

2022 MathWorks 中国汽车年会

数据驱动的动态系统建模-模型降阶之路

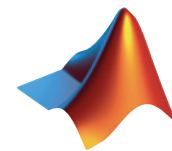
刘海伟, MathWorks 中国





Agenda

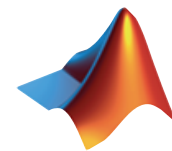
- 动态系统与数据驱动模型简介
- 基于AI的降阶模型开发与集成
- MathWorks的AI与MBD技术支持





Agenda

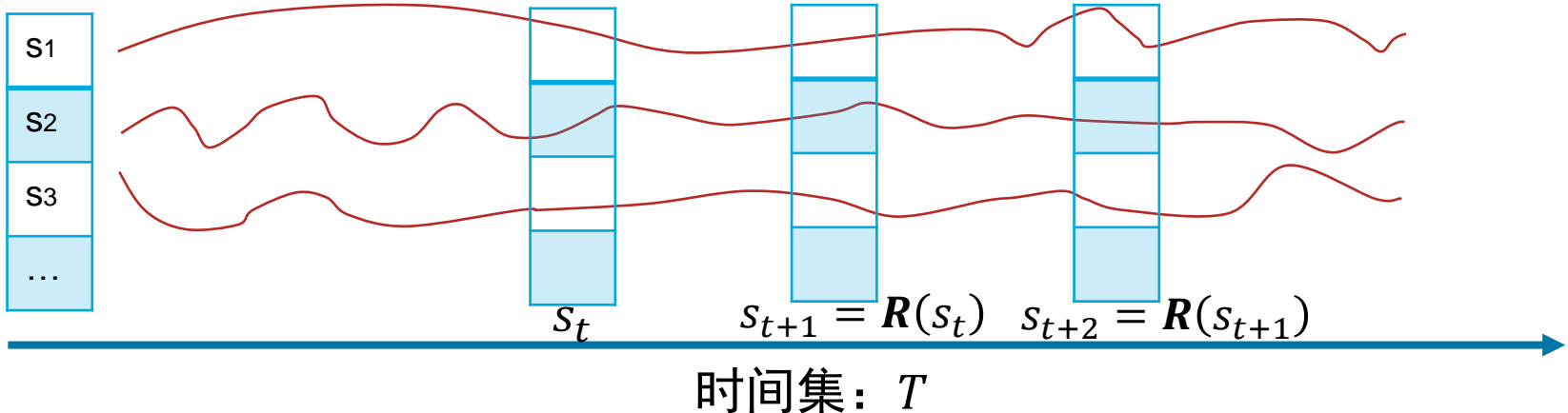
- 动态系统与数据驱动模型简介
- 基于AI的降阶模型开发与集成
- MathWorks的AI与MBD技术支持



动态系统

状态空间: S

$$R: S \times T \rightarrow S$$



```

x0 = [2; 0];
A = [-0.1 -1; 1 -0.1];
trueModel = @(t,y) A*y + B*controlsignal(t,ft,f);
% 定义系统函数，此处就是一个状态空间方程
numTimeSteps = 2000;
T = 15;
odeOptions = odeset(RelTol=1.e-7);
t = linspace(0, T, numTimeSteps);
[~, xTrain] = ode45(trueModel, t, x0,
odeOptions);

```

简单的线性系统

$$x(t)' = Ax(t) + Bu(t)$$

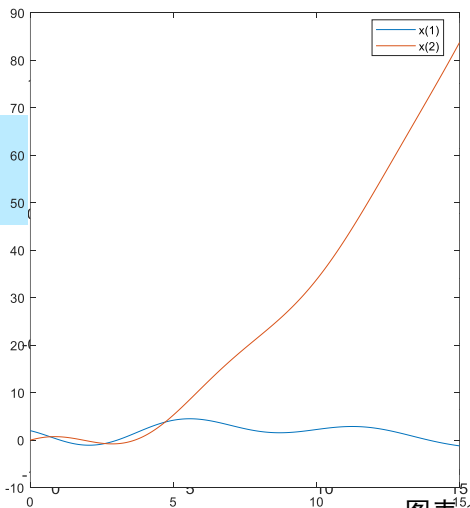
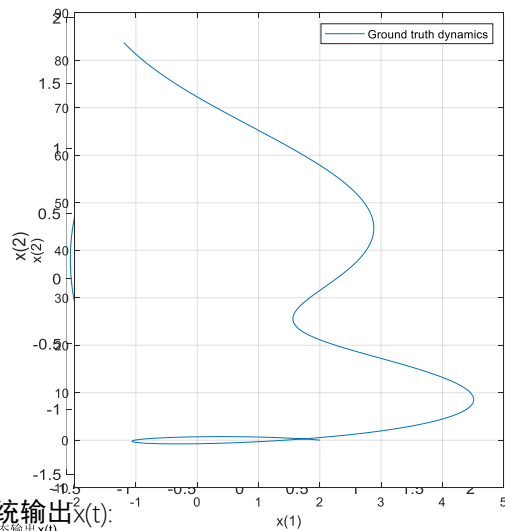


图 表 1 系 统 输 出 $x(t)$



何为数据驱动的动力系统建模

旋转体的状态空间模型 (线性系统)

$$J \frac{d\omega}{dt} + F\omega = T$$

$$y = \omega$$

$$\frac{dx}{dt} = Ax + Bu$$

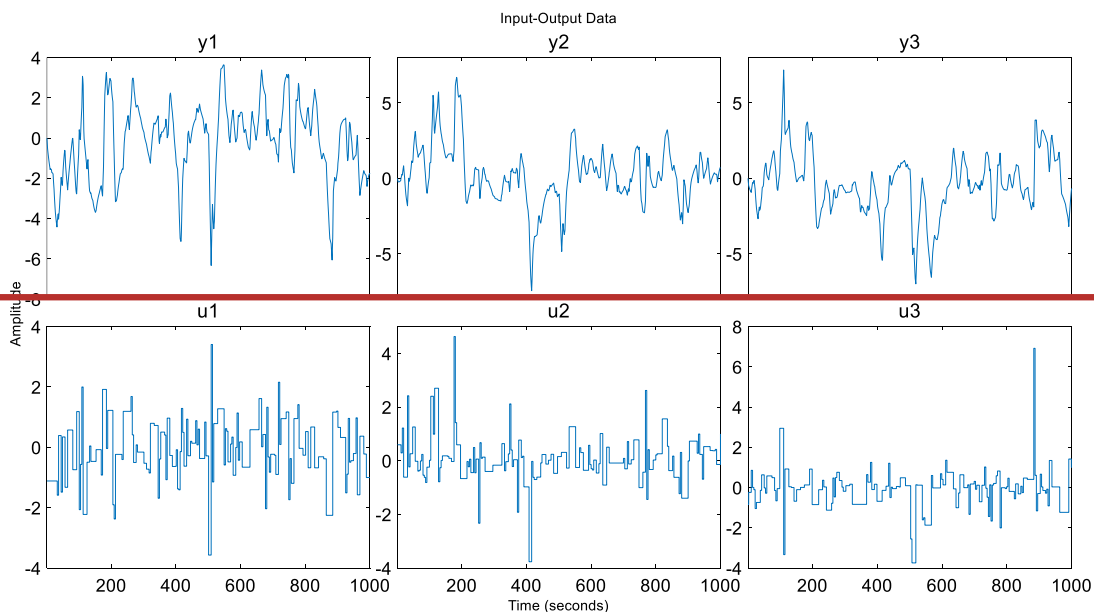
$$y = Cx + Du$$

$$A = -J^{-1}F, B = J^{-1}, C = I, D = 0$$

$$\frac{d\omega}{dt} = -J^{-1}F\omega + J^{-1}T$$

$$y = \omega$$

```
sys_mimo = ss(A,B,C,D);
[y,t] = lsim(sys_mimo,u,t);
```



数据驱动的动力系统 (Dynamical System) 建模(一): 深度学习

Original 刘海伟 MATLAB 2022-04-06 18:00

收录于合集

#MATLAB 数据科学与人工智能

78个 >



动力系统建模被各领域广泛应用, 例如电动汽车, 能源系统, 航空航天。我们本文提到动力系统主要是被控对象, 对被控对象进行建模是因为我们希望了解这个系统 (被控对象) 的物理特性以及接受一些外部输入 (力, 扭矩, 电流等等) 时会有什么样的动态响应, 基于此从而可以更好的给出控制输入得到我们期望的系统的输出, 以及理解系统的退化或最大化提升系统效率。

这些动力系统的行为是由多物理场复杂的交互作用决定的, 因此系统行为和系统响应建模通常需要复杂的第一原理支撑, 仿真时也需要大量的计算(例如有限元模型)。

这也是本文的出发点, 提供数据驱动 (主要介绍深度学习和系统辨识) 的模型降阶

(Reduced Order Modeling) 提速的方法, 通过数据得到具有一定保真度的数据模型, 在捕捉到系统动态特性的同时也提升仿真速度。

本文中涉及多个 demo, 数据以及脚本文件, 若您感兴趣进一步获取这些链接, 可以在文

记录编号

数据驱动的动力系统 (Dynamical System) 建模(二): 系统辨识

Original 刘海伟 MATLAB 2022-04-20 18:00

收录于合集

#MATLAB 数据科学与人工智能

78个 >





我们在上一篇深度学习用于动力系统建模 (点击跳转) 的文章中针对动力系统的特性与数据驱动的动机进行了论述。我们介绍了动力系统当前输出不仅依赖于当前的输入, 还依赖于系统过去的行为 (历史输入和历史输出)。我们也介绍了什么场景下使用深度学习/系统辨识来进行系统建模。

本文我们主要介绍数据驱动的另一主题: 系统辨识。

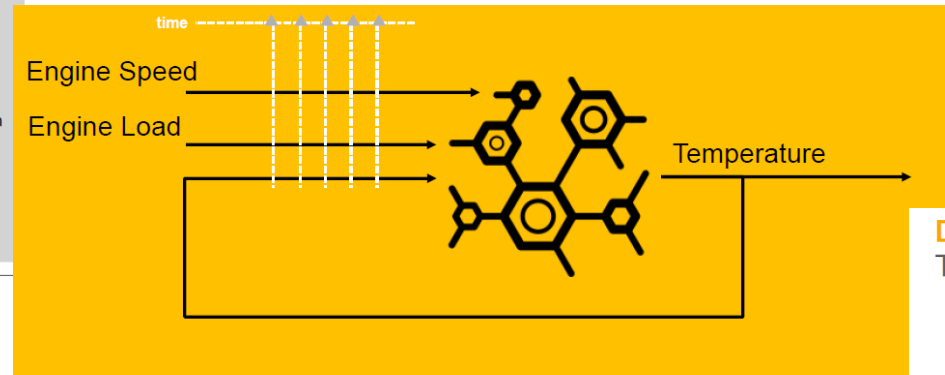
为了更好地理解, 我们可以设计一个简单的线性系统[链接1], 更具体的是一个连续时间状态空间模型, 来解释系统辨识的适用场景:

Continental 使用动态神经网络用于估计发动机温度

Classical ECU Functions Advantages/Disadvantes

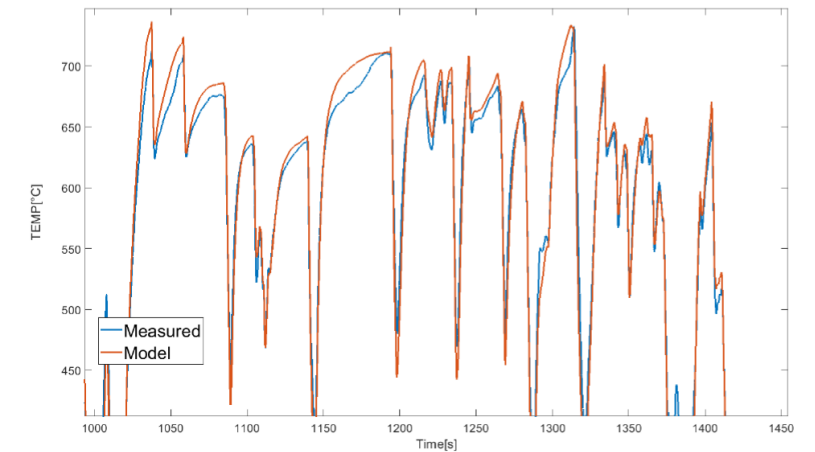
Advantages	Disadvantage
Physically motivated High understanding of whats going on (intermediate signals have typically physical units) Enabling "transfer learning" for single HW change 	<ul style="list-style-type: none"> › Require development (modelling + coding) › Require methodology development for calibration = training › Require tooling for the training (backpropagation) › Require very special measurements from engine test bench 


Electrification & Data Analytics Internal
11.04.19 Michael Wutz, © Continental AG




Electrification & Data Analytics Internal
11.04.19 Michael Wutz, © Continental AG

Deep Dynamical Systems Temperature example 40min of driving (validation)




Electrification & Data Analytics Internal
11.04.19 Michael Wutz, © Continental AG

Renault 使用 Deep Learning Networks 来估计 NO_x 排放: LSTM

挑战

Design, simulate, and improve aftertreatment systems to reduce oxides of nitrogen (NO_x) emissions

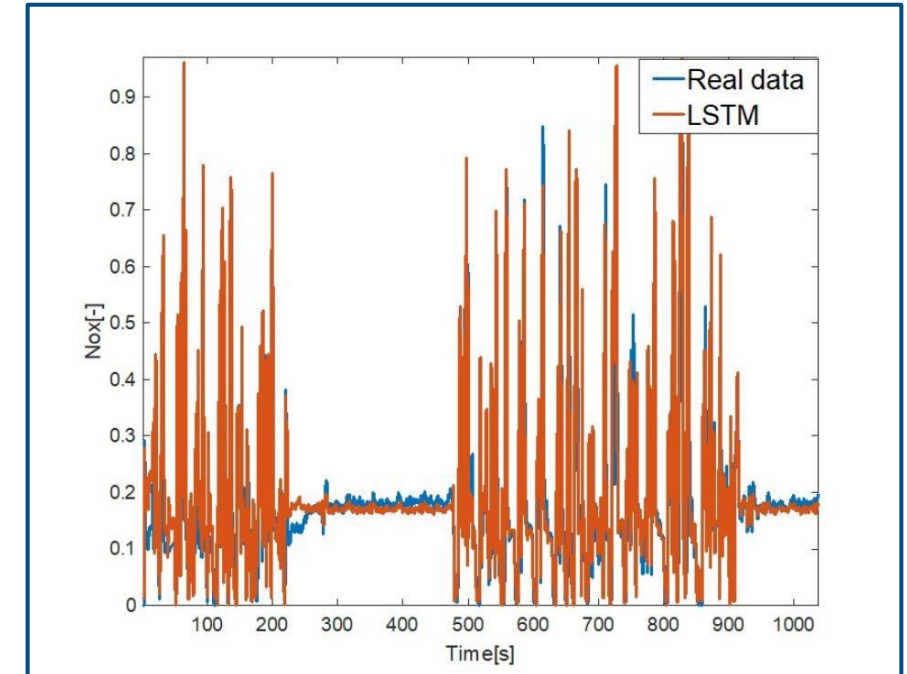
方案

Use MATLAB and Deep Learning Toolbox to model engine-out NO_x emissions using a long short-term memory (LSTM) network

结果

NO_x emissions predicted with close to 90% accuracy

- LSTM network incorporated into aftertreatment simulation model
- Code generated directly from network for ECU deployment



Measured NO_x emissions from an actual engine and modeled NO_x emissions from the LSTM network.

“Even though we are not specialists in deep learning, using MATLAB and Deep Learning Toolbox we were able to create and train a network that predicts NO_x emissions with almost 90% accuracy.”

