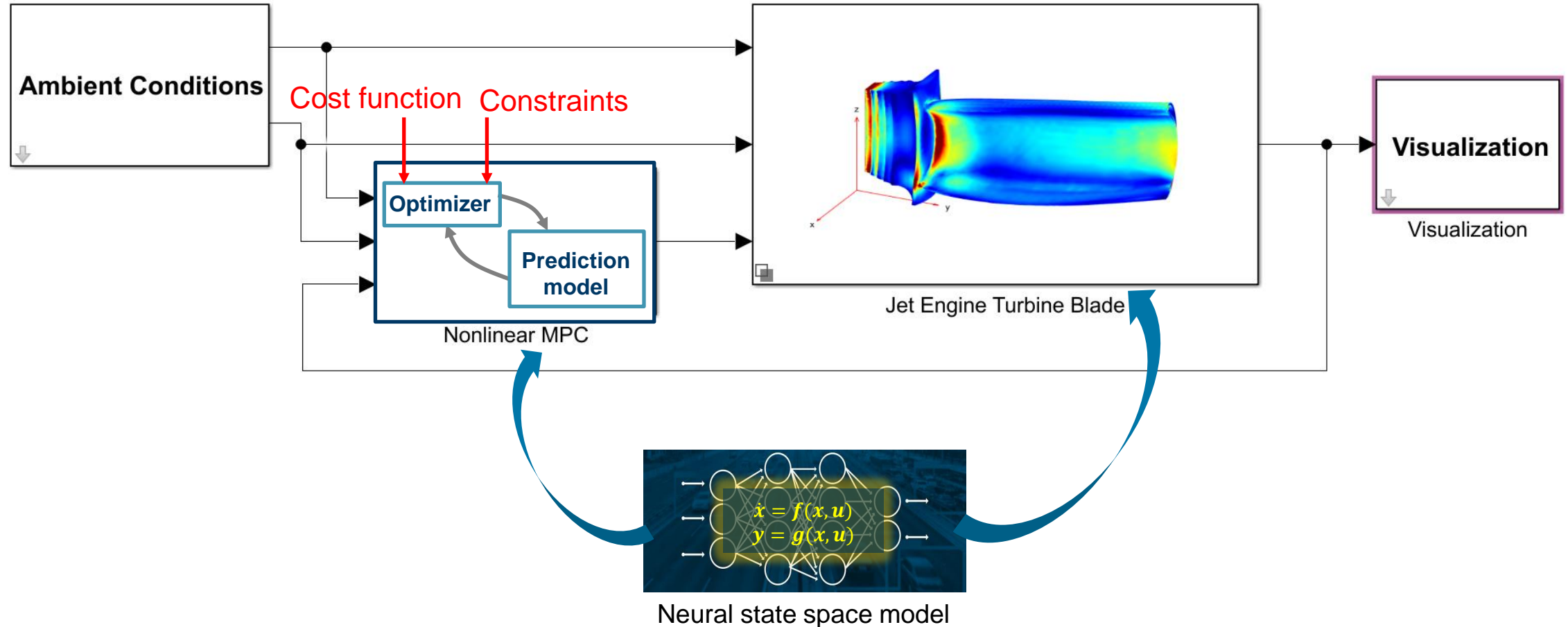
As deep neural networks become part of engineered systems, particularly safety-critical applications, it is crucial to ensure their reliability and robustness. Deep Learning Toolbox Verification Library lets you rigorously assess and test deep neural networks.

