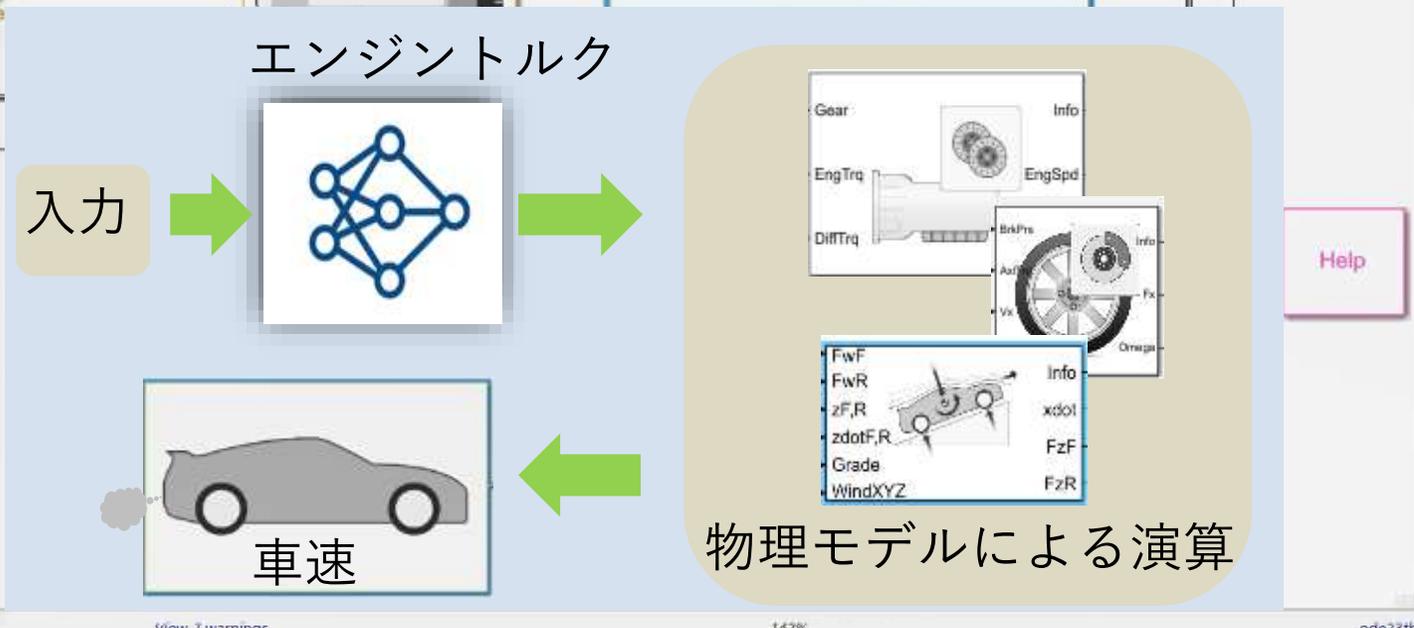
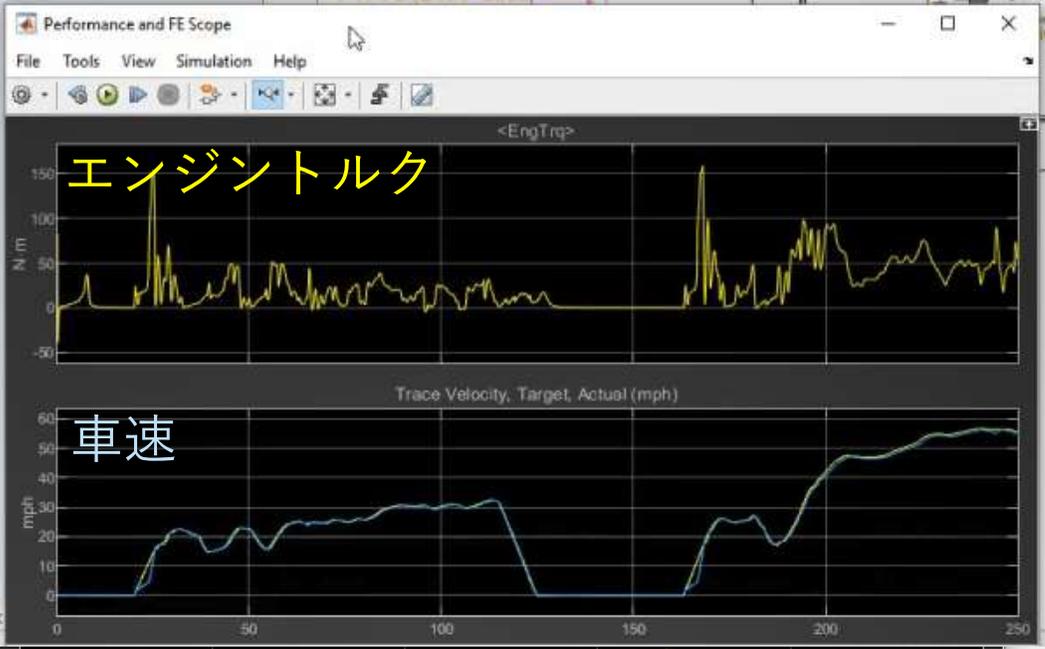
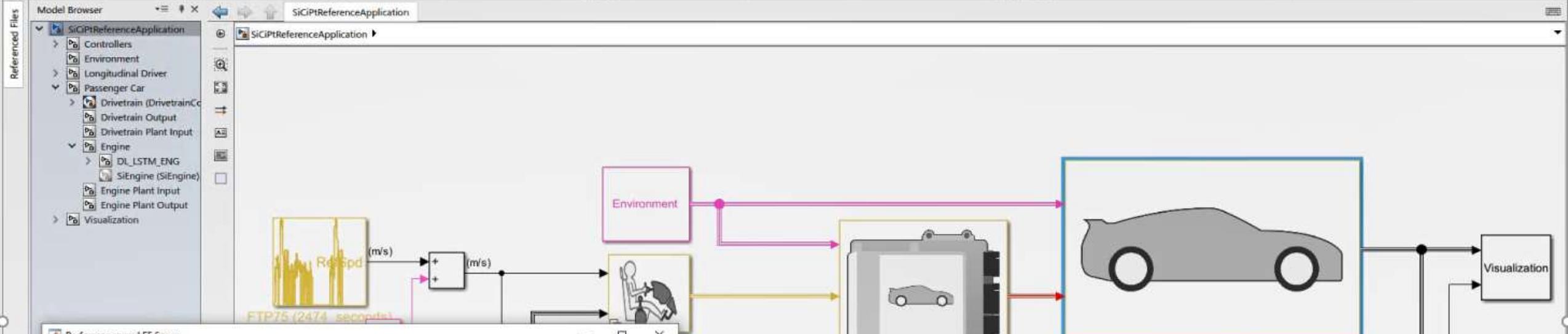


MATLAB EXPO 2021

MBDでのAI活用法：シミュレーションと実装

アプリケーションエンジニア
福本 拓司





AIは私たちの生活に欠かせないものに



音声認識



顔認識



自動運転

機械学習

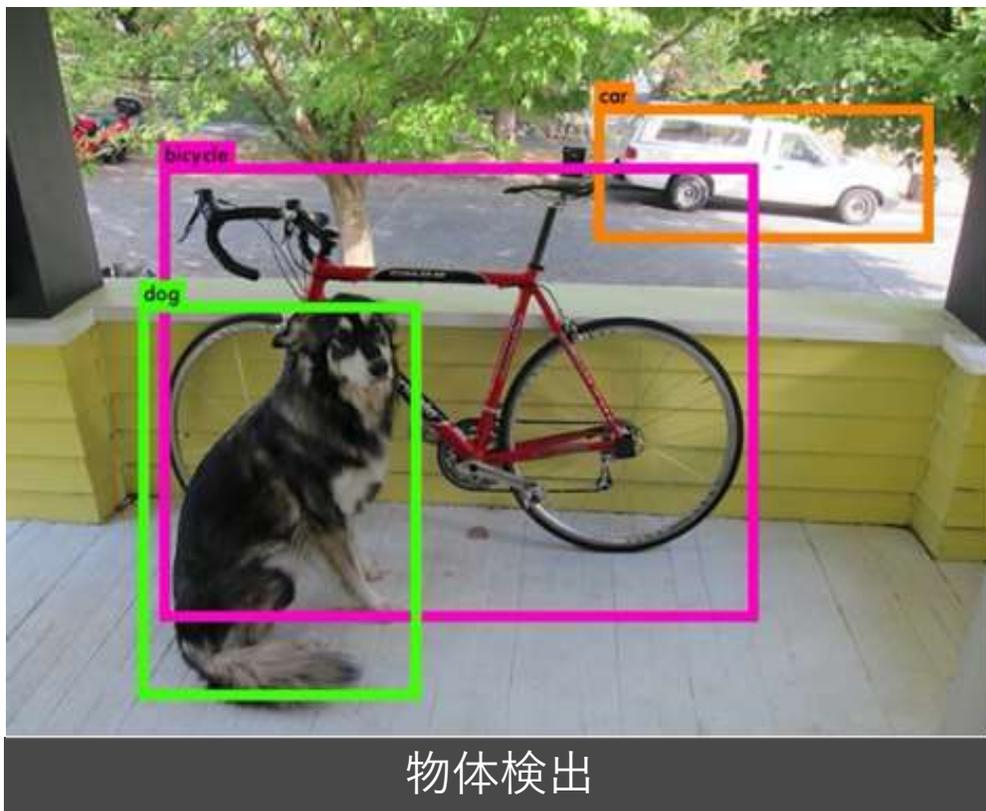
本セッション
での「AI」 =

ディープラーニング

強化学習

エンジニアリング分野でのAI活用

汎用



エンジニアリング分野



ディープラーニング検出

産業界で利用されるMATLABでのAI - 異常検出-

成功事例の
ダウンロード



武蔵精密工業、自動車部品の外観検査にディープラーニングを活用

武蔵精密工業株式会社

武蔵精密工業株式会社は、自動車の部品検査にディープラーニングを活用し、検査効率を向上させました。検査対象は130万個/月、本プロジェクトから前処理、学習、推論、結果生成機能と連携させました。

- 画像データ
- Class
- ラベル
- MATLAB



エアバス、AIとディープラーニングを欠陥検出を自動化

Nicolas Castet 氏, エアバス (Airbus)

航空機のパイプの欠陥を自動的に検出するために、ロバストなエンドツーエンドモデルをどのように構築すべきか。これはエアバスにとって大きな課題でした。エアバスは、MATLAB®を使用して、ディープラーニングモデルを迅速に構築し、開発しました。

エアバスはMATLABを採用し、MathWorksコンサルティングチームと協力により、3つの主要なステップに対処しました。最初のステップは、セグメンテーションなどのアプローチのために、ディープラーニングから構築し学習する統合ツールと、ビデオにラベルを付けるための視覚環境を用意することでした。MATLABのディープラーニングモデルにより、パイプ上のワイヤの位置を使用し、業界標準で必要とされる欠陥を測定しました。次に、欠陥の分析をリアルタイムで表示できる必要があり、最後のステップは、コーディングを必要とせずにMATLABコードを自動コードに変換し、組み込みシステムに直接展開することでした。

MATLAB®利用のメリット:

- 統合ツールを使用して、ディープラーニングモデルを設計、学習、展開
- 対話的なプロトタイピングとテストを非常に短時間で実行
- MATLABコードをCUDAコードに直接変換

» ディープラーニングのモデル化について学ぶ



八千代エンジニアリング、ダム・橋梁損傷検出にディープラーニングを適用

安野 貴人氏, 八千代エンジニアリング

八千代エンジニアリングは、ダム・橋梁の損傷検出にディープラーニングを適用しています。

取り組みでは、ディープラーニングを適用することで、ネットワークの適用は難しく、指しました。ムは3000以上の効率化が実現しています。

- セマンティック
- 多数の
- 複数GPU
- 任意のG

» 深層学習



関西電力、配管溶接部の損傷評価にディープラーニングを適用

木津 健一氏, 関西電力株式会社

関西電力は、高クロム鋼配管溶接部のクリープ損傷評価の実施に、ディープラーニングを適用しました。従来の外表面を視察する、非破壊検査では発見が困難だった肉厚内部での損傷を評価するため、外表面のひずみ分布計測で得られた画像を用いる手法が試みられています。

研究では、試験体のひずみ分布計測画像を使い、ディープラーニングのネットワークである、AlexNetにより抽出した特徴量を、サポートベクターマシンを使って、損傷小と損傷大に分類できることを確認しました。さらに、同様のネットワークにより損傷率を推定する回帰モデルを作成し、損傷率が大きいほど良い精度で推定できることを確認しました。今後、本手法の実機への適用性確認や、様々な材質の配管への応用などを目指しています。

MATLAB®利用のメリット:

- 転移学習をすぐに利用できるサンプルコード
- 学習済みネットワークの利用が容易
- 専用ツールボックスによる、その他の機械学習手法との組み合わせ
- 無料のセミナー、ヘルプドキュメントなど学習方法が充実

» AlexNet を使用した特徴抽出

“

MATLABを用いることで損傷の画像解析が容易に実現できました。機械学習からディープラーニングまで一つのフレーム

“

MATLABを導入して、ディープラーニングを使った開発を簡単に行えました。プログラミングが得意でなくても、無料セミナーとヘルプドキュメントだけで、十分学習することができました。

”

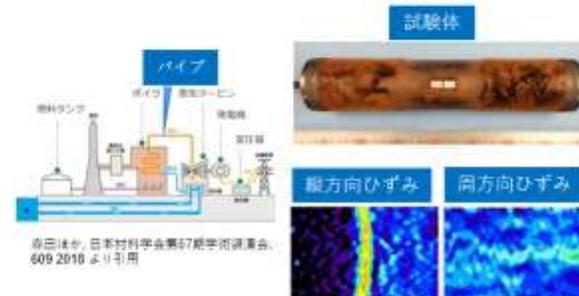


図1 木津 健一氏, 日本材料学会第67期学術講演会, 609 (2016) より引用

MATLAB EXPO 2019 Japan

» 講演資料を見る

産業界で利用されるMATLABでのAI

成功事例の
ダウンロード



立命館大学、CT撮影における放射線被ばくリスク低減にディープラーニング活用

中山良平准教授、立命館大学

コンピューター断層撮影(CTまたはCAT スキャン)は腫瘍生成するため、単なるレントゲン写真よりもはるかに高放射線量で、人体に有害である放射線の被ばく量が多くなります。

そこで立命館大学准教授中山良平氏は超低線量 CT スキャンから通常線量の CT スキャンと同程度の画質の画像を生成するニューラル ネットワーク (CNN) 回路をハイブリッドしたシステムを開発しました。CNNでの回帰は新しい画像生成時間が高速になるメリットがあり、肺領域と肺以外を対象とすることで結果の鮮明度を向上させました。

このシステムでは同程度の診断情報を95% 低減することが可能です。現在は画像管理システム(PACS)との連携の検証中。

MATLAB利用のメリット:

- 128種類のネットワーク構造から選択可能
- 画像の超解像処理とCNN回帰の高速化
- GUIインターフェースを作成し、誰でも簡単にWebアプリに実装可能

» 詳細はこちら

防衛装備庁、遠赤外線画像にGANを適用し、カラー画像を生成

課題
遠赤外線画像の視認性向上

ソリューション
深層学習のネットワークから高精度なカラー画像生成

結果

- 遠赤外線画像にGANを適用し、カラー画像の生成を実現
 - 学習データを作るための遠赤外線カメラと可視カメラの活用
- 自律移動ロボットの認識および制御に応用



ステック井上、Webカメラとディープラーニングで製造作業監視システムを実現

村井 浩一氏、株式会社ステック井上/長崎大学 工学研究科/有限会社秀工社

人手不足が進む近年、製造現場では労働生産性向上を目指したインダストリアルエンジニアリングに基づく改善活動が求められています。作業工程の把握はその第一歩です。作業分析にはストップウォッチでの計測や、ビデオを後から解析する方法が一般的ですが、製造現場での改善を促進するためにはリアルタイムで監視できる必要があります。

“ MATLABを使う事で、アルゴリズム検討から現場への展開までを迅速に行うことが可能になりました ”



宮崎大学、スマート酪農の実証実験にMATLABを活用

ティティ・ズイン氏、宮崎大学工学部システム工学科

宮崎大学のティティ・ズイン教授は、牛の健康状態を監視するために、乳牛の居場所の把握と遠隔監視システムを開発しました。このプロジェクトのために、画像認識アルゴリズムとシステムを開発しました。高精度な耳標の読み取りには、いくつかのDeep Learning Toolbox™は画像の取り込みを容易にするため、試行錯誤に時間をかけることを開発できました。システム実装は、開発者インターフェースをMATLAB Compiler™でWebブラウザで使用できるアプリに展開しました。

MATLAB利用のメリット:

- 牛の餌の検出100%、耳標の認識90%の認識精度の高いアルゴリズム開発と短期間で実現
- ITの専門家がチームにいないでも簡単にWebアプリに実装可能



資生堂、ディープラーニングによる化粧品開発

古田 光輝氏、資 麗氏、中谷 隆一氏、資生堂化粧品開発センター

安心・安全な化粧品を開発するためには、刺激や微生物汚染等を考慮し、必要最小量の防腐剤配合製品を開発することが求められます。そのため、チャレンジングな「保存効力試験」によって防腐力を評価しています。しかし、チャレンジングな試験には多くの種類の細菌、酵母、カビ等を長期間かけて培養し、判定する必要があります。

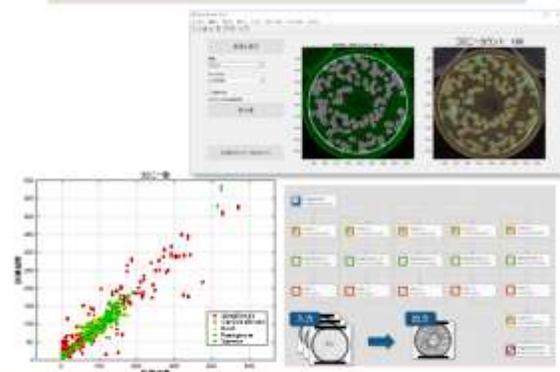
そこで資生堂は、微生物の生育の予測、試験省略可否の予測にAIの技術を利用する研究を行いました。MATLAB®の機械学習のモデルを用いて、微生物の生育予測を行い、菌の生育予測には、MATLABのGUIを作成し、微生物の生育予測のカウントには、MATLABにて

- Deep Learning Toolbox™
- 豊富なデータセット
- App Designer™による直感的なネットワーク構築

» MATLABによるディープラーニング



“ 化粧品開発に必要な微生物評価においてMATLABを活用し、試験の迅速化につながる良好な結果が得られました。 ”



Deep Network Designerによる直感的なネットワーク構築



MathWorksは、ガートナー社の2021年のデータ・サイエンスと機械学習プラットフォーム部門についてのマジック・クアドラントにおいてリーダーの1社と評価されました

MathWorksはリーダー・クアドラントにおいて2年連続でリーダーの1社に位置付けられ、ビジョンの完全性を基に高い評価をされました

この評価は、以下の実行能力を実証するものと考えています

- AIおよびデータサイエンスの経験が限られているチームを強化
- データの準備、AIモデリング、システム設計、量産のための包括的なワークフローの提供
- AIモデルを組み込み機器、AIモデリング、エッジデバイス、エンタープライズシステム、クラウドに実装
- インテグレーションの課題に取り組み、AIドリブンシステムの開発におけるリスクを軽減するためのSimulinkの活用

Gartner Magic Quadrant for Data Science and Machine Learning Platforms, Peter Krensky, Carlie Idoine, Erick Brethenoux, Pieter den Hamer, Farhan Choudhary, Afraz Jaffri, Shubhangi Vashisth, 1st March 2021.