- Nicoleta-Alexandra Stroe, Renault

AI驱动的系统建模

Data Preparation



Data cleansing and preparation



Human insight



Simulation-generated data

Modeling & Training



Model design and tuning



Hardware accelerated training



Interoperability

Simulation & Test



Integration with complex systems



System simulation



System verification and validation

Deployment



Embedded devices



Enterprise systems



Edge, cloud, desktop

使用AI模型进行动态系统建模

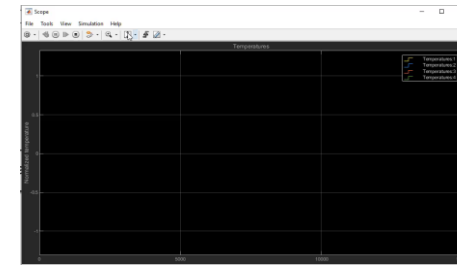
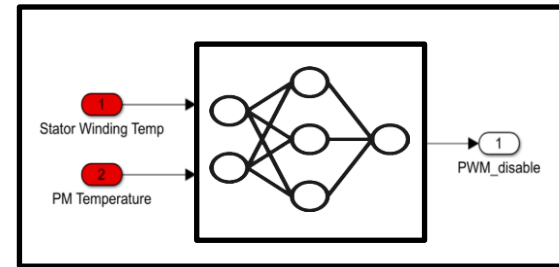
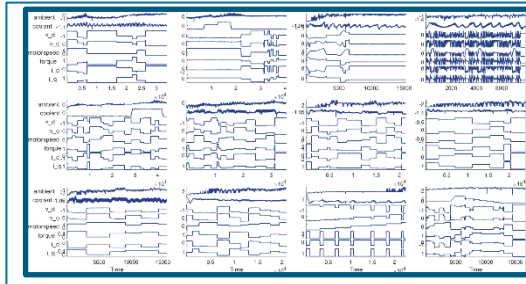
Data Preparation

Modeling & Training

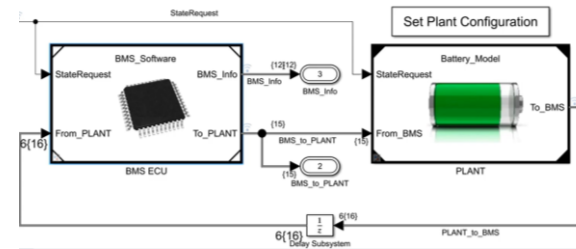
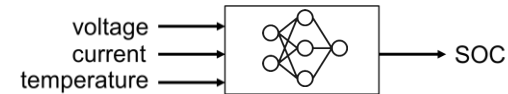
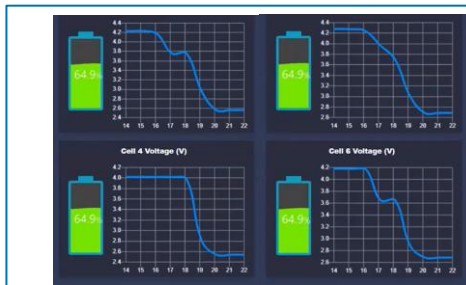
Simulation & Test

Deployment

电机温度预测



电池SOC估计



MATLAB中AI相关工具箱的发展

2016 - 2019

Toolboxes

- Deep Learning Toolbox
- Text Analytics Toolbox
- Reinforcement Learning
- Predictive Maintenance Toolbox

Code Generation

- GPU Coder
- MATLAB Coder

Apps

- Image Labeler
- Deep Network Designer
- Video Labeler
- Signal Labeler

Interoperability

- TensorFlow-Keras Importer
- ONNX Support

2020 - 2021

Apps

- Experiment Manager
- Lidar Labeler
- Reinforcement Learning Designer

Compression

- Quantization

Code Generation

- Deep Learning HDL Coder

Model-Based Design

- Image Classification & Model Prediction
- Recurrent Neural Networks
- Object Detectors

Interoperability

- TensorFlow Model Importer

2022

Accessibility

- Deep Learning Model Hub

Apps

- Data Cleaner

Compression

- Taylor Pruning

Code Generation

- TensorFlow Lite

Interoperability

- TensorFlow Lite inference for MATLAB and Simulink

Machine Learning

- Drift Detection

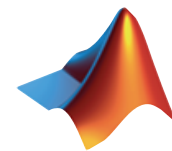
500+ examples, covering:

- **Computer Vision, LidarAudio, Signal and Wireless**
- **Computational Finance**
- **Controls**
- **Machine Learning**

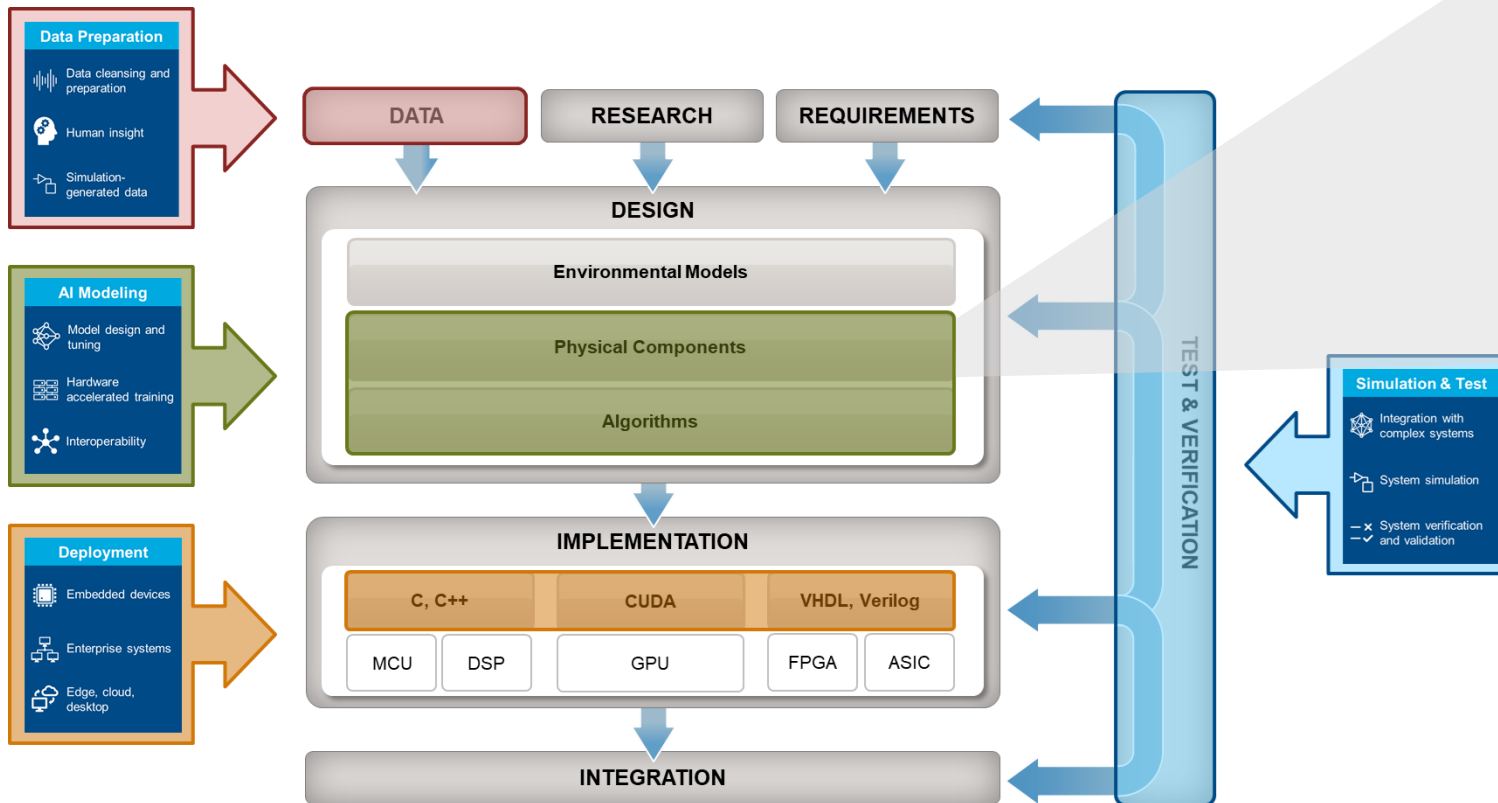


Agenda

- 动态系统与数据驱动模型简介
- 基于AI的降阶模型开发与集成
- MathWorks的AI与MBD技术支持



AI用于被控对象建模



AI用于被控对象建模

- 加速桌面与HIL仿真
- 当第一原理模型不能构建时通过数据进行动态系统（被控对象）建模

模型降阶

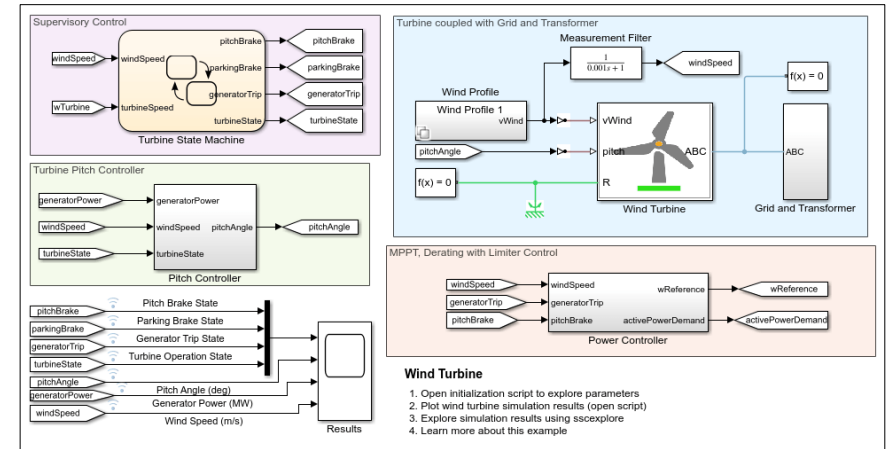
什么是模型降阶？

- 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

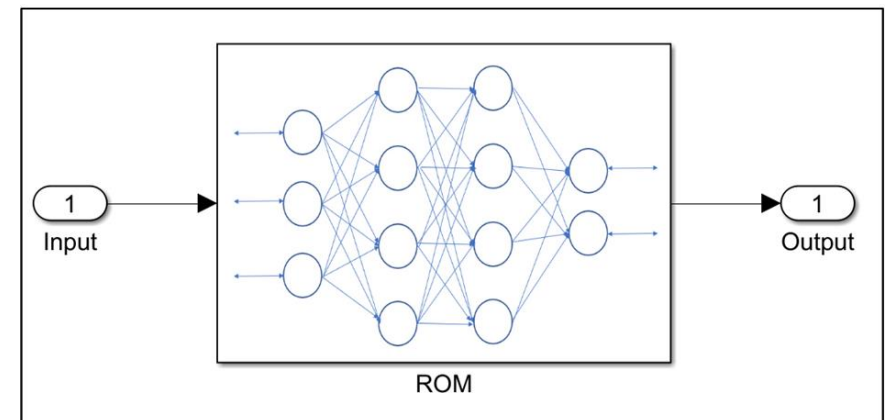
High-fidelity model



Simulation time

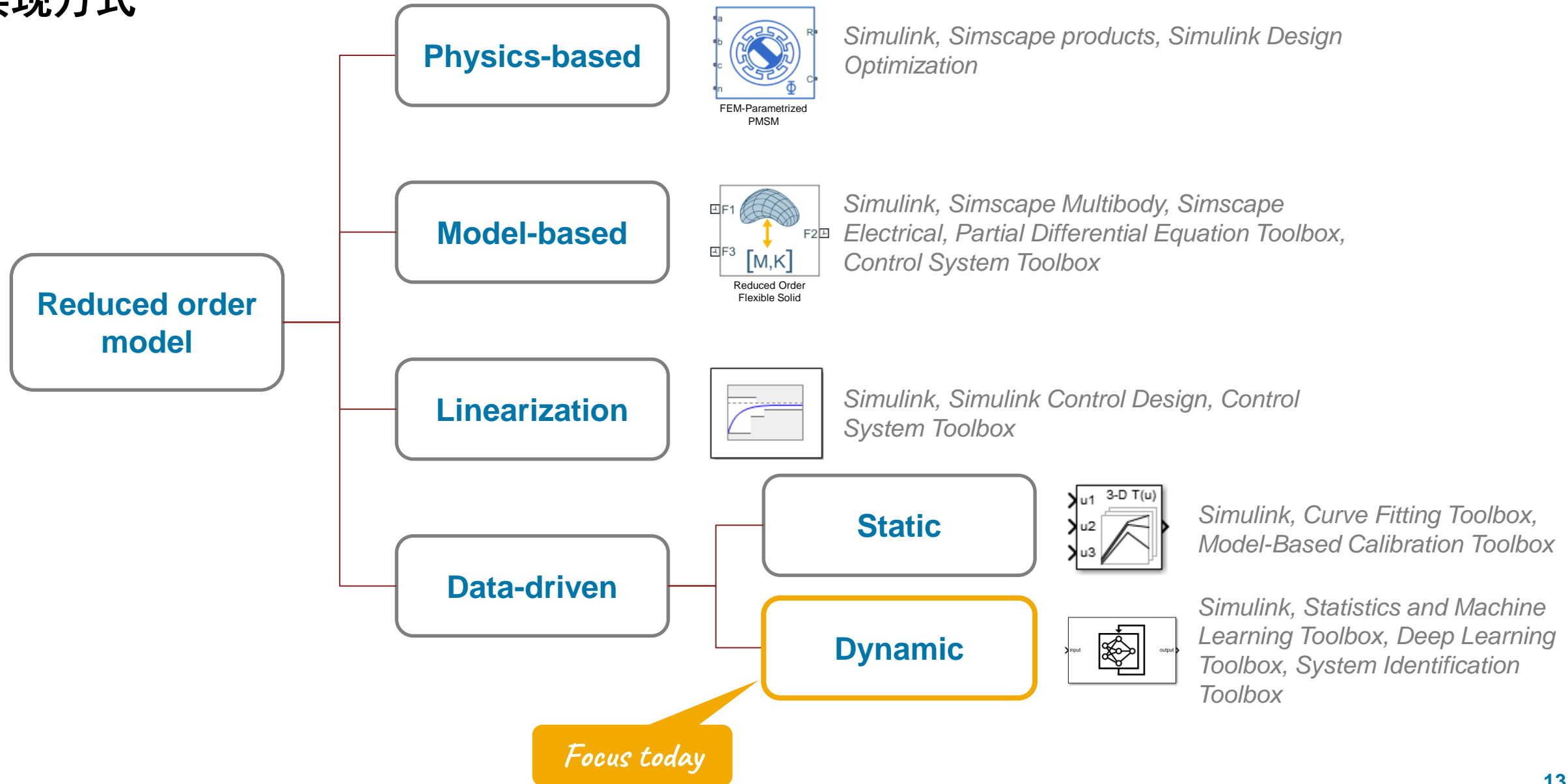


Reduced-Order Model (ROM)



