

# Simulink基礎

MathWorks Japan  
カスタマー サクセス部（教育機関）  
カスタマーサクセスエンジニア  
沖田 芳雄

# 概要

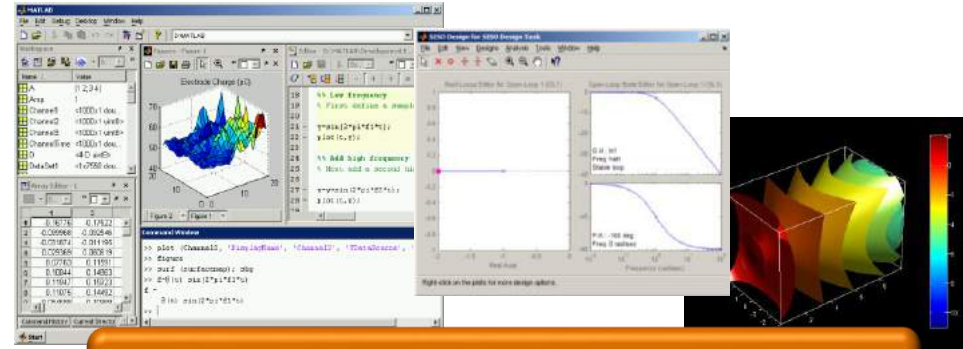
1. Simulinkとは
2. Simulinkハンズオントレーニング
  - sin関数の積分シミュレーション
  - 自由落下のモデリングとシミュレーション
  - 1自由度振動系のモデリングとシミュレーション

- 
- 可能

# MATLAB®とSimulinkの関係は？

## MATLAB

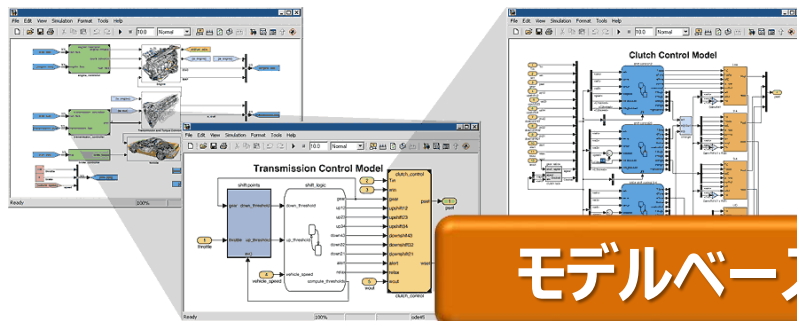
- 容易なデータ操作
- 簡潔なプログラミング言語
- 豊富な数学関数・ファイルI/O
- 2次元/3次元可視化機能



技術計算環境

## Simulink

- ブロック線図モデリング
  - 豊富なブロックライブラリ
- ⇒MATLAB上で動作



モデルベースデザイン 環境

## Stateflow®

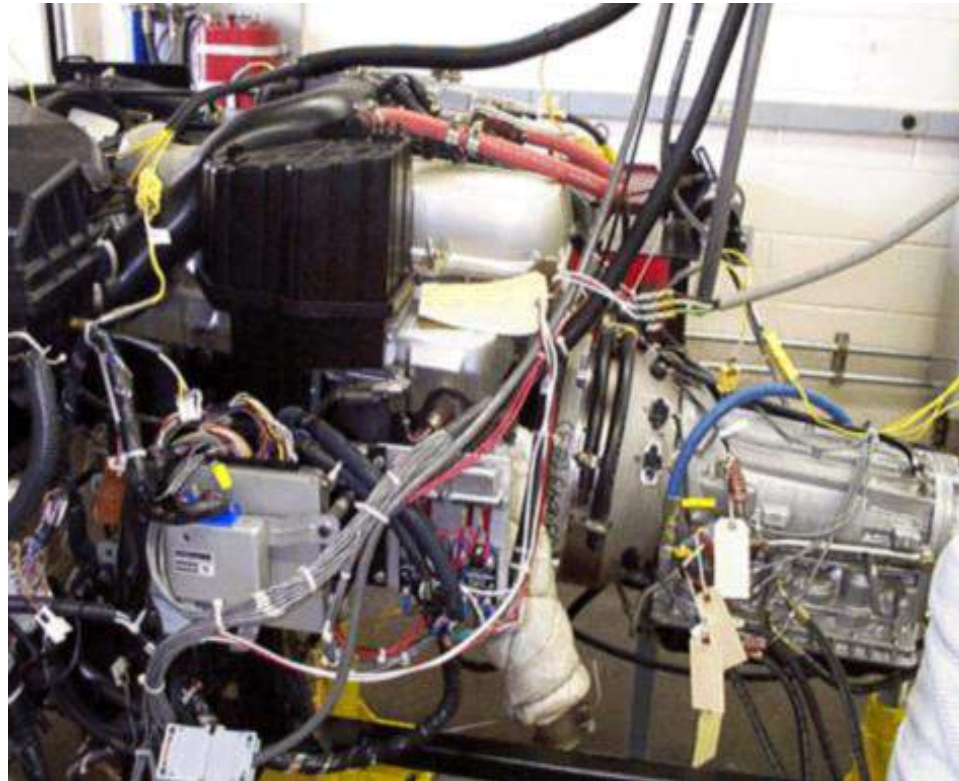
- フローチャート・状態遷移図を用いたモデリング環境
- ⇒Simulink上で動作





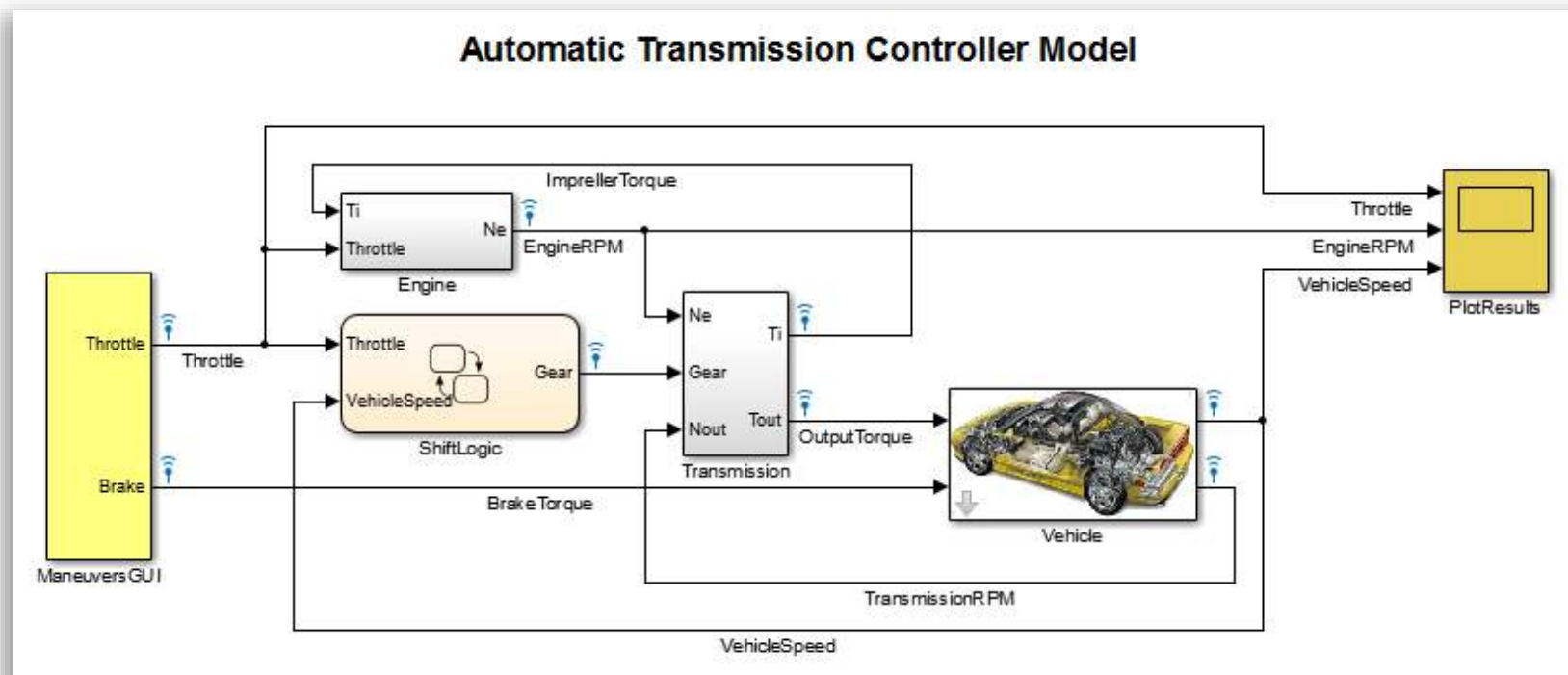
# Simulink

## 複雑な事をシンプルに



Credit: SwRI

# Simulinkによるモデル化

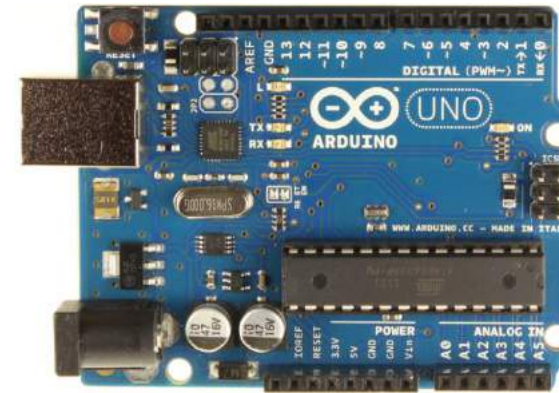


# ローコストハードウェアへの実装

ローコストハードウェア機器

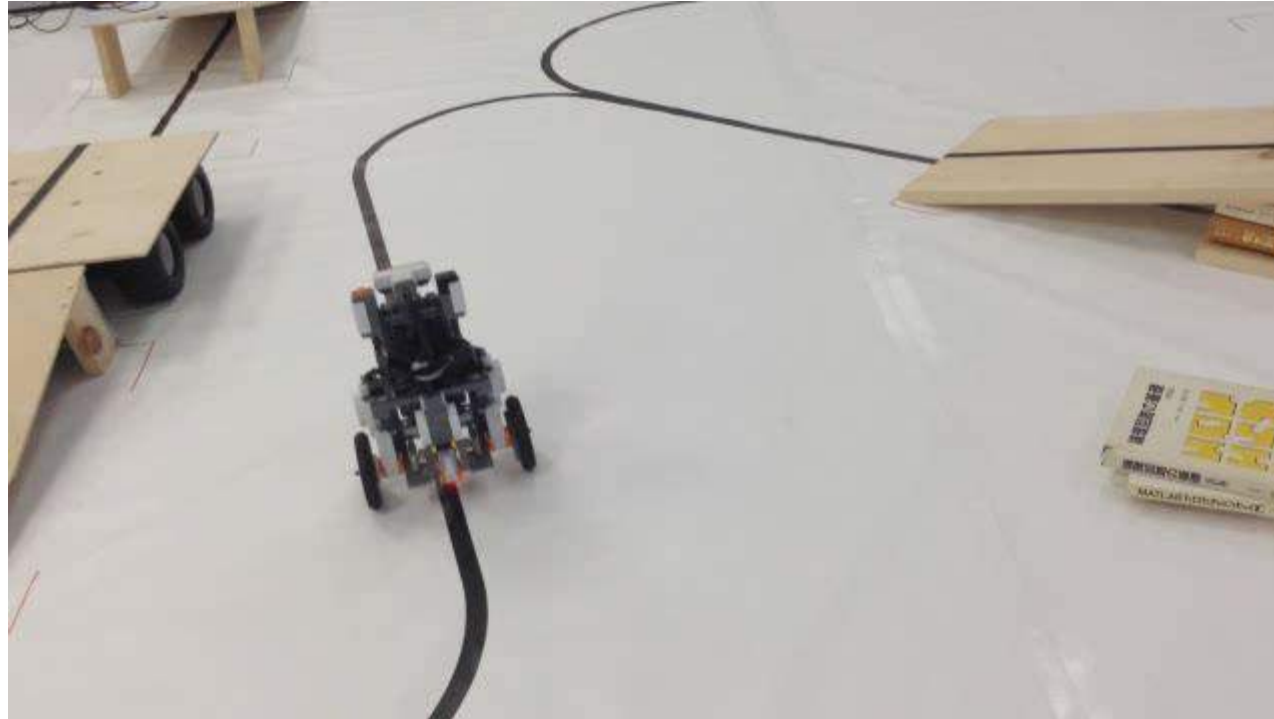


Arduinoなどのプログラマブルデバイス



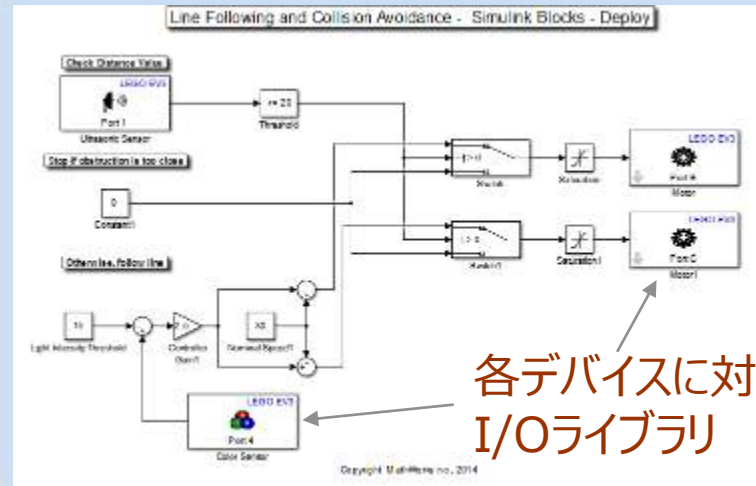
```
BOOL CMyMfc29BAuto::DisplayDialog()
{
    // TODO: Add your dispatch handler code here
    TRACE("Entering CMyMfc29BAuto::DisplayDialog %p\n", this);
    BOOL bRet = TRUE;
    AfxLockTempMaps(); // See MFC Tech Note #3
    CWnd* pTopWnd = CWnd::FromHandle(::GetTopWindow(NULL));
    try
    {
        CPromptDlg dlg /*(pTopWnd)*/;
        if (m_vaTextData.vt == VT_BSTR)
        {
            // converts double-byte character to single-byte character
            dlg.m_strData = m_vaTextData.bstrVal;
        }
        dlg.m_lData = m_lData;
        if (dlg.DoModal() == IDOK)
        {
            m_vaTextData = COleVariant(dlg.m_strData).Detach();
            m_lData = dlg.m_lData;
            bRet = TRUE;
        }
        else
        {
            bRet = FALSE;
        }
    }
    catch (CException* pe)
    {
        TRACE("Exception: failure to display dialog\n");
        bRet = FALSE;
        pe->Delete();
    }
    AfxUnlockTempMaps();
    return bRet;
}
```

# Simulinkモデルからの自動コード実装





# ハードウェア連携機器の例



アイデアを直ぐに実装

コード自動生成

各デバイスに対応した  
I/Oライブラリ



**PARROT® Minidrones**



**Arduino® Nano, Zero  
Uno, Mega 2560**



**Raspberry Pi™  
1/2/3**



**Apple® iOS**



**Android™ Devices**

# ハードウェア連携機能

## ハードウェアサポートパッケージの入手方法

ホーム プロット アプリ エディター パブリッシュ 表示

新規スクリプト 新規ライブスクリプト 新規作成 開く 比較 ファイルの検索 データのインポート ワークスペースの保存 新規変数 変数を開く ワークスペースのクリア コードの解析 実行および時間の計測 コマンドのクリア お気に入り Simulink レイアウト 設定 バスの設定 並列 アドオン ヘルプ コミュニティ サポートのリクエスト MATLABの学習

カテゴリでフィルター

MATLAB

- Data Import and Analysis 2
- External Language 1
- Interfaces

適用分野

- Science and Industry 9
- Image Processing and Computer Vision 22
- Signal Processing and Wireless Communications 18
- Robotics and Autonomous Systems 9
- Hardware Interfacing and IoT 131


Simulink

- Simulink Fundamentals 10
- Physical and Event-Based Modeling 1
- Code Generation 33
- Verification, Validation, and Test 2

タイプでフィルター

- ☒ ハードウェアサポートパッケージ 187
- ☐ オプション機能 2
- ☐ 関数 5

### MathWorks ハードウェア サポート パッケージ (187)

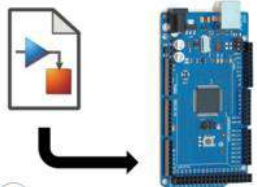


インストール済み

**MATLAB Support Package for Arduino Hardware**

Acquire inputs and send outputs on Arduino boards

ダウンロード: 2362




インストール済み

**Simulink Support Package for Arduino Hardware**

Run models on Arduino boards.

ダウンロード: 1202




インストール済み

**MATLAB Support Package for USB Webcams**

Acquire images and video from UVC compliant webcams.

ダウンロード: 627




インストール済み

**Communications Toolbox Support Package for RTL-SDR Radio**

Acquire RF data using RTL-SDR.


