2024 MathWorks
中国汽车年会

从系统辨识到AI建模 从最优控制到强化学习

刘海伟, MathWorks中国

MathWorks

客户场景

住友重工为液压挖掘机加快开发嵌入式模型预测控制(MPC)软件

他们需要设计控制器，使发动机在重载装卸过程中出现负载波动时保持所需的速度

Challenges

- 加快液压挖掘机嵌入式控制器的开发，以满足开发截止日期、排放标准和安全要求
- 没有挖掘机的解析模型（或许供应商没有提供）

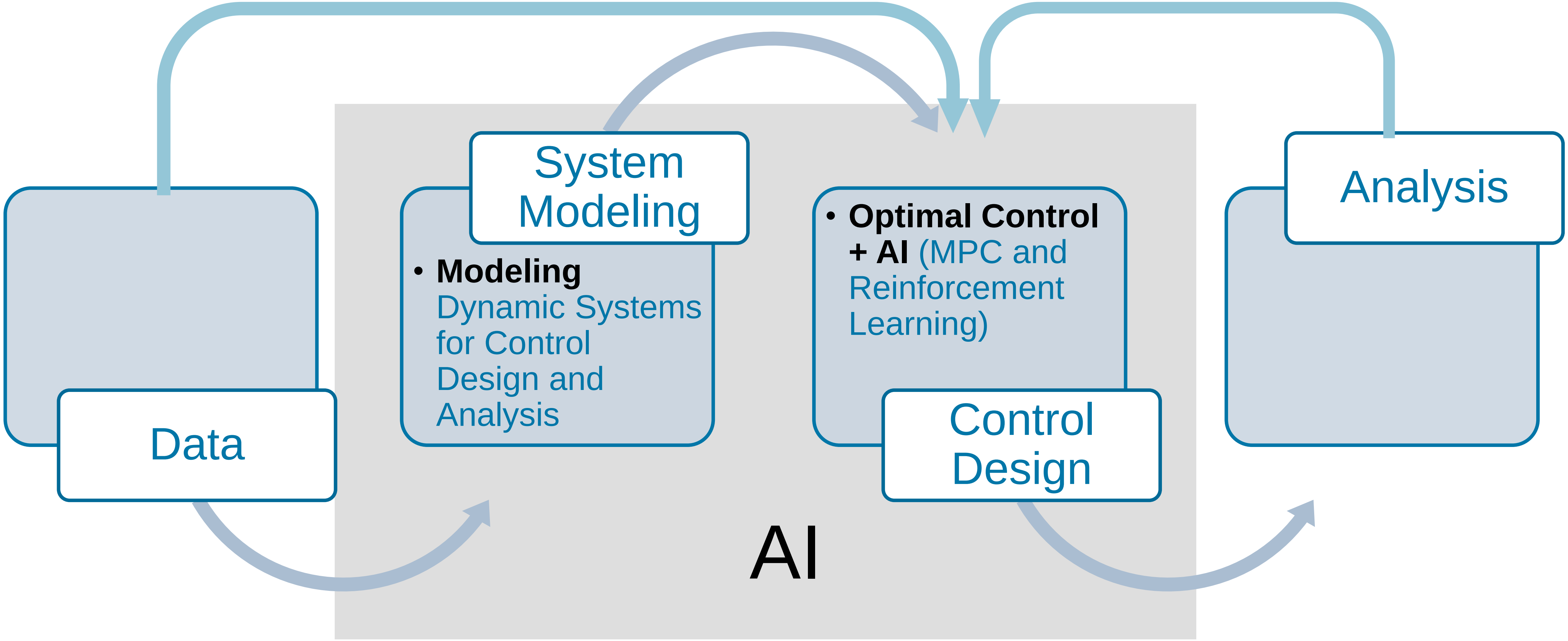
Solution

- 使用 Model Predictive Control Toolbox 开发 embedded MPC

?

A Sumitomo hydraulic excavator.

从系统辨识到AI建模 从最优控制到强化学习



Data

System Modeling

- **Modeling** Dynamic Systems for Control Design and Analysis

- **Optimal Control + AI** (MPC and Reinforcement Learning)

Control Design

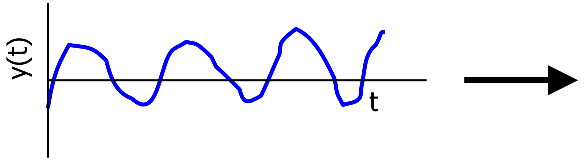
Analysis

AI

从系统辨识到AI建模

什么是系统辨识

- 使用实验数据离线或在线估计数学模型(黑盒)或在调整预知/预定义模型的参数(Grey Box)



$$\frac{s+2}{s^2+3s+4}$$

Data to Model

为什么使用系统辨识

控制系统设计

- 对象和噪声模型

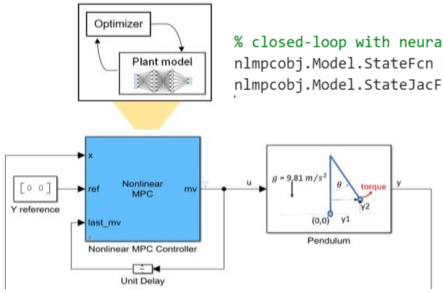
预测

- 模型降阶 噪声消除 金融分析

决策

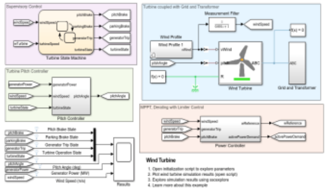
- 虚拟传感器 故障诊断 模态分析

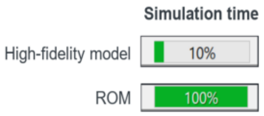
Nonlinear Control Design

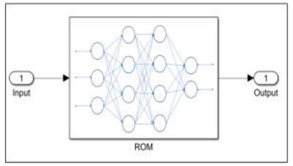


```
% closed-loop with neural state-space
nlmpcobj.Model.StateFcn = nssName;
nlmpcobj.Model.StateJacFcn = [nssName 'Jacobian'];
```

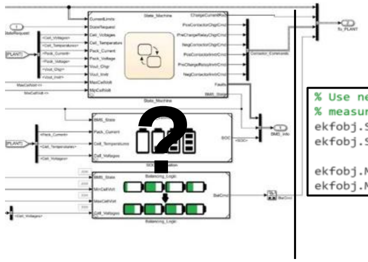
ROM







Virtual Sensor Modeling



Kalman Filters

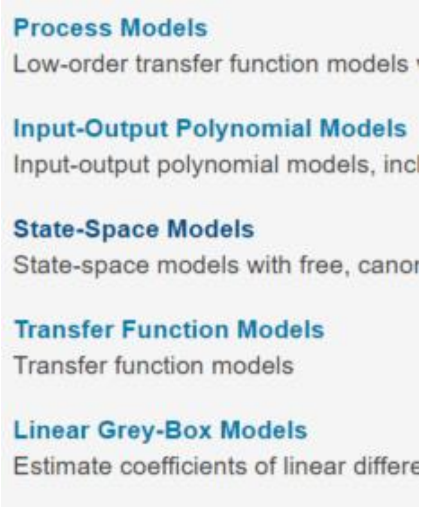
```
% Use neural state-space to specify state-transition and
% measurement function
ekfobj.StateTransitionFcn = nssStateFcn;
ekfobj.StateTransitionJacobianFcn = nssStateFcnJacobian;

ekfobj.MeasurementFcn = nssOutputFcn;
ekfobj.MeasurementJacobianFcn = nssOutputFcnJacobian;
```

从系统辨识到AI建模：系统辨识的模型类型

线性系统

- ARMAX, Box-Jenkins
- state-space, transfer function
- 过程模型 (process models)

链接



Process Models
Low-order transfer function models

Input-Output Polynomial Models
Input-output polynomial models, incl

State-Space Models
State-space models with free, canor

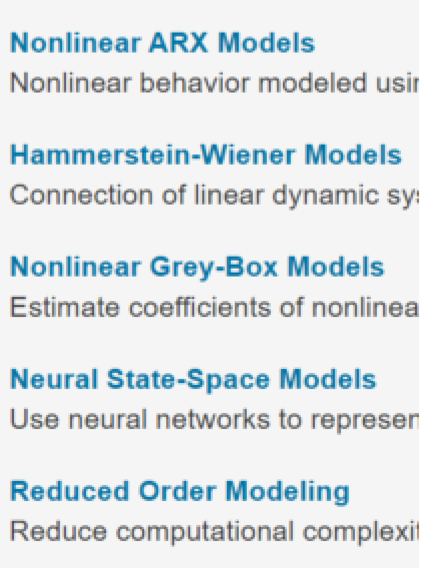
Transfer Function Models
Transfer function models

Linear Grey-Box Models
Estimate coefficients of linear differe

非线性系统

- Nonlinear ARX
- Hammerstein-Wiener

链接



Nonlinear ARX Models
Nonlinear behavior modeled usir

Hammerstein-Wiener Models
Connection of linear dynamic sy

Nonlinear Grey-Box Models
Estimate coefficients of nonlinea

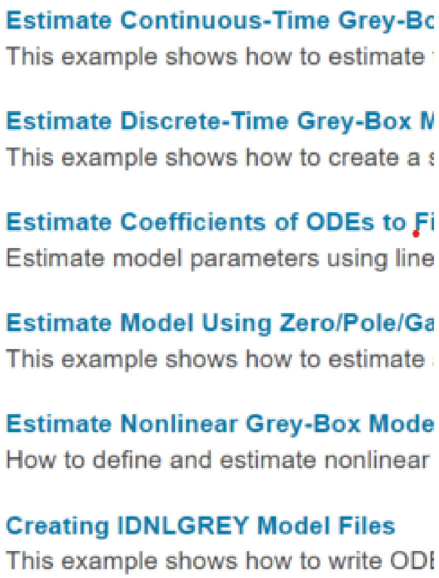
Neural State-Space Models
Use neural networks to represen

Reduced Order Modeling
Reduce computational complexit

Grey-Box估计

- 线性或非线性grey-box algorithms
- 估计预定义结构模型的待定参数

链接



Estimate Continuous-Time Grey-Bo
This example shows how to estimate

Estimate Discrete-Time Grey-Box M
This example shows how to create a

Estimate Coefficients of ODEs to Fi
Estimate model parameters using line

Estimate Model Using Zero/Pole/Ga
This example shows how to estimate

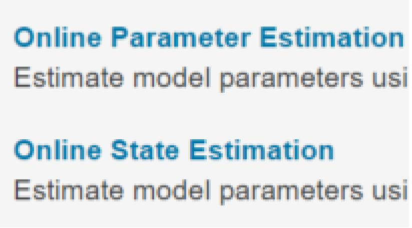
Estimate Nonlinear Grey-Box Mode
How to define and estimate nonlinear

Creating IDNLGREY Model Files
This example shows how to write ODI

在线估计

- Kalman Filter
- Recursive Least Square

链接



Online Parameter Estimation
Estimate model parameters usi

Online State Estimation
Estimate model parameters usi

深度学习

- RNN, LSTM, GRU
- 神经网络状态空间(Neural ODE)

链接

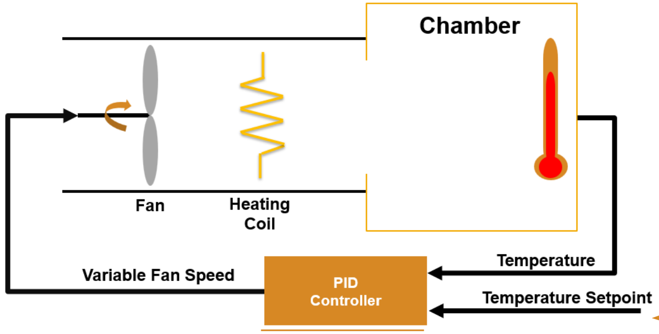
Neural State-Space Model of SI Engine T
This example describes reduced order mod

Neural State-Space Model of Simple Pen
This example shows how to design and trai

Reduced Order Modeling of Electric Vehi
This example shows a reduced order model

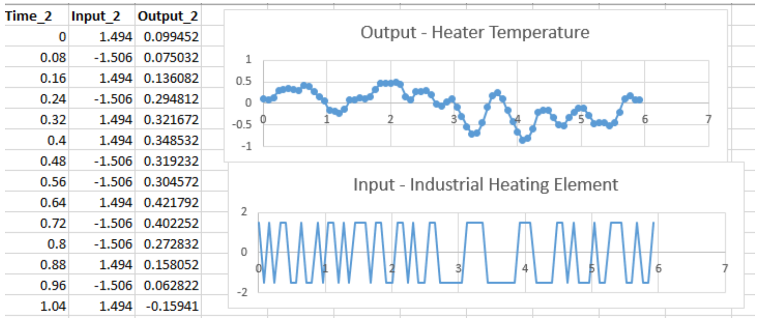
线性系统

示例:利用数据辨识温度模型提升控制器设计



实现

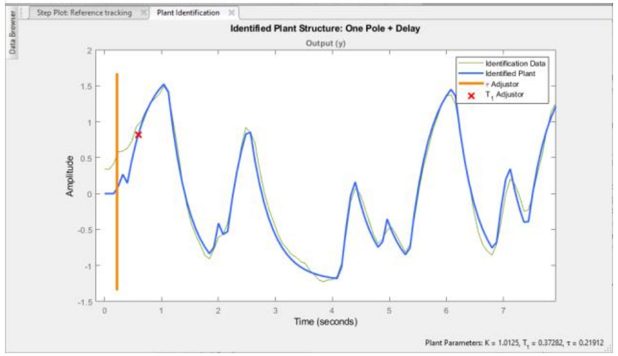
数据



Time_2	Input_2	Output_2
0	1.494	0.099452
0.08	-1.506	0.075032
0.16	1.494	0.136082
0.24	-1.506	0.294812
0.32	1.494	0.321672
0.4	1.494	0.348532
0.48	-1.506	0.319232
0.56	-1.506	0.304572
0.64	1.494	0.421792
0.72	-1.506	0.402252
0.8	-1.506	0.272832
0.88	1.494	0.158052
0.96	-1.506	0.062822
1.04	1.494	-0.15941

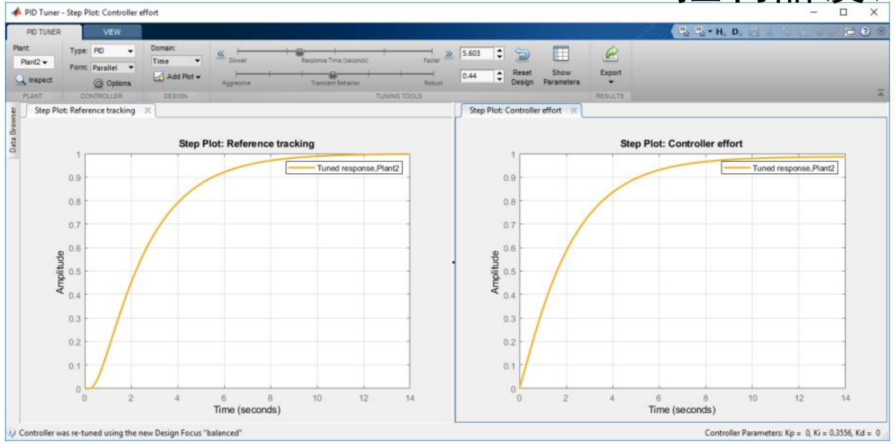
```
Process model with transfer function:
              Kp
 G(s) = ---------- * exp(-Td*s)
          1+Tp1*s

        Kp = 1.0125
       Tp1 = 0.37282
        Td = 0.21912
```

模型辨识



Plant Parameters: K = 1.0125, T_1 = 0.37282, τ = 0.21912

控制器设计



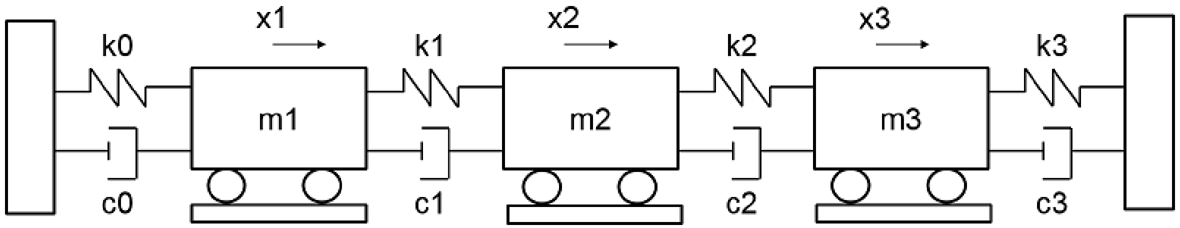
Plant: Plant2
Type: PID
Form: Parallel
Domain: Time
5.603
0.44

Controller was re-tuned using the new Design Focus "balanced"

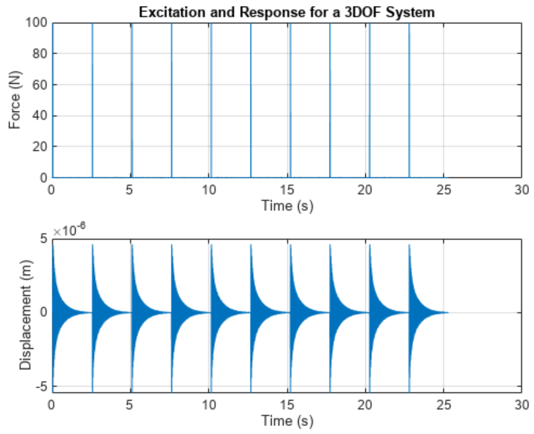
Controller Parameters: Kp = 0, Ki = 0.3556, Kd = 0