模型降阶

实现方式



数据驱动（Data-driven） vs. 第一原理建模（first-principles）

数据驱动和第一原理建模可以共存

数据驱动模型

Statistics, optimization, AI

第一原理模型

Physics, math, domain knowledge

Black-box

Gray-box

White-box

Advantages

- 当第一原理的模型难于构建时可以发挥作用
- 降低复杂度，仿真加速
- 可以利用当前已有的或者测得的数据
- 不需要领域知识

Challenges

- 需要大量的数据
- Are often not
 - 可解释
 - 包含物理意义的参数化
- 在训练数据之外的外插能力较弱

Advantages

- 可以以低/高保真度捕获(全局)可参数化的行为
- 清晰的物理含义（可解释）
- 无需数据工程

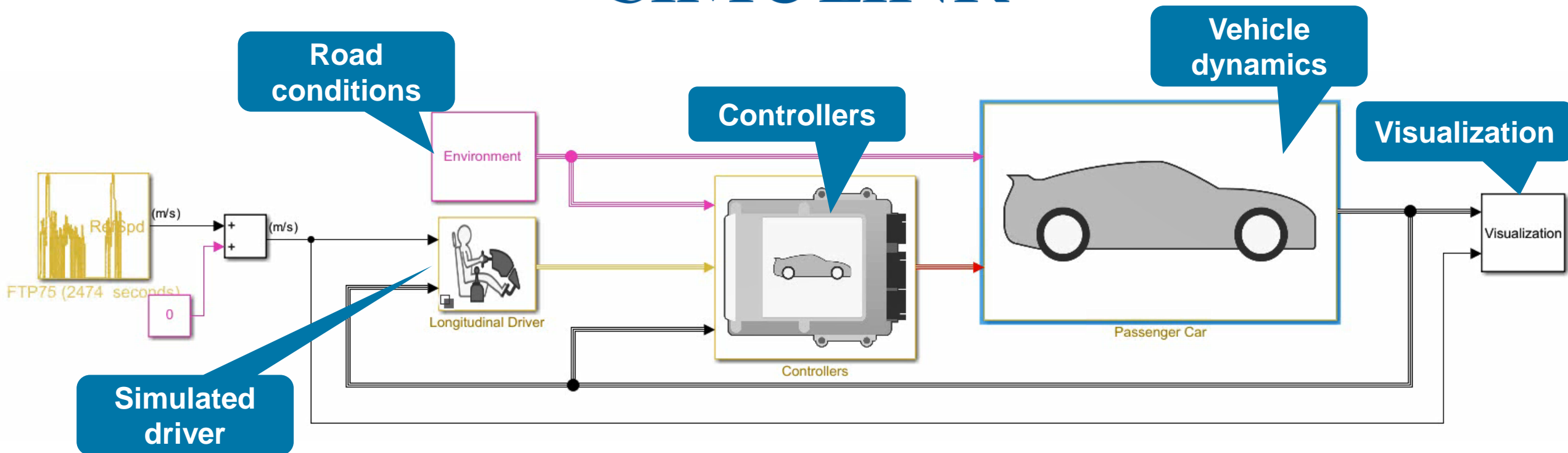
Challenges

- 难于获取/推导第一原理
- 需要强大的专业知识与理论背景

示例概览

使用基于AI的降阶模型替换第一原理发动机模型

SIMULINK®

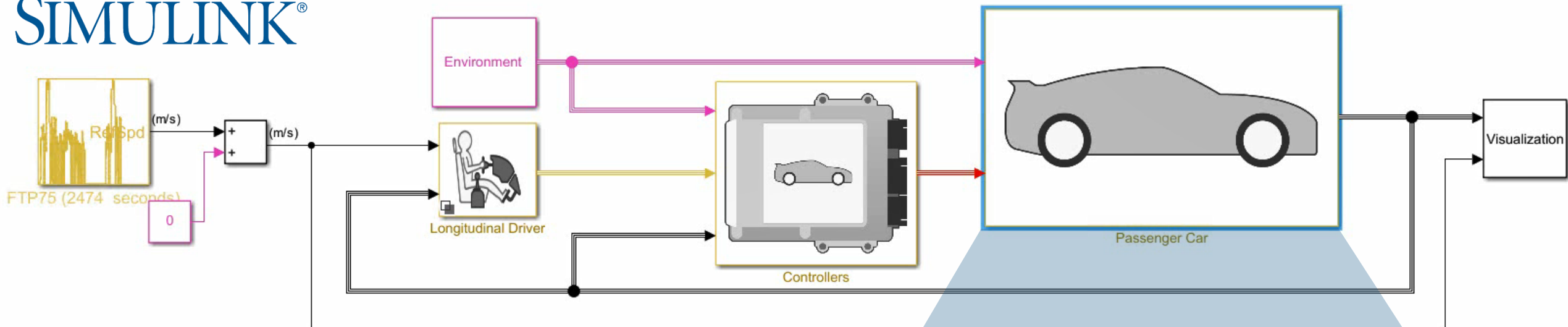


车速的闭环控制

示例概览

使用基于AI的降阶模型替换第一原理发动机模型

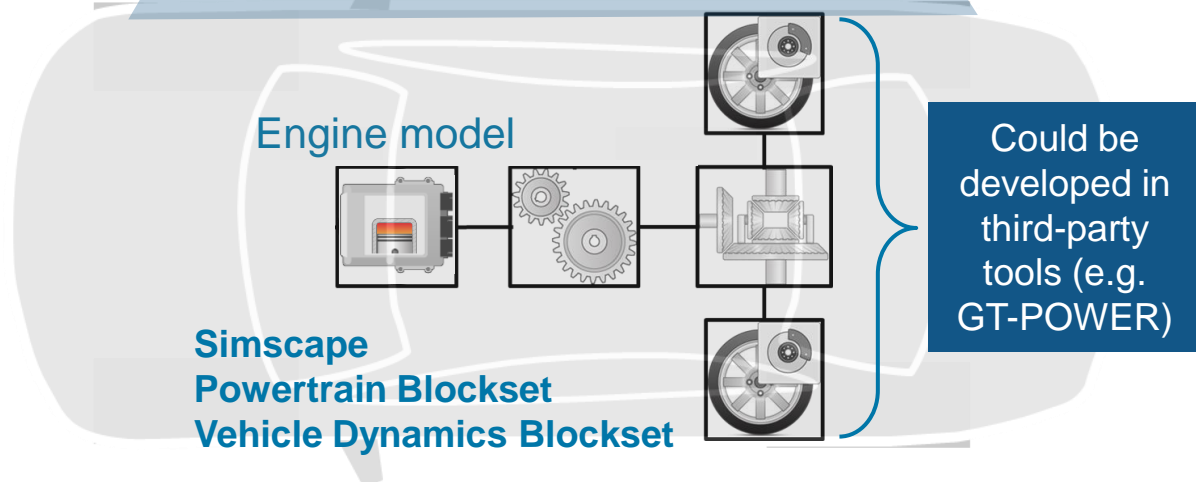
SIMULINK®



高保真

复杂模型

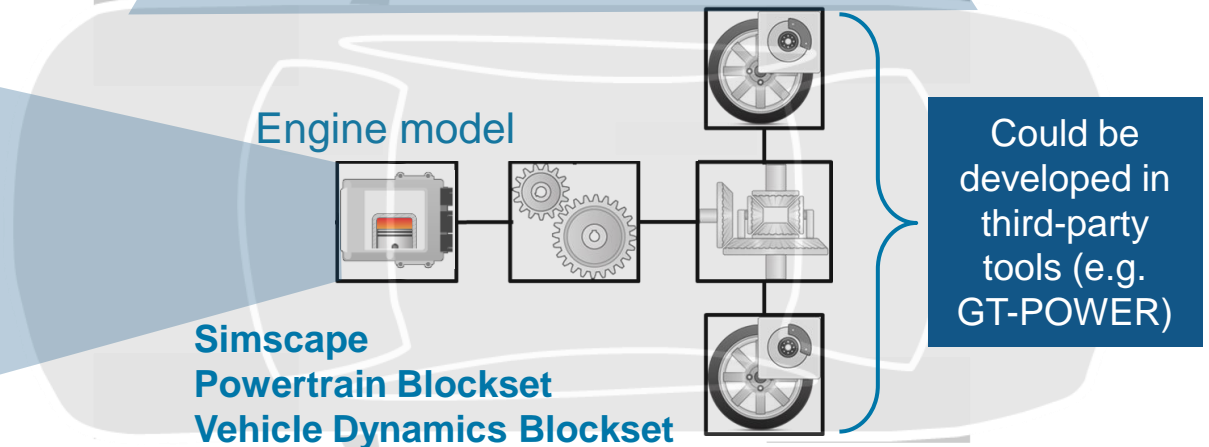
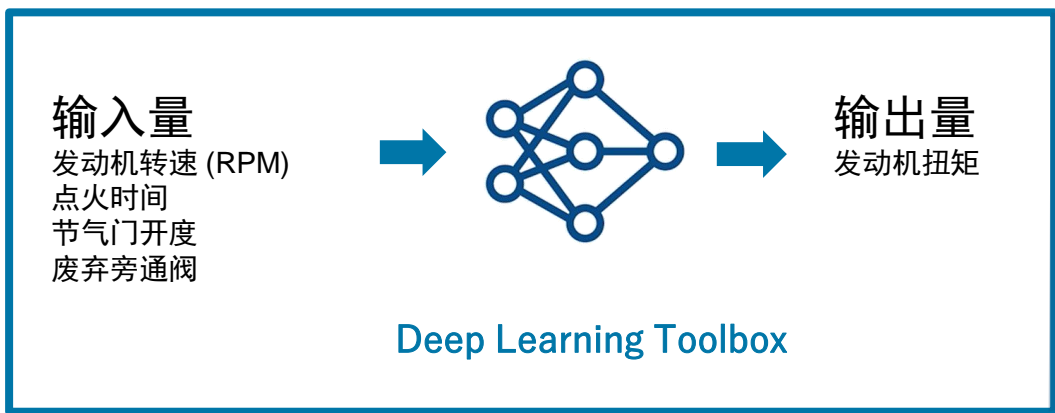
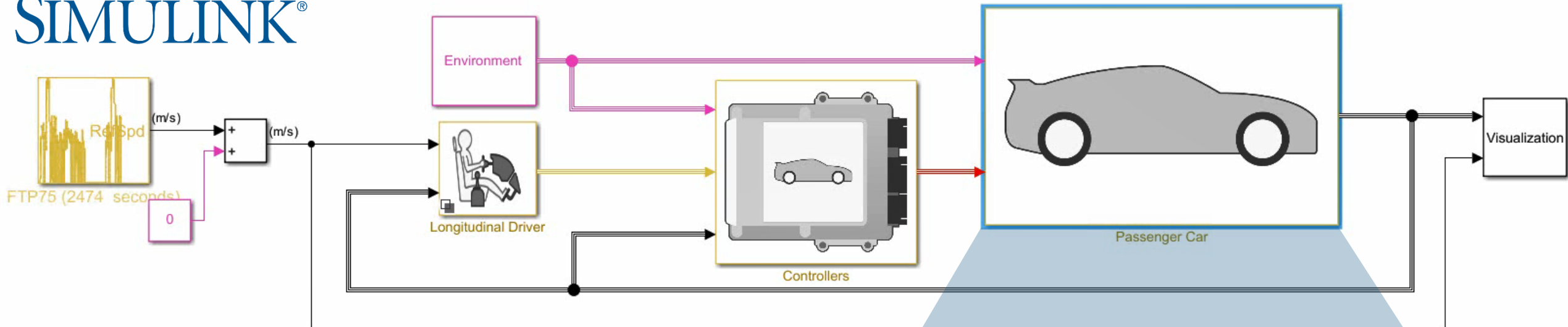
仿真慢



示例概览

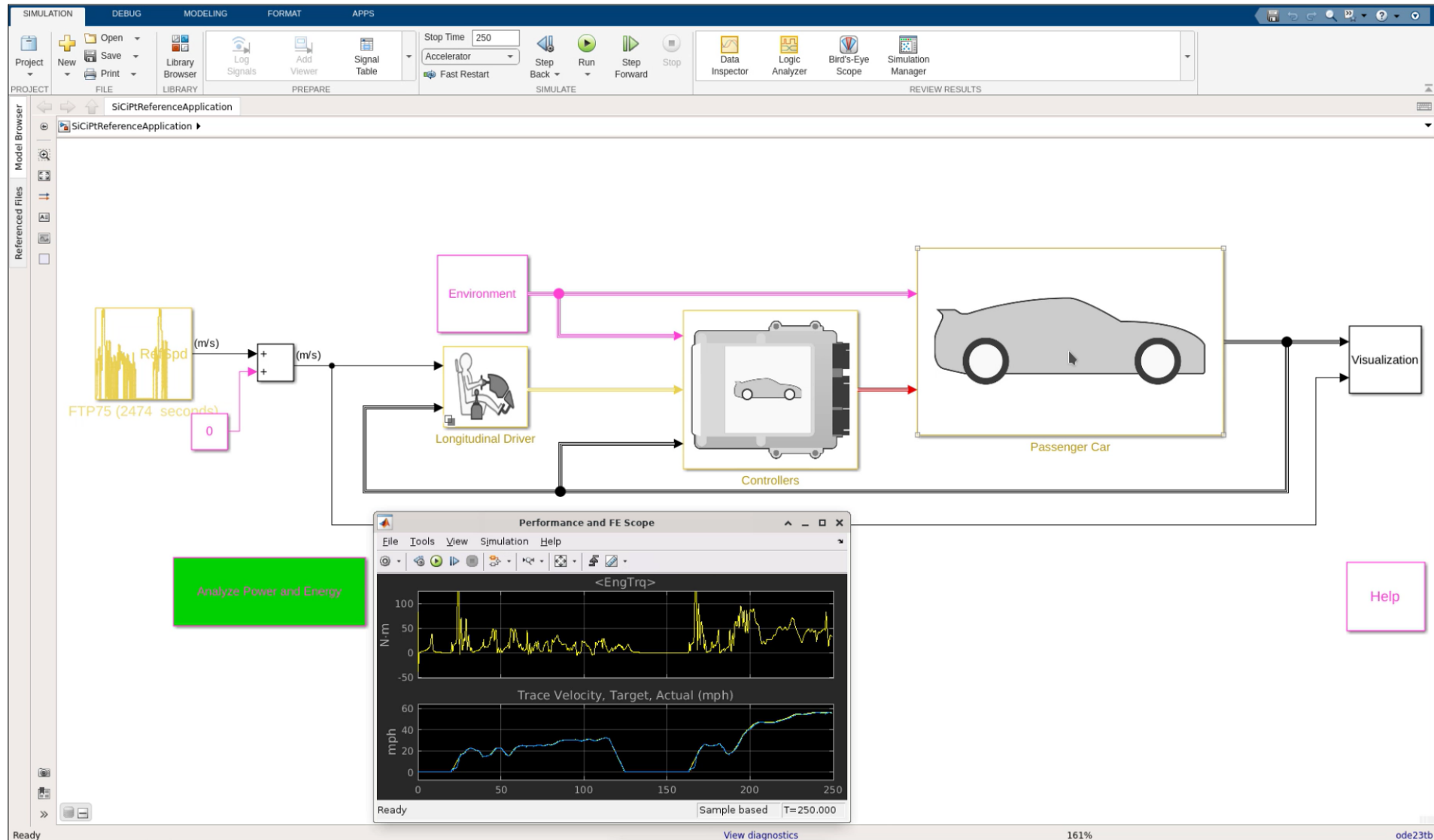
使用基于AI的降阶模型替换第一原理发动机模型

SIMULINK®

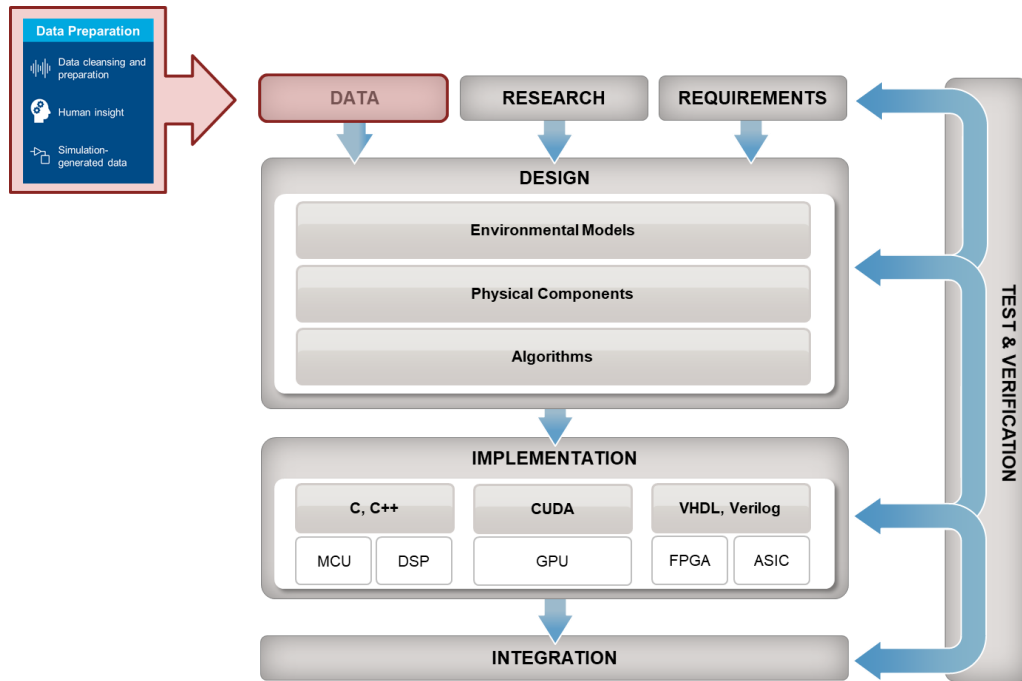


示例概览

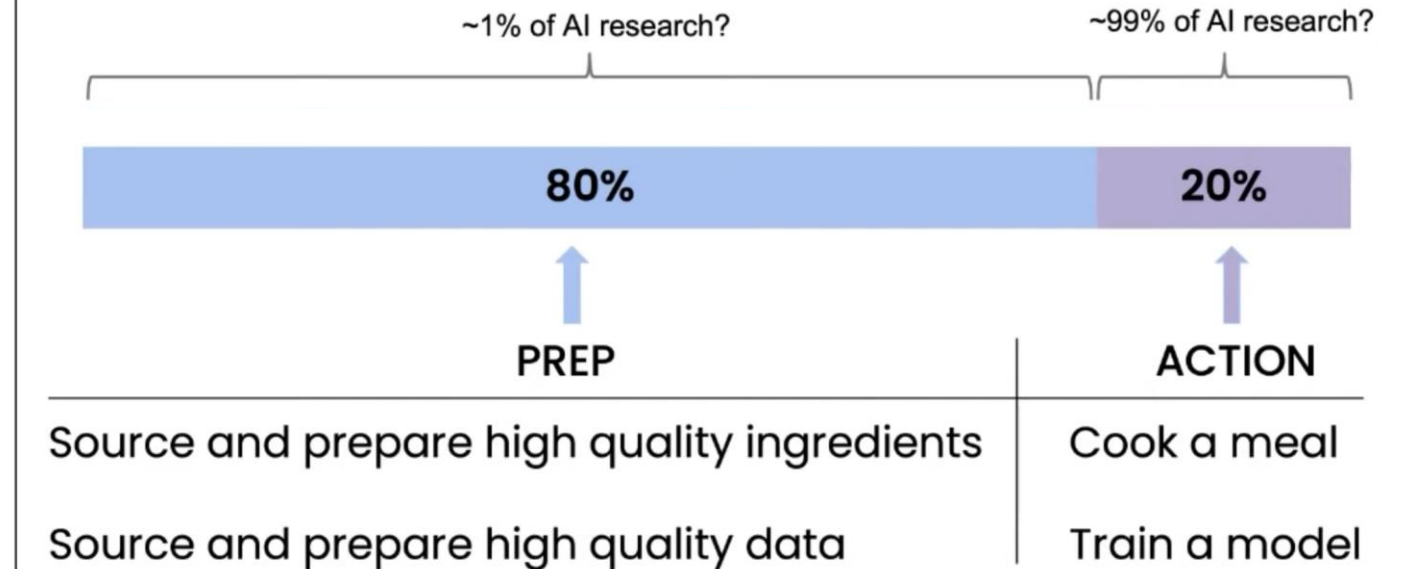
系统级仿真



数据准备的时间通常占大部分

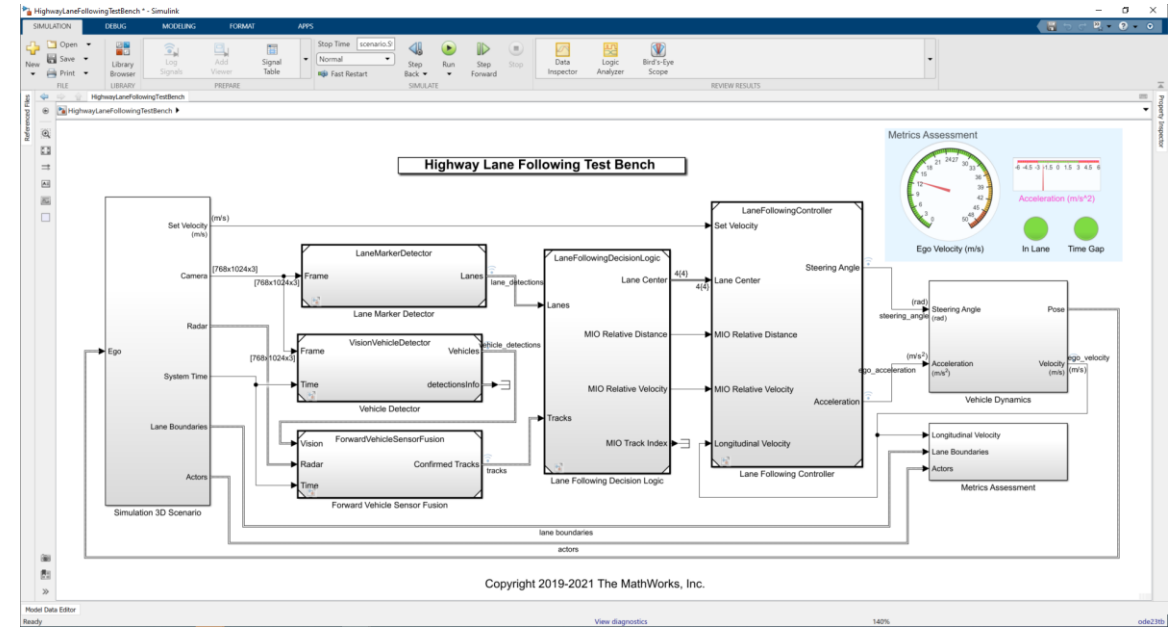
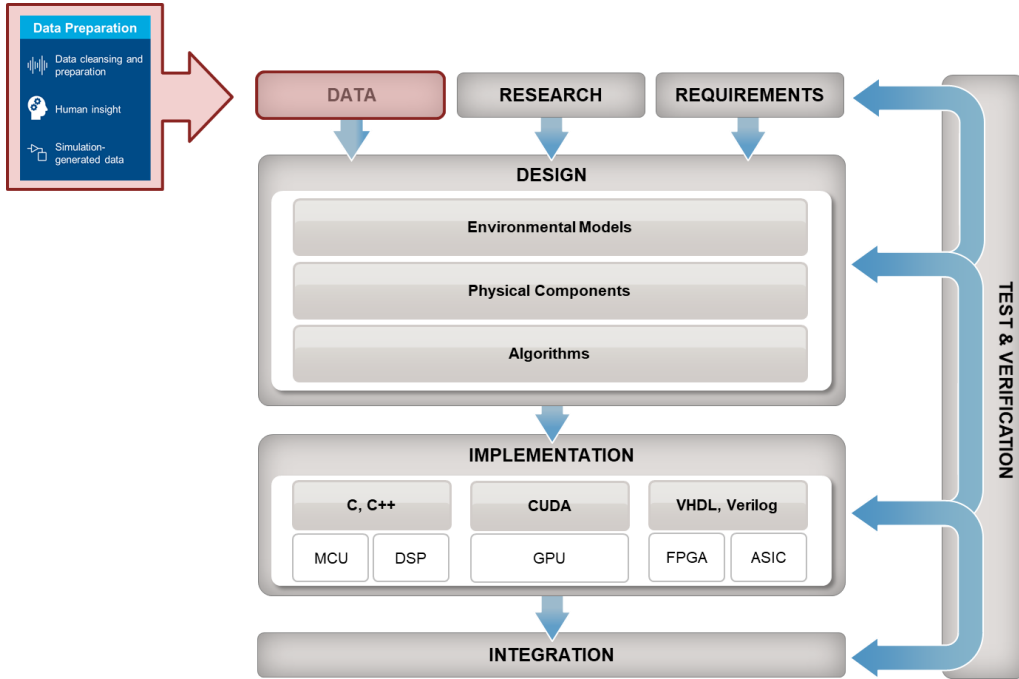