This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner document is available upon request from MathWorks.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner research organization and should not be construed as statements of fact. Gartner disclaims all warranties, express or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

Figure 1: Magic Quadrant for Data Science and Machine Learning Platforms

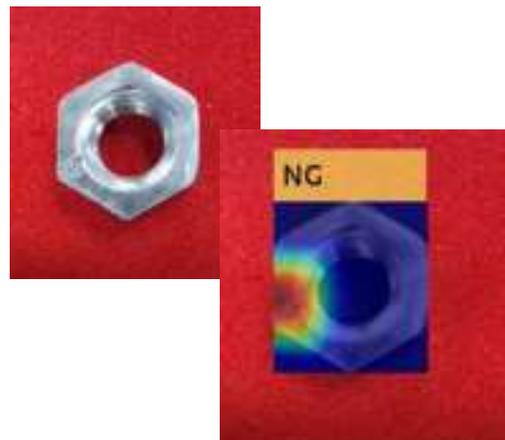
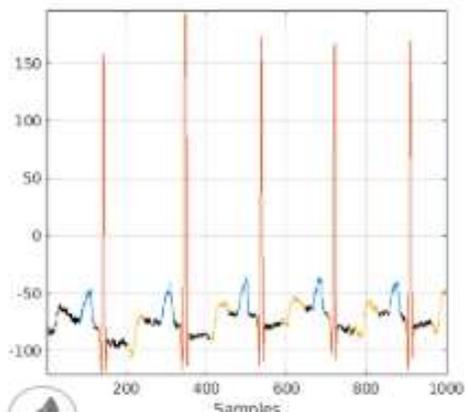


Source: Gartner (March 2021)

AIモデルのシミュレーションでの活用

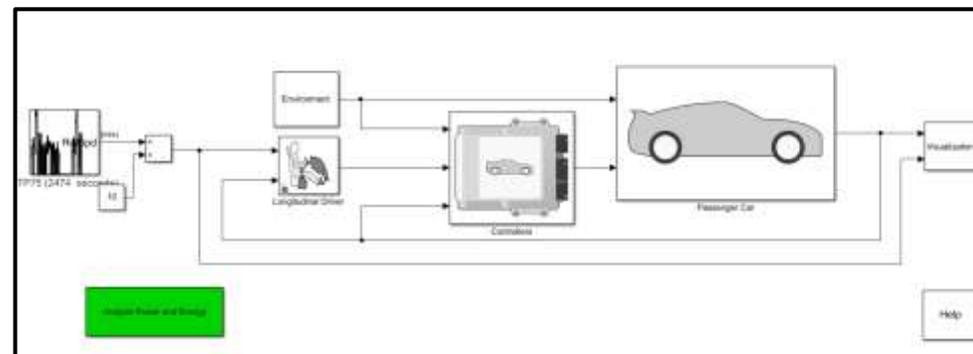
新たな実データに適用

- ・ 外観検査
- ・ 異常検知
- ・ 劣化予測
- ・ ...



シミュレーションに適用

- ・ AIモデルを含むアルゴリズム開発
- ・ プラントをAIモデルで表現
- ・ ...



AIを複雑なシステムの一部で利用する

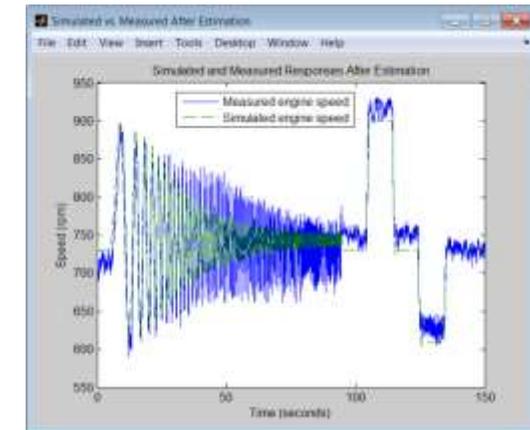
MATLAB/Simulinkによる システムモデリング・シミュレーション



Simulink によるモデルベース デザイン

テスト対象のシステムや物理的なプラントをモデル化し、動的な全体シミュレーション

- 実物/実世界では再現困難/不可能なテストを行える
 - 危険
 - 高コスト
 - 物理的に困難/不可能
- モデルベースデザインの主な適用領域
 - 制御システム
 - 信号処理
 - 通信システム
 - パワエレ
 - メカトロ
 - 自動運転
 - IoT
 - etc...

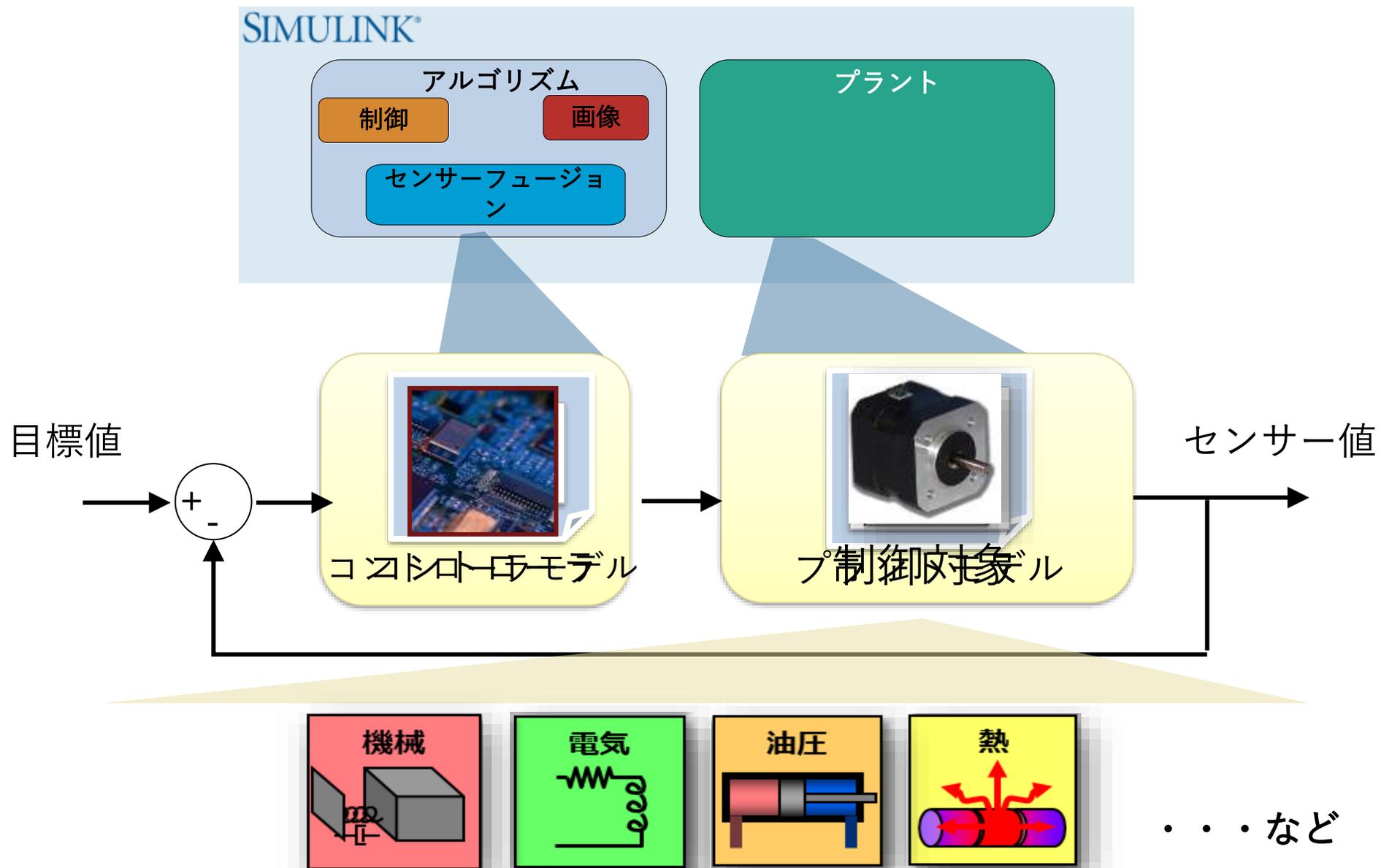


モデリング・シミュレーションの中で、
どのようにAIモデルを活用していただけるかを紹介します。

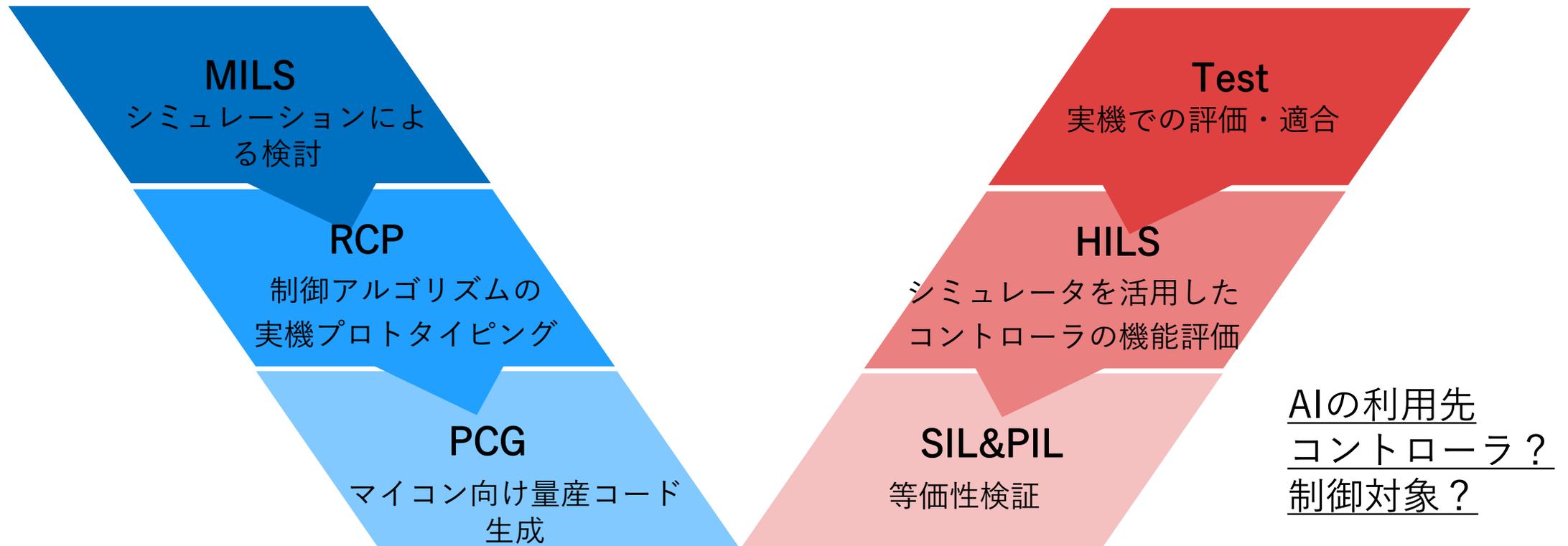
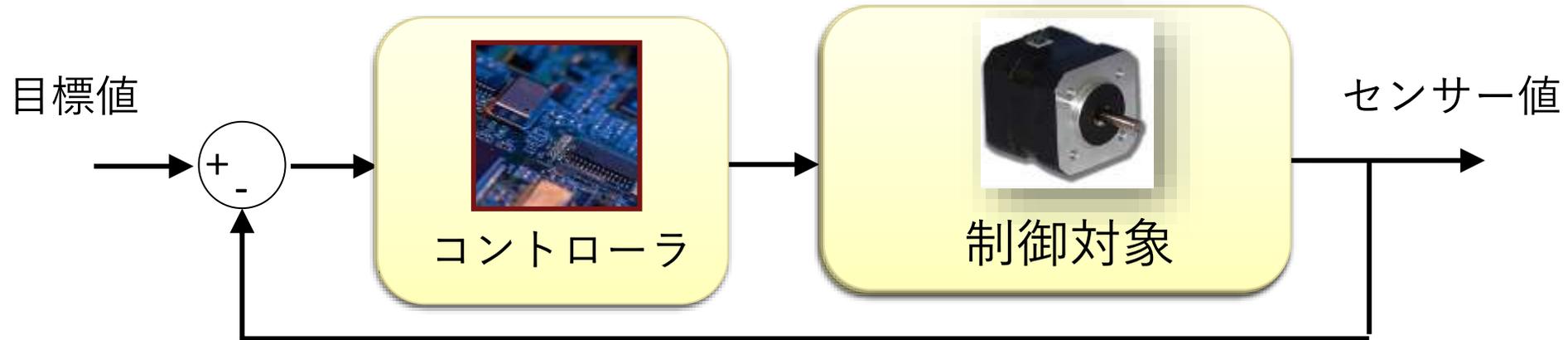
アジェンダ

- MBD開発でのAIモデル利用ケース
- エンジンモデルを題材にしたAI活用の3ステップ
 - AIモデルの構築
 - モデルドリブン+AIの統合シミュレーション
 - RCP/HILSのためのコード生成
- TensorFlow/PyTorchモデルの活用
- まとめ

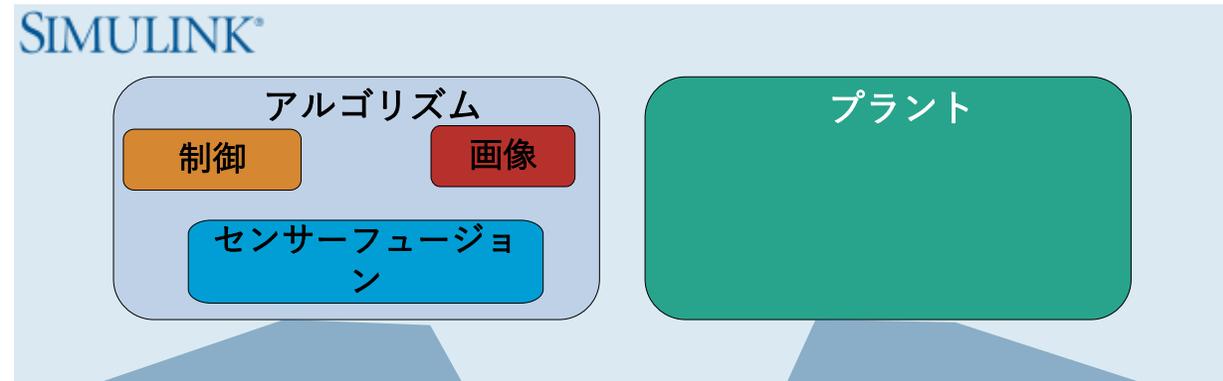
Simulinkでのモデリング



モデルベースデザインの開発フロー



シミュレーションの中でAIモデル活用の動機づけ



アルゴリズム

- 判断（コントローラ）にAIを利用
- ラピッドプロトタイピング(RCP)
- CPU/GPU/ECU/FPGAへの展開

プラントモデル

- 複雑な状態、応答の表現にAIを利用
- 高精度シミュレーション(3D-CAE 等)の高速化
- Hardware In the Loop (HILS)による検証
- コンポーネントの共有

ステップ：
|• 実測/シミュレーションデータでのAIモデル学習
|• システムレベルへの統合シミュレーション
|• Cコード生成

適用ステップはどちらも同じ
→プラントモデルで説明

プラントモデリングアプローチの利点とチャレンジ

モデルドリブンアプローチ

物理(第一原理)モデリング

データドリブンアプローチ

統計・最適化・AI

White-box

Gray-box

Black-box

◆ 利点

- ✓ 対象物の物理的挙動の本質を捉えたモデル構築が可能

◆ チャレンジ

- ✓ モデル導出・構築に対する労力
- ✓ 物理ドメインに関する専門知識が必要

◆ 利点

- ✓ データに対して高精度に適合したモデル構築が可能

◆ チャレンジ

- ✓ 大量の実機計測データが必要
- ✓ 数学モデルのパラメータの多くは物理的意味を持たない

プラントモデリングアプローチの利点とチャレンジ

モデルドリブンアプローチ

物理(第一原理)モデリング

データドリブンアプローチ

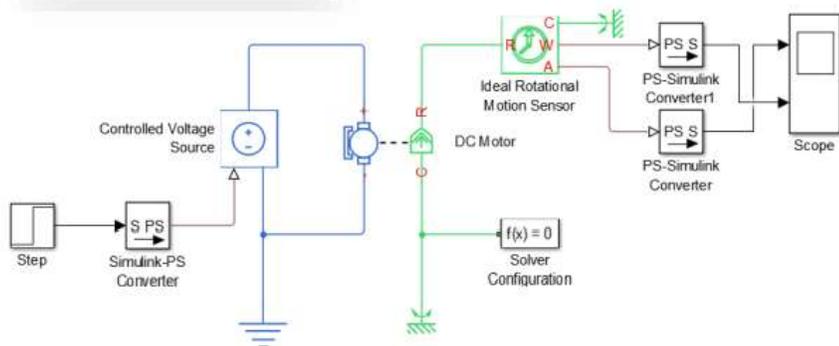
統計・最適化・AI

White-box

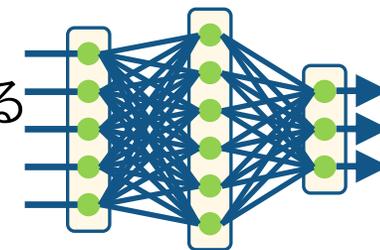
Gray-box

Black-box

基本的な物理式で
挙動を表現できるシステム



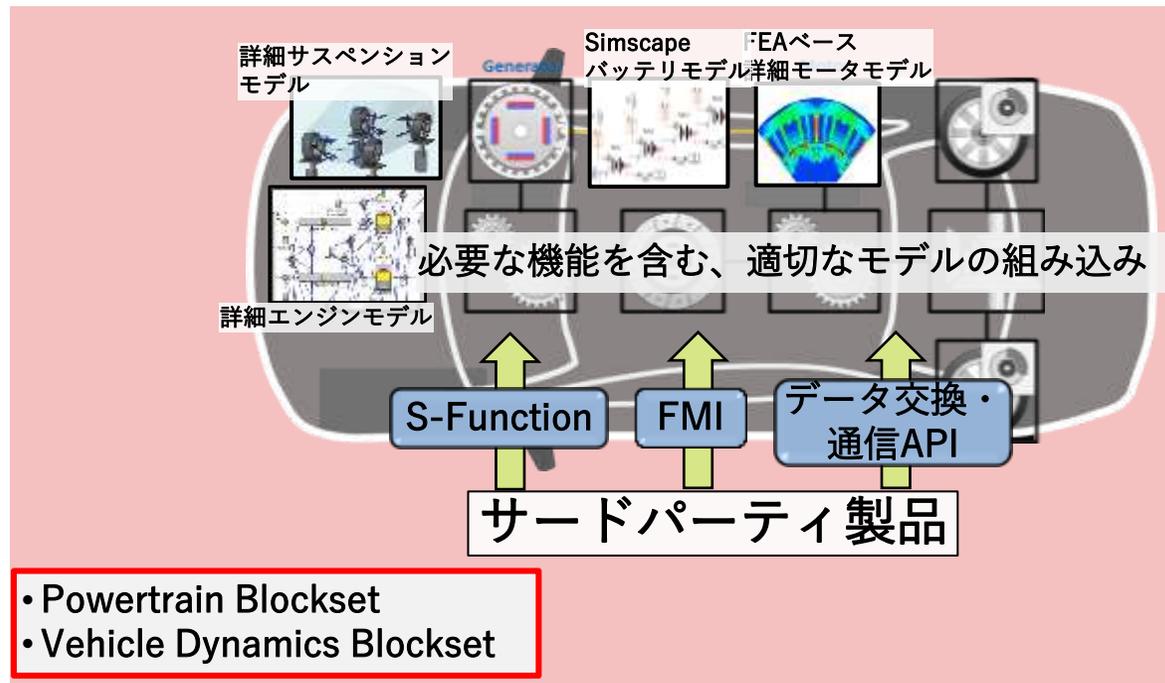
複数ドメインが混ざる
複雑なシステム



- 非線形応答のモデル化
- 3D-CAEシミュレーションの次元圧縮(ROM)と高速化

Simulinkで両アプローチを統合したシミュレーションが可能

モデルドリブンアプローチ 物理(第一原理)モデリング

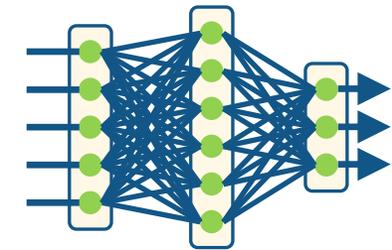
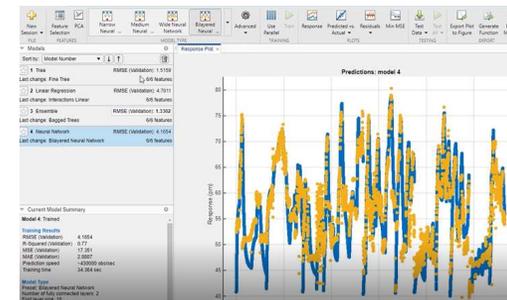