ダウンロード: 438



インストール済み

**MATLAB Support Package for Raspberry Pi Hardware**


Acquire sensor and image data from your Raspberry Pi.



インストール済み

**Embedded Coder Support Package for Texas Instruments C2000...**


Generate code optimized for C2000 MCU.



インストール済み

**Data Acquisition Toolbox Support Package for National Instruments...**

Acquire and analyze data from NI-DAQmx devices.



インストール済み

**Image Acquisition Toolbox Support Package for OS Generic Video Interface**

Acquire video and images from generic video capture devices.

アドオンの入手

アドオンの管理

ツールボックスのパッケージ化

アプリのパッケージ化

**ハードウェア サポート パッケージの入手**

# 企業における活用 モデル化とシミュレーションに基づくモノ作り (MBD)

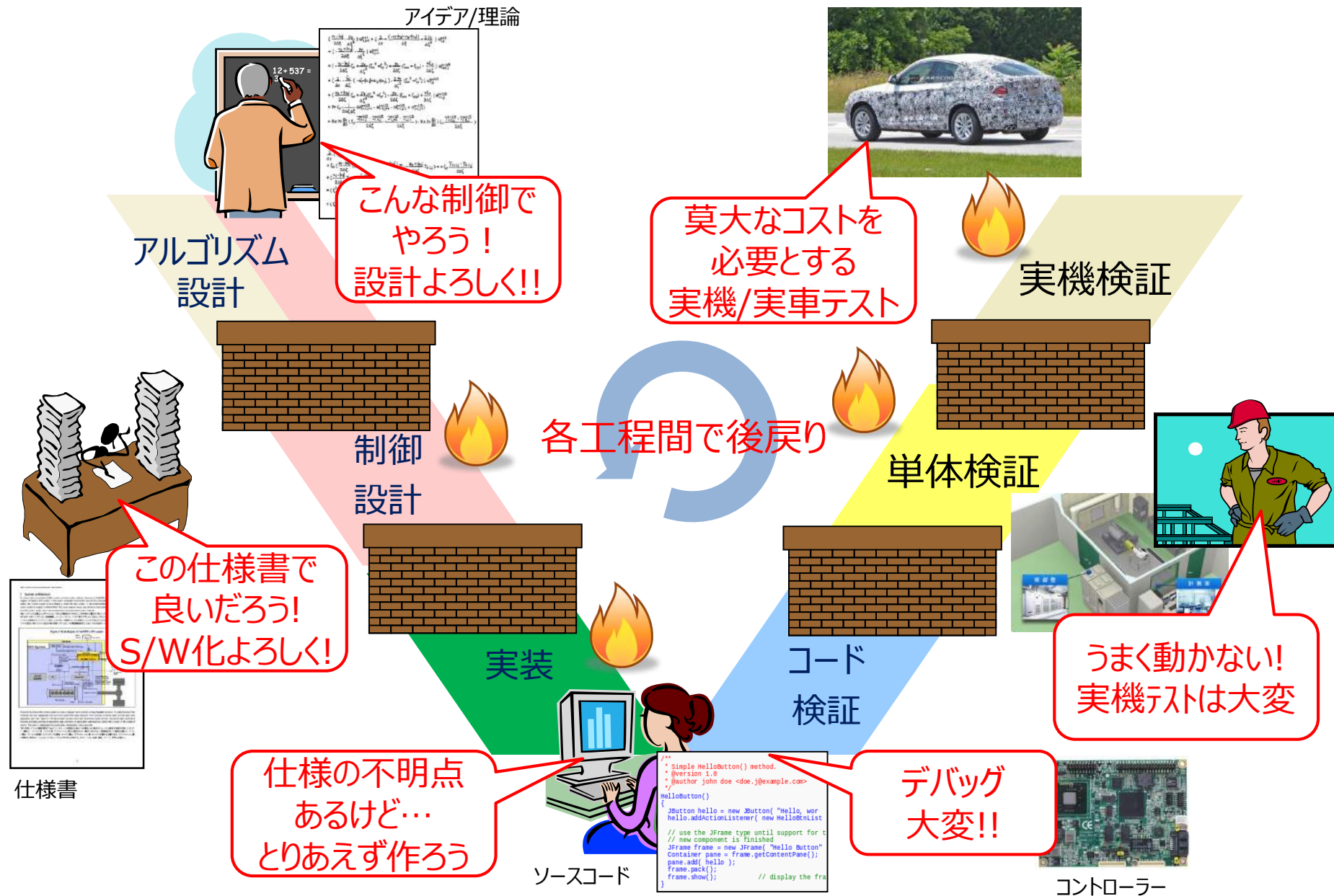
# Simulinkとは？

モデリング・シミュレーションのための環境です



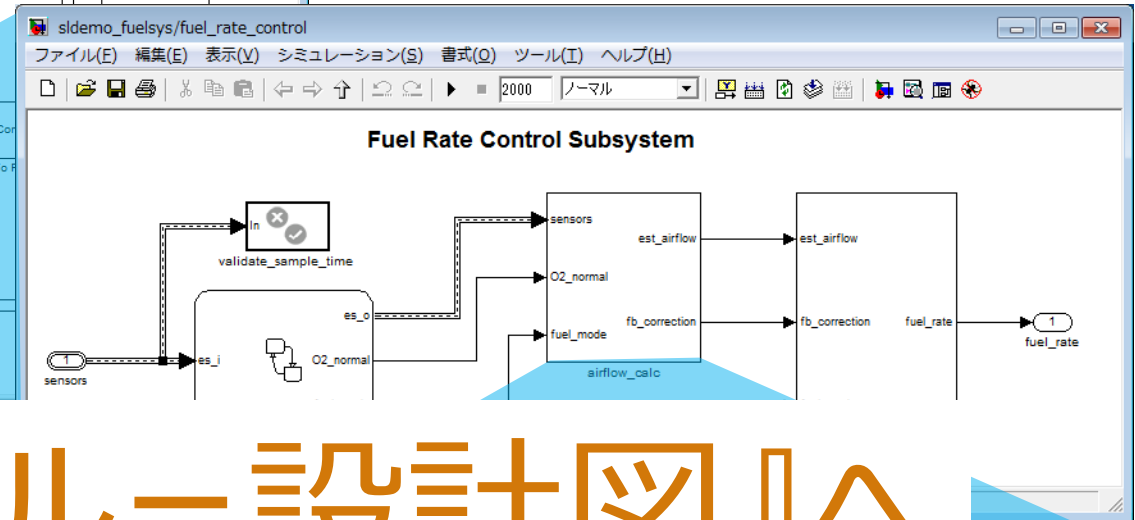
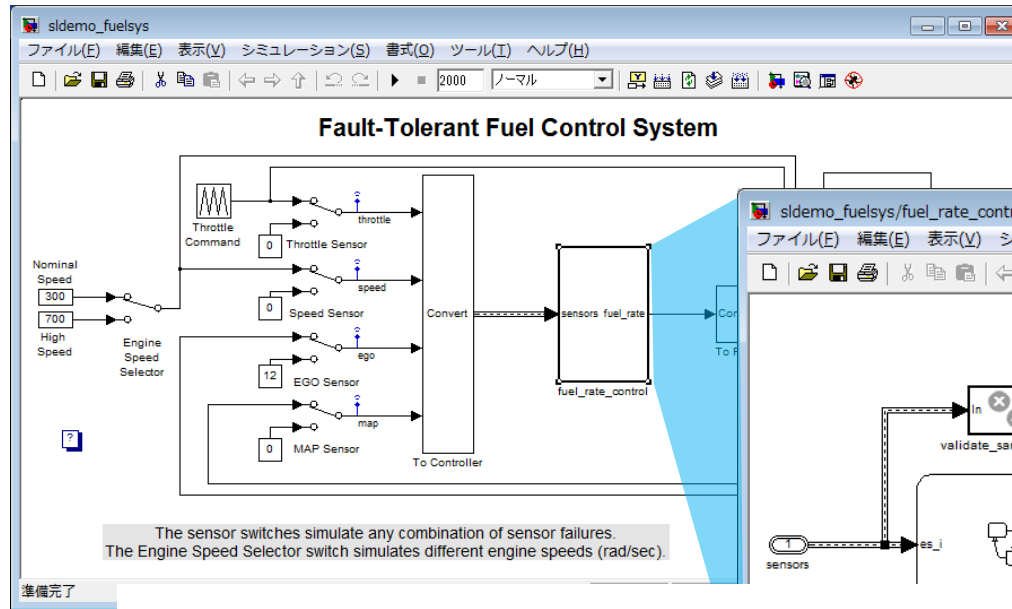


# 従来までのモノ作り... 分断されている、各設計/開発工程

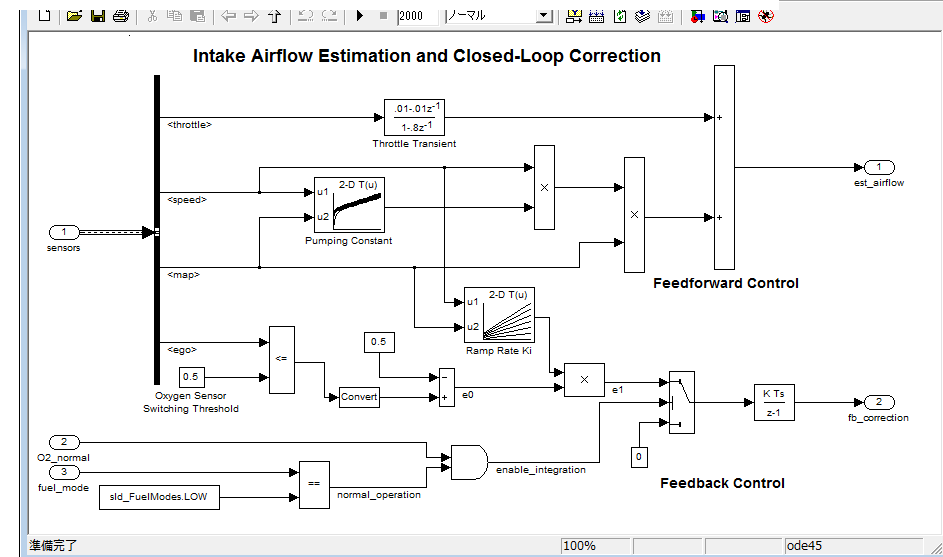
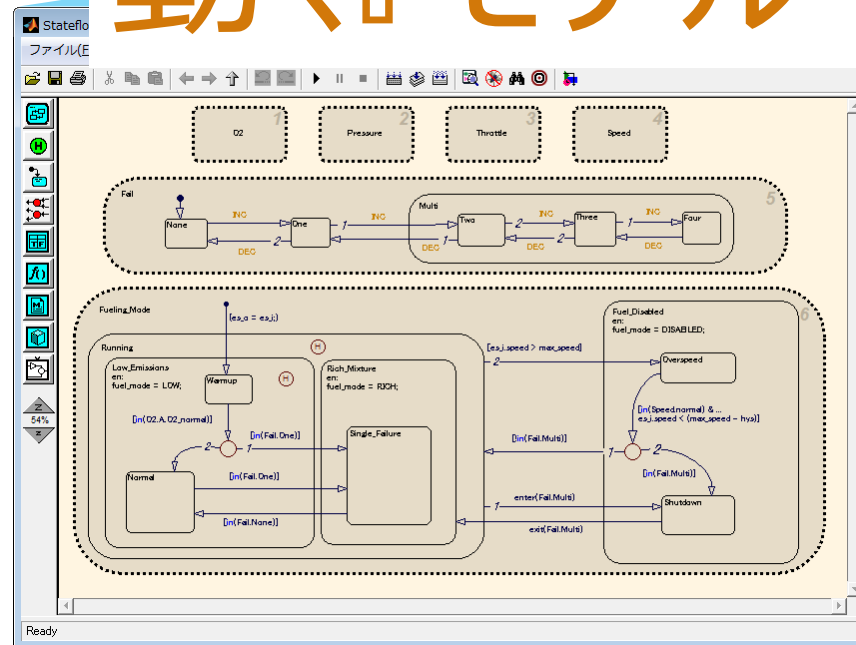