线性系统

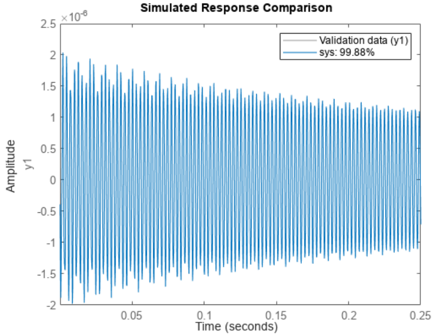
示例:利用数据辨识状态空间模型用于模态分析　ERA (特征系统实现算法)



获取数据　辨识模型　模态分析



```
order = 6;
sys = era(tt, order, 'Feedthrough', true)
```



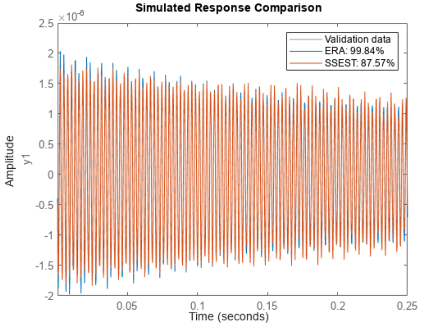
```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                x1          x2          x3          x4          x5          x6
   x1       0.8339     -0.5523   -0.001099    0.003053   -0.001497   0.0004633
   x2       0.5523      0.8323    -0.00714   -0.001638   -0.002676   0.0008873
   x3    -0.001099     0.00714      0.2293      0.9709  -0.0009399  -0.0006184
   x4    -0.003053   -0.001638     -0.9709      0.2289    0.006094   -0.002101
   x5    -0.001497    0.002676  -0.0009399   -0.006094     -0.5463     -0.8302
   x6   -0.0004633   0.0008873   0.0006184   -0.002101      0.8302     -0.5465
```

模态分析获得模态频率和阻尼系数

```
[~,f] = modalfrf(sys);
[fd, zeta] = modalfit(sys,f,3);
```

降阶模型的模态分析并对比原模型

```
order = 4;
sys = era(tt, order, 'Feedthrough', true);
```



非线性系统

Nonlinear ARX

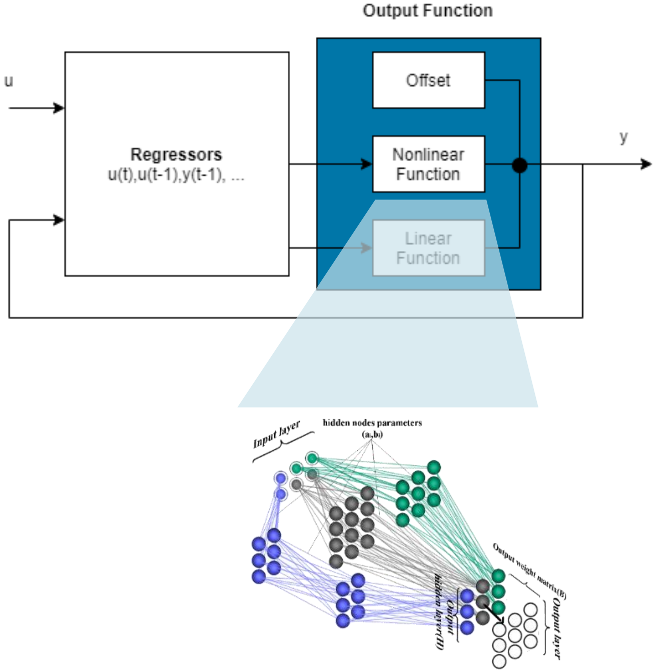
- 动态非线性估计，如小波网络，树划分，和sigmoid网络
- 机器学习和深度学习用于建模非线性动态特性

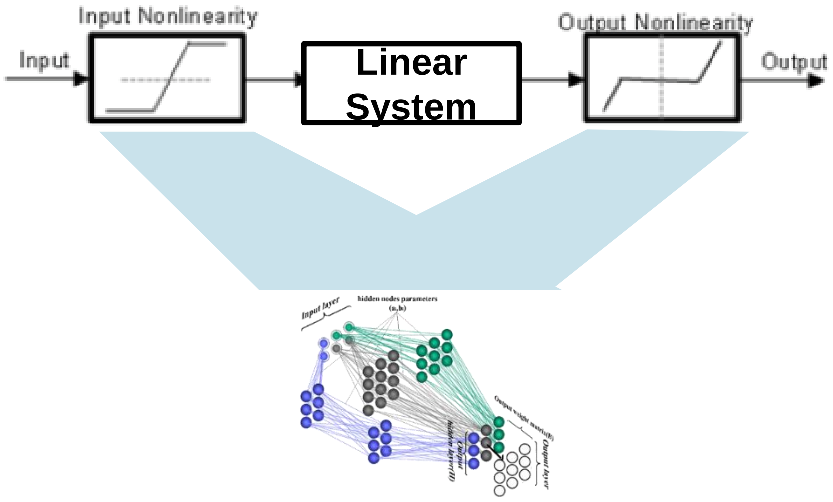
Hammerstein-Wiener

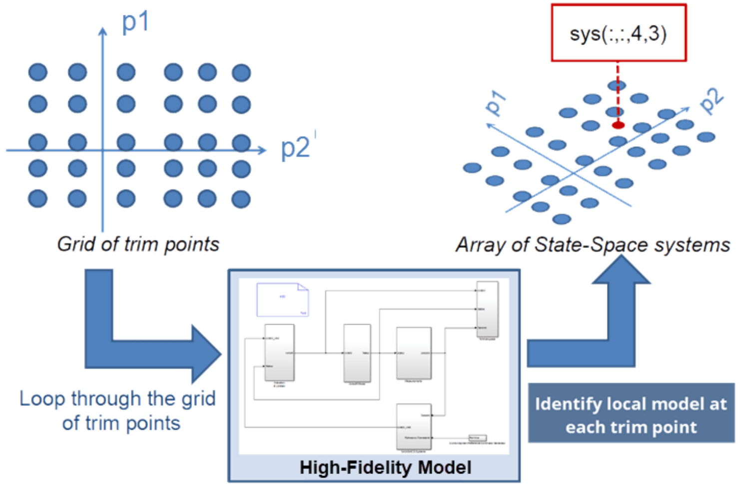
- 用线性传递函数表示动力学，用线性系统输入和输出的非线性函数捕捉非线性

LPV

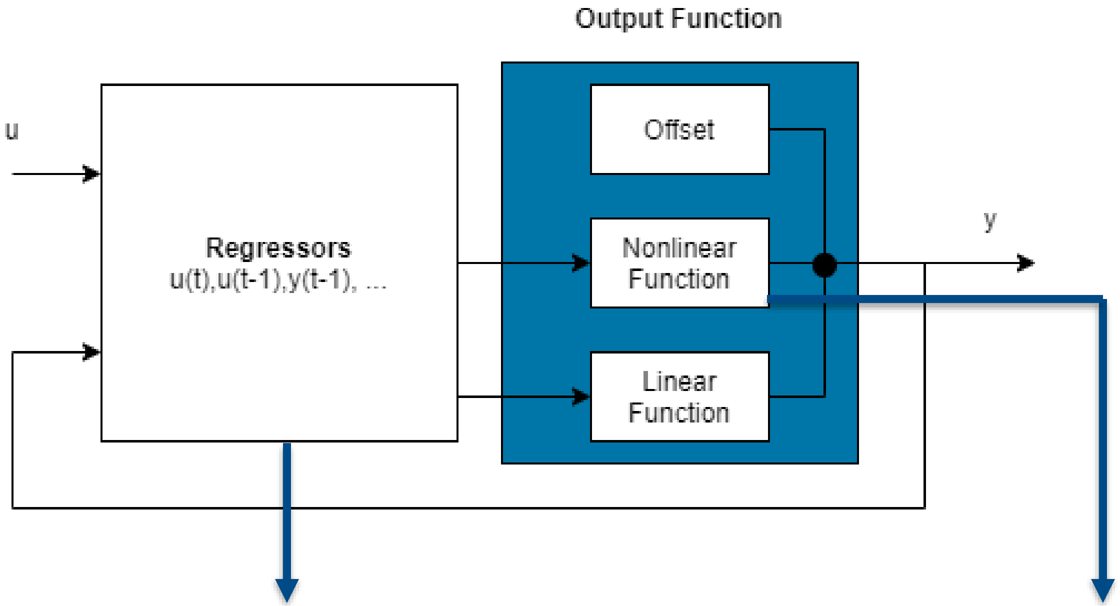
- LPV是用一组线性状态空间模型，其动态变化是某些时变参数(称为调度参数)的函数







使用非线性ARX模型建模非线性系统并嵌入物理见解和知识



Nonlinear ARX Models:

使用灵活的非线性函数扩展线性模型和模型复杂的非线性行为

- Linear regressors
- Polynomial regressors
- Periodic regressors
- Custom regressors

Ex: $\max(\min(u(t-1),100)-100)$

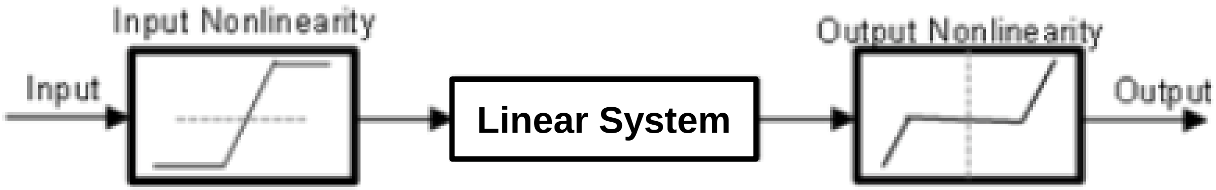
- Wavelet Network
- Sigmoid Network
- …..
- Gaussian Process
- Support Vector Machine
- Regression Tree Ensemble

General purpose nonlinear function estimators

State-of-the-art machine learning algorithms to capture nonlinearities

Requires Statistics and Machine Learning Toolbox

使用Hammerstein Wiener模型建模非线性系统并嵌入物理见解和知识



Hammerstein Wiener Models:
通过串联非线性静态修正扩展线性动态模型

- Wavelet Network
- Sigmoid Network
- …..
- Saturation
- Dead Zone
- …..
- Gaussian Process

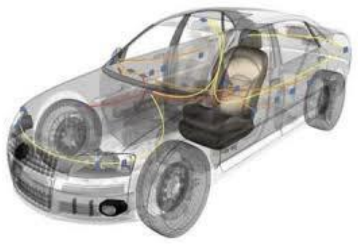
General purpose nonlinear function estimators

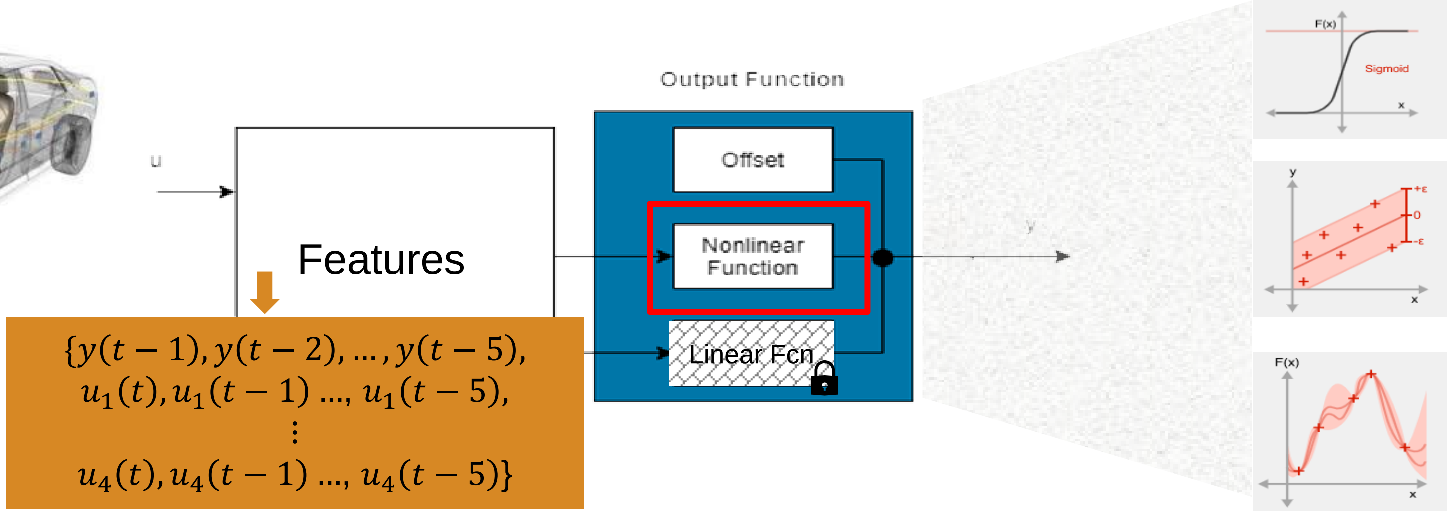
Physics inspired nonlinear estimators

Machine learning algorithm to capture nonlinearities

Requires Statistics and Machine Learning Toolbox

基于NLARX 和Machine Learning 的 SI Engine 动态建模示例





$$\begin{gathered}\{y(t-1), y(t-2), \ldots, y(t-5),\\ u_1(t), u_1(t-1) \ldots, u_1(t-5),\\ \vdots\\ u_4(t), u_4(t-1) \ldots, u_4(t-5)\}\end{gathered}$$

拟合线性模型

使用线性模型初始化NARX并固定线性模型

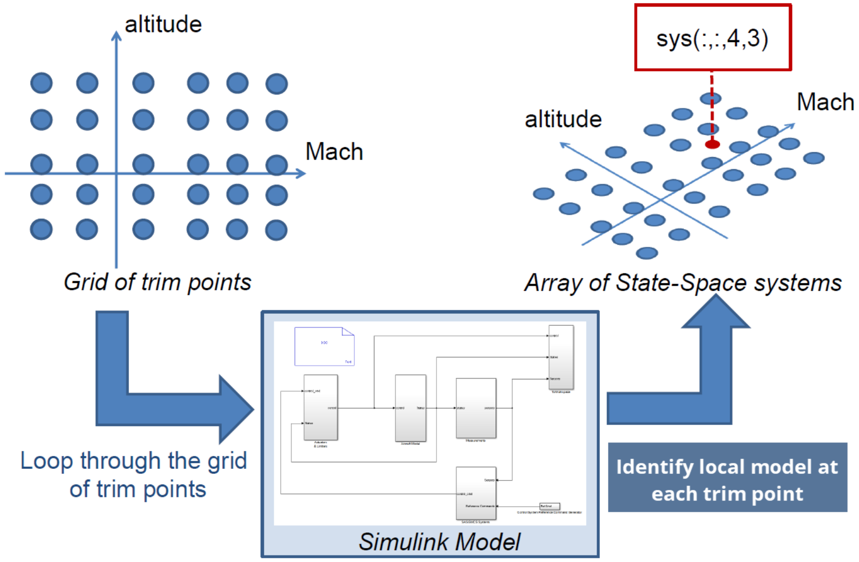
尝试GP, SVM, Sigmoid网络有用于非线性建模

使用NLARX训练模型

在验证集上比较结果

```
linsys = ssest(data, 1:10, 'Ts', 0.1);  % Linear model with automatic order selection

sys = idnlarx(linsys, idGaussianProcess); % Nonlinear ARX structure using GP
sys.OutputFcn.LinearFcn.Free = false;     % Freeze the linear component

sys.OutputFcn = idSupportVectorMachine; % Replace GP with SVM ...

opt = nlarxOptions('Display', 'on', 'SparsifyRegressors', true); % Training options
sys = nlarx(data, sys, opt);                      % Tune the free parameters of sys

compare(test_data, sys)  % validate
```

LPV(*linear parameter-varying*)