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

构建模型预测控制器 Model Predictive Controller

SIMULINK®



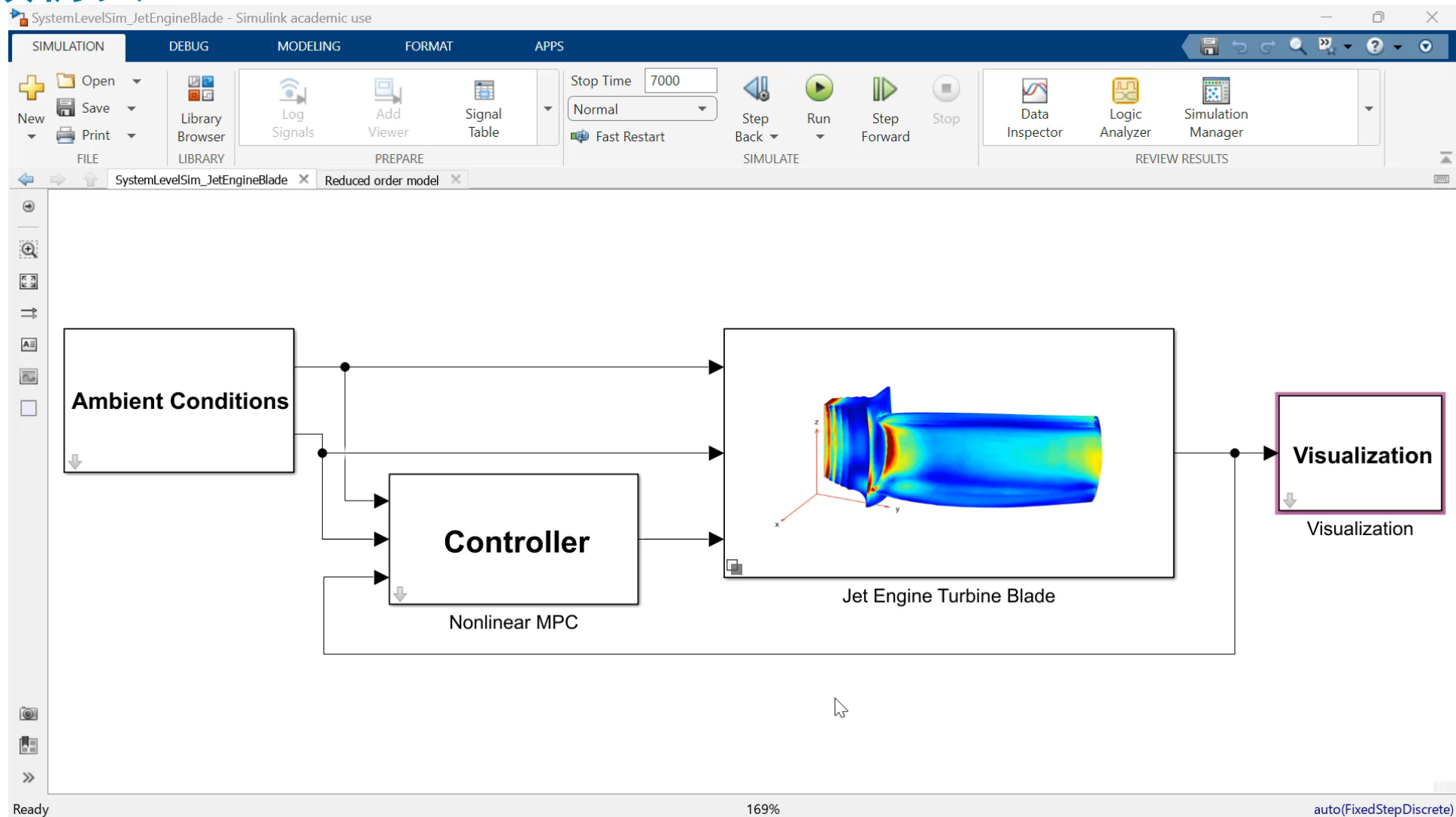
Data Preparation

AI Modeling

Simulation & Test

Deployment

系统级仿真



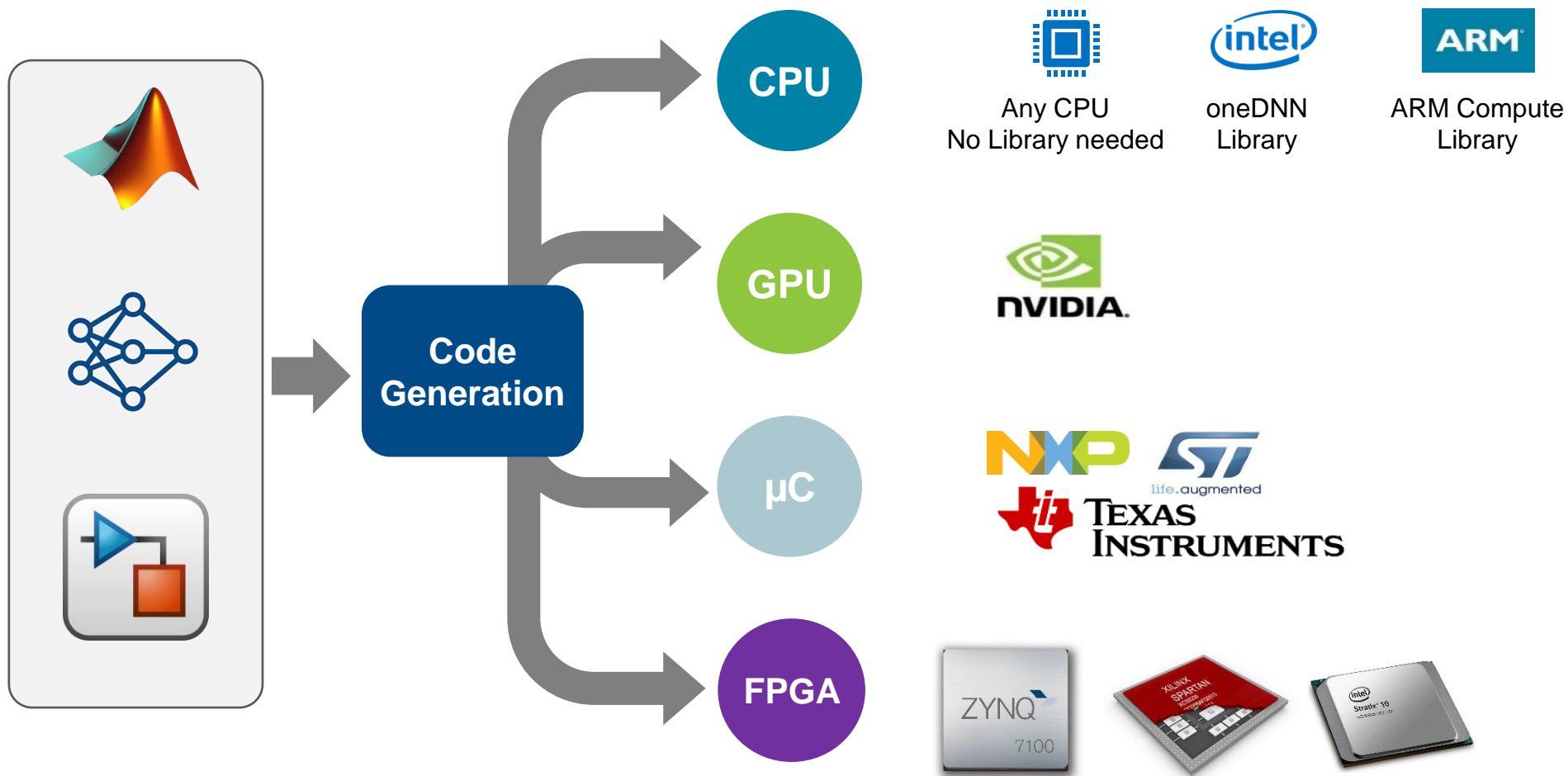
Data Preparation

AI Modeling

