【三菱電機シーケンサ×MATLAB/Simulink】
人手作業の自動化/自律化を効率的に実現する
～ロバストなアルゴリズム・システム開発とは～

MathWorks Japan
アプリケーションエンジニアリング部

×

Agenda

1.自動化/自律化におけるトレンドと課題

2.ロバストなシステム開発のためのワークフロー

- MATLAB/Simulinkとは？
- 三菱電機シーケンサとの連携機能
- 両社の強みを生かしたロバストな開発フロー

3.人手作業の自律化を想定したシステム開発例

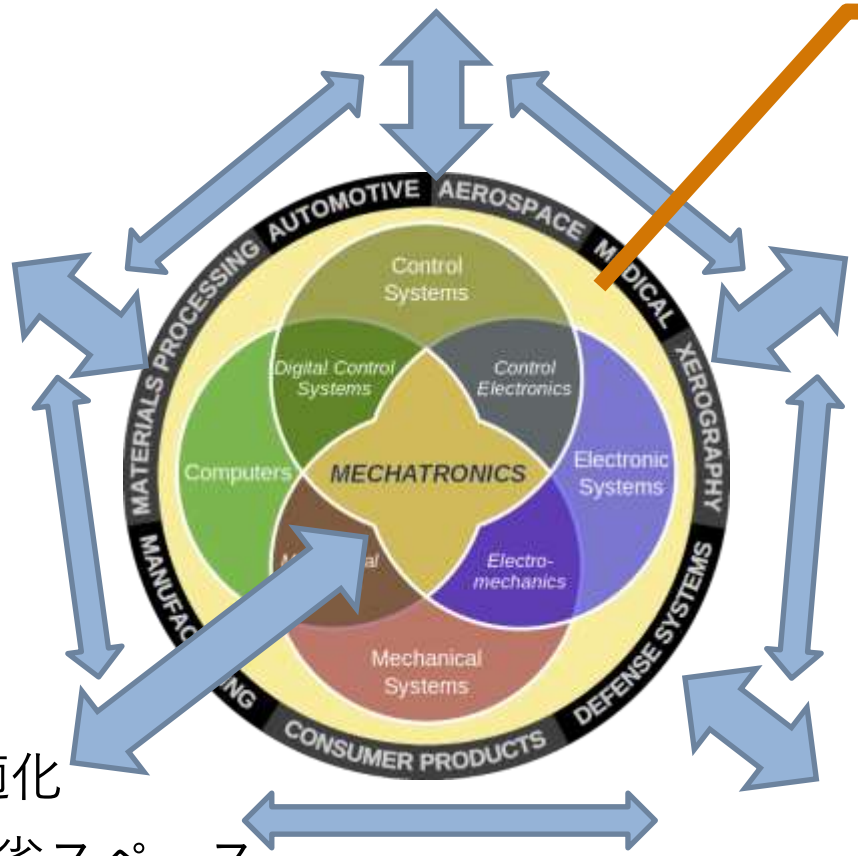
トレンド: 高度・デジタル技術への期待

複雑・高度化したシステム開発

複合領域にまたがるシステムに更に新たな技術を付加

AI

画像認識



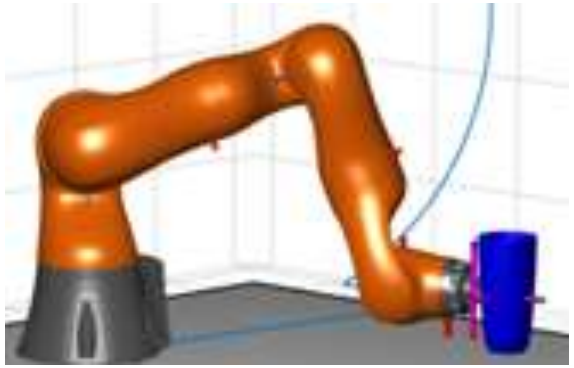
最適化

協調作業・省スペース

ヴァーチャル
コミッショニング

予知保全

デジタル化＝モデルの活用



デジタルツイン

自動化⇒自律化

自動化/自律化の例題：ピッキングタスク

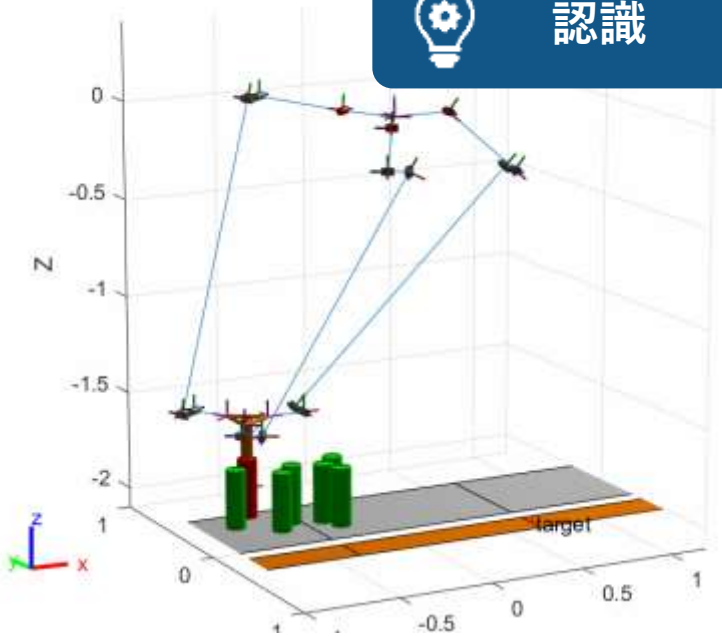
デルタロボットの軌道計画

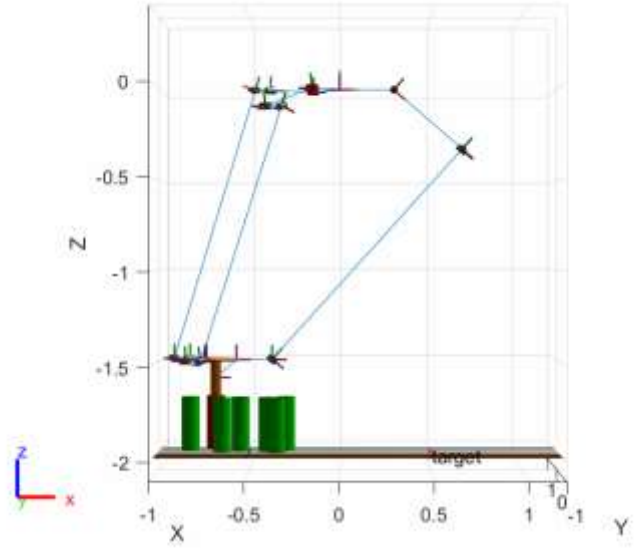
ロボット&環境

KRONES

センシング

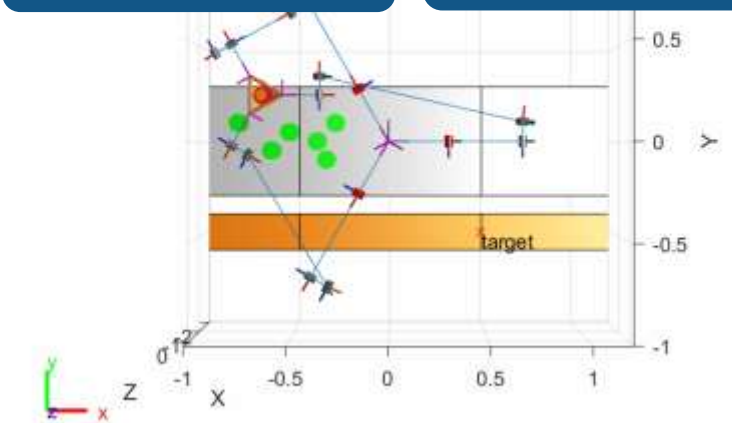
認識





計画 & 判断

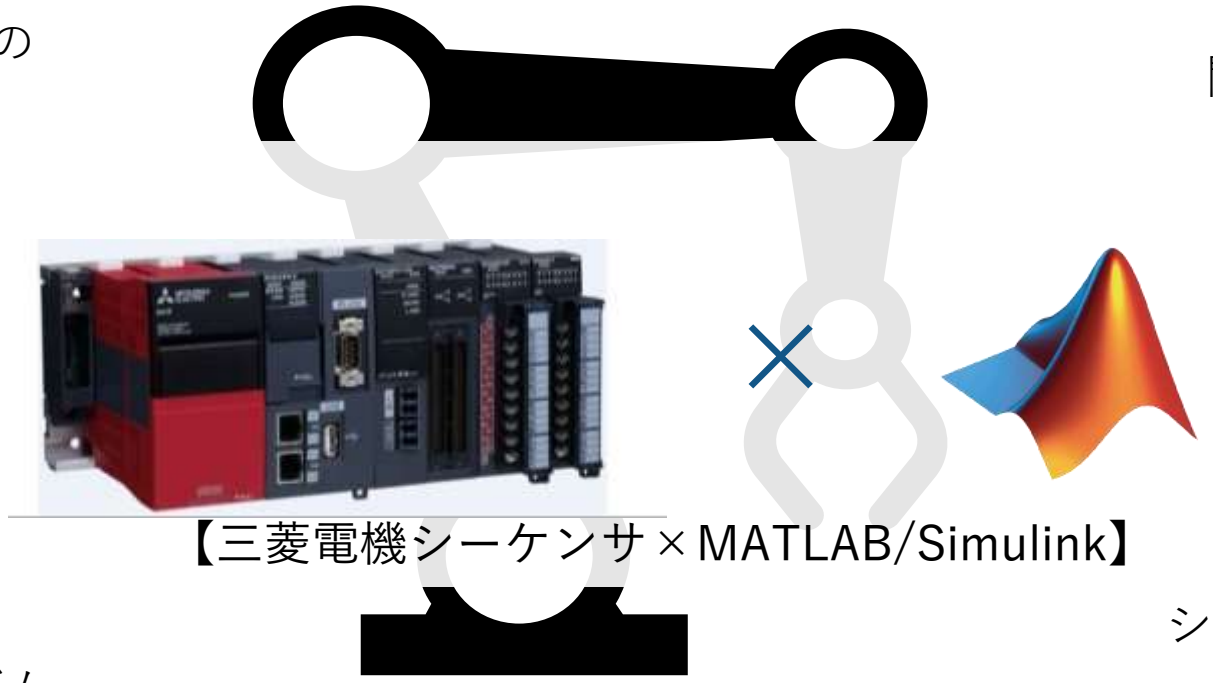
制御



自律システム開発における課題とは？

