

Simulink®/Stateflow®入門

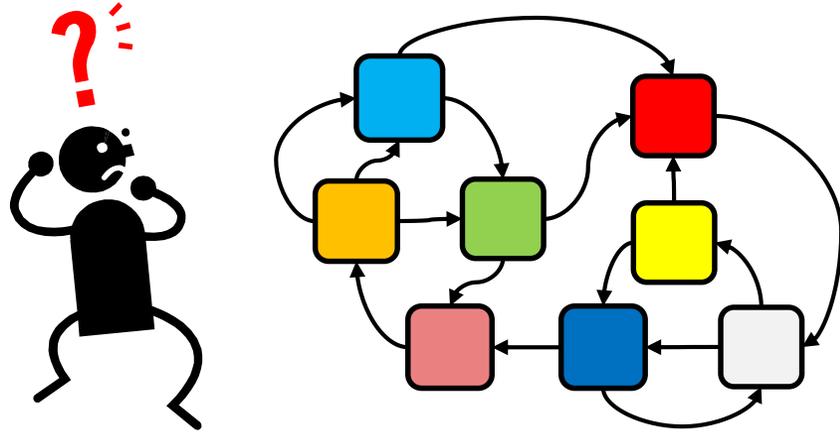
MathWorks Japan

アプリケーションエンジニアリング部 制御

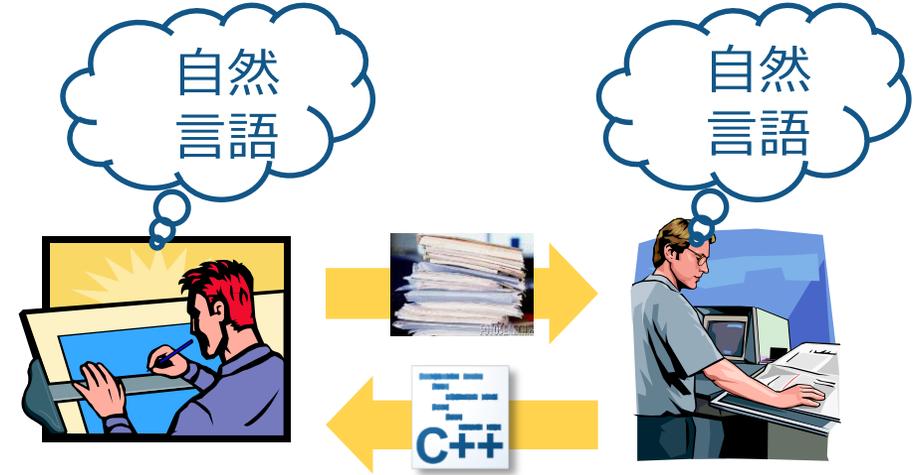
新井 克明

製品の研究・開発工程に立ちはだかる様々な課題

開発対象の高度化・複雑化



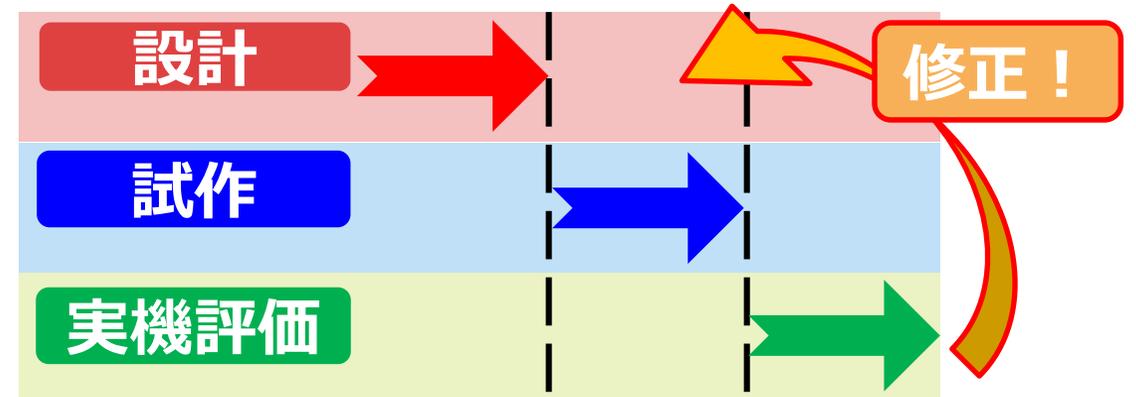
自然言語での仕様のやりとりによる誤解釈



実機テストまで詳細な動作確認ができない

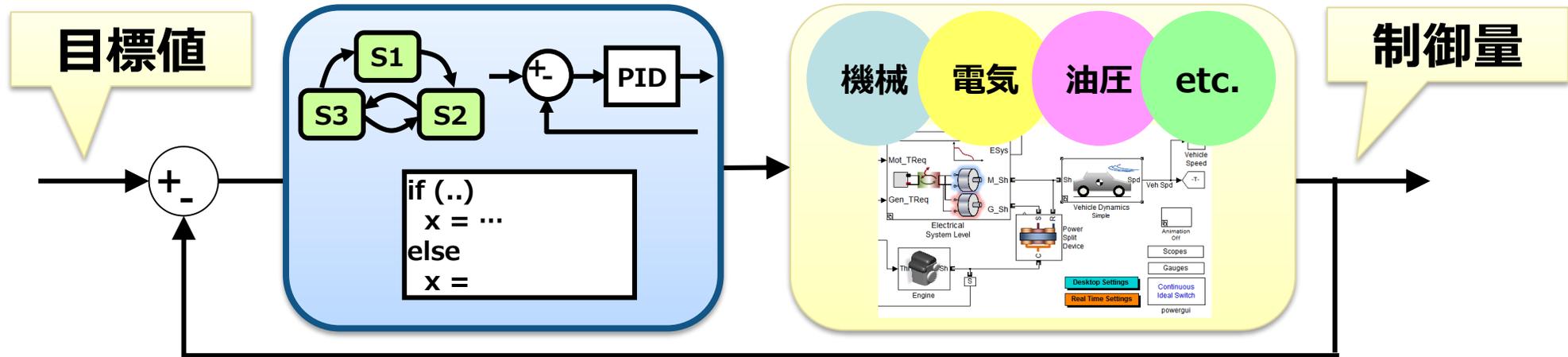


頻発する実機テストでの大きな手戻り

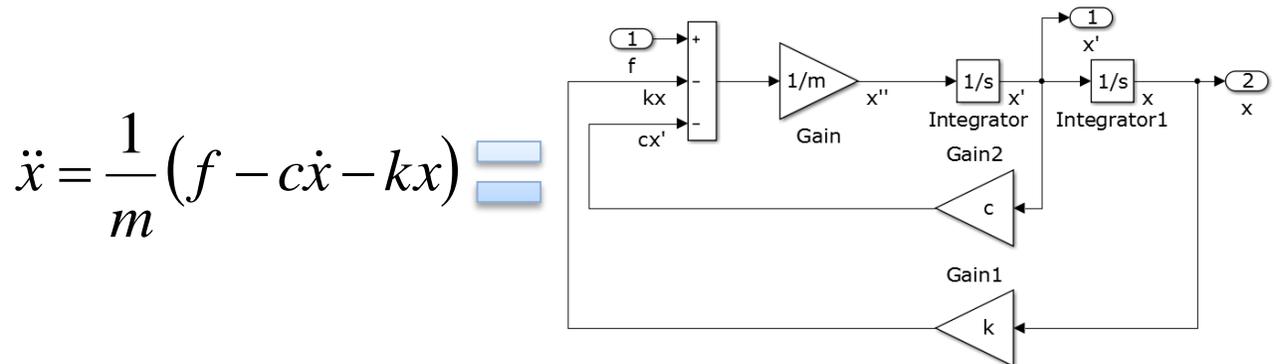


モデルを使ったシステム設計で研究・開発プロセスの高精度化/高効率化

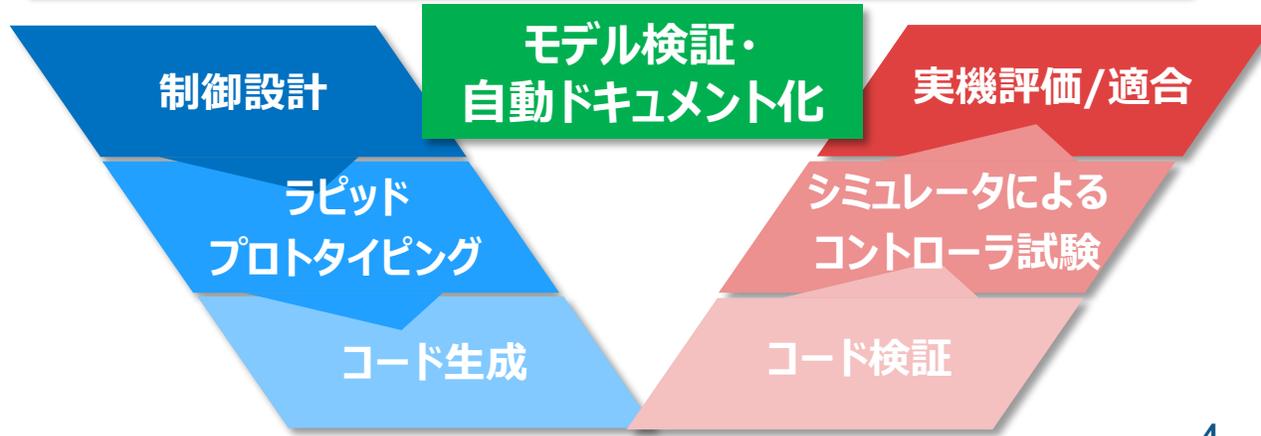
システムレベルでの詳細な動作検証



曖昧性のない数式モデルによる意思疎通

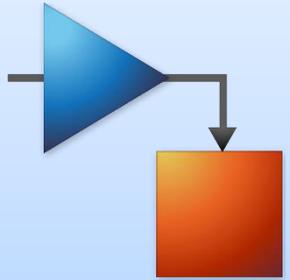


モデルを活用した一貫した開発フロー



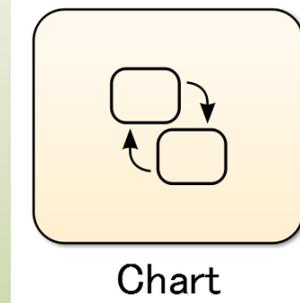
MathWorksのモデルベースデザイン(MBD)環境

Simulink® GUIベースのMBD基本環境



- ブロックダイアグラムによるグラフィカルなシステム設計
- 時間軸ベースでのシミュレーション

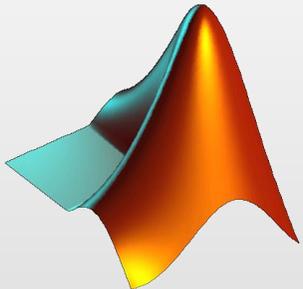
Stateflow® モードロジック設計用アドオン



- Simulinkとシームレスに接続
- フローチャート
- 状態遷移図/表
- 真理値表

Simulinkによるシステム設計を更に強力にサポート!

MATLAB® コマンドベース技術計算環境

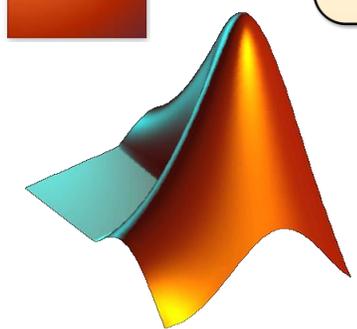
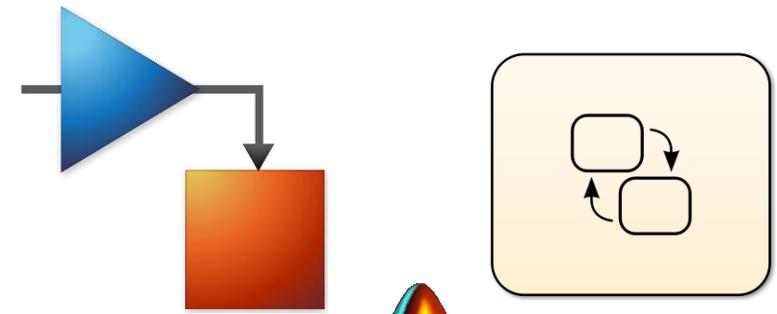


- パラメータの定義
- パラメータスタディ (繰り返しシミュレーション)
- モデルの探索・検査/自動結線
- 自動チューニング
- 最適化
- 機械学習
- 画像処理
- ...etc

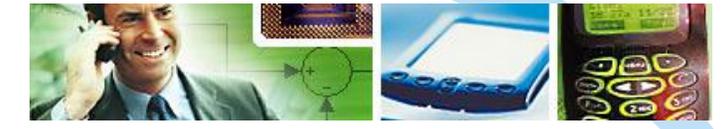
あらゆる技術領域で活用されるMATLAB&Simulink / Stateflow