Data is Food for AI



Source: Andrew NG slide from MLOps 2021

生成合成数据用于训练

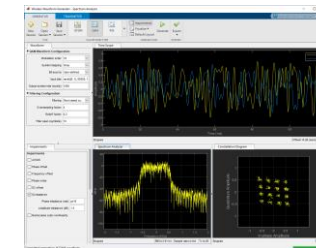


Simulink/Simscape

Other techniques:



GANs



Wireless Waveform Generator



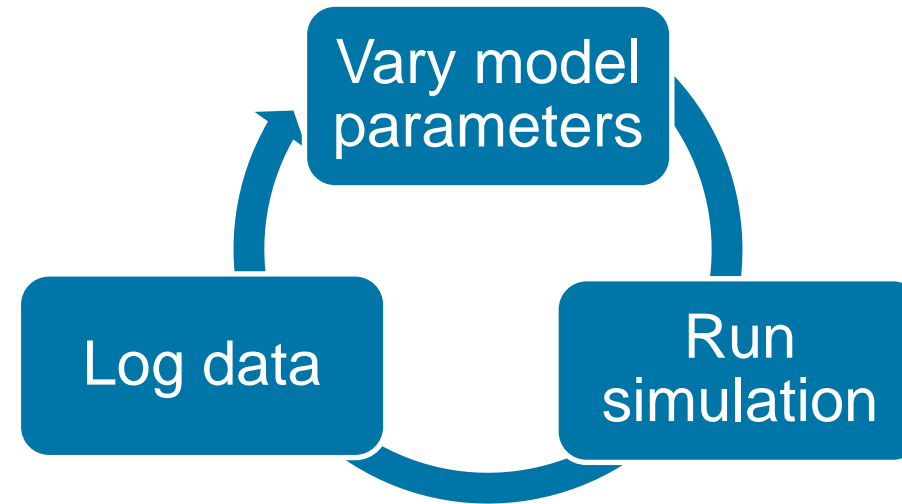
Unreal Engine®

合成数据生成

Design of Experiments

DoE = 512x3 table

	EngTrqReq	EngSpdR...	SpkAdvOfst
1	60	2000	-30
2	128	2500	15
3	94	2750	8
4	111	2875	-19
5	77	2625	-11
6	144	2125	4
7	85	2563	-21
8	119	3313	-28
9	68	2938	21

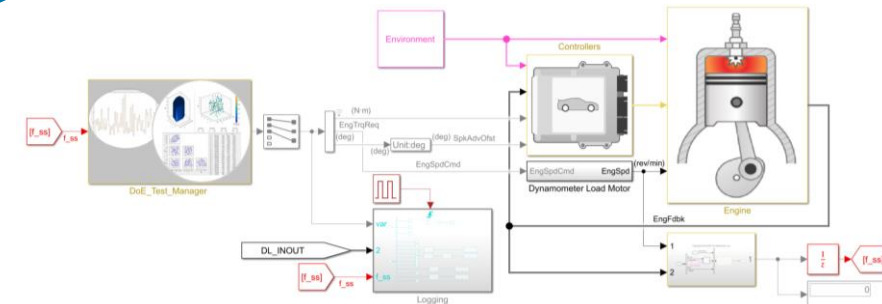


Input features

Engine speed (RPM)
Ignition timing
Throttle position
Wastegate valve

Response

Engine Torque



合成数据生成

Design of Experiments

The screenshot displays the MATLAB Simulink Simulation Manager interface. The main window shows a 2D scatter plot of **EngTrqReq** (Y-axis, 60 to 150) versus **EngSpdReq** (X-axis, 2000 to 4000). The plot contains numerous green data points, with a few blue points scattered throughout. A 3D scatter plot is also visible in the background, showing **SpkAdvOfst** (Z-axis, 0 to 4000) versus **EngSpdReq** and **EngTrqReq**.

The **Plot Properties** panel on the right is configured as follows:

- Title:** (empty)
- Grid:** X Y
- X-AXIS:**
 - Data: EngSpdReq
 - X Label: EngSpdReq
 - X Limits: 2000 to 4000 Auto
- Y-AXIS:**
 - Data: EngTrqReq
 - Y Label: EngTrqReq
 - Y Limits: 60 to 150 Auto
- MARKER COLOR:**
 - Data: Simulation Status
 - Label: (empty)
 - Limits: 0 to 1 Auto
 - Colormap: parula(default)
 - Colorbar:

The **Live Editor** at the bottom left shows the following code:

```

36 open_system('DoE_Engine_Test.slx')
37 % Run parallel simulations, 300s
38 simout = helper.simulation(DoE, 'Rapid')

```

The **Command Window** at the bottom shows the execution progress:

```

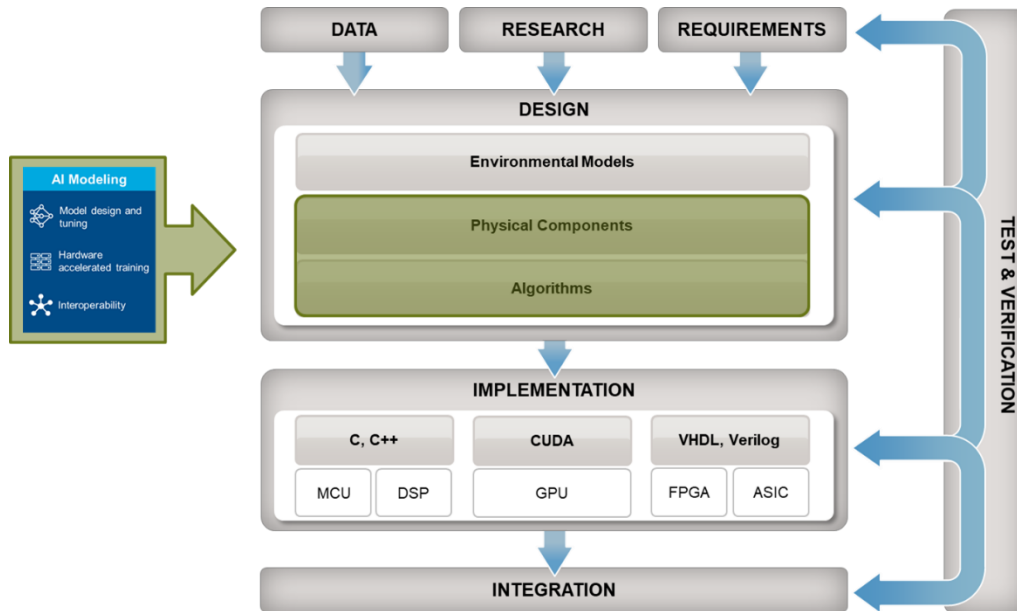
Starting parallel pool (parpool) using...
Connected to the parallel pool (number of workers: 32)
[10-Jun-2022 03:33:09] Checking for...
[10-Jun-2022 03:33:09] Starting Simulink on parallel workers...
[10-Jun-2022 03:33:56] Loading project on parallel workers...
[10-Jun-2022 03:33:56] Configuring simulation cache folder on parallel workers...
[10-Jun-2022 03:34:18] Transferring base workspace variables used in the model to parallel workers...
[10-Jun-2022 03:34:26] Loading model on parallel workers...
[10-Jun-2022 03:34:53] Running simulations...

```

The **Simulation Manager** window shows the simulation is running on 32 parallel workers with a speedup of 19.5x and 97% completion. The remaining time is 12 seconds.

The **Check and Save Simulation Result** button is visible at the bottom of the interface.

数据驱动的ROM



Static

Look up table

Surface Fitting

Machine Learning

etc.

Dynamic

LSTM

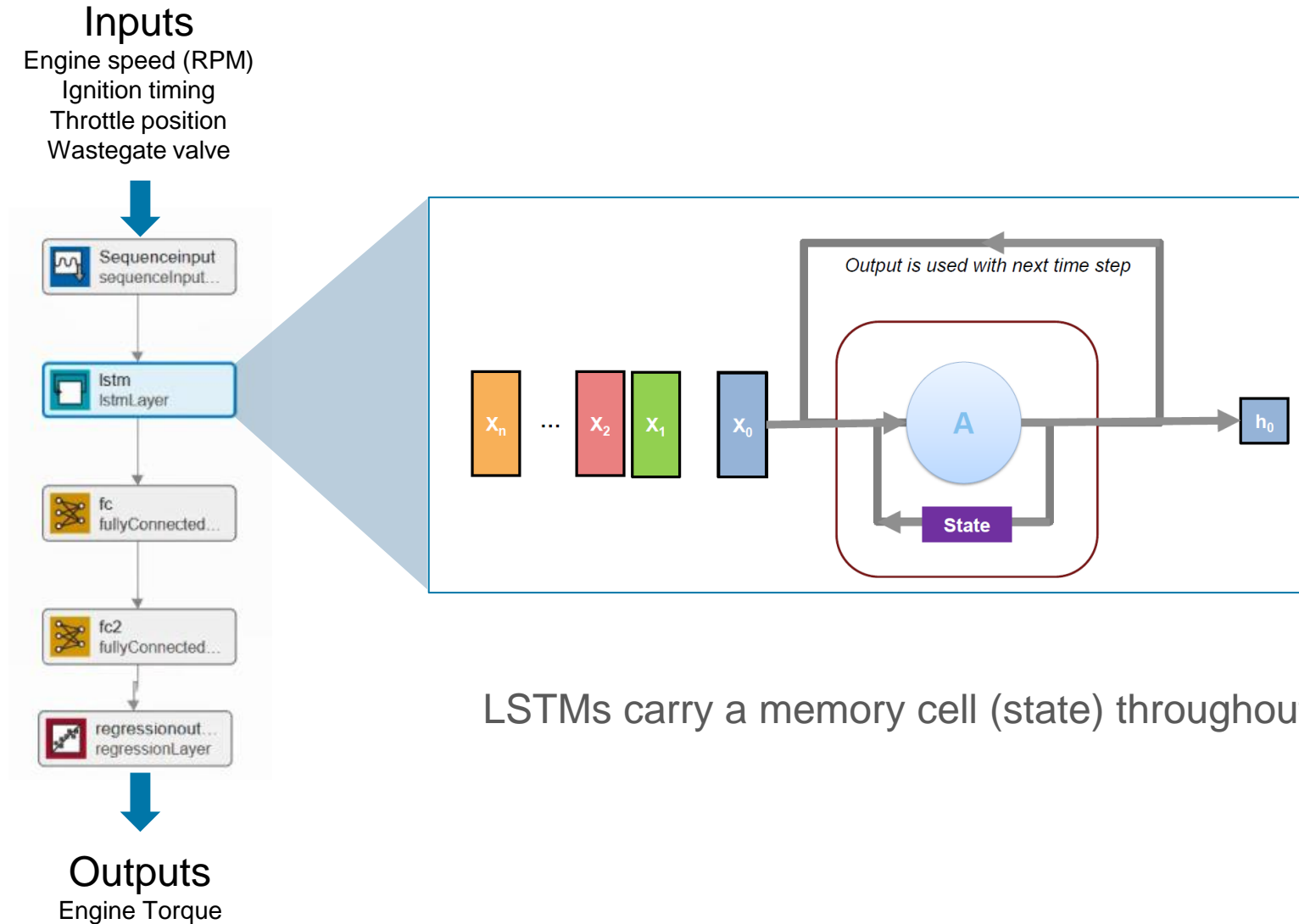
Neural ODE

NLARX models

etc.

基于AI的ROM: LSTM

从时间序列数据中学习时间上的依赖



基于AI的ROM: LSTM

从时间序列数据中学习时间上的依赖

Open Deep Network Designer app from the MATLAB Toolstrip to interactively design your neural network.

```

deepNetworkDesigner

48 numFeatures = 4;
49 numResponses = 1;
50 numHiddenUnits = 60;
51 dropoutProbability = 0.2;
52
53 layers = [
54     sequenceInputLayer(numFeatures, "Name", "input")

```

基于AI的ROM: Neural ODEs

使用神经网络学习系统动态特性

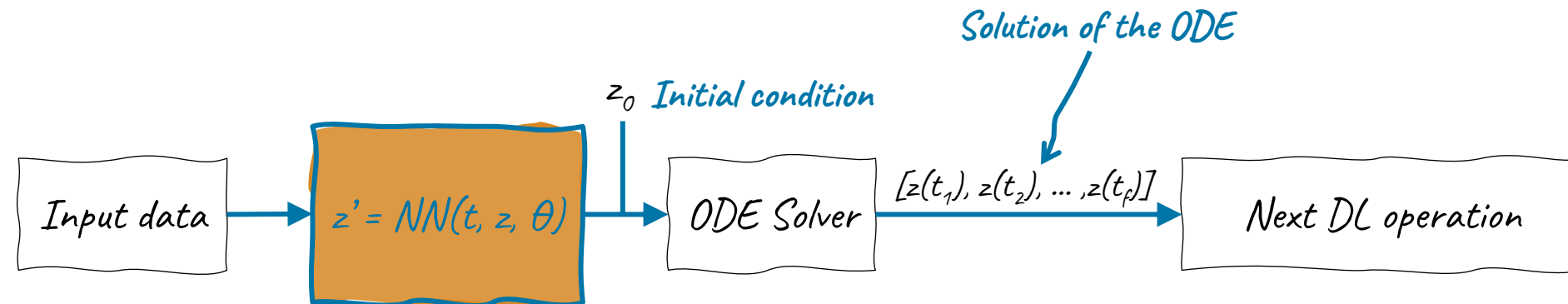
Ordinary Differential Equations (ODEs)

$$\begin{cases} z' = f(t, z), & t \in (t_0, t_f) \\ z(t_0) = z_0 \end{cases}$$

- Model phenomena in many domains: physics, finance, biology, chemistry, ...
- In real-world applications, ODEs are solved with numerical integration techniques, approximating $z(t)$

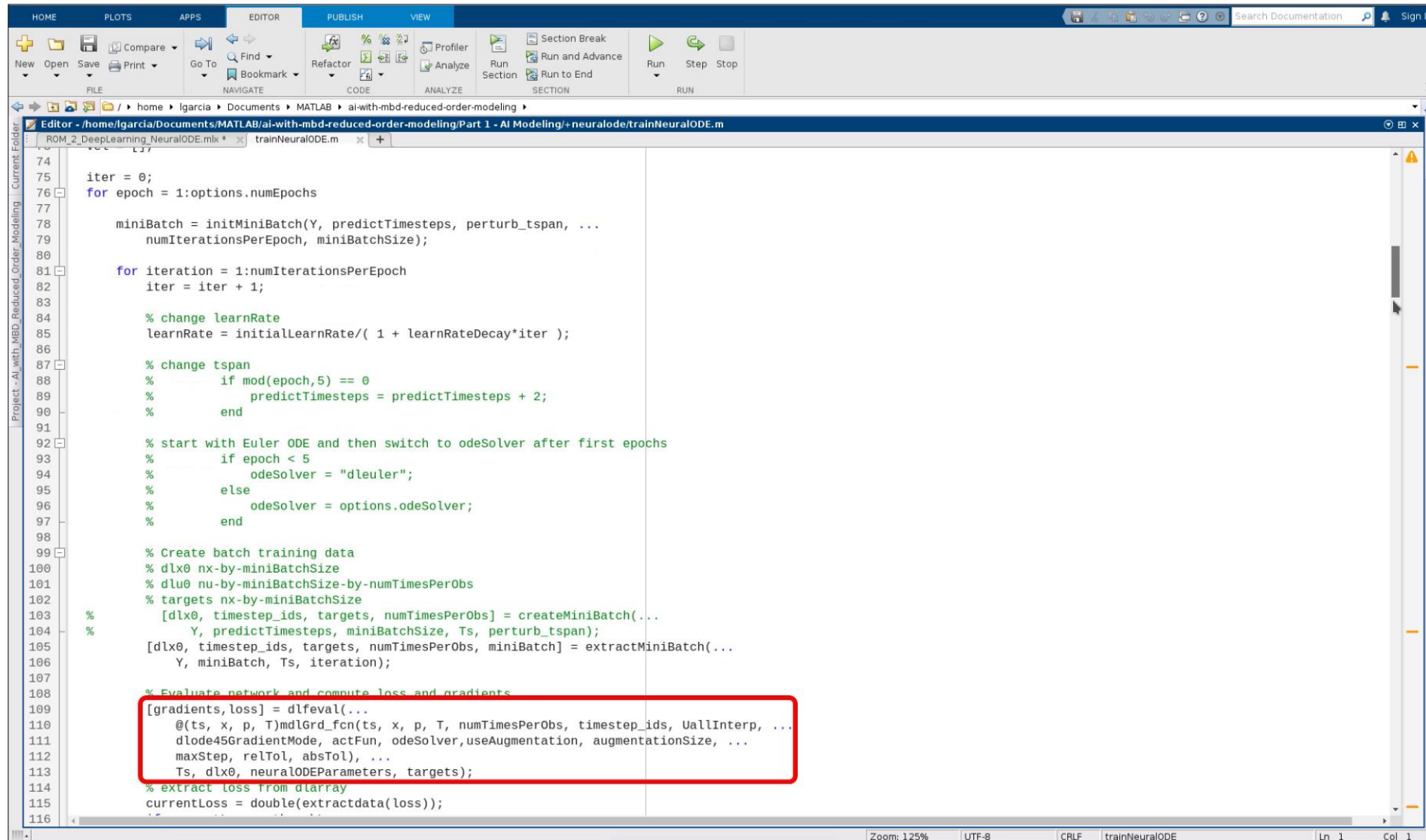
Neural ODEs

- Solving an ODE from first principles may be challenging / not always possible
- Train a neural network from data to represent the function $f(t, z)$
- Solve the ODE using the trained network $NN(t, z, \theta)$ instead of $f(t, z)$