データドリブンアプローチ 統計・最適化・AI

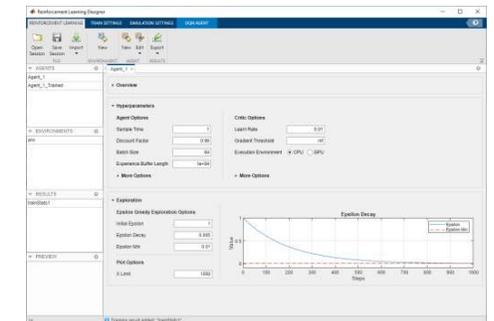
ディープラーニング



機械学習



強化学習



両方のアプローチ方法を

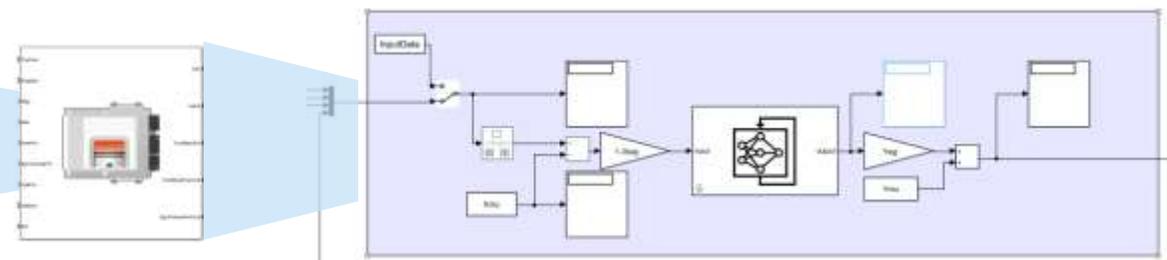
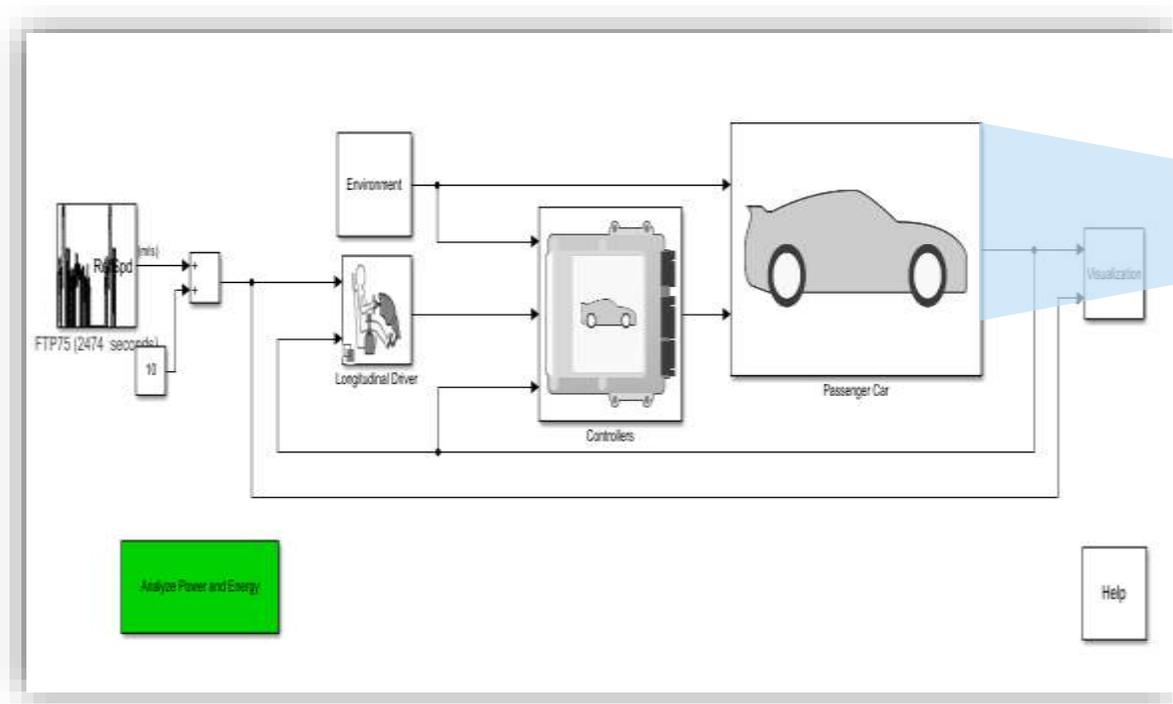
SIMULINK®

で統合シミュレーション

ステップごとに解説： モデルドリブン+AIの統合した活用フロー



ケース：ドライバ、コントローラ、自動車モデルの統合シミュレーション

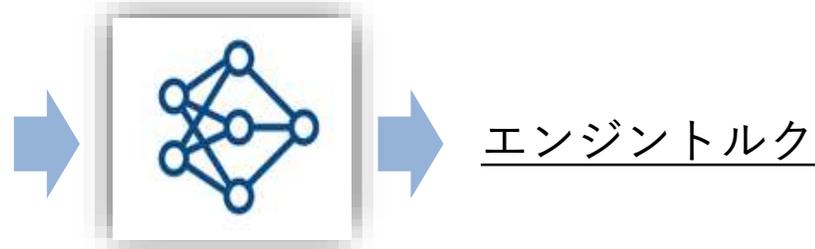


- エンジン出力のデータを持っている→AI化
- モデルドリブン+AIの統合シミュレーション
- HILSでのリアルタイムシミュレーション

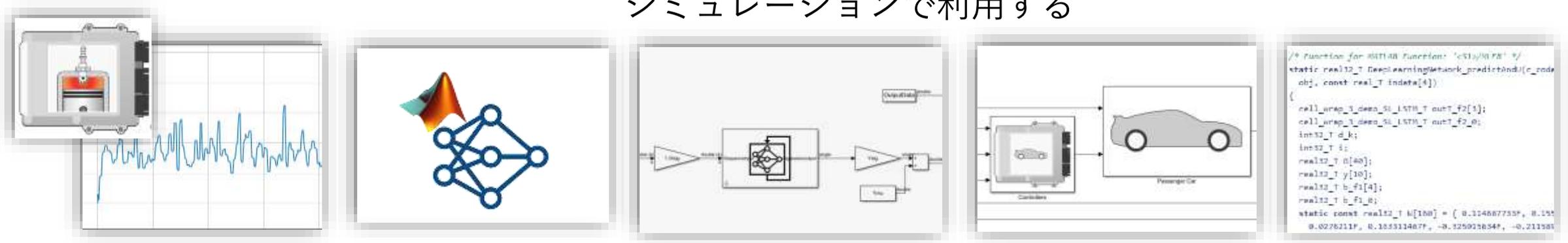
例題

エンジントルクの出力の応答を実データからAIモデル化して利用
自動車を目標速度で動作させた際のシステム全体の応答シミュレーション。

エンジン回転
点火時期
スロットルポジション
ウエストゲートバルブ



トルク出力を予測する LSTMモデル 学習後
シミュレーションで利用する



データ準備

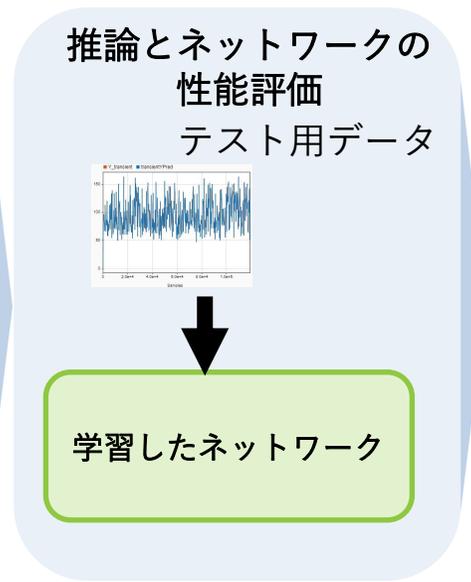
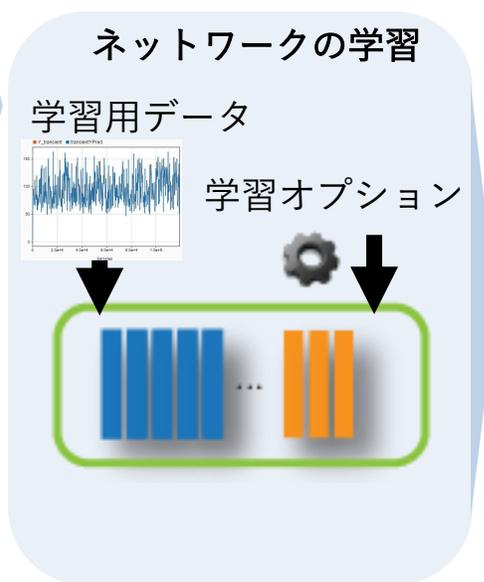
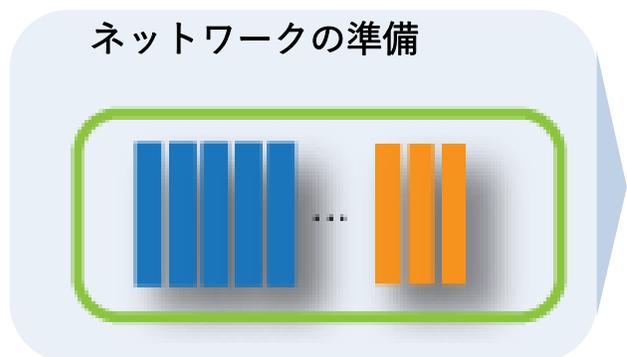
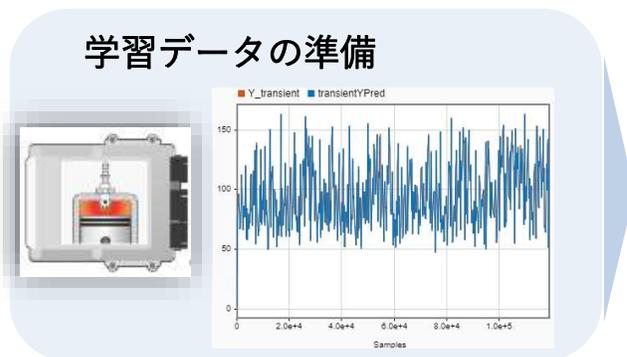
AIモデルの
学習

Simulinkへの
実装

システムモデル
への統合

HILS向け
コード生成

AIモデル「教師あり学習」のフロー



シミュレーション, etc
目的の用途で利用

※推論：学習済みのネットワークで予測すること

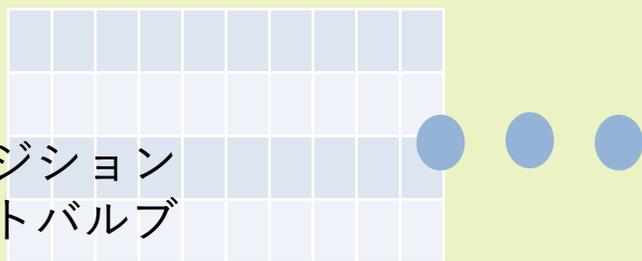
データの準備



データ：

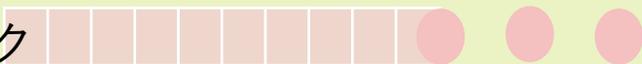
入力特徴量

- エンジン回転
- 点火時期
- スロットルポジション
- ウェストゲートバルブ



応答

- エンジントルク



大半のデータが $-1 \sim 1$ に入るように平均、分散でデータを正規化

データ読み込み

■ データ量が少ない：

- txt, csv ファイル
- matファイル

■ データ量が多い、複数ファイル：

• datastore

- tabularTextDatastore (表形式)
- signalDatastore (信号用)
- audioDatastore (オーディオ用)
- fileDatastore (カスタムのファイル形式)
- etc...

→メモリ消費を抑えた効率的な読み込み

Current Folder

- _Copy_of_Demo_LSTM_f...
- Demo_LSTM_for_Engine...
- LogData20210204_doe5...
- net.mat
- params.mat
- Y.mat

入力特徴量

- エンジン回転
- 点火時期
- スロットルポジション
- ウエストゲートバルブ

応答

- エンジントルク

WgAreaPct 1x118962 do...
X_transient 4x118962 do...
Y_transient 1x118962 do...
Ymu 94.1296
Ysig 25.4823

```
load LogData20210204_doe512.mat
data = loglogout;
EngSpd = data{7}.Values.Data';
SpkAdv = data{11}.Values.Data';
ThrPosPct = data{14}.Values.Data';
WgAreaPct = data{15}.Values.Data';
EngTrq_fil = data{4}.Values.Data';
% 入力特徴量データの準備
X_transient = [ThrPosPct; ...スロットルポジション
               WgAreaPct; ...ウエストゲートバルブ
               EngSpd; ...エンジン回転
               SpkAdv]; % 点火時期

% 応答
Y_transient = EngTrq_fil; % エンジントルク

tdatalength = 100000;

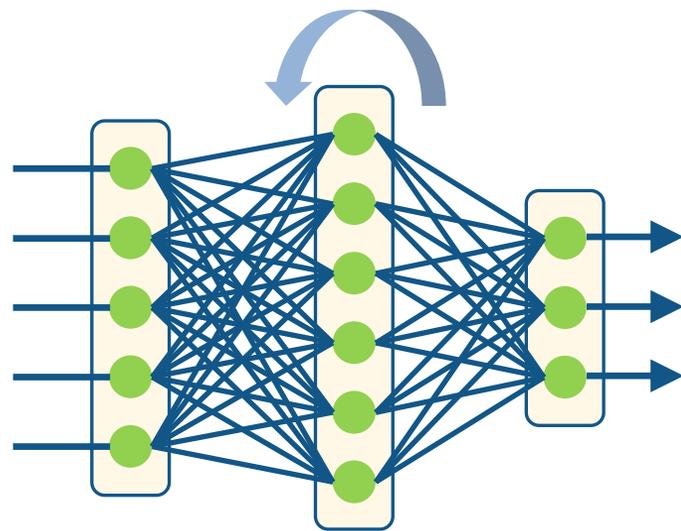
testdataidx = 100001;
idx_traindata = zeros(1,numel(EngTrq_fil));
idx_testdata = idx_traindata;
```