複合領域の
専門性

開発環境



【三菱電機シーケンサ×MATLAB/Simulink】

高度な
アルゴリズム

システムの
安全性

Agenda

1.自動化/自律化におけるトレンドと課題

2.ロバストなシステム開発のためのワークフロー

- MATLAB/Simulinkとは？
- 三菱電機シーケンサとの連携機能
- 両社の強みを生かしたロバストな開発フロー

3.人手作業の自律化を想定したシステム開発例

MATLABが研究開発を加速

アクセス

各種データソース

専用ハードウェア

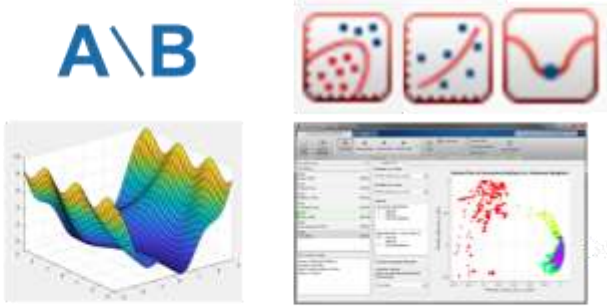
ライブラリ活用

C/Fortran/Python/AI/CAE

研究開発

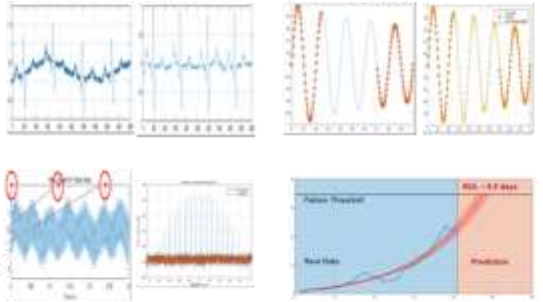
数値解析

簡潔な表現 / 統計・最適化 / 開発環境 / GUIアプリ



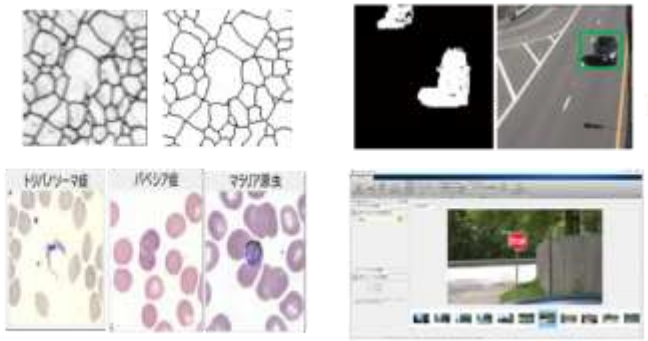
信号・時系列データへの拡張

フィルタ / 補完 / 変換 / 時系列予測



画像処理・深層学習

フィルタ・変換 / 認識・判定 / 分類 / GUIアプリ



その他

性能・並列化 / 容易 / ワンパッケージ / サポート

実装

クラウド利用

ブラウザ表示 / IaaS

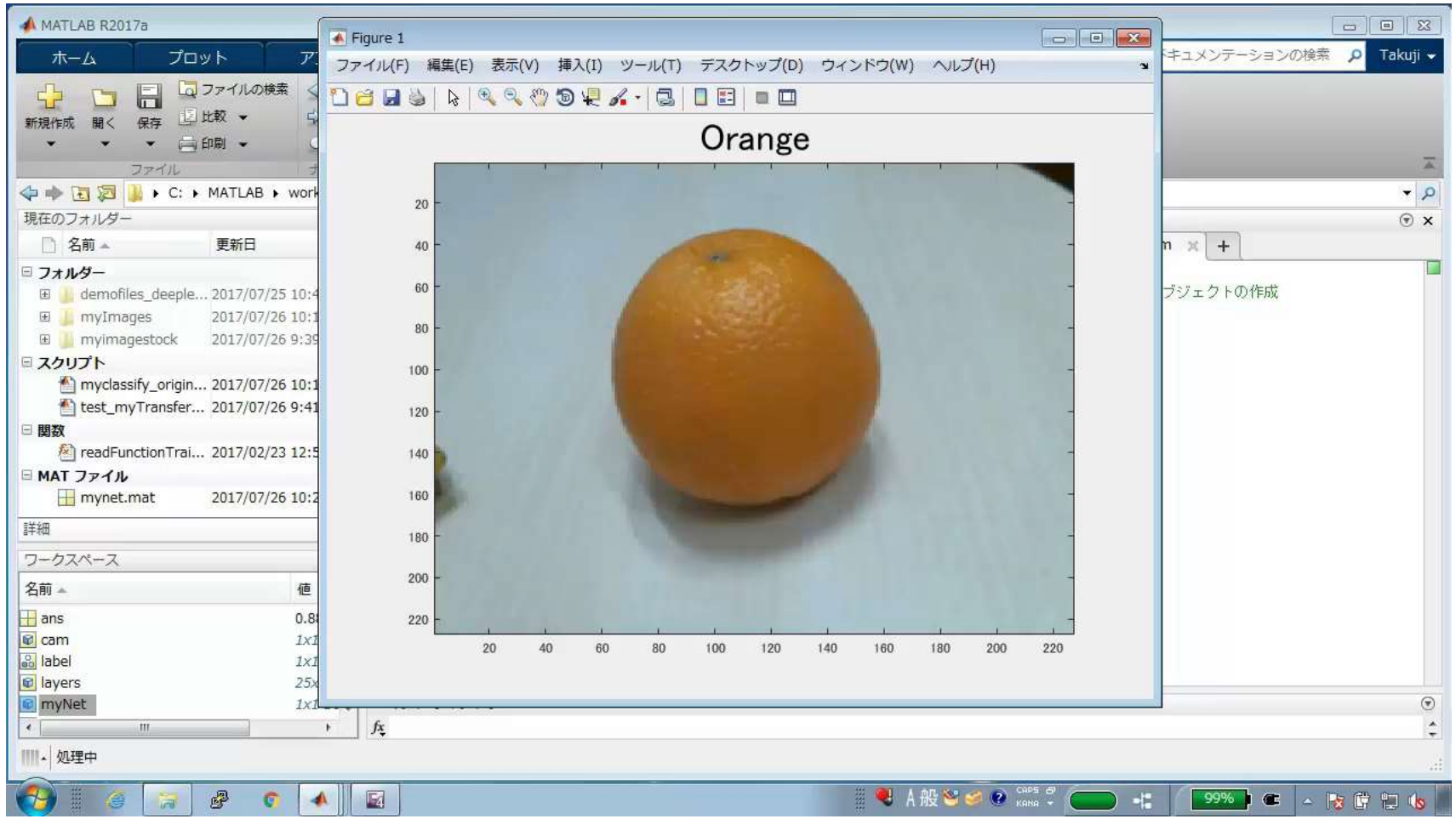
MATLAB なしのPC

組込実装

GPU / FPGA / マイコン

ディープラーニングによる物体認識

ディープラーニング：10行でできる転移学習 ～画像分類タスクに挑戦～