基于AI的ROM: Neural ODEs

使用神经网络学习系统动态特性



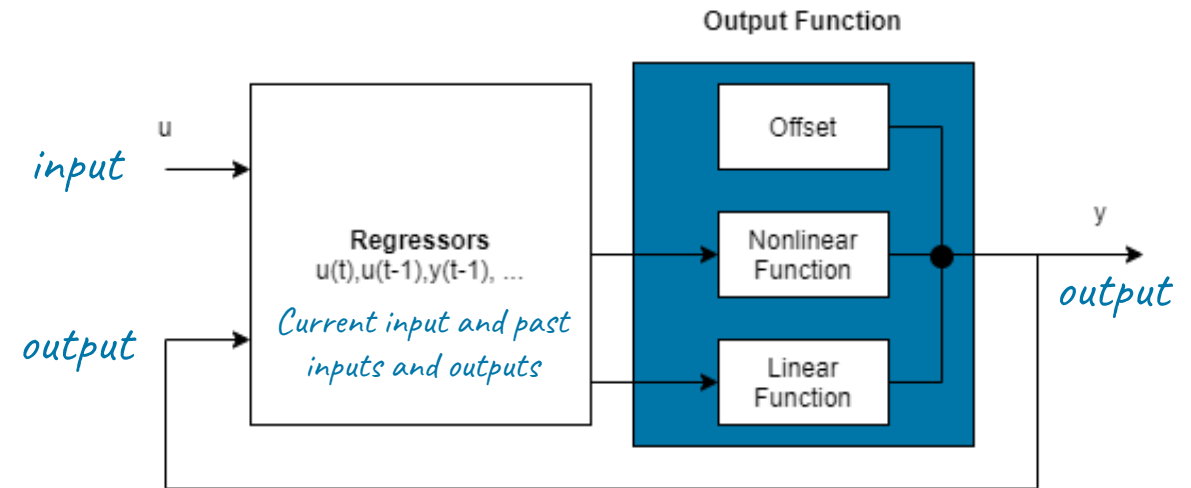
```
74
75
76 for epoch = 1:options.numEpochs
77
78     miniBatch = initMiniBatch(Y, predictTimesteps, perturb_tspan, ...
79         numIterationsPerEpoch, miniBatchSize);
80
81     for iteration = 1:numIterationsPerEpoch
82         iter = iter + 1;
83
84         % change learnRate
85         learnRate = initialLearnRate/( 1 + learnRateDecay*iter );
86
87         % change tspan
88         %     if mod(epoch,5) == 0
89         %         predictTimesteps = predictTimesteps + 2;
90         %     end
91
92         % start with Euler ODE and then switch to odeSolver after first epochs
93         %     if epoch < 5
94         %         odeSolver = "dleuler";
95         %     else
96         %         odeSolver = options.odeSolver;
97         %     end
98
99         % Create batch training data
100        % dlx0 nx-by-miniBatchSize
101        % dlu0 nu-by-miniBatchSize-by-numTimesPerObs
102        % targets nx-by-miniBatchSize
103        % [dlx0, timestep_ids, targets, numTimesPerObs] = createMiniBatch(...
104        %     Y, predictTimesteps, miniBatchSize, Ts, perturb_tspan);
105        % [dlx0, timestep_ids, targets, numTimesPerObs, miniBatch] = extractMiniBatch(...
106        %     Y, miniBatch, Ts, iteration);
107
108        % Evaluate network and compute loss and gradients
109        [gradients,loss] = dlfeval(...
110            @(ts, x, p, T)mdlGrd_fcn(ts, x, p, T, numTimesPerObs, timestep_ids, UallInterp, ...
111                dlode45GradientMode, actFun, odeSolver,useAugmentation, augmentationSize, ...
112                maxStep, relTol, absTol), ...
113            Ts, dlx0, neuralODEParameters, targets);
114        % extract loss from darray
115        currentLoss = double(extractdata(loss));
116
```

基于AI的ROM: Nonlinear ARX

使用Sigmoid-based Nonlinear ARX Model进行代理模型创建

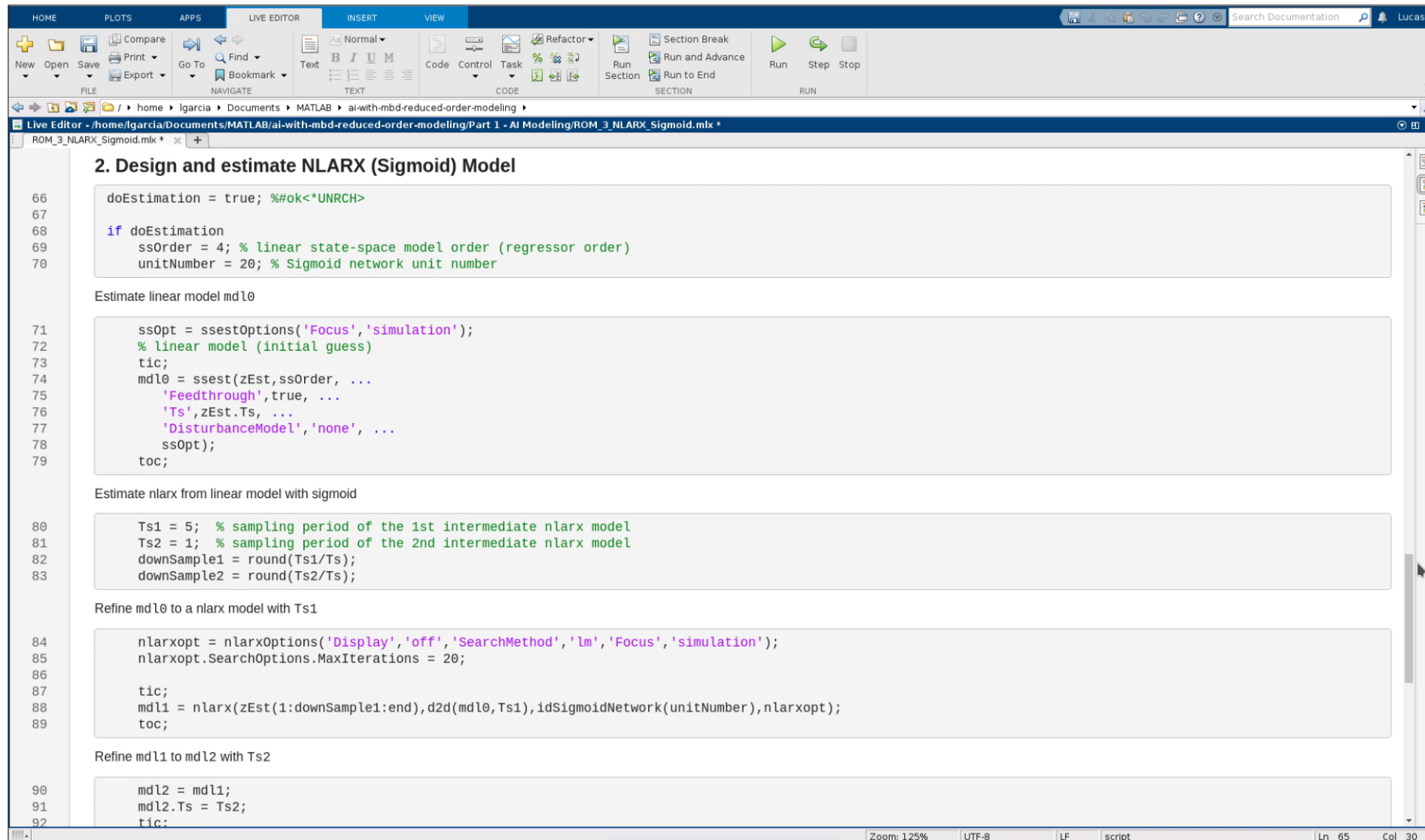
Nonlinear ARX (NLARX)

- Extends linear ARX to the nonlinear case
- Flexible choice of nonlinear functions
- May be more interpretable than Deep Learning models
- Potentially faster training and simulation



基于AI的ROM: Nonlinear ARX

使用Sigmoid-based Nonlinear ARX Model进行代理模型创建



```
HOME PLOTS APPS LIVE EDITOR INSERT VIEW Search Documentation Lucas
New Open Save Print Go To Find Export FILE NAVIGATE TEXT CODE SECTION RUN
doEstimation = true; %#ok<UNRCH>
if doEstimation
    ssOrder = 4; % linear state-space model order (regressor order)
    unitNumber = 20; % Sigmoid network unit number
Estimate linear model mdl0
ssOpt = ssestOptions('Focus','simulation');
% linear model (initial guess)
tic;
mdl0 = ssest(zEst,ssOrder, ...
    'Feedthrough',true, ...
    'Ts',zEst.Ts, ...
    'DisturbanceModel','none', ...
    ssOpt);
toc;
Estimate nlarx from linear model with sigmoid
Ts1 = 5; % sampling period of the 1st intermediate nlarx model
Ts2 = 1; % sampling period of the 2nd intermediate nlarx model
downSample1 = round(Ts1/Ts);
downSample2 = round(Ts2/Ts);
Refine mdl0 to a nlarx model with Ts1
nlarxopt = nlarxOptions('Display','off','SearchMethod','lm','Focus','simulation');
nlarxopt.SearchOptions.MaxIterations = 20;
tic;
mdl1 = nlarx(zEst(1:downSample1:end),d2d(mdl0,Ts1),idSigmoidNetwork(unitNumber),nlarxopt);
toc;
Refine mdl1 to mdl2 with Ts2
mdl2 = mdl1;
mdl2.Ts = Ts2;
tic;
```

2. Design and estimate NLARX (Sigmoid) Model

66 doEstimation = true; %#ok<UNRCH>

67

68 if doEstimation

69 ssOrder = 4; % linear state-space model order (regressor order)

70 unitNumber = 20; % Sigmoid network unit number

Estimate linear model mdl0

71 ssOpt = ssestOptions('Focus','simulation');

72 % linear model (initial guess)

73 tic;

74 mdl0 = ssest(zEst,ssOrder, ...

75 'Feedthrough',true, ...

76 'Ts',zEst.Ts, ...

77 'DisturbanceModel','none', ...

78 ssOpt);

79 toc;

Estimate nlarx from linear model with sigmoid

80 Ts1 = 5; % sampling period of the 1st intermediate nlarx model

81 Ts2 = 1; % sampling period of the 2nd intermediate nlarx model

82 downSample1 = round(Ts1/Ts);

83 downSample2 = round(Ts2/Ts);

Refine mdl0 to a nlarx model with Ts1

84 nlarxopt = nlarxOptions('Display','off','SearchMethod','lm','Focus','simulation');

85 nlarxopt.SearchOptions.MaxIterations = 20;

86

87 tic;

88 mdl1 = nlarx(zEst(1:downSample1:end),d2d(mdl0,Ts1),idSigmoidNetwork(unitNumber),nlarxopt);

89 toc;

Refine mdl1 to mdl2 with Ts2

90 mdl2 = mdl1;

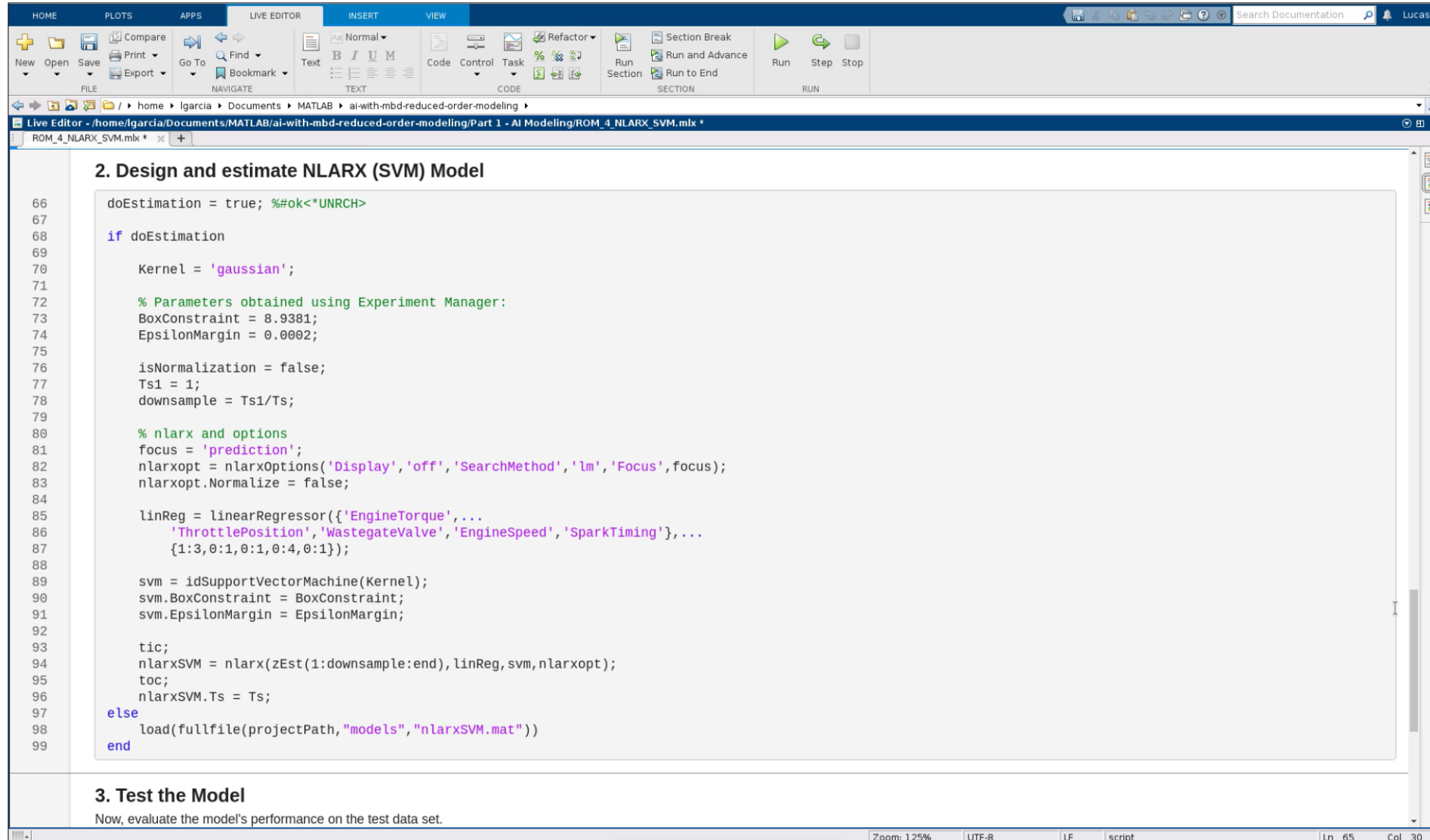
91 mdl2.Ts = Ts2;

92 tic;