Grid of trim points

Array of State-Space systems

Loop through the grid of trim points

Simulink Model

Identify local model at each trim point

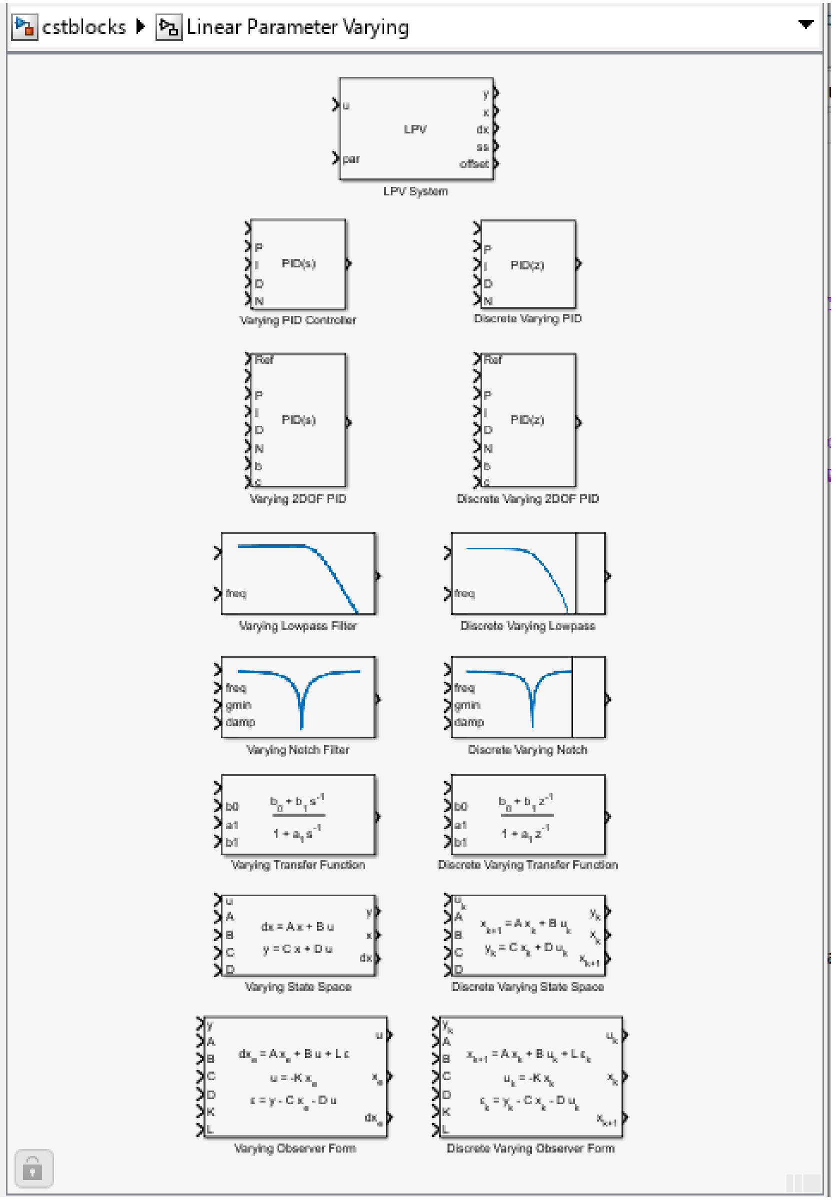
Simulink Control Design Toolbox

system

$$\dot{x} = \delta(p) + A(p)(x - x_0(p)) + B(p)(u - u_0(p))$$

$$y = C(p)(x - x_0(p)) + D(p)(u - u_0(p))$$

$$p = h(t, x, u)$$



Grey-Box估计

- 使用场景

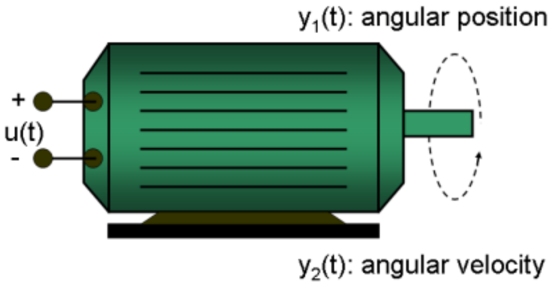
动力学模型（ODE）框架是已知或预定义的m脚本/c mex file,模型中的系数需要使用数据进行标定

$$x^{\dagger}(t) = F(t, x(t), u(t), par1, par2, ..., parN)$$
$$y(t) = H(t, x(t), u(t), par1, par2, ..., parN) + e(t)$$
$$x(0) = x0$$

$$x^{\dagger}(t) = {dx(t)}/_{dt}$$

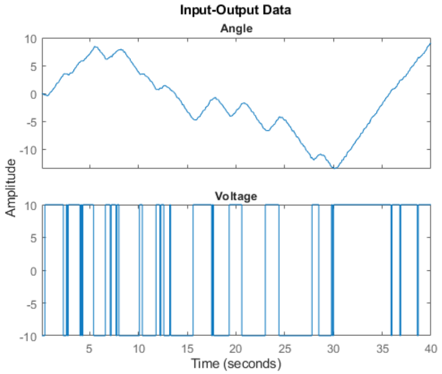
$$x^{\dagger}(t) = x(t + T_s)$$

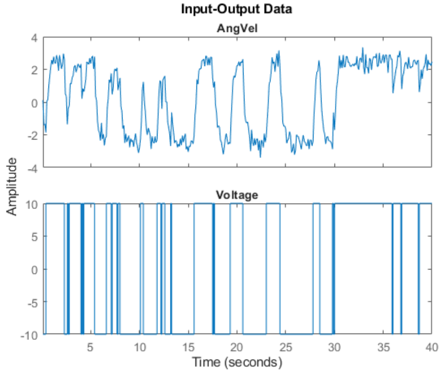
预定义的模型 实测数据 辨识模型参数

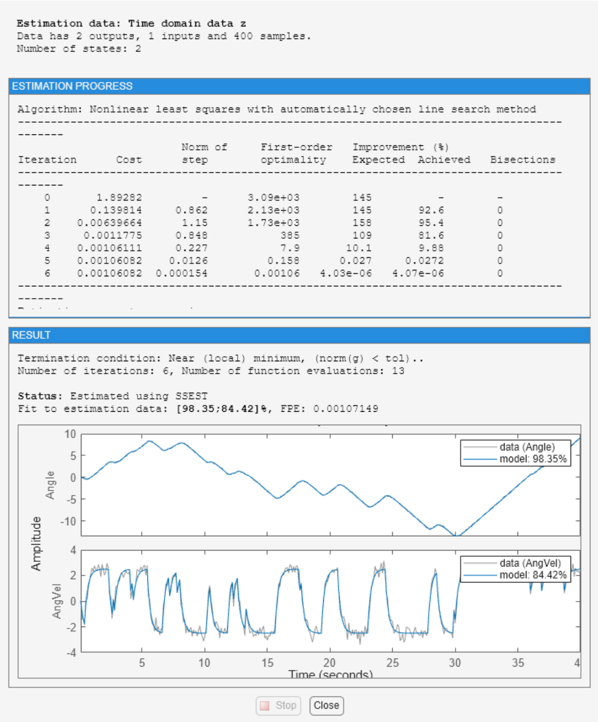


$$\frac{d}{dt}x = \begin{vmatrix} 0 & 1 \\ 0 & -th1 \end{vmatrix} x + \begin{vmatrix} 0 \\ th2 \end{vmatrix} u$$

$$y = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} x$$







```
Estimation data: Time domain data z
Data has 2 outputs, 1 inputs and 400 samples.
Number of states: 2
```

ESTIMATION PROGRESS

```
Algorithm: Nonlinear least squares with automatically chosen line search method
--------------------------------------------------------------------------------------
-------
                          Norm of     First-order    Improvement (%)
Iteration       Cost        step       optimality    Expected  Achieved    Bisections
--------------------------------------------------------------------------------------
-------
    0        1.89282          -        3.09e+03         145         -            -
    1       0.139814      0.862        2.13e+03         145      92.6            0
    2     0.00639664       1.15        1.73e+03         158      95.4            0
    3      0.0011775      0.848             385         109      81.6            0
    4     0.00106111      0.227             7.9        10.1      9.88            0
    5     0.00106082     0.0126           0.158       0.027    0.0272            0
    6     0.00106082   0.000154         0.00106    4.03e-06  4.07e-06            0
--------------------------------------------------------------------------------------
-------
```

RESULT

```
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 6, Number of function evaluations: 13

Status: Estimated using SSEST
Fit to estimation data: [98.35;84.42]%, FPE: 0.00107149
```

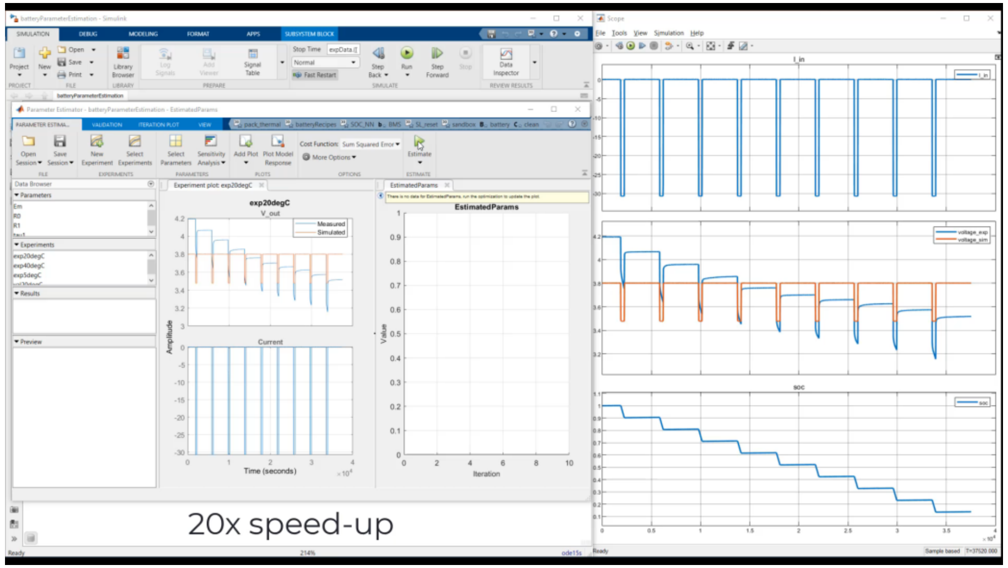
Grey-Box估计

针对Simulink模型的Grey-Box Estimation

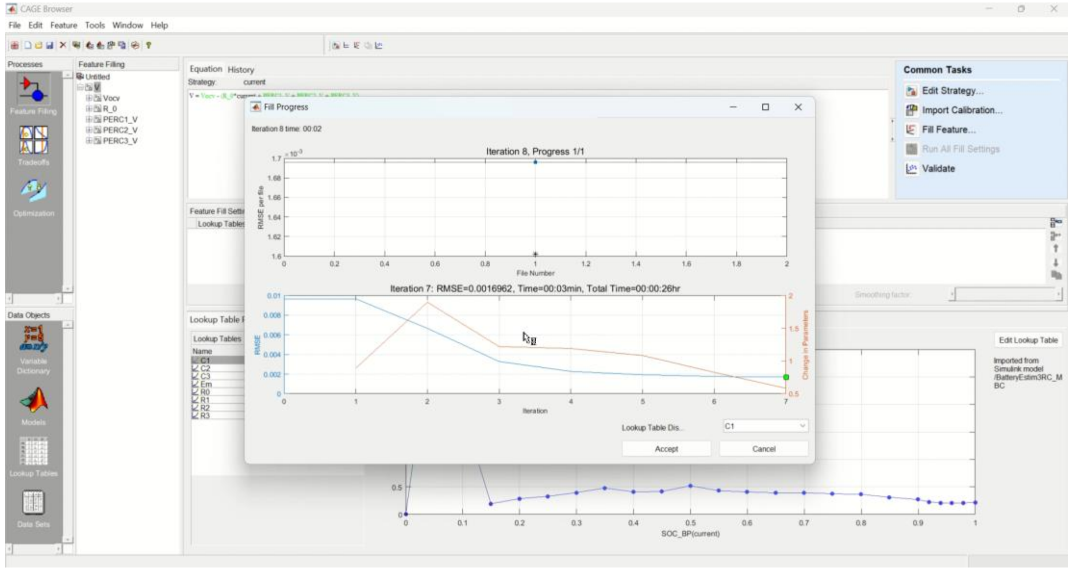
- **如果你的模型不是m脚本而是Simulink模型**

Simulink Design Optimization Toolbox

Model-Based Calibration Toolbox



Simulink Design Optimization会利用数据对模型中的位置参数进行优化求解，从而到最优参数



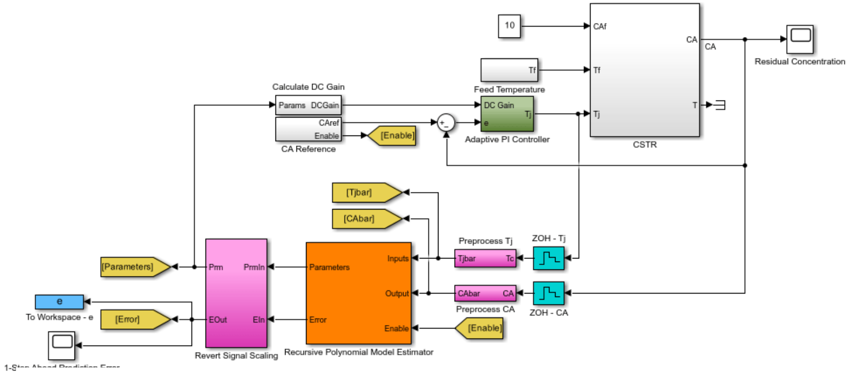
如果你的Simulink模型中有很多Look-up Table需要使用数据进行标定，可以使用Model-Based Calibration的Feature Filling

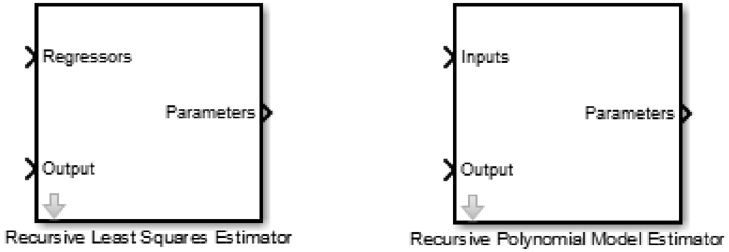
在线估计

Online Parameter Estimation

$$A(q)\bar{y}(t) = B(q)\bar{u}(t) + C(q)\bar{e}(t)$$

$$A(q) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na}$$
$$B(q) = (b_0 1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + a_{nb-1} z^{-nb+1}) z^{-nk}$$
$$C(q) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_{nc} z^{-nc}$$

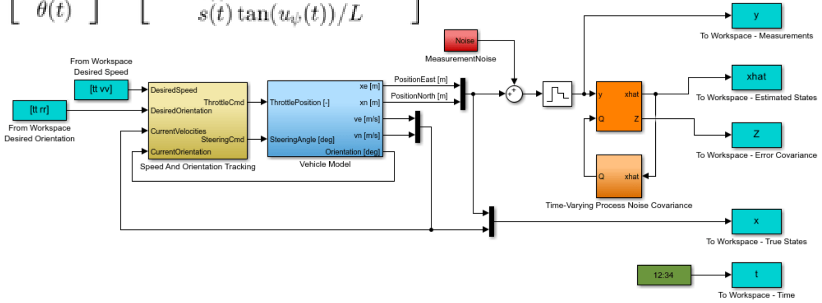


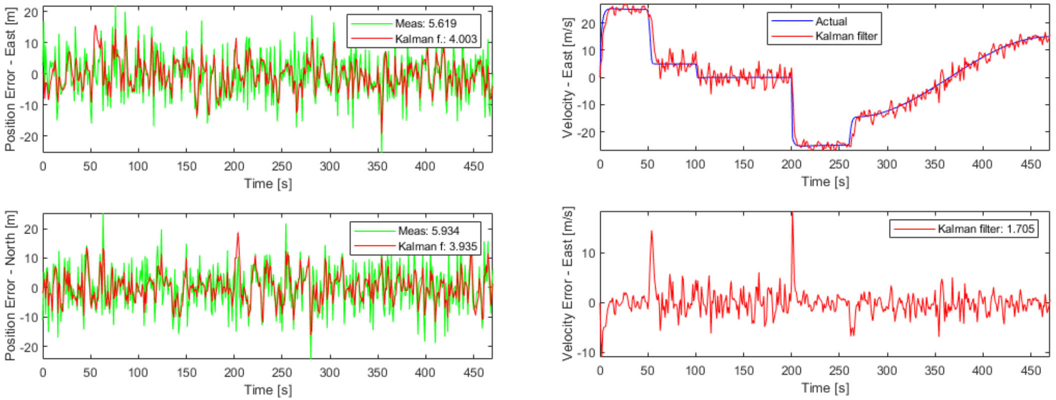


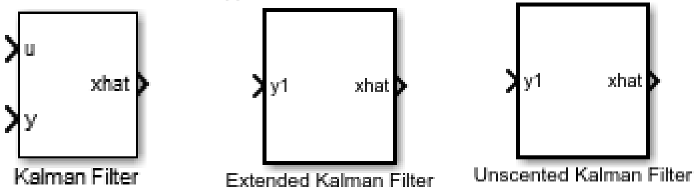
- Recursive Least Squares and Recursive Polynomial Models blocks and functions
- Recursive AR, ARX, ARMA, ARMAX, OE functions
- Blocks and functions support code generation

Online State Estimation

$$\frac{d}{dt}\begin{bmatrix} x_e(t)\\ x_n(t)\\ s(t)\\ \theta(t)\end{bmatrix} = \begin{bmatrix} s(t)\cos(\theta(t))\\ s(t)\sin(\theta(t))\\ (P\,\frac{u_T(t)}{s(t)} - A\,C_d\,s(t)^2)/m \\ s(t)\tan(u_\psi(t))/L\end{bmatrix}$$



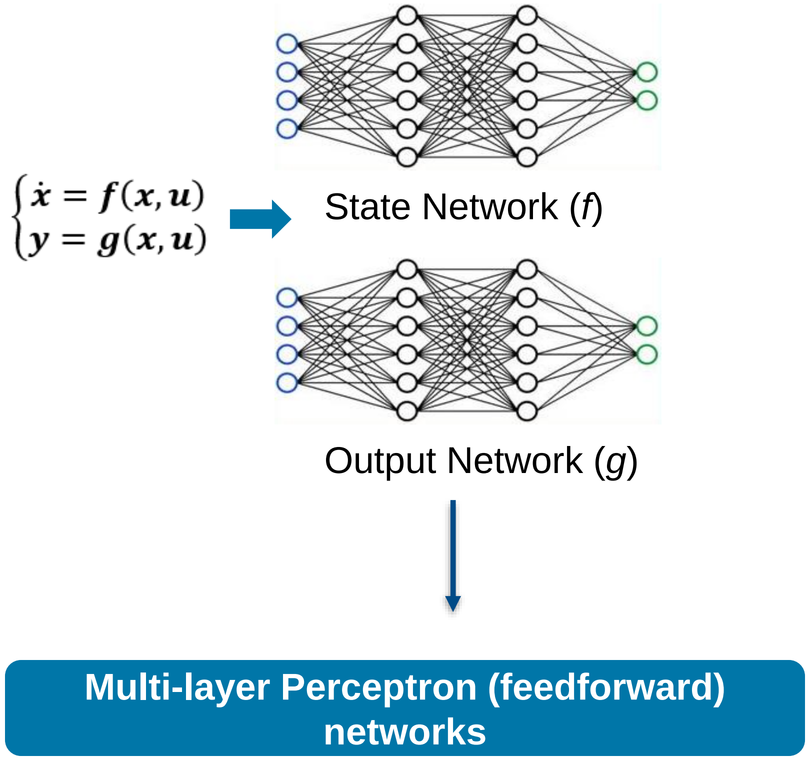




- Kalman Filter, EKF, UKF, Particle Filter

深度学习

创建基于深度学习的非线性状态空间模型



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

State Network (*f*)