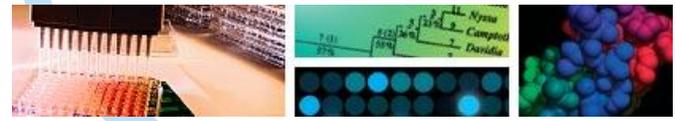
科学技術計算



金融解析



通信/エレクトロニクス/半導体



生命工学



制御関連



画像処理



実験・計測



2輪走行ロボットのシステムデザインをしてみましょう

- ◆ 2輪走行ロボットのシステム仕様の検討
 - ✓ 2輪駆動の自走式ロボット
 - ✓ ユーザーのボタン操作で前進・後退・方向転換

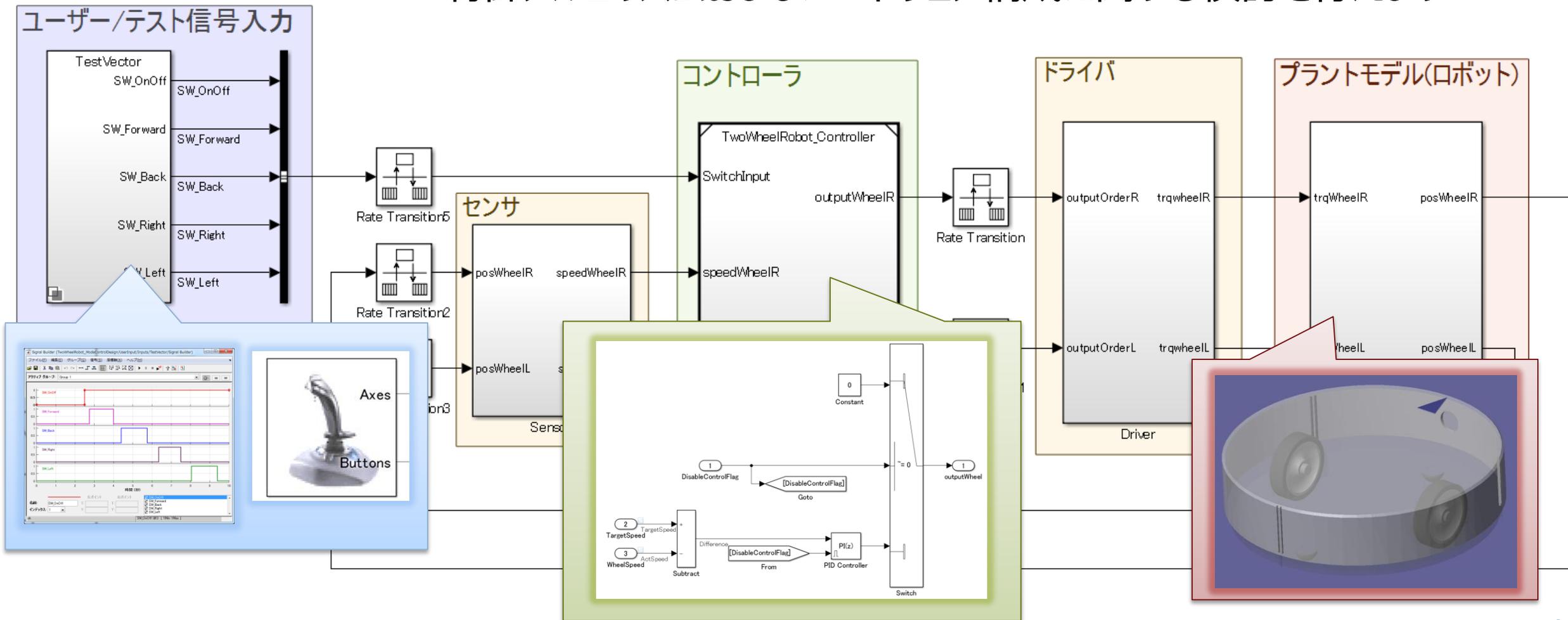


アジェンダ

- モデルを活用したシステム開発
 - Simulinkによるフィードバック制御の設計
 - Stateflowを使ったモードロジックの設計
- まとめ

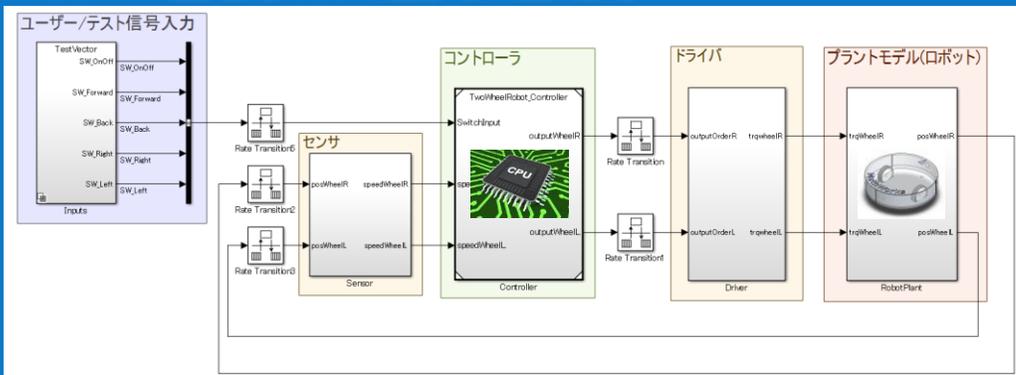
システムレベルシミュレーションによるシステム設計

- ◆ Simulinkならシステム（製品全体）を俯瞰したシミュレーションを通して制御アルゴリズムおよびハードウェア構成に関する検討を行えます



モデルを活用した一貫した開発フローで業務の効率化が図れます

モデルを使った詳細設計



制御設計

ラピッド
プロトタイピング

コード生成

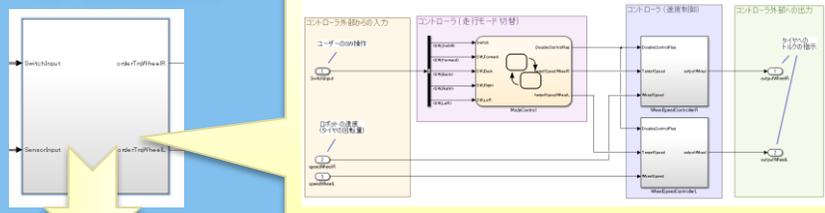
実機評価/適合

シミュレータによる
コントローラ試験

コード検証

モデル検証・成果物の作成

モデルからコード実装へ

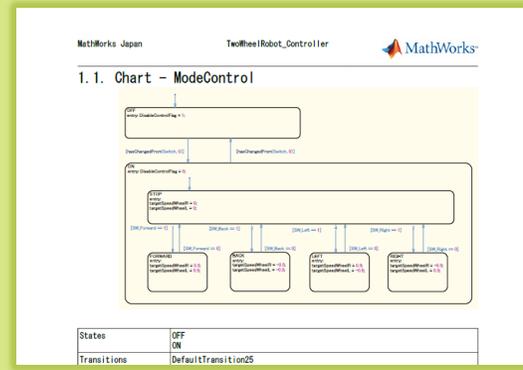
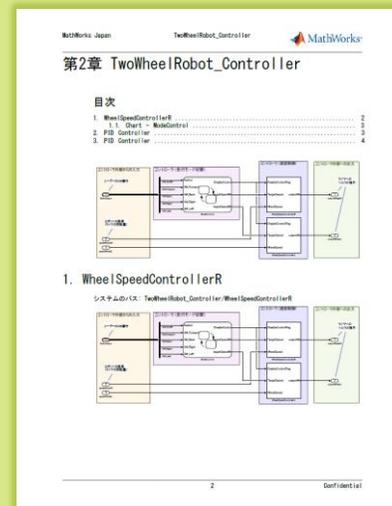


```

194
195 /* Sum: '<S3>/Subtract' incorporates:
196 * Inport: '<Root>/speedWheelR'
197 */
198 rtb_Difference = TwoWheelRobot_Controller_B.targetSpeedWheelR -
199 TwoWheelRobot_Controller_U.speedWheelR;
200

```

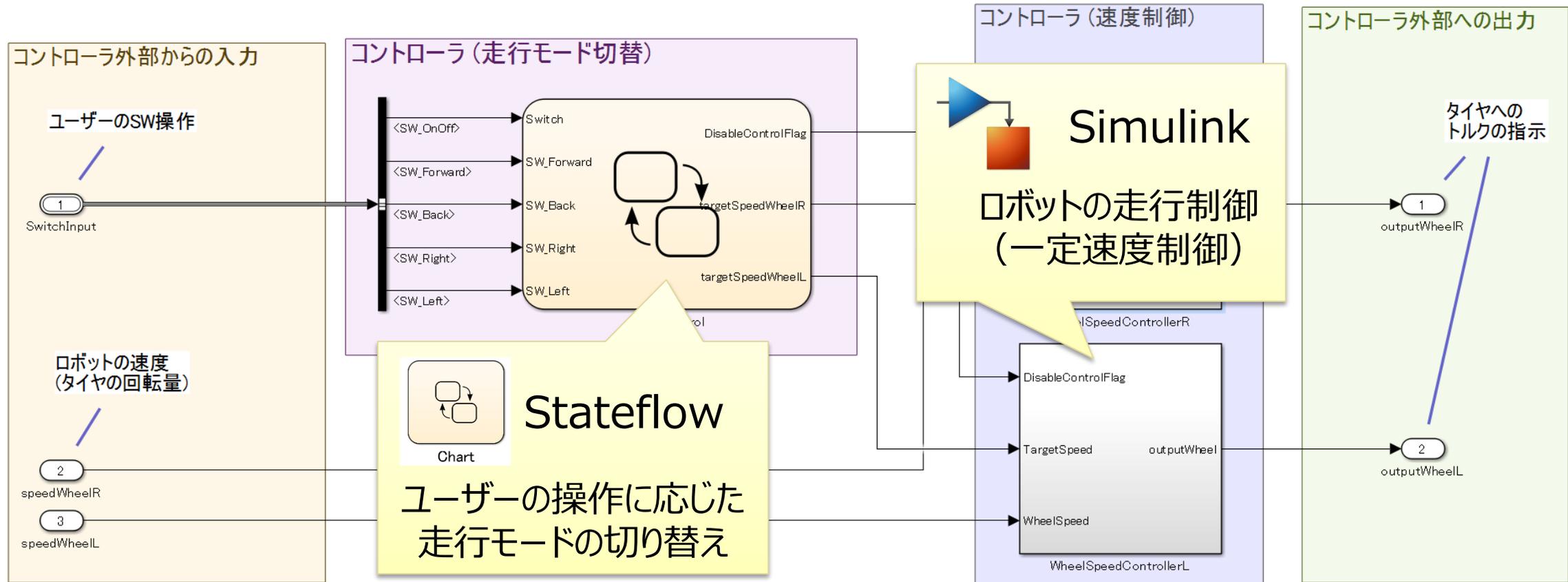
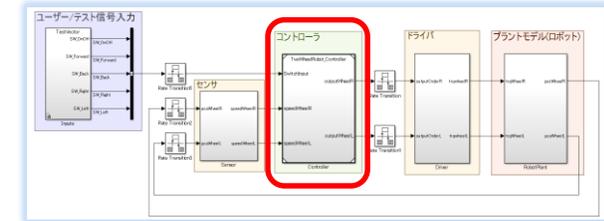
成果物のドキュメント化



