AT International 2009

MapleSim による数式モデルを用いた モデルベース開発
～シミュレーションから実機検証まで～

サイバネットシステム株式会社

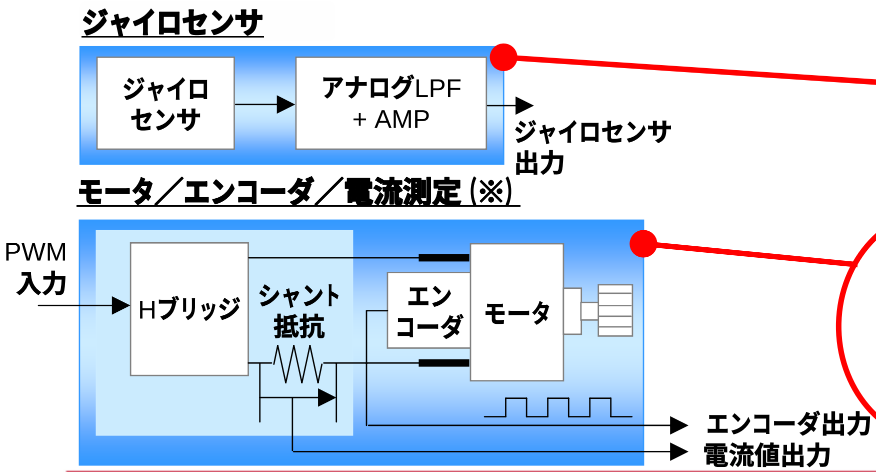
アドバンスドソリューション統括部
モデルベース開発推進室

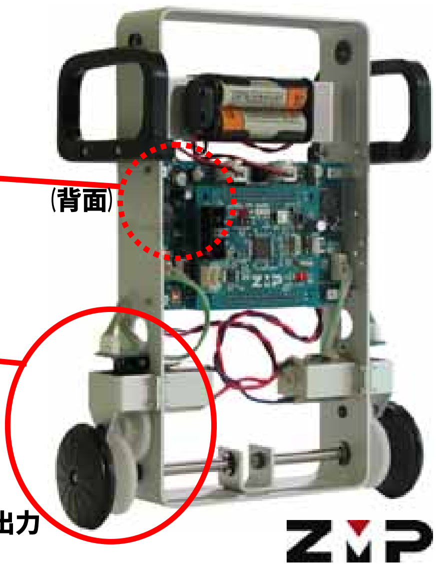
つくる情熱を、支える情熱。
CYBERNET

はじめに

■ 本セミナーでは、倒立二輪ロボットe-nuvo WHEEL の制御設計を例に、MapleSim をはじめとした具体的なツールチェーンによるモデルベース開発の流れをご紹介します。

- 名称： e-nuvo WHEEL
- 開発元： 株式会社ZMP





アジェンダ

- モデルベース開発とは
- MapleSim によるプラントモデル開発
  - MapleSim とは
  - プラントモデルの作成
  - モデルの線形化と状態空間モデルの取得
- コントローラ設計
  - MATLAB を用いた最適レギュレータの設計
  - シミュレーションによるコントローラの検証
- Speedgoatによる実機検証
  - Speedgoatとは
  - 倒立二輪ロボット制御の実機検証
- モデルベース開発の課題と方向性

CYBERNET

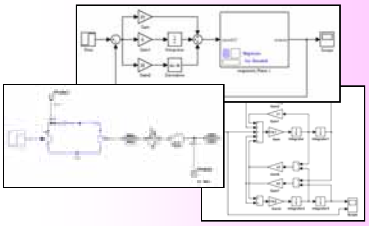
アジェンダ

- モデルベース開発とは
- MapleSim によるプラントモデル開発
  - MapleSim とは
  - プラントモデルの作成
  - モデルの線形化と状態空間モデルの取得
- コントローラ設計
  - MATLAB を用いた最適レギュレータの設計
  - シミュレーションによるコントローラの検証
- Speedgoatによる実機検証
  - Speedgoatとは
  - 倒立二輪ロボット制御の実機検証
- モデルベース開発の課題と方向性

モデルベース開発とは

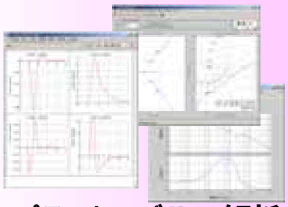
構想・設計・試作・検証といった開発プロセスを数理モデルに基づき実施する設計手法

モデリング



・プラントモデル
・コントローラモデル

解析・シミュレーション



・プラントモデルの解析
・コントローラ設計
・制御系の検証

自動コード生成・実装

・実機検証
(リアルタイムシミュレーション)
・製品向け組込コード作成

ツールチェーン

Maple
MapleSim
MATLAB & SIMULINK
speedgoat

モデルベース開発とは

■ モデルベース開発のメリット

シミュレーションによる仮想実験ができる

- 試作実験の回数、コストを軽減できる.
- 実験・観測が困難な状況も検討することができる.

試行錯誤ではないシステマティックな設計ができる

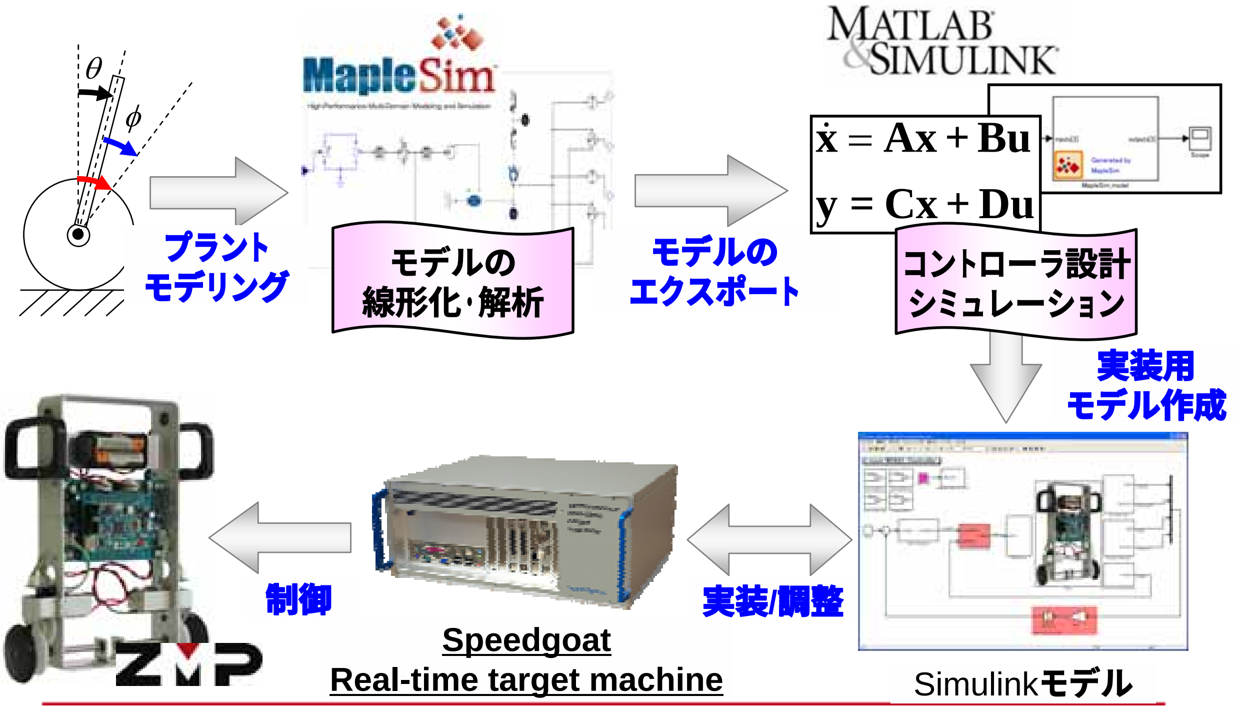
- 手戻りを削減できる.
- 設計レベルの人依存度を軽減できる.

限界性能が掌握できる.

- 性能・安全性の向上.

モデルベース開発とは

■ 本セッションでご紹介する設計の流れ



アジェンダ

■ モデルベース開発とは

■ MapleSim によるプラントモデル開発
- MapleSim とは
- プラントモデルの作成
- モデルの線形化と状態空間モデルの取得

■ コントローラ設計
- MATLAB を用いた最適レギュレータの設計
- シミュレーションによるコントローラの検証

■ Speedgoatによる実機検証
- Speedgoatとは
- 倒立二輪ロボット制御の実機検証

■ モデルベース開発の課題と方向性

MapleSim とは

MapleSim 2

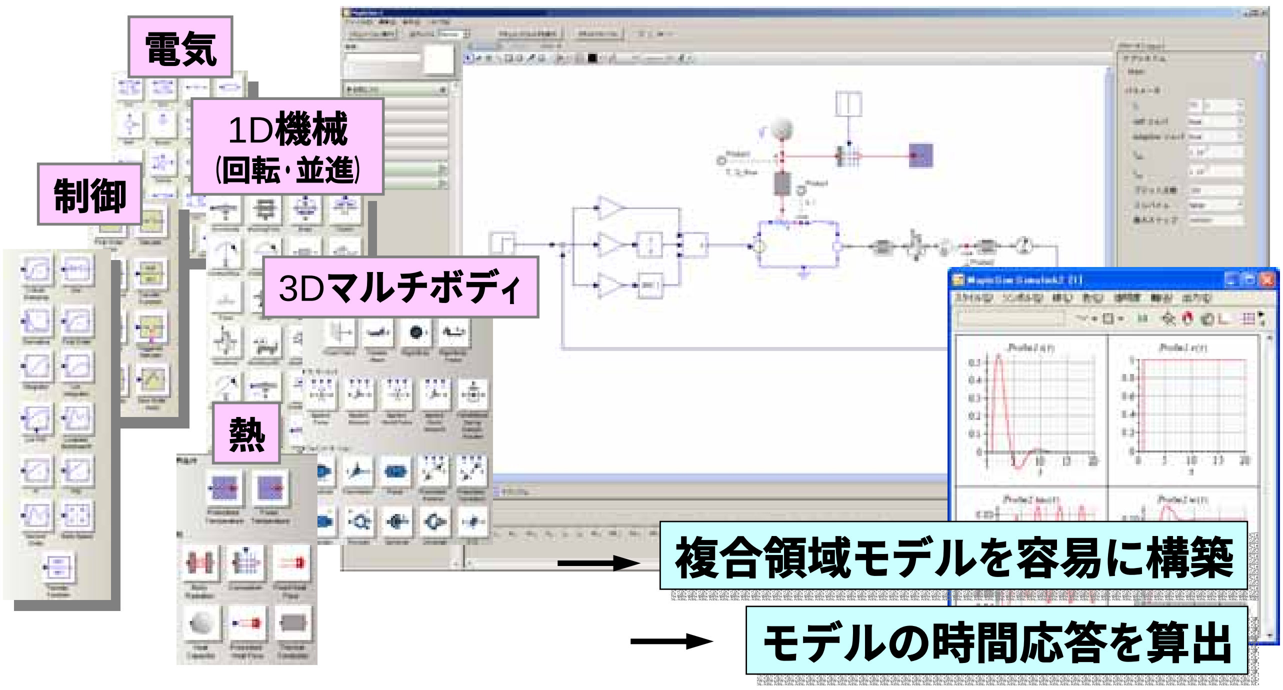
High-Performance Multi-Domain Modeling and Simulation