matファイルからのデータ読み出し

入力特徴量の準備

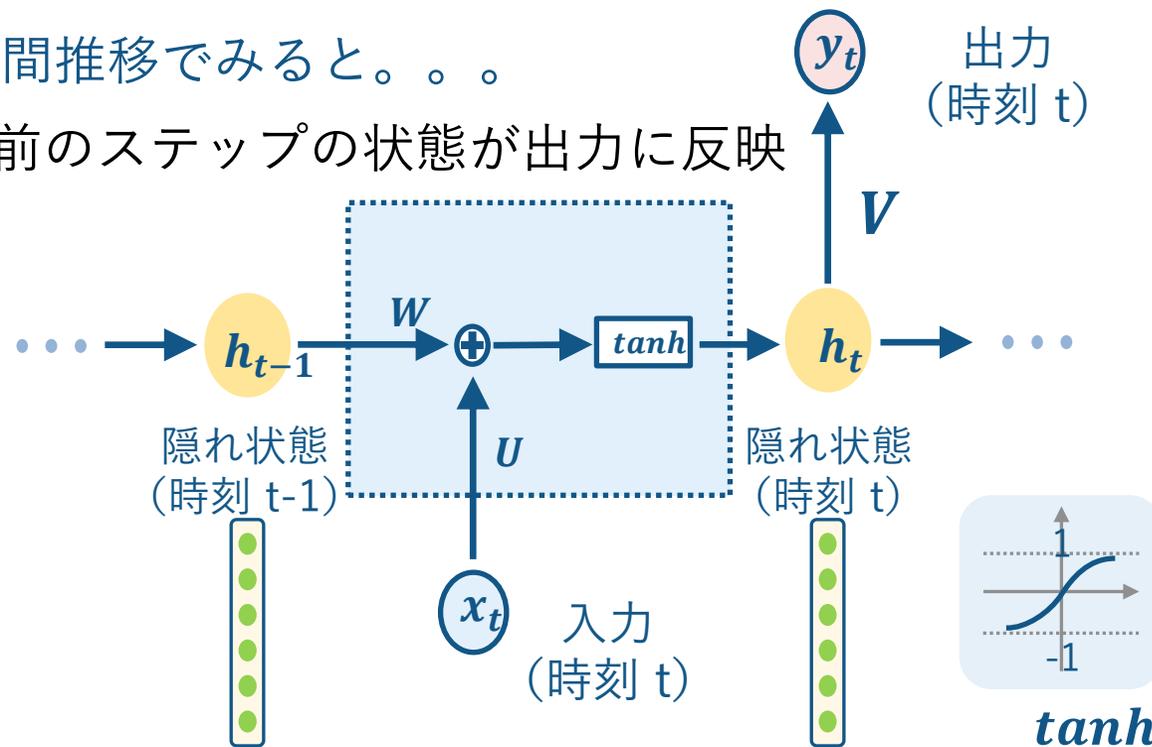
ネットワークの準備：なぜLSTM？

時系列シミュレーションに向いているRecurrent Neural Network



エルマン型 RNN

時間推移でみると。。。
RNNでは前のステップの状態が出力に反映

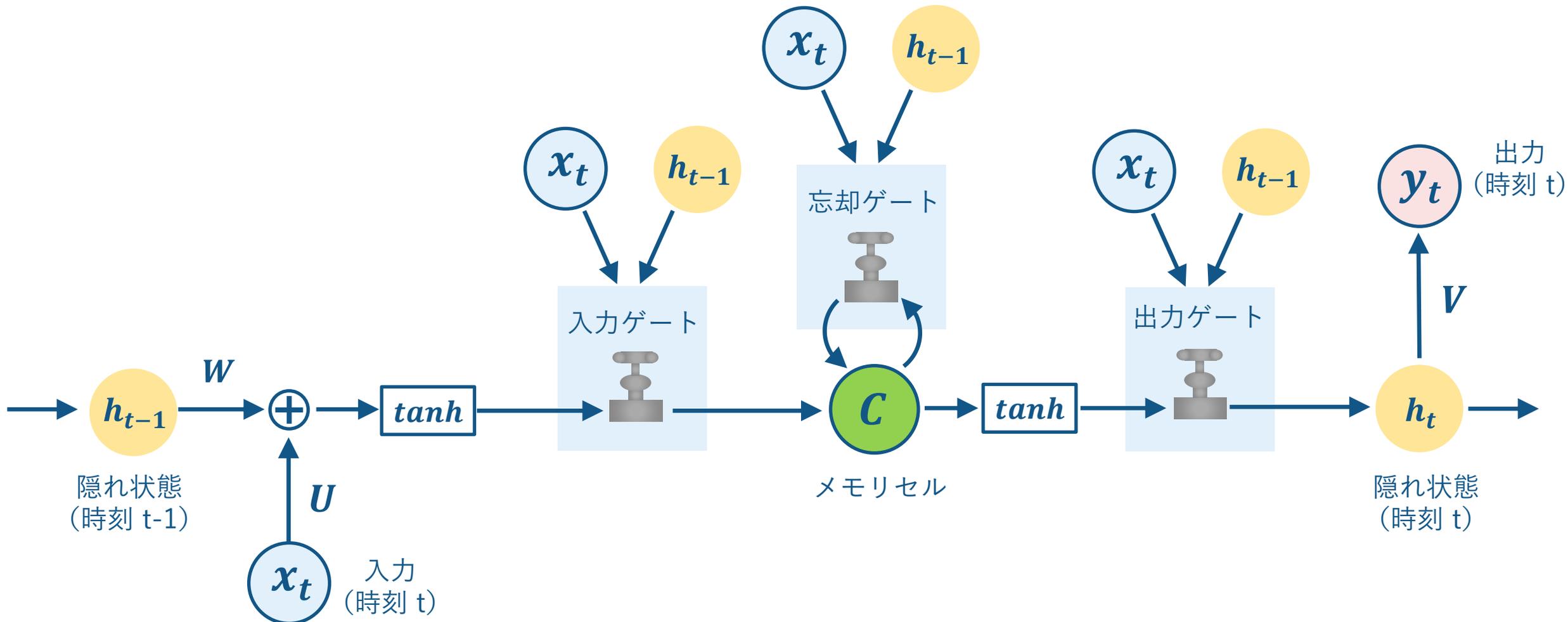


➡ Simulinkでの時系列シミュレーションに向いている

直前の入力への依存度が高い

LSTM (Long Short Term Memory) とは？

通常の RNN に「メモリセル」を導入し、
入力・忘却・出力を3つのゲートで制御するようにしたもの



Current Folder

- _Copy_of_Demo_LSTM_f...
- Demo_LSTM_for_Engine...
- LogData20210204_doe5...
- net.mat
- params.mat
- X.mat
- Y.mat

Demo_LSTM_for_EngineTrq.mlx (Live Script)

Name	Value
tdatale...	100000
testdataidx	100001
ThrPosPct	1x118962 do...
WgAreaPct	1x118962 do...
X_transient	4x118962 do...
Y_transient	1x118962 do...
Ymu	94.1296
Ysig	25.4823

```
33 % 平均分散からデータを正規化する
34 std_transientX = (X_transient - Xmu) ./ Xsig;
35 std_transientXtrain = std_transientX(:,1:tdatale...);
36 std_transientY = ((Y_transient - Ymu) ./ Ysig);
37 std_transientYtrain = std_transientY(:,1:tdatale...);
38 std_transientXtest = std_transientX(:,testdataidx:end);
39
```

ネットワークの作成

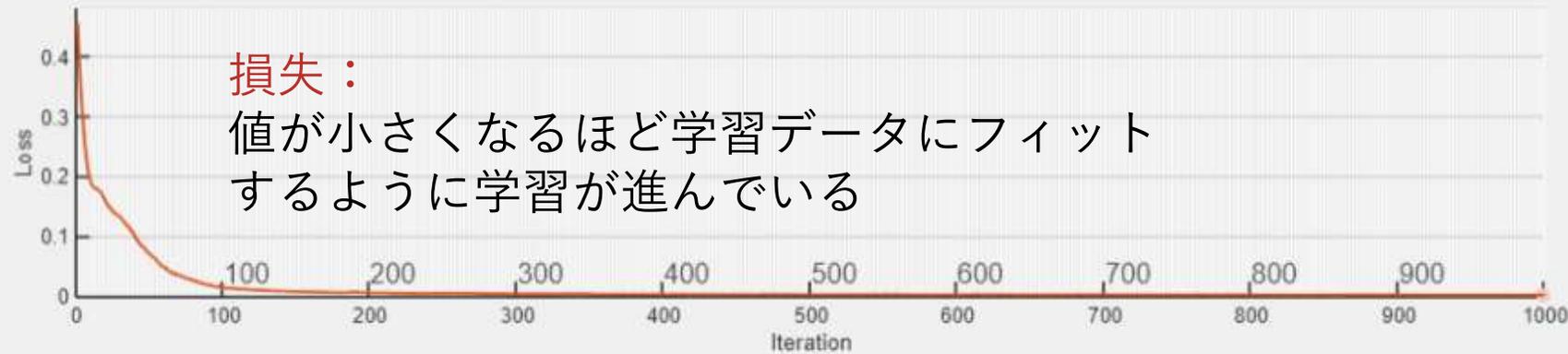
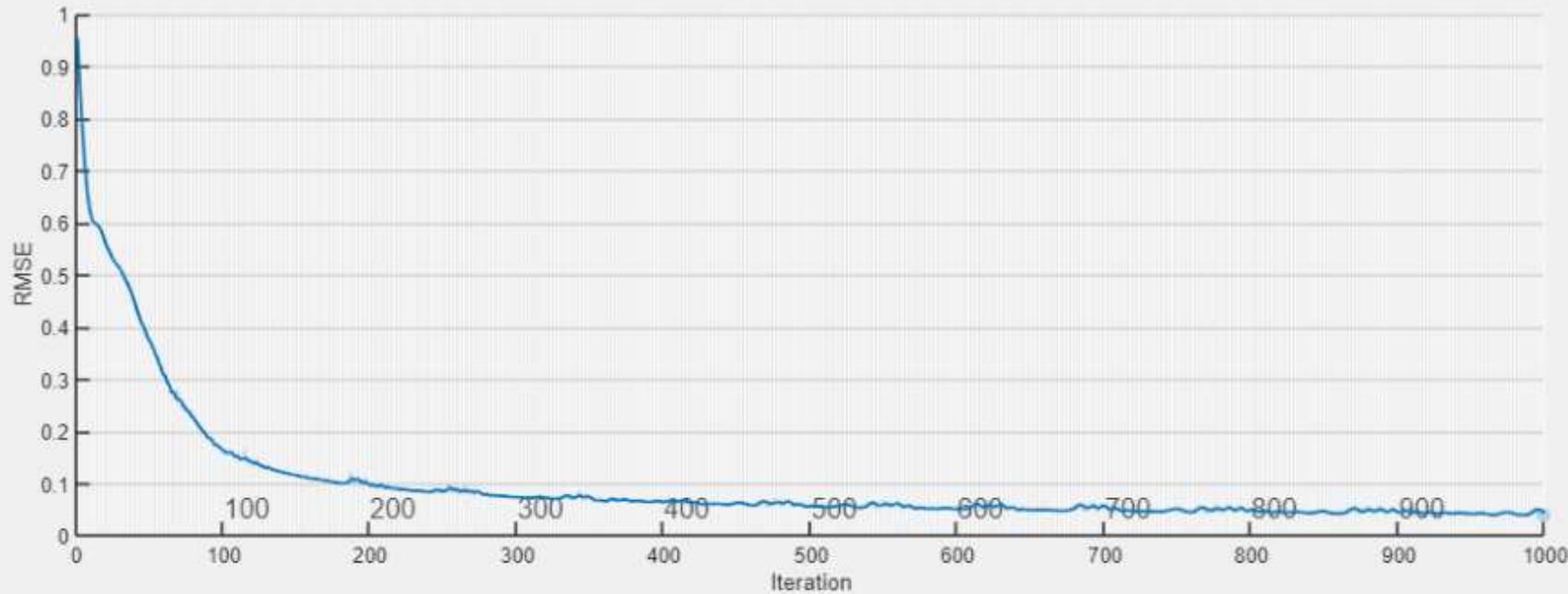
deepNetworkDesigner

ディープネットワークデザイナー
でネットワークを作成します

ネットワークの構築



Training Progress (2021/05/09 23:08:41)



損失：
値が小さくなるほど学習データにフィット
するように学習が進んでいる

Results

Validation RMSE: N/A
Training finished: Reached final iteration

Training Time

Start time: 2021/05/09 23:08:41
Elapsed time: 199 min 1 sec

Training Cycle

Epoch: 1000 of 1000
Iteration: 1000 of 1000
Iterations per epoch: 1
Maximum iterations: 1000

Validation

Frequency: N/A

Other Information

Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

[Learn more](#)

RMSE

— Training (smoothed)
—•— Training
- -•- - Validation

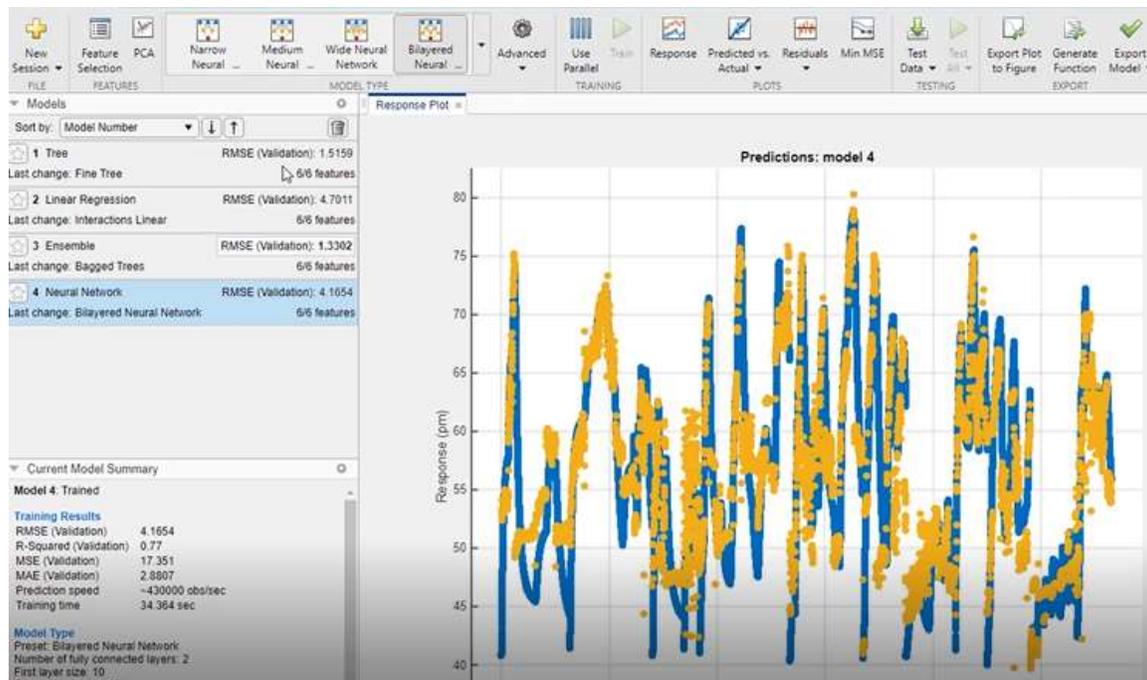
Loss

```
save('net.mat','net');
```

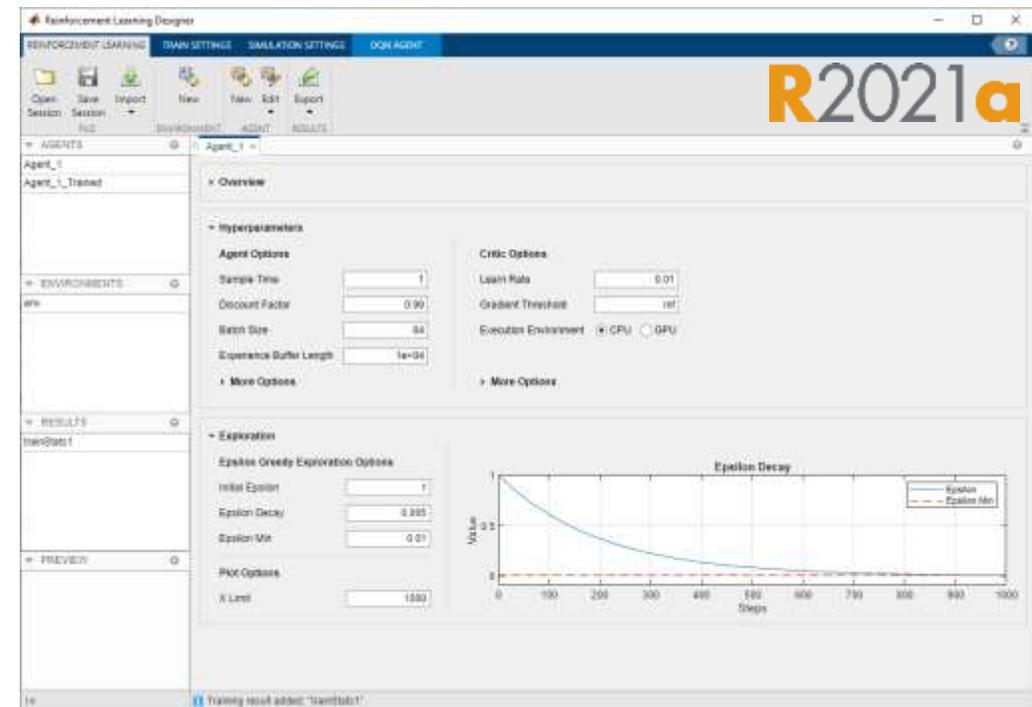
← 学習したモデルはnet.matでファイルで保存。
Simulinkでのシミュレーションに利用

機械学習・強化学習のモデル化で使えるGUIのApps

分類学習器/回帰学習器



強化学習アプリ



初めての方でも、手軽にAI開発に取りかかれます

学習済みネットワークでの推論と評価

データ準備

AIモデルの
学習

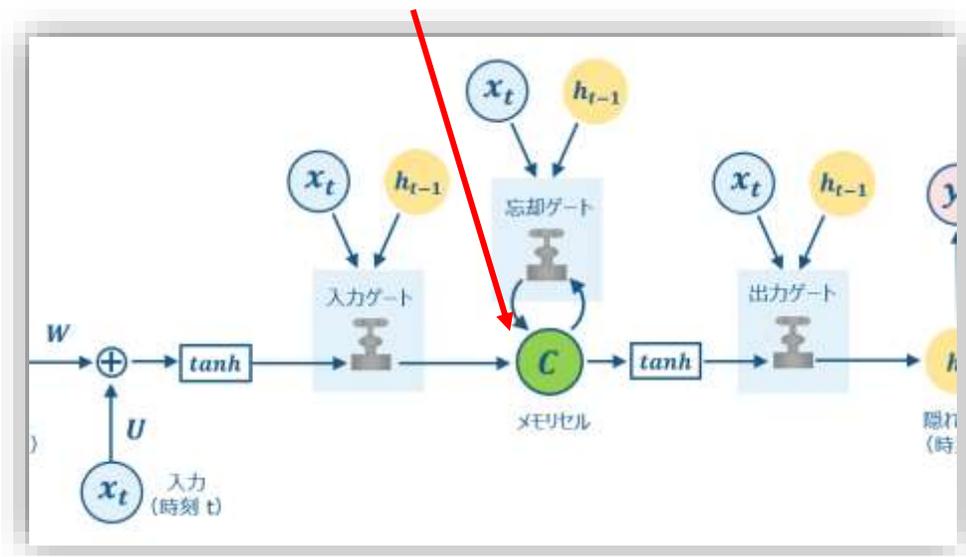
Simulinkへの
実装

システムモデル
への統合

コード生成

predictAndUpdateState関数

推論を実行しメモリセルのパラメータを更新



- 評価
 - 利用適するレベルか判断基準を設ける

今回：
RMSE (二乗平均平方根誤差)

※推論：学習済みのネットワークで予測すること

推論

```
73 % 推論してLSTMのメモリセルの状態をアップデート
74 [transientnet,std_transientYPred] = predictAndUpdateState(net,std_transientX,'MiniBatchSize',96)
75 % 正規化データを元のスケールに戻す
76 transientYPred = (std_transientYPred .* Ysig) + Ymu;
77 % RMSEの計算
78 rmse_transient = sqrt(mean((Y_transient(testdataidx:end)-transientYPred(testdataidx:end)).^2))

rmse_transient = single
    2.0056
```

```
79 % グラフ&評価結果
80 figure,
81 subplot(1,2,1),plot(Y_transient(1:tdatalength),'DisplayName','standardizedYtrain2');hold on;plot
82 title(["学習データ長："+num2str(tdatalength)," 過渡モデル："+modeltype ,"学習データ",""])
83 subplot(1,2,2),plot(Y_transient(testdataidx:end),'DisplayName','standardizedYtrain2');hold on;pl
84 title(["データ長さ："+num2str(tdatalength) ,"過渡モデル："+modeltype ,"評価データ",...
85 "RMSE："+num2str(rmse_transient) ])
```