制御器(コントローラ)のモデリング

本セッションでメインでご紹介します！

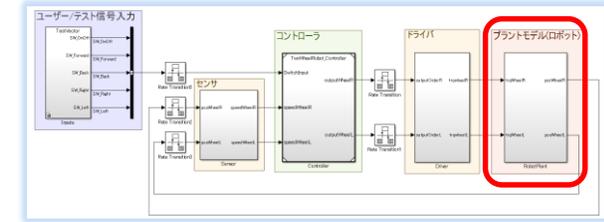
- ◆ SimulinkとStateflowで2輪走行ロボットの制御アルゴリズム
 - 機能ごとのサブシステム化で見通しの良くデザインできます



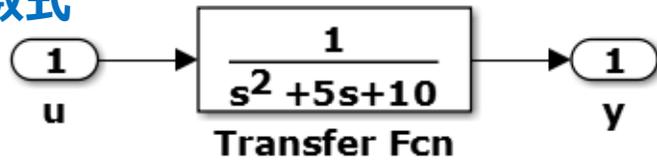
制御対象(プラントモデル)のモデリング

プラントモデリング関連セッションをご覧ください！

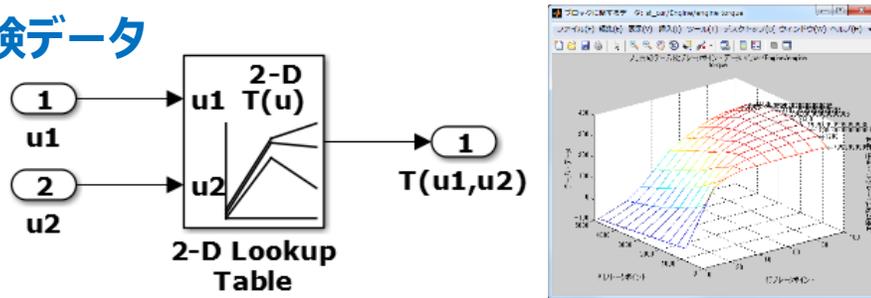
- ◆ プラントモデリングのための様々なアプローチを提供します



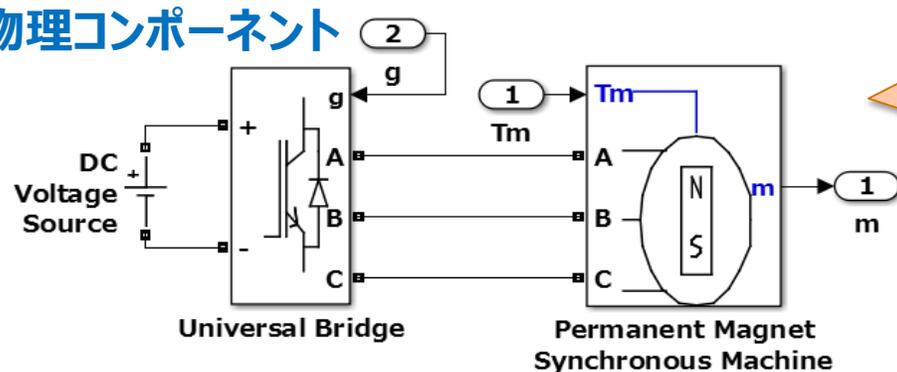
数式



実験データ



物理コンポーネント

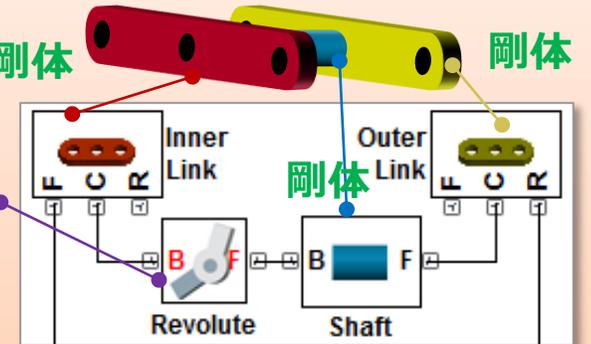


SimMechanics

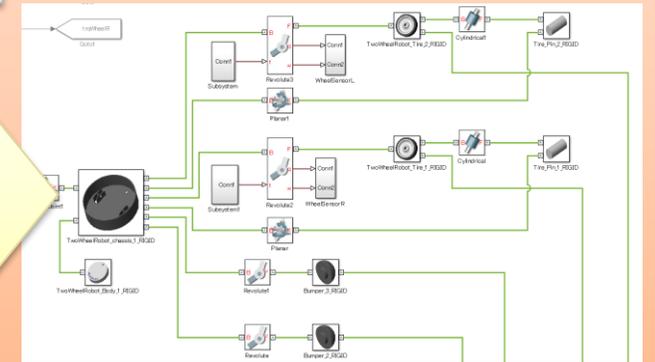
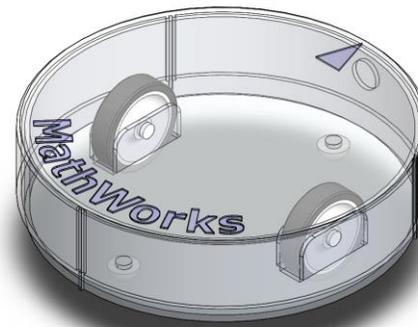
剛体

剛体

回転
ジョイント



※CADデータからインポートしたモデルをベースにカスタマイズ

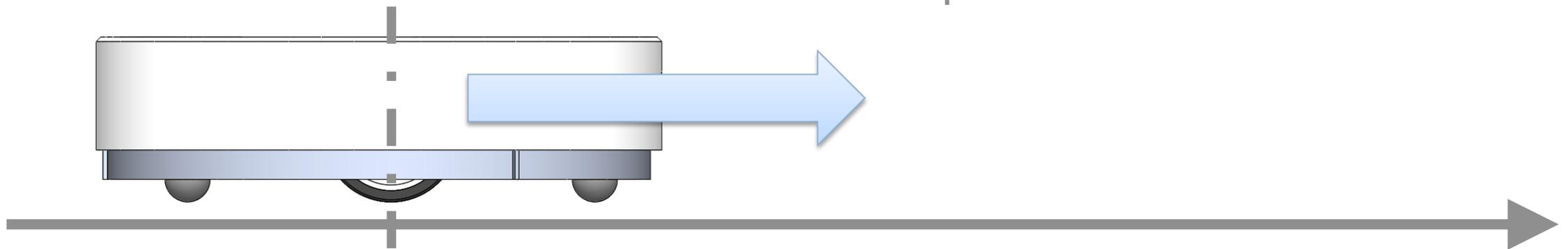
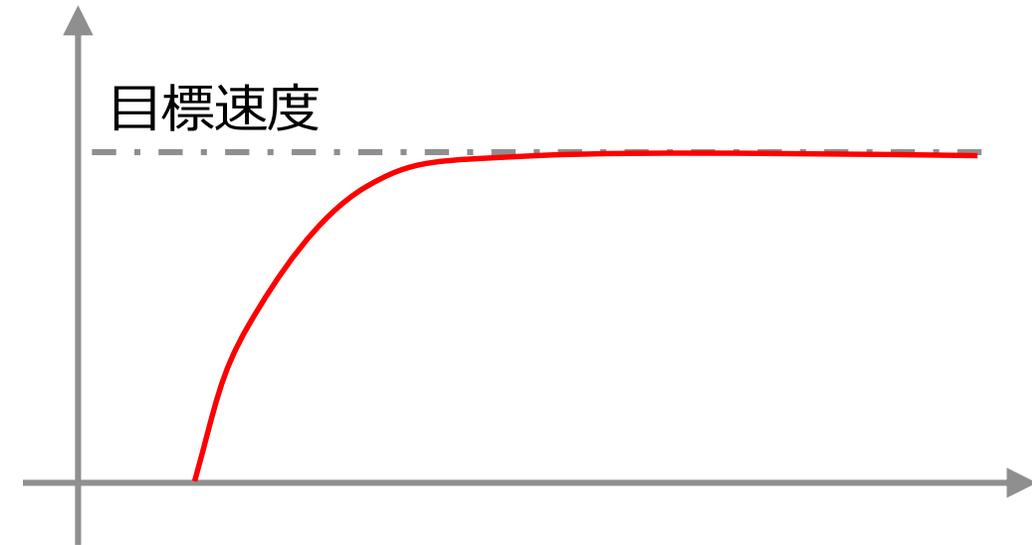