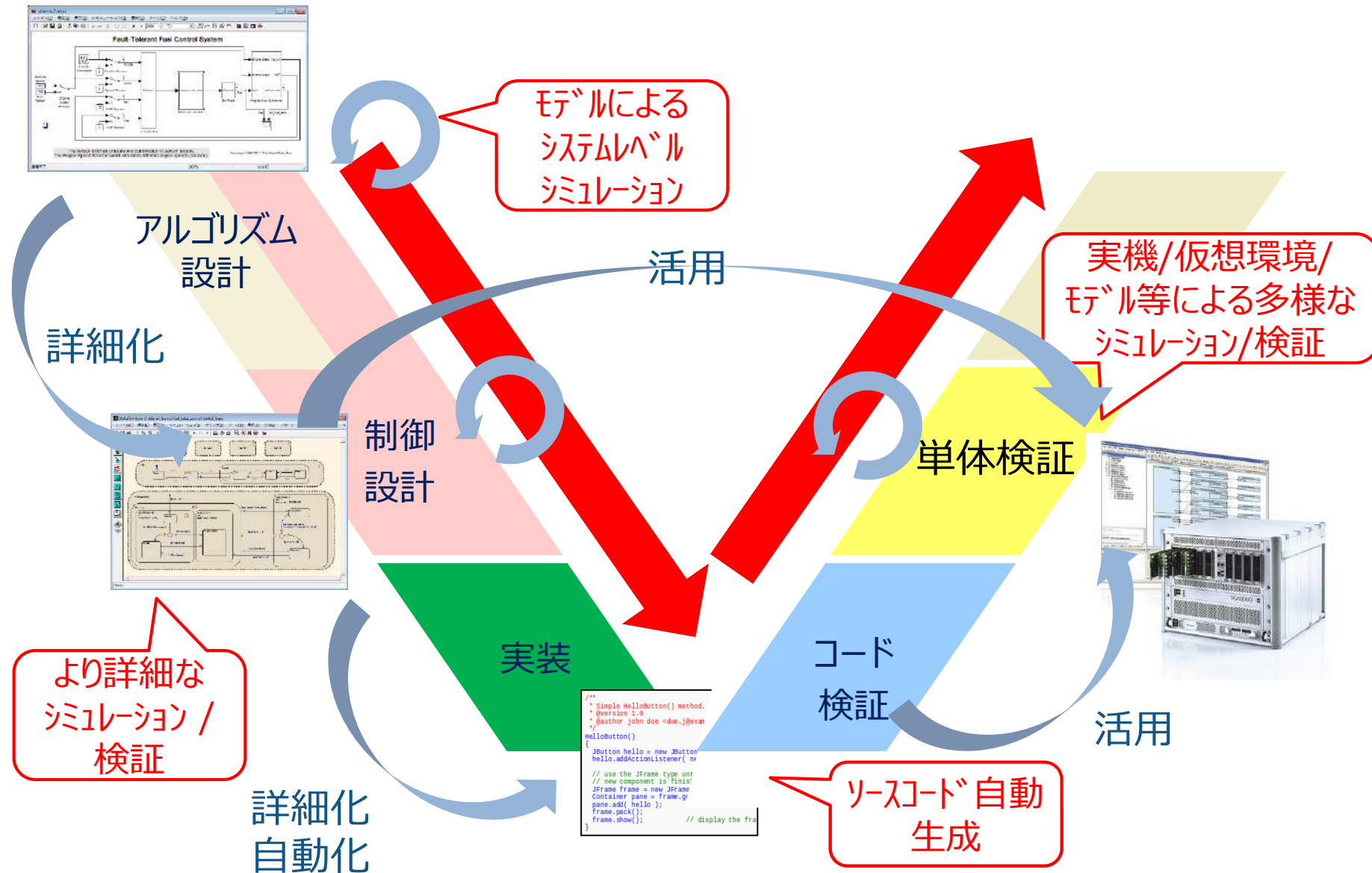
# 動かない『紙』から



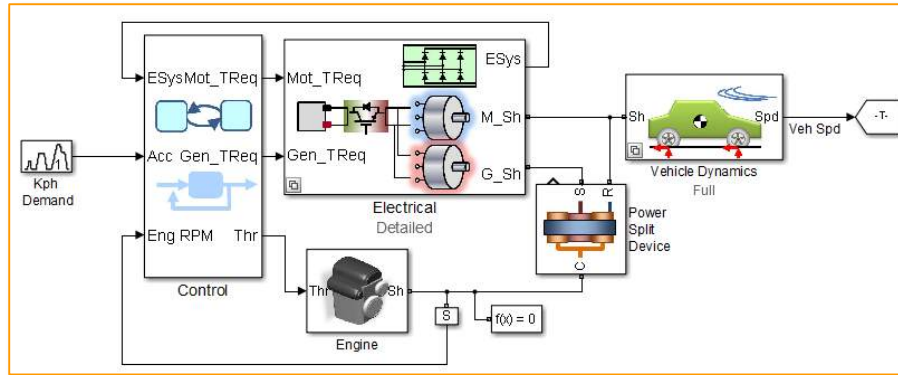
# 動く『モデル＝設計図』へ



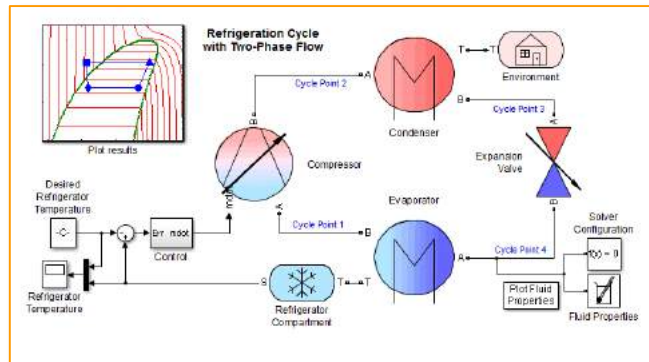
# MBDとは / 実行可能な仕様書によるシミュレーションと自動化



# 自動車関連では...

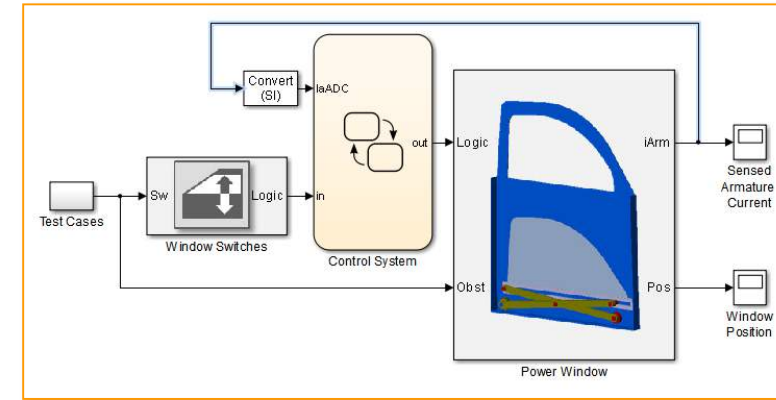
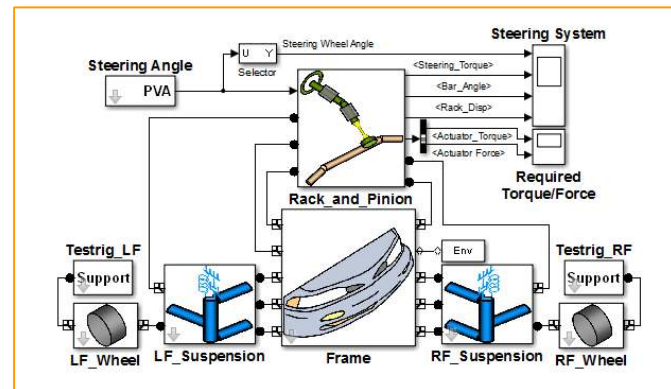


ハイブリッド自動車

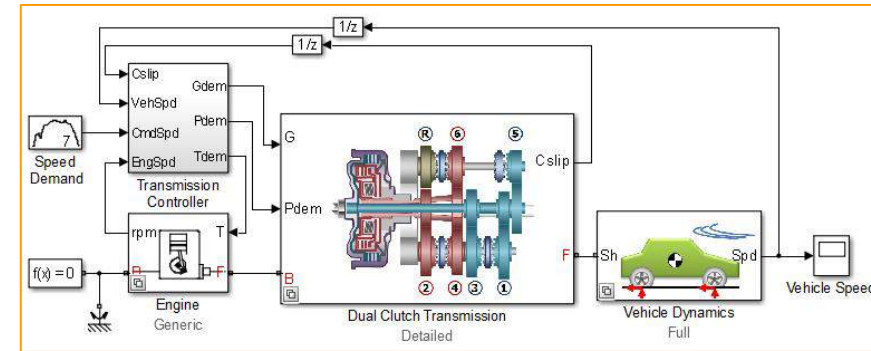


冷却/空調サイクル

パワーステアリング



パワーウィンドウ



デュアル・クラッチ・  
トランスミッション



# Using Model-Based Design to develop high quality and reliable Active Safety & Automated Driving Systems

**Jonny Andersson**

Senior Engineer

Scania

**Thorsten Gerke**

Automotive Industry Manager EMEA

MathWorks



# Scania's AEB - The final Product under Test

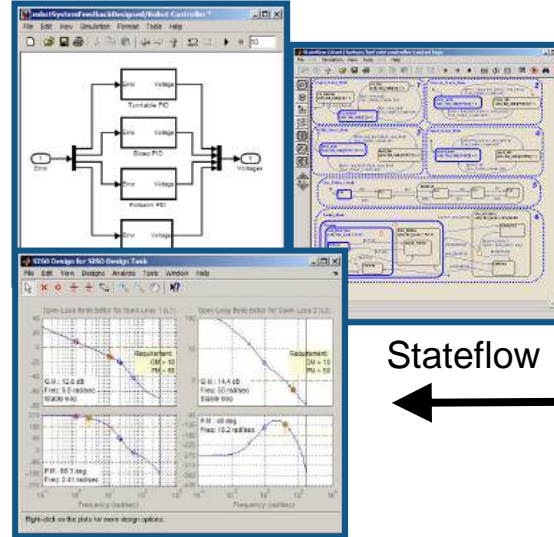


# シミュレーションによる机上検証・解析 (MILS)

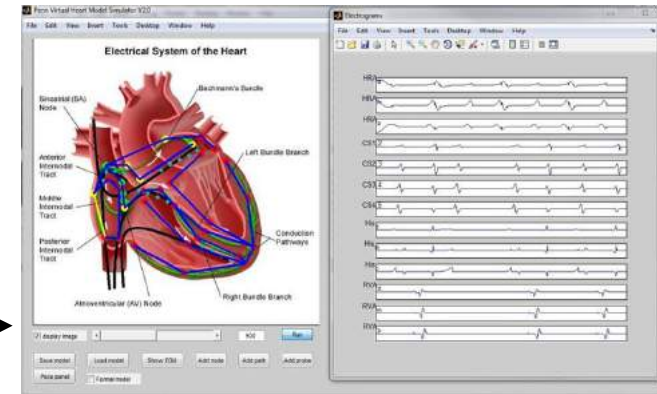
Virtual World

Simulink

設計製品



Stateflow



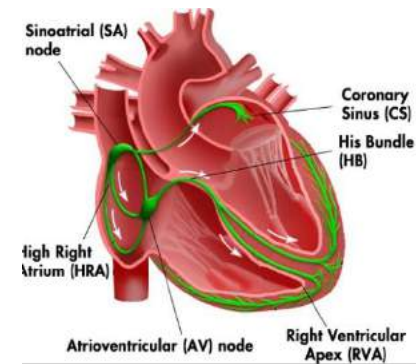
プラントモデル

コントローラモデル

Real World



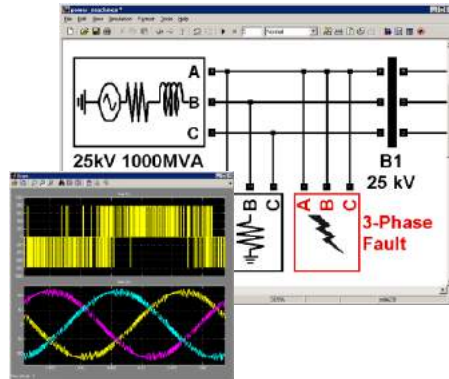
コントローラ



# 物理モデリング製品

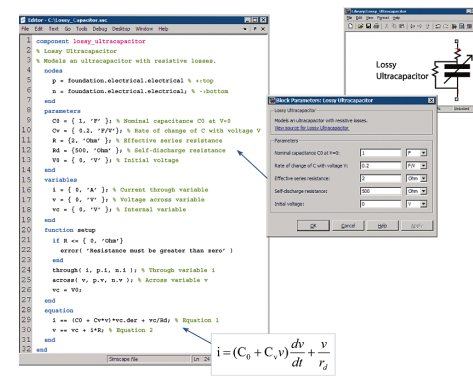
## SimPowerSystems

### ■ 電力・モータ制御系



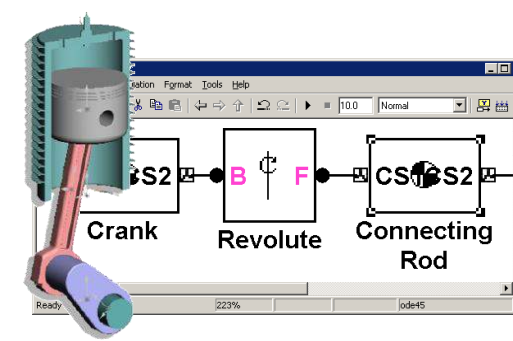
## Simscape

### ■ 物理モデリング基本製品



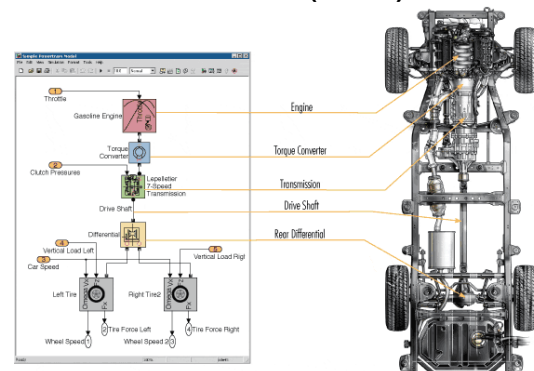
## SimMechanics

### ■ 機構系 (3-D)



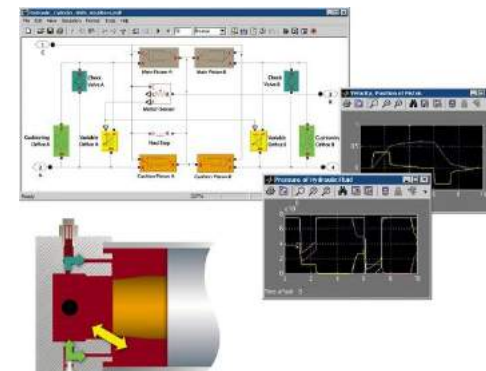
## SimDriveline

### ■ 車両駆動系 (1-D)



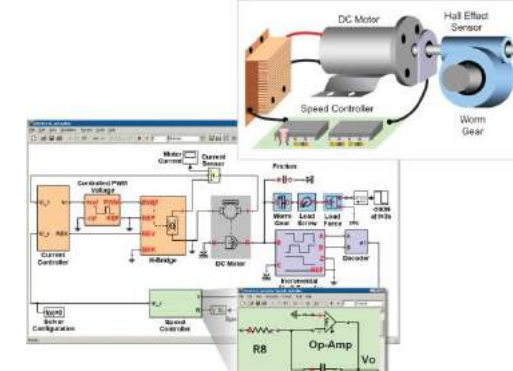
## SimHydraulics

### ■ 油圧系



## SimElectronics

### ■ 電気回路・メカトロ系



数式導出不要、直感的・効率的なモデリング

# Simulinkハンズオントレーニング

# Simulinkの起動方法

**Simulink**の起動方法（2通り）.

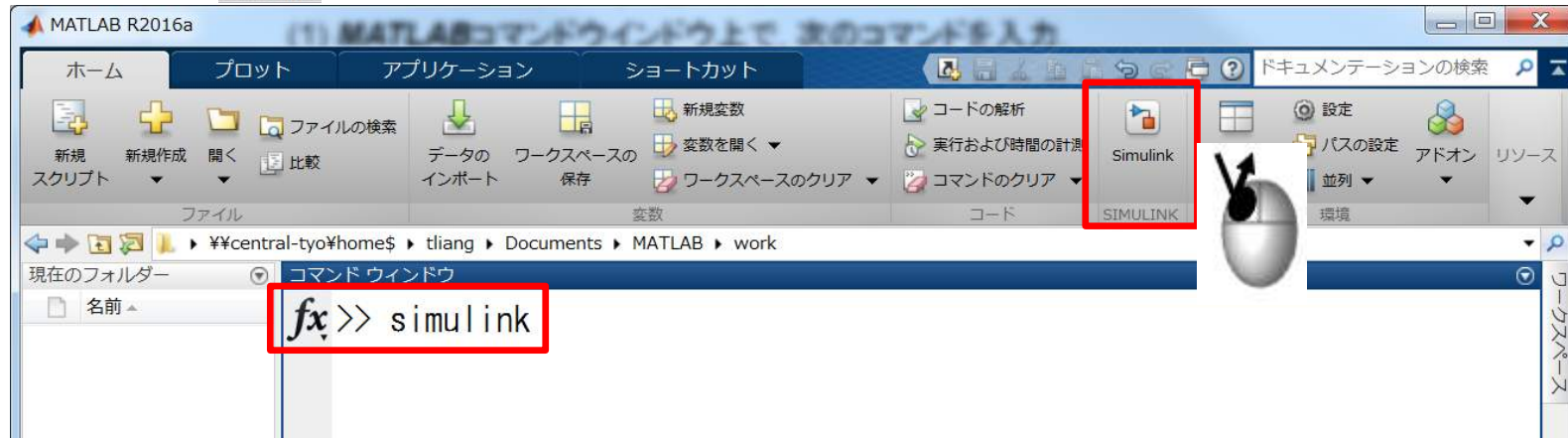
(1) **MATLAB**コマンドウィンドウ上で、次のコマンドを入力. **または**

**>> simulink**

(2) ツールバーの,



をクリック.



**Simulink**を起動すると, **Simulink**スタートページ が起動する.



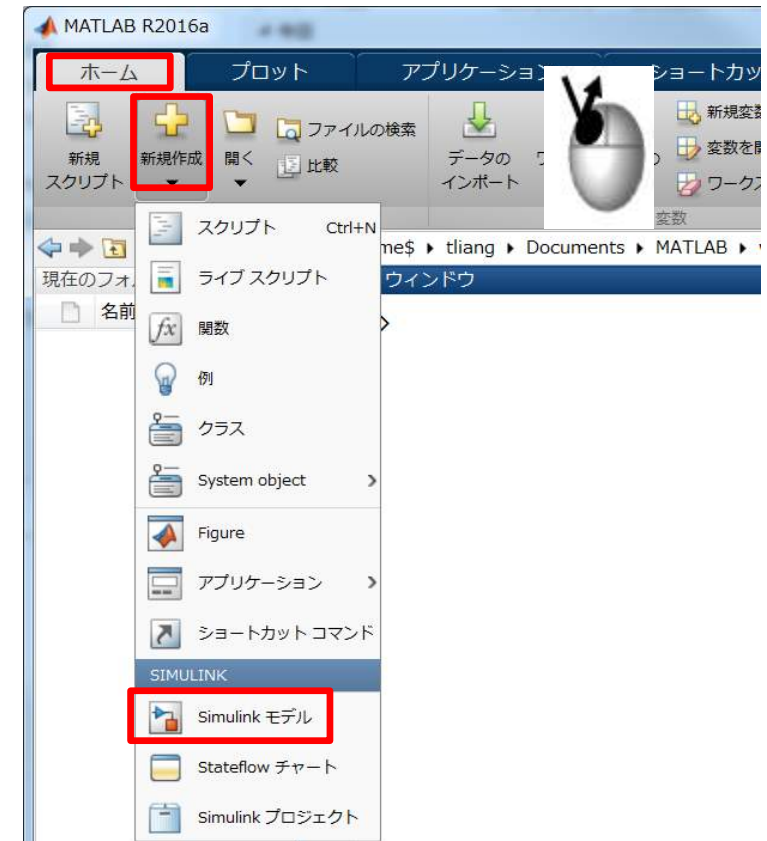
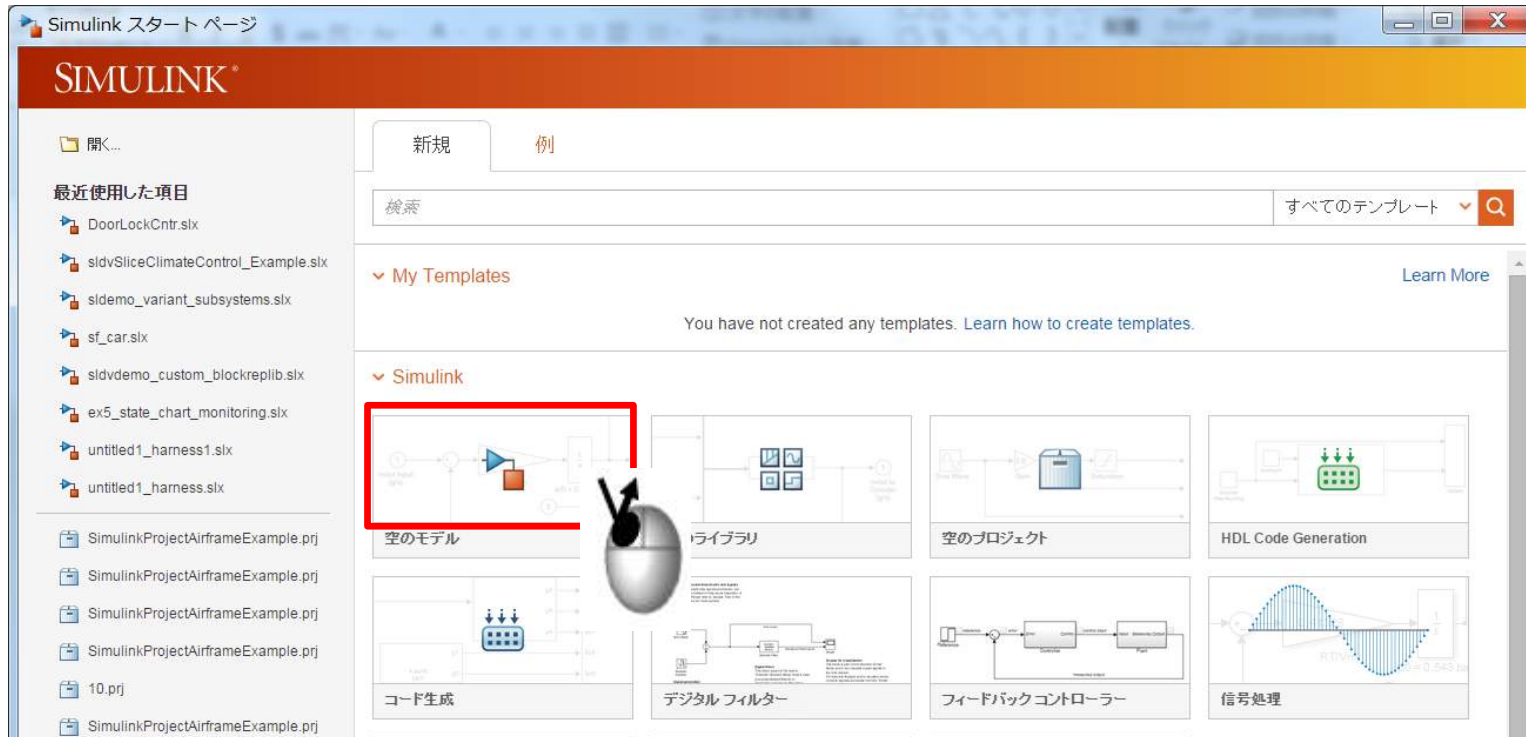
# モデルウィンドウの起動