学習した種類：
-オレンジ
-みかん
-グレープフルーツ(ルビー)
-グレープフルーツ(ホワイト)
-レモン

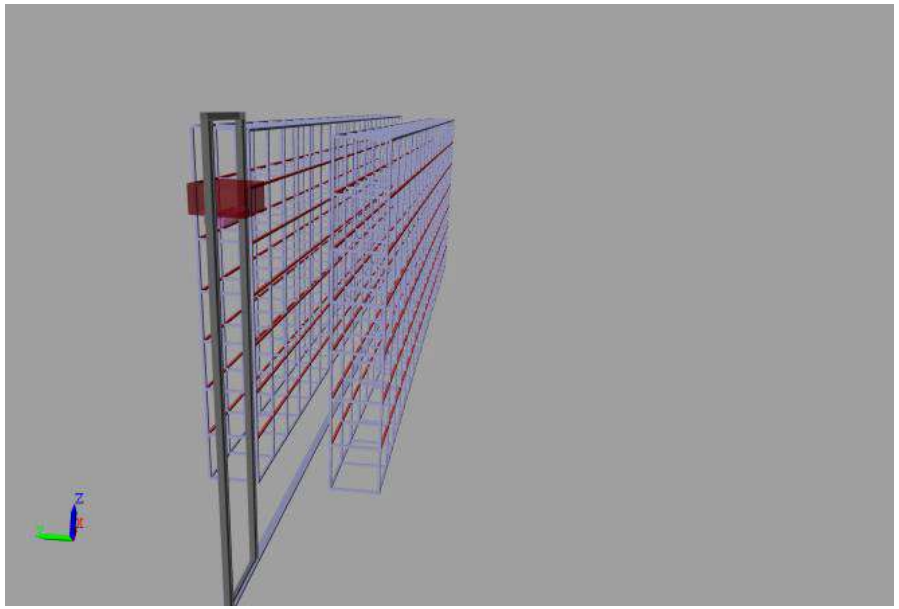
学習画像数：各 20 枚

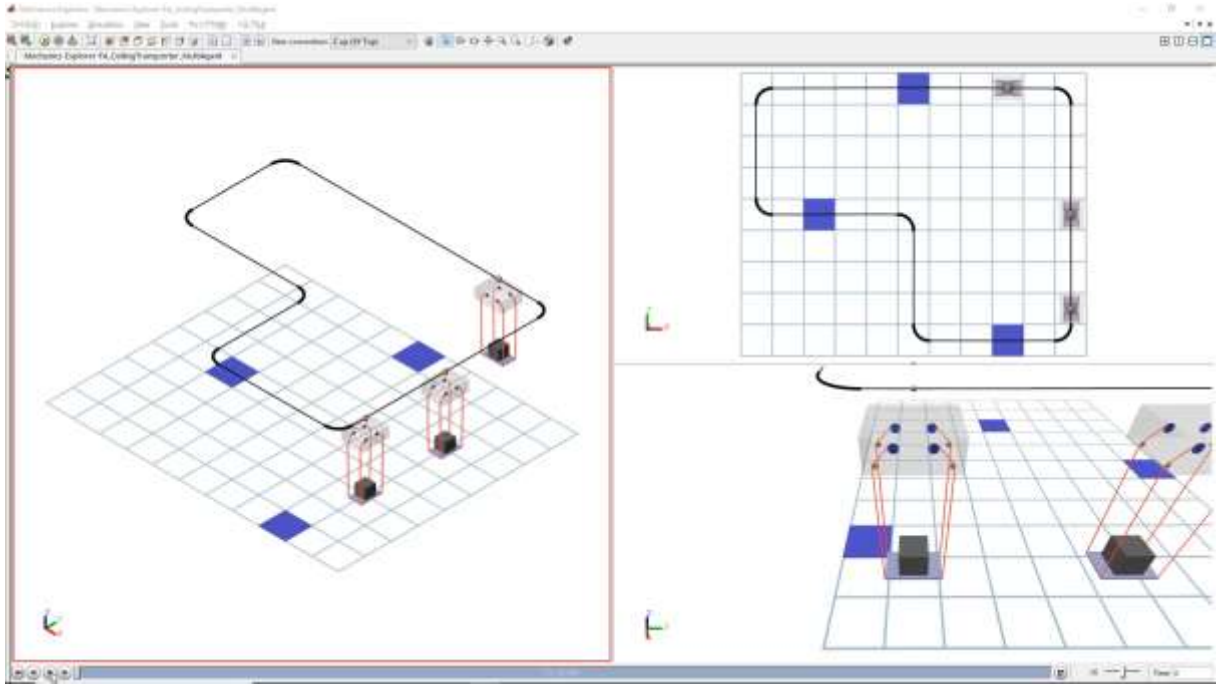
●要件を満たすPC&MATLAB環境
●学習済みAlexnet
●画像セット
で10行のコーディングで始められます

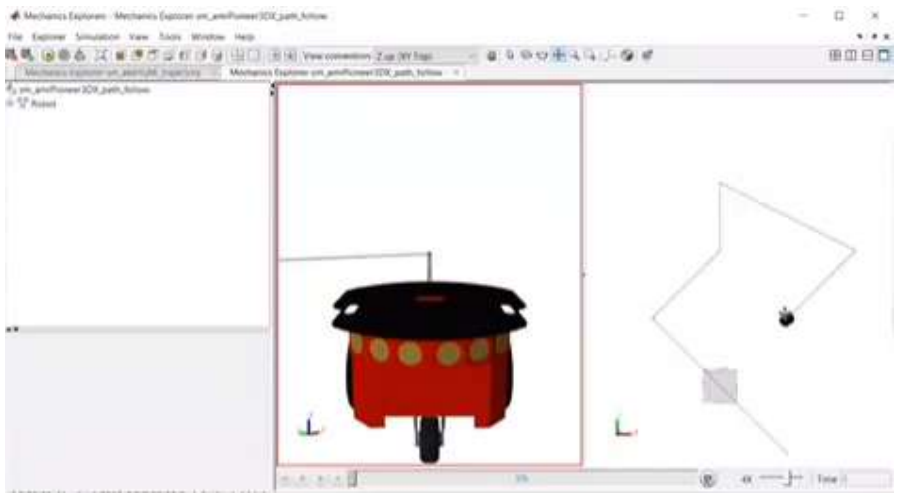
https://www.mathworks.com/videos/deep-learning-transfer-learning-in-10-lines-of-matlab-code-1503069304524.html

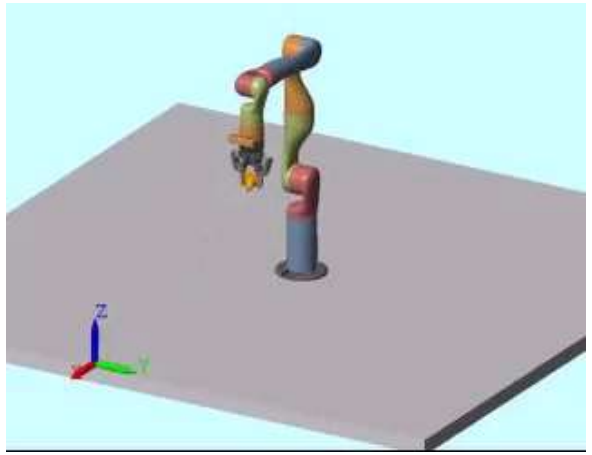
Simulinkとは？

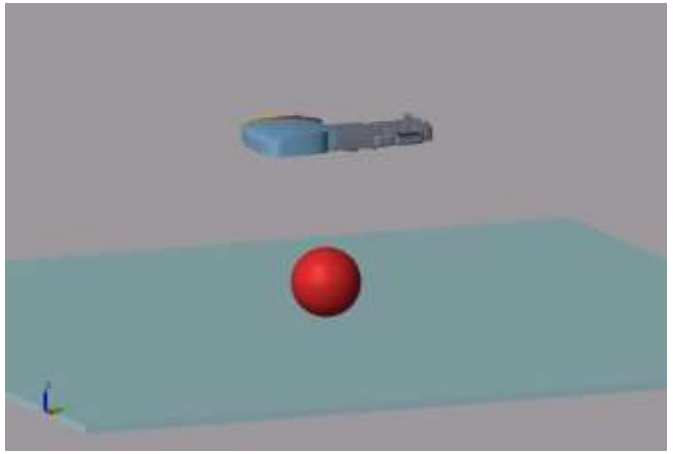
関連サンプルデモ







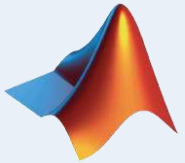




連携機能を活用した自律システムのロバストな開発フロー

STEP1.机上検討

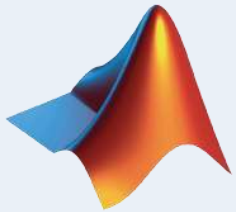
MATLAB/Simulinkを活用して
アルゴリズムの選定や初期の検証を加速



STEP2.プロトタイプ

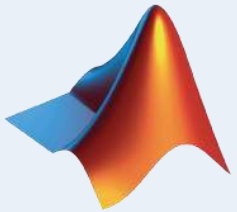
三菱電機シーケンサ×MATLAB/Simulink
ロバストな試験環境を早期に構築