Output Network (*g*)

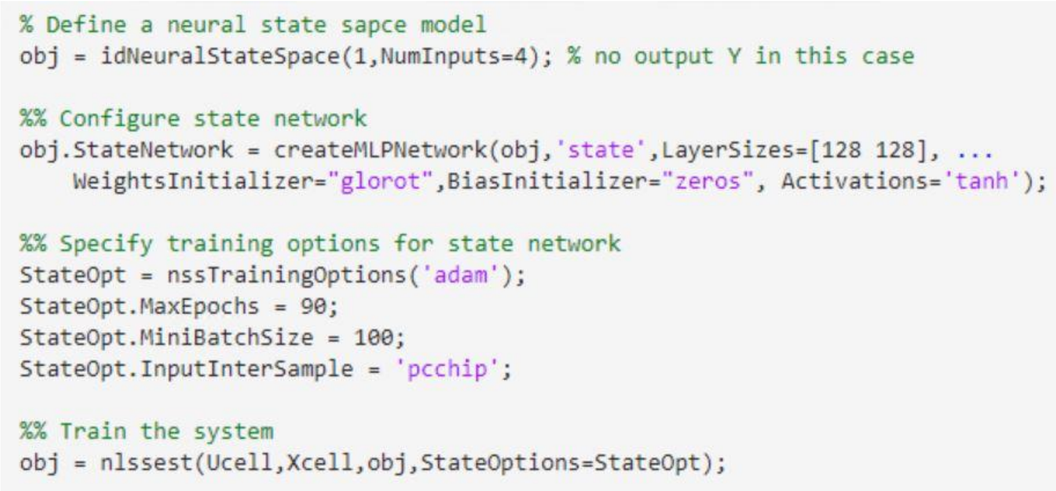
Multi-layer Perceptron (feedforward) networks

Requires Deep Learning Toolbox

Neural State-Space Models:

创建非线性状态空间模型，其中非线性状态函数f和非线性输出函数g是从数据中学习的神经网络

(popularly known as Neural ODE in deep learning community)



```
% Define a neural state sapce model
obj = idNeuralStateSpace(1,NumInputs=4); % no output Y in this case

%% Configure state network
obj.StateNetwork = createMLPNetwork(obj,'state',LayerSizes=[128 128], ...
    WeightsInitializer="glorot",BiasInitializer="zeros", Activations='tanh');

%% Specify training options for state network
StateOpt = nssTrainingOptions('adam');
StateOpt.MaxEpochs = 90;
StateOpt.MiniBatchSize = 100;
StateOpt.InputInterSample = 'pcchip';

%% Train the system
obj = nlssest(Ucell,Xcell,obj,StateOptions=StateOpt);
```

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

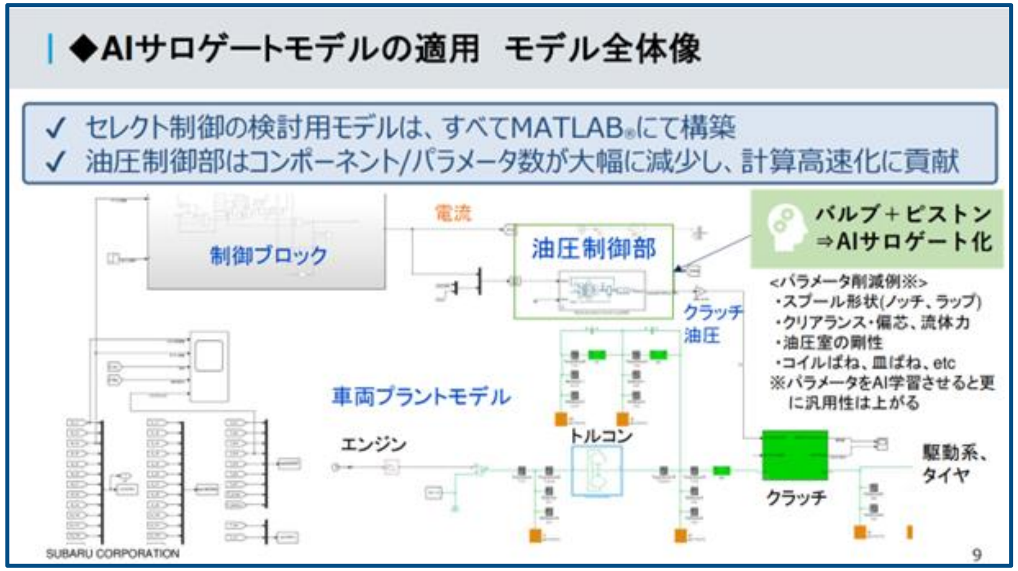
在自动档车辆中，变速器液压控制系统调节液压流体的流量，确保在不同行驶工况下平稳的换挡和高效的动力传递。

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



◆AIサロゲートモデルの適用　モデル全体像

✓ セレクト制御の検討用モデルは、すべてMATLAB®にて構築
✓ 油圧制御部はコンポーネント/パラメータ数が大幅に減少し、計算高速化に貢献

バルブ＋ピストン
⇒AIサロゲート化

<パラメータ削減例※>
・スプール形状(ノッチ、ラップ)
・クリアランス・偏芯、流体力
・油圧室の剛性
・コイルばね、皿ばね、etc
※パラメータをAI学習させると更に汎用性は上がる

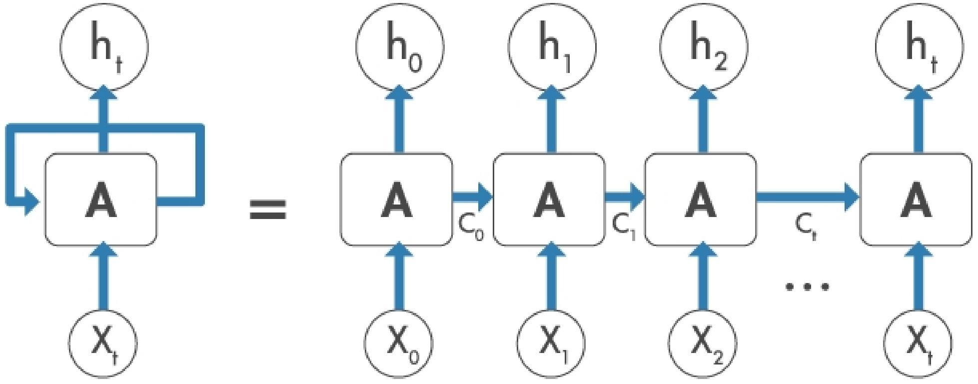
The AI surrogate model for studying selective control was built completely in MATLAB.

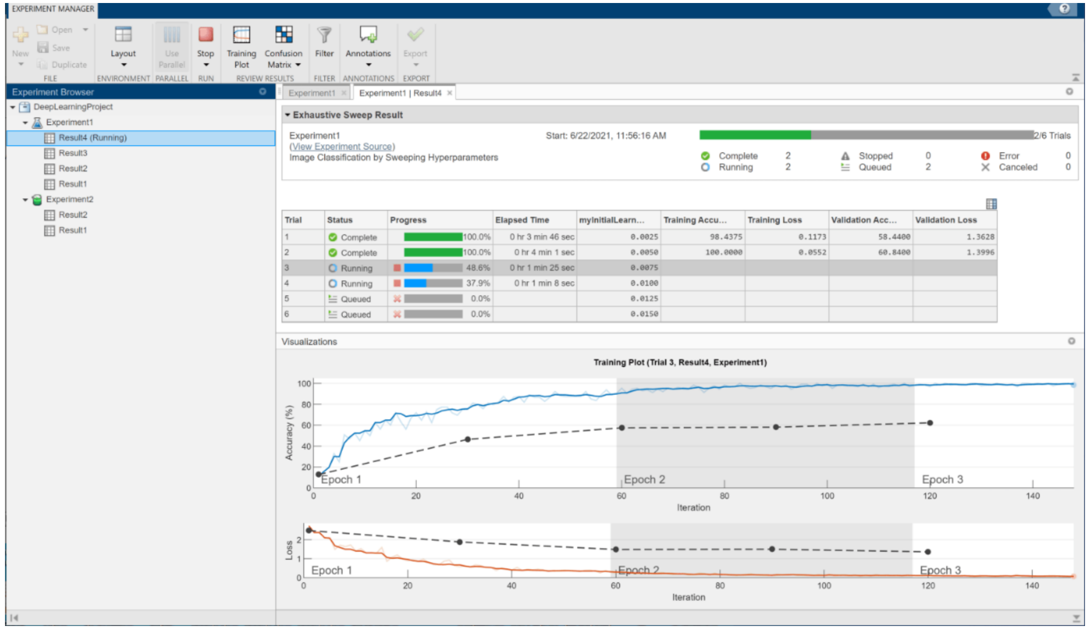
> *The AI model can now reproduce waveforms at any source pressure, oil temperature, and current. The calculation time can be significantly reduced while ensuring the accuracy of hydraulic waveforms.*

Link to user story

神经网络用于动态模型：LSTM

- E.g. Long Short Term Memory (LSTM) networks
  - \+ Memory learnt during training
  - \+ Useful for highly nonlinear dynamics
  - – Large training data sets required
  - – Long training time
  - – Low interpretability





Exhaustive Sweep Result

Experiment1
(View Experiment Source)
Image Classification by Sweeping Hyperparameters

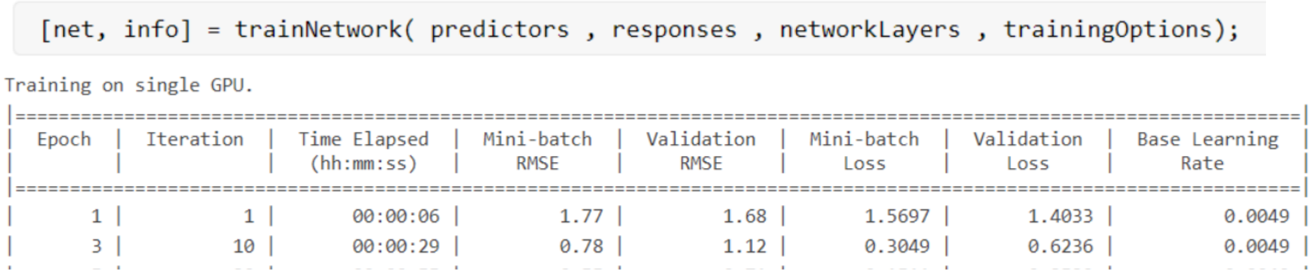
Start: 6/22/2021, 11:56:16 AM

2/6 Trials

✓ Complete	2	⚠ Stopped	0	! Error	0
○ Running	2	☰ Queued	2	✕ Canceled	0

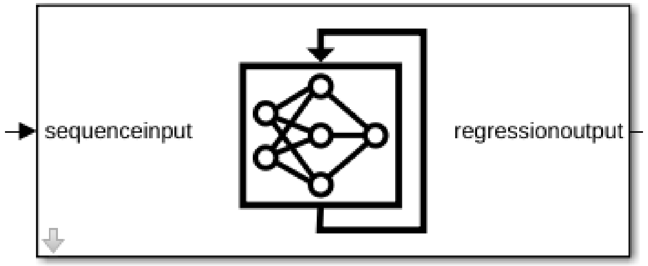
Trial	Status	Progress	Elapsed Time	myInitialLearn...	Training Accu...	Training Loss	Validation Acc...	Validation Loss
1	✓ Complete	100.0%	0 hr 3 min 46 sec	0.0025	98.4375	0.1173	58.4400	1.3628
2	✓ Complete	100.0%	0 hr 4 min 1 sec	0.0050	100.0000	0.0552	60.8400	1.3996
3	○ Running	48.6%	0 hr 1 min 25 sec	0.0075				
4	○ Running	37.9%	0 hr 1 min 8 sec	0.0100				
5	☰ Queued	0.0%		0.0125				
6	☰ Queued	0.0%		0.0150				

[Experiment Manager App]



```
[net, info] = trainNetwork( predictors , responses , networkLayers , trainingOptions);
```

```
Training on single GPU.
|=========================================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Validation  |  Mini-batch  |  Validation  |  Base Learning  |
|         |             |   (hh:mm:ss)   |     RMSE     |     RMSE     |     Loss     |     Loss     |      Rate       |
|=========================================================================================================|
|       1 |           1 |       00:00:06 |         1.77 |         1.68 |       1.5697 |       1.4033 |          0.0049 |
|       3 |          10 |       00:00:29 |         0.78 |         1.12 |       0.3049 |       0.6236 |          0.0049 |
```

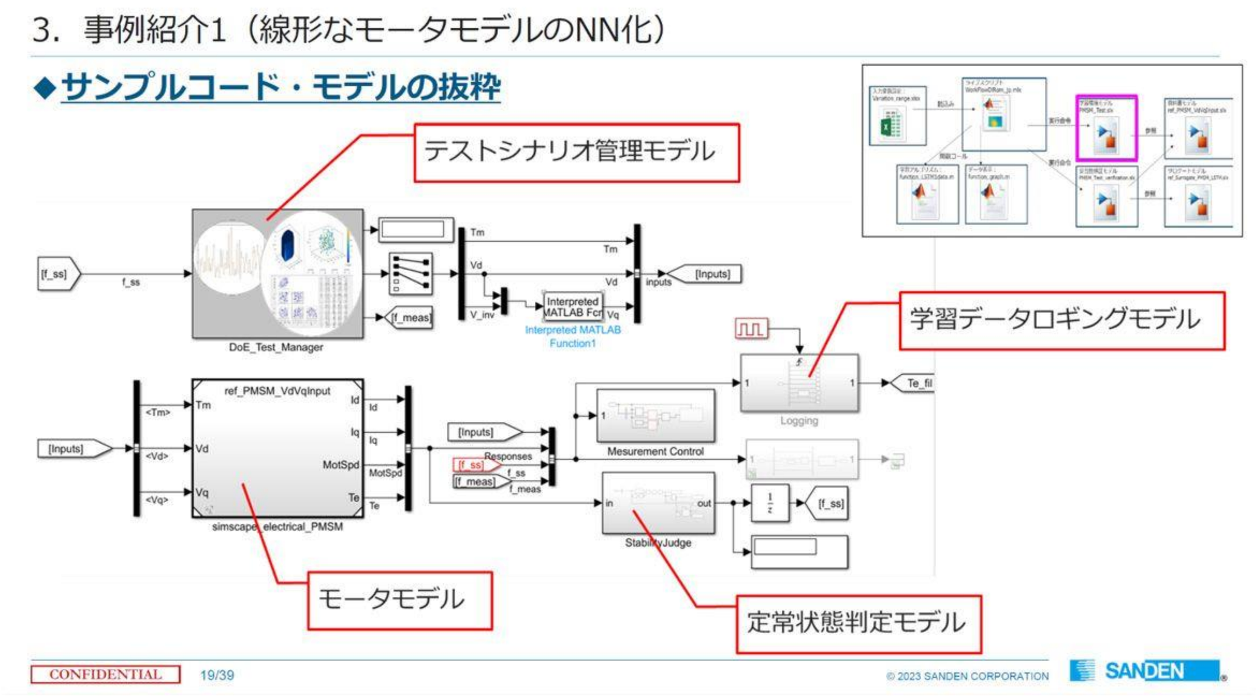


LSTM Network - Stateful Predict

Deep Learning Toolbox

Sanden 利用AI代理模型提高汽车空调的建模效率

- 建立了高精度的汽车空调系统LSTM替代模型
- 计算速度提高了100倍
- 成功的建模建立了一个纳入现实世界的参数的方法，包括热量



3. 事例紹介1（線形なモータモデルのNN化）

◆サンプルコード・モデルの抜粋

CONFIDENTIAL 19/39 © 2023 SANDEN CORPORATION SANDEN

链接

客户场景

住友重工为液压挖掘机加快开发嵌入式模型预测控制(MPC)软件

缺乏发动机模型，但他们需要设计控制器，使发动机在重载装卸过程中出现负载波动时保持所需的速度

Challenges

- 加快液压挖掘机嵌入式控制器的开发，以满足开发截止日期、排放标准和安全要求
- 没有挖掘机的解析模型（或许供应商没有提供）

Solution

- **使用系统辨识从发动机数据中估计模型**
- **在MPC中，使用辨识的模型作为内部预测模型**
- **使用Embedded Coder生成代码并部署到控制器**