数式ベースの複合領域モデリング環境

- GUIによる直感的なモデル作成
- 複合領域のモデル化
- 数式モデルによる高速シミュレーションとモデル解析

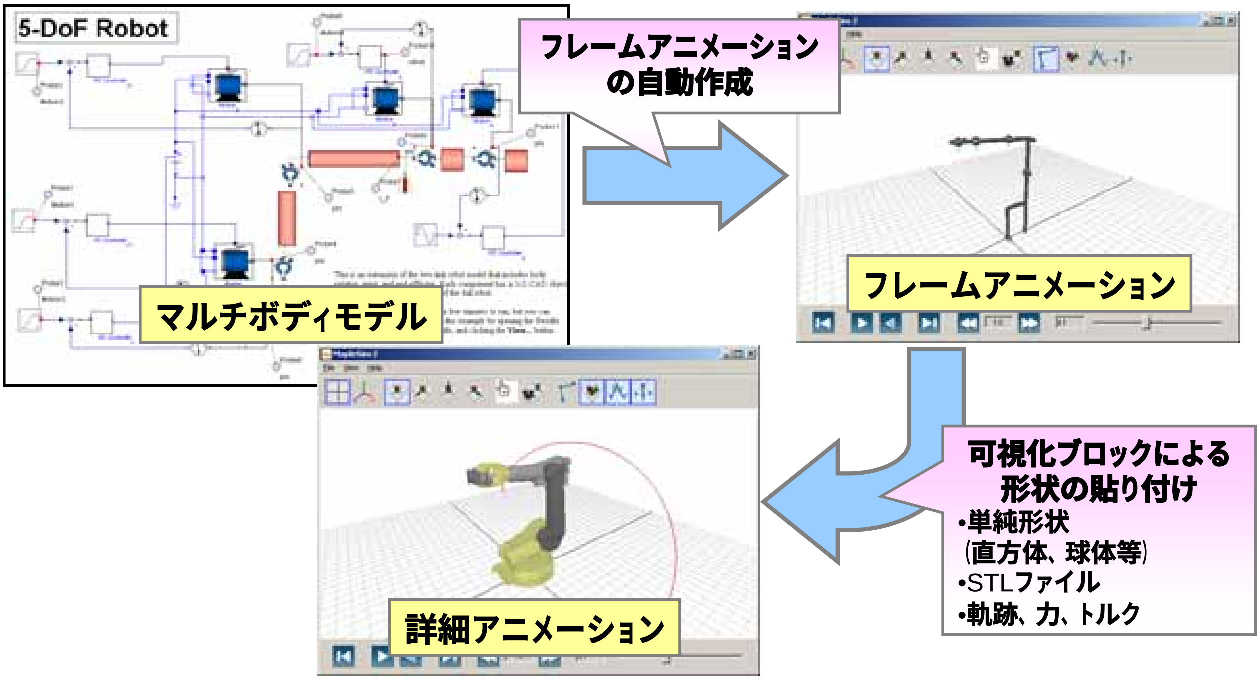
MapleSim とは

■ GUI操作による直感的モデル作成



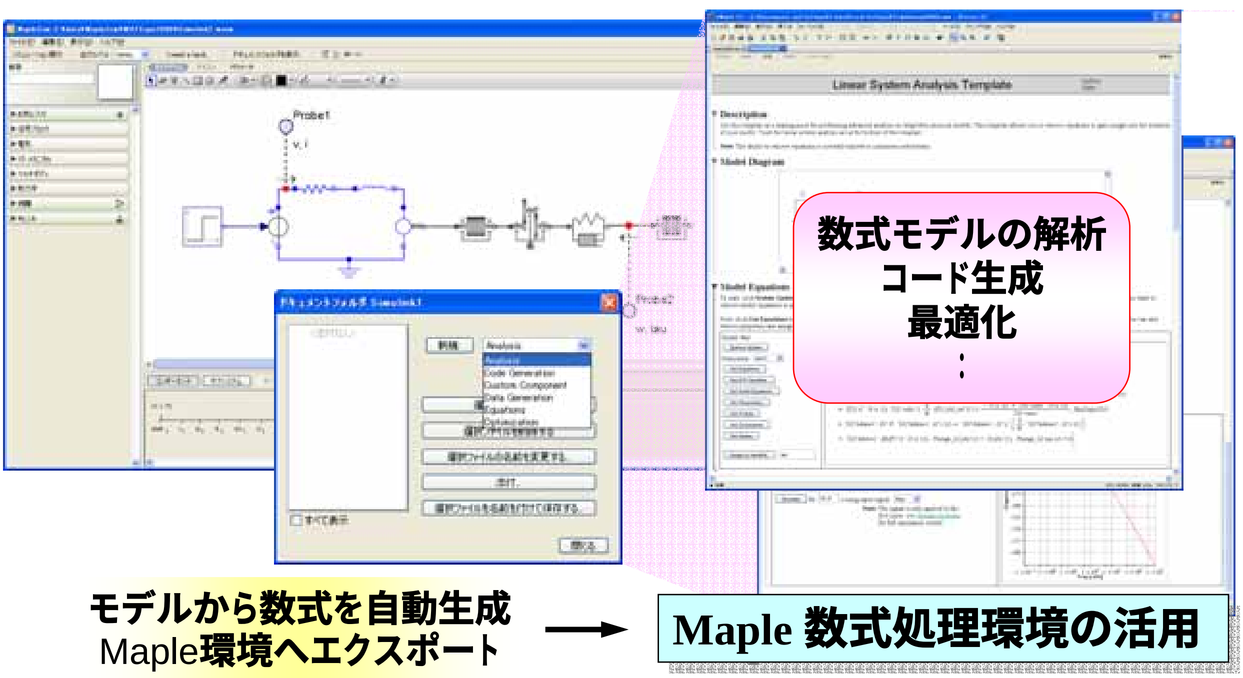
MapleSim とは

■ 可視化（マルチボディモデル）



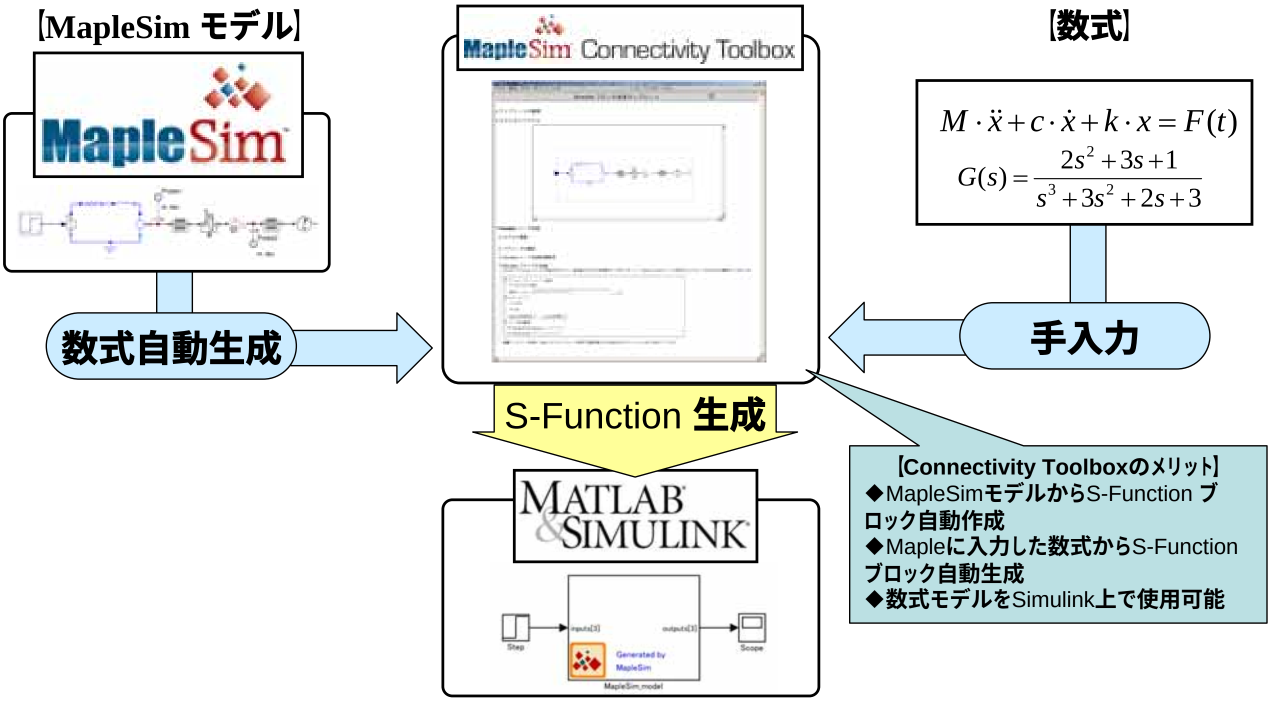
MapleSim とは

■ 数式の自動生成・Maple 環境との連携



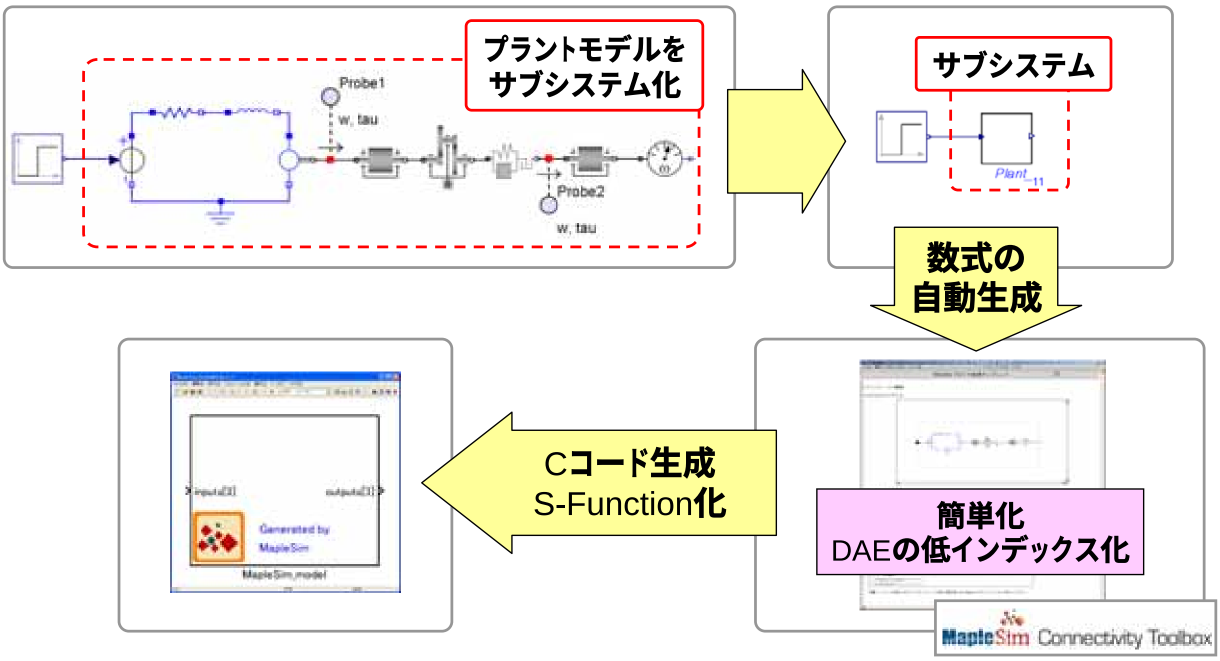
MapleSim とは

■ MapleSim Connectivity Toolboxについて



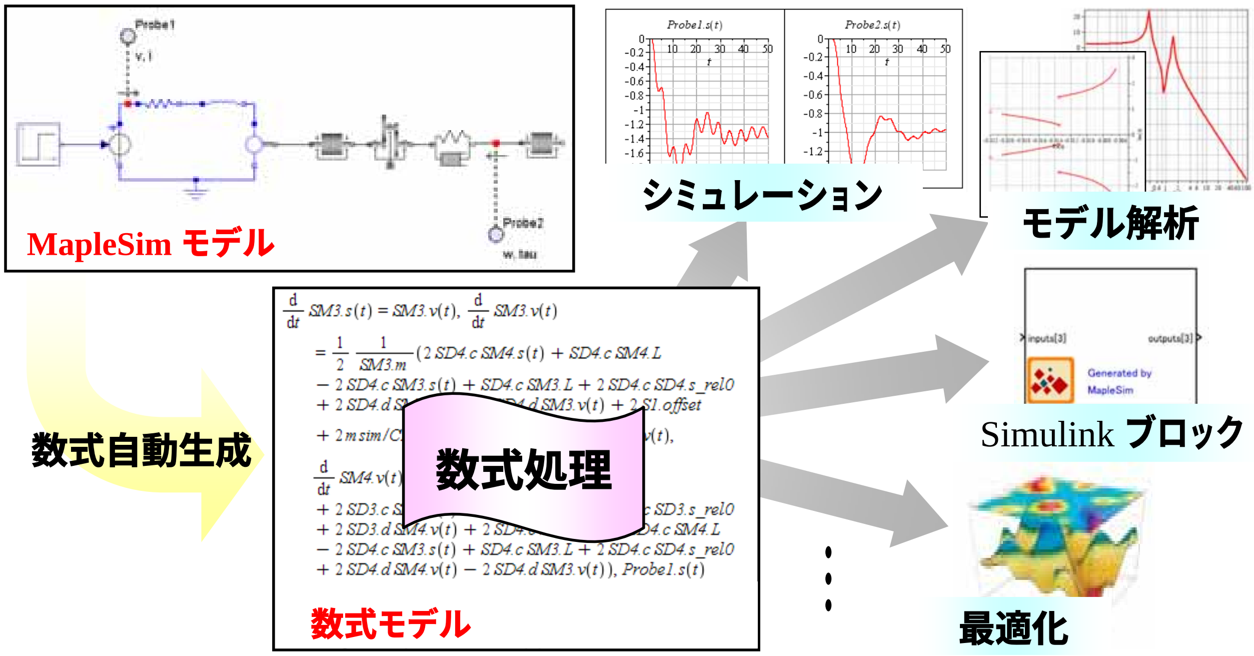
MapleSim とは

■ MapleSim Connectivity ToolboxによるSimulink ブロック生成



MapleSim とは

■ 数式ベースのモデリング･シミュレーション環境



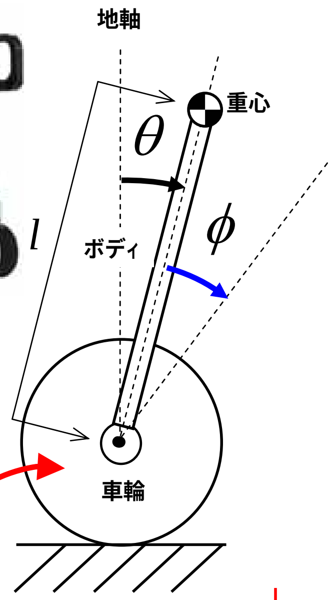
アジェンダ

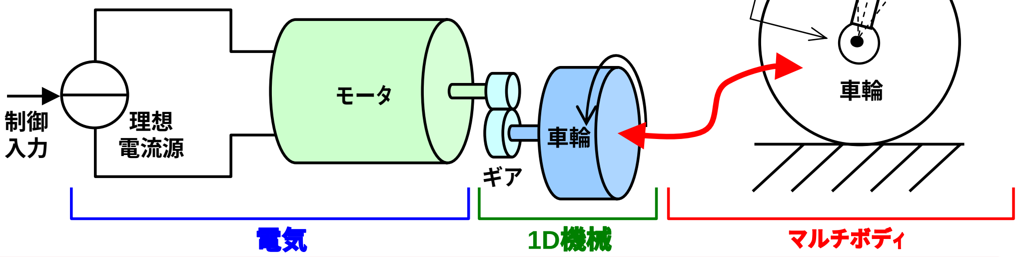
- ■ モデルベース開発とは
- ■ MapleSim によるプラントモデル開発
  - MapleSim とは
  - プラントモデルの作成
  - モデルの線形化と状態空間モデルの取得
- ■ コントローラ設計
  - MATLAB を用いた最適レギュレータの設計
  - シミュレーションによるコントローラの検証
- ■ Speedgoatによる実機検証
  - Speedgoatとは
  - 倒立二輪ロボット制御の実機検証
- ■ モデルベース開発の課題と方向性

プラントモデルの作成

■ e-nuvo WHEEL の概念モデル

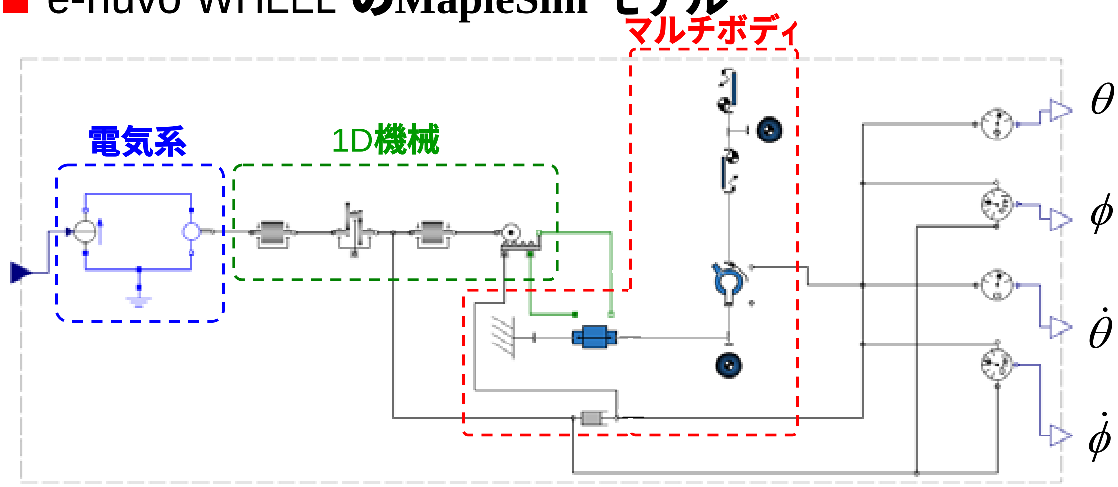
パラメータ名	変数	単位	値
ボディの質量	m	[kg]	0.5241
車輪の質量	M	[kg]	0.071
ボディの慣性モーメント	J_p	[kg･m²]	2.155E-03
車輪の慣性モーメント	J_t	[kg･m²]	8.632E-06
モータ (ロータ) の慣性モーメント	J_m	[kg･m²]	1.30E-07
車軸とボディ重心との距離	l	[m]	0.1204
車輪の半径	r_t	[m]	0.02485
車軸の粘性抵抗	c	[kgm²/s]	1.00E-04