Simulation & Test

Deployment

Deploy to target with zero coding errors



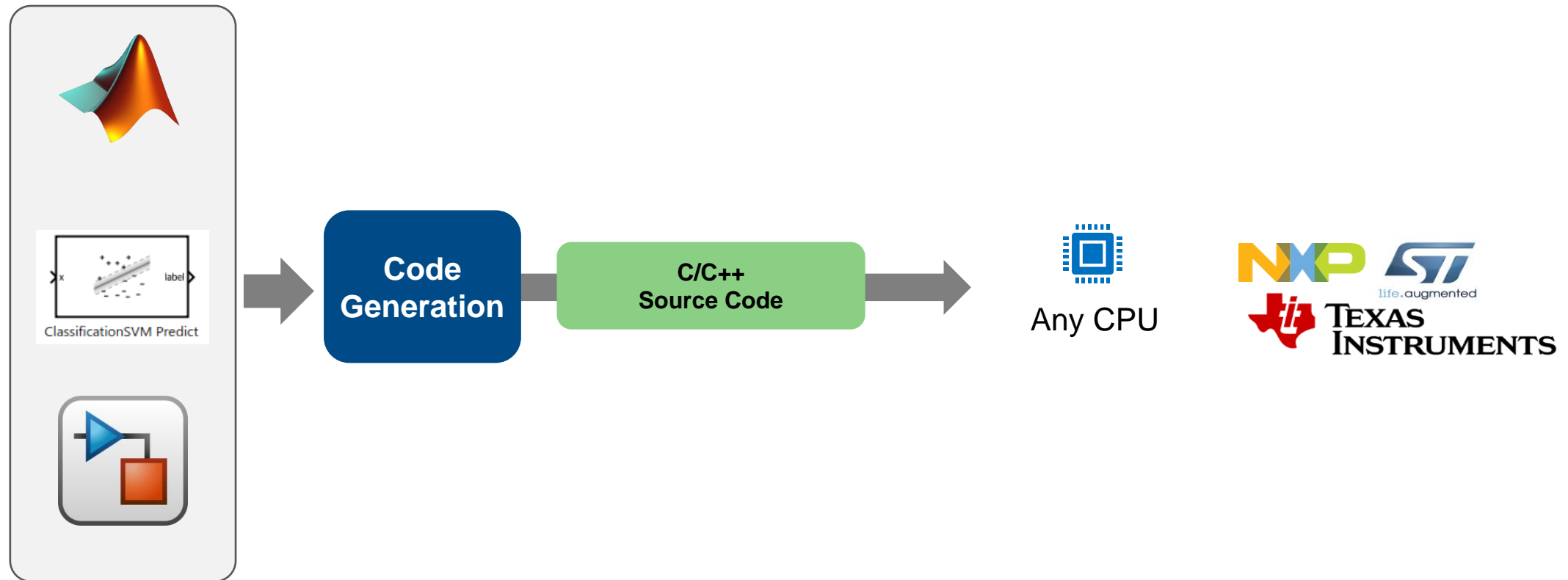
Data Preparation

AI Modeling

Simulation & Test

Deployment

Use Embedded Coder to Generate Code for Machine Learning



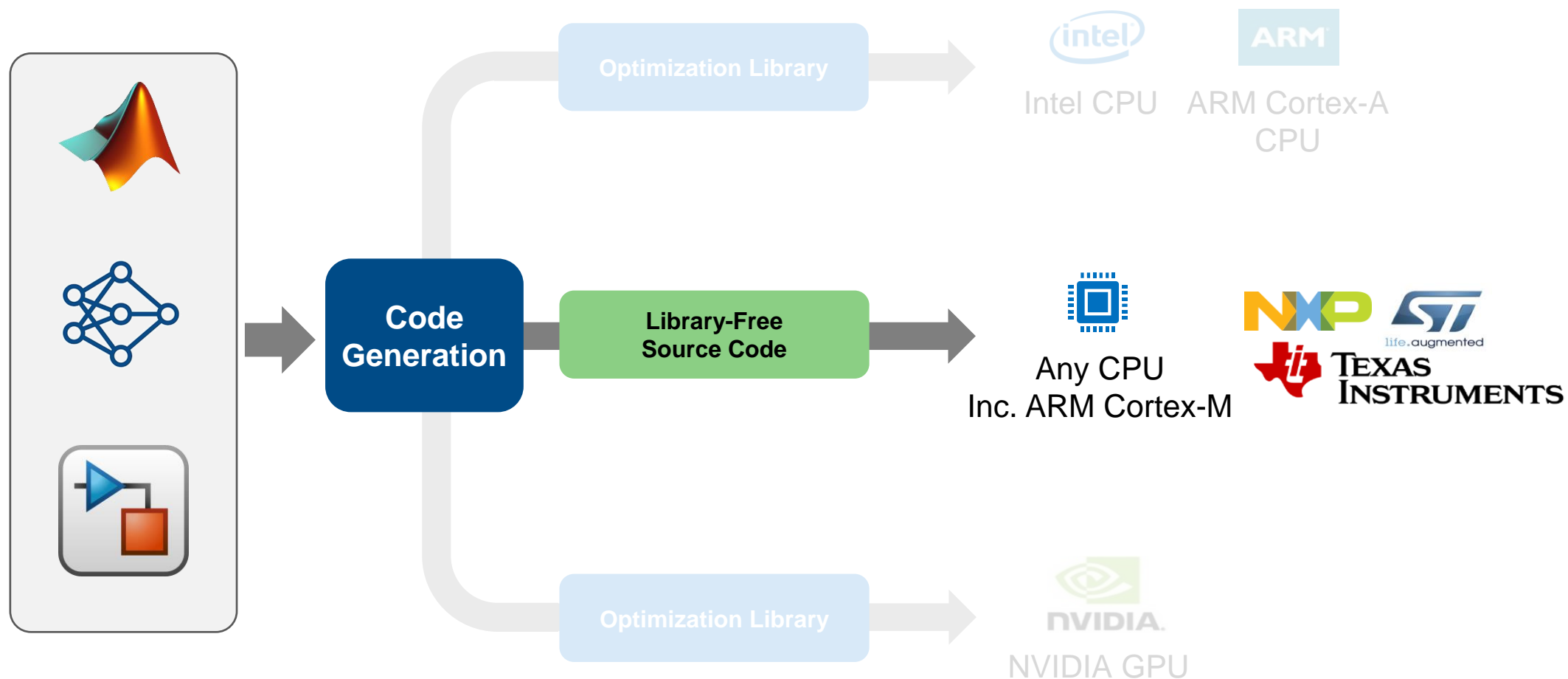
Data Preparation

AI Modeling