新規のモデルウィンドウを起動する方法は、2通りある。

- (1) **Simulinkライブラリブラウザー**から、空のモデル
- (2) **MATLAB**のメニューバーより、  
[ホーム]→[新規作成]→[Simulinkモデル]を選択。



をクリック。



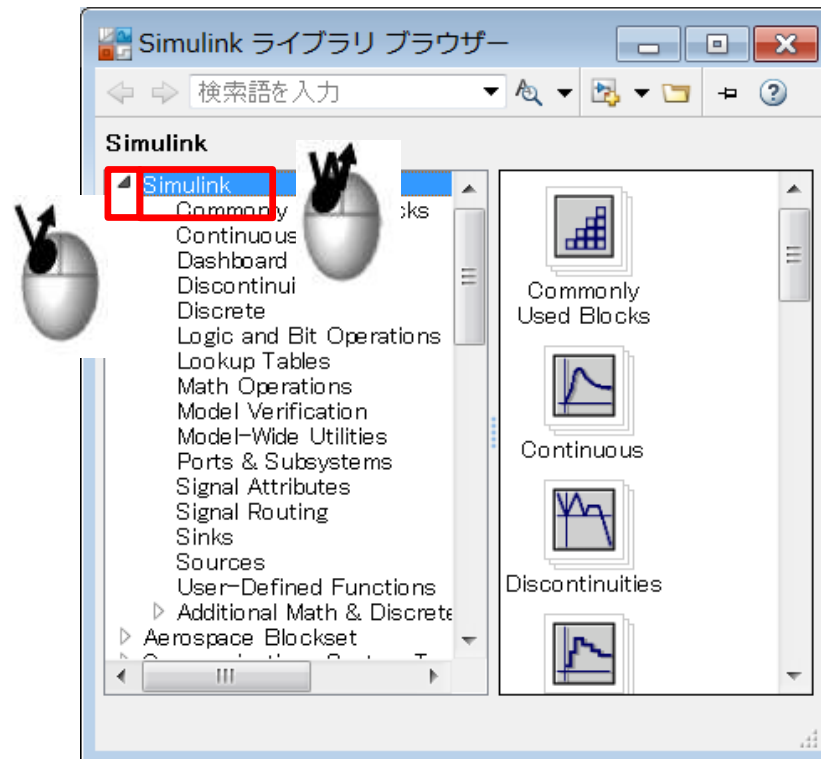
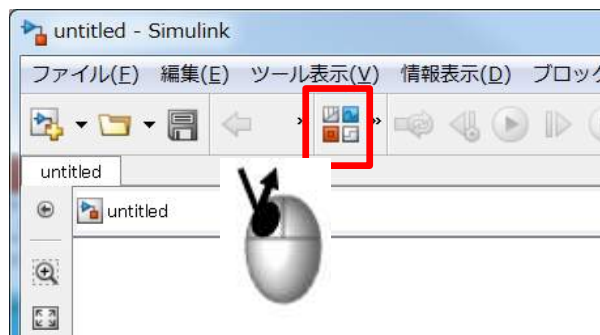
# Simulinkのブロックライブラリ

 をクリックして、**Simulinkライブラリ ブラウザー** (※) を起動し、次のいずれかの操作をする。

- (1) [Simulink]をダブルクリック。
- (2) [Simulink]の左側の[▶]マークをクリック。

その後、**Simulink**のブロックライブラリがツリー構造で現れる。

(※) 積分器などの各種のブロック要素を備えたライブラリ



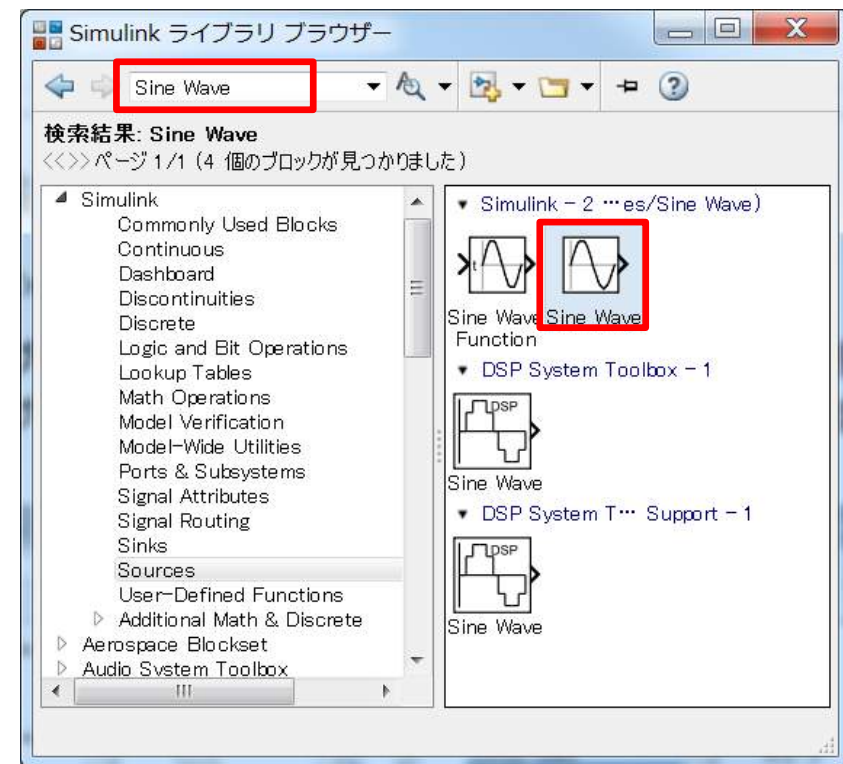
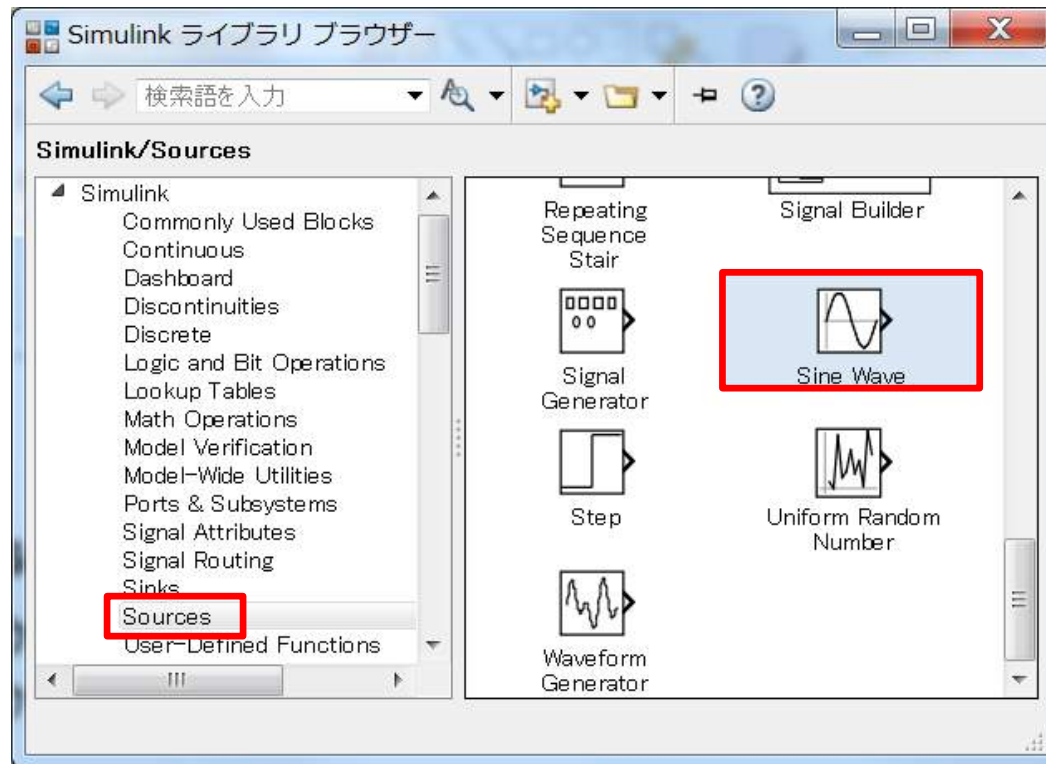
Commonly Used Blocks	利用頻度の高いブロック群
Continuous	連続時間で処理を実行
Discontinuities	不連続要素
Discrete	離散時間で処理を実行
Logic and Bit Operations	論理演算、ビット演算
Lookup Tables	ルックアップテーブル
Math Operations	数学演算、代数演算、複素数の扱い
Model Verification	信号の上下限等の検証
Model-Wide Utilities	モデル情報記述、線形化
Ports & Subsystems	入出力ポート、サブシステム、 Model Reference
Signal Attributes	データタイプ変換、レート変換、 初期状態、信号の調査
Signal Routing	信号の抽出・並び替え、切替
Sinks	信号の観測、ロギング
Sources	信号源
User-Defined Functions	ユーザ定義関数
Additional Math & Discrete	その他数学演算、離散系ブロック

# ブロックの選択, 検索

(例) モデルウィンドウ上に, [Sources]→[Sine Wave]を置く.

(1) *Simulink*ライブラリブラウザーで, [Simulink]→[Sources]より探す.

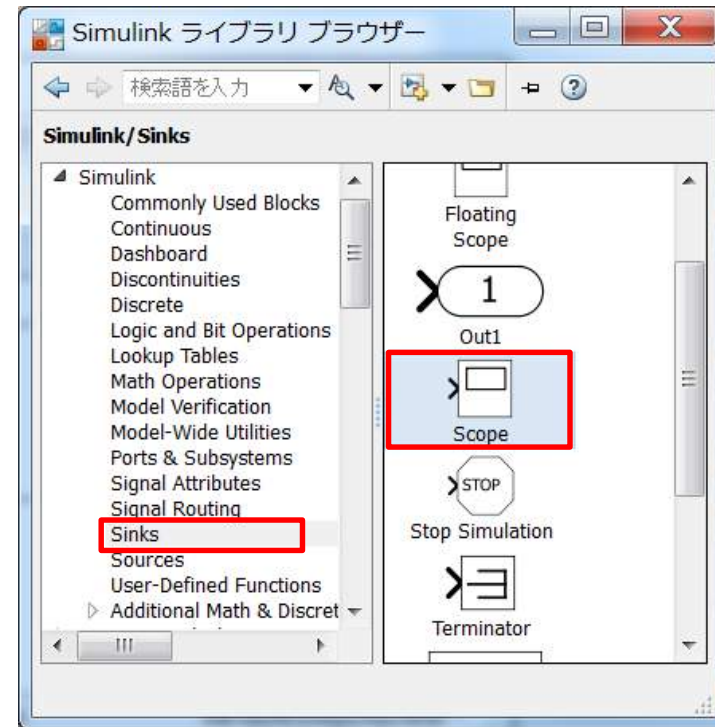
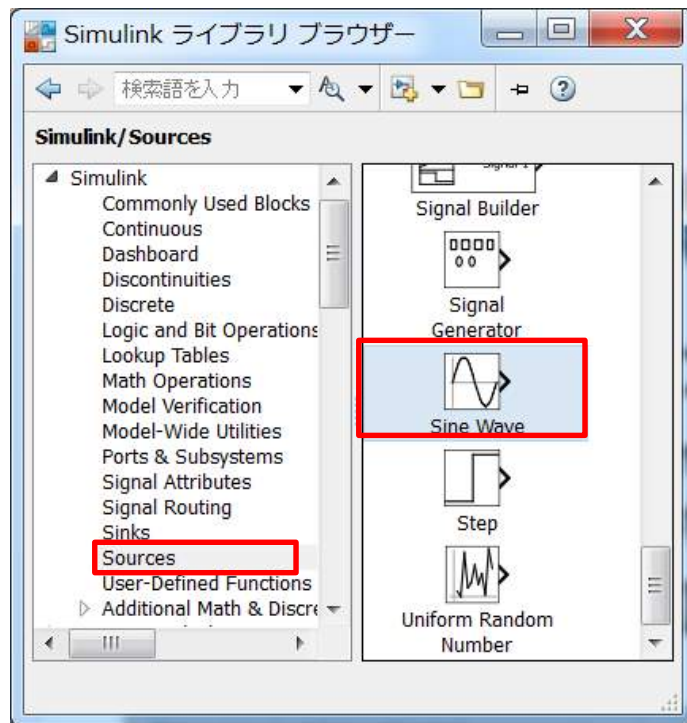
(2) *Simulink*ライブラリブラウザーの  で, ブロックの検索を行う.