Results

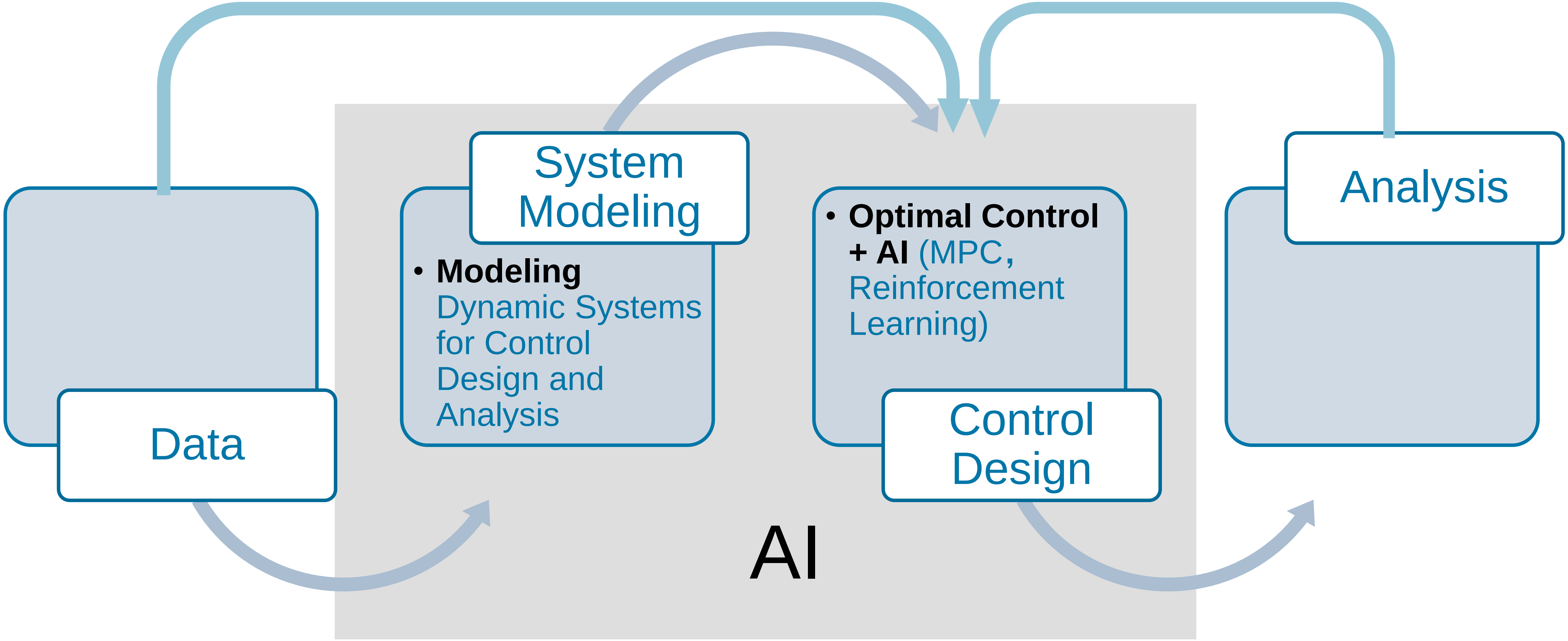
- 燃油效率提高了15%
- 工程工作量减少了50%
- 紧迫的截止日期

Link to user story

A Sumitomo hydraulic excavator.

> *"Sumitomo Construction Machinery achieved a 15% reduction in fuel consumption without sacrificing the excavator's dynamic performance. The increase in efficiency was due, in part, to a 50% reduction in engine speed fluctuations made possible by Model Predictive Control Toolbox and our improved control design.*
>
> *- Eisuke Matsuzaki, Sumitomo Heavy Industries*

从系统辨识到AI建模 从最优控制到强化学习



Data

System Modeling

- **Modeling** Dynamic Systems for Control Design and Analysis

- **Optimal Control + AI** (MPC, Reinforcement Learning)

Control Design

Analysis

AI

不同类型的最优控制问题

Linear Quadratic Control (LQ)

Optimal control where both the system model and performance objective are represented by linear and quadratic functions, respectively.

$$J(u) = \int_0^\infty (x^TQx + u^TRu + 2x^TNu)dt$$

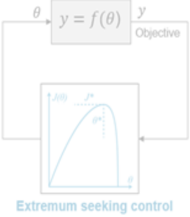
Subject to: $\dot{x} = Ax + Bu$
$u = -Kx$

H-infinity synthesis

An optimal control tool/technique for designing single-input single-output (SISO) or MIMO feedback controllers to achieve robust performance and stability

Extremum Seeking Control (ESC) 极值搜索

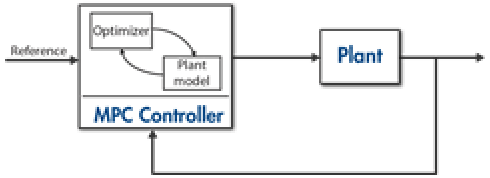
Optimal control method that learns optimal solution through perturbation of control parameters or input signals



Extremum seeking control

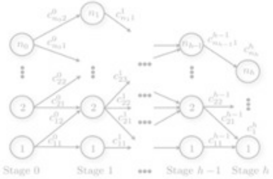
Model Predictive Control 强化学习

Receding horizon approach that solves a constrained optimization problem.



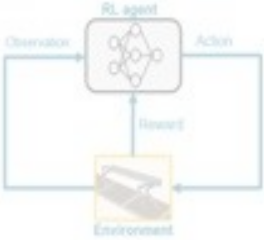
Dynamic Programming 动态规划

Problem-solving technique that involves breaking down a complex problem into smaller subproblems and storing the solutions to these subproblems in memory.



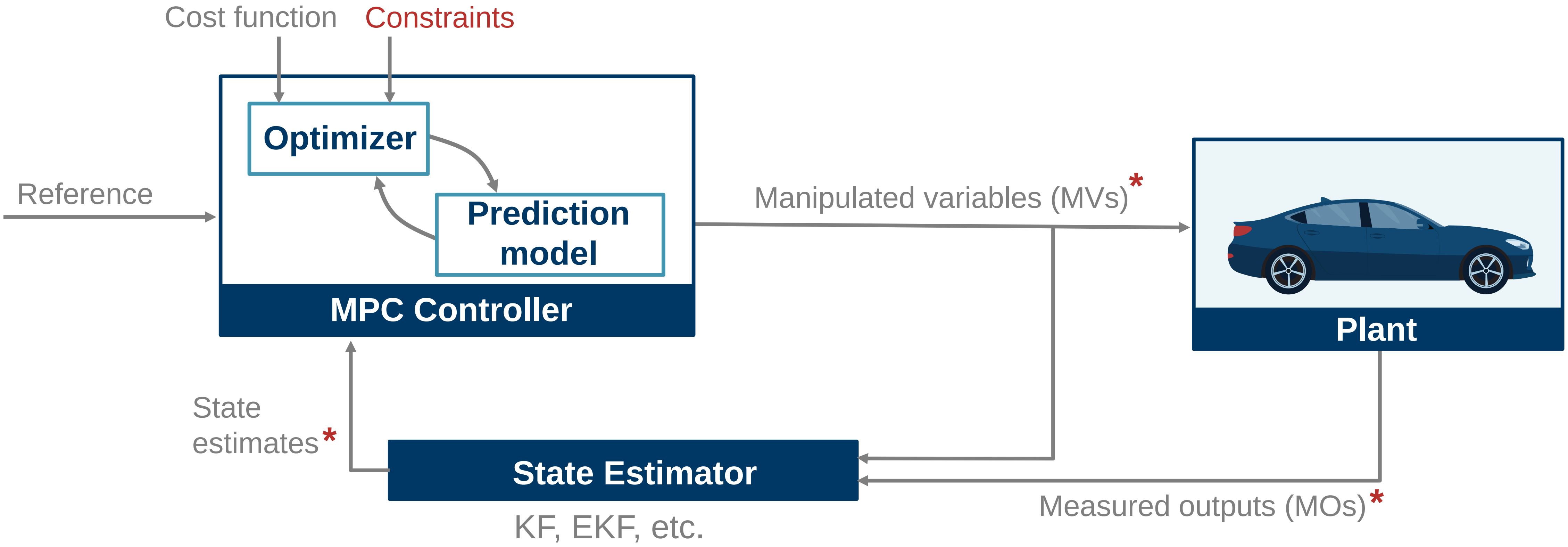
Reinforcement Learning 强化学习

Optimal control technique that trains an ‘agent’ through trial & error interactions with an environment



MPC模型预测控制

- MIMO control technique that solves a constrained optimization problem in receding horizon fashion



Cost function

Constraints

Reference

Optimizer

Prediction model

MPC Controller

Manipulated variables (MVs)*

Plant

State estimates*

State Estimator

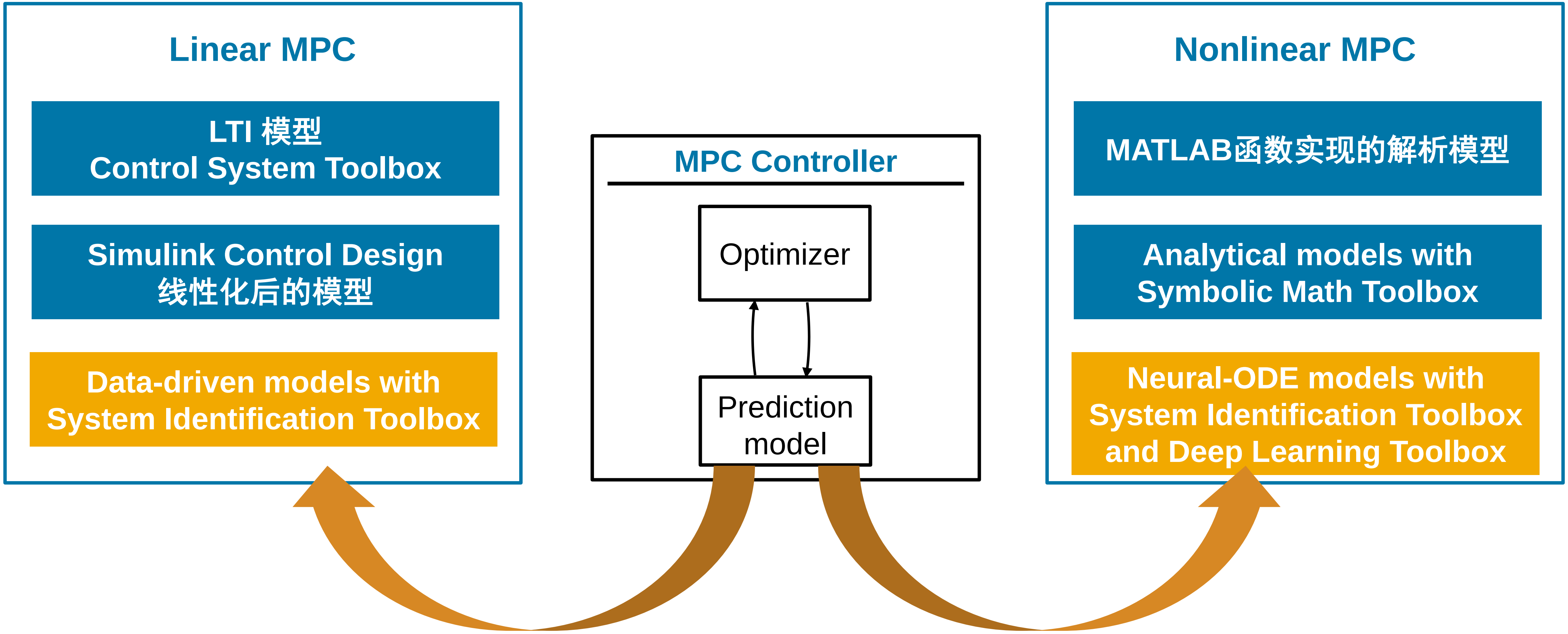
KF, EKF, etc.

Measured outputs (MOs)*

Prediction model specification

🟦 第一原理模型

🟧 数据驱动模型



Linear MPC

LTI 模型
Control System Toolbox

Simulink Control Design
线性化后的模型

Data-driven models with
System Identification Toolbox

MPC Controller

Optimizer

Prediction model

Nonlinear MPC

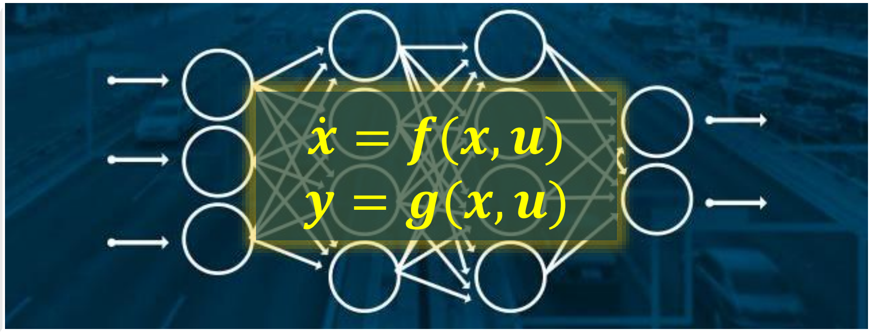
MATLAB函数实现的解析模型

Analytical models with
Symbolic Math Toolbox

Neural-ODE models with
System Identification Toolbox
and Deep Learning Toolbox

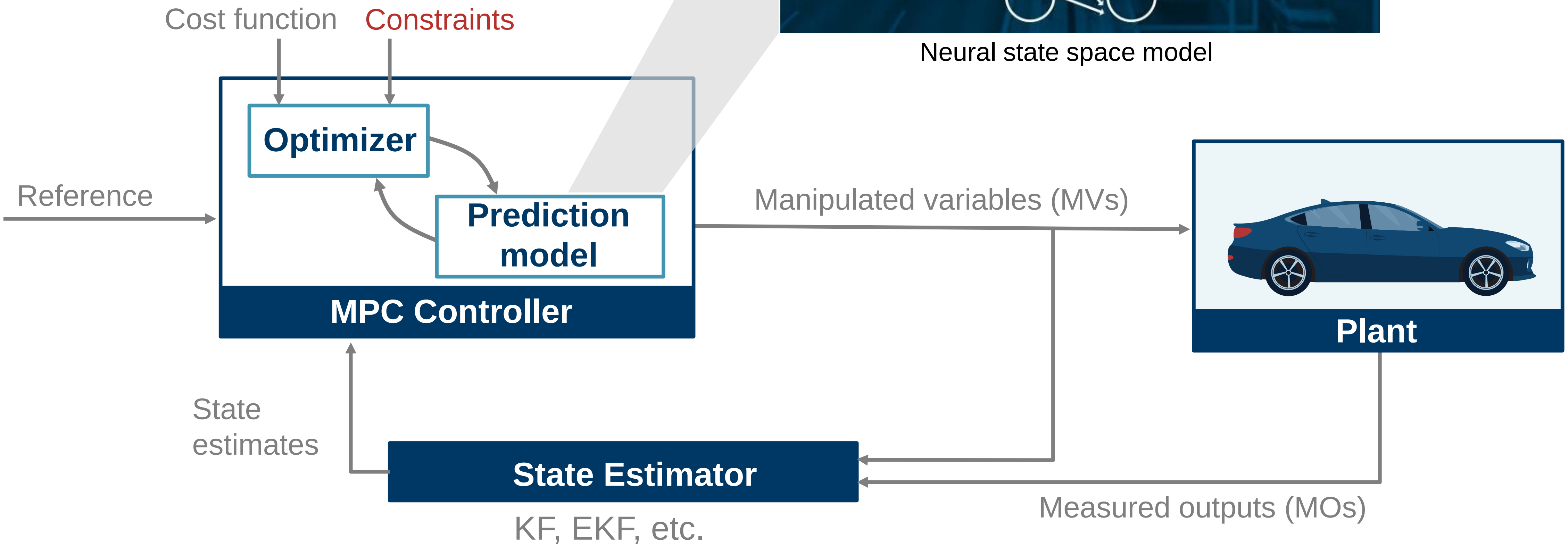
MPC with system identification

Neural state-space prediction model



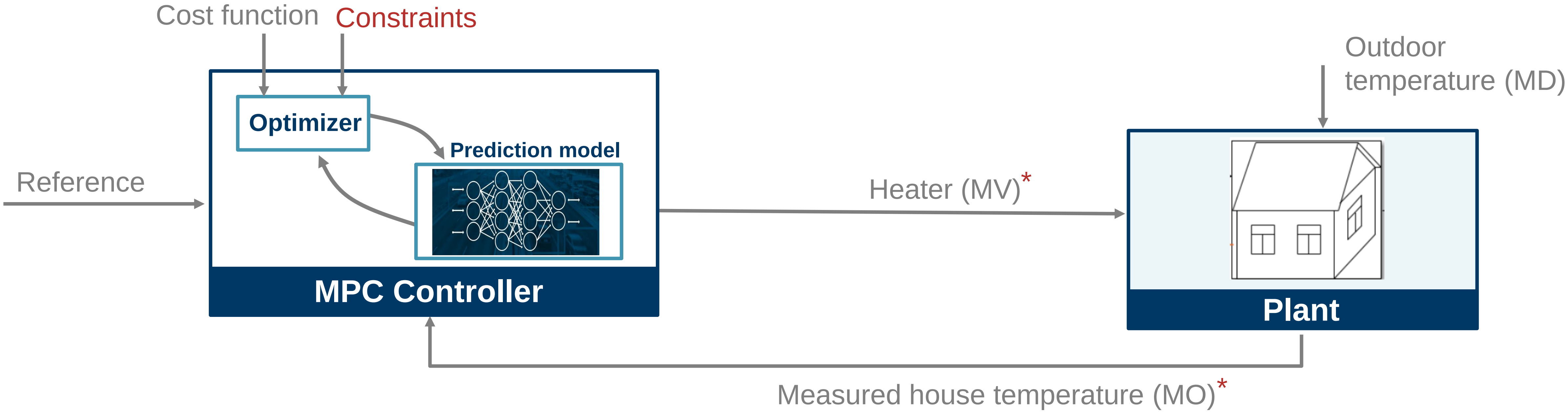
$$\dot{x} = f(x, u)$$
$$y = g(x, u)$$

Neural state space model



结合Neural State Space预测模型的非线性MPC用于控制空调加热系统

- **Plant:** 房屋空调加热系统
- **Goal:** Minimize energy cost and maintain house temperature within a range, e.g. [20 °C, 22 °C]