モータのトルク/電流比	K_t	[N･m/A]	2.79E-03
ギア比	i	[-]	30





プラントモデルの作成

■ e-nuvo WHEEL のMapleSim モデル



数式自動生成

$$-3.368209256\,input(t) - 0.01119571714\left(\frac{d}{dt}outdtheta(t)\right) - 0.01119571714\left(\frac{d}{dt}outdphi(t)\right) + 0.03943661972\left(\frac{d}{dt}outphi(t)\right) - 0.06748752000\cos(outtheta(t))\left(\frac{d}{dt}outdtheta(t)\right) + 0.06748752000\sin(outtheta(t))\left(\frac{d}{dt}outtheta(t)\right)^2 = 0$$

$$-0.6620525712\sin(outtheta(t)) + 0.001677064872\cos(outtheta(t))\left(\frac{d}{dt}outdtheta(t) + \frac{d}{dt}outdphi(t)\right) - 0.0009800000000\left(\frac{d}{dt}outphi(t)\right) + 0.007450622209\left(\frac{d}{dt}outdtheta(t)\right)\cos(outtheta(t))^2 + 0.002650000000\left(\frac{d}{dt}outdtheta(t)\right) + 0.007450622209\left(\frac{d}{dt}outdtheta(t)\right)\sin(outtheta(t))^2 = 0$$

$$outdtheta(t) = \frac{d}{dt}outtheta(t)$$

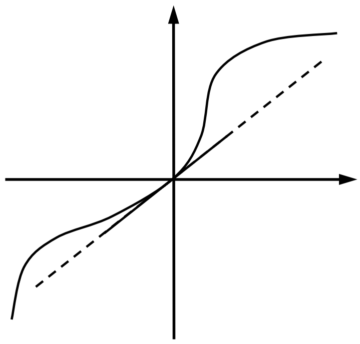
$$outdphi(t) = \frac{d}{dt}outphi(t)$$

アジェンダ

- モデルベース開発とは
- MapleSim によるプラントモデル開発
  - MapleSim とは
  - プラントモデルの作成
  - モデルの線形化と状態空間モデルの取得
- コントローラ設計
  - MATLAB を用いた最適レギュレータの設計
  - シミュレーションによるコントローラの検証
- Speedgoatによる実機検証
  - Speedgoatとは
  - 倒立二輪ロボット制御の実機検証
- モデルベース開発の課題と方向性