Ethernet通信等



STEP3. 実運用

コード生成技術によりアルゴリズムをシームレスに展開



コード生成
実装

連携機能を活用した自律システムのロバストな開発フロー

STEP1.机上検討

MATLAB/Simulinkを活用して
アルゴリズムの選定や初期の検証を加速

STEP2.プロトタイプ

三菱電機シーケンサ×MATLAB/Simulink
ロバストな試験環境を早期に構築

Ethernet通信等

STEP3. 実運用

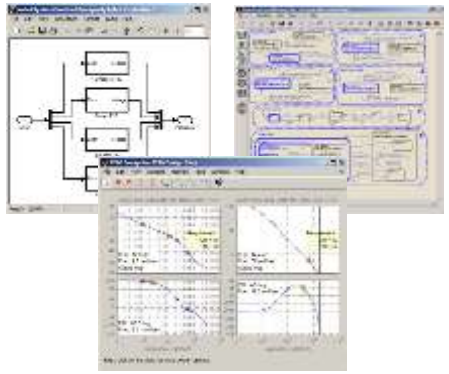
コード生成技術によりアルゴリズムをシームレスに展開

コード生成
実装

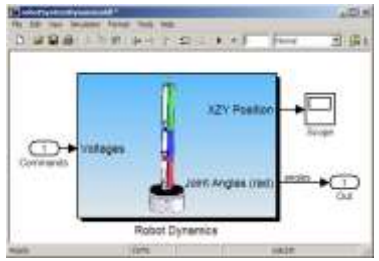
STEP1.机上検討

MATLAB/Simulinkを活用してアルゴリズムの選定や初期の検証を加速

仮想環境



ソフトウェアモデル



ハードウェアモデル

実環境

制御装置

制御対象

<u>高度なアルゴリズムを早期に”机上検証”</u>

- ✓ ライブラリや例題の活用
- ✓ UI・可読性の高い統合開発環境

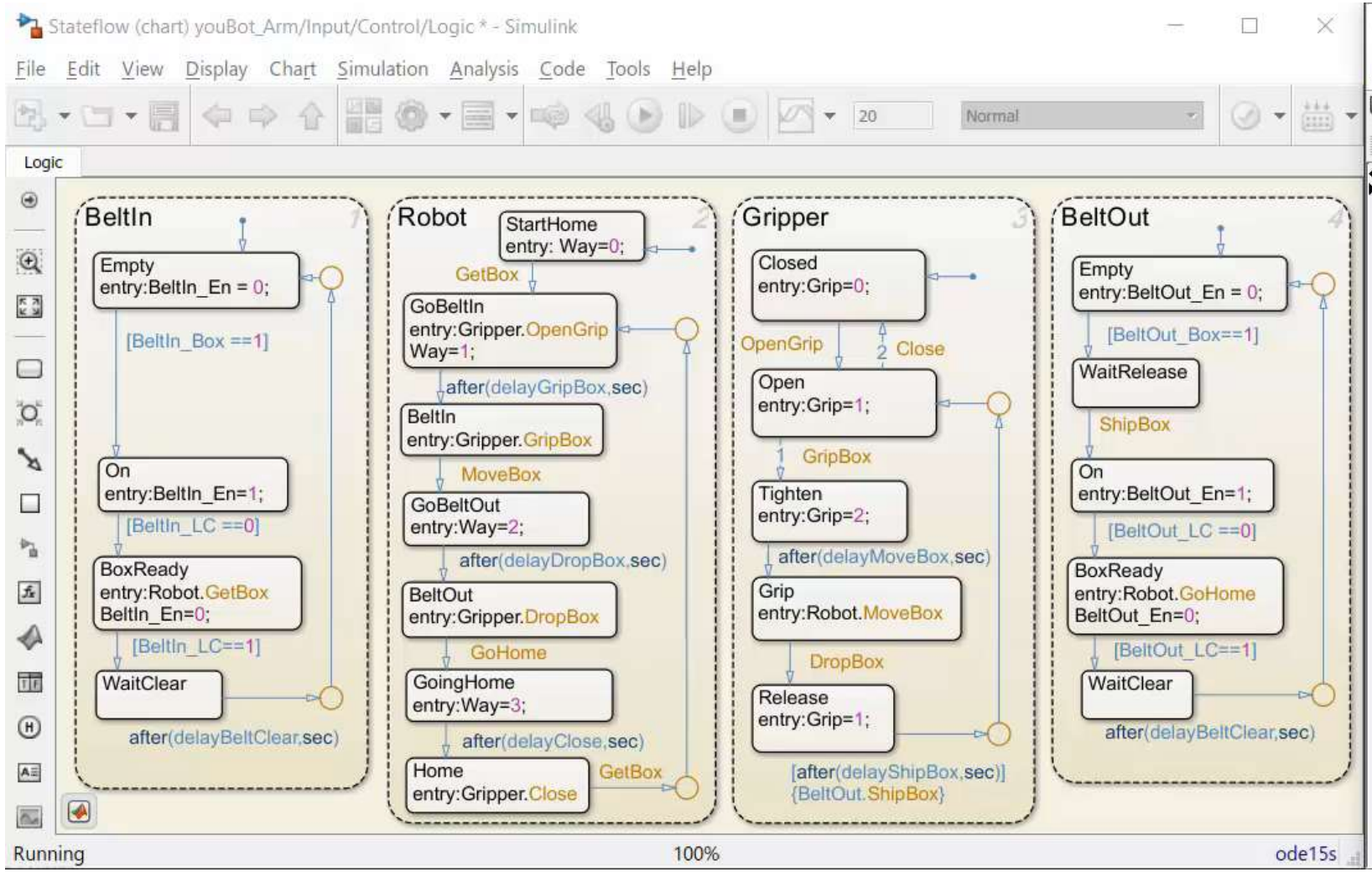
<u>シミュレーションによる”コスト削減”</u>

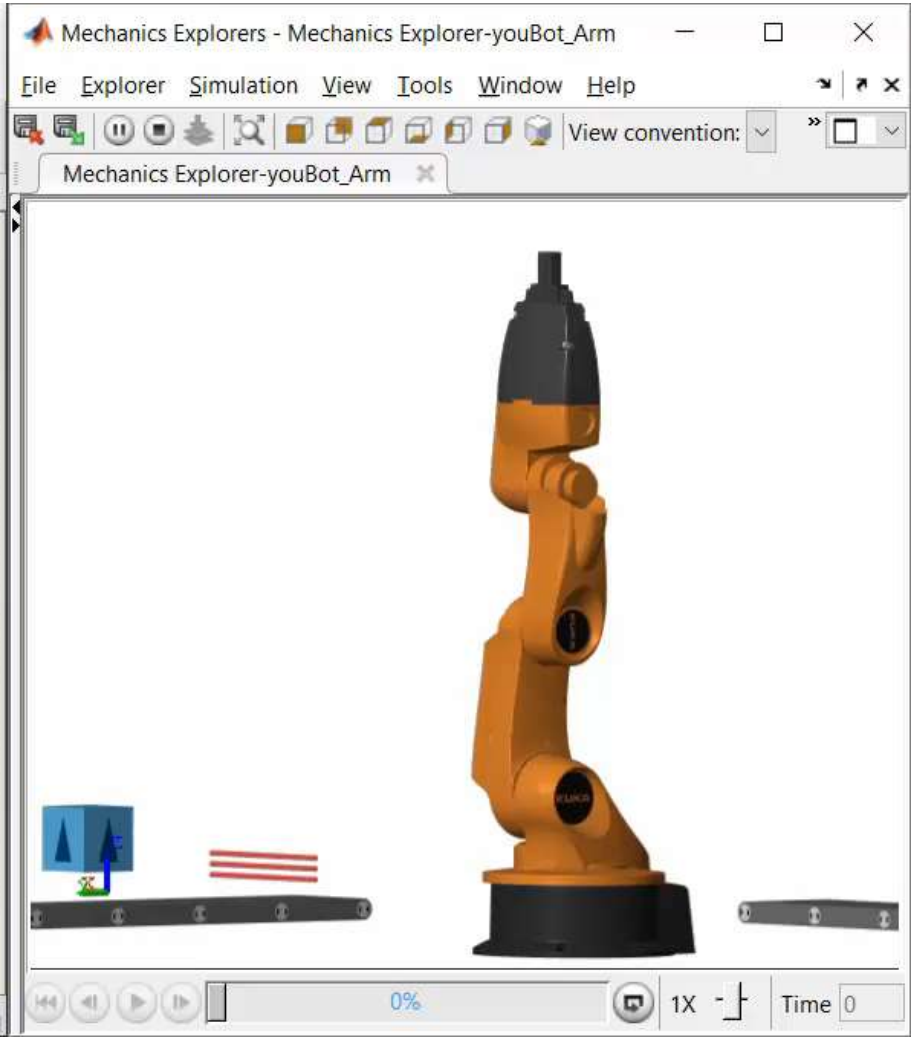
- ✓ 試作・実験回数を削減
- ✓ 大規模試験の仮想化
- ✓ メカ・エレキ・制御含めた
  トレードオフ設計を促進

⇒MATLAB/Simulinkの各種機能が
シミュレーションの活用を支援・促進

STEP1.机上検討

MATLAB/Simulinkを活用してアルゴリズムの選定や初期の検証を加速





連携機能を活用した自律システムのロバストな開発フロー

STEP1.机上検討

MATLAB/Simulinkを活用して
アルゴリズムの選定や初期の検証を加速

STEP2.プロトタイプ

三菱電機シーケンサ×MATLAB/Simulink
ロバストな試験環境を早期に構築

Ethernet通信等

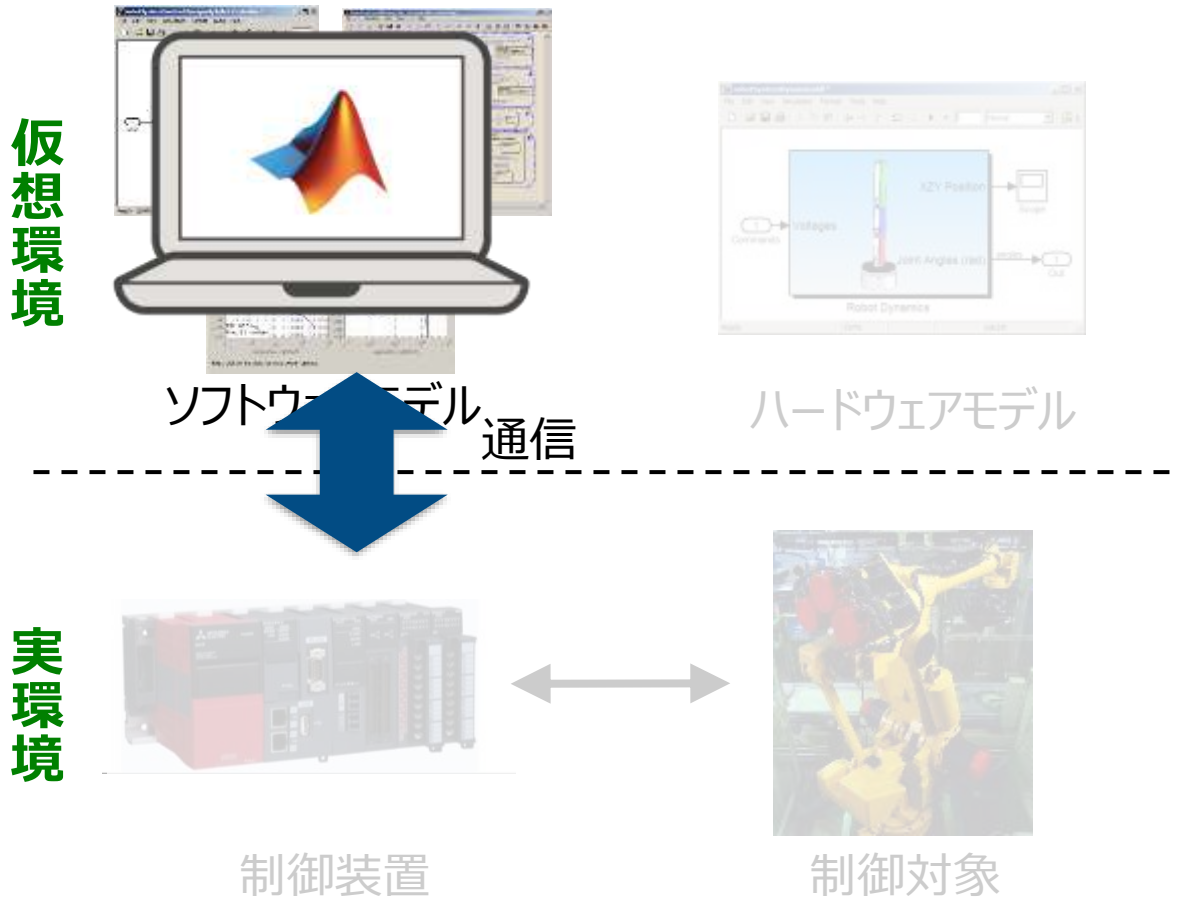
STEP3. 実運用

コード生成技術によりアルゴリズムをシームレスに展開

コード生成
実装

STEP2.プロトタイプ

1)MATLAB/Simulinkとシーケンサの通信ユニットを介した連携



高度なアルゴリズムを早期に”実機検証”