アジェンダ

- モデルを活用したシステム開発
 - Simulinkによるフィードバック制御の設計
 - Stateflowを使ったモードロジックの設計
- まとめ

2 輪走行ロボットの走行制御

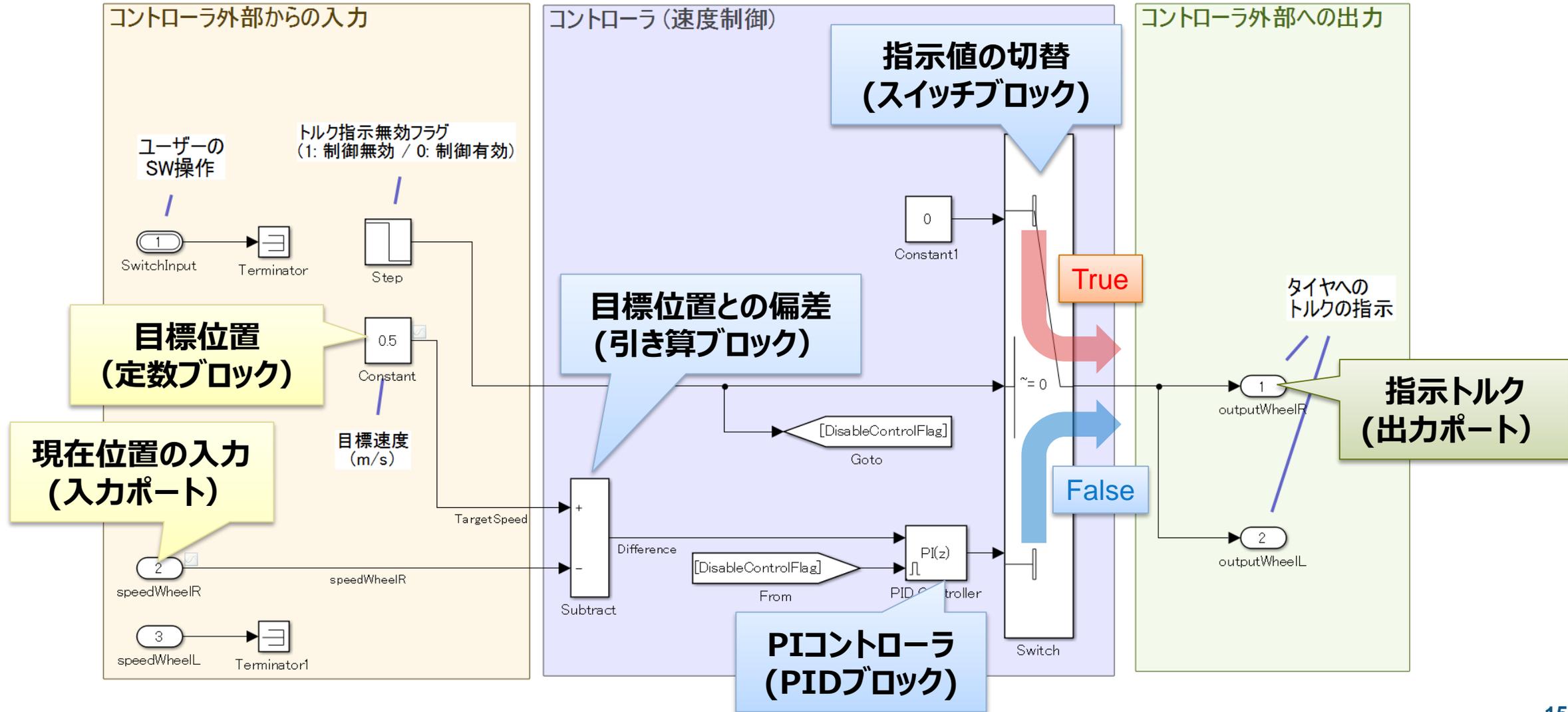
- ◆ ユーザー指示で前進走行する制御を検討します
 - タイヤの回転量から速度を計算
 - タイヤの駆動は理想的なトルク源があると仮定
 - タイヤのスリップは考慮しない
 - 一定速度を目標にしたフィードバック制御