Zoom: 125% UTF-8 LF script Ln 65 Col 30

基于AI的ROM: Nonlinear ARX

使用SVM-based Nonlinear ARX Model进行代理模型创建



```

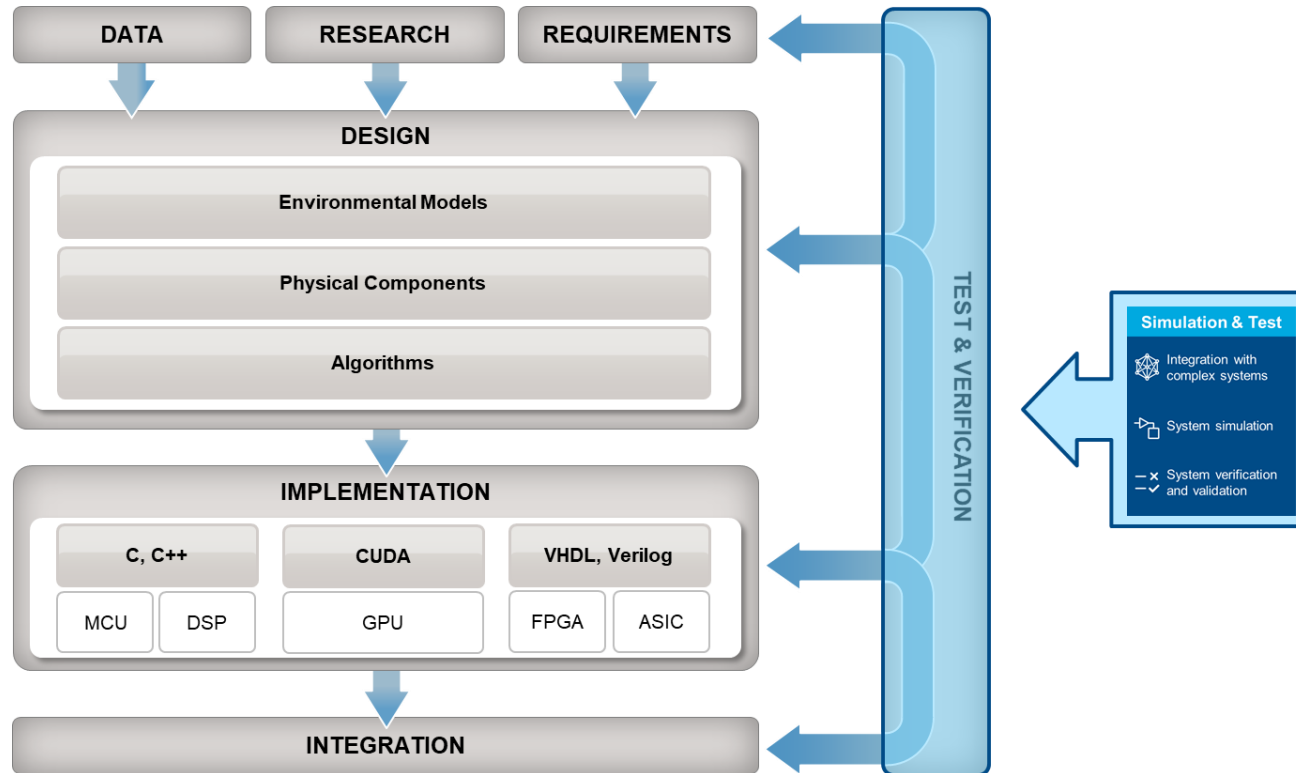
2. Design and estimate NLARX (SVM) Model

66 doEstimation = true; %#ok<*UNRCH>
67
68 if doEstimation
69
70     Kernel = 'gaussian';
71
72     % Parameters obtained using Experiment Manager:
73     BoxConstraint = 8.9381;
74     EpsilonMargin = 0.0002;
75
76     isNormalization = false;
77     Ts1 = 1;
78     downsample = Ts1/Ts;
79
80     % nlarx and options
81     focus = 'prediction';
82     nlarxopt = nlarxOptions('Display','off','SearchMethod','lm','Focus',focus);
83     nlarxopt.Normalize = false;
84
85     linReg = linearRegressor({'EngineTorque',...
86         'ThrottlePosition','WastegateValve','EngineSpeed','SparkTiming'},...
87         {1:3,0:1,0:1,0:4,0:1});
88
89     svm = idSupportVectorMachine(Kernel);
90     svm.BoxConstraint = BoxConstraint;
91     svm.EpsilonMargin = EpsilonMargin;
92
93     tic;
94     nlarxSVM = nlarx(zEst(1:downsample:end), linReg, svm, nlarxopt);
95     toc;
96     nlarxSVM.Ts = Ts;
97 else
98     load(fullfile(projectPath,"models","nlarxSVM.mat"))
99 end

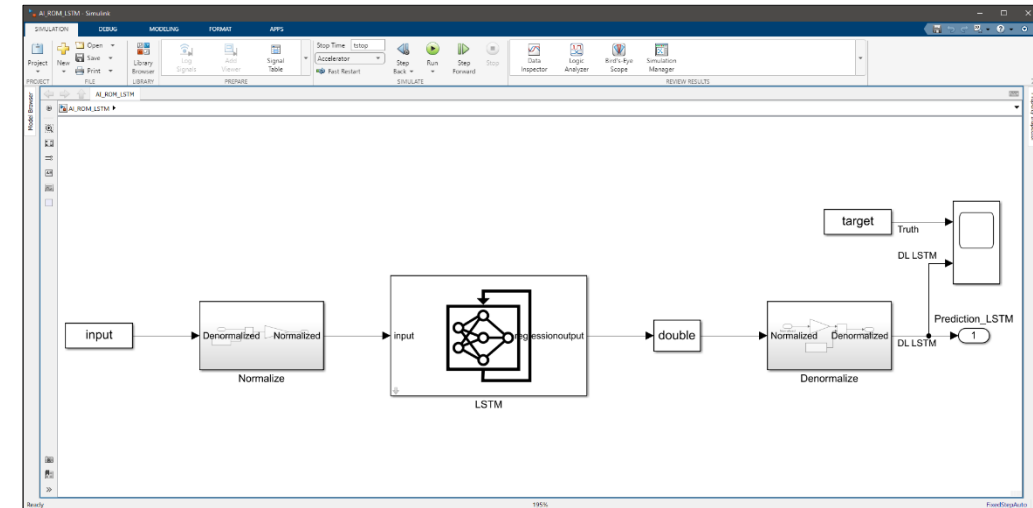
3. Test the Model
Now, evaluate the model's performance on the test data set.

```

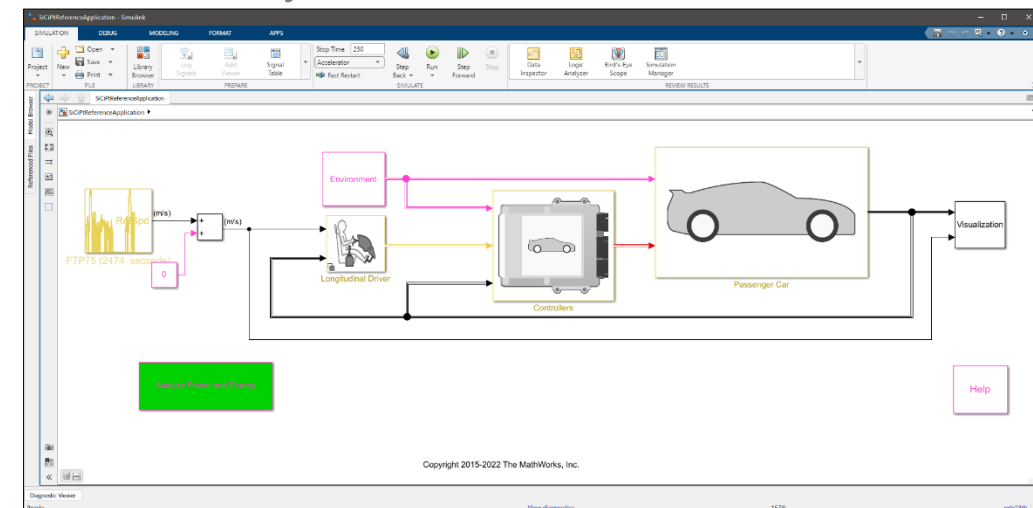
将AI模型集成进行系统级仿真与测试



Integration of trained AI model into Simulink

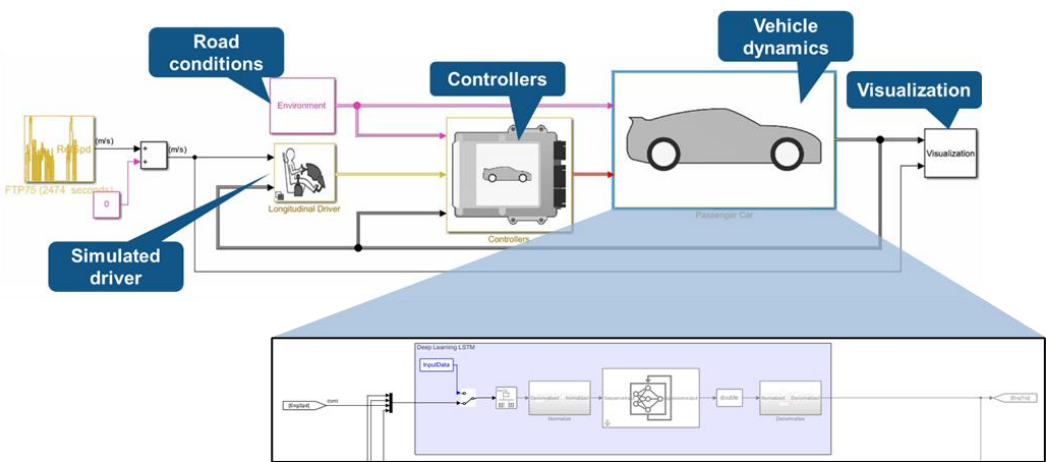


System-level simulation



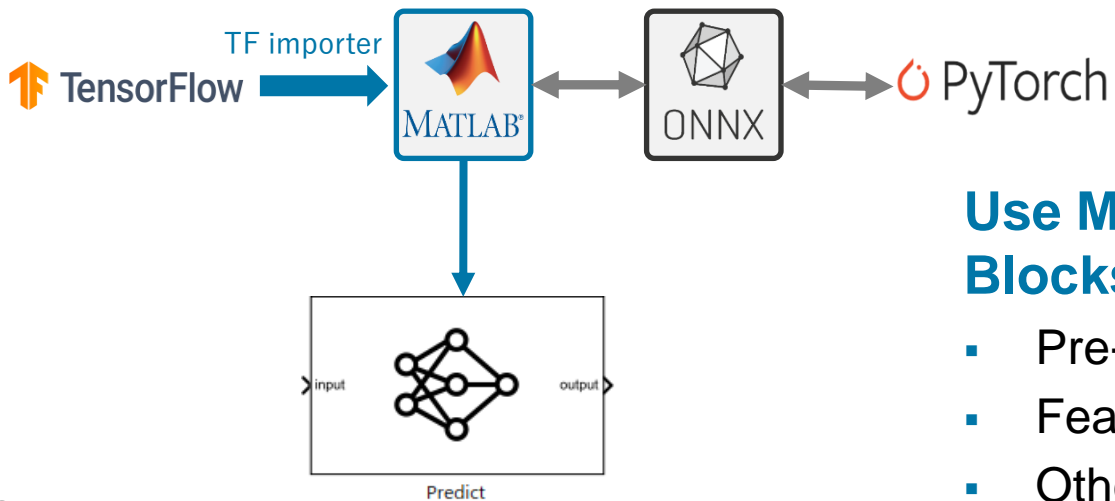
在Simulink的系统模型中集成AI模型

将AI模型集成其他算法用于系统级仿真与测试



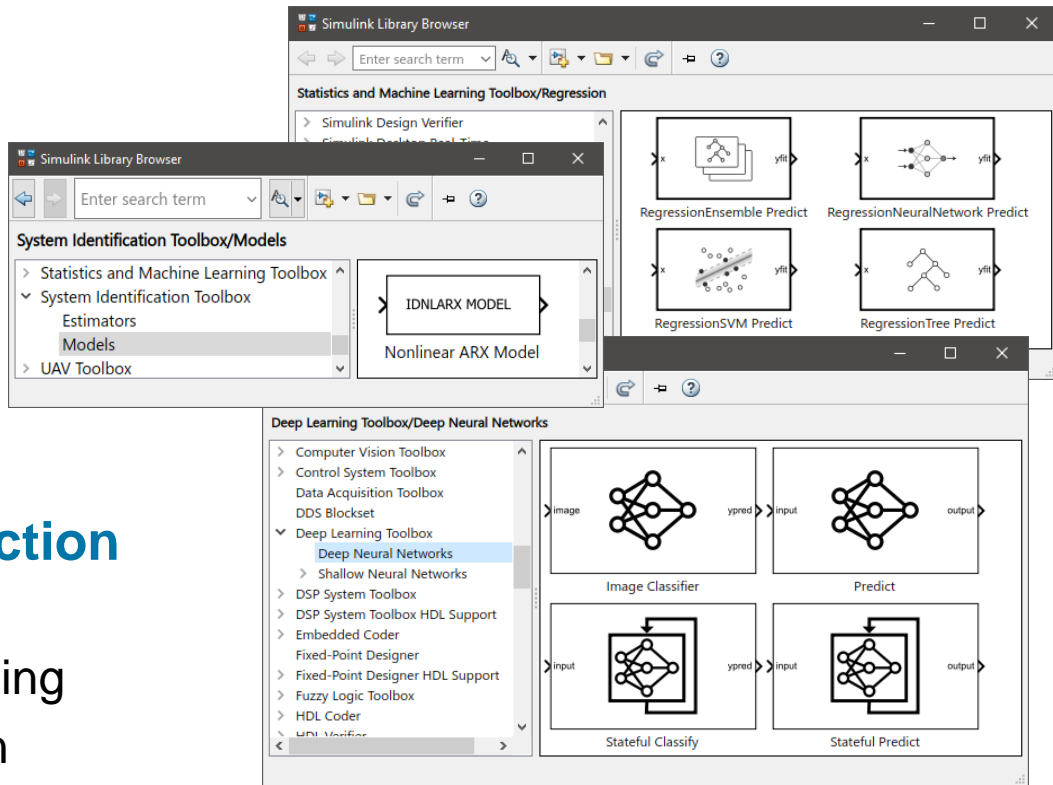
Use **Statistics and Machine Learning Toolbox**, **System Identification Toolbox** and **Deep Learning Toolbox** block libraries to bring AI models into Simulink

和其他框架互操作

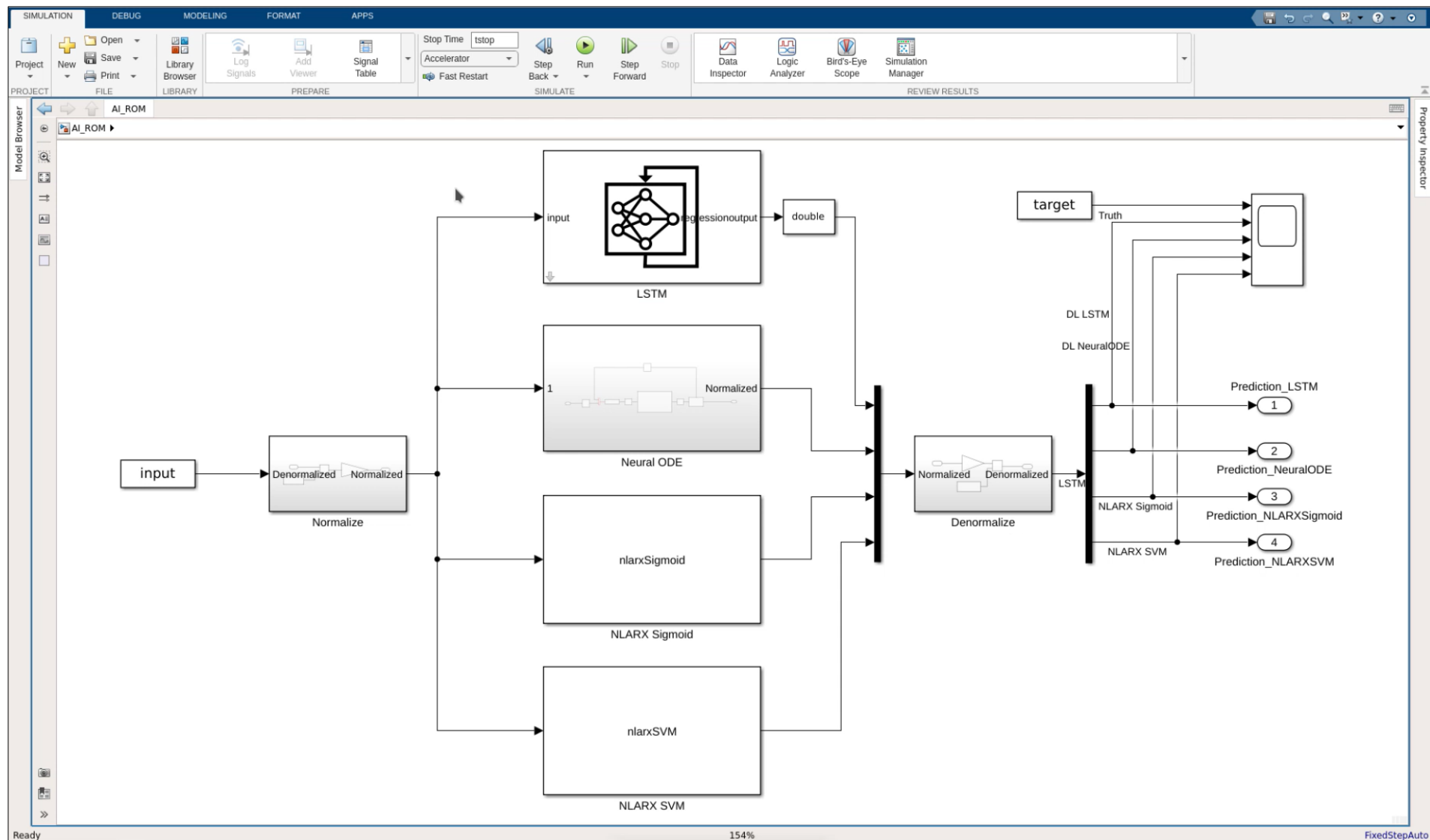


Use MATLAB Function Blocks for:

- Pre-/Post-processing
- Feature Extraction
- Other models



训练好的AI模型集成到Simulink

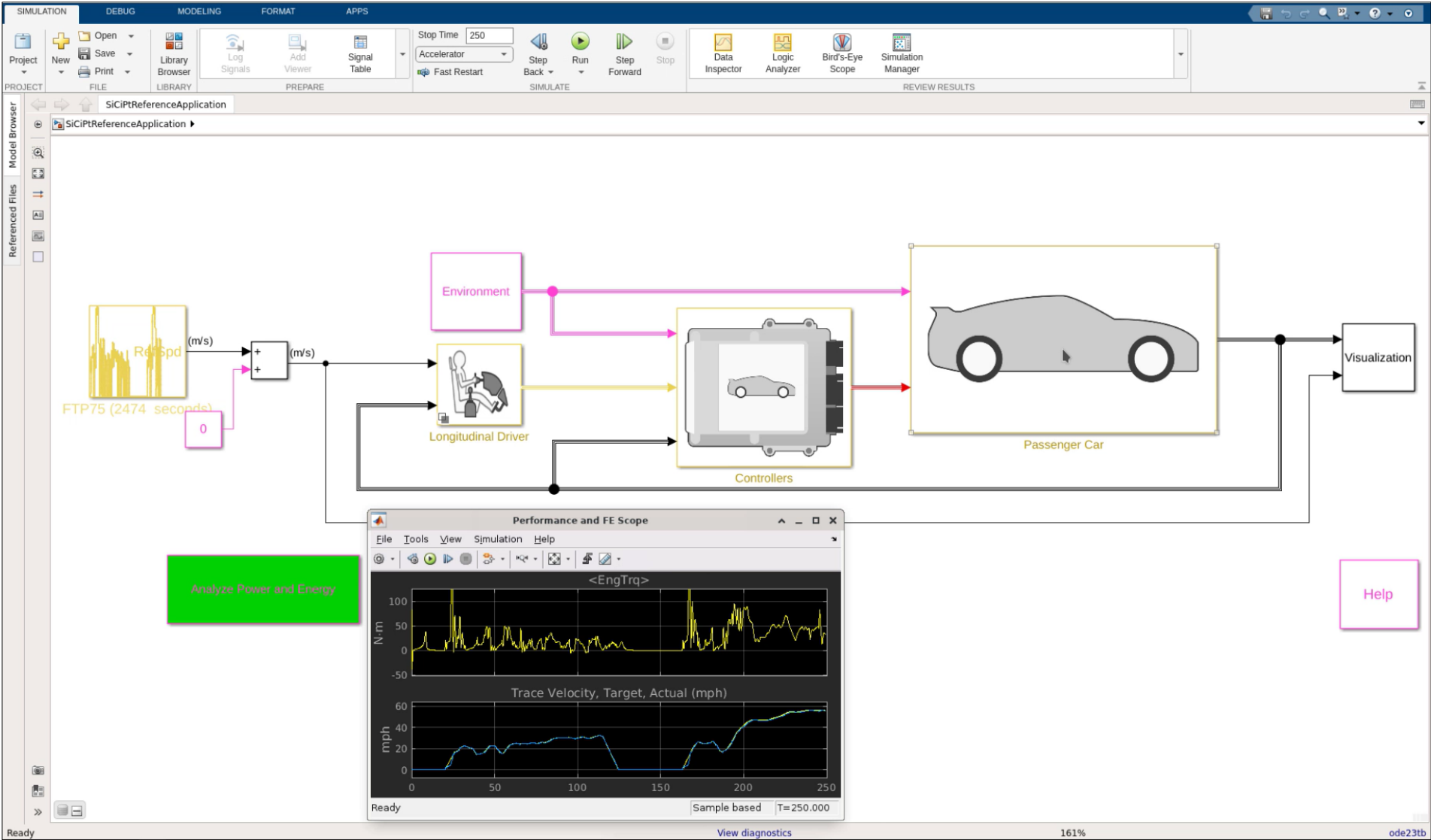


训练好的AI模型集成到Simulink

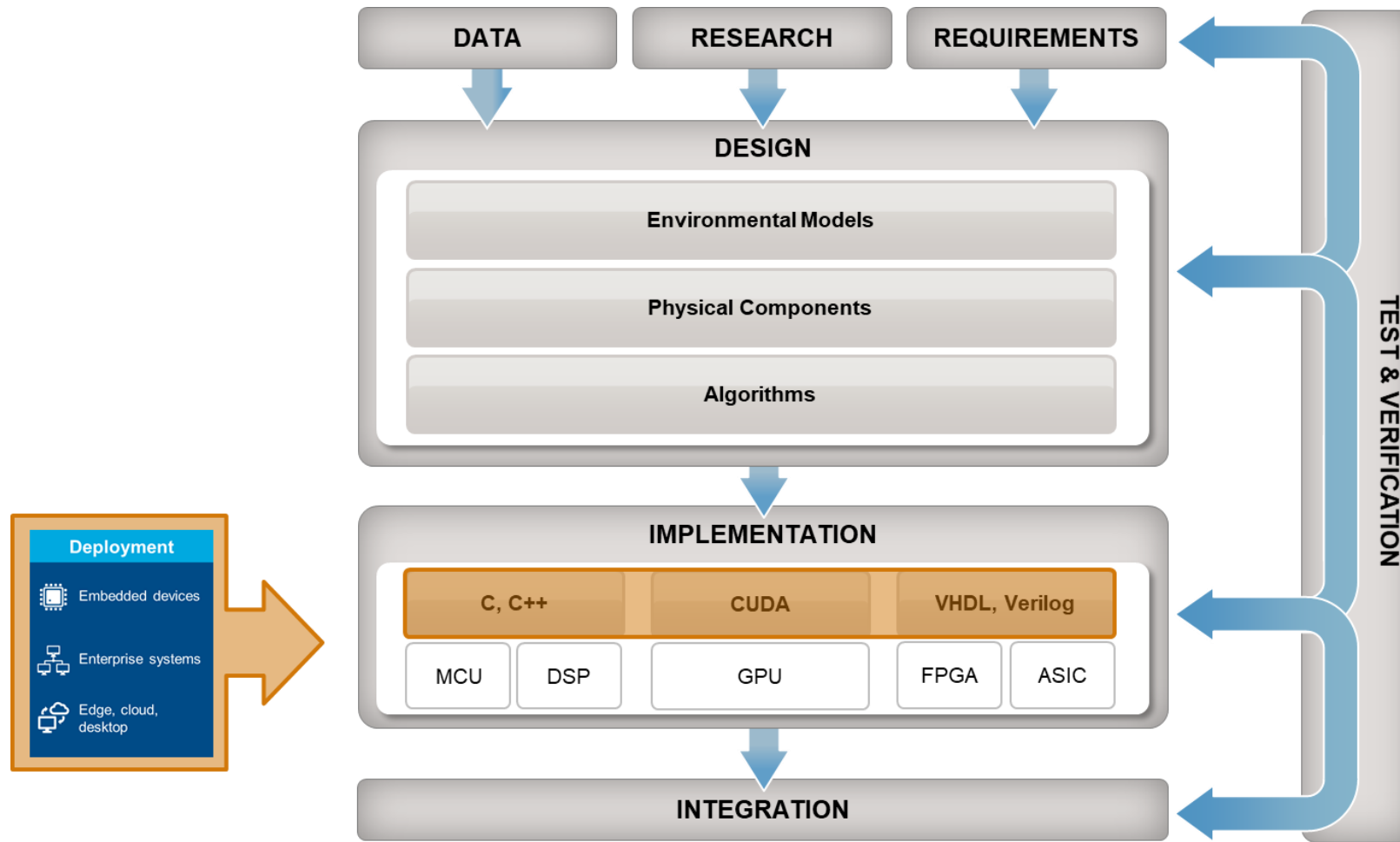
Simulink Profiler

Path	Time Plot (Dark Band = Self Time)	Total Time (s)	Self Time (s)	Number of Calls
AI_ROM		37.952	35.394	142587
> Neural ODE		1.441	0.000	0
> LSTM		0.798	0.000	0
> NLARX Sigmoid		0.148	0.000	0
From Workspace1		0.086	0.086	23794
Demux		0.057	0.057	95184
Prediction_LSTM		0.019	0.019	23793
Prediction_NeuralODE		0.003	0.003	23793
Prediction_NLARXSigmoid		0.003	0.003	23793
Prediction_NLARXSVM		0.003	0.003	23793
> Normalize		0.000	0.000	0
Cast To Double		0.000	0.000	3
> NLARX SVM		0.000	0.000	0
> Denormalize		0.000	0.000	0