Simulation & Test

Deployment

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



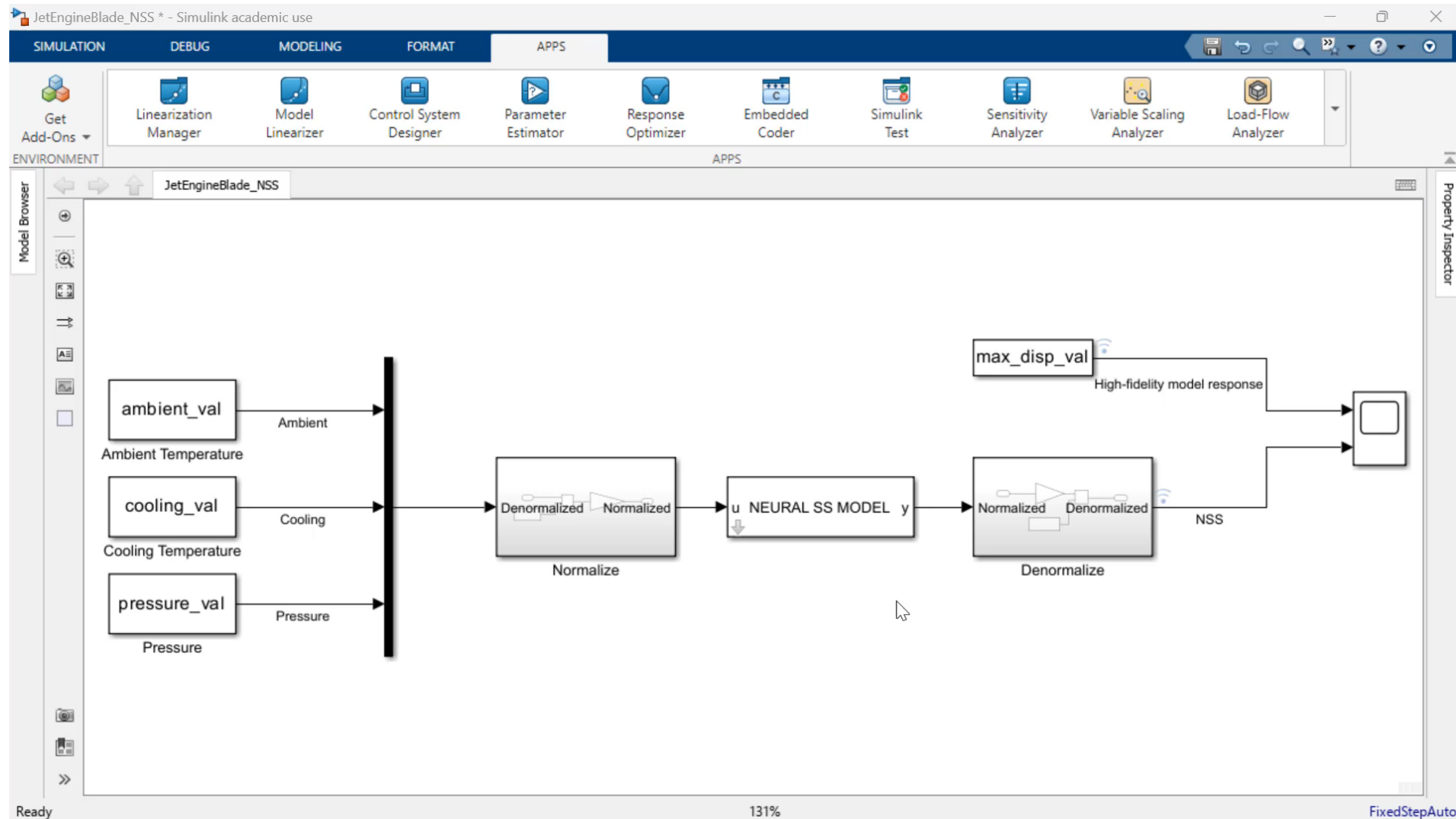
Data Preparation

AI Modeling

Simulation & Test

Deployment

Generate Library-Free C Code for Deep Learning Networks



Data Preparation

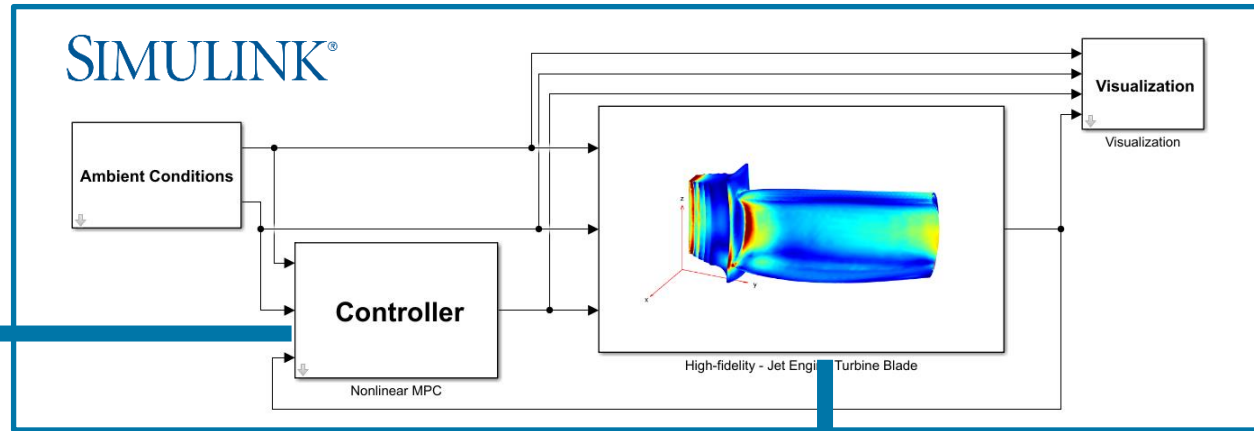