モデルの線形化と状態空間モデルの取得

- テイラー展開により非線形モデルを線形化



テイラー展開の1次近似線形化のイメージ

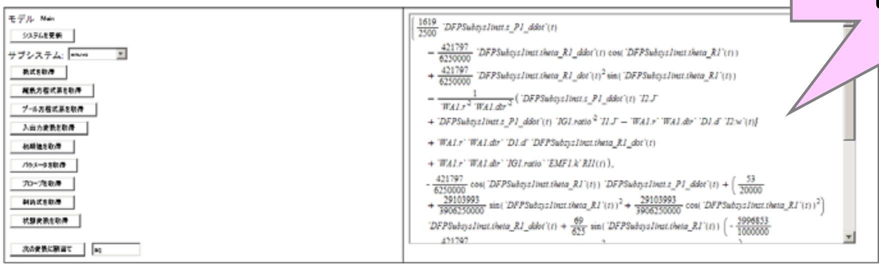
  - 非線形微分方程式の非線形項を
    一次テイラー展開することにより線形化

※線形化機能について
MapleSim2.0ではモデルの線形化機能が含まれていないため、本セミナーではマニュアルで線形化を行っています。
この機能は今後のバージョンアップで対応予定です。

モデルの線形化と状態空間モデルの取得

■ 数式の整理

▼モデルの数式



数式解析テンプレートを用いて
モデルの数式を取得

整理された数式

整理された数式
seq(print(systemeq[j]), j = 1 ..nops(systemeq))

$$-3.368209256\,\mathit{input}(t) - 0.01119571714\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) - 0.01119571714\left(\frac{d}{dt}\mathit{outdphi}(t)\right) + 0.03943661972\left(\frac{d}{dt}\mathit{outphi}(t)\right) - 0.06748752000\cos(\mathit{outtheta}(t))\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) + 0.06748752000\sin(\mathit{outtheta}(t))\left(\frac{d}{dt}\mathit{outtheta}(t)\right)^2 = 0$$

$$-0.6620525712\sin(\mathit{outtheta}(t)) + 0.001677064872\cos(\mathit{outtheta}(t))\left(\frac{d}{dt}\mathit{outdtheta}(t) + \frac{d}{dt}\mathit{outdphi}(t)\right) - 0.0009800000000\left(\frac{d}{dt}\mathit{outphi}(t)\right) + 0.007450622209\left(\frac{d}{dt}\mathit{outdtheta}(t)\right)\cos(\mathit{outtheta}(t))^2 + 0.002650000000\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) + 0.007450622209\left(\frac{d}{dt}\mathit{outdtheta}(t)\right)\sin(\mathit{outtheta}(t))^2 = 0$$

$$\mathit{outdtheta}(t) = \frac{d}{dt}\mathit{outtheta}(t)$$

$$\mathit{outdphi}(t) = \frac{d}{dt}\mathit{outphi}(t)$$

モデルの線形化と状態空間モデルの取得

■ 非線形微分方程式の線形化

▼ 非線形システムの線形化
※モデルの自動線形化機能は次期バージョンで追加予定です.
本セミナーでは標準的なMapleコマンドを用いてモデルの線形化を行います.
seq(print(systemeq[j]), j = 1 ..2) :

非線形
微分方程式

$$-3.368209256\,\mathit{input}(t) - 0.01119571714\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) - 0.01119571714\left(\frac{d}{dt}\mathit{outdphi}(t)\right) + 0.03943661972\left(\frac{d}{dt}\mathit{outphi}(t)\right) - 0.06748752000\boxed{\cos(\mathit{outtheta}(t))}\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) + 0.06748752000\boxed{\sin(\mathit{outtheta}(t))}\boxed{\left(\frac{d}{dt}\mathit{outtheta}(t)\right)^2} = 0$$

$$-0.6620525712\boxed{\sin(\mathit{outtheta}(t))} + 0.001677064872\boxed{\cos(\mathit{outtheta}(t))}\left(\frac{d}{dt}\mathit{outdtheta}(t) + \frac{d}{dt}\mathit{outdphi}(t)\right) - 0.0009800000000\left(\frac{d}{dt}\mathit{outphi}(t)\right) + 0.007450622209\left(\frac{d}{dt}\mathit{outdtheta}(t)\right)\boxed{\cos(\mathit{outtheta}(t))^2} + 0.002650000000\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) + 0.007450622209\left(\frac{d}{dt}\mathit{outdtheta}(t)\right)\boxed{\sin(\mathit{outtheta}(t))^2} = 0$$

```
nlterm := [sin(temptheta), cos(temptheta), cos(temptheta)^2, sin(temptheta)^2]
dnlterm := [dtemptheta^2]
linterm := map(a -> convert(taylor(a, temptheta = 0, 2), polynom), nlterm)
```
[temptheta, 1, 1, 0]
```
dlinterm := map(a -> convert(taylor(a, dtemptheta = 0, 2), polynom), dnlterm)
```
[0]

非線形項の線形化

```
linearize := [seq(nlterm[j] = linterm[j], j = 1 ..nops(nlterm)), seq(dnlterm[j] = dlinterm[j], j = 1 ..nops(dnlterm))] :
linearize := eval(linearize, [temptheta = outtheta(t), dtemptheta = diff(outtheta(t), t)]) :
linearizedeq := eval(systemeq, linearize) :
```

線形化された
微分方程式

線形化された数式
seq(print(linearizedeq[j]), j = 1 ..2)

$$-3.368209256\,\mathit{input}(t) - 0.07868323714\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) - 0.01119571714\left(\frac{d}{dt}\mathit{outdphi}(t)\right) + 0.03943661972\left(\frac{d}{dt}\mathit{outphi}(t)\right) = 0$$

$$-0.6620525712\,\mathit{outtheta}(t) + 0.01177768708\left(\frac{d}{dt}\mathit{outdtheta}(t)\right) + 0.001677064872\left(\frac{d}{dt}\mathit{outdphi}(t)\right) - 0.0009800000000\left(\frac{d}{dt}\mathit{outphi}(t)\right) = 0$$

モデルの線形化と状態空間モデルの取得

■ 数式オブジェクトの作成と状態空間オブジェクトへの変換

▼ 状態空間モデルの取得

with(DynamicSystems)

[AlgEquation, BodePlot, CharacteristicPolynomial, Chirp, Coefficients, ControllabilityMatrix, Controllable, DiffEquation, DiscretePlot, FrequencyResponse, GainMargin, Grammians, ImpulseResponse, ImpulseResponsePlot, IsSystem, MagnitudePlot, NewSystem, ObservabilityMatrix, Observable, PhaseMargin, PhasePlot, PrintSystem, Ramp, ResponsePlot, RootContourPlot, RootLocusPlot, RouthTable, SSModelReduction, SSTransformation, Simulate, Sinc, Sine, Square, StateSpace, Step, System, SystemOptions, ToDiscrete, TransferFunction, Triangle, Verify, ZeroPoleGain, ZeroPolePlot]

Dynamic Systems **パッケージの読み込み**

deobj := *DiffEquation*(*linearizedeq*, [*input*(*t*)], [*outtheta*(*t*), *outphi*(*t*), *outdtheta*(*t*), *outdphi*(*t*)])

Diff. Equation
continuous
4 output(s); 1 input(s)
inputvariable = [*input*(*t*)]
outputvariable = [*outtheta*(*t*), *outphi*(*t*), *outdtheta*(*t*), *outdphi*(*t*)]

数式オブジェクト

tfobj := *TransferFunction*(*deobj*)

Transfer Function
continuous
4 output(s); 1 input(s)
inputvariable = [*input*(*s*)]
outputvariable = [*outtheta*(*s*), *outphi*(*s*), *outdtheta*(*s*), *outdphi*(*s*)]

伝達関数オブジェクト

ssobj := *StateSpace*(*deobj*)

State Space
continuous
4 output(s); 1 input(s); 4 state(s)
inputvariable = [*input*(*t*)]
outputvariable = [*outtheta*(*t*), *outphi*(*t*), *outdtheta*(*t*), *outdphi*(*t*)]
statevariable = [*x1*(*t*), *x2*(*t*), *x3*(*t*), *x4*(*t*)]

状態空間オブジェクト

モデルの線形化と状態空間モデルの取得

■ 数式オブジェクトの作成と状態空間オブジェクトへの変換

取得した伝達関数

trfc := *tfobj*:-*tf*

$$\left[\begin{array}{c}
\dfrac{-1.765220445\,10^{19}\,s + 1.031514085\,10^{19}}{3.038747238\,10^{14}\,s^3 + 1.210508107\,10^{18}\,s^2 + 2.316297912\,10^{19}\,s - 8.159098589\,10^{19}} \\
\dfrac{6.198392910\,10^{14}\,s^2 - 3.484268122\,10^{16}}{1.519373619\,10^{9}\,s^4 + 6.052540535\,10^{12}\,s^3 + 1.158148956\,10^{14}\,s^2 - 4.079549294\,10^{14}\,s} \\
\dfrac{-1.765220445\,10^{19}\,s^2 + 1.031514085\,10^{19}\,s}{3.038747238\,10^{14}\,s^3 + 1.210508107\,10^{18}\,s^2 + 2.316297912\,10^{19}\,s - 8.159098589\,10^{19}} \\
\dfrac{6.198392912\,10^{14}\,s^2 - 3.484268122\,10^{16}}{1.519373619\,10^{9}\,s^3 + 6.052540535\,10^{12}\,s^2 + 1.158148956\,10^{14}\,s - 4.079549294\,10^{14}}
\end{array}\right]$$

取得した伝達関数

取得した状態空間マトリクス

SSA := *ssobj*:-*a*

$$\begin{bmatrix} -3983.5761641065122259 & 0 & 0 & 5.35710477181344457 58\,10^5 \\ 567.31815036127950179 & 0 & 0 & -76225.422205052366150 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