停止状態

Simulinkによる速度制御コントローラ的设计

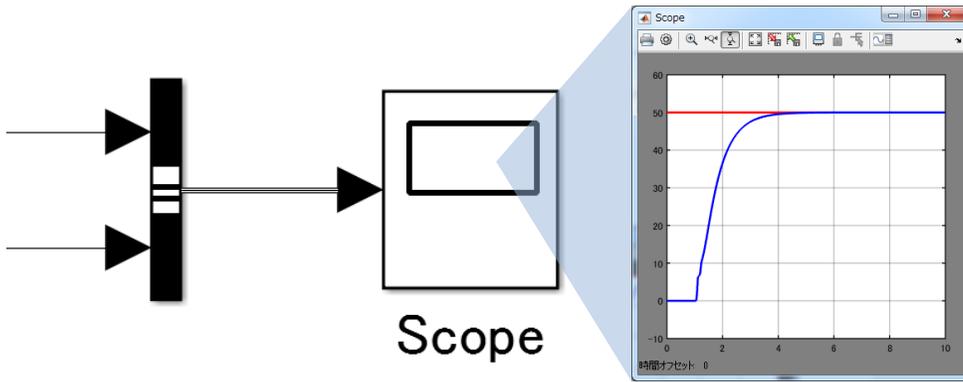
- ◆ ライブラリブロックを活用してアルゴリズムを設計できます



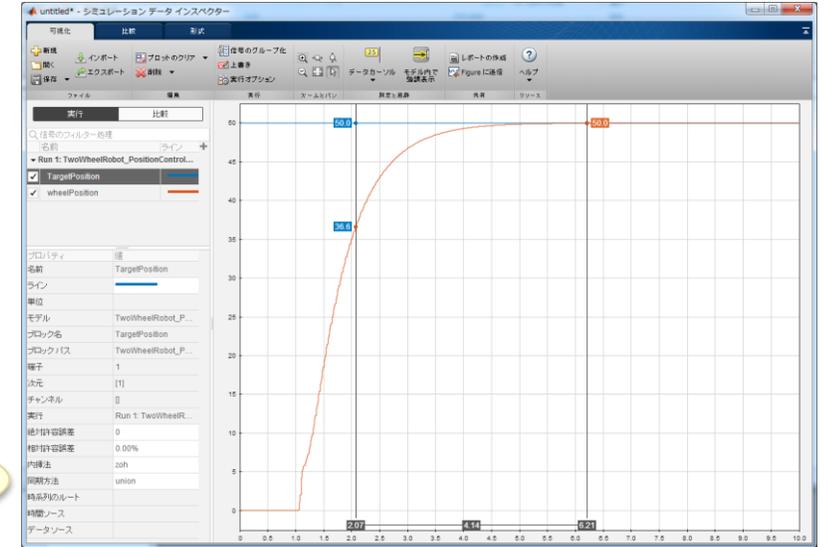
シミュレーション結果の確認の仕方

Simulinkでは様々なデータの取得 & 可視化方法を提供しています

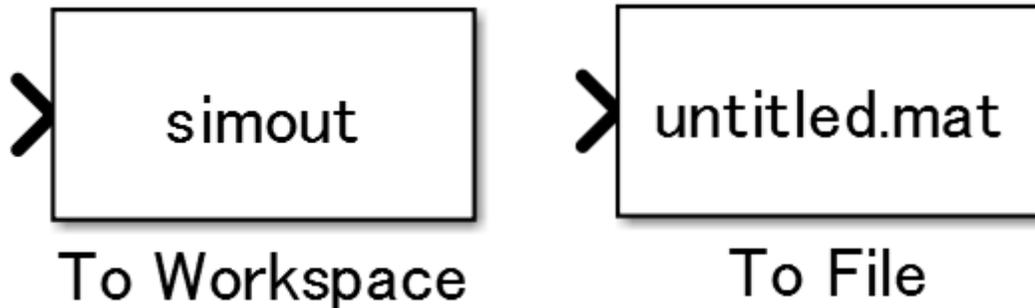
Scopeブロックによるロギング



Simulation Data Inspectorによる可視化



MATLABワークスペース/ファイルへの出力



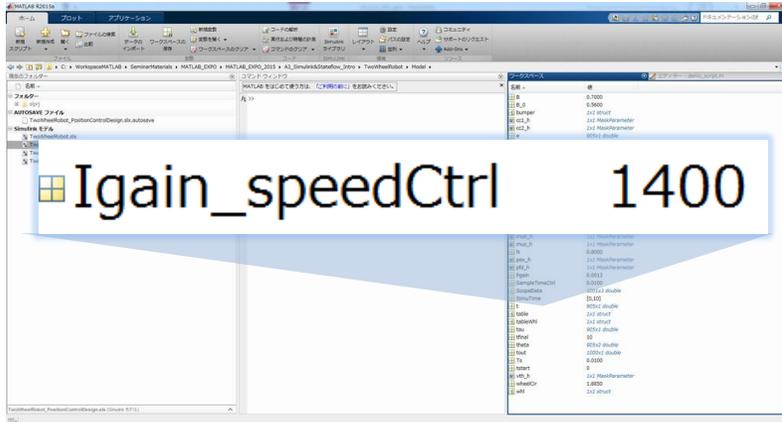
1
wheelPosition

The screenshot shows the context menu for the 'wheelPosition' signal in the SDI. The menu items are:

- 10 ノーマル
- シミュレーション データ インспекター
- 選択した信号をデータ インспекターにストリーミング** (highlighted with an orange box)