AI Modeling

Simulation & Test

Deployment

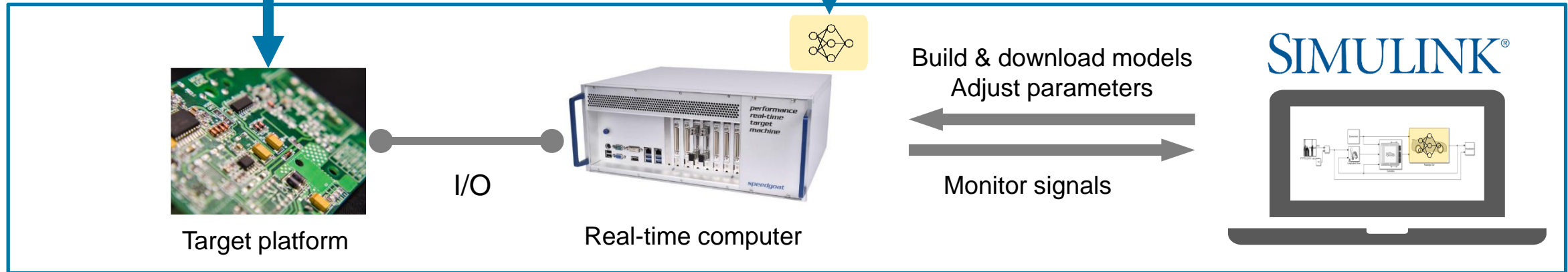
Hardware-in-the-loop 仿真

System-level integration and test



Code generation from algorithm

Code generation from plant model



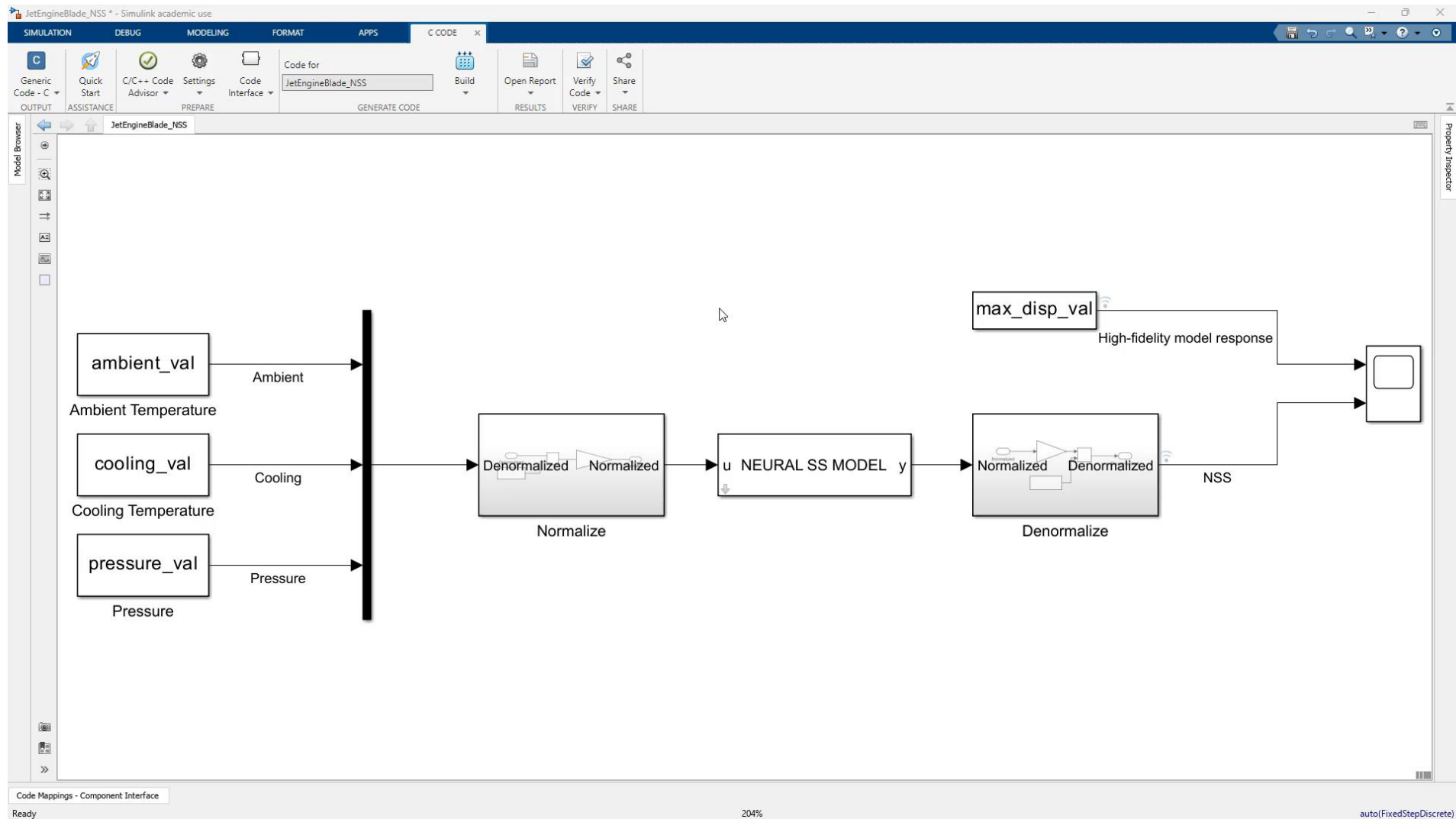
Data Preparation

AI Modeling