结合Neural State Space预测模型的非线性MPC用于控制空调加热系统

NMPC with neural state-space prediction model control design workflow

- **训练预测模型**
  - 构建状态空间模型网络框架
  - 配置网络超参数
  - 使用实验数据训练网络
  - 生成状态方程函数
- **使用神经网络状态空间预测模型设计MPC**
  - 指定训练好的预测模型状态方程
  - 指定cost, prediction horizon, constraints, 和 weights
  - 使用自动微分生成雅可比函数
- **Closed-loop 桌面仿真**
- **代码生成和部署**

结合Neural State Space预测模型的非线性MPC用于控制空调加热系统

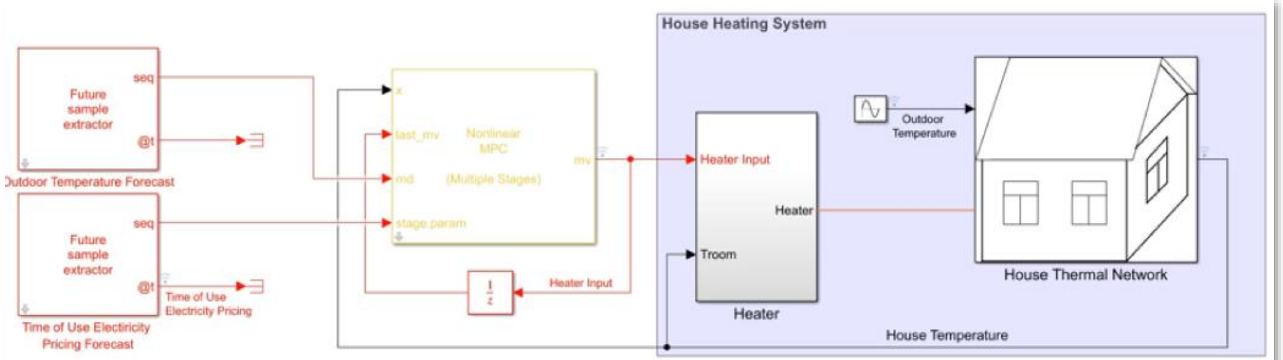
训练预测模型

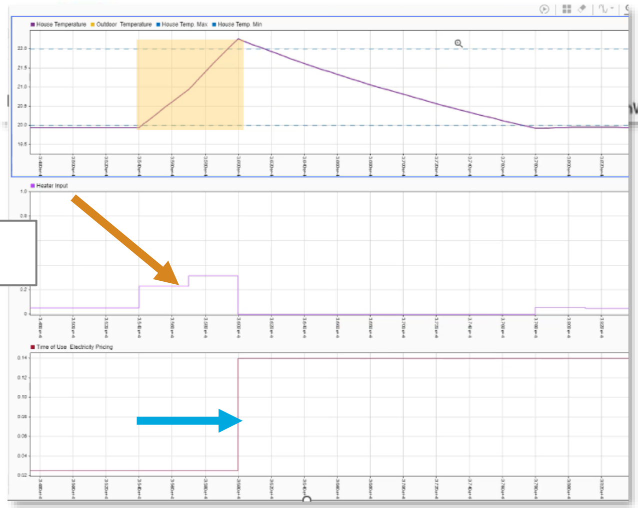
使用神经网络状态空间预测模型设计MPC

Closed-loop 桌面仿真

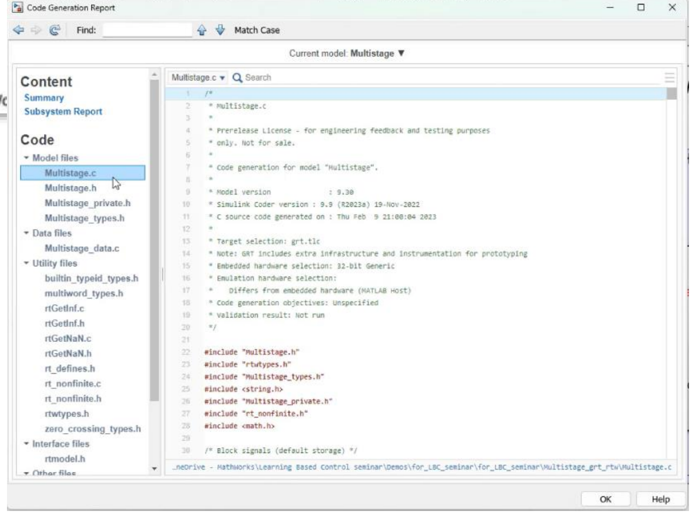
代码生成和部署

- **内置的Simulink block 用于实现非线性MPC**
- **无需深度学习专家知识**轻松训练 neural state space 预测模型
- **Automatic differentiation 和 code generation**





在**电价涨价**之前**提前预热**



```
/*
 * Multistage.c
 *
 * Prerelease License - for engineering feedback and testing purposes
 * only. Not for sale.
 *
 * Code generation for model "Multistage".
 *
 * Model version              : 9.30
 * Simulink Coder version : 9.9 (R2023a) 19-Nov-2022
 * C source code generated on : Thu Feb  9 21:00:04 2023
 *
 * Target selection: grt.tlc
 * Note: GRT includes extra infrastructure and instrumentation for prototyping
 * Embedded hardware selection: 32-bit Generic
 * Emulation hardware selection:
 *    Differs from embedded hardware (MATLAB Host)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "Multistage.h"
#include "rtwtypes.h"
#include "Multistage_types.h"
#include <string.h>
#include "Multistage_private.h"
#include "rt_nonfinite.h"
#include <math.h>

/* Block signals (default storage) */
```

_neDrive - MathWorks\Learning Based Control seminar\Demos\for_LBC_seminar\for_LBC_seminar\Multistage_grt_rtw\Multistage.c

示例链接

示例视频链接

不同类型的最优控制问题

Linear Quadratic Control (LQ)

Optimal control where both the system model and performance objective are represented by linear and quadratic functions, respectively.

$$J(u) = \int_0^{\infty} (x^TQx + u^TRu + 2x^TNu)dt$$

Subject to: $\dot{x} = Ax + Bu$

$u = -Kx$

H-infinity synthesis

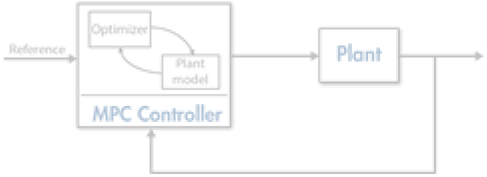
An optimal control tool/technique for designing single-input single-output (SISO) or MIMO feedback controllers to achieve robust performance and stability

Extremum Seeking Control (ESC) 极值搜索

Optimal control method that learns optimal solution through perturbation of control parameters or input signals

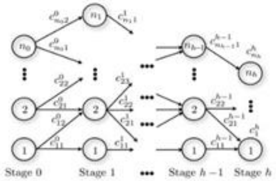
Model Predictive Control 强化学习

Receding horizon approach that solves a constrained optimization problem.



Dynamic Programming 动态规划

Problem-solving technique that involves breaking down a complex problem into smaller subproblems and storing the solutions to these subproblems in memory.



Reinforcement Learning 强化学习

Optimal control technique that trains an ‘agent’ through trial & error interactions with an environment

HEV Control – 能量管理

- 函数解：求解对执行器(发动机、电机)的瞬时扭矩(或功率)指令

$$T_{demand} = T_{eng} + T_{mot} \ ???$$

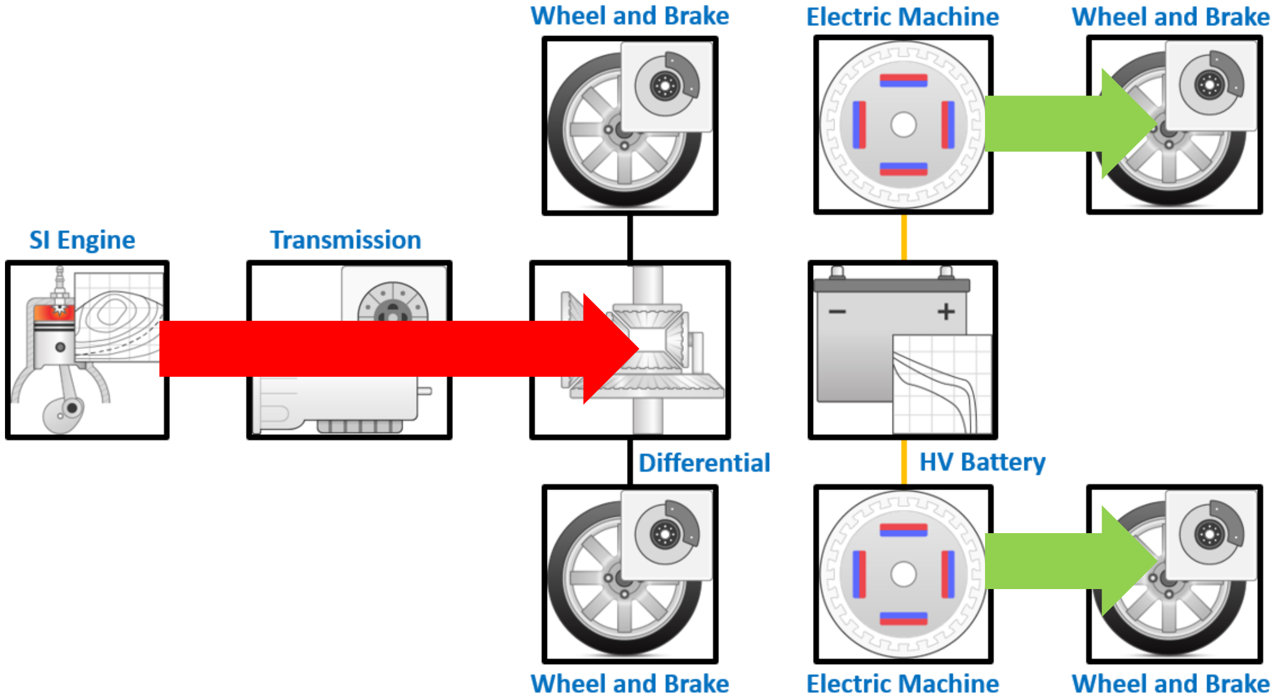
- 满足约束:

$$\tau_{min}(\omega) \le \tau_{act} \le \tau_{max}(\omega)$$

$$P_{chg}(SOC) \le P_{batt} \le P_{dischg}(SOC)$$

$$I_{chg}(SOC) \le I_{batt} \le I_{dischg}(SOC)$$

$$SOC_{min} \le SOC \le SOC_{max}$$



- 目标：尽量减少能源消耗，保持驾驶性能

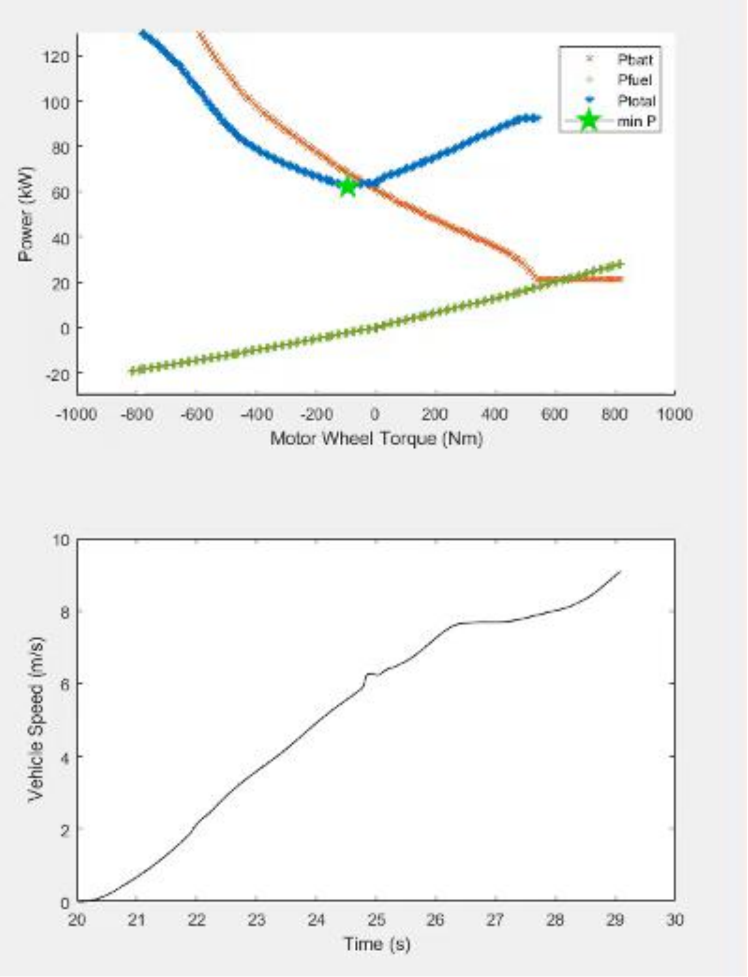
动态规划和庞特里亚金极小值原理

Equivalent Consumption Minimization Strategy (ECMS)

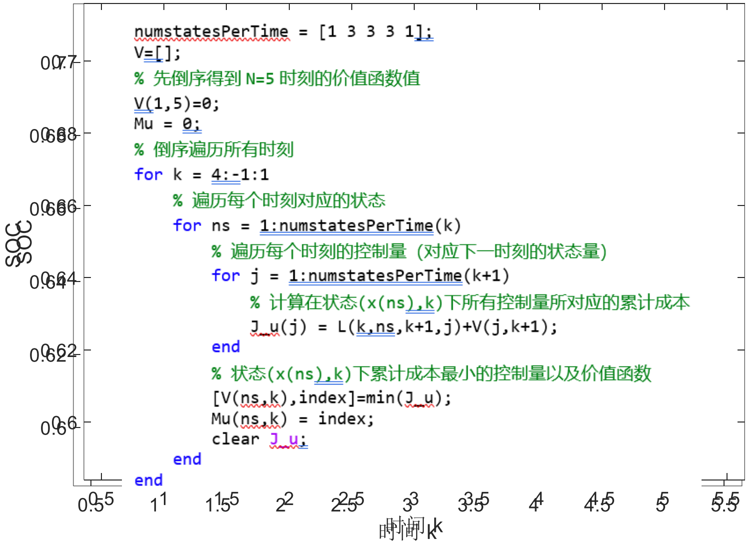
- 将最优控制问
  时优化求解
- 如果驱动周期
  优的解决方案
- 对于未知的驾
  进行增强