SSB := *ssobj*:-*b*

$$\begin{bmatrix} 4.0795712350311584877\,10^5 \\ -58090.400641654776629 \\ 0 \\ 0 \end{bmatrix}$$

SSC := *ssobj*:-*c*

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

SSD := *ssobj*:-*d*

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

取得した
状態空間マトリクス

アジェンダ

- モデルベース開発とは
- MapleSim によるプラントモデル開発
  - MapleSim とは
  - プラントモデルの作成
  - モデルの線形化と状態空間モデルの取得
- **コントローラ設計**
  - **MATLAB を用いた最適レギュレータの設計**
  - シミュレーションによるコントローラの検証
- Speedgoatによる実機検証
  - Speedgoatとは
  - 倒立二輪ロボット制御の実機検証
- モデルベース開発の課題と方向性

MATLAB を用いた最適レギュレータの設計

- **状態空間マトリクスをMATLAB へ出力**

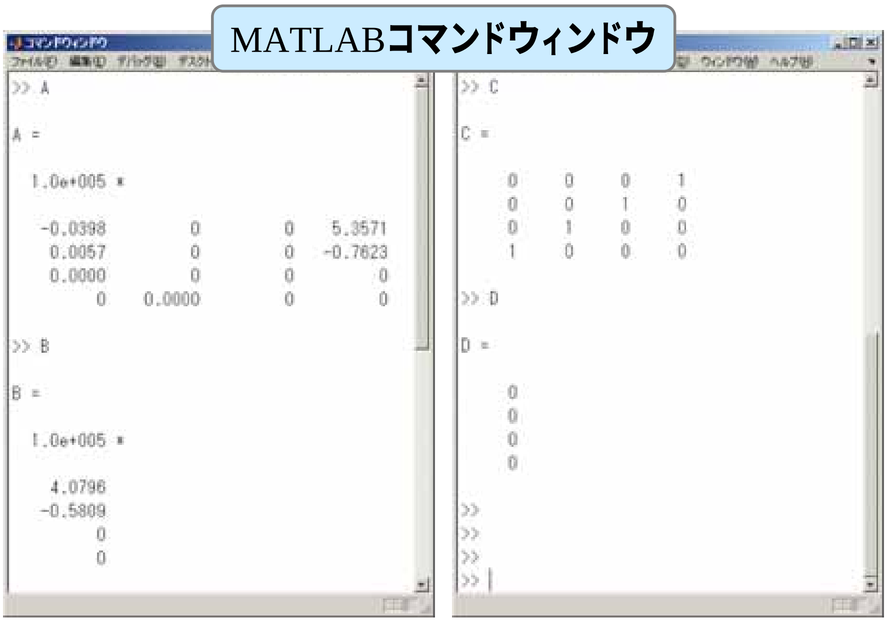
MATLAB パッケージ読込

▼ 状態空間モデルのMATLABへの出力

```
with(Matlab)
    [AddTranslator, FromMFile, FromMatlab, chol, closelink, defined, det, dimensions, eig, evalM, fft, getvar, inv, lu, ode15s, ode45, openlink, qr, setvar, size, square, transpose]
evalM("desktop")
setvar("A", SSA)
setvar("B", SSB)
setvar("C", SSC)
setvar("D", SSD)
```

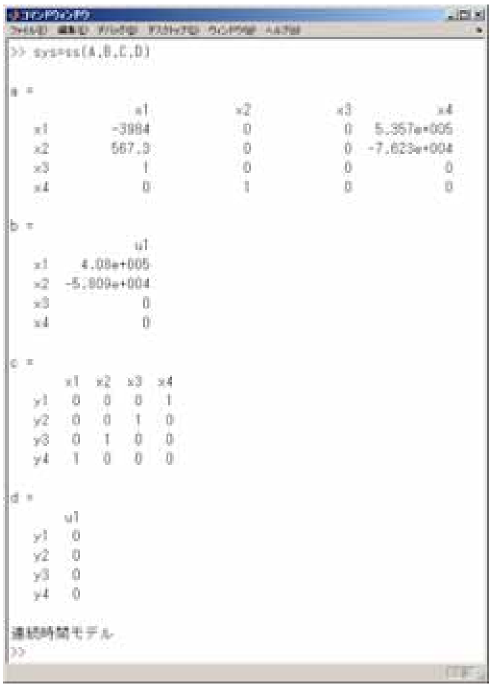
A,B,C,D行列を MATLAB に出力

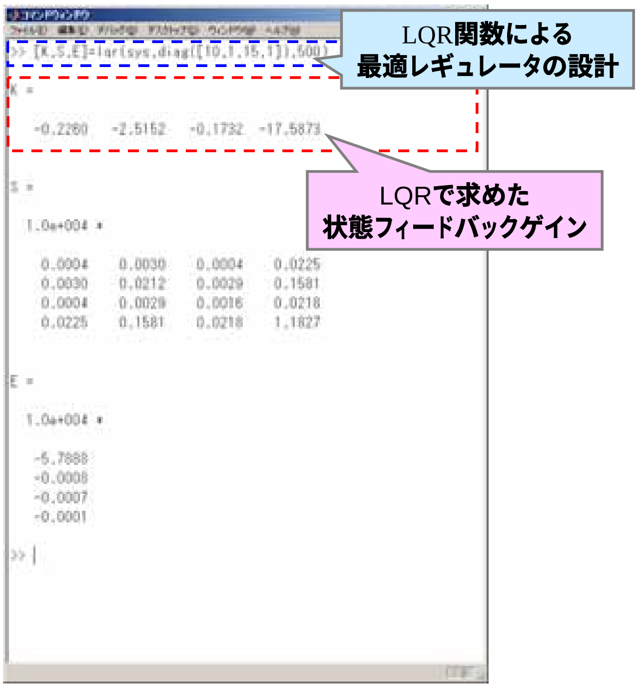
MATLABコマンドウィンドウ



MATLAB を用いた最適レギュレータの設計

■ 状態空間オブジェクトの作成と最適レギュレータの設計





アジェンダ

■ モデルベース開発とは
■ MapleSim によるプラントモデル開発
- MapleSim とは
- プラントモデルの作成
- モデルの線形化と状態空間モデルの取得

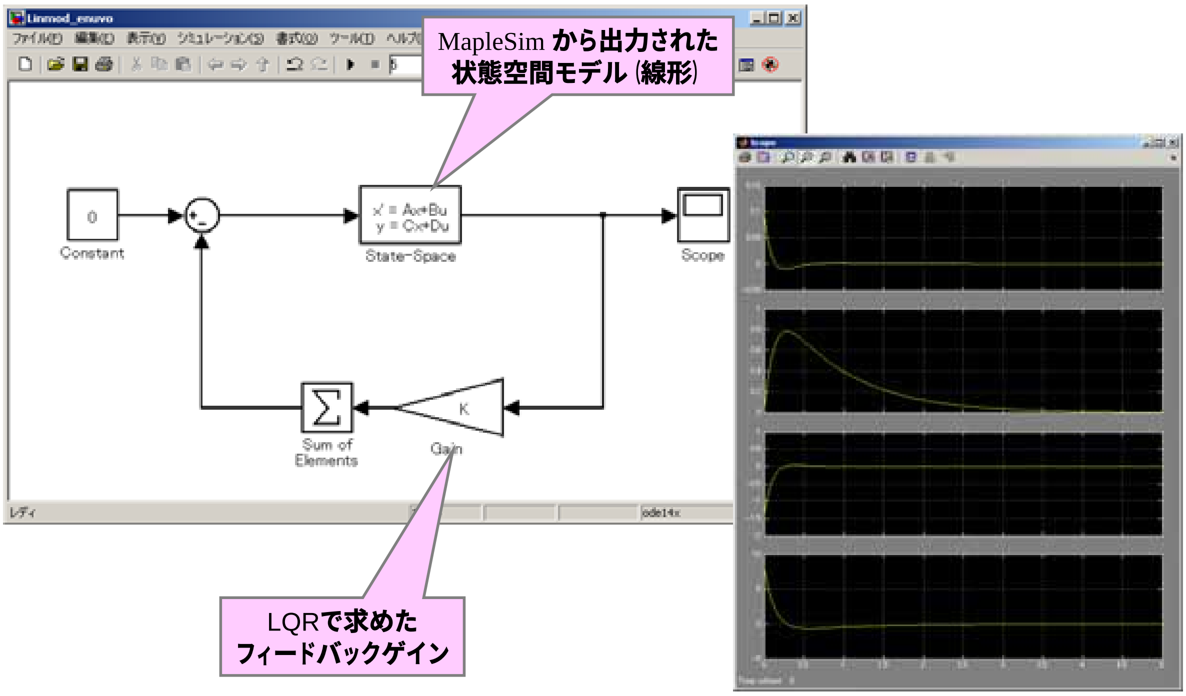
■ **コントローラ設計**
- MATLAB を用いた最適レギュレータの設計
- **シミュレーションによるコントローラの検証**