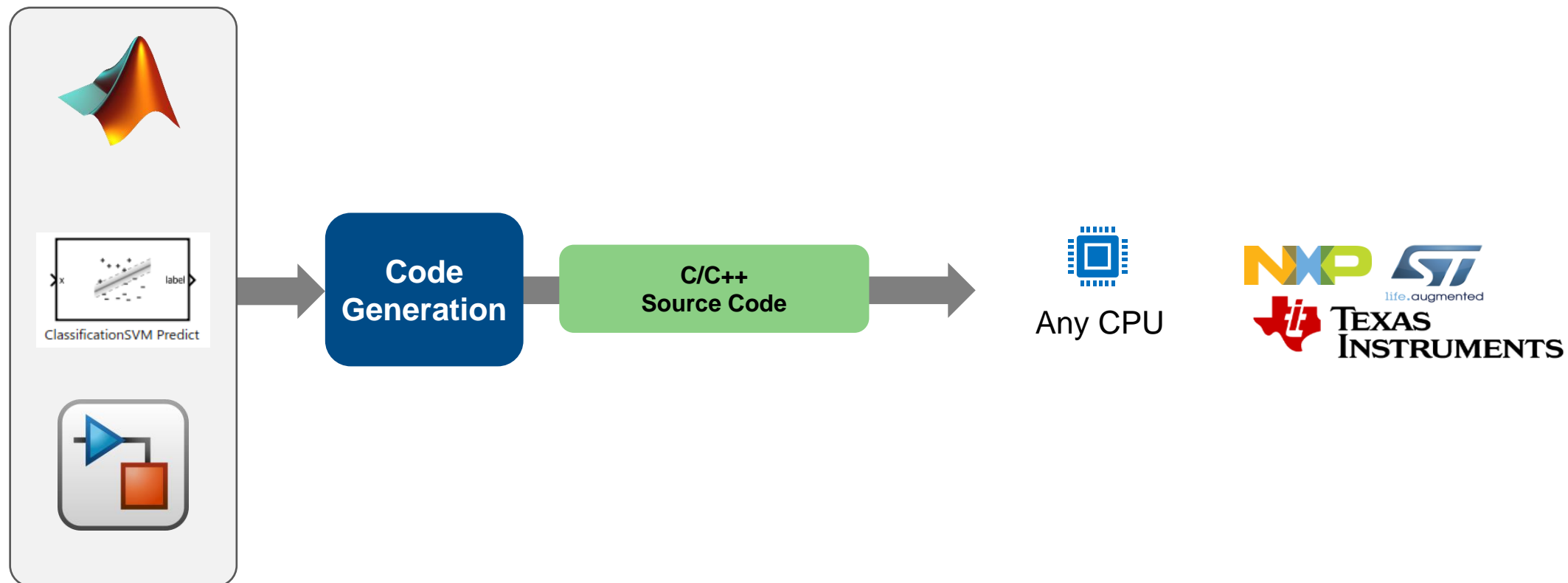
系统级仿真



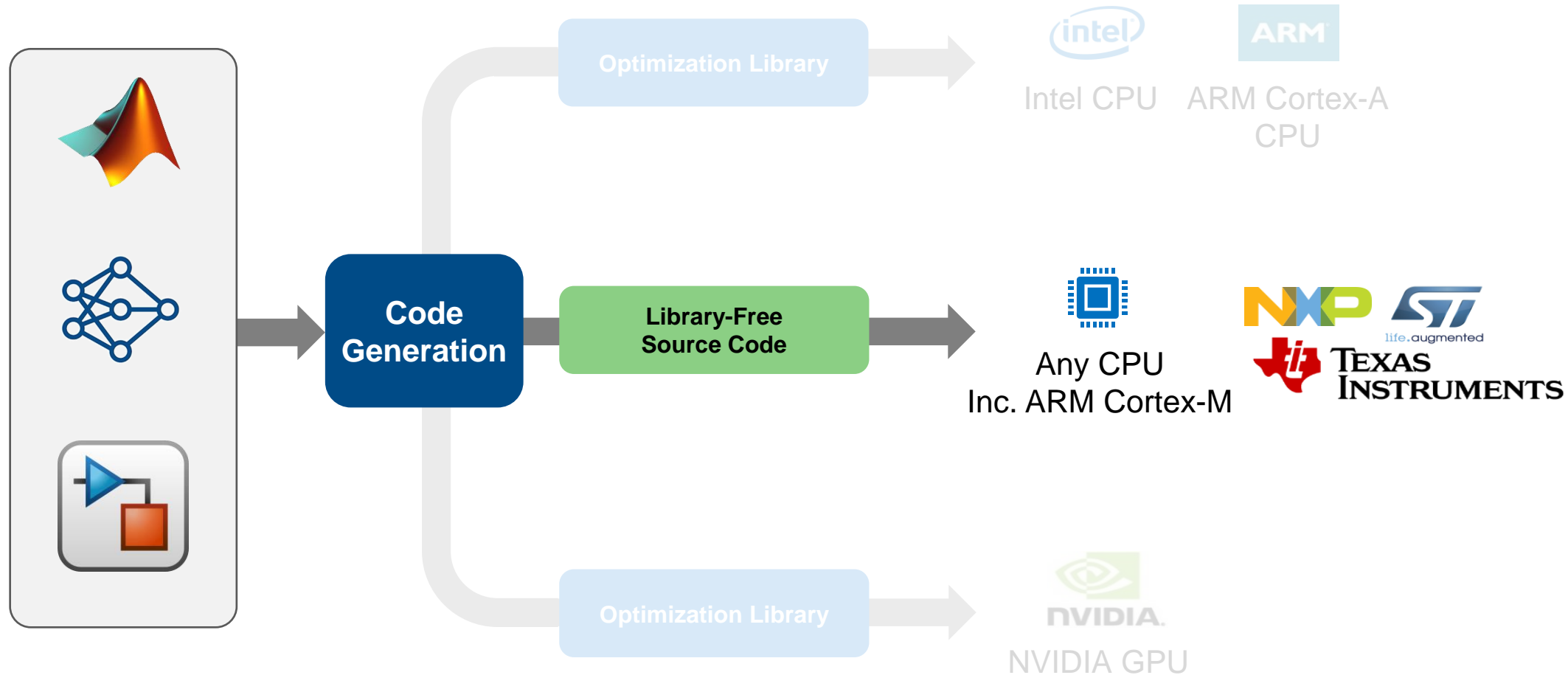
自动代码生成部署到硬件



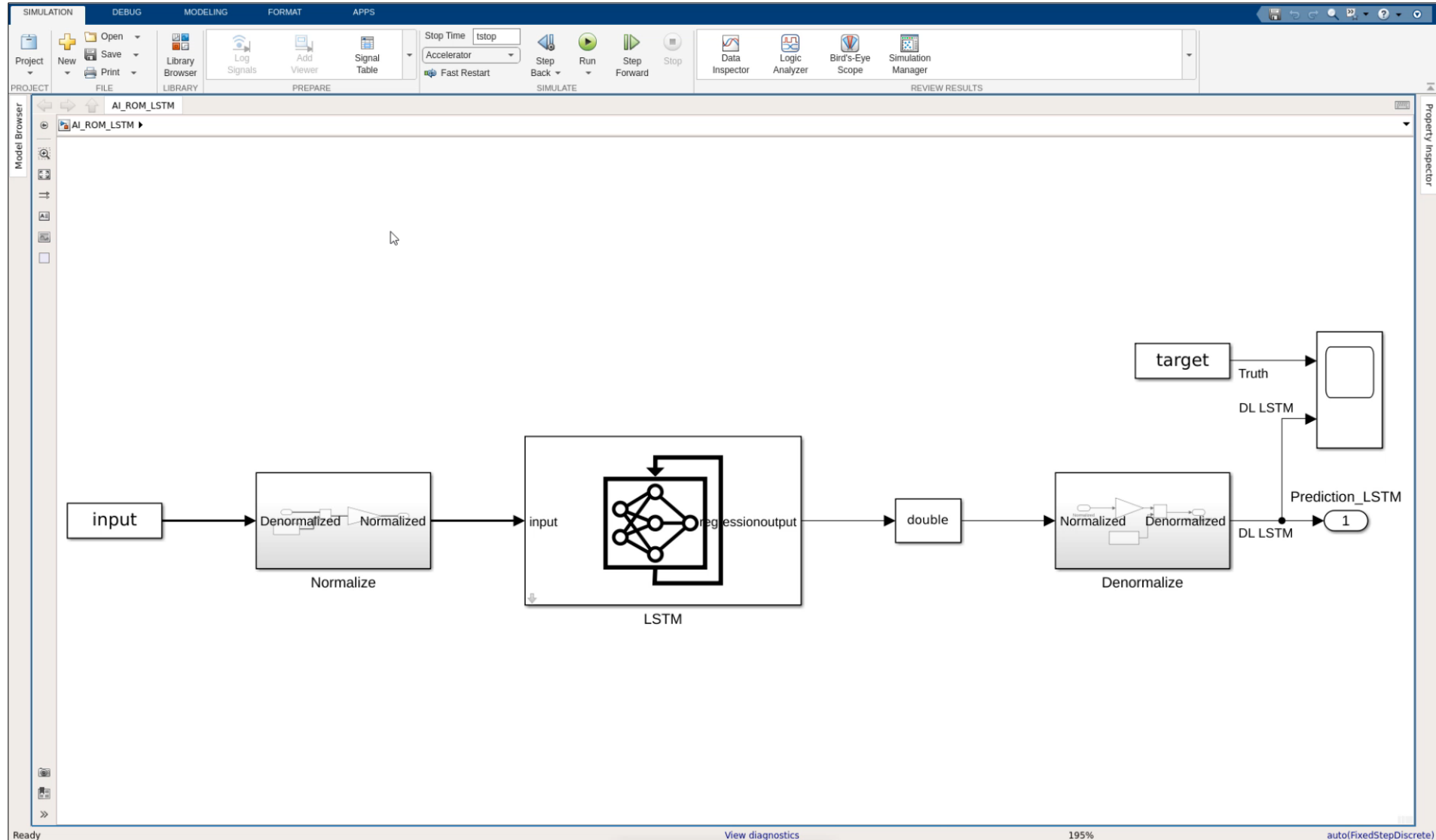
使用 Embedded Coder 生成机器学习代码



生成深度学习 Library-Free 的 C 代码

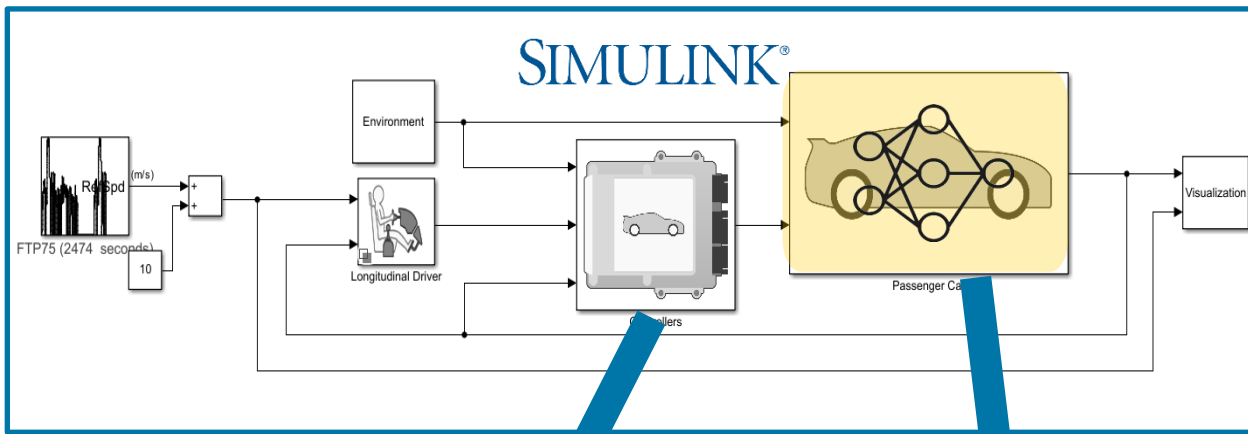


生成深度学习 Library-Free 的 C 代码



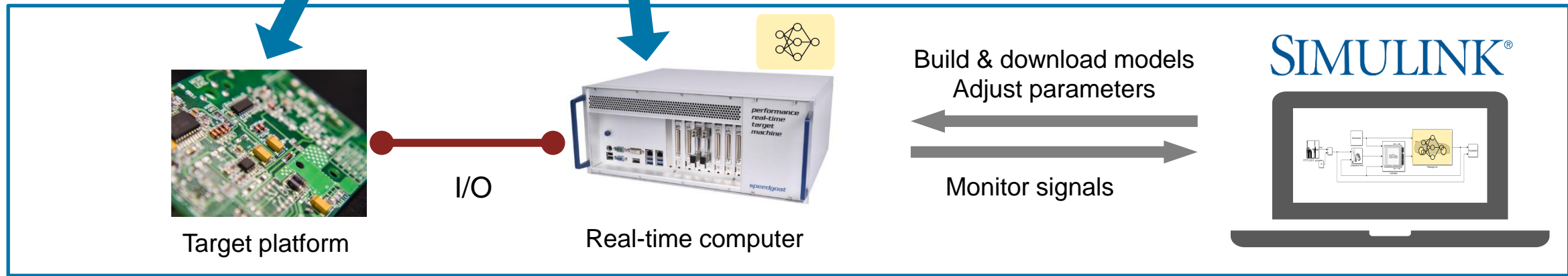
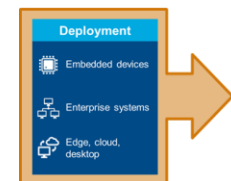
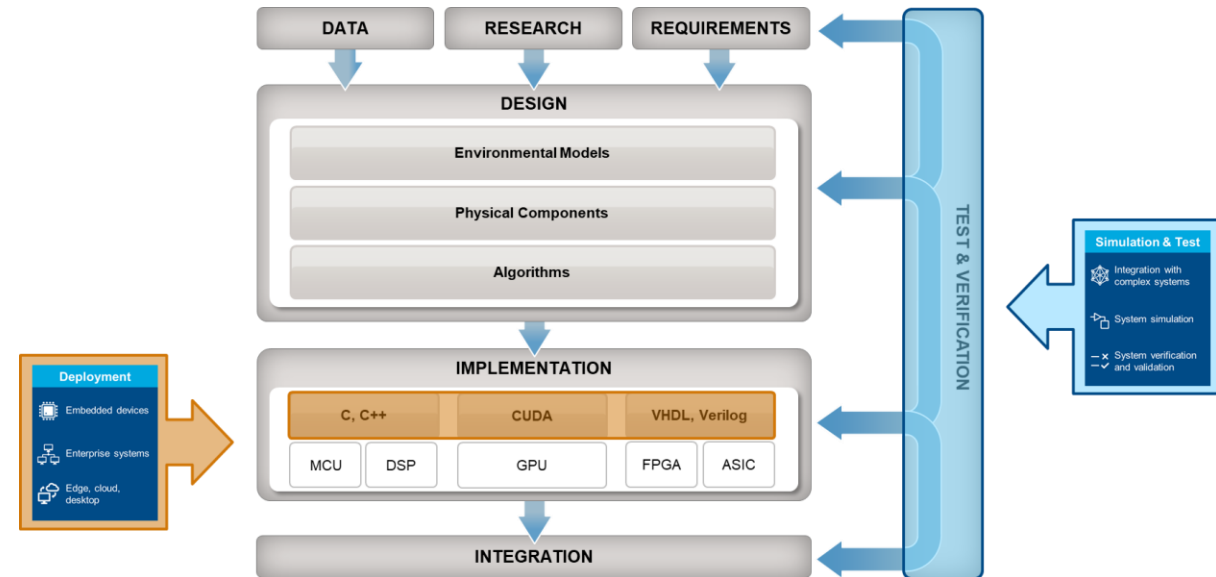
Hardware-in-the-loop simulation

系统级集成和测试



Code generation from algorithm

Code generation from plant model



Hardware-in-the-loop simulation

The screenshot displays the Simulink environment for a hardware-in-the-loop simulation. The main workspace shows a Simulink model with the following components:





















- InputData** and **Xmu** blocks feeding into a summing junction.
- A gain block labeled $1./Xsig$.
- A **Sequenceinput** block receiving the output of the gain block.
- A connection to a hardware target (Speedgoat).

Overlaid on the workspace is a large text box with the text: **Connect to HIL Simulator & Run Simulation**.

The **Simulation Data Inspector** window is open, showing a plot of the simulation results. The plot compares the **LSTM** (blue line) and **REF** (orange line) signals. The y-axis ranges from 0 to 120, and the x-axis ranges from 2 to 77. The plot shows a step-like signal that increases from approximately 45 to 120 over time, with the LSTM signal closely following the REF signal.

The Speedgoat hardware target is shown at the bottom of the workspace. It is a rack-mounted unit with the text "performance real-time target machine" and "speedgoat" visible on its front panel.

选择系统匹配的AI模型

	LSTM Long Short-Term Memory Network	Neural ODE Neural Ordinary Differential Equation	NLARX Sig Nonlinear ARX Sigmoid	NLARX SVM Nonlinear ARX Support Vector Machine
Training Speed	 *			
Interpretability				
Inference Speed				
Model Size				
Accuracy (RSME)				

Results are specific to Vehicle Engine ROM example

总结

使用基于AI的代理模型替换第一原理的发动机模型

第一原理模型和数据模型可并存

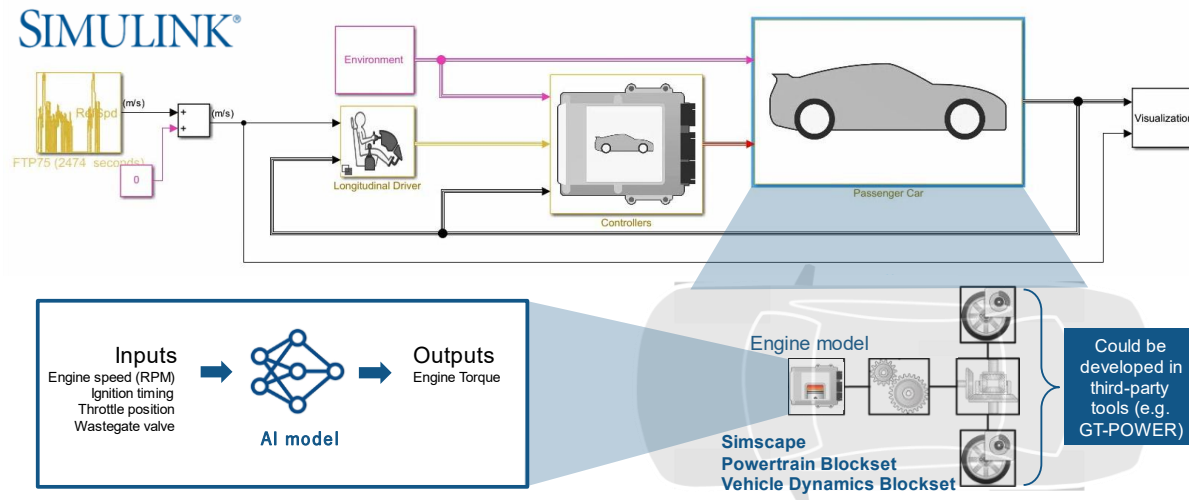
Data-driven models

First-principles models

Black-box

Gray-box

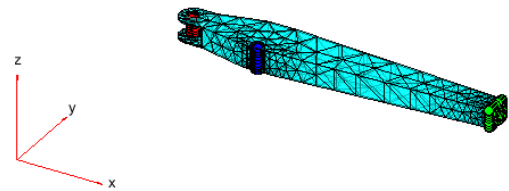
White-box



- 预处理Simulink合成数据
- 训练好的机器学习/深度学习/系统辨识模型来部分替换复杂的车辆发动机动态系统
- 将训练好的模型集成到Simulink，结合第一原理模型用于系统级仿真
- Generated C code and performed HIL tests

其他的参考示例

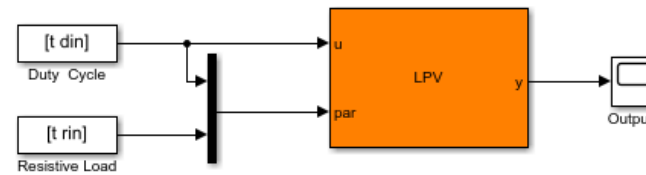
Model an Excavator Dipper Arm as a Flexible Body



Simscape Multibody,
Partial Differential Equation Toolbox

[Link](#)

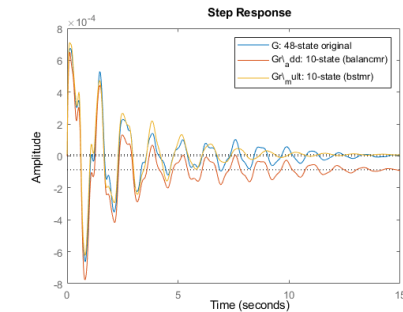
LPV Approximation of Boost Converter Model



Simscape Electrical,
Simulink Control Design

[Link](#)

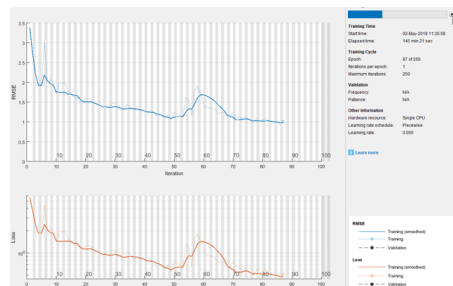
Simplifying Higher-Order Plant Models



Robust Control Toolbox

[Link](#)

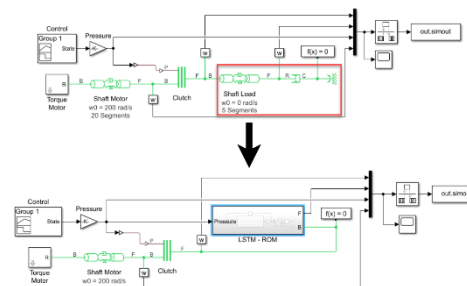
Generate a Deep Learning SI Engine Model



Deep Learning Toolbox,
Statistics and Machine Learning Toolbox

[Link](#)

Physical System Modeling Using LSTM Network in Simulink



Simulink, Simscape,
Deep Learning Toolbox

[Link](#)

Surrogate Modeling Using Gaussian Process-Based NLARX Model



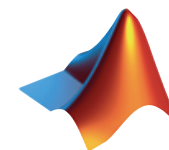
Simulink, Simscape, System Identification Toolbox,
Statistics and Machine Learning Toolbox

[Link](#)

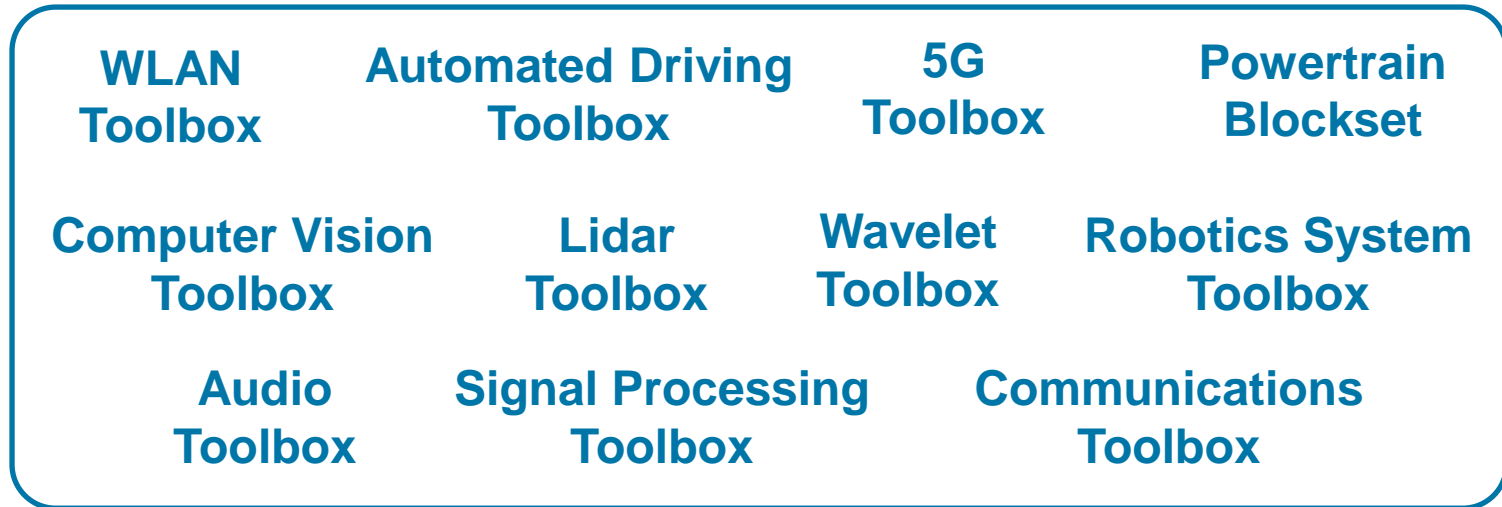
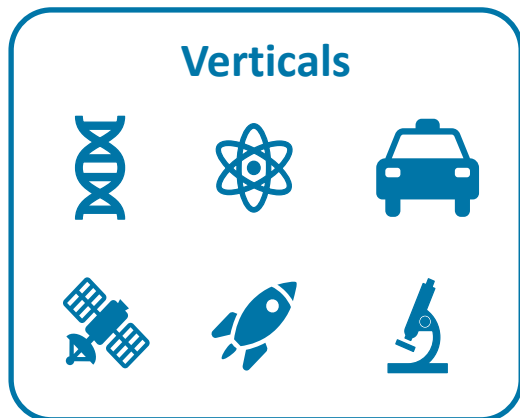


Agenda

- 动态系统与数据驱动模型简介
- 基于AI的降阶模型开发与集成
- MathWorks的AI与MBD技术支持



MathWorks Provides Capabilities to Develop and Use AI Models

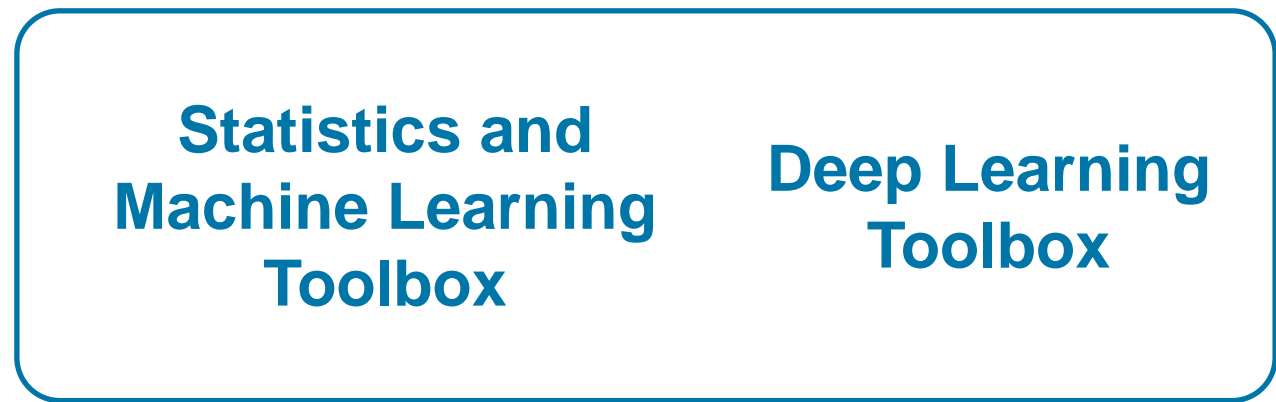


MACHINE LEARNING

K-Means	Gaussian Processes
Logistic Regression	Decision Trees
Random Forests	Support Vector Machines
PCA	Generalized Additive Models
Linear Regression	Naive Bayes
	Feedforward Neural Networks
	NARX Networks
	Autoencoders

DEEP LEARNING

CNNs	Transformers	
LSTMs	Neural ODEs	
TCN	VAE	GANs



MathWorks Engineering Support



Training



Guided Evaluations



Onsite Workshops



Consulting



Technical Support

免费入门MATLAB AI

无需下载，无需安装，只需要浏览器...



Deep Learning Onramp

Get started quickly using deep learning methods to perform image recognition.

[Details and launch](#)



Machine Learning Onramp

An interactive introduction to practical machine learning methods for classification problems.

[Details and launch](#)



Reinforcement Learning Onramp

An interactive introduction to reinforcement learning methods for control problems.

[Details and launch](#)

其他资源

深度学习入门之旅(2小时免费上机操作):

<https://www.mathworks.com/learn/tutorials/deep-learning-onramp.html>

Deep Learning Workshop (3小时上手实践)

需要的话欢迎联系我们

Deep Learning 培训 (16小时深度课程, 线上或线下)

<https://ww2.mathworks.cn/learn/training/deep-learning-with-matlab.html>

使用MATLAB进行深度学习教学 (课程开发支持, 欢迎联系我们)

<https://www.mathworks.com/academia/courseware/teaching-deep-learning-with-matlab.html>

MATLAB自带示例 (可以直接在MATLAB中运行)

<https://ww2.mathworks.cn/help/deeplearning/examples.html>

Github MATLAB深度学习资源

<https://github.com/matlab-deep-learning/>

MATLAB微信公众号

数据驱动的动态系统建模 (一) : 深度学习

数据驱动的动态系统建模 (二) : 系统辨识

2022 MathWorks 中国汽车年会

Thank you