- ✓ PC上のMATLABを直接接続
- ✓ リアルタイム専用ハード(Speedgoat)の接続にも対応可能

三菱電機シーケンサの”柔軟な接続性”

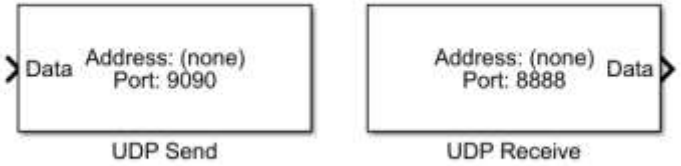
- ✓ Ethernetを始めとした柔軟な接続性
- ✓ 設備で利用されている制御装置をそのまま活用可能

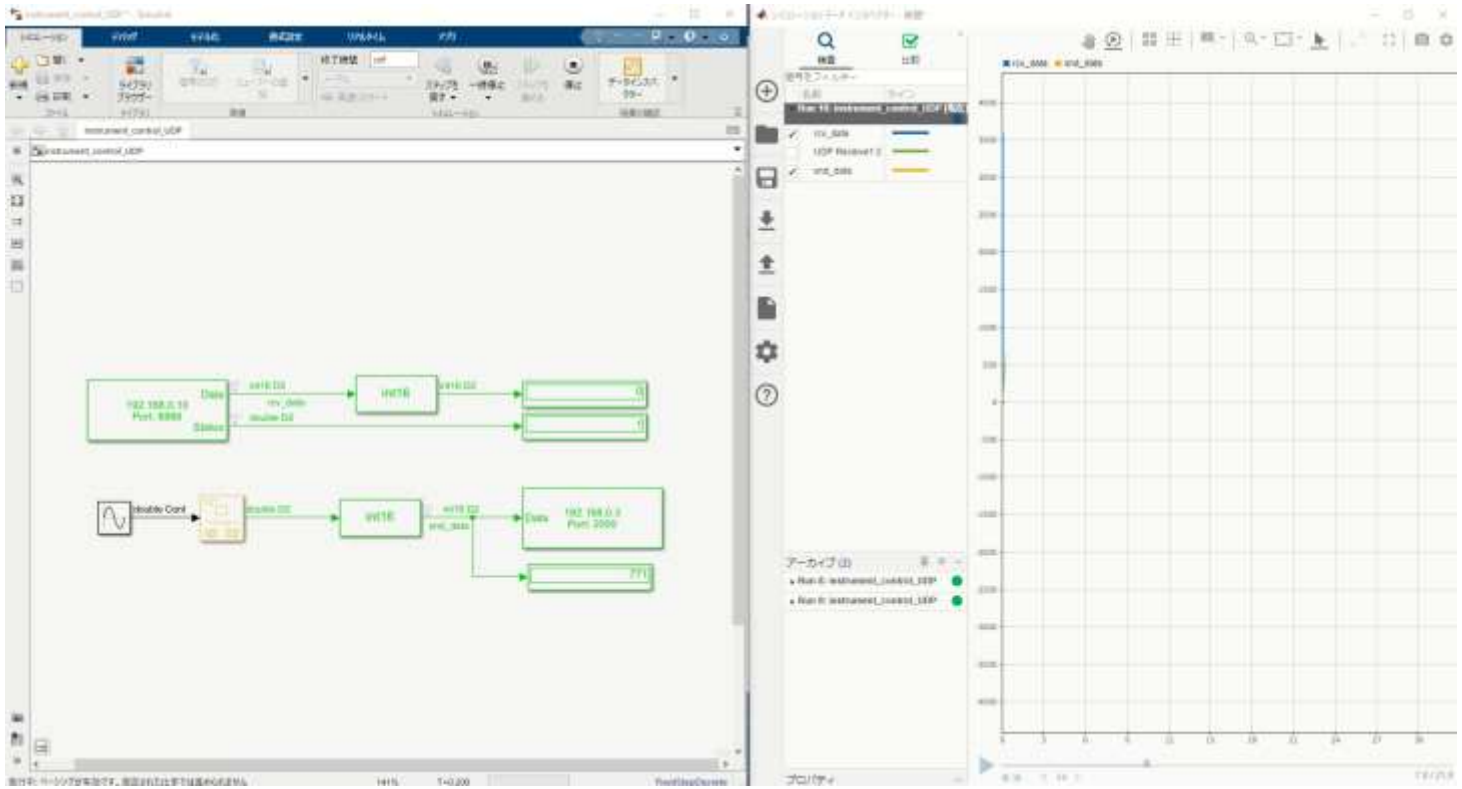
⇒MATLAB/Simulink、三菱電機シーケンサそれぞれ提供する柔軟なインターフェースにより、簡単に接続が可能。

STEP2.プロトタイプ

1)MATLAB/Simulinkとシーケンサの通信ユニットを介した連携

Ethernet (UDP)経由での
サイン波のループバック





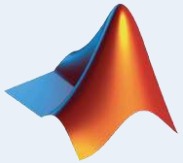
- ✓ Ethernet接続の基本的な設定だけで簡単に接続

※Ethernetユニット設定例
UDP通信(オープン待機)
ポートペアリングON
固定バッファ
無手順
生存確認なし

連携機能を活用した自律システムのロバストな開発フロー

STEP1.机上検討

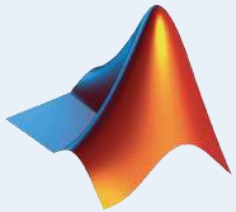
MATLAB/Simulinkを活用して
アルゴリズムの選定や初期の検証を加速



STEP2.プロトタイプ

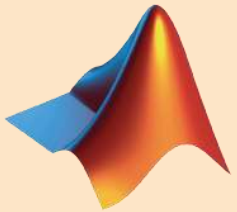
三菱電機シーケンサ×MATLAB/Simulink
ロバストな試験環境を早期に構築