■ Speedgoatによる実機検証
- Speedgoatとは
- 倒立二輪ロボット制御の実機検証

■ モデルベース開発の課題と方向性

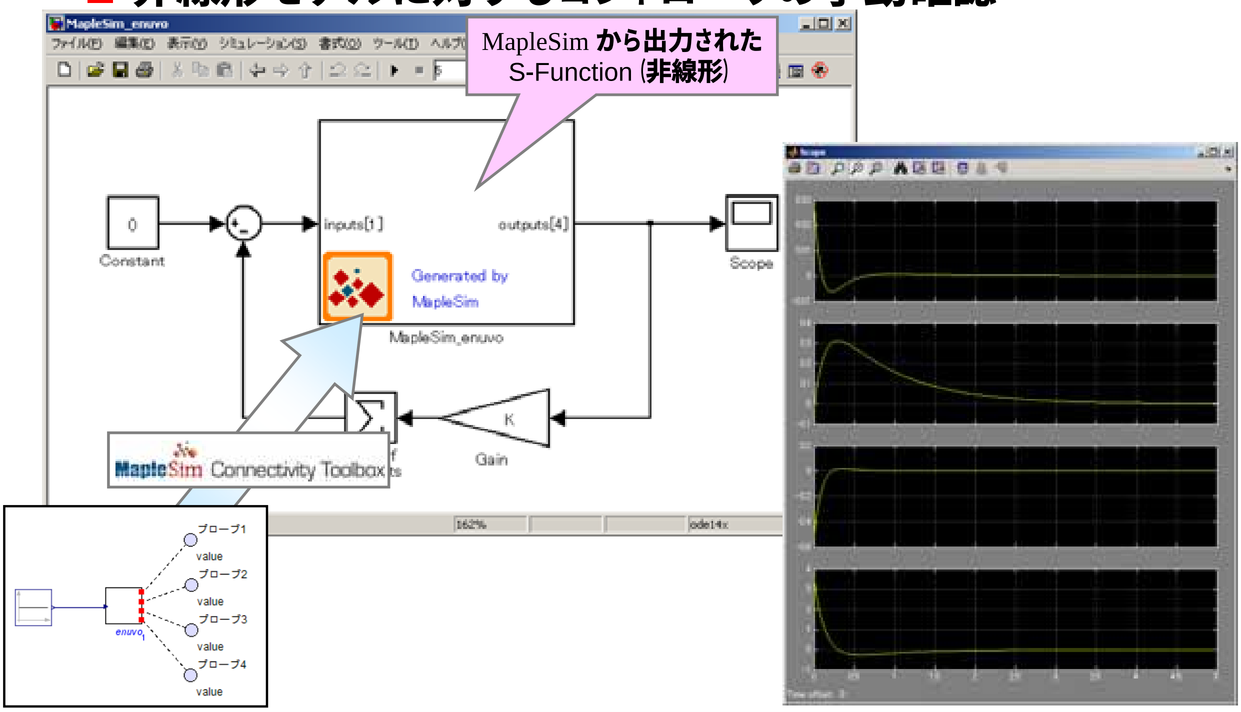
シミュレーションによるコントローラの検証

■ 線形モデルに対するコントローラの挙動確認



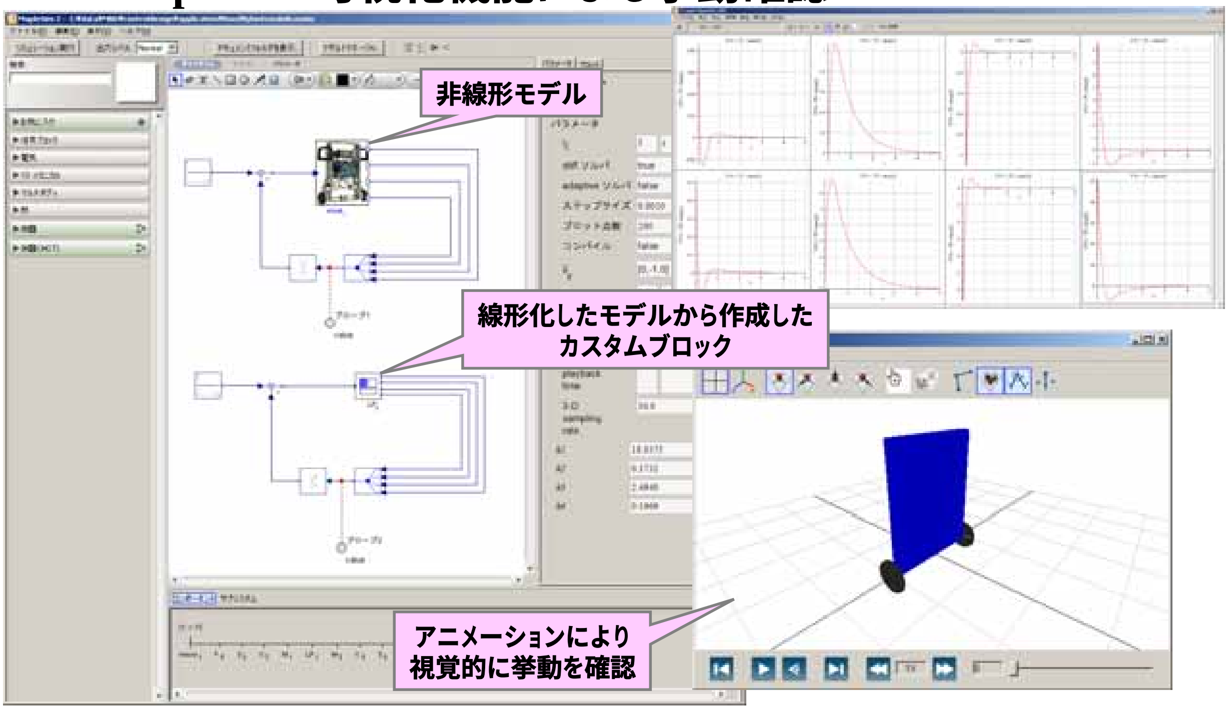
シミュレーションによるコントローラの検証

■ 非線形モデルに対するコントローラの挙動確認



シミュレーションによるコントローラの検証

■ MapleSim 可視化機能による挙動確認

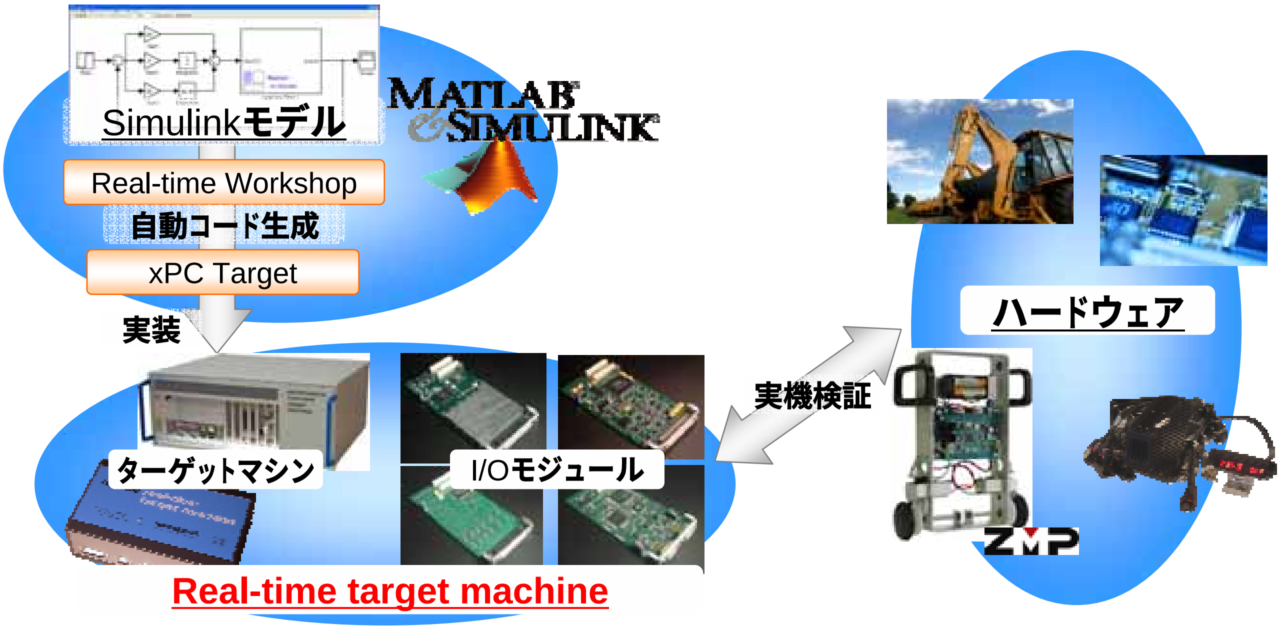


アジェンダ

- ■ モデルベース開発とは
- ■ MapleSim によるプラントモデル開発
  - ● MapleSim とは
  - ● プラントモデルの作成
  - ● モデルの線形化と状態空間モデルの取得
- ■ コントローラ設計
  - ● MATLAB を用いた最適レギュレータの設計
  - ● シミュレーションによるコントローラの検証
- ■ **Speedgoatによる実機検証**
  - ● **Speedgoatとは**
  - ● 倒立二輪ロボット制御の実機検証
- ■ モデルベース開発の課題と方向性

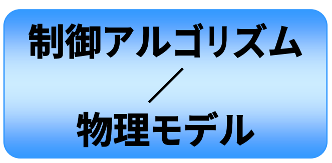
Speedgoatとは

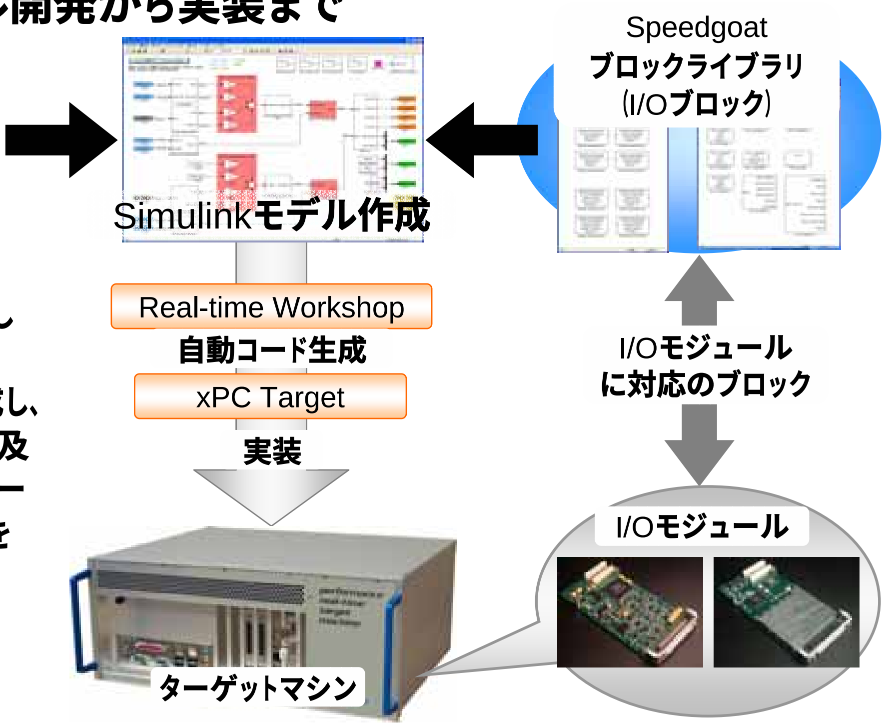
Speedgoat社は、モデルベース開発のためのSimulinkモデルを用いた実機検証用ハードウェアを提供します。



Speedgoatとは

Simulinkモデル開発から実装まで





- I/Oモジュールに対応したブロックを使用してSimulinkモデルを作成し、Real-Time Workshop及びxPC Targetによりターゲットマシンにモデルを実装します。

Speedgoatとは

製品体系

ハード

ターゲットマシン

- Performance

- Mobile

- Modular

I/Oモジュール

- IO10x (アナログ)
- IO20x (デジタル)
- IO30x (パルストレイン)
- IO40x (エンコーダ)
- IO50x (シリアル通信)
- IO60x (CAN) etc...