## 演習1 : sin信号をスコープブロックで表示

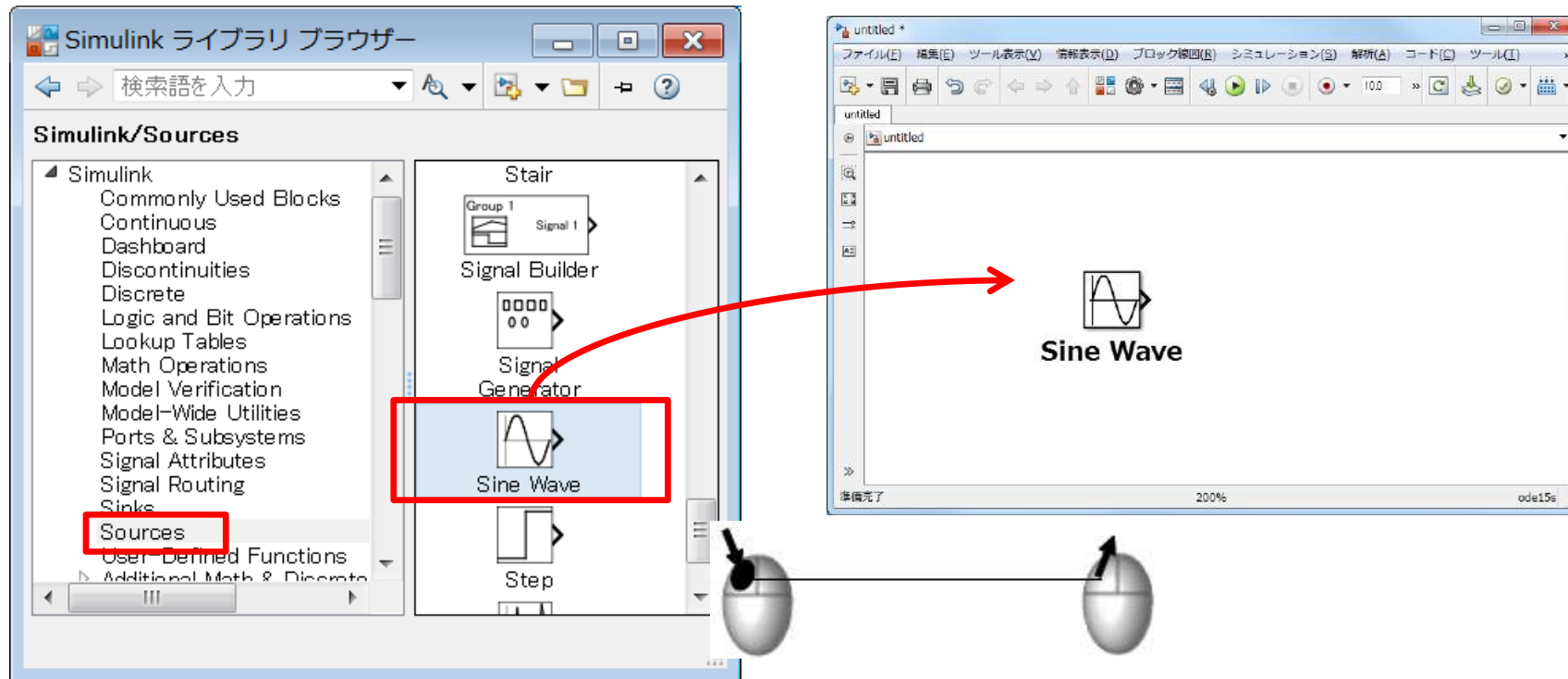
*Simulink*ライブラリブラウザー から

1. **[Simulink]→[Sources]→[Sine Wave]**選びモデルウィンドウ上まで、マウスの左ボタンを使い、ドラッグ & ドロップ。
2. **[Simulink]→[Sink]→[Scope]**を選び同様にドラッグ & ドロップ



# ブロックの配置

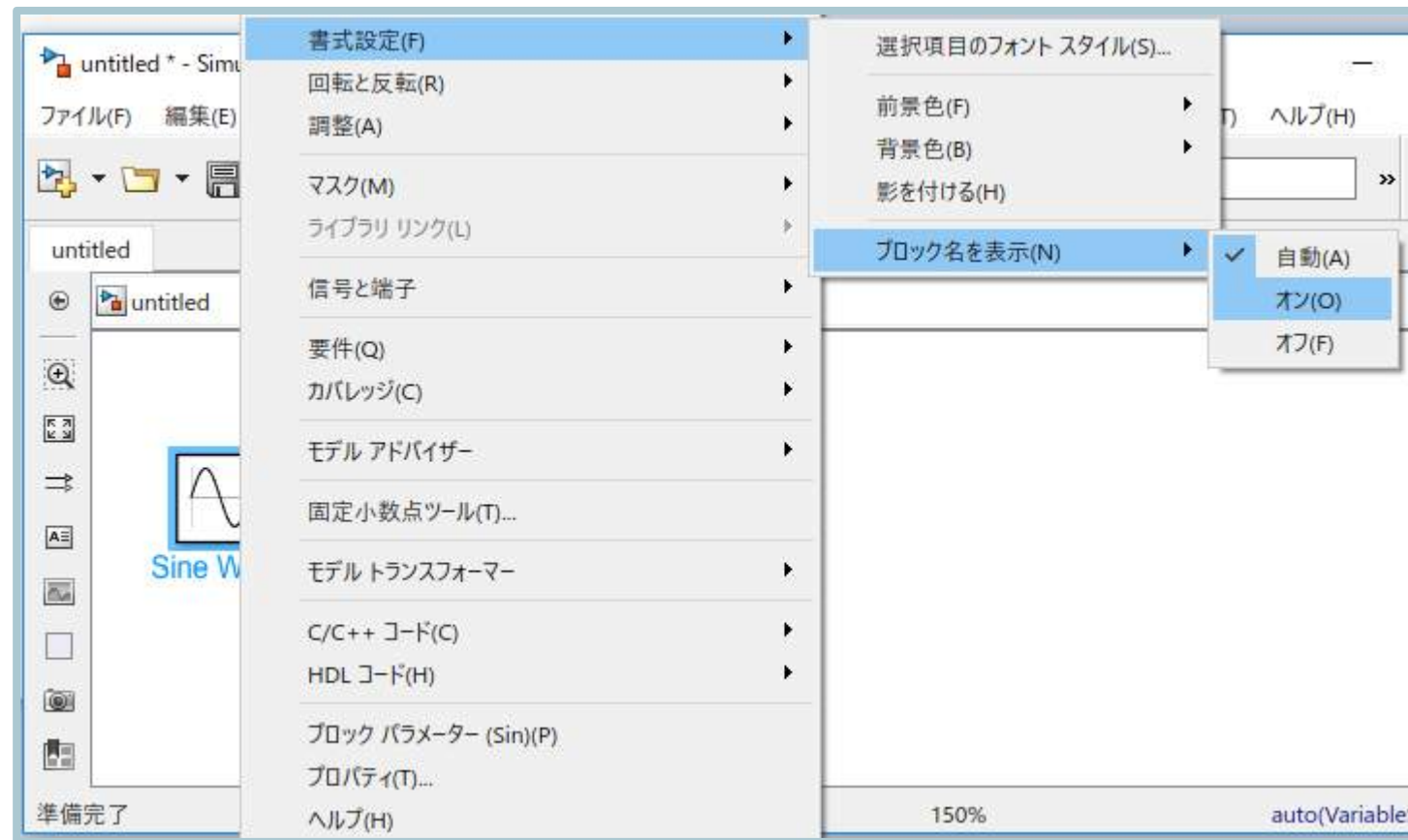
[Sine Wave]を選択後、モデルウィンドウ上まで、マウスの左ボタンを使い、ドラッグ＆ドロップ。





## 補足：ブロック名の表示

ブロックを選択してマウスを右クリックして、書式設定を選択し、  
ブロック名を表示を選択後、オンを選択する

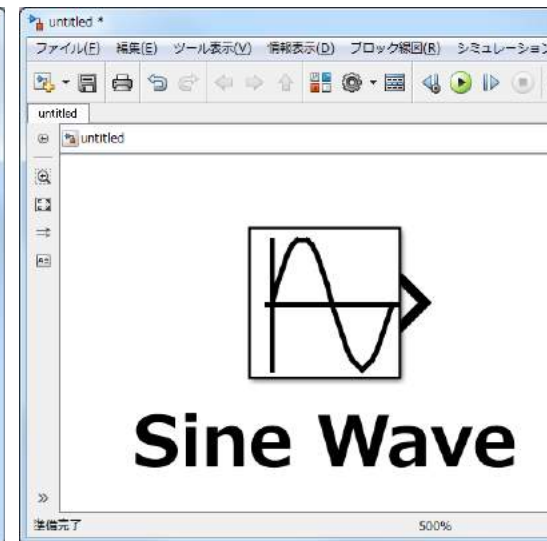
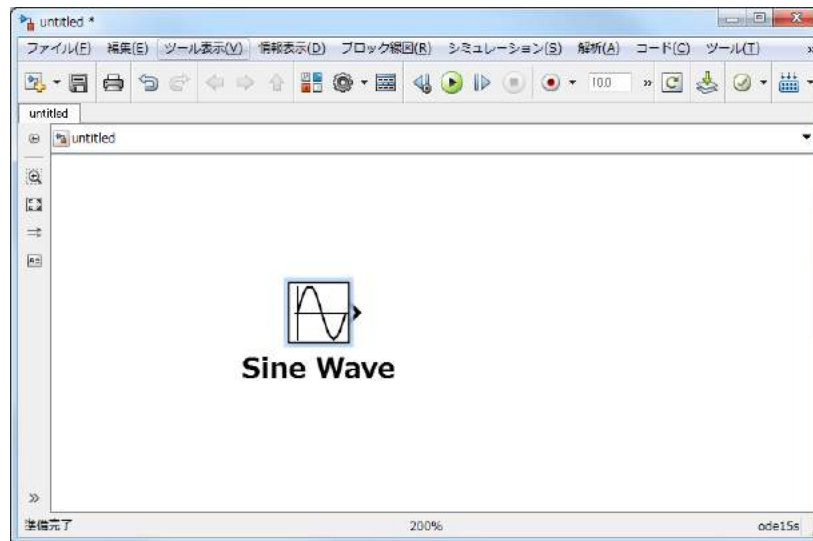


# ブロックのリサイズ

ブロックの大きさを変えるには、ブロック選択後、四隅のうちいずれかにマウスカーソルを近づけ、左マウスボタンでドラッグする。

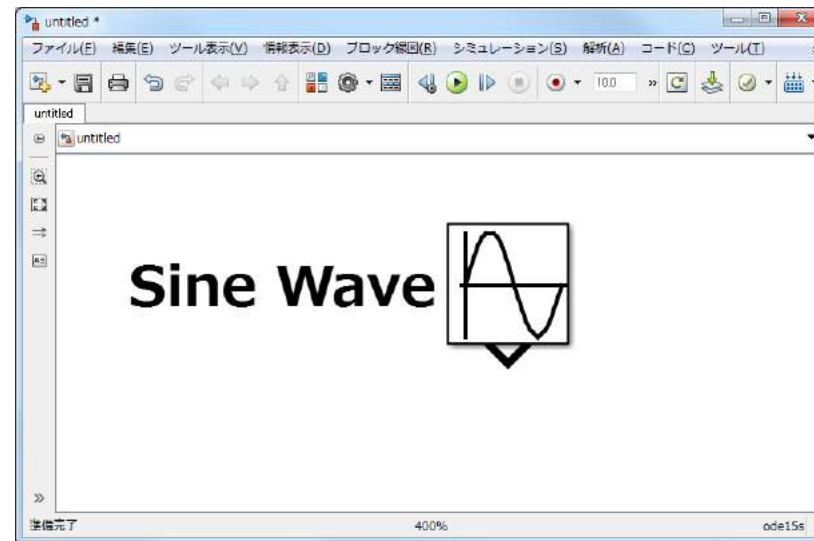
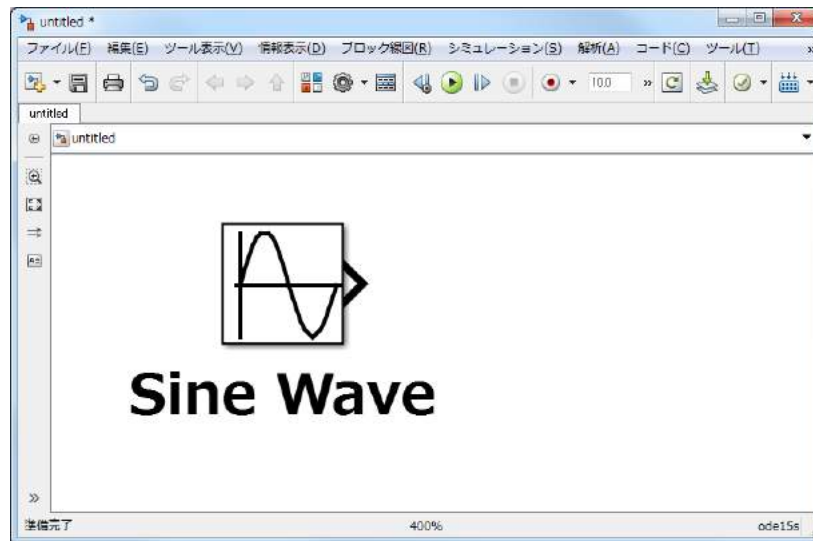
または、

ブロック選択後、マウスのホイール回転で zoom in ・ zoom out.



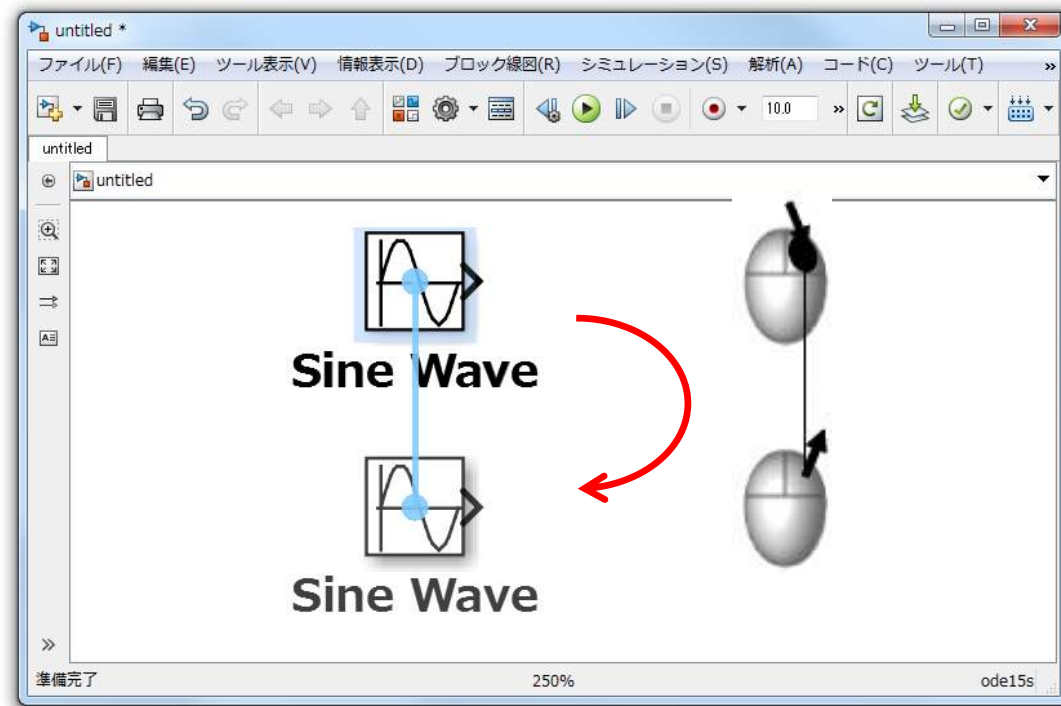
# ブロックの回転

フィードバック部に配置するブロックを結線する場合など、ブロックを回転させるには、ブロックを選択後、キーボードの **[Ctrl+R]** キーで時計回りに90度回転。



# ブロックのコピー

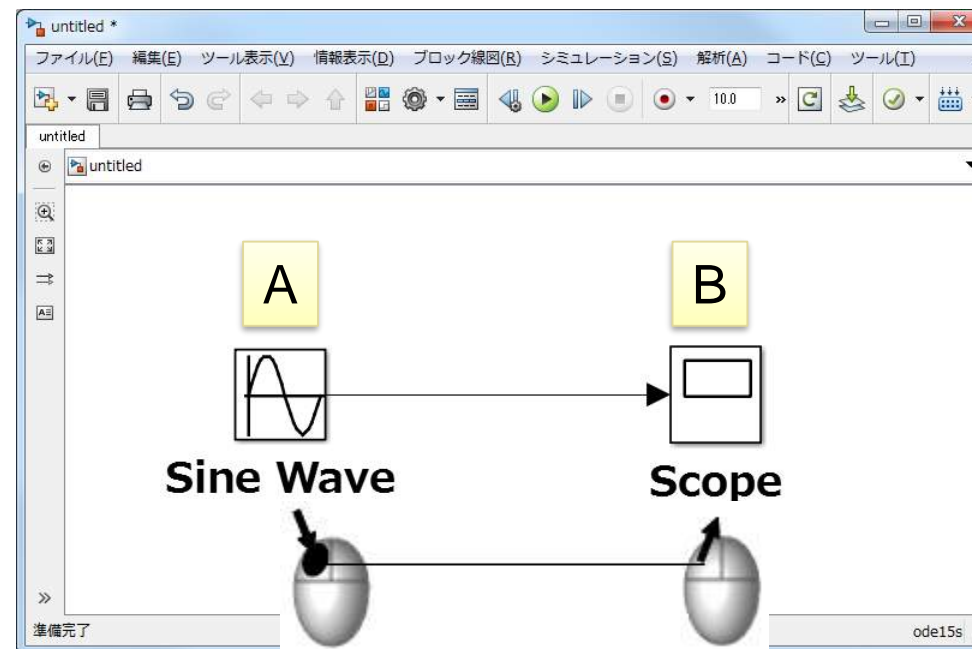
同じブロックをモデル内にコピーするには、右マウスボタンを使ってドラッグする。



## ブロックの結線

同様に, **[Sinks]**→**[Scope]**をコピー後, 結線をする.

- (1) 結線をしたいブロックの片方のポートを選択し, もう片方のポートまで, 左マウスボタンでドラッグする.
- (2) AからBへ結線したい場合は, [Ctrl]キーを押しながら, マウスでA, Bブロックを順に左クリックすると, 簡単に結線できる.