Ethernet通信等



STEP3. 実運用

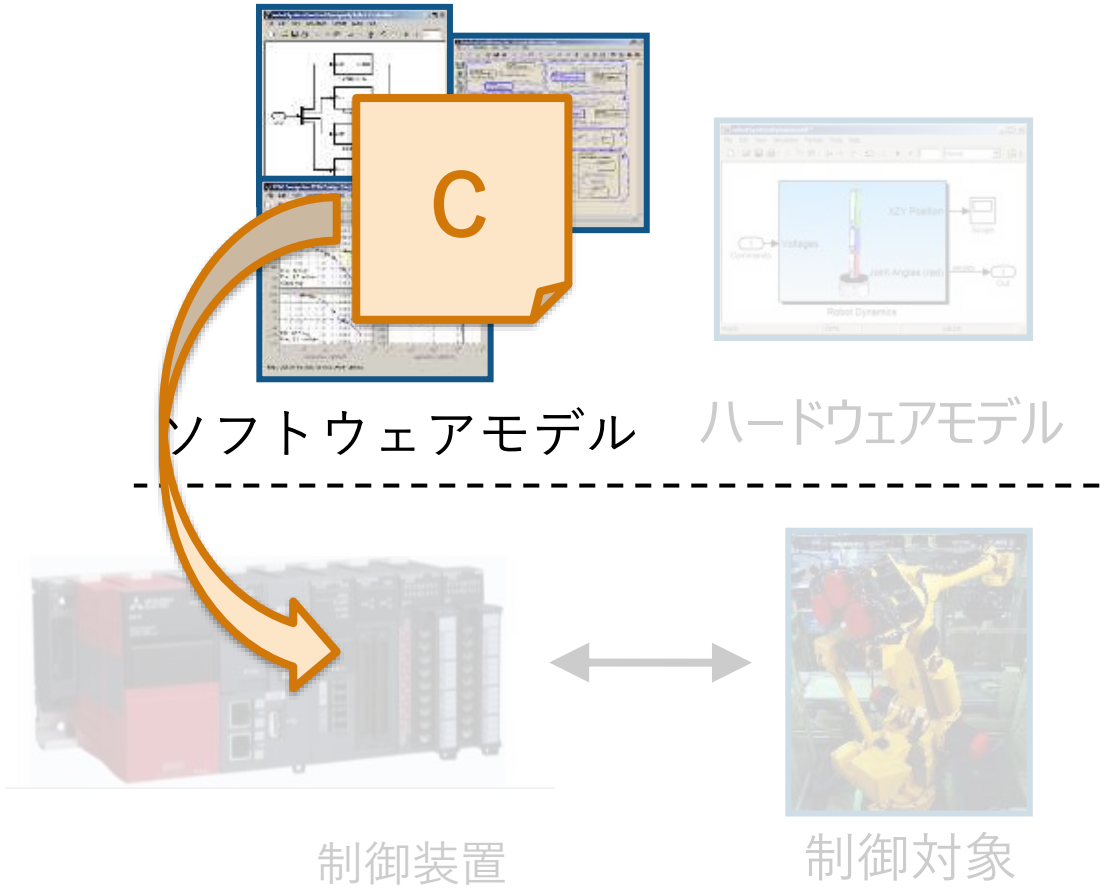
コード生成技術によりアルゴリズムをシームレスに展開



コード生成
実装

STEP2.プロトタイプ ⇒ STEP3.実運用

2)MATLAB/SimulinkのCコード生成機能を通じた C言語対応ユニットへのプログラムの実装



MATLAB/Simulinkからの自動Cコード生成

- ✓ STEP1,2で検証したアルゴリズムをシームレスに活用
- ✓ 自動コード生成機能を活用して、コーディングの手間やミスを最小化

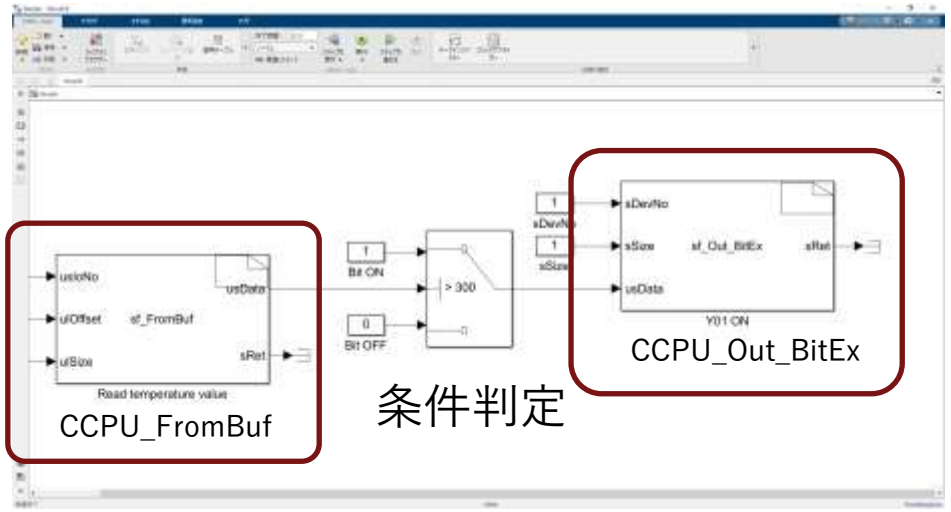
三菱電機シーケンサC言語対応ユニットの活用

- ✓ Simulinkブロックから生成されたC言語を直接利用可能

⇒MATLAB/SimulinkのCコード生成機能
三菱電機シーケンサのC言語対応ユニット
の協調によりシームレスな実装フローを実現

STEP2.プロトタイプ　⇒　STEP3. 実運用

2) MATLAB/SimulinkのCコード生成機能を通じた C言語対応ユニットへのプログラムの実装



C
自動生成

三菱電機シーケンサ専用命令を含む状態で自動生成

変数宣言

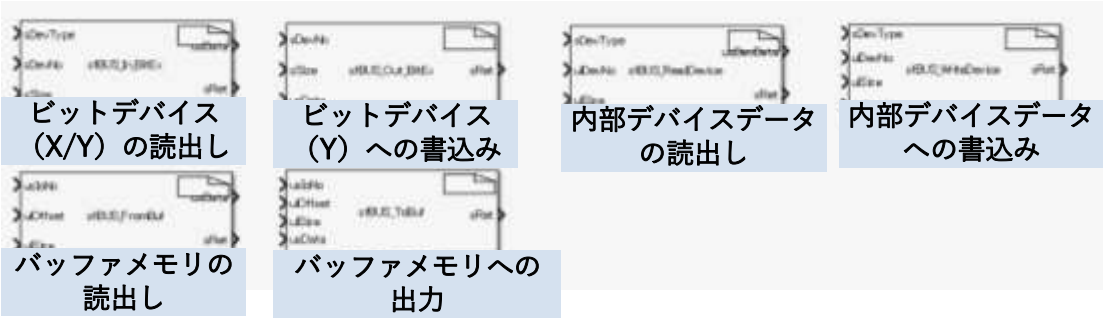
CCPU_FromBuf

条件判定

CCPU_Out_BitEx

```
#include "Sa
#include "Sa

/* Block sig
B_Sample_T Sample_B;

/* Real-time model */
static RT_MODEL_Sample_T Sample_M_;
RT_MODEL_Sample_T *const Sample_M = &Sample_M_;

/* Model step function */
void Sample_step(void)
{
  /* S-Function (sf_FromBuf): '<Root>/Read temperature value' incorporates:
   *  Constant: '<Root>/ulOffset'
   *  Constant: '<Root>/ulSize'
   *  Constant: '<Root>/usIoNo'
   */
  sf_FromBuf_Outputs_wrapper(&Sample_P.usIoNo_Value, &Sample_P.ulOffset_Value,
    &Sample_P.ulSize_Value, &Sample_B.Readtemperaturevalue_o1,
    &Sample_B.Readtemperaturevalue_o2);

  /* Switch: '<Root>/Switch' */
  if (Sample_B.Readtemperaturevalue_o1 > Sample_P.Switch_Threshold) {
    /* Switch: '<Root>/Switch' incorporates:
     *  Constant: '<Root>/Bit ON'
     */
    Sample_B.Switch = Sample_P.BitON_Value;
  } else {
    /* Switch: '<Root>/Switch' incorporates:
     *  Constant: '<Root>/Bit OFF'
     */
    Sample_B.Switch = Sample_P.BitOFF_Value;
  }

  /* End of Switch: '<Root>/Switch' */

  /* S-Function (sf_Out_BitEx): '<Root>/Y01 ON' incorporates:
   *  Constant: '<Root>/sDevNo'
   *  Constant: '<Root>/sSize'
   */
  sf_Out_BitEx_Outputs_wrapper(&Sample_P.sDevNo_Value, &Sample_P.sSize_Value,
    &Sample_B.Switch, &Sample_B.Y01ON);
```

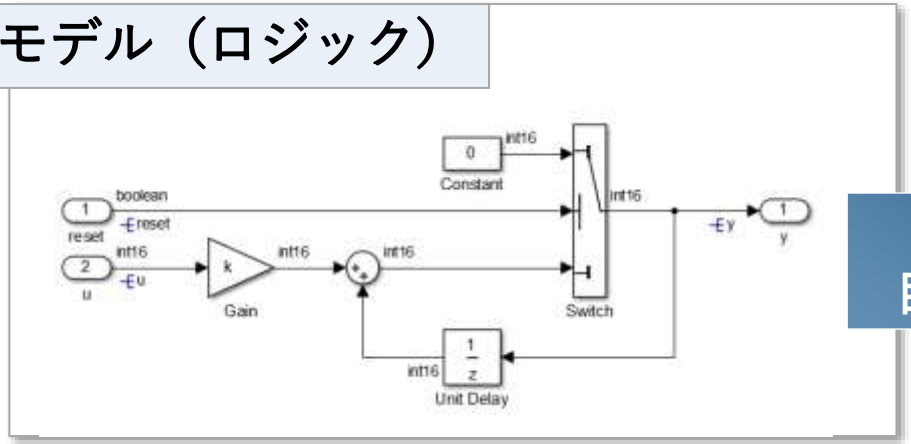
Simulink Library for　MELSEC C言語コントローラ
http://www.mitsubishielectric.co.jp/fa/products/cnt/plceng/download/library/



ご参考：Simulink PLC Coder™：
モデルからPLC用STコードを自動生成します

※ラダー図は特定のPLC IDEのみ対応

モデル（ロジック）



＋

コード仕様

- ターゲットPLC IDE
- 変数/定数属性
  - 名前、記憶クラス、データ型
- テストベンチ用コード有無

コード
自動生成

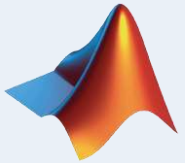
STコード

```
FUNCTION_BLOCK rst_cntr
VAR_INPUT
  ssMethodType: SINT;
  reset: BOOL;
  u: INT;
END_VAR
VAR_OUTPUT
  y: INT;
END_VAR
VAR
  UnitDelay_DSTATE: INT;
END_VAR

CASE ssMethodType OF
  0:
    UnitDelay_DSTATE := 0;
  1:
    IF reset THEN
      y := 0;
    ELSE
      y := DINT_TO_INT(INT_TO_DINT(2 * u) +
      INT_TO_DINT(UnitDelay_DSTATE));
    END_IF;
    UnitDelay_DSTATE := y;
END_CASE;
END_FUNCTION_BLOCK
```

連携機能を活用した自律システムのロバストな開発フロー

STEP1.机上検討

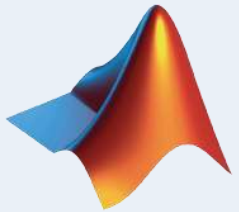
MATLAB/Simulinkを活用して
アルゴリズムの選定や初期の検証を加速



STEP2.プロトタイプ

三菱電機シーケンサ×MATLAB/Simulink
ロバストな試験環境を早期に構築

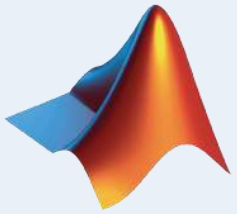
Ethernet通信等



実運用のその前に

STEP3. 実運用

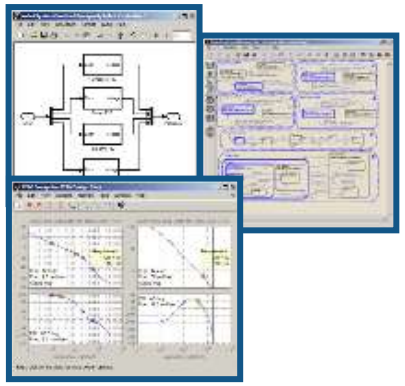
コード生成技術によりアルゴリズムをシームレスに展開

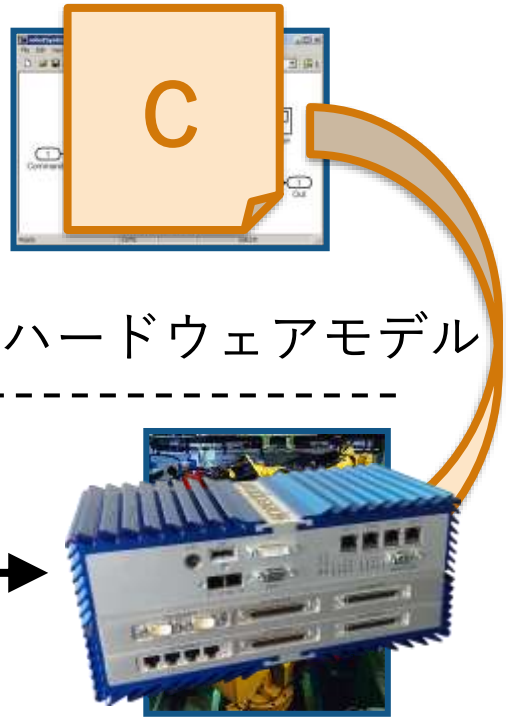


コード生成
実装

STEP2.プロトタイプ ⇒ STEP3.実運用

実機レス開発を促進するHardware In the Loop





ソフトウェアモデル　　ハードウェアモデル

制御装置　　制御対象

制御対象モデルを実機の代替として活用

- ✓ 実機模擬用ハードに制御対象モデルをコードし実装
- ✓ テスト対象はハンドコード/自動生成コードどちらもOK
- ✓ 本格運用前の仮試験や故障・異常系処理のテストに有用
- ✓ 特に大規模設備や、危険な試験にて適用効果大