- 選択した信号をワークスペースにログ
- ワークスペース データのログをデータ インспекターに送信
- ログおよびストリーミングの設定...
- シミュレーション データ インспекターのヘルプ...

MATLABと連携したSimulinkのシミュレーション自動化

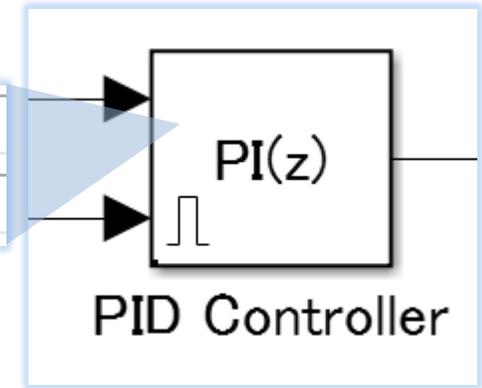
- ◆ MATLABで制御変数を自動的にスイープすることでシミュレーション検証を効率化できます



Simulinkは
MATLABワークスペースの
変数を参照できます

比例項 (P): Pgain_speedCtrl

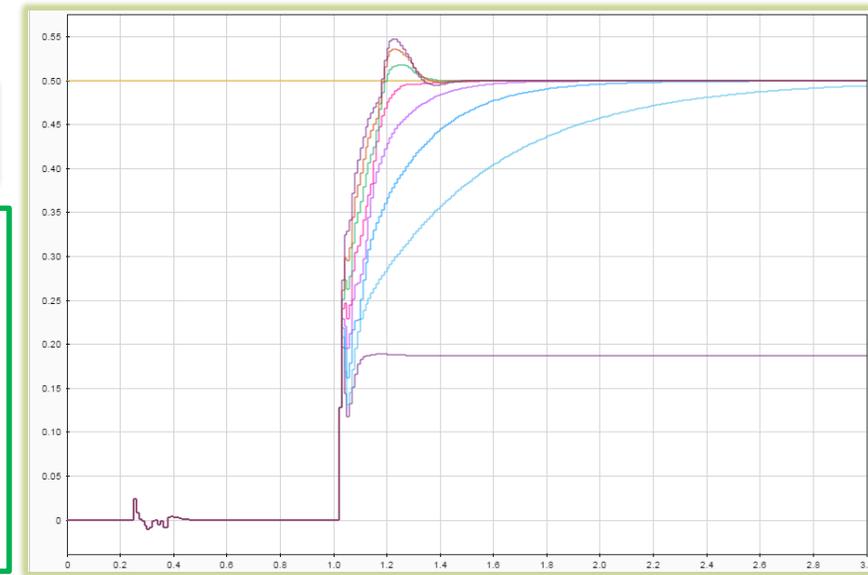
積分項 (I): Igain_speedCtrl



パラメータスイープしながら処理を繰り返す

```
for Igain_speedCtrl = 0:200:1400
    sim(gcs);
end
```

シミュレーション実行

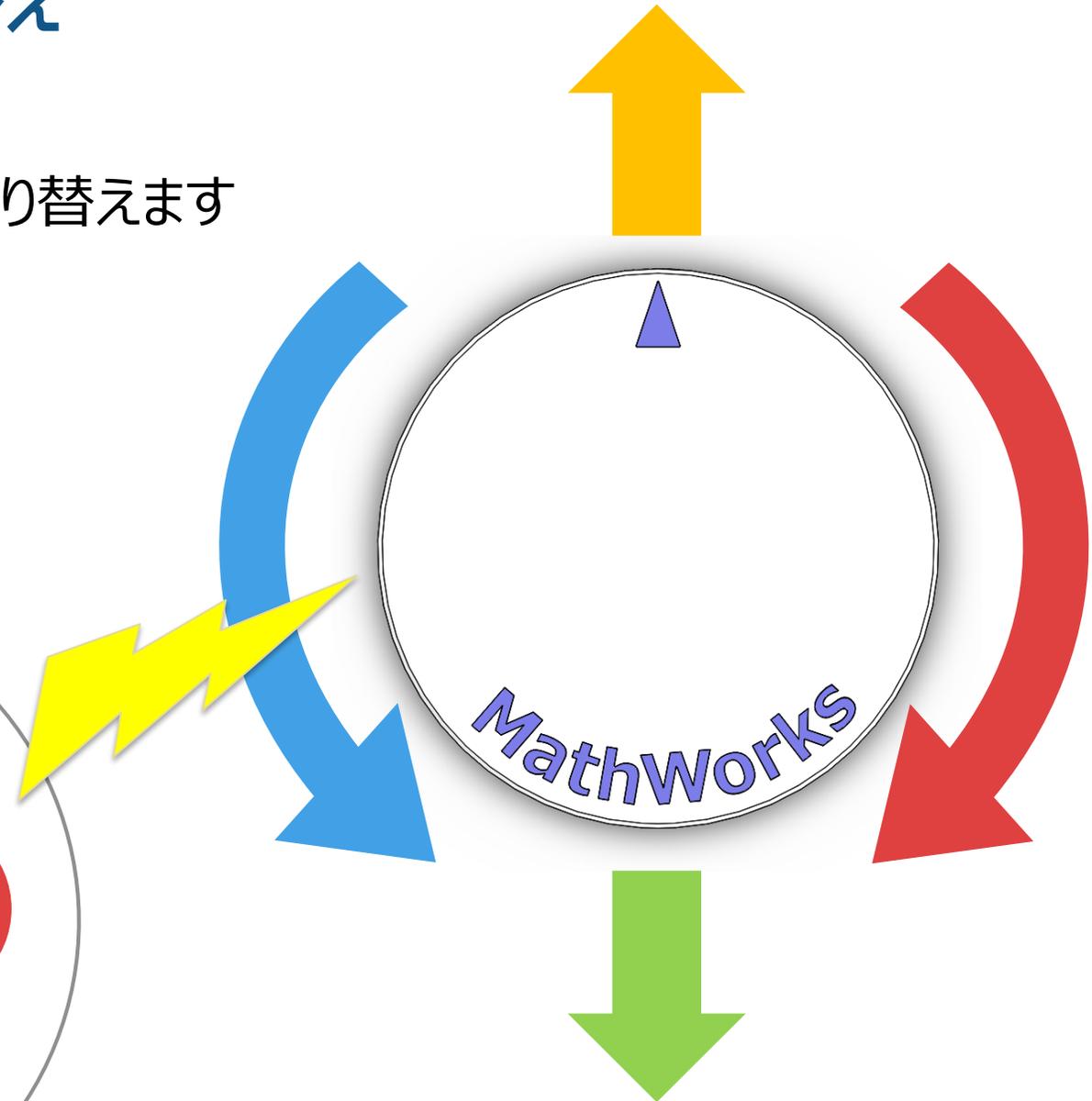
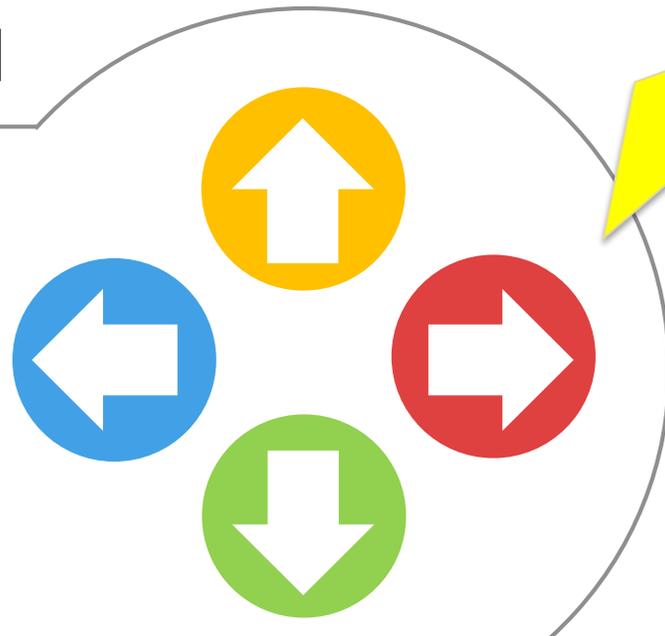
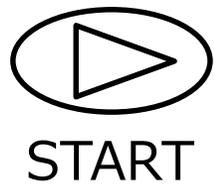