# 様々なブロックの結線操作

結線	マウス操作	図の番号
信号線を直角に曲げる	左マウスボタンでドラッグ	図1
信号線を途中から引き出す	右マウスボタンでドラッグ	図2

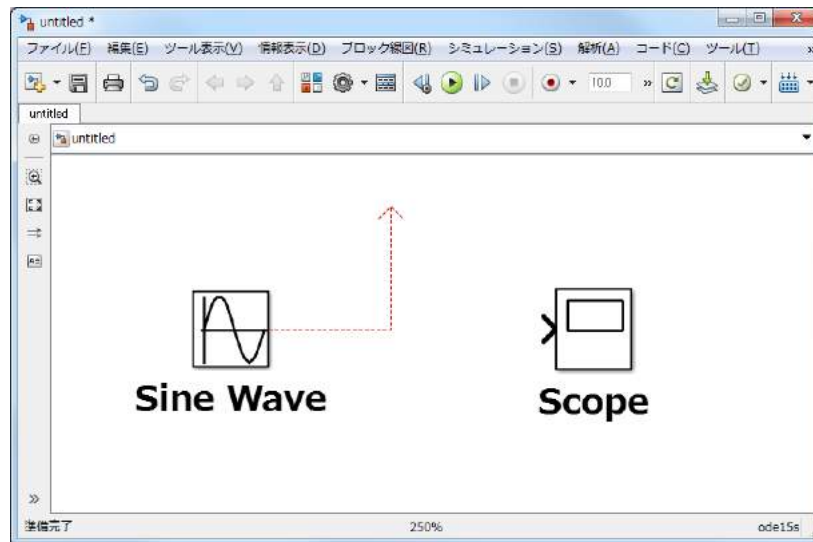


図1

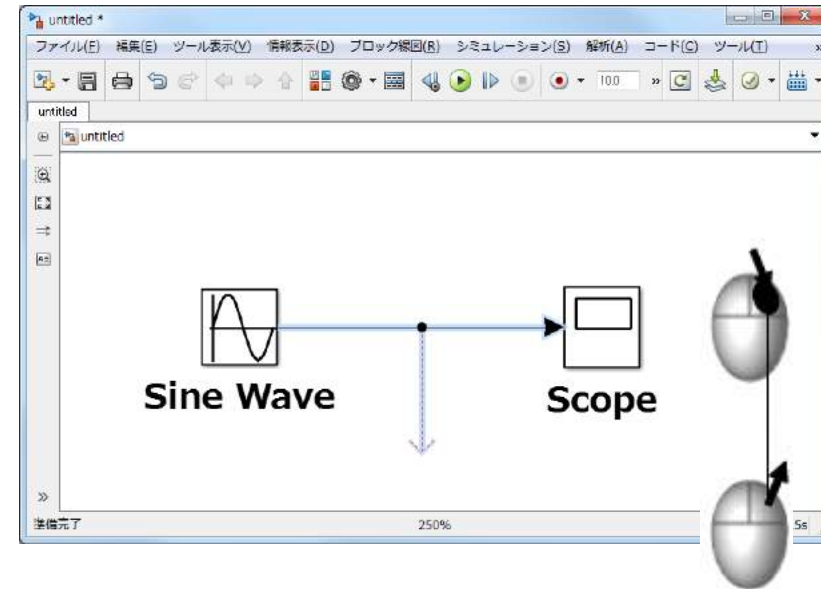
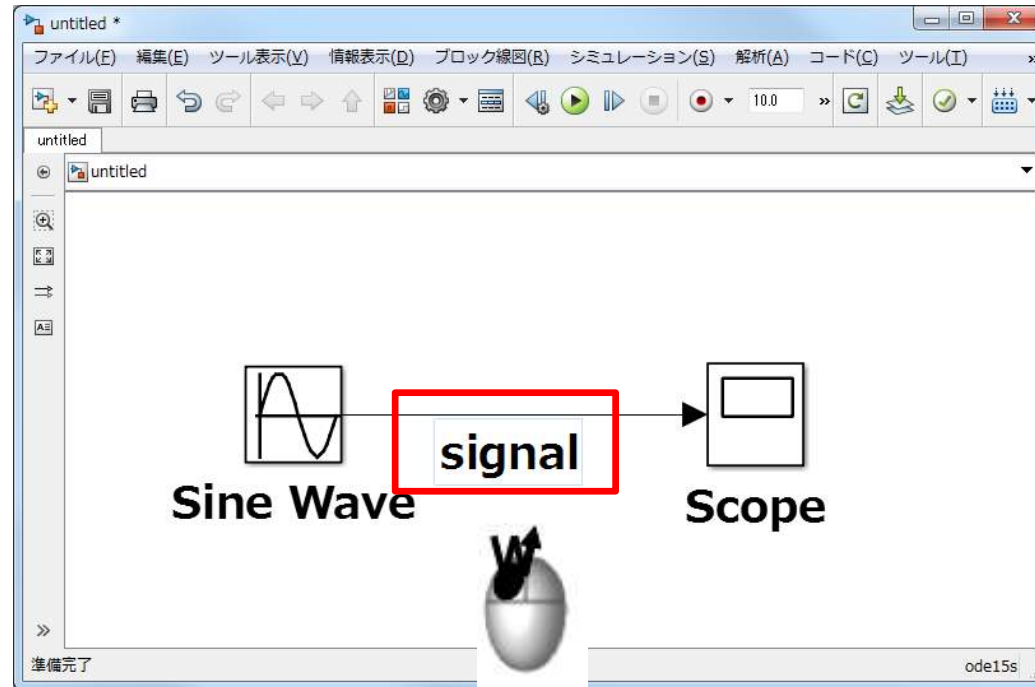


図2

## 信号線にラベルを付ける

信号線にラベルを付けるには、信号線上でダブルクリックし、文字を入力する。  
ラベルの位置は、マウスでドラッグすることにより、変更できる。



## ブロックパラメータの設定

パラメータを変更したいブロックをダブルクリックする。  
(例) **Sine Wave**ブロックのパラメータの中で、  
次のように正弦波の振幅, 周波数を変更する。



ブロック パラメーター: Sine Wave

パラメーター

正弦波タイプ: 時間ベース

時間 (t): シミュレーション時間を使用

振幅:

1

バイアス:

0

周波数 (rad/sec):

2\*pi\*1

位相 (rad):

0

サンプル時間:

0

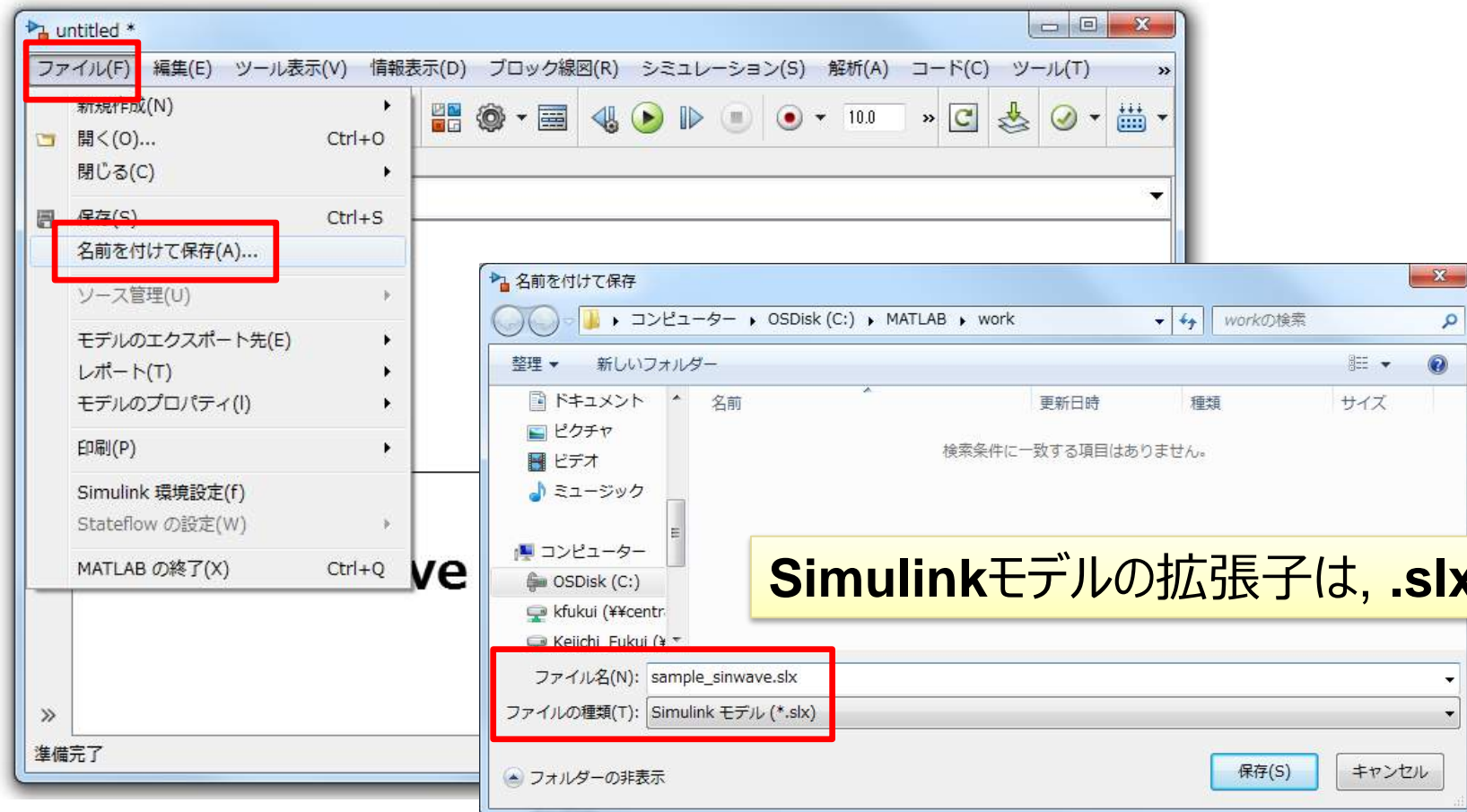
☒ ベクトル パラメーターを 1 次元として解釈

OK(O) キャンセル(C) ヘルプ(H) 適用(A)


周波数は角周波数[rad/sec]で入力  
 $f$  [Hz] を指定する場合  $\omega = 2\pi f$  を入力

# モデルの保存と起動

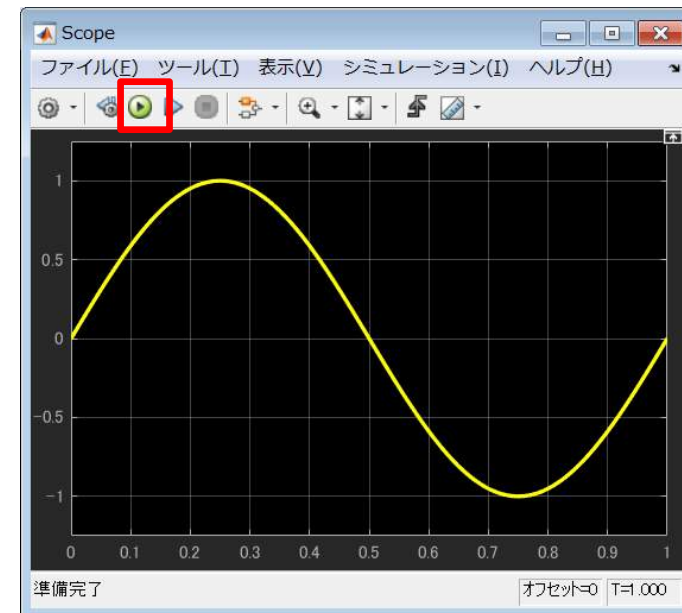
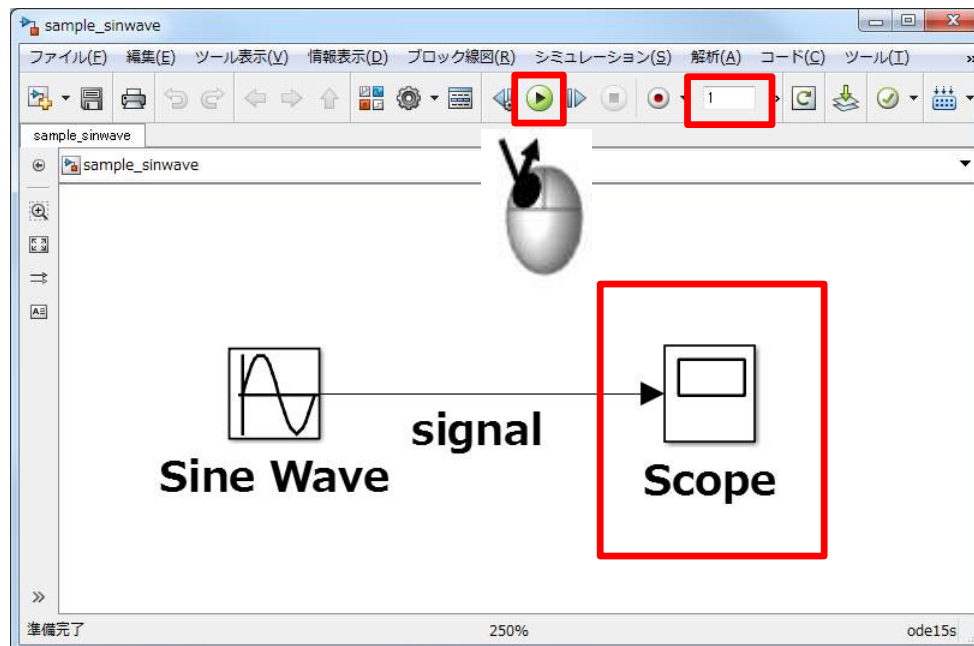
作成したモデルを保存するには、メニューの**[ファイル]**→**[名前を付けて保存]**を選択し、適当なファイル名を付けて保存する。



# シミュレーション

シミュレーション終了時間  を設定し、 クリックして、シミュレーションを実行する。 **Scope**ブロックをダブルクリックして、ウィンドウを開いておくと、シミュレーションの経過を見ることができる。

## sample\_sinwave.slx

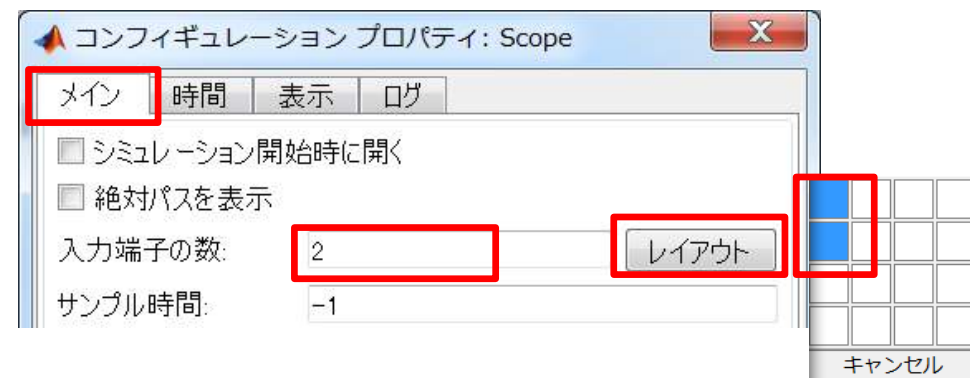
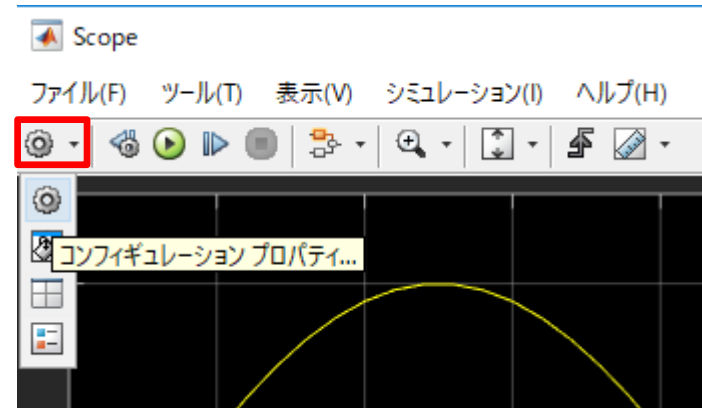
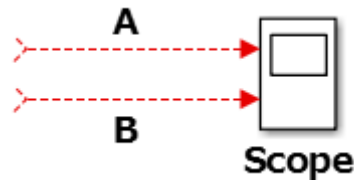




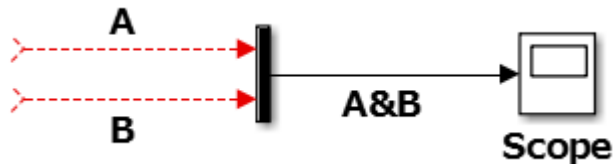
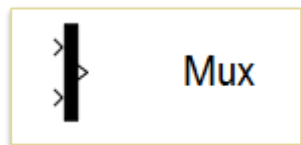
# Scopeブロックによる複数信号の観測

Scopeウィンドウに2つの応答を一度に表示するに次の2通りの方法がある.

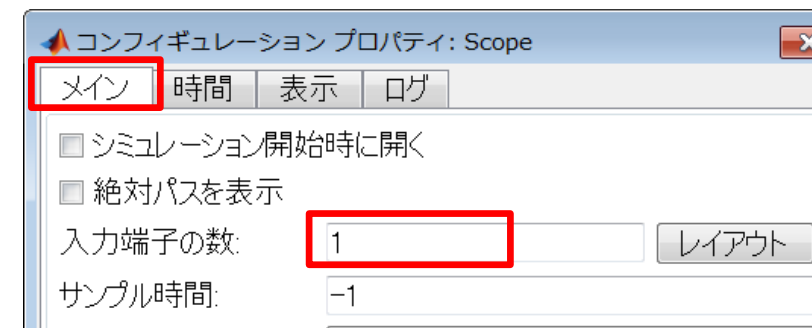
(1) 2つの座標軸に, それぞれの応答を表示する.