⇒MATLAB/SimulinkのCコード生成機能
ハードウェアモデルを再利用して、
実機レス試験をさらに加速

Agenda

1.自動化/自律化におけるトレンドと課題

2.ロバストなシステム開発のためのワークフロー

- MATLAB/Simulinkとは？
- 三菱電機シーケンサとの連携機能
- 両社の強みを生かしたロバストな開発フロー

3.人手作業の自律化を想定したシステム開発例

連携機能を活用したロバストな開発フロー～実践～

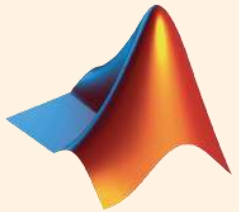
STEP1.机上検討

MATLAB/Simulinkを活用して
アルゴリズムの選定や初期の検証を加速

STEP2.プロトタイプ

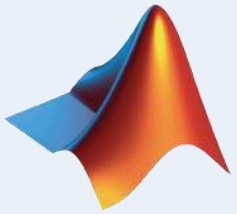
三菱電機シーケンサ×MATLAB/Simulink
ロバストな試験環境を早期に構築

Ethernet通信等



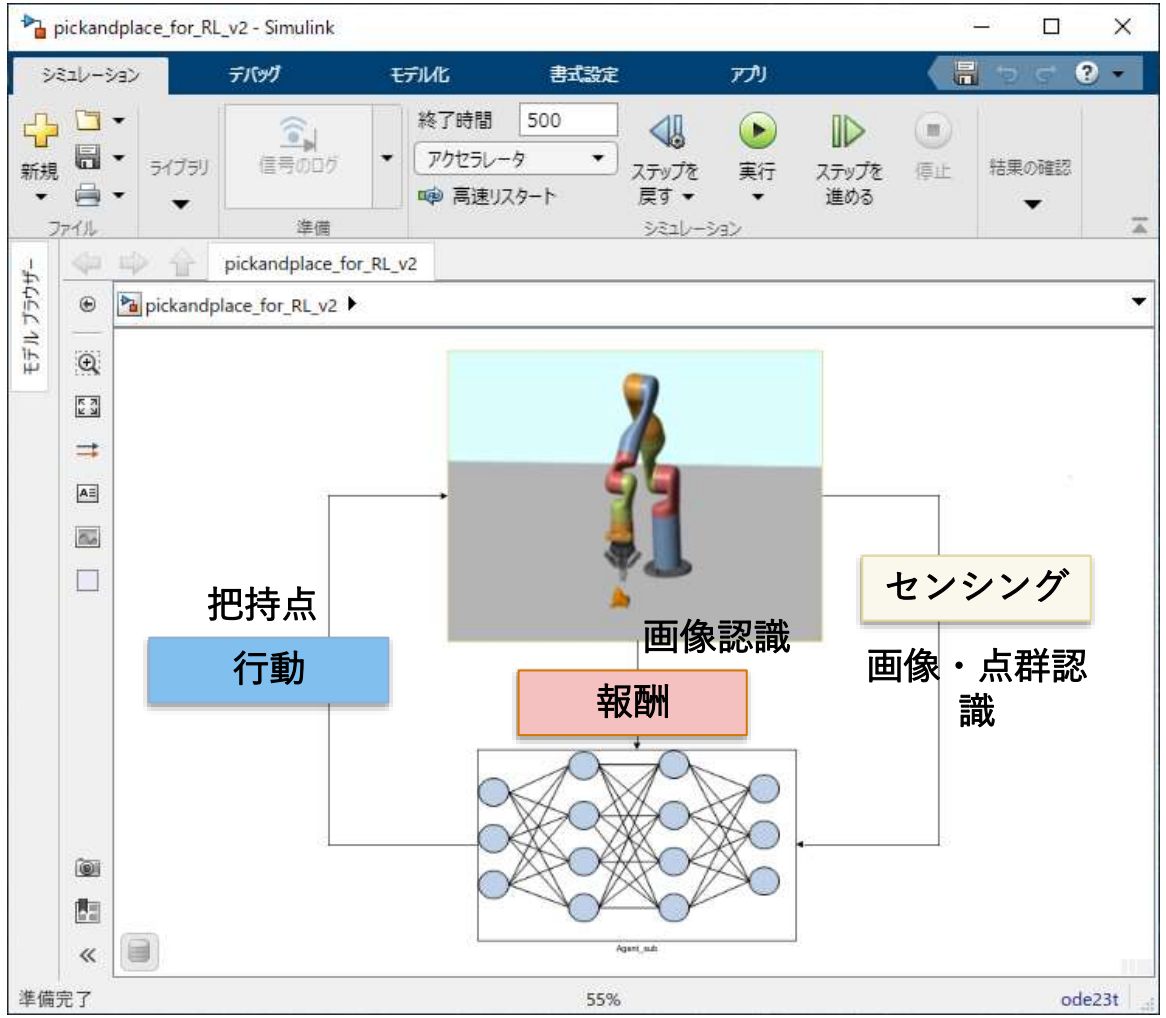
STEP3. 実運用

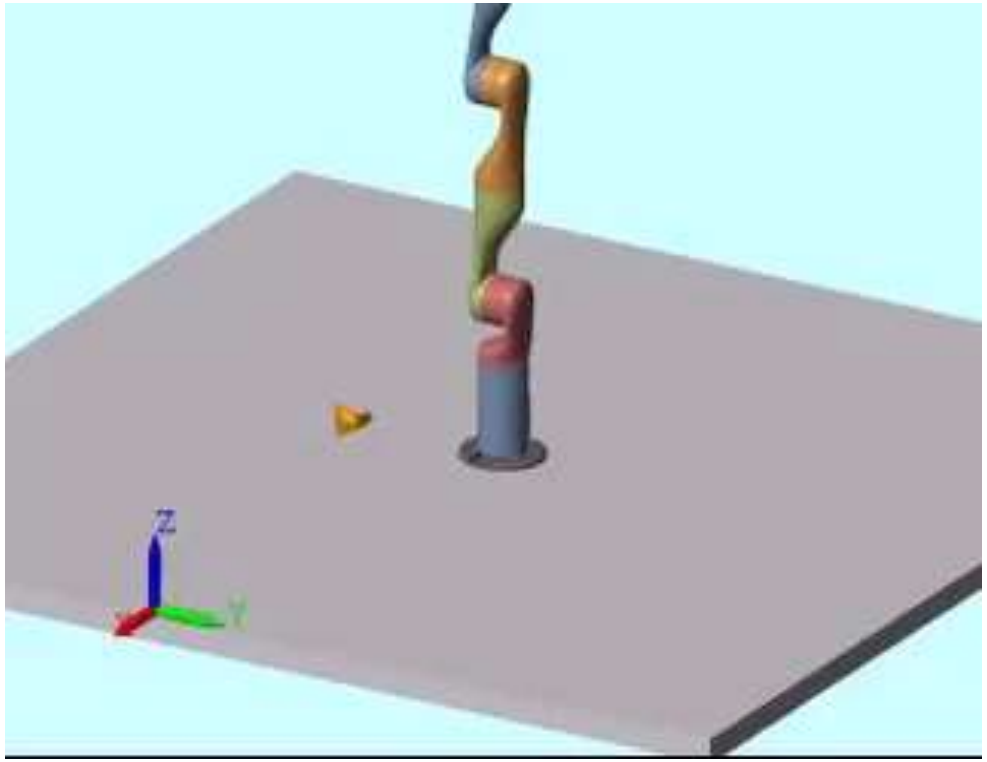
コード生成技術によりアルゴリズムをシームレスに展開



コード生成
実装

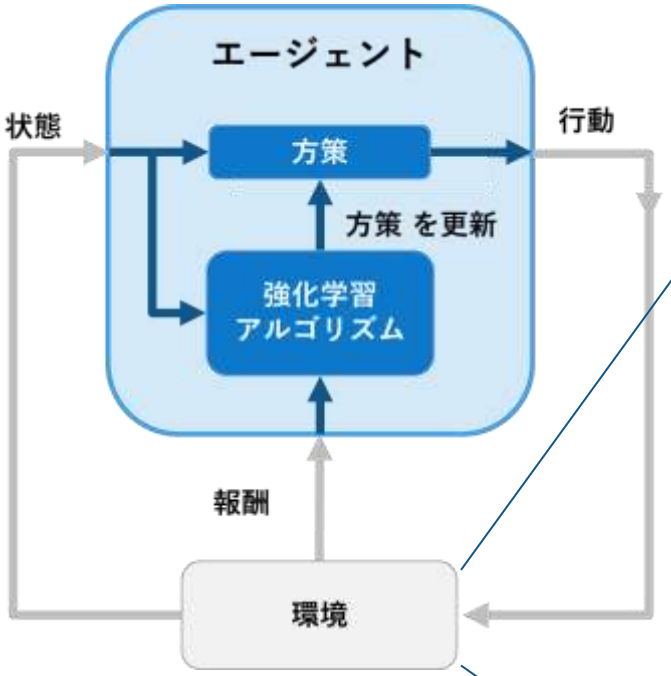
自律システムの例(強化学習によるチューニング)