アジェンダ

- モデルを活用したシステム開発
 - Simulinkによるフィードバック制御の設計
 - Stateflowを使ったモードロジックの設計
- まとめ

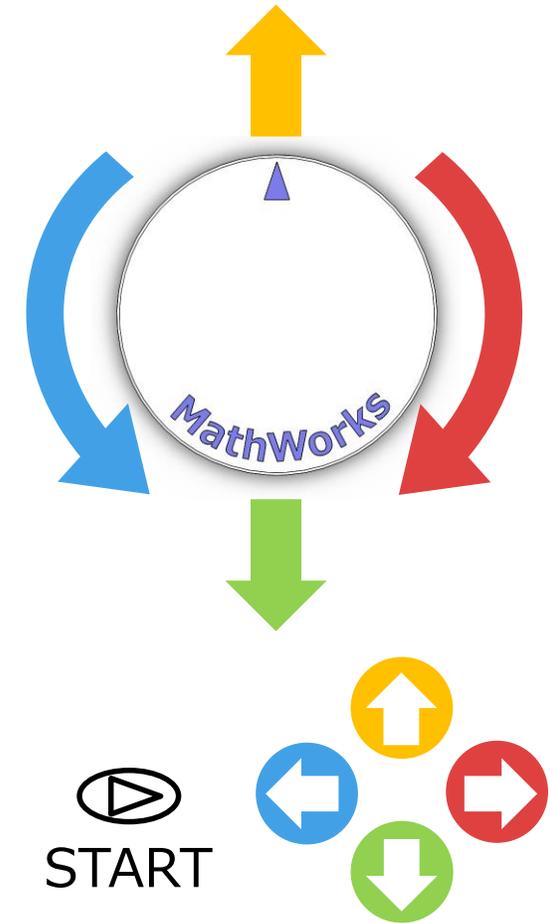
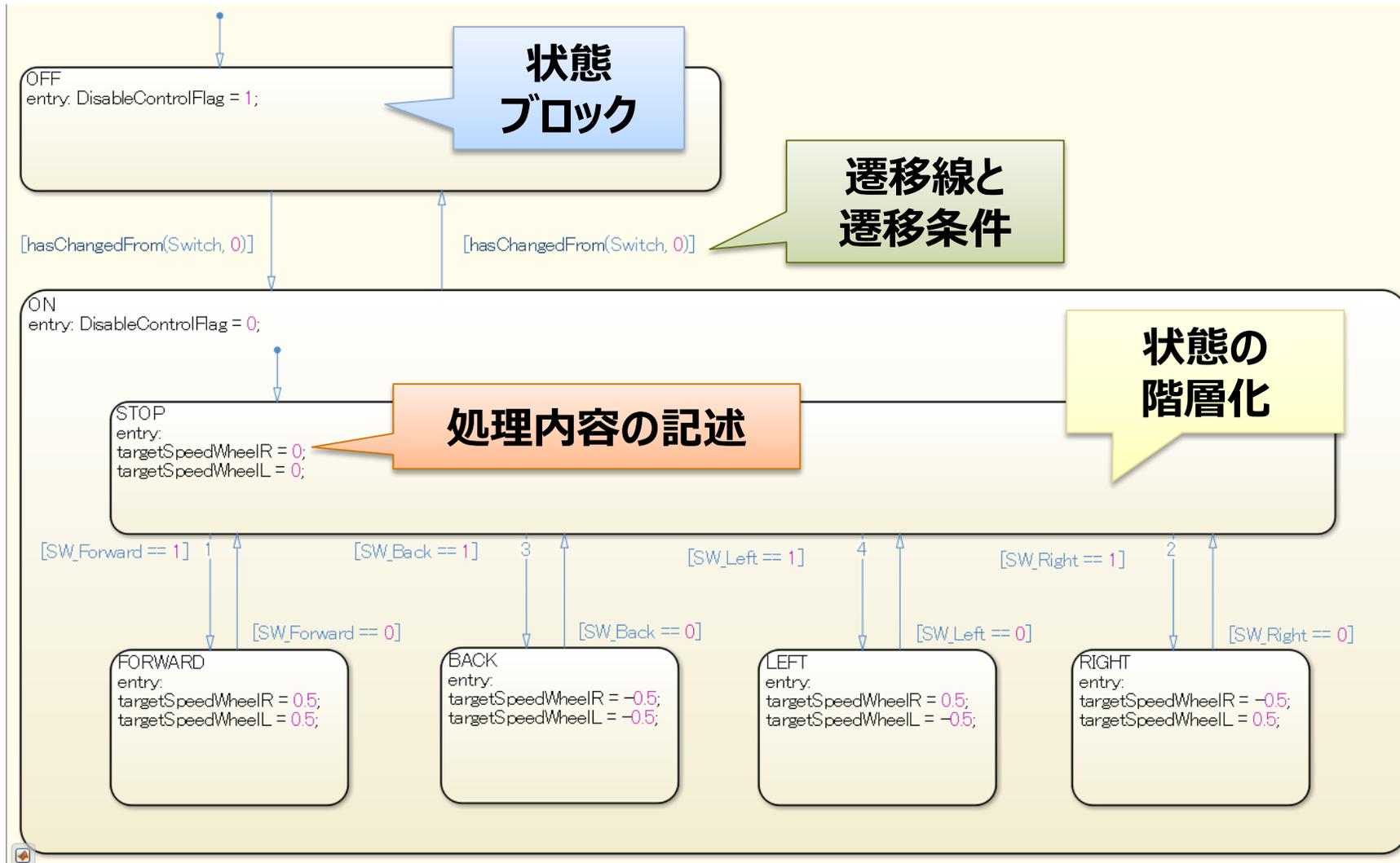
2 輪走行ロボットの走行モードの切り替え