Simulation & Test

Deployment

Hardware-in-the-loop 仿真



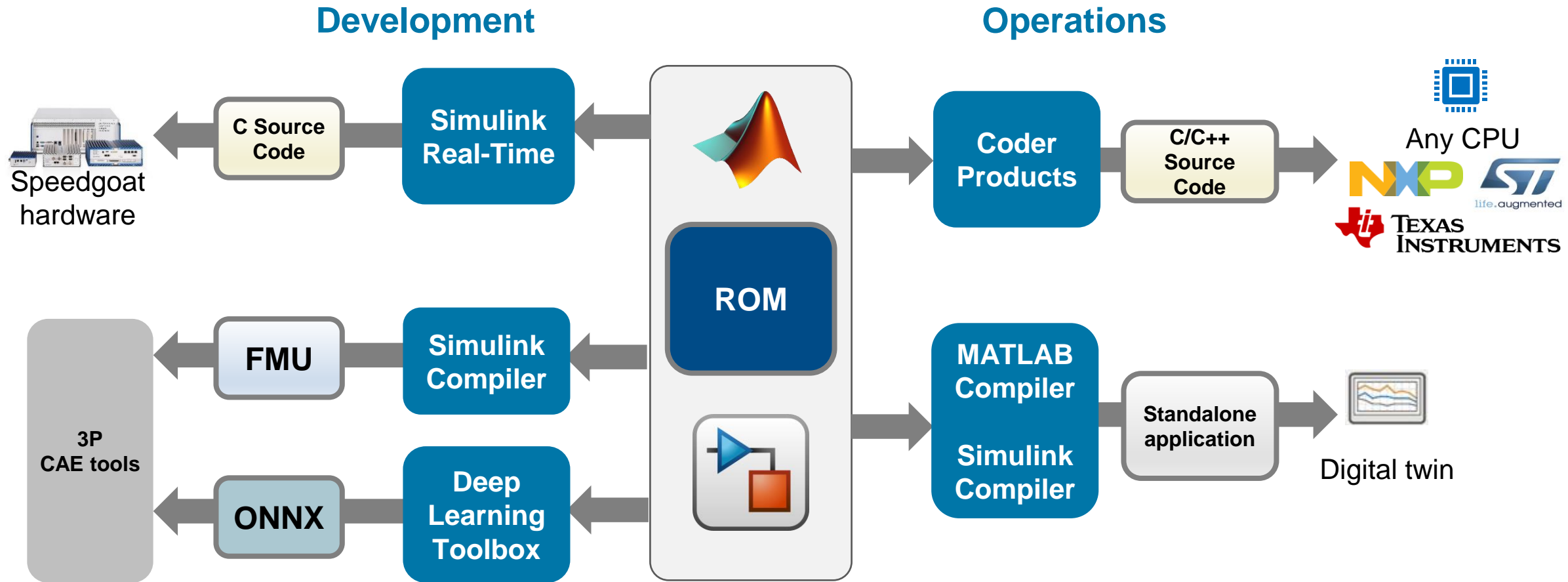
Data Preparation

AI Modeling

Simulation & Test

Deployment

在其他环境（开发或生产）中使用降阶后的模型



Data Preparation

AI Modeling

Simulation & Test

Deployment

SUBARU利用AI Surrogate Model减少变速器控制系统分析时间