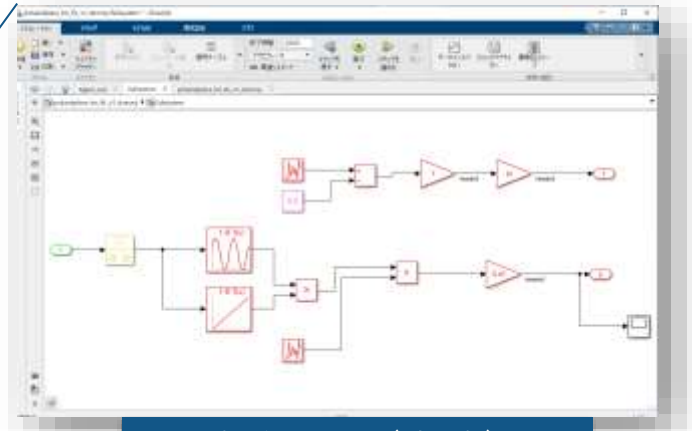




高度自律システム開発：Reinforcement Learning

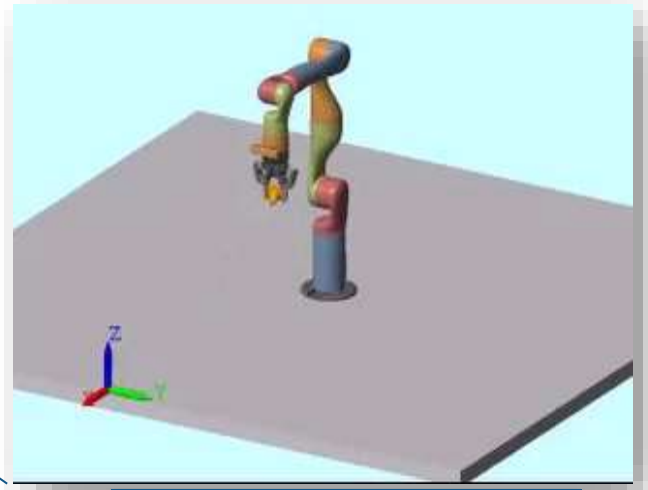
モデルの詳細度とパラメータ設計



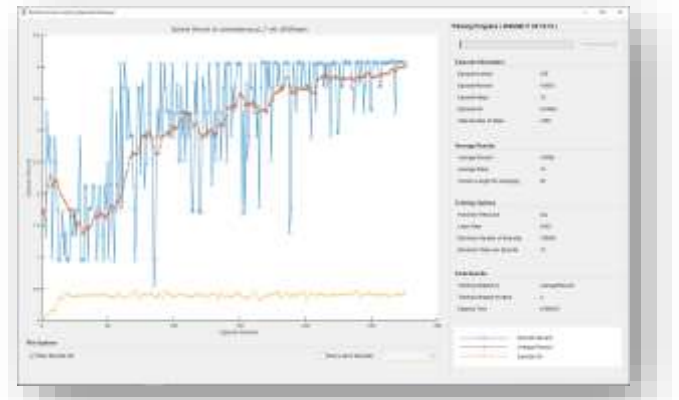


類似問題(高速)

パラメータ設計検討



厳密な問題

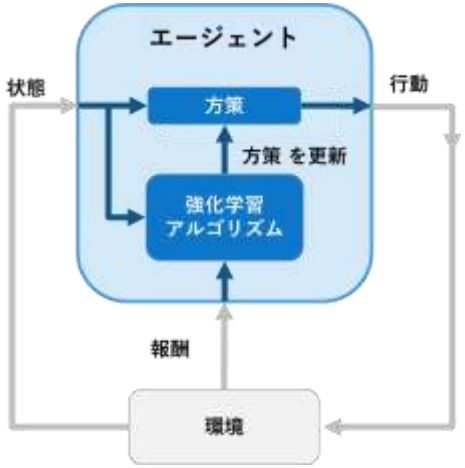


実問題へのトライ

29

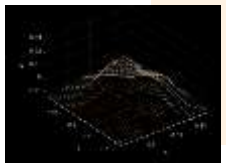
試験設備のシステム構成案

画像は三菱電機FAサイトより引用
http://www.mitsubishielectric.co.jp/news/2017/0302-b.html



強化学習アルゴリズム

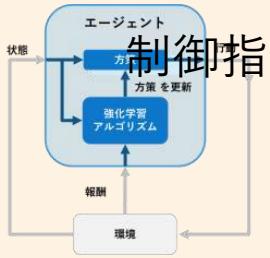
センシング



認識

MATLAB/Simulink

報酬



制御指令

Ethernet

完了信号

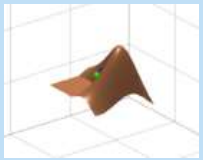
ビジョンシステムなど

産業用ロボット

ロボットコントローラ

シーケンサ
ダイレクト機能

三菱電機シーケンサ



コンベヤ等

把持対象などの環境

各種機器との
柔軟な接続

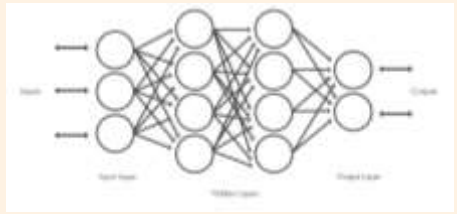
STOP

停止スイッチ

環境

全体の処理の流れ

強化学習アルゴリズム



MATLAB/Simulink

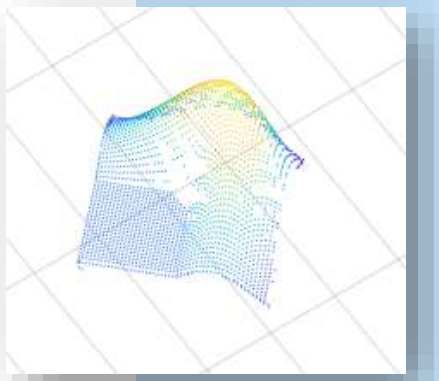
制御指令

④Ethernet経由で
指令値を送信

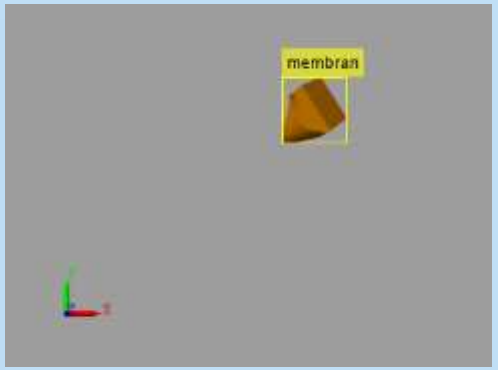
⑥完了信号

③センシング
情報として
NNに入力

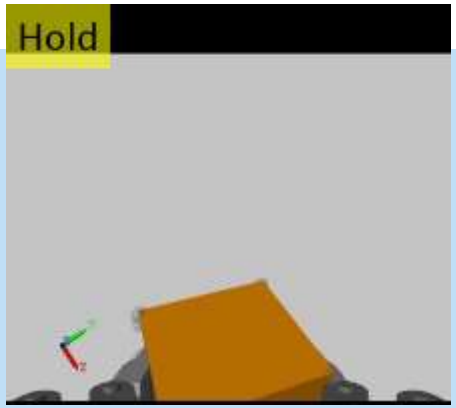
報酬



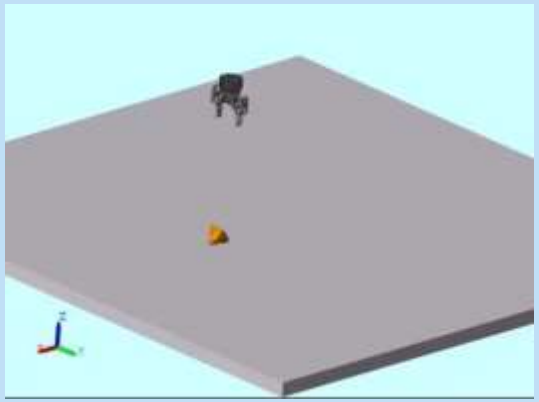
②点群から姿勢を
推定



①画像データから
位置座標を取得



⑦画像データから
成否を判定



⑤指令に基づき
ピッキングにトライ

⑧再度センシングへ

ポイント

STEP2.プロトタイプ

三菱電機シーケンサ×MATLAB/Simulink
ロバストな試験環境を早期に構築

MATLAB/Simulinkのアルゴリズム開発の強みと、
柔軟な接続性など三菱電機シーケンサの強みを活用

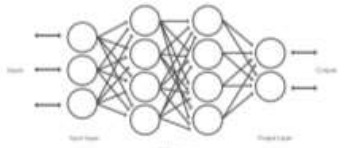
×

学習したネットワークの実運用

STEP3.実運用

コード生成技術によりアルゴリズムをシームレスに展開

-学習したNeural networkなどの高度アルゴリズムのコード生成および実装に対応



強化学習
認識

コード生成
実装

まとめ：自律システムにおける課題とは？

複合領域の
専門性

開発環境

×

【三菱電機シーケンサ×MATLAB/Simulink】

高度な
アルゴリズム

システムの
安全性

Accelerating the pace of engineering and science