ソフト

Simulinkブロックライブラリ

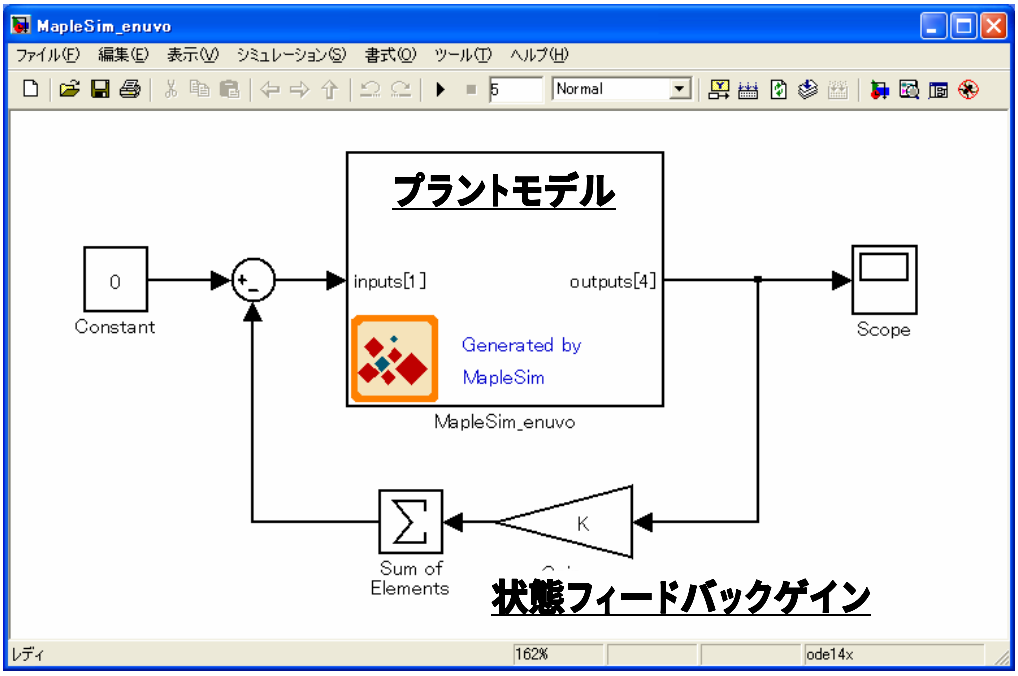
Interface Designer for xPC Target

アジェンダ

- モデルベース開発とは
- MapleSim によるプラントモデル開発
  - MapleSim とは
  - プラントモデルの作成
  - モデルの線形化と状態空間モデルの取得
- コントローラ設計
  - MATLAB を用いた最適レギュレータの設計
  - シミュレーションによるコントローラの検証
- **Speedgoatによる実機検証**
  - Speedgoatとは
  - **倒立二輪ロボット制御の実機検証**
- モデルベース開発の課題と方向性