(2) 1つの座標軸に, 2つの応答を表示する.



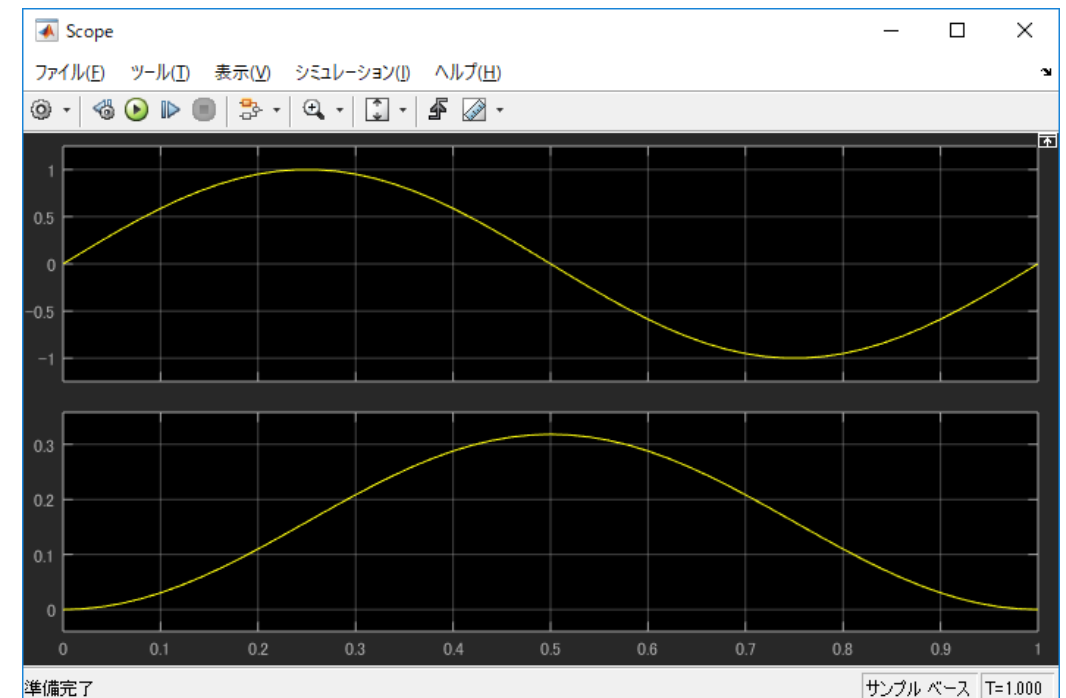
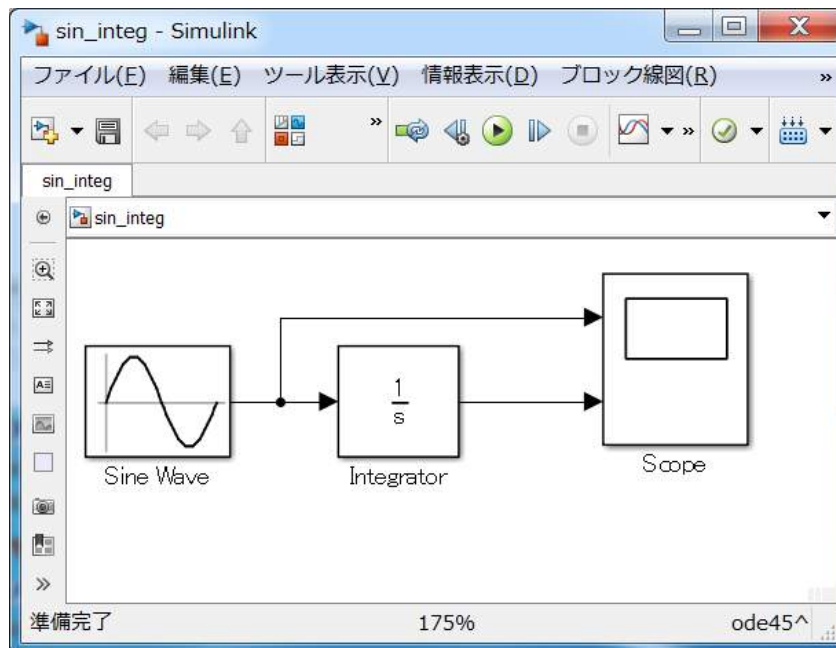
Signal Routingライブラリにある,  
**Mux**ブロックを使用する.



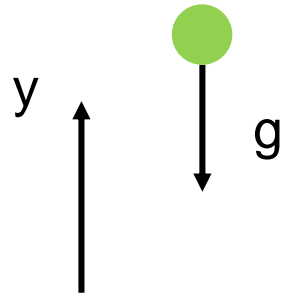
## 演習2 : sin関数の積分シミュレーション

**Simulink**ライブラリブラウザー から

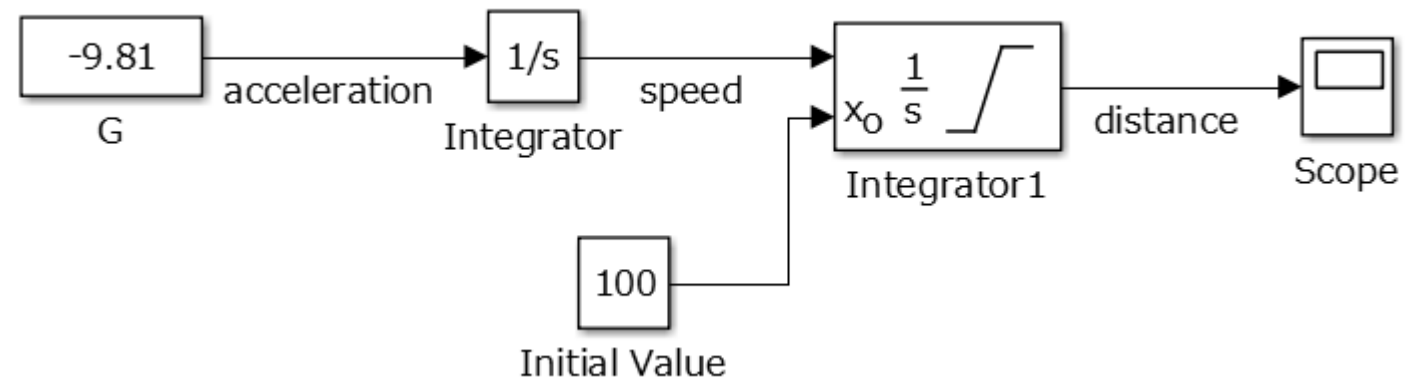
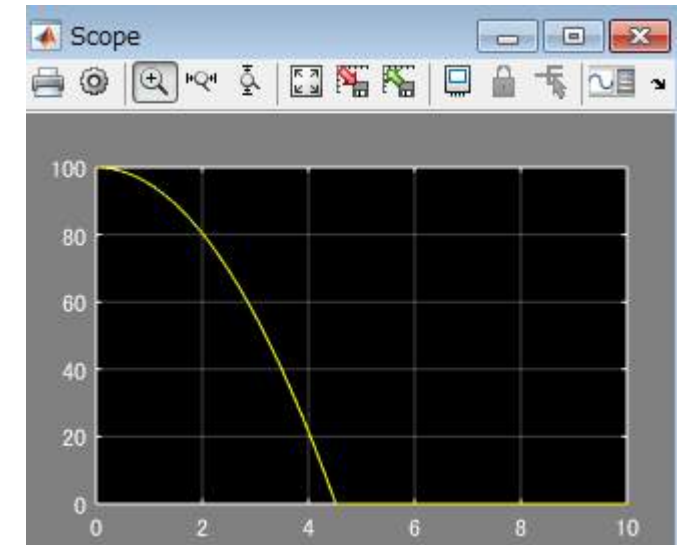
1. **[Simulink]→[Continuous]→[Integrator]**選び,配置
2. **Scope**をダブルクリック、パラメータ設定で入力端子の数を2と指定し、下図のように結線してシミュレーション



## 演習3: 自由落下のモデリングとシミュレーション



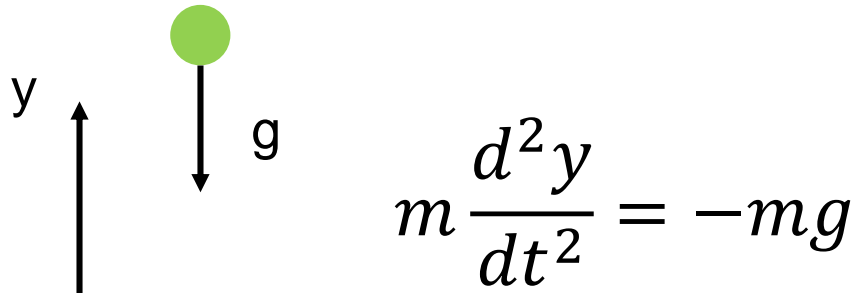
$$m \frac{d^2 y}{dt^2} = -mg$$



# *Simulink*によるシミュレーション (時間応答解析)

- (1) 自由落下の***Simulink***モデルの作成.
- (2) コンフィギュレーションパラメータの設定.
- (3) シミュレーション実行, 出力信号の観測.

# 自由落下のモデリング手順 (1)



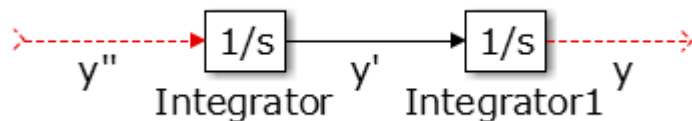
物体の質量	$m$ [kg]
重力加速度	$g$ [m/s <sup>2</sup> ]
変位(出力)	$y$ [m]

微分方程式からブロック線図に置き換える方法

- (1) 最高次の微係数について整理する.

$$\ddot{y} = -g$$

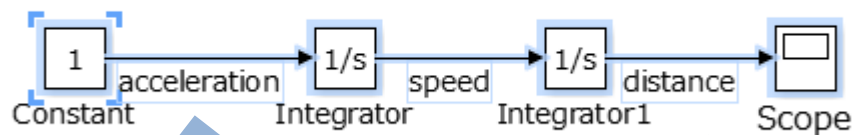
- (2) 積分器を中心にモデリングする.



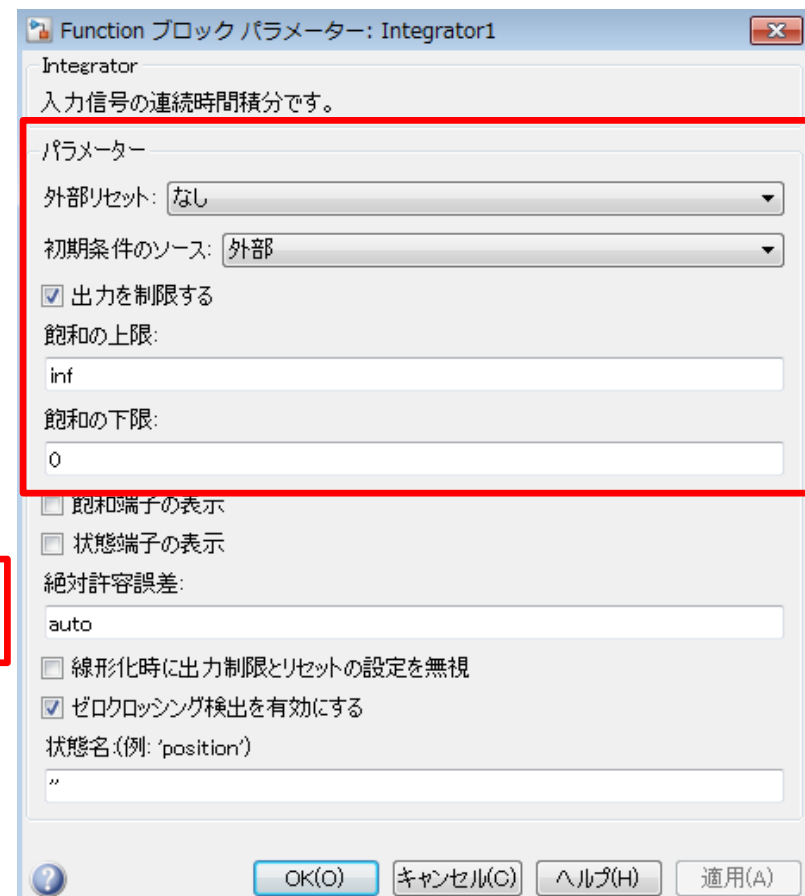
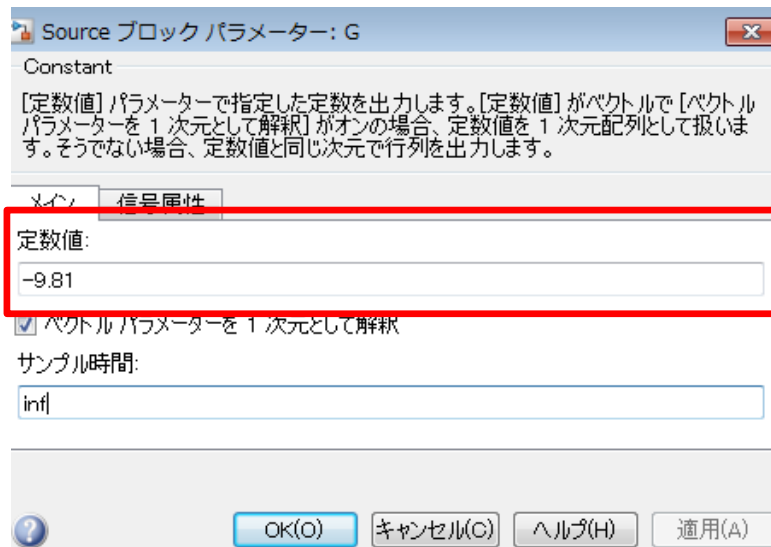


## 自由落下のモデリング手順 (2)

(3) 下図のように微分方程式を結線する

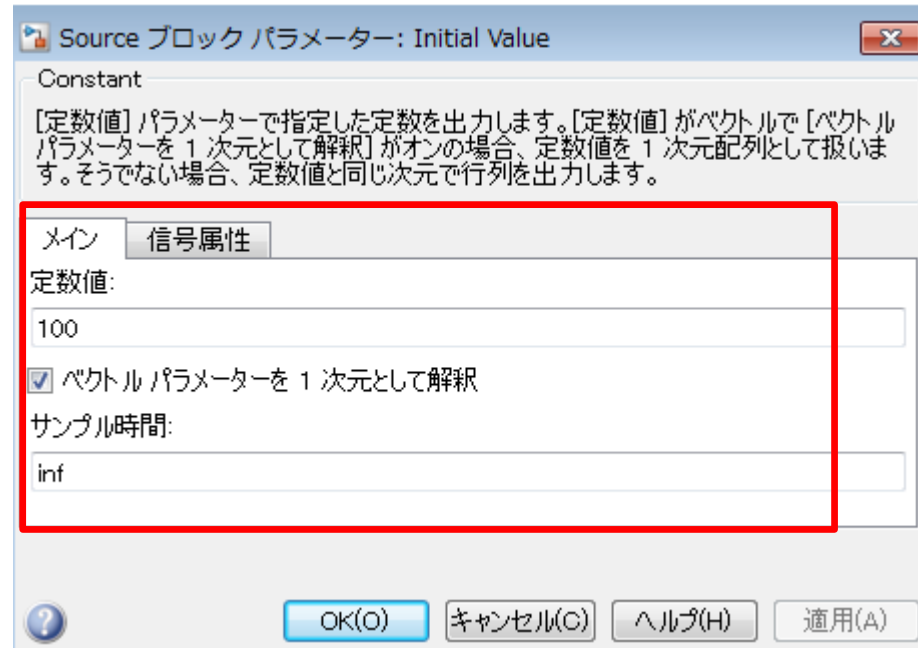
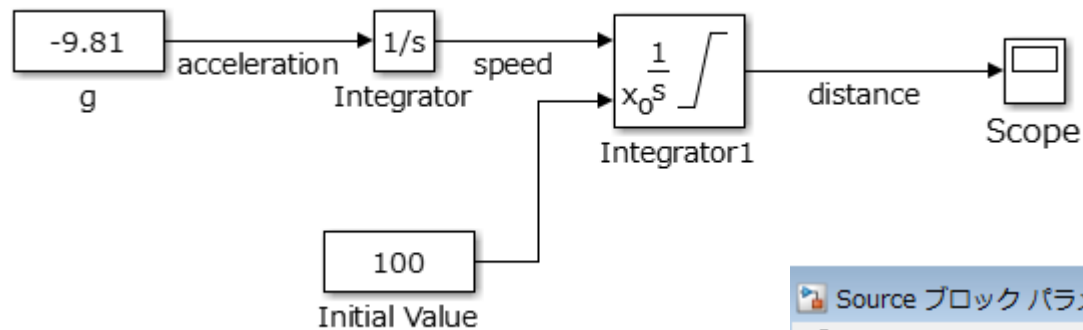