学習データ長:100000
過渡モデル: LSTM

データ長さ:100000
過渡モデル: LSTM

学習モデルのSimulink実装

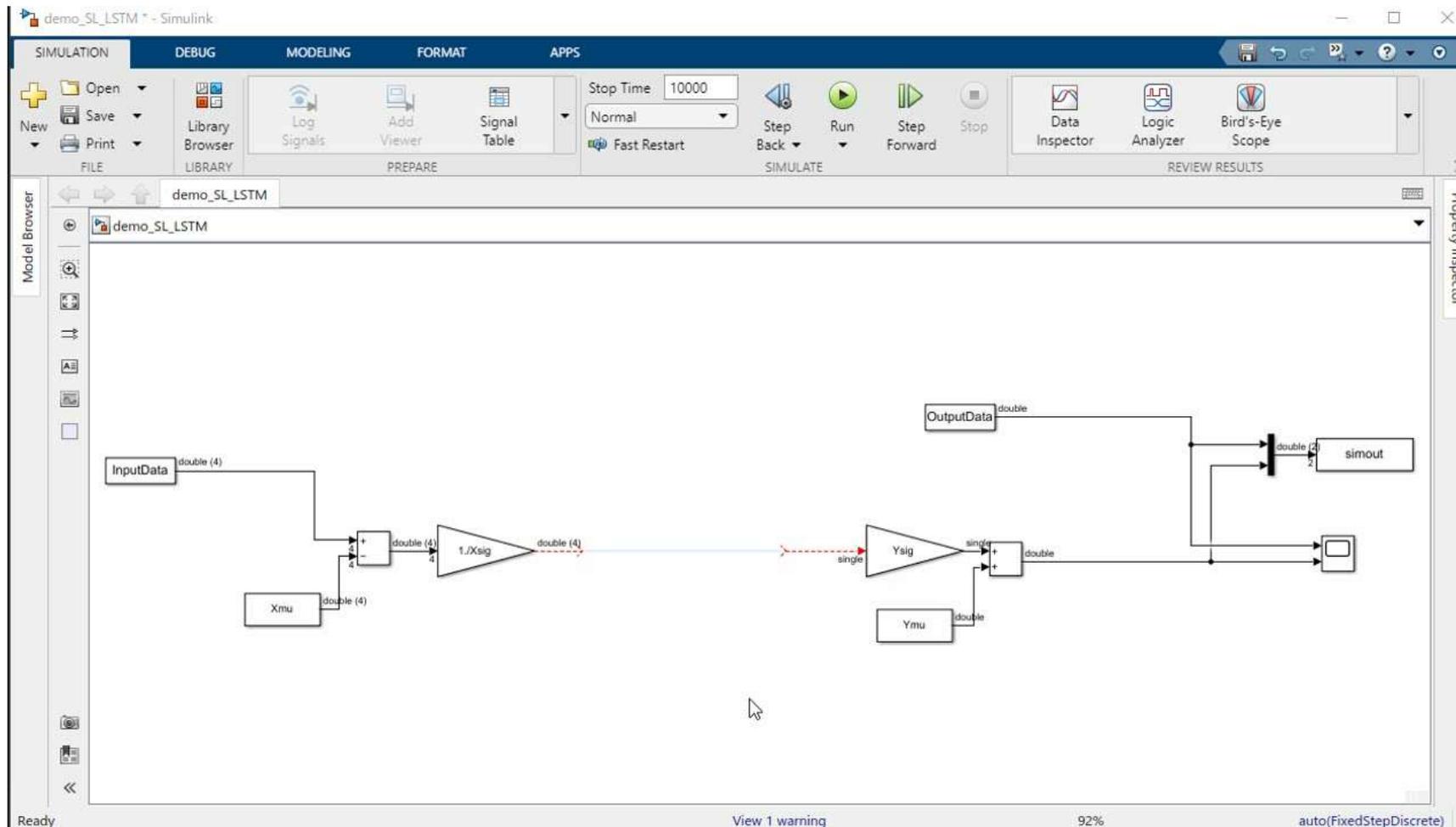
データ準備

AIモデルの
学習

Simulinkへの
実装

システムモデル
への統合

コード生成



データを準備

→ Stateful Predict Block

→ ネットワーク呼び出し

→ Simulationの実行

システムレベルのモデルに統合して全体シミュレーション

データ準備

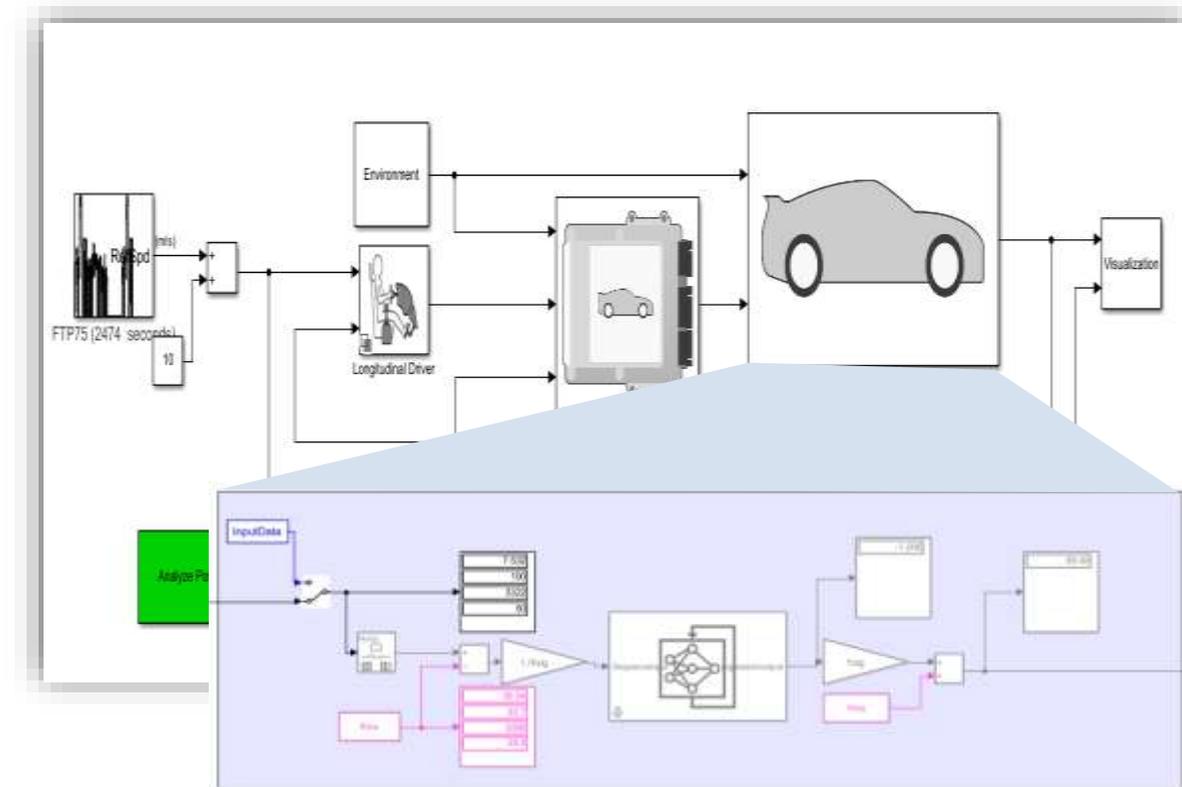
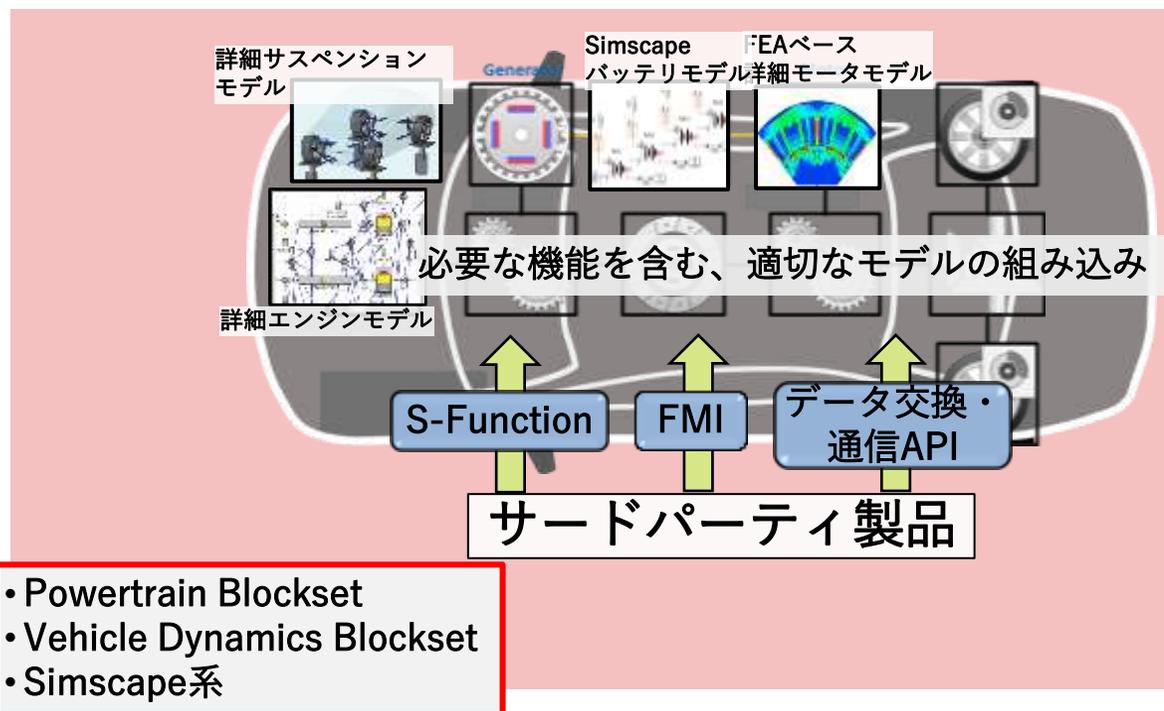
AIモデルの
学習

Simulinkへの
実装

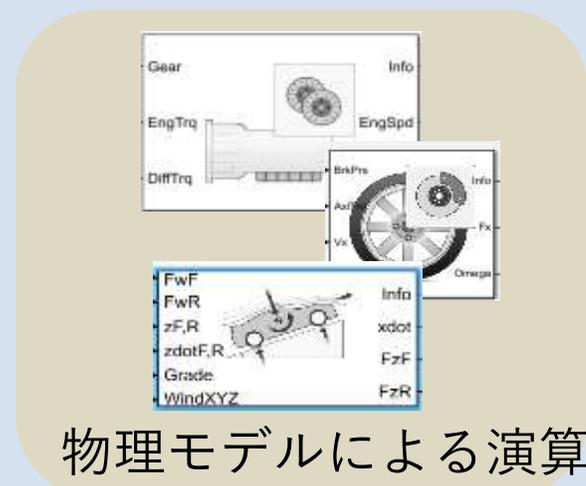
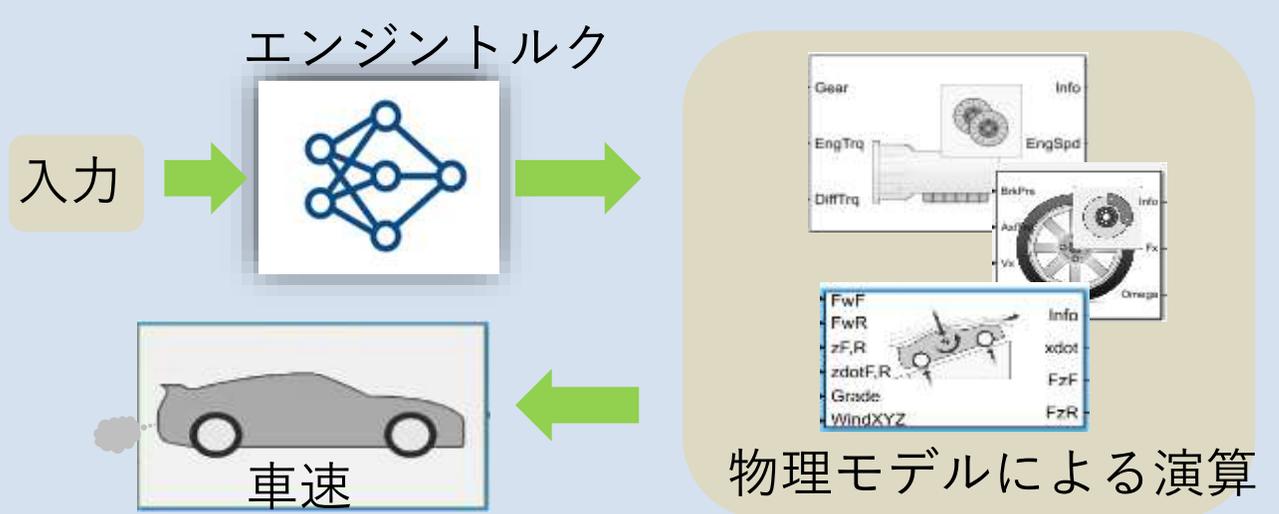
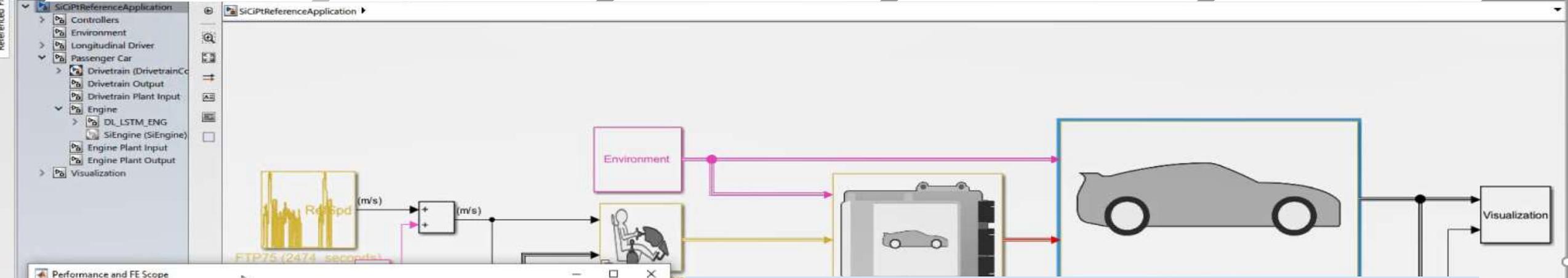
システムモデル
への統合

コード生成

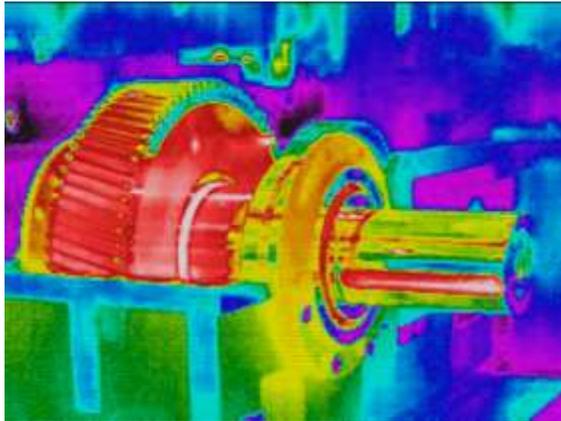
全体シミュレーションの環境を提供



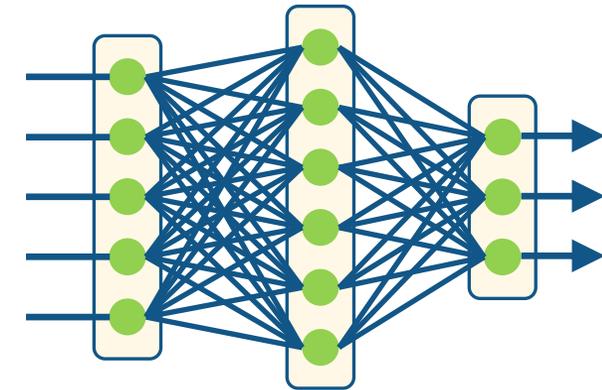
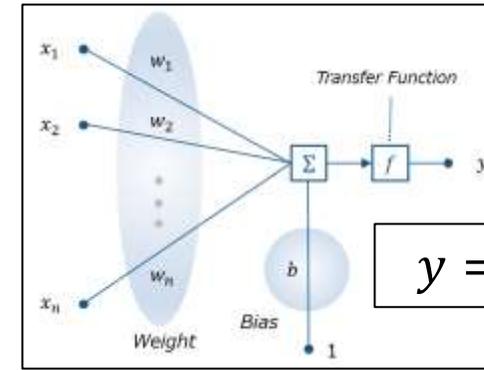
システムレベルのモデルに統合して全体シミュレーション



ROM(Reduced Order Model)での高速化



3D-CAEによる
詳細シミュレーション
(1ステップごとの計算負荷大)