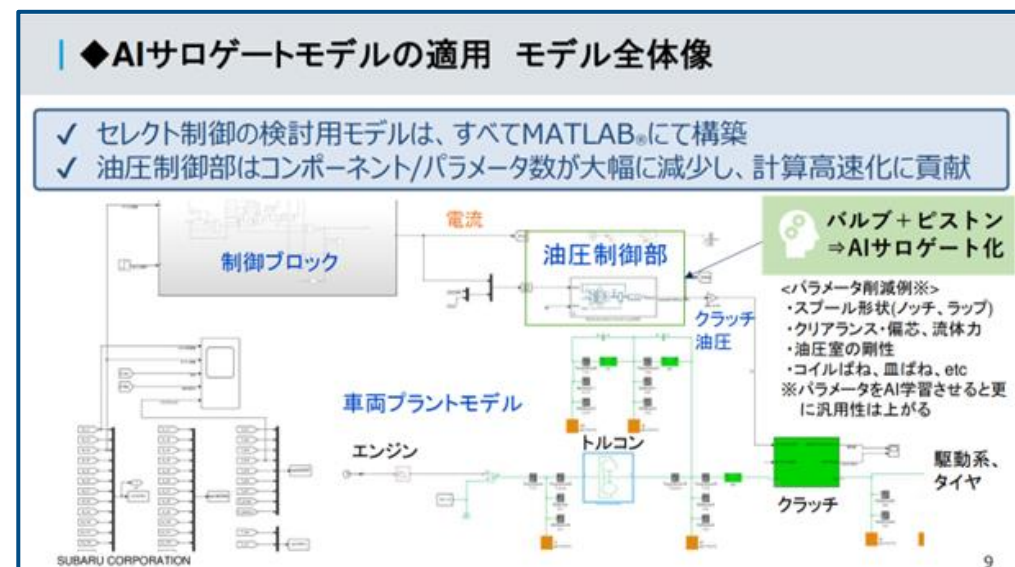
在自动挡车辆中，变速器液压控制系统调节液压流体的流量，确保在不同行驶工况下平稳的换挡和高效的动力传递。

方案

- 该 AI 替代模型是通过 MATLAB® 使用神经 ODE 模型构造的。应用这种 AI 替代模型比之前使用第三方一维物理模型进行分析大幅缩短了计算时间。

关键成果

- 与原来的一维模型相比，计算时间减少 99%
- 在 MATLAB 中构造的 AI 替代模型可以重现具有任意电流、油温和源压力读数的波形
- 准确重现波形，即使在模型未经训练的油温范围内也是如此