- ◆ ユーザーのボタン入力に応じて走行方向を切り替えます
 - STARTボタン: 制御開始
 - ↑ボタン: 前進
 - ↓ボタン: 後退
 - ←ボタン: 左旋回
 - →ボタン: 右旋回



Stateflowによるモードロジック設計

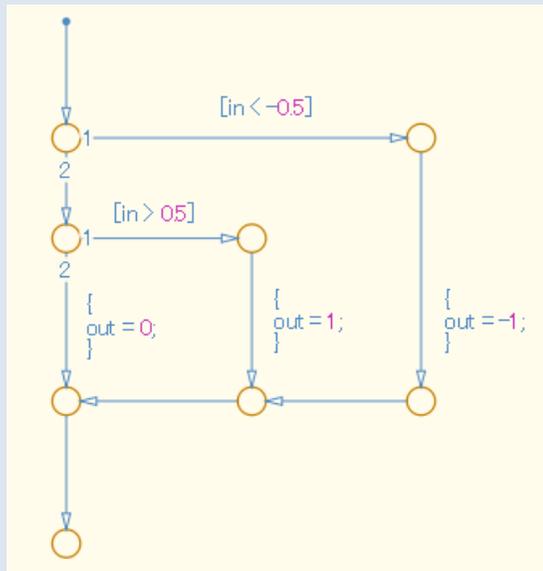
- ◆ 直感的なモデリングスタイルでモードロジックを設計できます



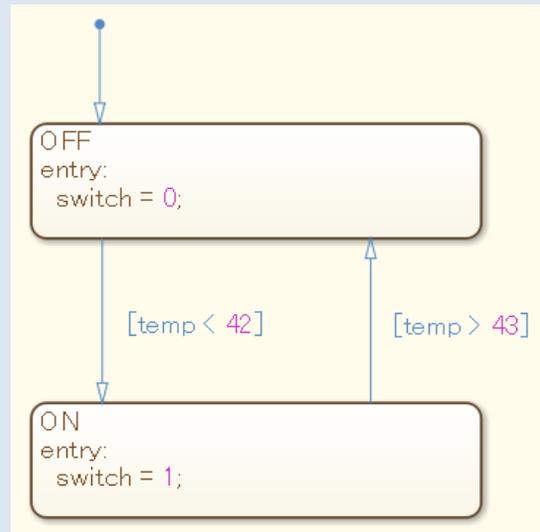
Stateflowによる色々なロジック表現

- ◆ 設計したいアルゴリズムに応じて様々なデザインスタイルを活用できます

フローチャート



状態遷移図



状態遷移表

ステート	遷移 (条件/アクション/遷移先ステート)	
	if	else-if (1)
state1	[in > 1]	[in > 2]
state1	en: out = 1;	
state1_1	[flag]	[x > 0]
state1_1	en, du: cnt = 0;	[x = x + 1;]
state1_2	[flag]	
state1_2	en: cnt = cnt + 1;	
state2	[in < -1]	
state3	[after(5, sec)]	
state3	en: out = 3;	
	state1	\$PREV
		\$NEXT
		\$SELF
		state1
		state3
		% IGNORE %

真理値表

条件テーブル				
	説明	条件	D1	D2
1	Hot	t > T_thresh	T	T
2	Dry	h < H_thresh	T	-
		アクション: アクション テーブルから行を指定	CoolOn, HumidOn	CoolOn

アクション テーブル		
#	説明	アクション
1	Turn On Cooling (This implicitly reduces humidity)	CoolOn: cooler = 1; heater = 0; humidifier = 0;
2	Turn On Heater (This implicitly reduces humidity)	HeatOn: heater = 1; cooler = 0; humidifier = 0;
3	Turn On Humidifier	HumidOn: humidifier = 1;

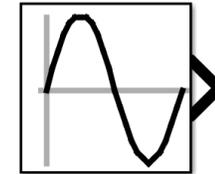
Simulinkへのテストパターン入力

目的に応じた様々な入力信号を作成できます

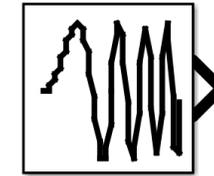
SignalBuilderによる任意波形の作成

期待値テストやカバレッジ測定に活用

Sourceライブラリブロックによる周期波形



Sine Wave



Chirp Signal



Counter Free-Running

ジョイスティック/ゲームパッドからの入力



Joystick Input

ユーザーの操作をシミュレーション中に入力

※ オプションツールボックスのSimulink 3D Animationが必要です

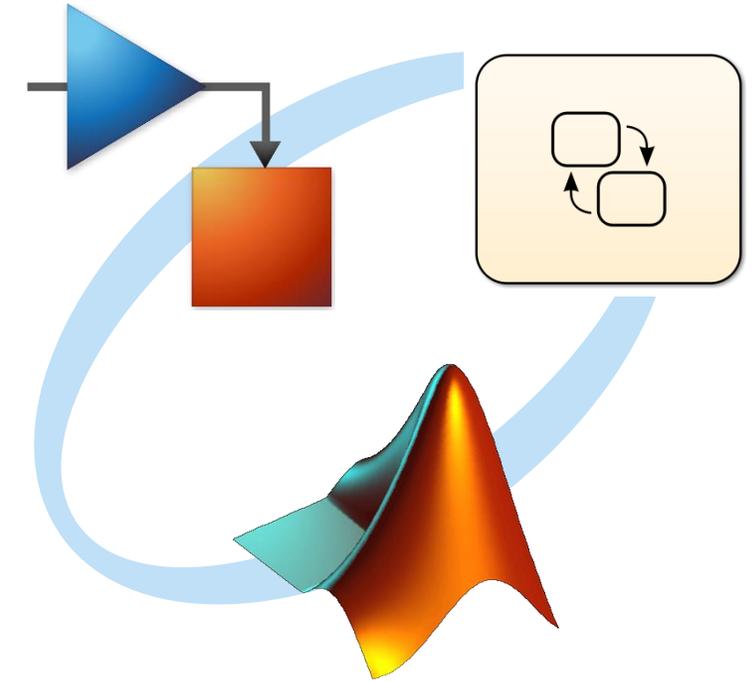
アジェンダ

- モデルを活用したシステム開発
 - Simulinkによるフィードバック制御の設計
 - Stateflowを使ったモードロジックの設計
- まとめ

まとめ

◆ SimulinkとStateflowによるシステム開発……

- グラフィカルで曖昧性のないシステムデザイン
- システムレベルシミュレーションによる早期作りこみ
- MATLABによるシミュレーションの自動化
- ソフトウェア実装などモデルの一貫した活用



モデルを使ったシステム設計で
研究・開発プロセスを高精度化/高効率化！

Simulink / Stateflow は 習って慣れば、すぐマスターできます。 (MathWorks トレーニング - 2分間CM)

● Exercise demo -

付録 D: 演習

ステート マシンのモデル化 – パワー ウィンドウ

参照先: ステート マシンのモデル化

Stateflow® チャートを使用して、自動車のパワー ウィンドウ システムをモデル化します。

- 1 つの入力 (u) – ウィンドウの移動方向
 - -1 – ウィンドウを下げる。
 - 0 – ウィンドウを動かさない。
 - 1 – ウィンドウを上げる。
- 1 つの出力 (pos) – ウィンドウの現在の位置
 - pos = 0 – ウィンドウは完全に開いている。
 - pos = 20 – ウィンドウは完全に閉じている。

この Stateflow チャートには以下の特性があります。

- ウィンドウを上げるように指示された場合 (u = 1)、ウィンドウが完全に閉じていなければ、pos の値を 1 増やす。
- ウィンドウを下げるように指示された場合 (u = -1)、ウィンドウが完全に開いていなければ、pos の値を 1 減らす。

練習

次のモデルを開いて演習を表示
>> model_state_start

- 毎月開催！
「Simulink 基礎」
- 2ヶ月に1回開催！
「Stateflow 基礎」
- 具体的な例題から
モデルを作る方法が
学べます

受講者が解けるようになる演習の一例



Accelerating the pace of engineering and science

ご清聴ありがとうございました

© 2015 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See 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.