$\min P_{equivalent}(t)$

where s(



动态规划



```
numstatesPerTime = [1 3 3 3 1];
V=[];
% 先倒序得到 N=5 时刻的价值函数值
V(1,5)=0;
Mu = 0;
% 倒序遍历所有时刻
for k = 4:-1:1
    % 遍历每个时刻对应的状态
    for ns = 1:numstatesPerTime(k)
        % 遍历每个时刻的控制量 (对应下一时刻的状态量)
        for j = 1:numstatesPerTime(k+1)
            % 计算在状态(x(ns),k)下所有控制量所对应的累计成本
            J_u(j) = L(k,ns,k+1,j)+V(j,k+1);
        end
        % 状态(x(ns),k)下累计成本最小的控制量以及价值函数
        [V(ns,k),index]=min(J_u);
        Mu(ns,k) = index;
        clear J_u;
    end
end
```

$$x_{k+1} = f_k(x_k, u_k)$$

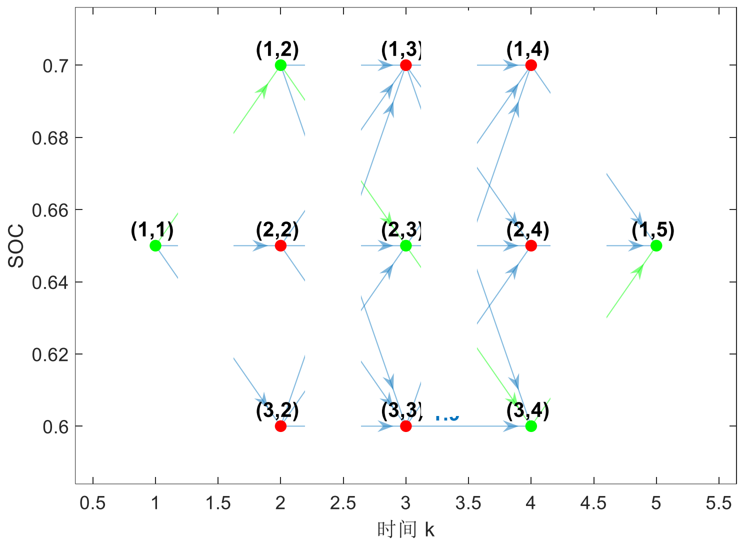
$$V(x_N, N) = h(x_N, N)$$

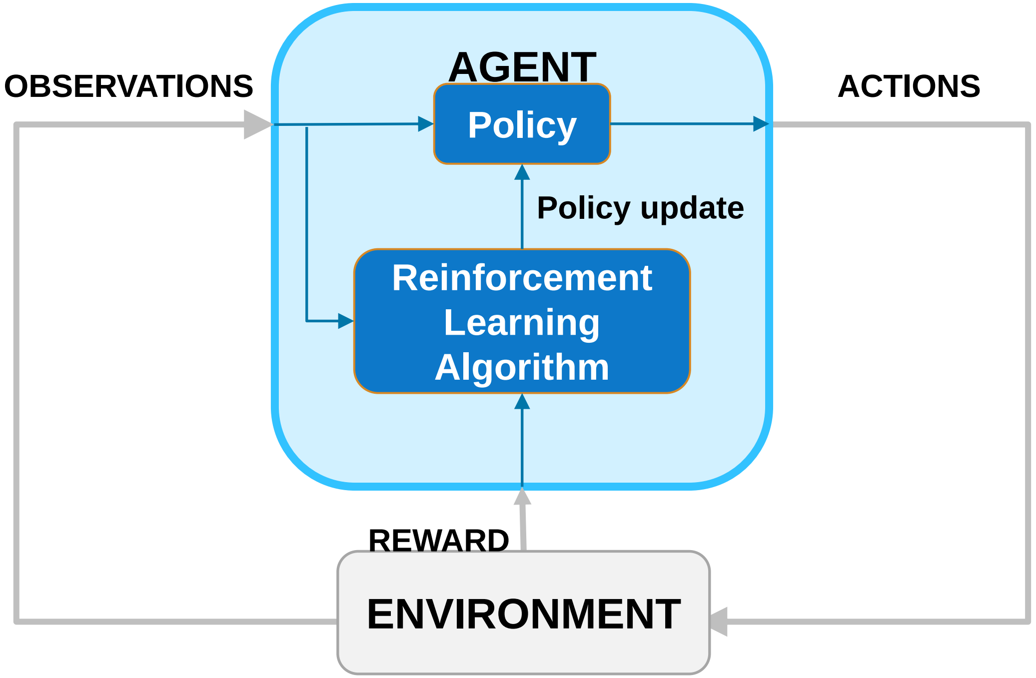
$$V(x_k, k) = \min_{u_k \in U_k(x_k)} \{L_k(x_k, u_k) + V(f_k(x_k, u_k), k+1)\}$$

$$k = N-1, \ldots, 1$$

倒推：先计算k=4时刻的各个不同SOC(状态)下的值函数$V(SOC_k, k|k=4)$. 然后继续倒推，计算任意k时刻，遍历所有离散化的状态和离散化的控制组合，可以得到每个时刻各个状态(x_k, k)对应的值函数$V(x_k, k)$和各个状态下最优的控制量矩阵$Mu(x_k, k)$

什么是强化学习




OBSERVATIONS

AGENT

Policy

ACTIONS

Policy update

Reinforcement Learning Algorithm

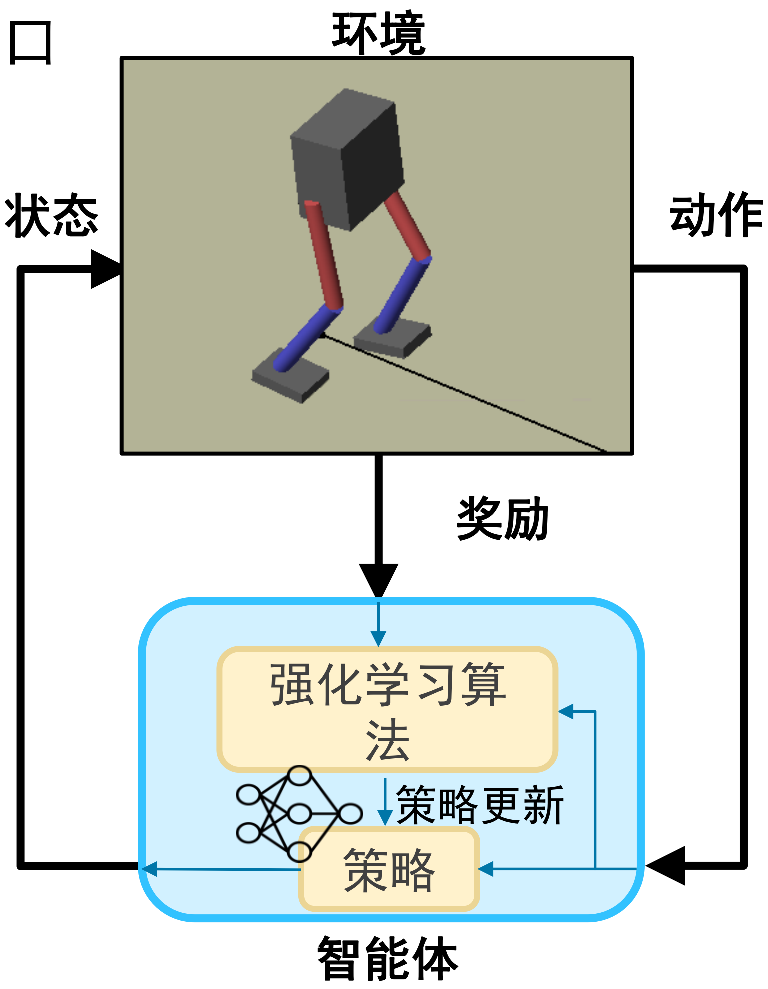
REWARD

ENVIRONMENT

- 适用于通常需要启发式的**序列决策**问题
- 一种机器学习技术，**通过与环境的试错交互来训练“智能体”**

Reinforcement Learning Toolbox

- MATLAB函数/ Simulink模型与环境的接口
  - 强化学习用“RL Agent”模块
- 创建智能体的网络
- 智能体
  - Q-Learning
  - DQN / Double DQN
  - SARSA
  - REINFORCE*1
  - DDPG
  - A2C*1,*2
  - 并行学习(GORILA / A3C*1,*2)
  - PPO
- 策略发布



问题构建

创建环境

定义奖励

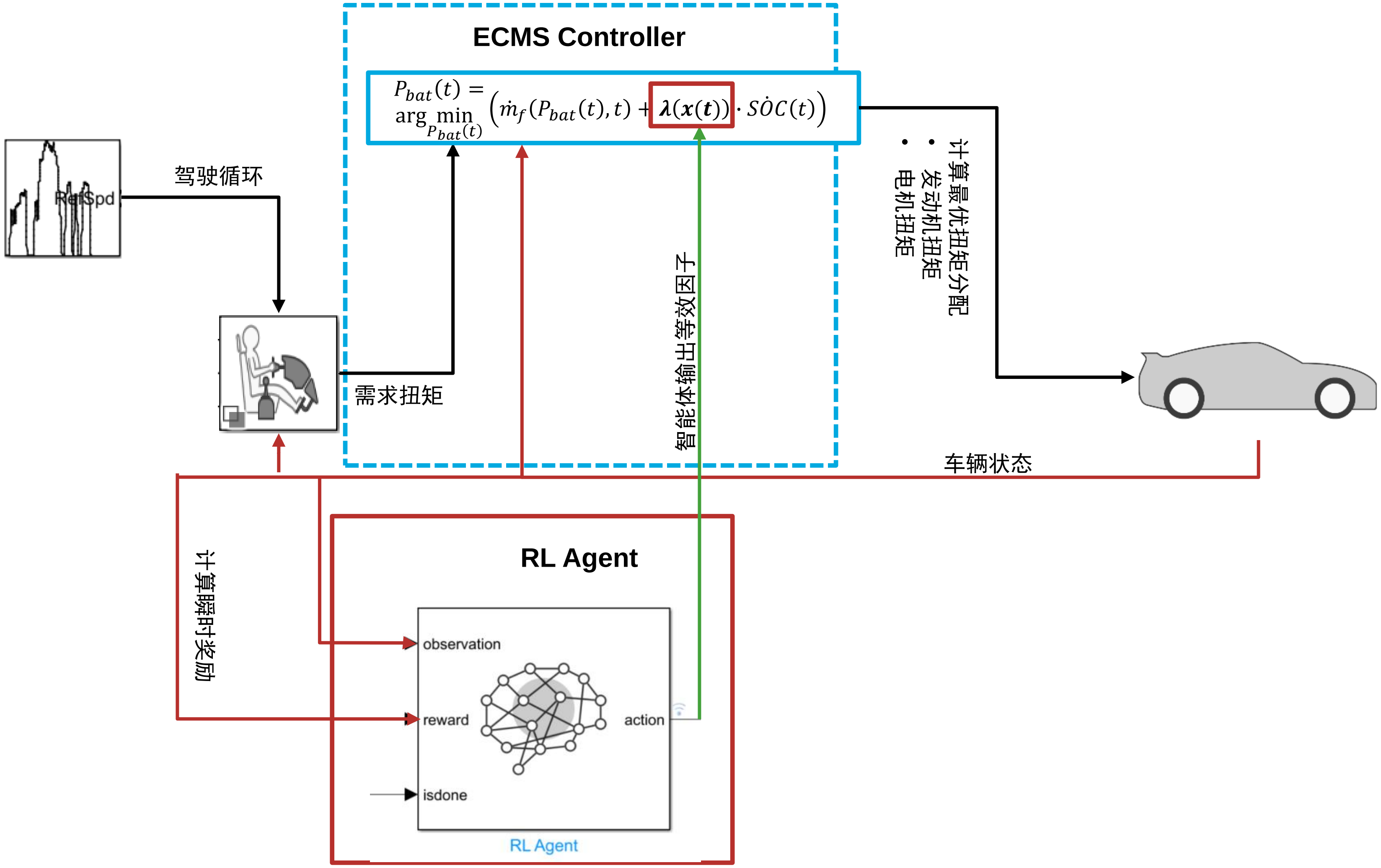
构建策略网络

训练

结果验证

策略部署

Reinforcement Learning 结合 ECMS算法用于标定等效燃油因子λ



ECMS Controller

$$P_{bat}(t) = \arg\min_{P_{bat}(t)} \left(\dot{m}_f(P_{bat}(t),t) + \boxed{\boldsymbol{\lambda}(\boldsymbol{x}(\boldsymbol{t}))} \cdot S\dot{O}C(t)\right)$$

RefSpd

驾驶循环

需求扭矩

智能体输出等效因子

计算最优扭矩分配
- 发动机扭矩
- 电机扭矩

车辆状态

计算瞬时奖励

RL Agent

observation

reward

action

isdone

RL Agent

利用Simulink的强化学习流程(DDPG的例子)

控制系统建模

- 环境-被控对象
- 奖励-奖励函数
- 智能体

环境(被控对象)接口创建

输入输出数据规范

信号/连续、离散/上下限等

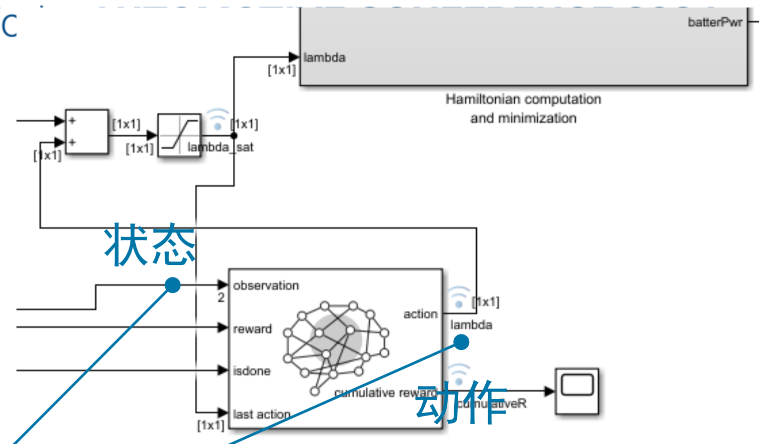
智能体设计

- Critic 网络
- Actor 网络
- DDPG 智能体 **rlDDPGAgent**

训练智能体

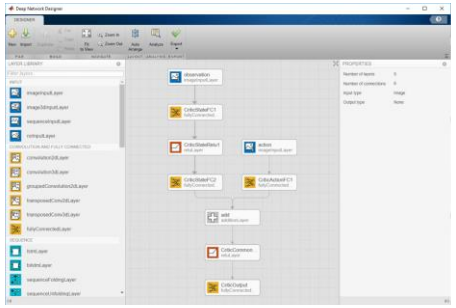
train 学习状态监视器

控制性能评价



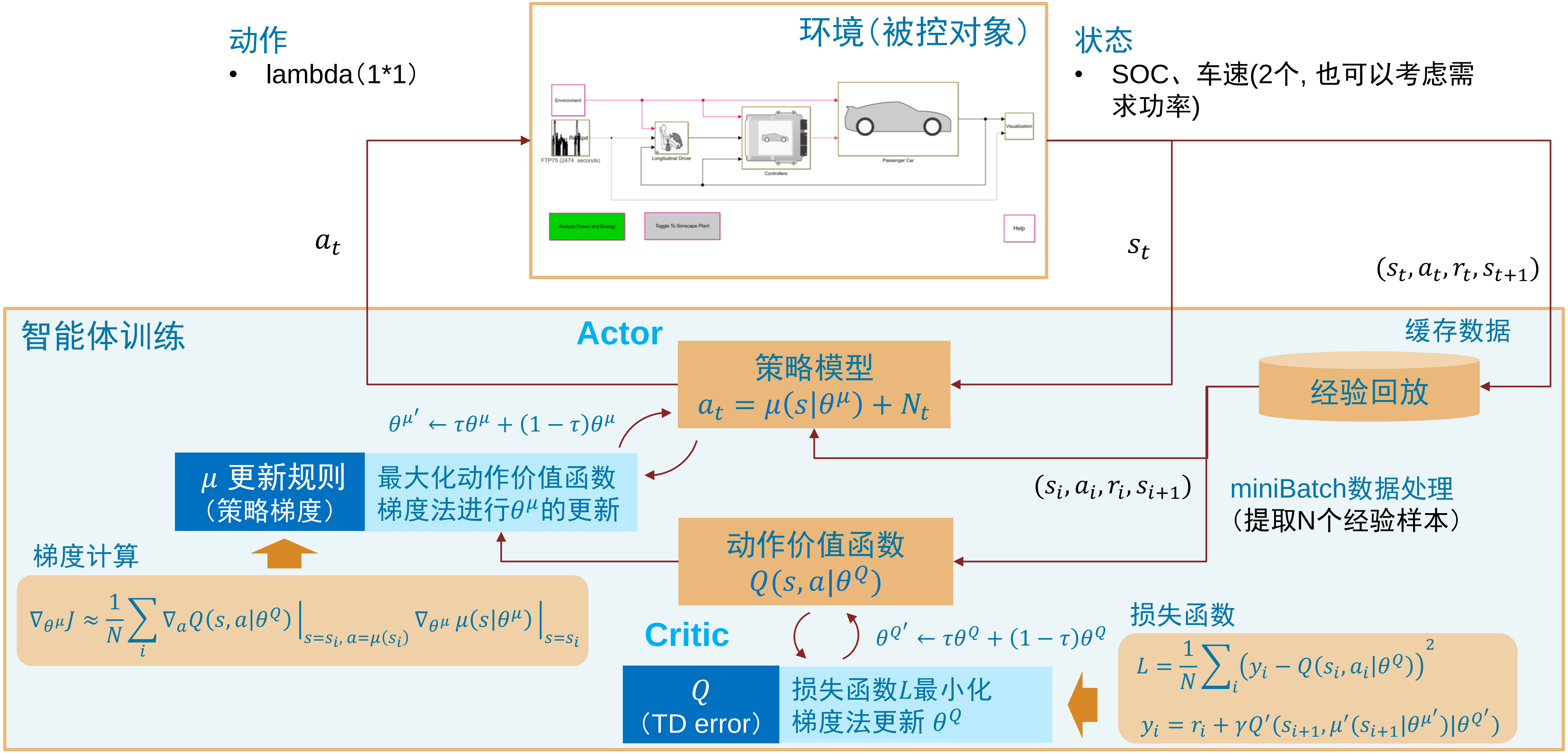
对象

RL Agent
智能体



Deep Network Designer
(Deep Learning Toolbox)

利用Simulink进行强化学习流程(DDPG 智能体示例)



动作
- lambda（1*1）

环境（被控对象）

状态
- SOC、车速(2个，也可以考虑需求功率)

a_t

s_t

(s_t, a_t, r_t, s_{t+1})

智能体训练

Actor

策略模型
$a_t = \mu(s|\theta^\mu) + N_t$

缓存数据

经验回放

$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^\mu$

μ 更新规则（策略梯度）

最大化动作价值函数梯度法进行θ^μ的更新

(s_i, a_i, r_i, s_{i+1})

miniBatch数据处理（提取N个经验样本）

梯度计算

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)\Big|_{s=s_i,\, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)\Big|_{s=s_i}$$