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.

总结

增效

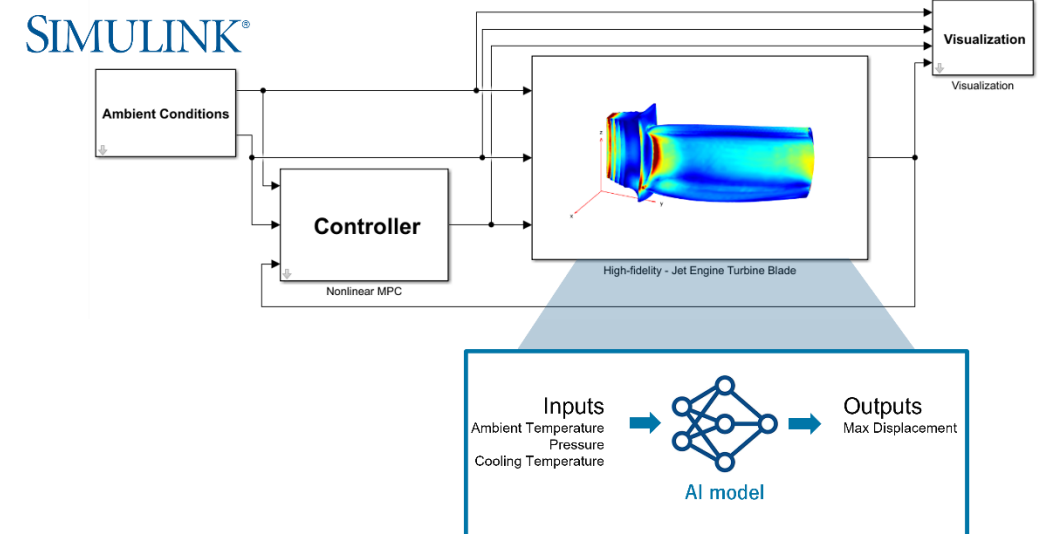
系统级仿真、硬件在环(HIL)测试、非线性控制设计和虚拟传感器建模

探索

MATLAB中不同的ROM的技术来找到效果最好的。

Reduced Order Modeler App

- **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

Thank you



© 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.