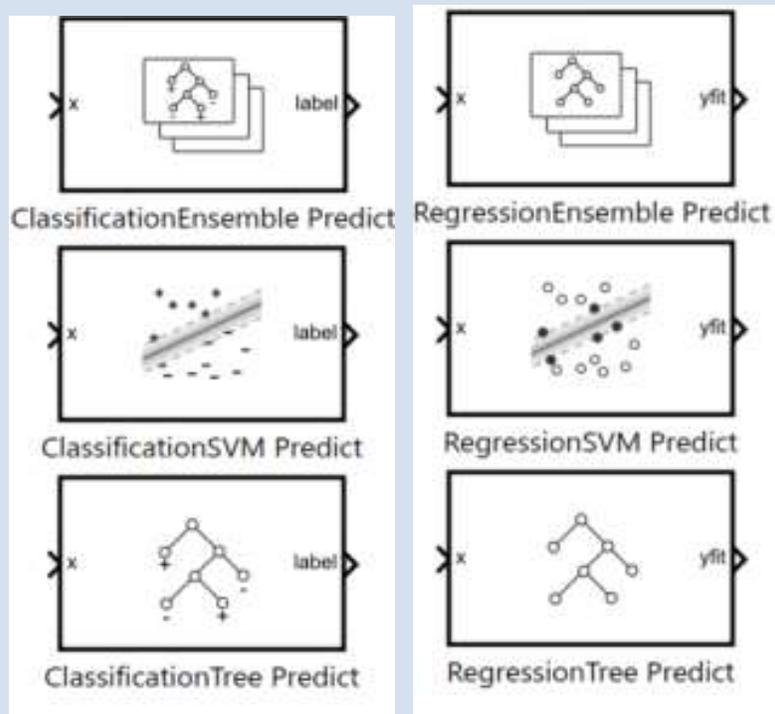


- AIモデル
 - 次元圧縮
 - 専用ライブラリ
 - GPUでの並列化

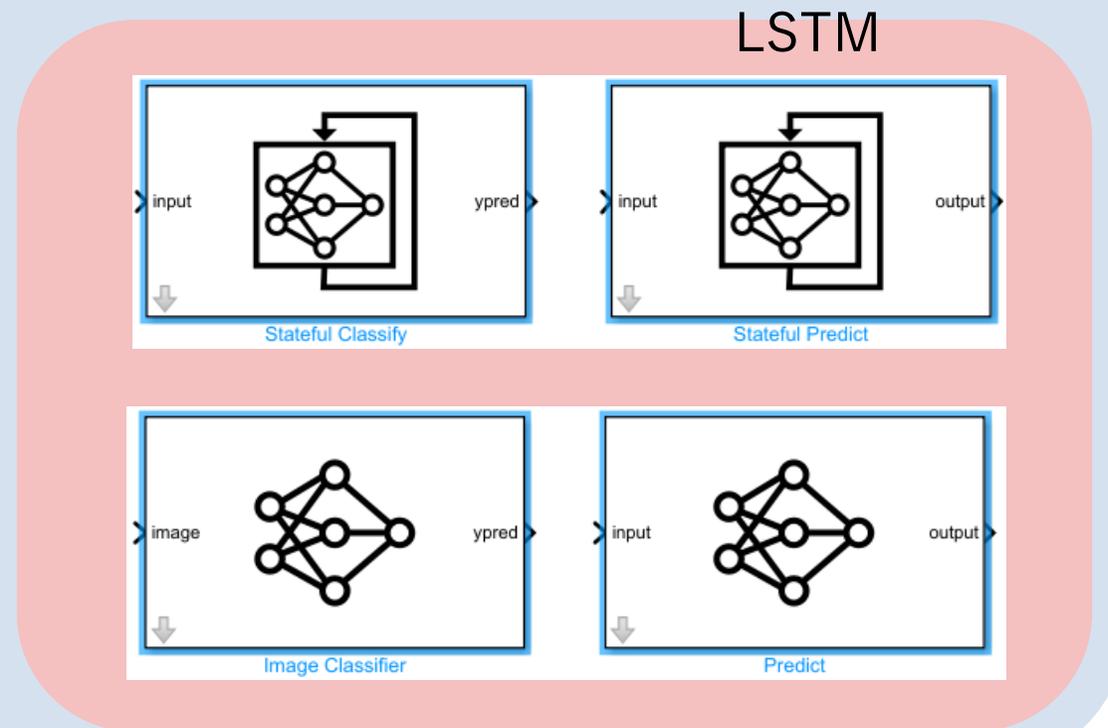
AIモデル活用による次元圧縮とシミュレーション高速化に期待

AIモデル推論用のSimulink

機械学習



ディープラーニング

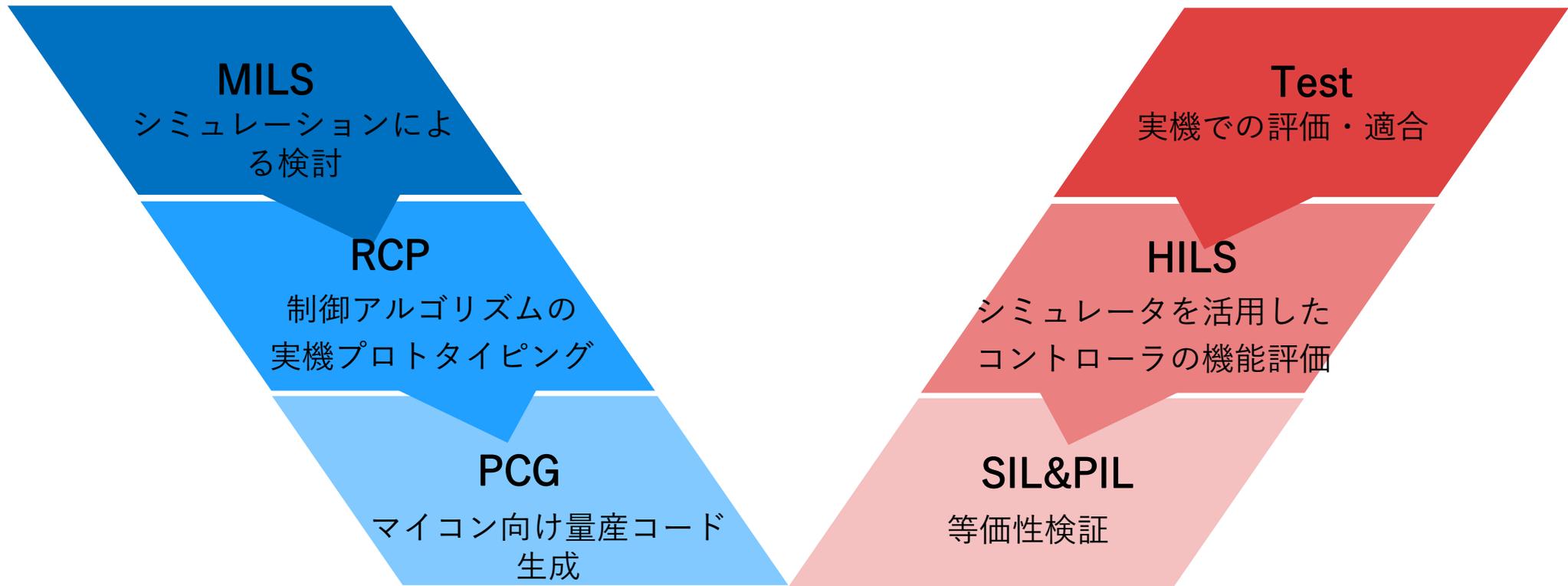


前処理 & 後処理



モデルベースデザインにおけるVプロセス

- 自動コード生成機能を活用して
コンセプト設計から機能実装・実機評価までカバー



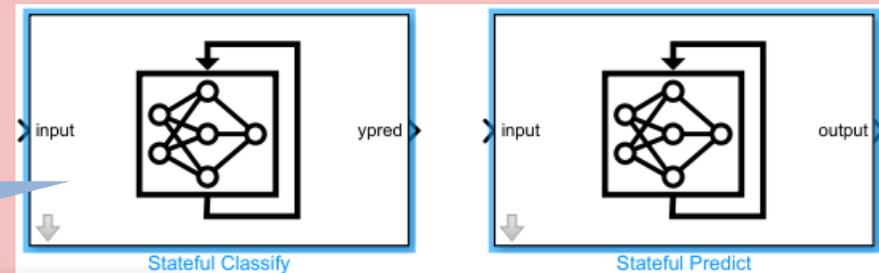
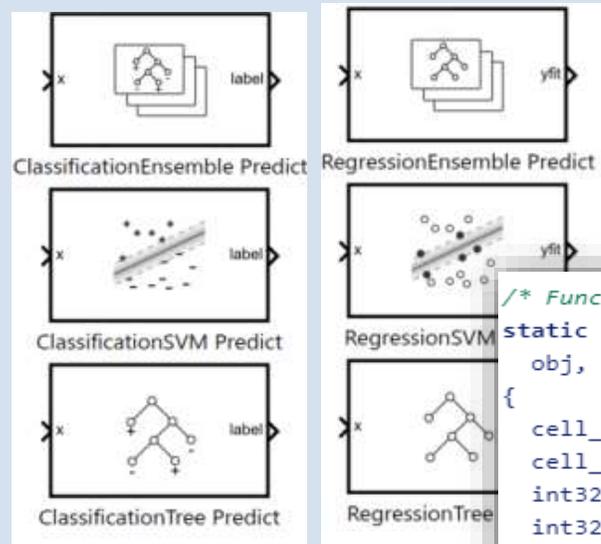
MILS: Model In the Loop Simulation
RCP: Rapid Control Prototyping
PCG: Product Code Generation

SILS: Software In the Loop Simulation
PILS: Processor In the Loop Simulation
HILS: Hardware In the Loop Simulation

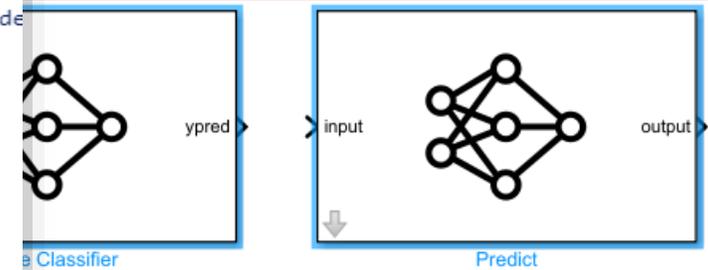
推論用ブロックはANSI C/C++コード生成に対応

機械学習

ディープラーニング



```
/* Function for MATLAB Function: '<S1>/MLFB' */
static real32_T DeepLearningNetwork_predictAndU(c_code
obj, const real_T indata[4])
{
    cell_wrap_3_demo_SL_LSTM_T outT_f2[3];
    cell_wrap_3_demo_SL_LSTM_T outT_f2_0;
    int32_T d_k;
    int32_T i;
    real32_T G[40];
    real32_T y[10];
    real32_T b_f1[4];
    real32_T b_f1_0;
    static const real32_T W[160] = { 0.114687733F, 0.155
0.0276211F, 0.163311467F, -0.325015634F, -0.211589
```



ANSI C/C++コード生成により、AIモデルの活用の自由度UP

デンソーテン、AI制御における モデルベース開発適用に向けたプロセスを構築

横山 夏軌氏, 株式会社デンソーテン

複雑化が進む自動車制御において、熟練者の経験に頼られていたり、定式化が難しい課題は多くあり、ディープラーニングは解決手法の一つとして期待されています。デンソーテンでは当初Python言語でのアルゴリズム開発に取り組みましたが、車載ECUで動作するCコードに変換できない、車載制御開発で肝となるモデルベース開発におけるシミュレーションに適用できない課題がありました。

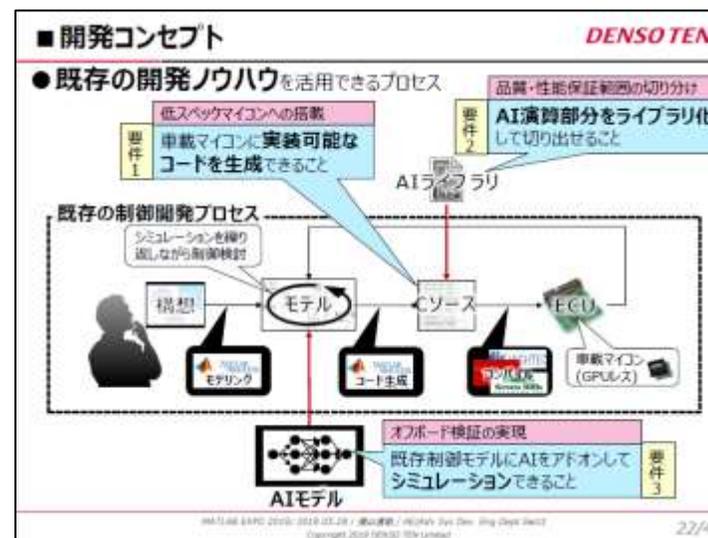
そこで既存制御モデルを構築していたSimulink®にMATLAB®で作成したAIモデルを取り込むプロセス構築に取り組みました。MATLABで作成したAIモデルからSimulink APIを用いてブロック作成、結線、学習値の追加を自動で行い、ECUへの実装を想定した専用のAIライブラリを利用してモデル全体でのシミュレーションに成功しました。加えてSimulinkモデルからMATLABでのAIモデルに逆変換する手順も構築しています。AI制御において設計から実装まで一貫性のあるモデルベース開発のためのプロセスが完成し、製品適用に向けて開発を進めています。

MATLAB/Simulink利用のメリット:

- 既存制御モデルにAIモデルを統合し効率よくモデルベース開発に適用
- ディープネットワークデザイナーを用いたマウス操作でのネットワーク構築
- MATLAB/Simulink間のAPIによるディープラーニングモデル双方向変換
- S-functionによる独自AIライブラリの利用

“ AIを制御用ECUで利用するにはモデルベース開発のフロー適用は不可欠です。既存の制御モデルとAIモデルを合わせてシミュレーションができる環境が整い、今後の製品開発を加速していくことができます。

”



LSTMモデルのコード生成

データ準備

AIモデルの
学習

Simulinkへ
の実装

システムモデル
への統合

コード生成

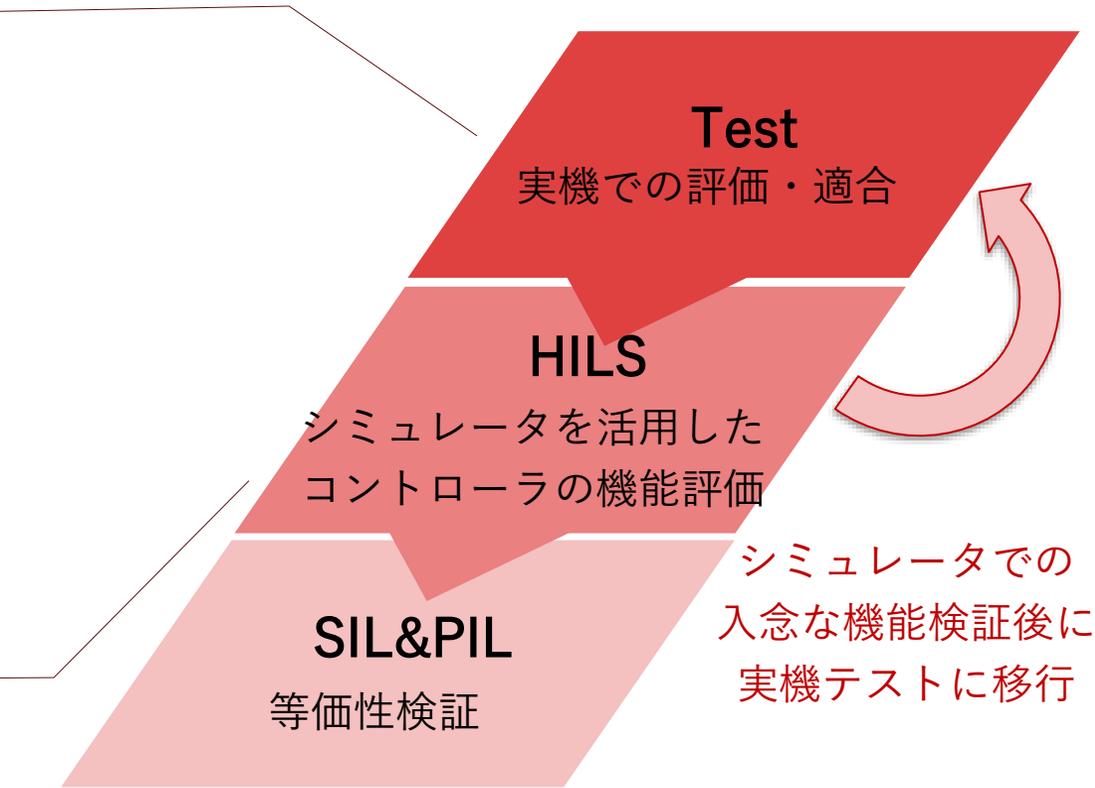
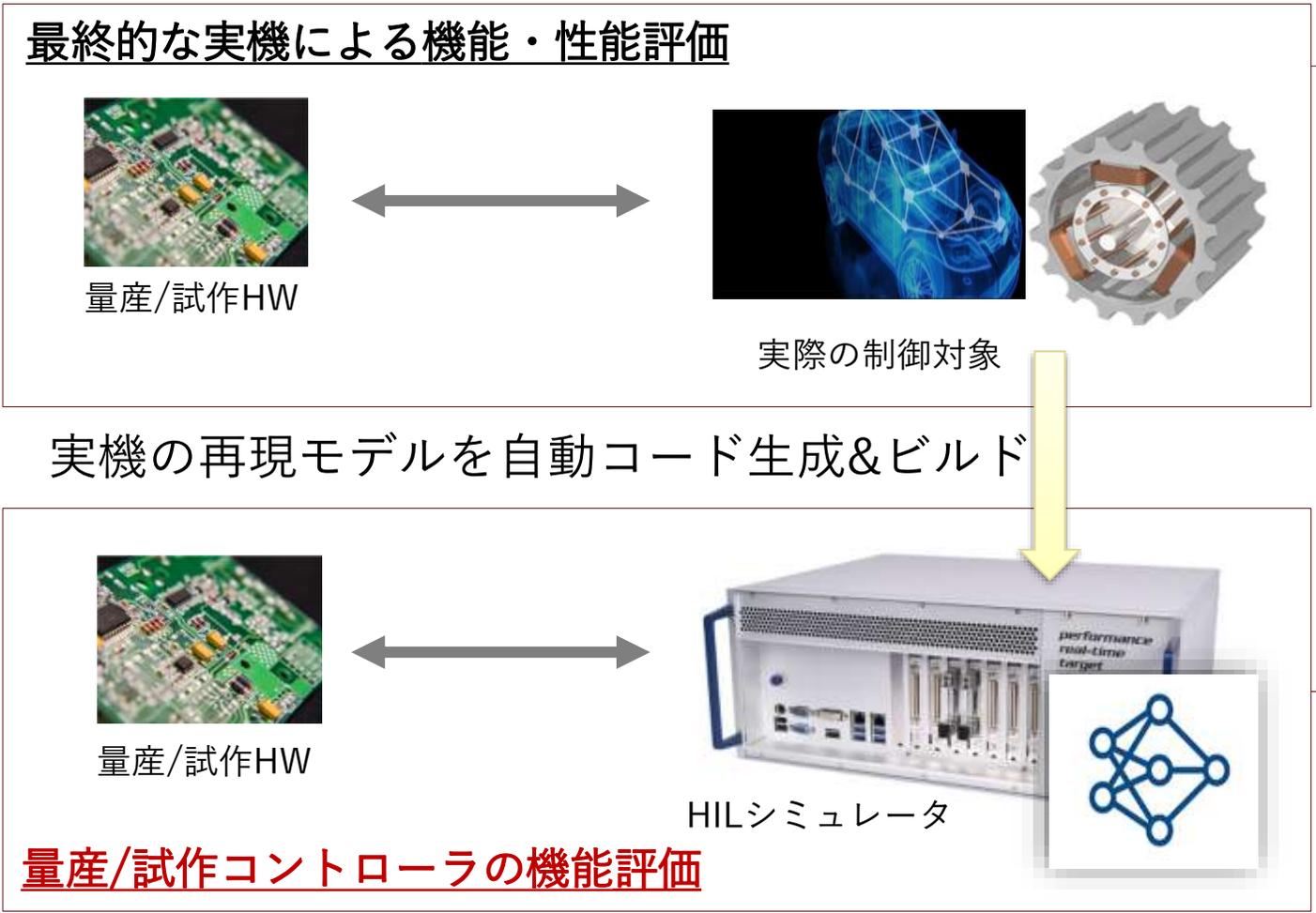
The screenshot displays the Simulink environment for a project named 'demo_SL_LSTM'. The 'C CODE' tab is active, showing a 'Generate C code' dialog box. The main workspace contains a Simulink block diagram with the following components and connections:

- InputData** (double (4)) and **Xmu** (double (4)) are inputs to a summing junction.
- The summing junction output goes to a gain block **1./Xsig**.
- The output of the gain block is connected to the **SequenceIn** input of the **SequenceIn** block.
- The **SequenceIn** block contains an LSTM neural network model.
- The **SequenceIn** block outputs **SequenceOutput** (single) to a gain block **Ysig**.
- The **Ysig** block output goes to another summing junction.
- Ymu** (double) is also an input to this summing junction.
- The summing junction output is connected to the **OutputData** (double) block.
- The **OutputData** block is connected to a **simout** block.

The bottom status bar shows 'Code Mappings - C', 'Ready', 'View 2 warnings', '94%', and 'auto(FixedStepDiscrete)'.