动作价值函数
$Q(s,a|\theta^Q)$

Critic

$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^Q$

损失函数

$$L = \frac{1}{N}\sum_i \left(y_i - Q(s_i, a_i|\theta^Q)\right)^2$$

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$$

Q（TD error）

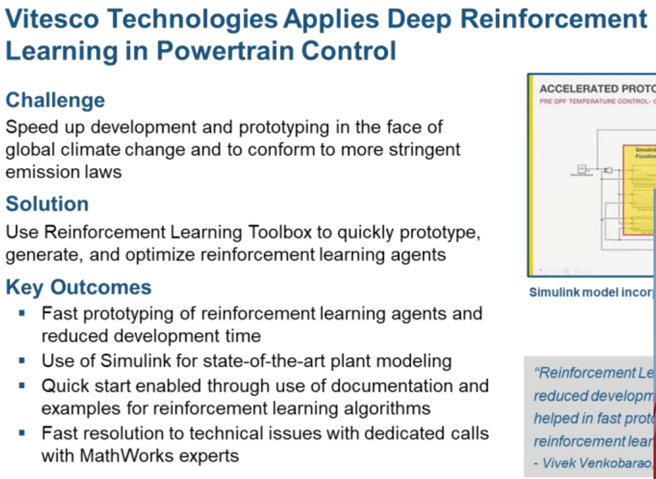
损失函数L最小化梯度法更新 θ^Q

使用开箱即用的算法
自定义算法
开发新算法

Overview of Reinforcement Learning Toolbox Features

	2019	2020	2021	2022	2023
Agents and Algorithms	Q, SARSA, DQN, REINFORCE, DDPG, PPO C++/GPU code gen 自定义智能体	A2C, A3C, TD3, SAC Modular objects for custom training loop	TRPO	MBPO Prioritized and Hindsight replay memory	
Environments	Create MATLAB and Simulink environments		Automatic reward generation		
Training	Train built-in agents	Multi-agent training	Reinforcement Learning Designer app	Parameter tuning Enhanced data logging and visualization	Learn from data Evolution strategy

Reinforcement Learning Toolbox用于解决一系列决策问题和应用



Vitesco Technologies Applies Deep Reinforcement Learning in Powertrain Control

Challenge

Speed up development and prototyping in the face of global climate change and to conform to more stringent emission laws

Solution

Use Reinforcement Learning Toolbox to quickly prototype, generate, and optimize reinforcement learning agents

Key Outcomes

- Fast prototyping of reinforcement learning agents and reduced development time
- Use of Simulink for state-of-the-art plant modeling
- Quick start enabled through use of documentation and examples for reinforcement learning algorithms
- Fast resolution to technical issues with dedicated calls with MathWorks experts

"Reinforcement Le
reduced developm
helped in fast prot
reinforcement lear
- Vivek Venkobarao

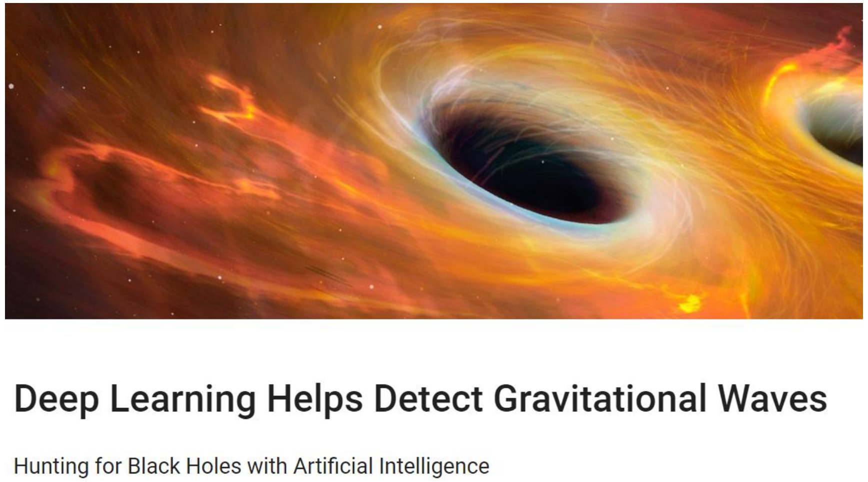
RL replacing classical controls
(Link to customer presentation)

A perspective on deploying Machine Learning to augment classic control design

Ali Borhan
Manager – Cummins R&T

November 5, 2020

RL augmenting classical controls
(Link to customer presentation)

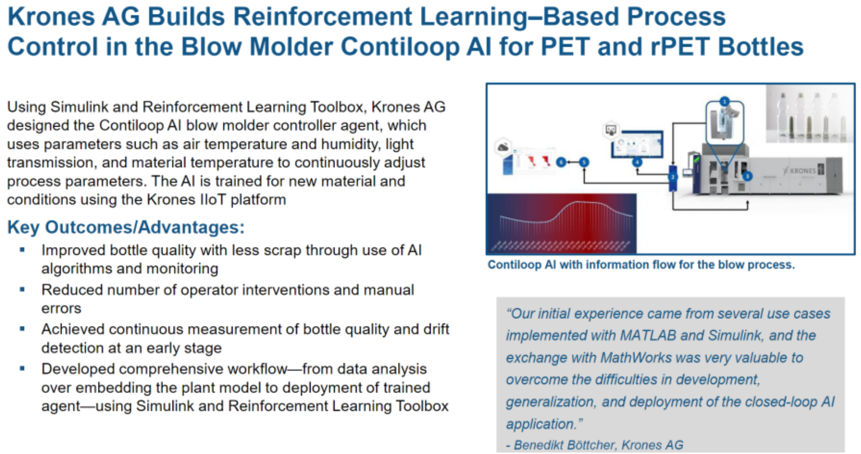


Deep Learning Helps Detect Gravitational Waves

Hunting for Black Holes with Artificial Intelligence

RL for complex nonlinear control
(Link to article)

Reinforcement Learning Toolbox 用于解决一系列决策问题和应用



Krones AG Builds Reinforcement Learning–Based Process Control in the Blow Molder Contiloop AI for PET and rPET Bottles

Using Simulink and Reinforcement Learning Toolbox, Krones AG designed the Contiloop AI blow molder controller agent, which uses parameters such as air temperature and humidity, light transmission, and material temperature to continuously adjust process parameters. The AI is trained for new material and conditions using the Krones IIoT platform

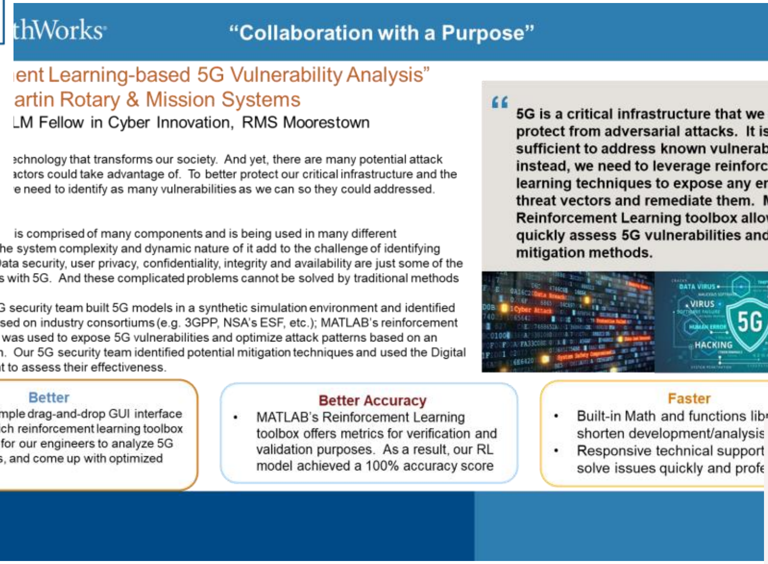
Key Outcomes/Advantages:

- Improved bottle quality with less scrap through use of AI algorithms and monitoring
- Reduced number of operator interventions and manual errors
- Achieved continuous measurement of bottle quality and drift detection at an early stage
- Developed comprehensive workflow—from data analysis over embedding the plant model to deployment of trained agent—using Simulink and Reinforcement Learning Toolbox

Contiloop AI with information flow for the blow process.

> *"Our initial experience came from several use cases implemented with MATLAB and Simulink, and the exchange with MathWorks was very valuable to overcome the difficulties in development, generalization, and deployment of the closed-loop AI application."*
>
> *- Benedikt Böttcher, Krones AG*

RL blow molder control
(Link to customer story)



thWorks "Collaboration with a Purpose"

ent Learning-based 5G Vulnerability Analysis"
artin Rotary & Mission Systems
LM Fellow in Cyber Innovation, RMS Moorestown

echnology that transforms our society. And yet, there are many potential attack
actors could take advantage of. To better protect our critical infrastructure and the
e need to identify as many vulnerabilities as we can so they could addressed.

is comprised of many components and is being used in many different environments. The system complexity and dynamic nature of it add to the challenge of identifying vulnerabilities. Data security, user privacy, confidentiality, integrity and availability are just some of the obvious concerns with 5G. And these complicated problems cannot be solved by traditional methods

Solution: Our 5G security team built 5G models in a synthetic simulation environment and identified threat vectors based on industry consortiums (e.g. 3GPP, NSA's ESF, etc.); MATLAB's reinforcement learning tool box was used to expose 5G vulnerabilities and optimize attack patterns based on an objective function. Our 5G security team identified potential mitigation techniques and used the Digital Twin environment to assess their effectiveness.

> **"5G is a critical infrastructure that we must protect from adversarial attacks. It is not sufficient to address known vulnerabilities; instead, we need to leverage reinforcement learning techniques to expose any emerging threat vectors and remediate them. MATLAB's Reinforcement Learning toolbox allows us to quickly assess 5G vulnerabilities and identify mitigation methods."**

Better

- MATLAB's simple drag-and-drop GUI interface and feature-rich reinforcement learning toolbox made it easy for our engineers to analyze 5G vulnerabilities, and come up with optimized solutions.

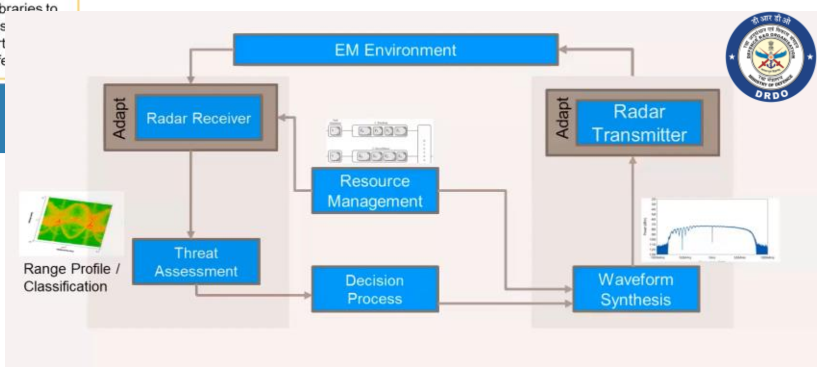
Better Accuracy

- MATLAB's Reinforcement Learning toolbox offers metrics for verification and validation purposes. As a result, our RL model achieved a 100% accuracy score

Faster

- Built-in Math and functions libraries to shorten development/analysis
- Responsive technical support
solve issues quickly and profe

RL in **cybersecurity**
(Link to customer presentation)

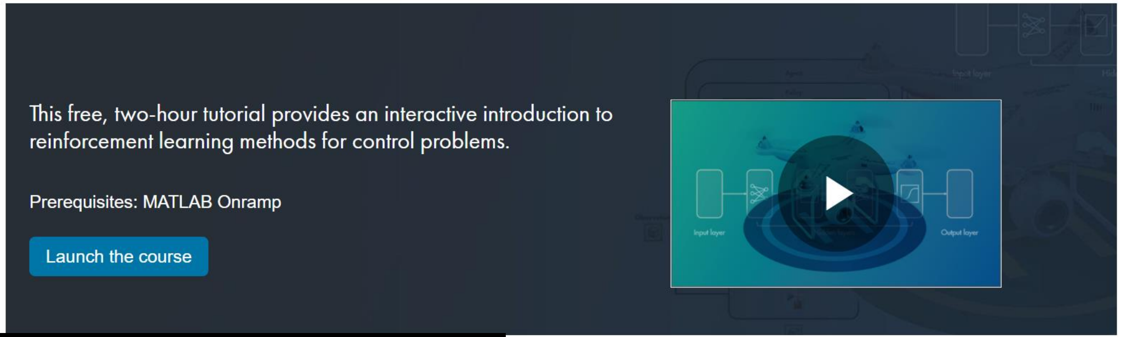


RL for **radar** design
(Link to customer presentation)

相关资源

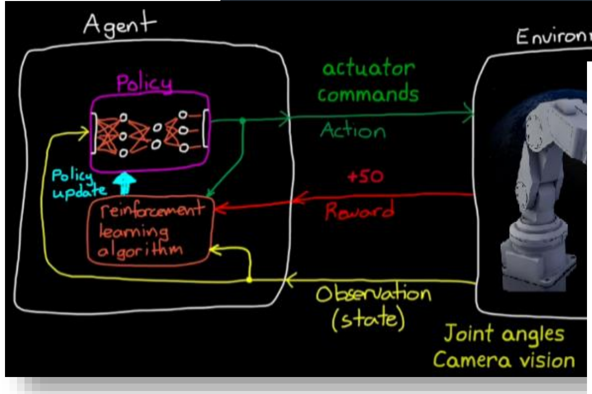
- Reinforcement Learning Toolbox product page
- Reinforcement Learning Onramp
- Instructor-led training
- Demo videos
- Documentation and examples written for engineers and domain experts
- Tech Talk video series on Reinforcement Learning concepts for engineers
- Reinforcement Learning e-books
- 微信文章
  - AI与控制（二）：从优化到最优控制，从动态规划到强化学习

Reinforcement Learning Onramp



This free, two-hour tutorial provides an interactive introduction to reinforcement learning methods for control problems.

Prerequisites: MATLAB Onramp



AI与控制（二）从优化到最优控制，从动态规划到强化学习

Original 刘海伟 MATLAB 2023-11-15 18:03 Posted on 上海

收录于合集

#MATLAB 数据科学与人工智能 117个 >

优化问题，尤其静态优化问题，在控制系统设计中随处可见，例如基于燃油经济性和驾驶体验的多目标优化的汽车发动机 MAP 标定，基于性能指标优化的飞行器结构设计参数优化，以实验数据与模型输出匹配为目标的电池 RC 等效电路模型标定等等，他们都是通过构建目标函数（某个值的性能最大，或者某两个值之间的差距最小），然后调用优化算法实现设计变量寻优。

当设计变量变成一个函数，而这个函数对系统指标的影响又在时间先后上取决于一个动力学约束，这时候我们可能依然可以通过离散的方式将问题变成静态优化问题，当然这类问题也可以通过最优控制理论来实现，利用庞特里亚金极小值原理，动态规划来求解。例如混动车辆的 ECMS 算法，输出电池和发动机的能量分配序列满足电池 SOC 平衡的基础上油耗最低。

随着 AI 的引入，即使我们对于系统或模型一无所知，我们又可以通过试错的方式来获得一个长期奖励较优的控制器，用于处理序列决策问题，例如自动驾驶车辆或智能机器人的控制器控制序列，也就是强化学习的思路。本文接下来通过MATLAB示例来简单介绍这些概念的思想。

◆ ◆ ◆ ◆

优化问题

对于一个普通的静态优化问题，可以描述为求解最优变量 x 使得 f(x) 最小[1]，

$$\min f(x)$$

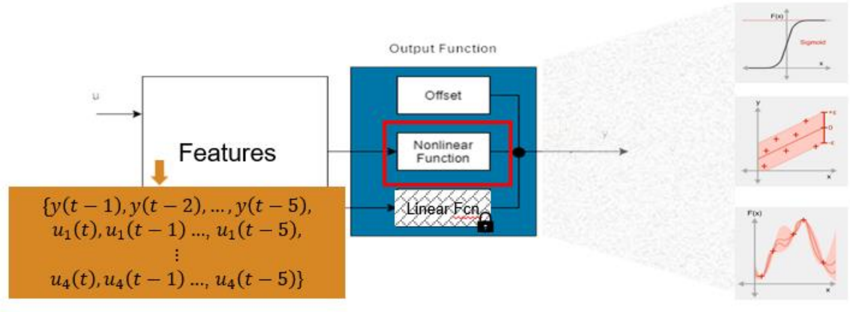
总结

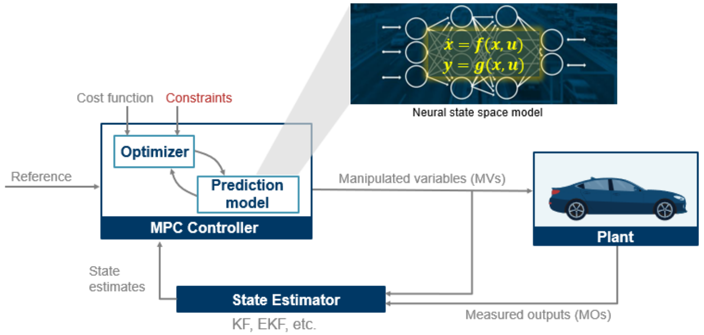
数据驱动的被控对象建模用于控制系统设计

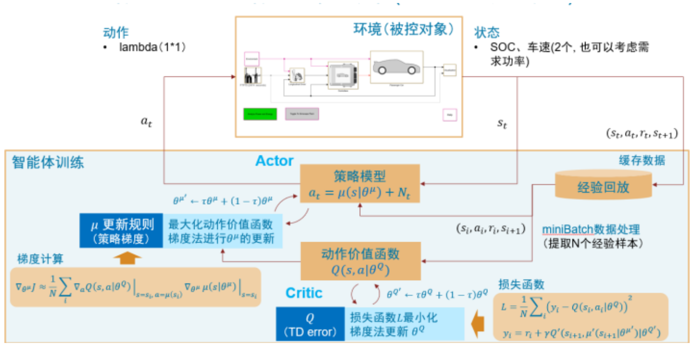
- 结合AI的一些经典的系统辨识（NLARX+AI, NLHW+AI）
- 基于AI的系统辨识算法（神经网络状态空间等等）

最优控制+AI

- 模型预测控制+AI
- 强化学习







Q & A

2024 MathWorks
中国汽车年会

Thank you

MathWorks®