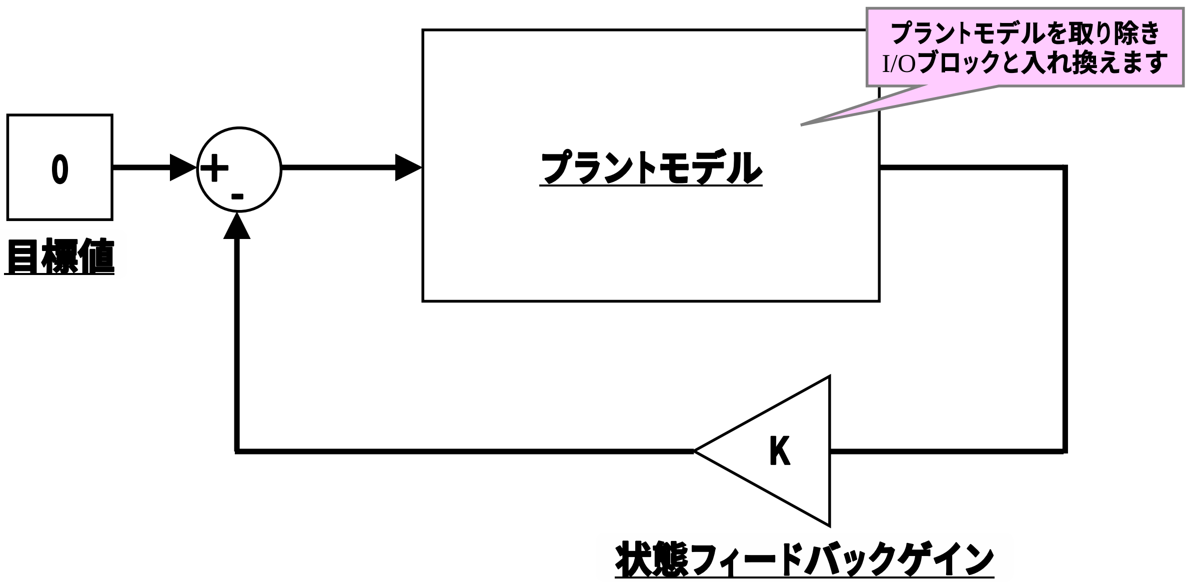
倒立二輪ロボット制御の実機検証

コントローラ設計により得られたSimulinkモデル



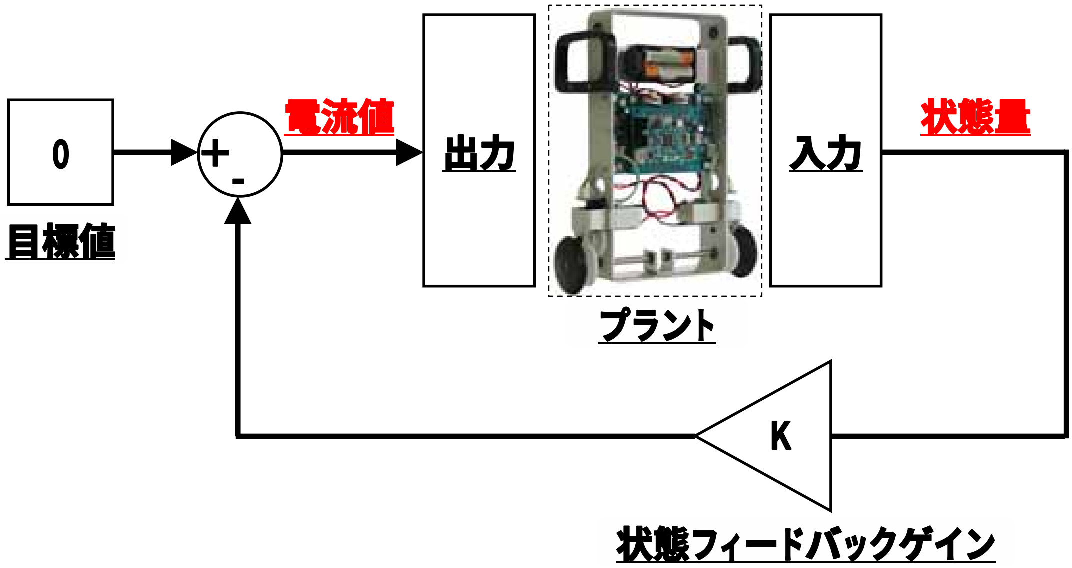
倒立二輪ロボット制御の実機検証

プラントモデルに換えて、実際の倒立二輪ロボットと接続するために、ターゲットマシンのI／Oブロックを配置します。



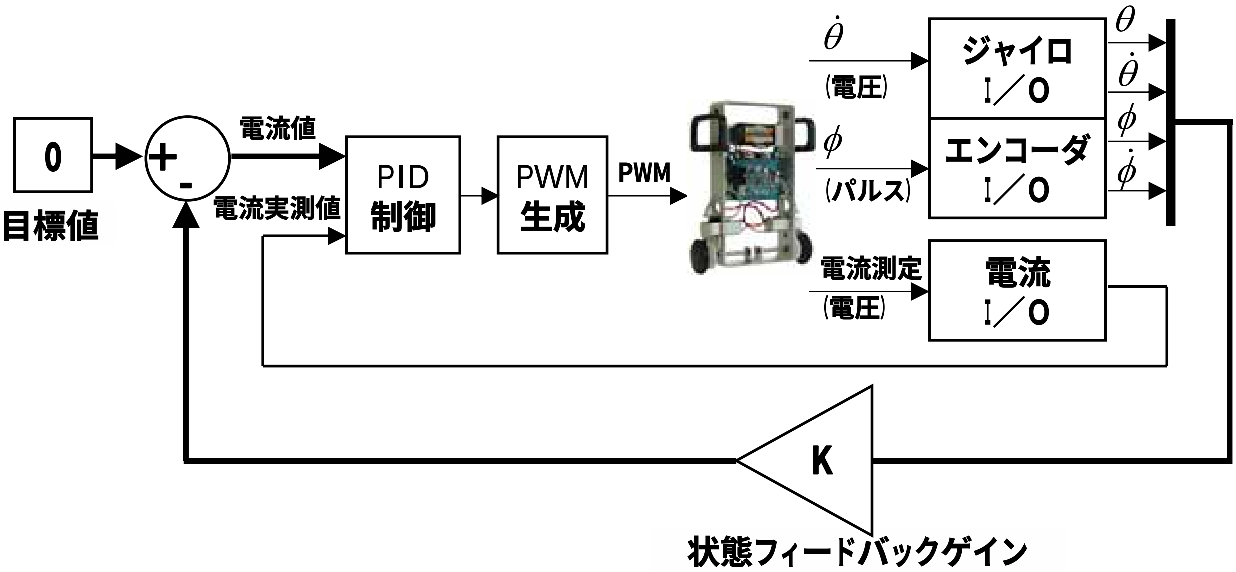
倒立二輪ロボット制御の実機検証

入力ブロックは、状態量を出力し、出力ブロックは、目標となる電流値が入力されます。



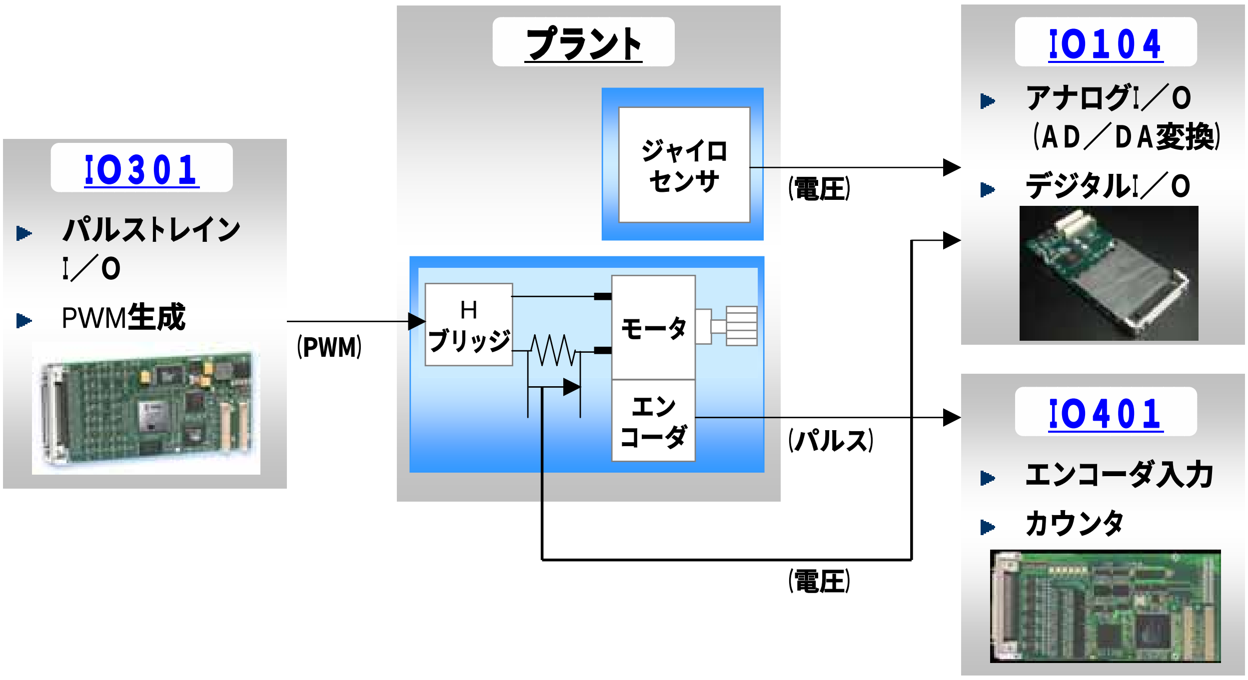
倒立二輪ロボット制御の実機検証

実際に実装するモデルでは、理想電流源を実現する為に、PIDによる電流制御を行っています。



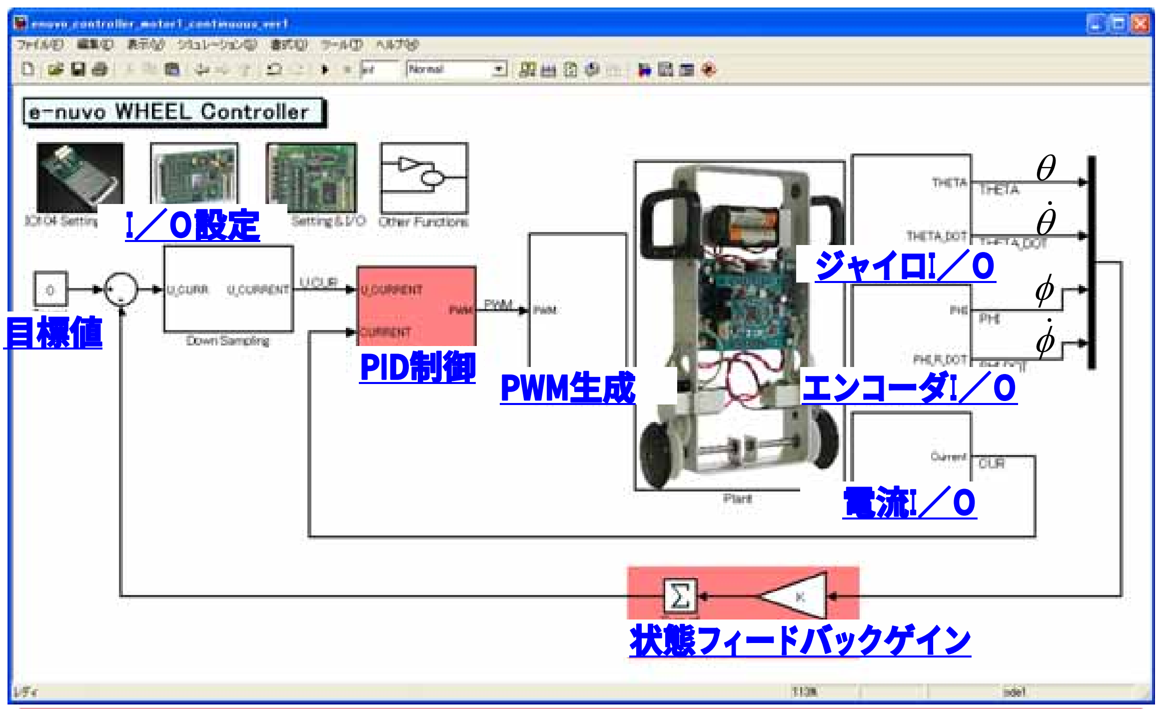
倒立二輪ロボット制御の実機検証

必要となるI／Oに合せて、I／Oモジュールを選択します。

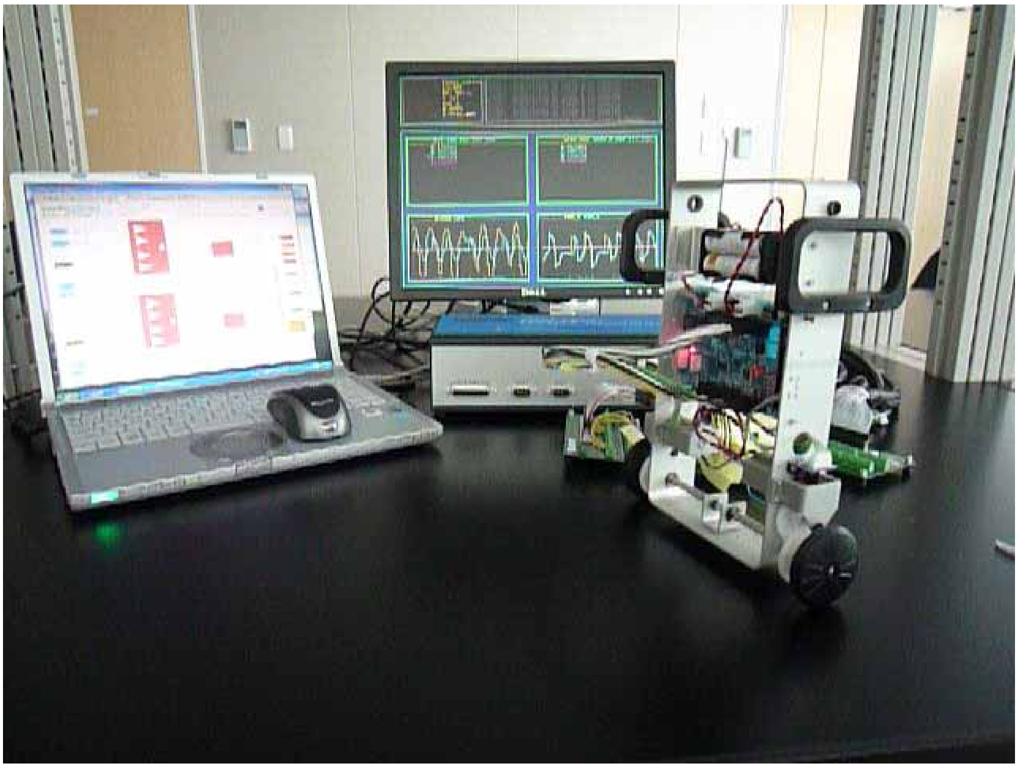


倒立二輪ロボット制御の実機検証

作成したSimulinkモデルは以下のモデルになります。



倒立二輪ロボットの実機検証



アジェンダ

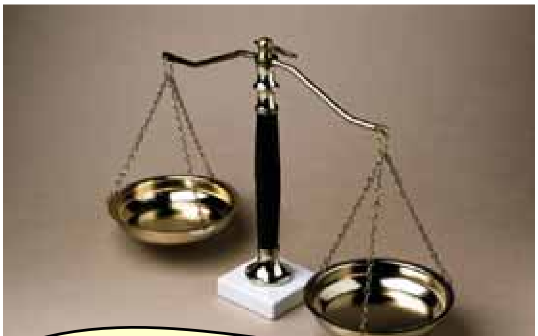
- モデルベース開発とは
- MapleSim によるプラントモデル開発
  - MapleSim とは
  - プラントモデルの作成
  - モデルの線形化と状態空間モデルの取得
- コントローラ設計
  - MATLAB を用いた最適レギュレータの設計
  - シミュレーションによるコントローラの検証
- Speedgoatによる実機検証
  - Speedgoatとは
  - 倒立二輪ロボット制御の実機検証
- **モデルベース開発の課題と方向性**

モデルベース開発の課題と方向性

■ どこまで詳細にモデル化するのか、してはいけないのか

モデルは詳細であればあるほど良い.

- モデルの詳細度
- シミュレーション計算コスト
- 算出されるコントローラの次数
- モデルの見通し



何事もバランスが大事

モデルベース開発の課題と方向性

■ モデルベース開発は役に立つ？

モデル化誤差

シミュレーションだけでは
設計できない

詳細なモデルで多数のシミュレーションを繰り返し、
最適な制御パラメータを算出した.

シミュレーションばかりに目が行きがちですが…

モデルベース開発の本質的なメリット

- □ 簡略化されたモデルによって、
  プラントのカラクリ（本質的な仕組み）を理解することができる.
- □ 数理的な工学理論を実設計に適用することができる.
- □ プラントの本質を理解した上で適切な設計仕様を設定.
  →限界性能を把握することができる.
- □ 設計レベルの人依存性を低減.

モデルベース開発の課題と方向性

■ MapleSim、Speedgoatによって…

Maple

MapleSim

High-Performance Multi-Domain Modeling and Simulation

数式自動生成による
見通しの良いホワイトボックスモデル

モデルベース開発の
本質的なメリットを享受

speedgoat

simulation and real-time systems

開発した数理モデルを
即時に実機に接続可能

工学理論に基づく設計を
実際の製品に適用

各製品へのお問合せは

つくる情熱を、支える情熱。

CYBERNET

サイバネットシステム株式会社
アドバンスドソリューション統括部
モデルベース開発推進室

TEL: 03-5297-3255　FAX: 03-5297-3637

E-Mail(Maple): infomaple@cybernet.co.jp
E-Mail(Speedgoat) ecsales@cybernet.co.jp

Web(Maple): www.cybernet.co.jp/Maple
Web(Speedgoat):
www.cybernet.co.jp/matlab-thirdparty/product/speedgoat/

※製品価格、評価版、ご利用形態、ご契約内容の確認などは上記窓口までお気軽にお問合せください。
※お客様の業務内容に合わせた利用方法のご提案もご用意しております。詳しくは担当営業までご相談ください。