HILS: Hardware-In-the-Loop Simulation

- シミュレータを用いて量産/試作コントローラの機能チェックを実施



HILS 導入のメリット



任意の試験条件の素早い再現



実際の設備の占有時間を削減
(現場に行かずに最大限のデバッグ)



再現性の高い繰り返し試験



繰り返し試験の自動化



実物を壊さずに故障モードを再現

実際のコントローラ



通信/I/O

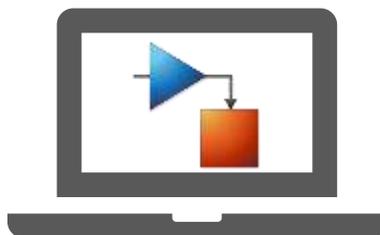


HILシミュレーター

モデルのビルド & ダウンロード
パラメータの調整



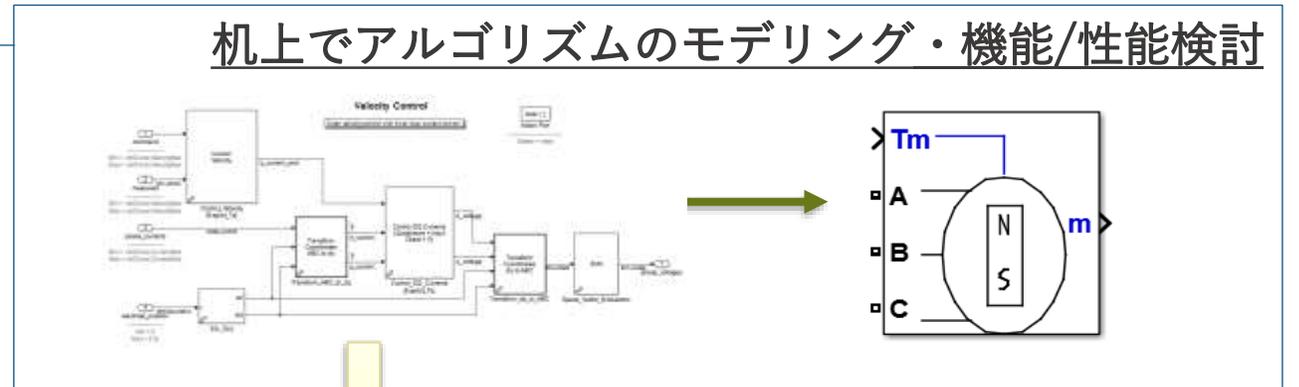
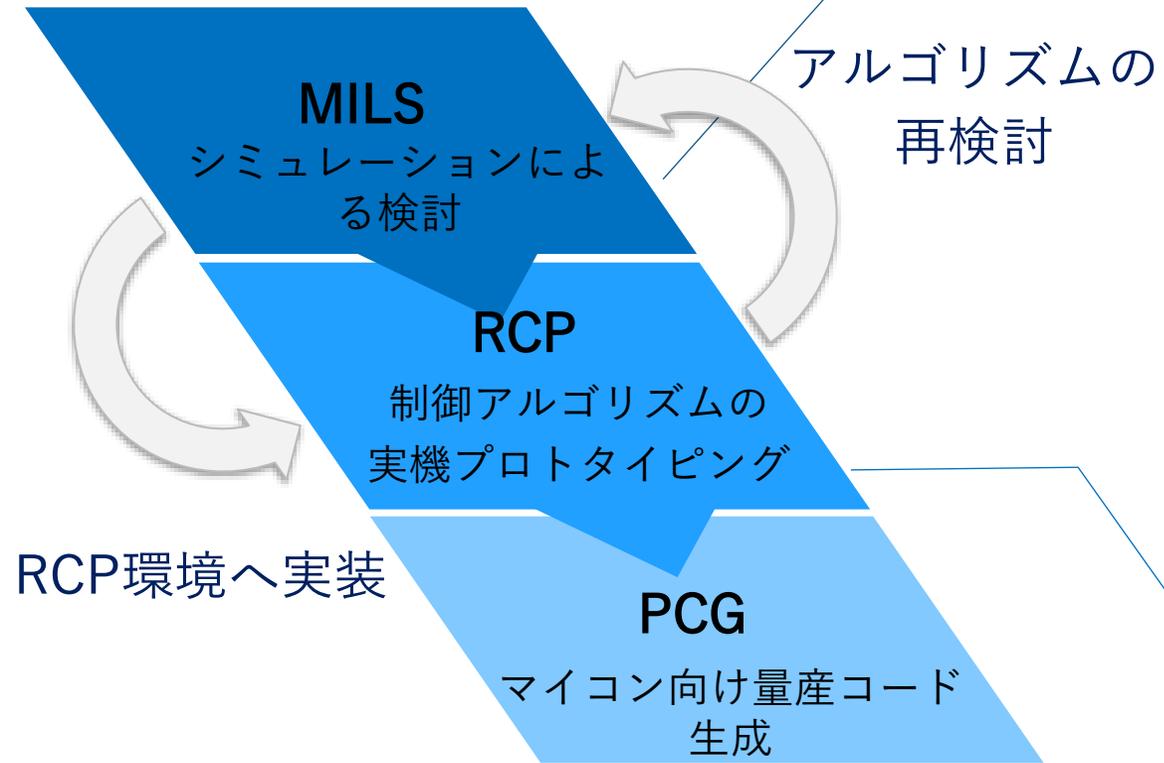
信号のモニタ・ロギング



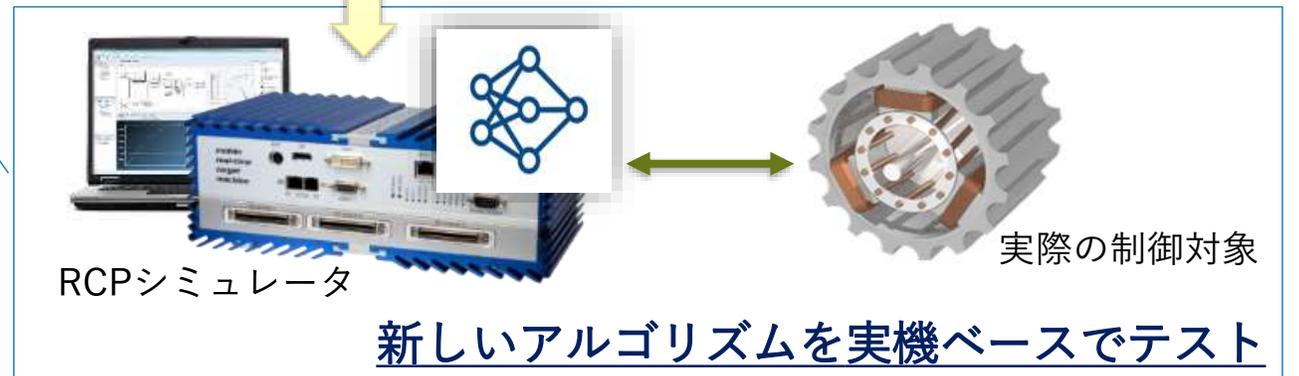
MATLAB & Simulink

RCP: Rapid Control Prototyping

- 検討したアルゴリズムを汎用性の高いRCPハードウェアへ素早く実装&テスト

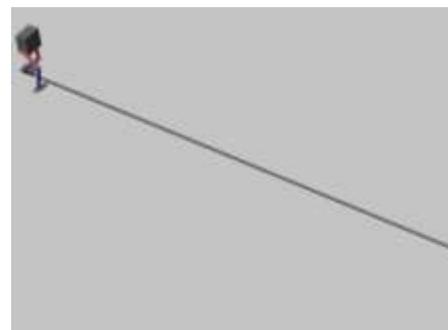
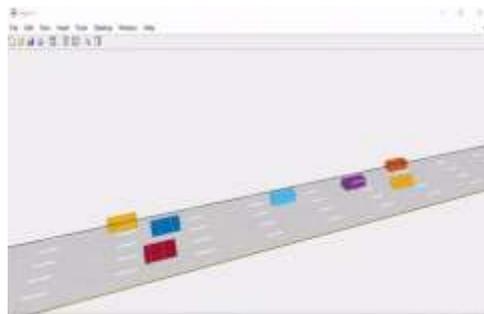
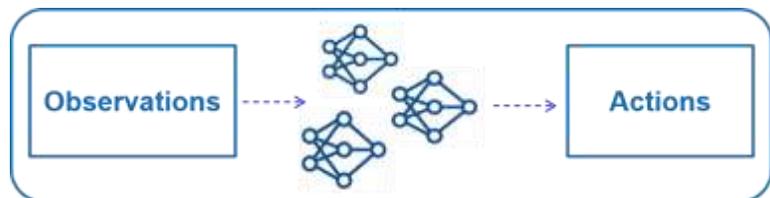


コントローラモデルを自動コード生成 & ビルド



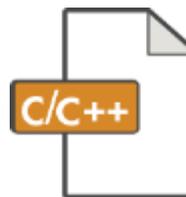
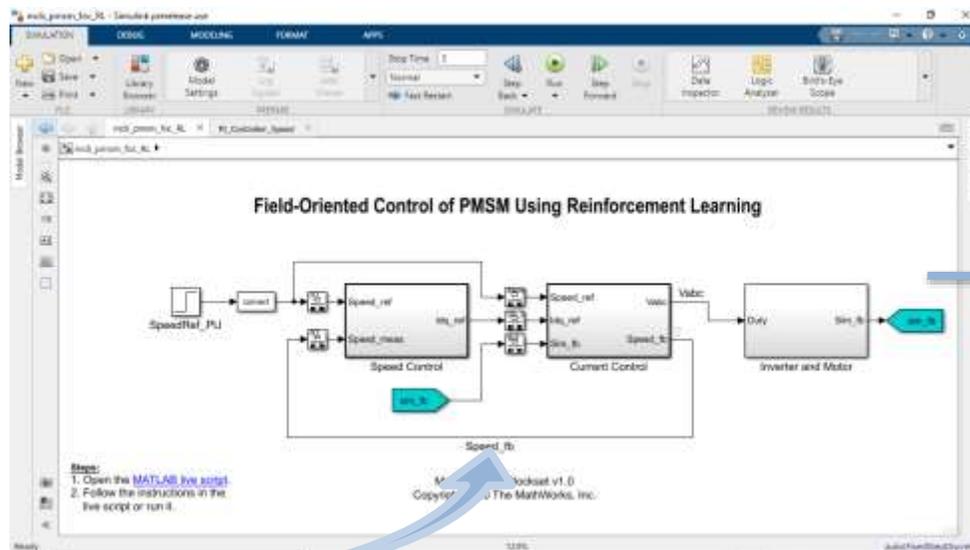
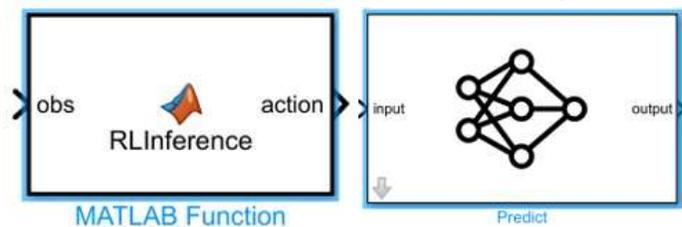
アルゴリズム開発での強化学習とラピッドプロトタイピング (RCP)

Agentの作成と学習



`generatePolicyFunction(agent);`

Policy



RCP 導入のメリット



シミュレーション/実機テスト
双方へシームレスに移行



ハンドコーディングなしで
モデルを素早く実装

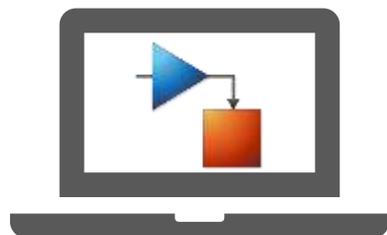


テスト・計測環境の素早い構築



現場で判明したロジックの問題に
モデル・パラメータ変更で即時対応

MATLAB & Simulink



モデルのビルド & ダウンロード
パラメータの調整



信号のモニタ・ロギング



RCPシミュレーター

実際の制御対象



通信/I/O

Vitesco Technologies、パワートレイン制御に深層強化学習を適用

Vivek Venkobarao, Gautham T. Sidharthan, Vitesco Technologies

Vitesco Technologiesは、クローズドループのパワートレイン制御に深層強化学習を適用しました。

パワートレイン制御は、非常に多様な環境条件を考慮しなければならない難しい課題です。さらに、地球規模での気候変動や排ガス規制が厳しくなる中、開発期間を短縮する必要があり、プロトタイピングを迅速に行うことが求められています。

Reinforcement Learning Toolboxにより、強化学習エージェントのプロトタイプ作成、生成、最適化を迅速に行い、開発期間を大幅に短縮することができました。

MATLABおよびSimulinkを強化学習に適用するメリット

- 強化学習エージェントの迅速なプロトタイピングと開発期間の短縮
- Simulinkを使用した最先端のプラントモデリング
- 強化学習アルゴリズムのドキュメントとサンプル
- MathWorksのエキスパートによるテクニカルサポート

“

Reinforcement Learning Toolboxにより、開発時間を大幅に短縮することができました。強化学習エージェントの迅速なプロトタイピングと生成に非常に役立ちました。

”

CHOICE IN IMPLEMENTATION

REINFORCEMENT LEARNING REALIZATION OPTIONS

Complexity	Traditional Methods	MATLAB/SIMULINK
Time series Models	Need to derive time vector	Simulink to rescue
Solve the ODE	Numerical methods to be introduced	Solvers
Environment Models	Need to be recreated with time series	State of Art plant models
Action, State	Needs to be derived and implemented	Toolbox generates
Observations, Reward	Needs to be derived with time vector	Simulink to rescue
A2C(Actor and Critic)	Can use library	DL API Actor and Critic -> RL toolbox
DDPG Agent	Can use library	RL toolbox with good flexibility of playing with parameters
Component Integration	Complex - Time series and cross sectional data	Simple - MATLAB, Simulink, RL blocks in same platform

組み込み機器のコードを自動的に生成する

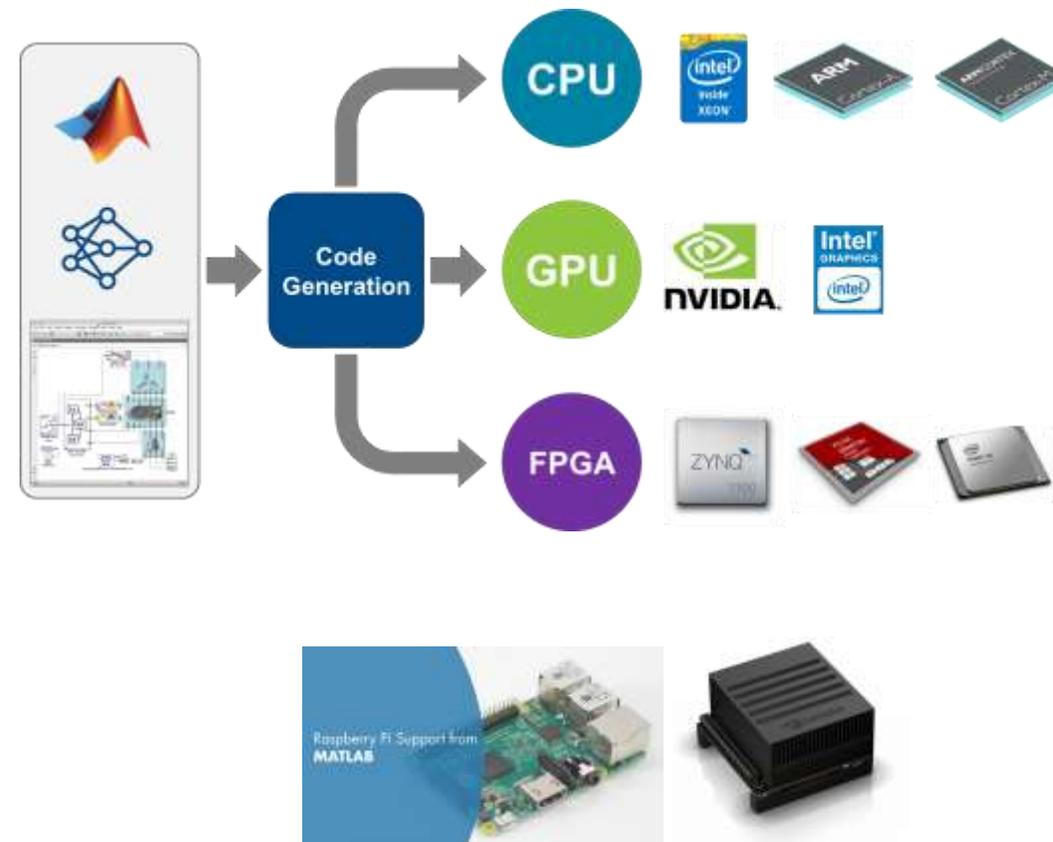
学習したモデルを低コストの試作ボードでテストしたり、モデルを再コーディングせずに製品化

最適化されたネイティブコードを生成します:

GPU – GPU Coder	R2017b
CPU – MATLAB Coder	R2018b
FPGA – Deep Learning HDL Coder	R2020b
ANSI C/C++コード生成	R2021a

低コストのボードでモデルをテスト:

Raspberry Pi	R2019a
ARM Targets	



展開されたモデルのメモリと電力消費の削減

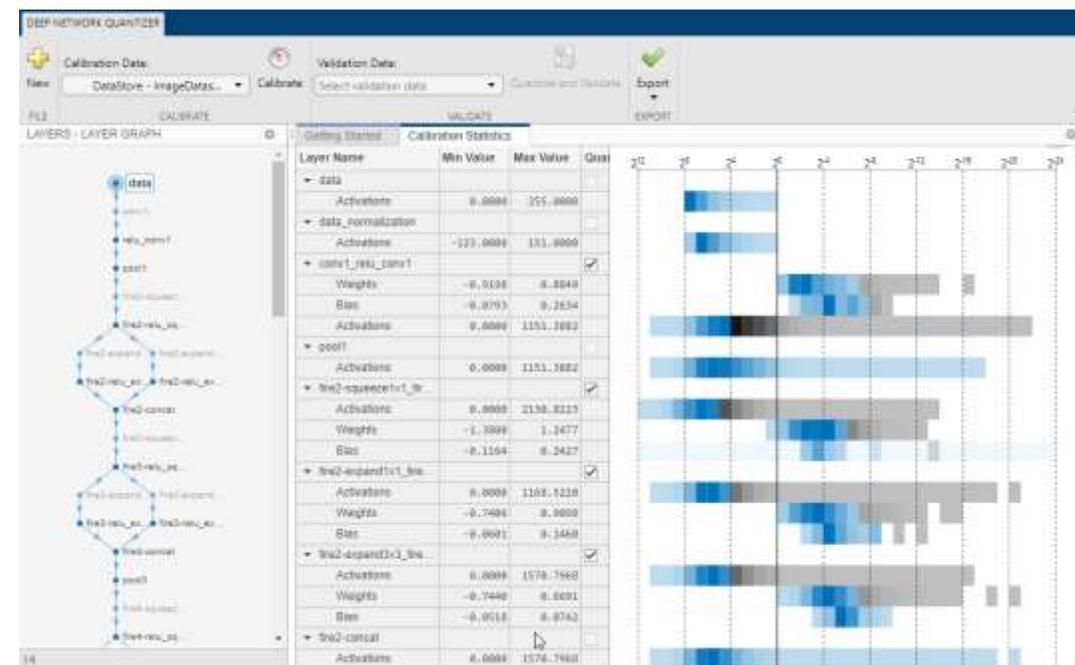
ネットワークの量子化と圧縮を行い、低消費電力のマイクロコントローラーやFPGAに展開

求められる精度を満たすために、適切な量子化手法を選択し、検証

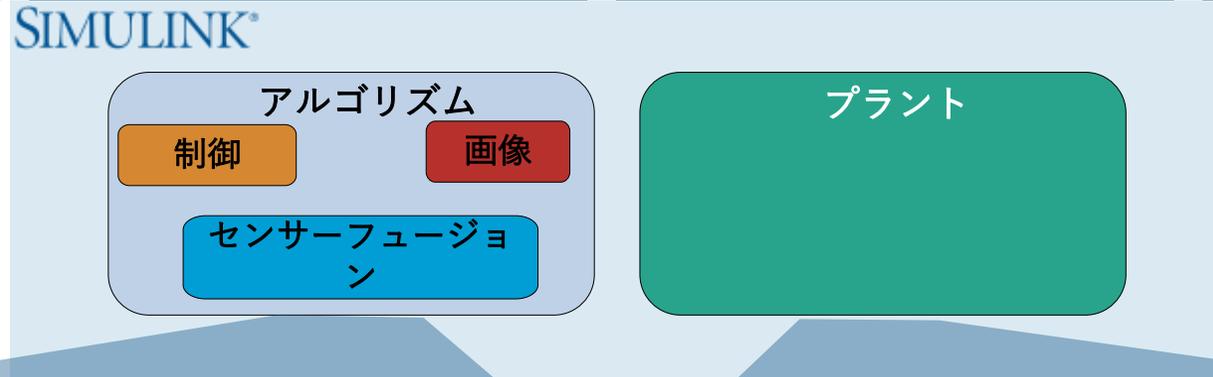
Deep Network Quantizer App

R2020a

- 畳み込みレイヤーのダイナミックレンジを可視化
- 量子化するネットワークレイヤーを個別に選択
- パフォーマンスの評価
- GPUコードを生成して展開 (GPU Coder)



MBDでのAI活用法：シミュレーションと実装フロー



アルゴリズムにAIを利用

- 判断（コントローラ）にAIを利用
- ラピッドプロトタイピング(RCP)
- CPU/GPU/ECU/FPGAへの展開

プラントモデルにAIを利用

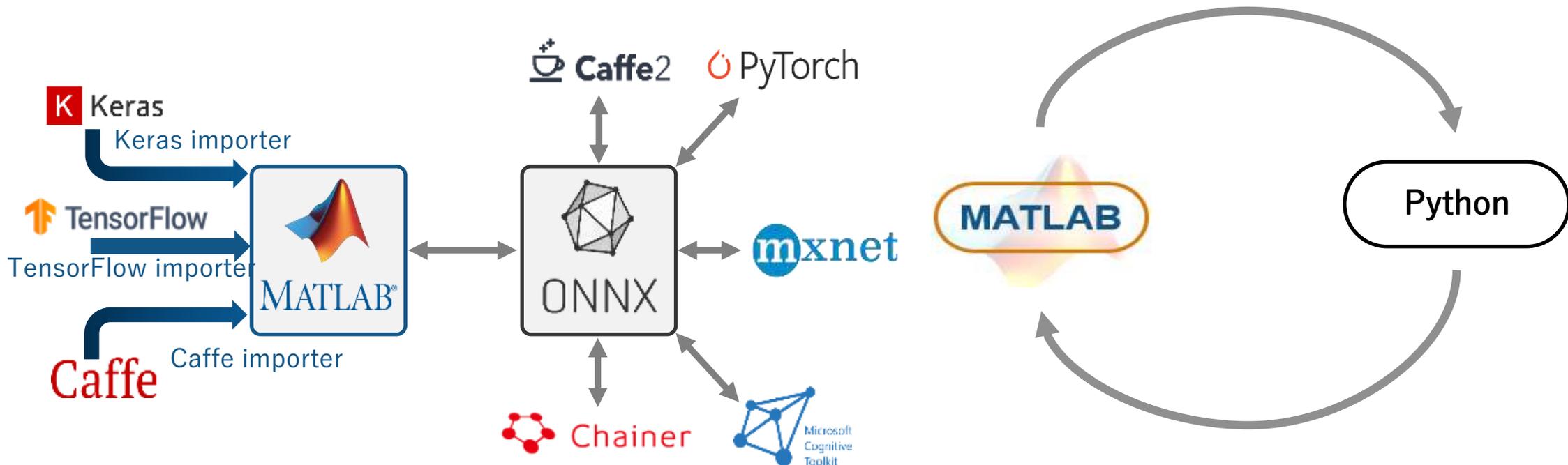
- 複雑な状態、応答の表現にAIを利用
- 高精度シミュレーション(3D-CAE 等)の高速化
- Hardware In the Loop (HILS)による検証
- コンポーネントの共有

AIの活用のためのシームレスな開発フローをサポート

アジェンダ

- MBD開発でのAIモデル利用ケース
- エンジンモデルを題材にしたAI活用の3ステップ
 - AIモデルの構築
 - モデルドリブン+AIの統合シミュレーション
 - RCP/HILSのためのコード生成
- TensorFlow/PyTorchモデルの活用
- まとめ

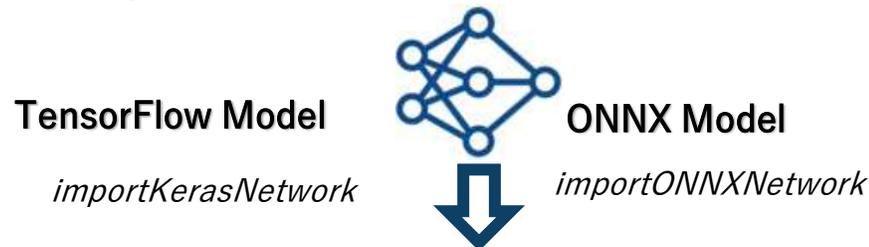
TensorFlow/PyTorchとMATLABの相互運用性



モデルインポート/エクスポート

MATLAB-Pythonの同時実行

コード生成、可視化とデバッグ、再トレーニング、システム統合に MATLAB & Simulinkを使用



MATLAB DNNモデル

Simulinkでの
システム統合

コード生成
配布・展開

可視化とデバッグ

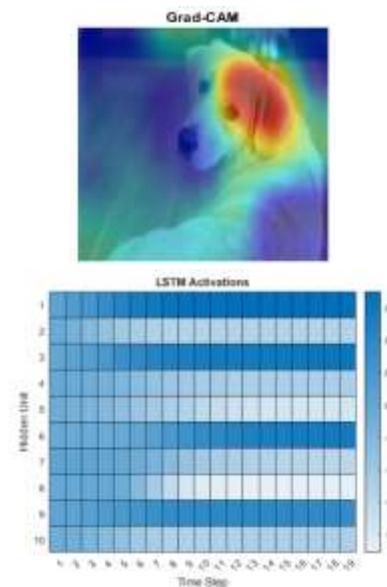
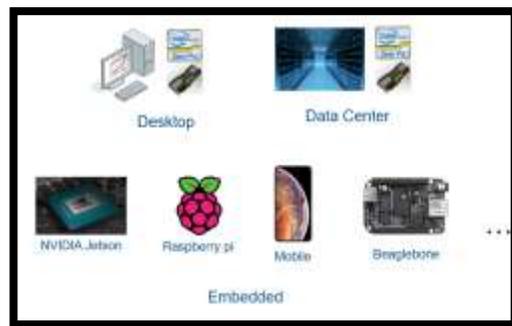
ネットワーク解析
再学習



Reinforcement Learning

Automated Driving

Control Systems



Automatic Differentiation

Custom Training Loop

Weight Sharing

三井化学、TensorFlowとMATLABを使用してAI・自動化技術を現場展開

課題

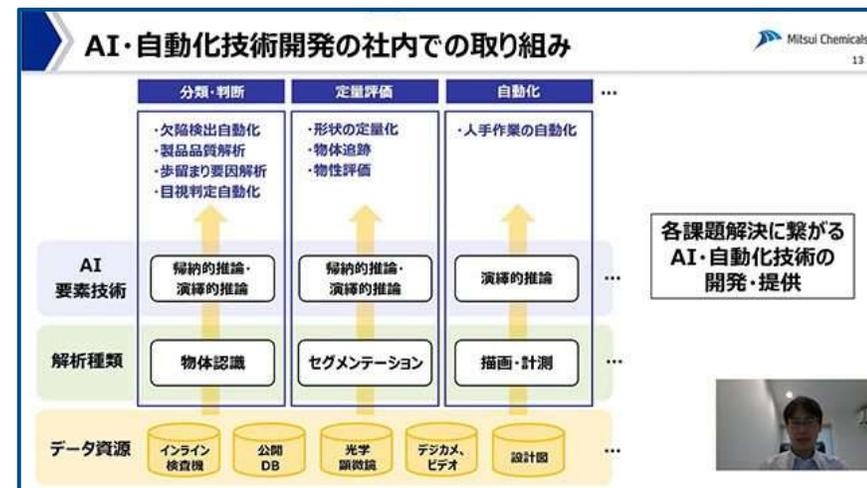
シート状製品の外観検査の自動化における開発モデルのユーザー利便性およびメンテナンス性

ソリューション

TensorFlow-Keras向けインポーターを使用してモデルをMATLABに取り込み、GUIアプリケーションとして配布して運用

結果

- 外観検査時間の8割削減
- 他のフレームワークで学習したモデルの有効活用
- 誰もが利用できるGUIアプリケーションとして展開



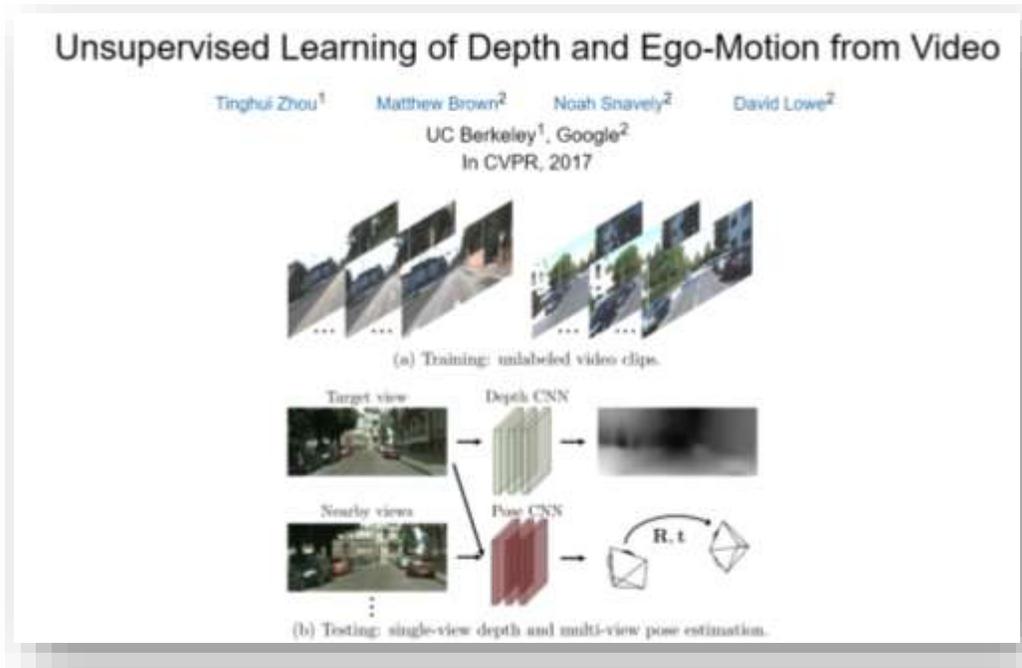
Python(TensorFlow-Keras)を使用したモデル開発と、MATLABを使用したモデルの効率的な現場実装。

“MATLABを活用することで、現場実装上の課題が解決され、工数を掛けることなく、確度の高い開発に繋がれるようになりました。”

- 前川 真太郎氏, 三井化学株式会社

例:単眼カメラからの姿勢・深度推定

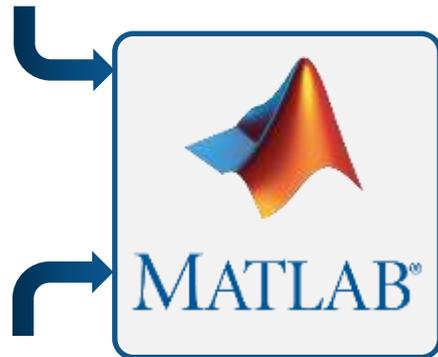
- 深層学習に基づく単眼画像シーケンスを用いた奥行き・姿勢の推定
 - TensorFlow ベースのコードが公開. (<https://github.com/tinghuiz/sfMLearner>)
- 適用アプリケーション：自動運転、移動ロボットなど



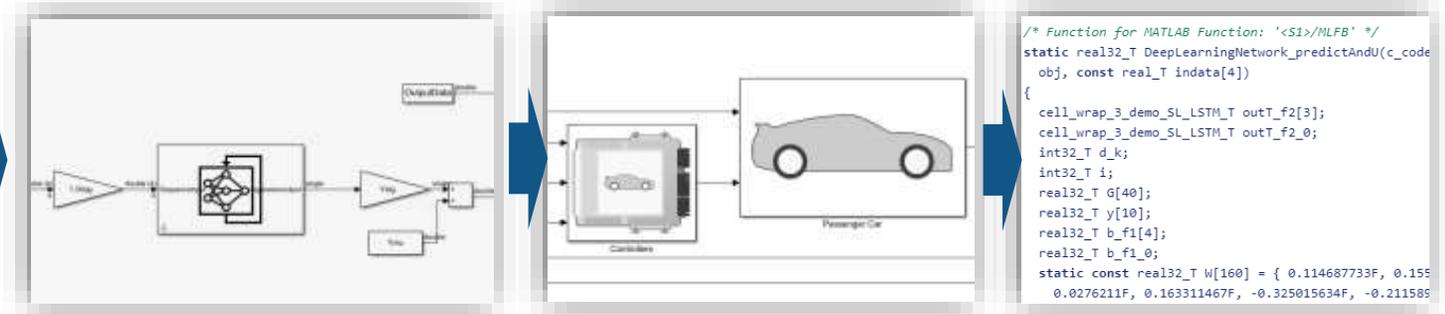
TensorFlow & PyTorchからも 統合シミュレーション & 実装をサポート



TensorFlow

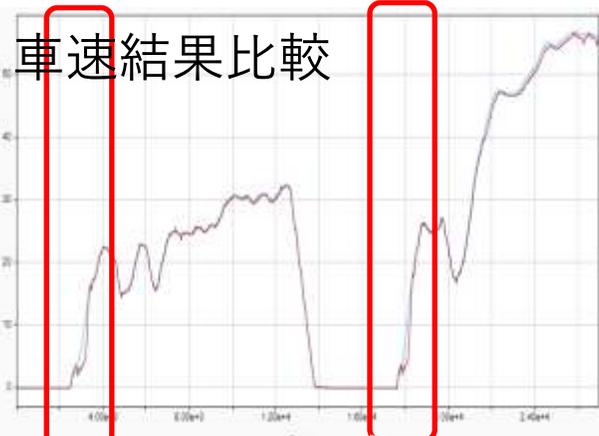


PyTorch



- RCP
- HILS
- 量産向けコード生成

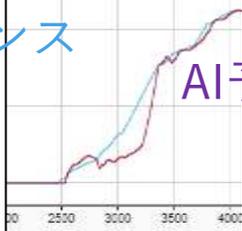
AI適用におけるチャレンジ



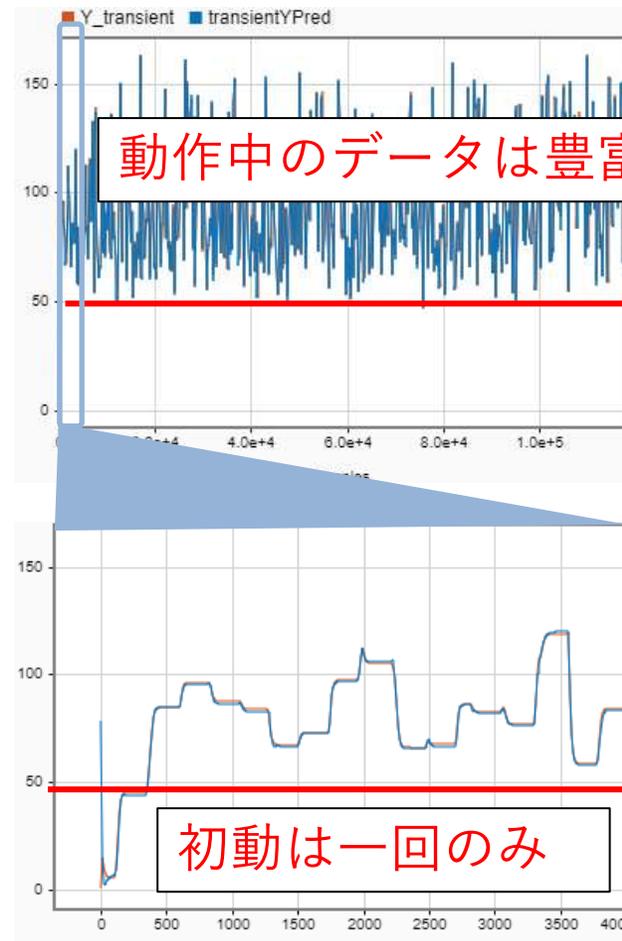
車速0からの立ち上がり
ずれ大

リファレンス

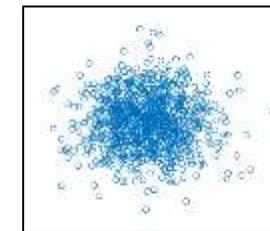
AI予測+計算



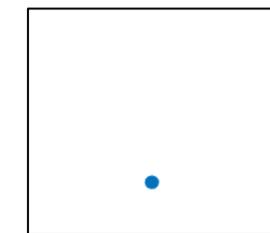
LSTMの学習用 トルクデータ



動作中の現象



車速0付近の現象

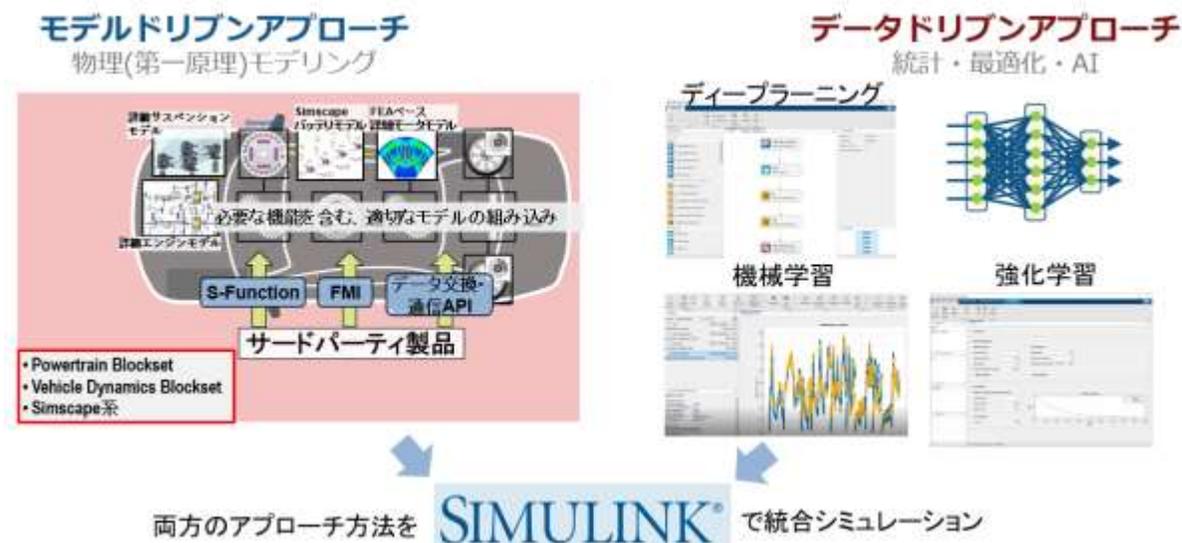


機械学習・ディープラーニングは統計的なアプローチ

- 外挿は精度に期待しづらい
- 「起こり得る動作を網羅するデータ」で学習させる (データ量ではない)

まとめ

- 実業務の課題解決にAI適用が進み、非線形性の表現、シミュレーション高速化を目的に、MBDのフローでもく使われ始めています
- モデル構築や取り込み、モデルドリブンとAIを統合したシミュレーション、HILS/RCP向けコード生成をサポート
- ドメイン知識とデータ解析テクニック両方を兼ね備えたサービスを提供



MATLAB EXPO 2021

Thank you



© 2021 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.