(4) ブロックパラメータを次のように設定する。



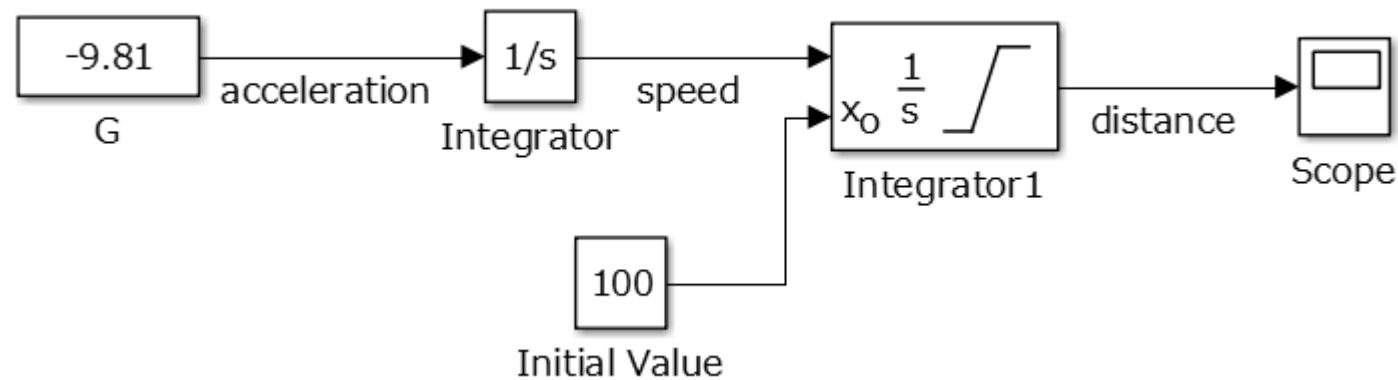
## 自由落下のモデリング手順 (3)

- (5) 後ろのIntegratorブロックに新たに、Constant ブロックを結線する。  
パラメータを以下の通りにする。



## 自由落下のモデリングで使用するブロック

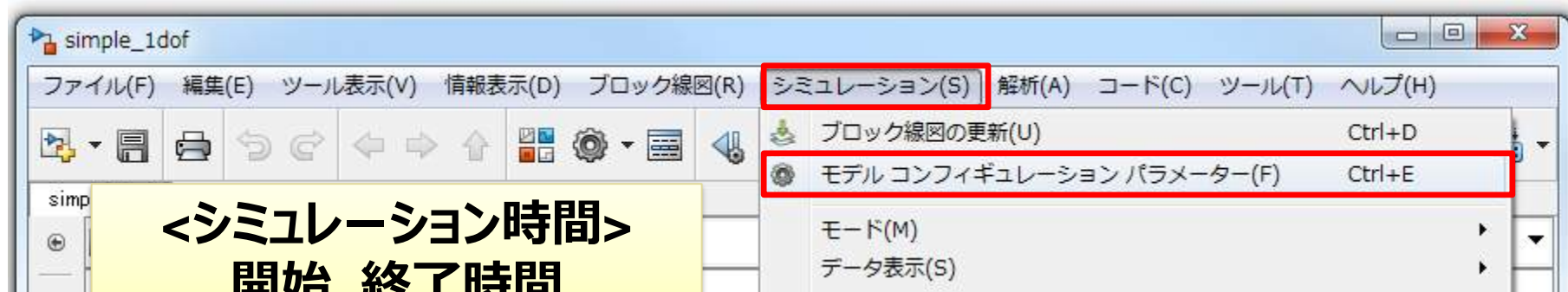
<i>Simulink</i> ライブラリの名前	ブロック名	機能
Continuous	Integrator	積分
Sources	Constant	定数
Sinks	Scope	波形観測



# **Simulinkによるシミュレーション (時間応答解析)**

- (1) 自由落下の**Simulink**モデルの作成.
- (2) コンフィギュレーションパラメータの設定.
- (3) シミュレーション実行, 出力信号の観測.

# シミュレーション時間, ソルバの設定



**<ソルバーオプション>  
積分手法の選択, ステップサイズ**

(注) **[最大ステップサイズ]**は, デフォルトの**[auto]**に設定すると,  
上記の例では, 終了時間の10.0[sec]を50で割った0.2[sec]に自動で設定される.  
今回は, **[最大ステップサイズ]**を**0.01[sec]**としてシミュレーションする.

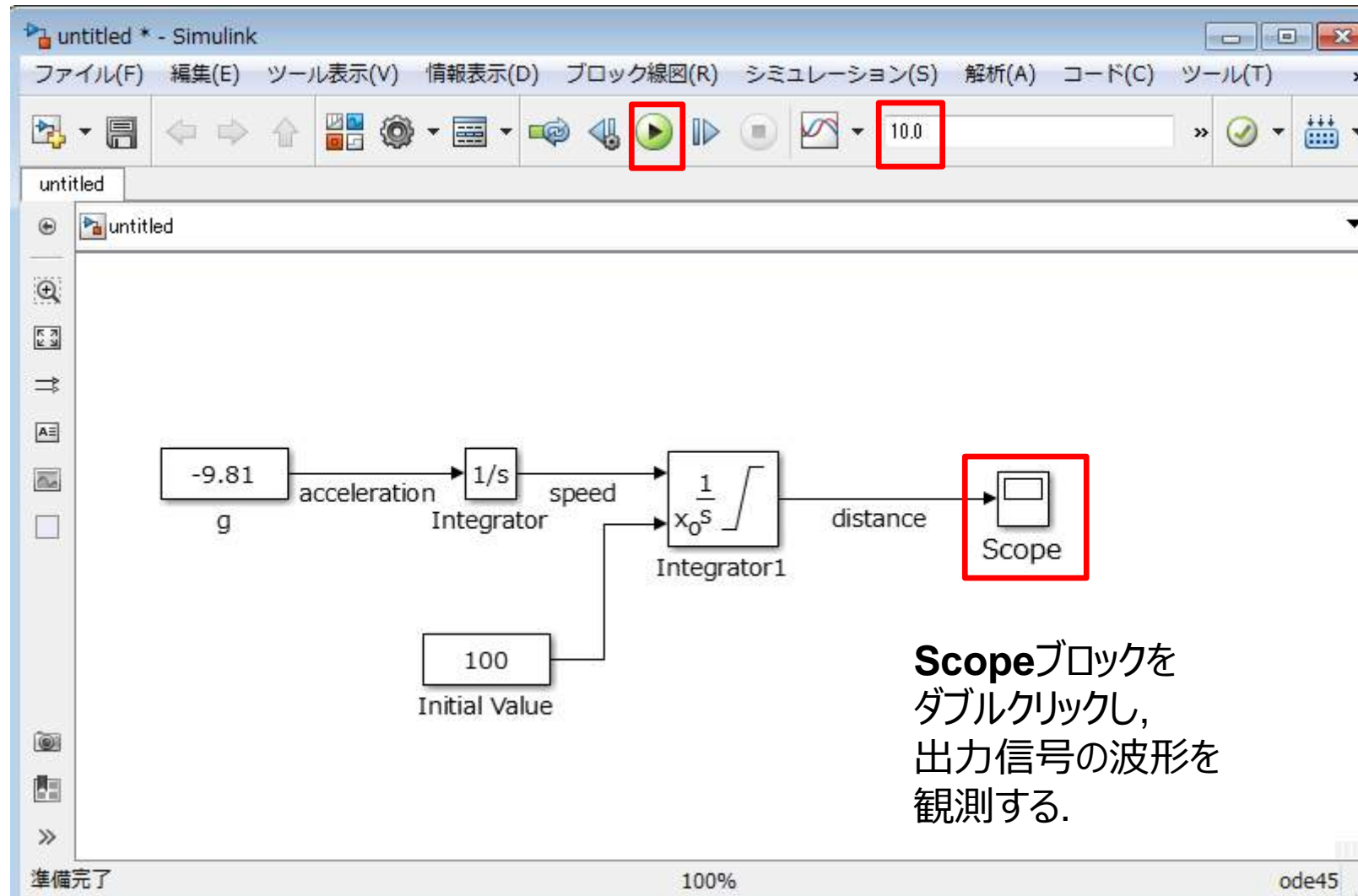


# **Simulinkによるシミュレーション (時間応答解析)**

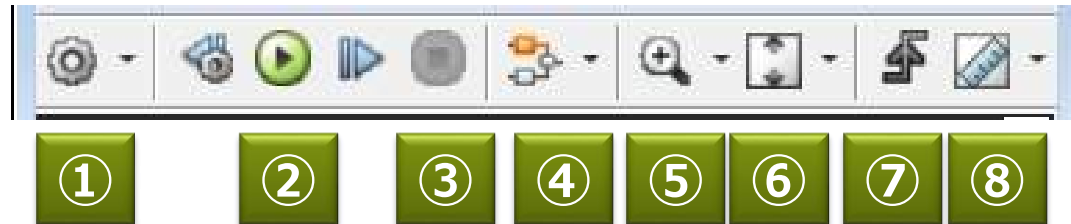
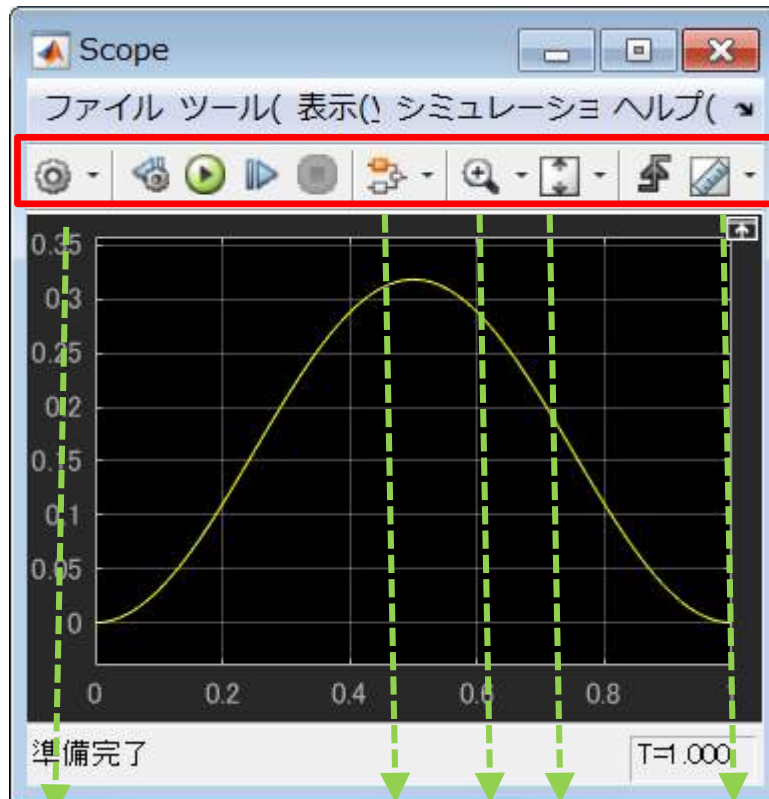
- (1) 自由落下の**Simulink**モデルの作成.
- (2) コンフィギュレーションパラメータの設定.
- (3) シミュレーション実行, 出力信号の観測.

# シミュレーションの実行

シミュレーション時間を設定し, **Simulink**モデルのシミュレーションを実行する.

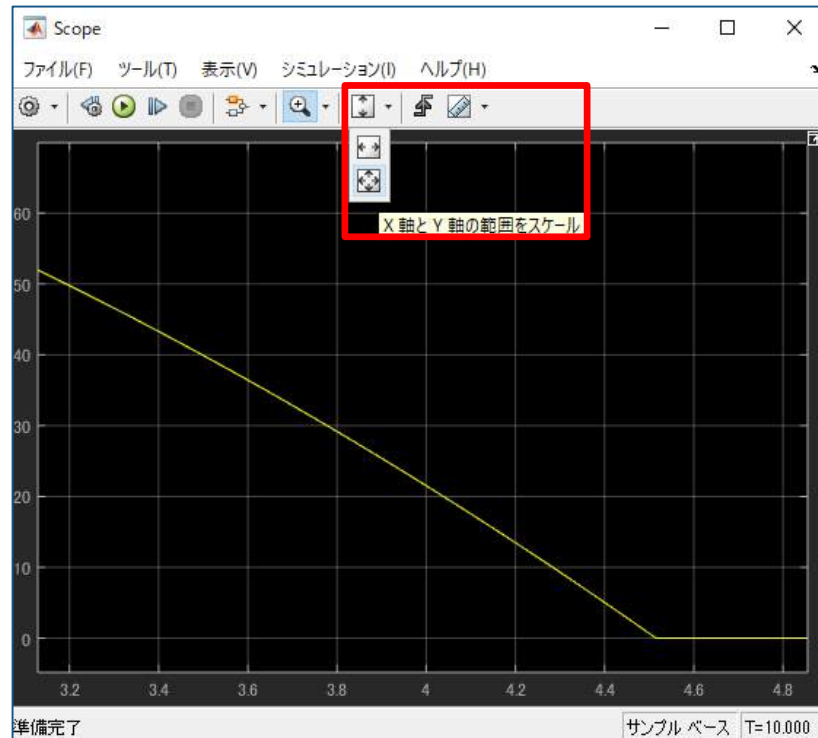


# Scopeブロックのスケール変更 (1)

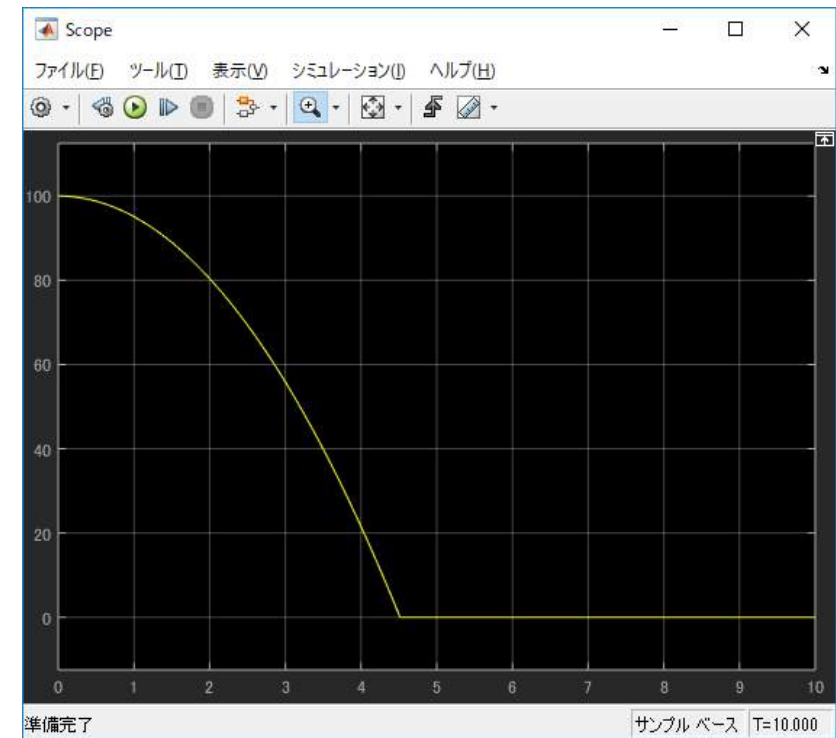


番号	説明
①	Scopeのプロパティ
②	ステップオプション - 再生 - 1ステップ
③	停止
④	ブロックの強調 (+ Scopeのスナップショット)
⑤	ズーム(+ 関連オプション)
⑥	自動スケーリング(X軸,Y軸,XY軸)
⑦	トリガー
⑧	カーソルによる測定

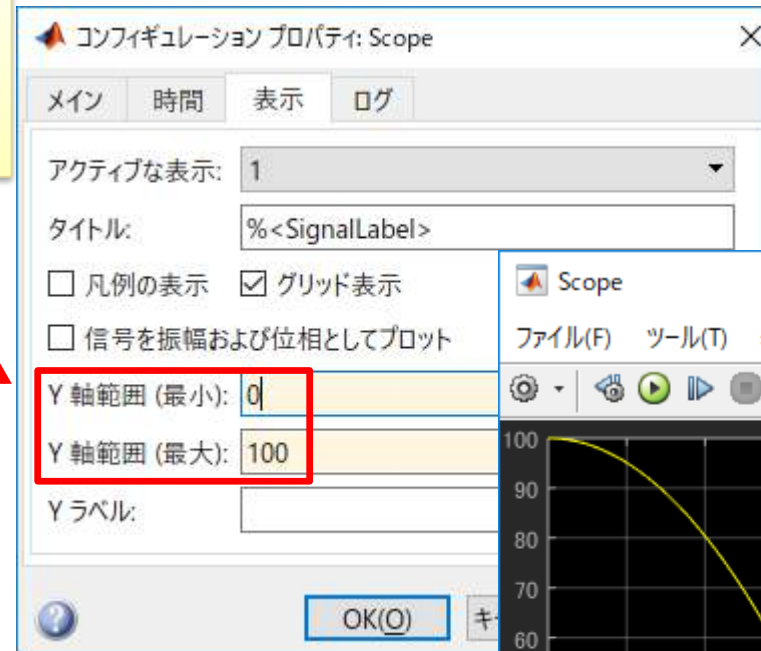
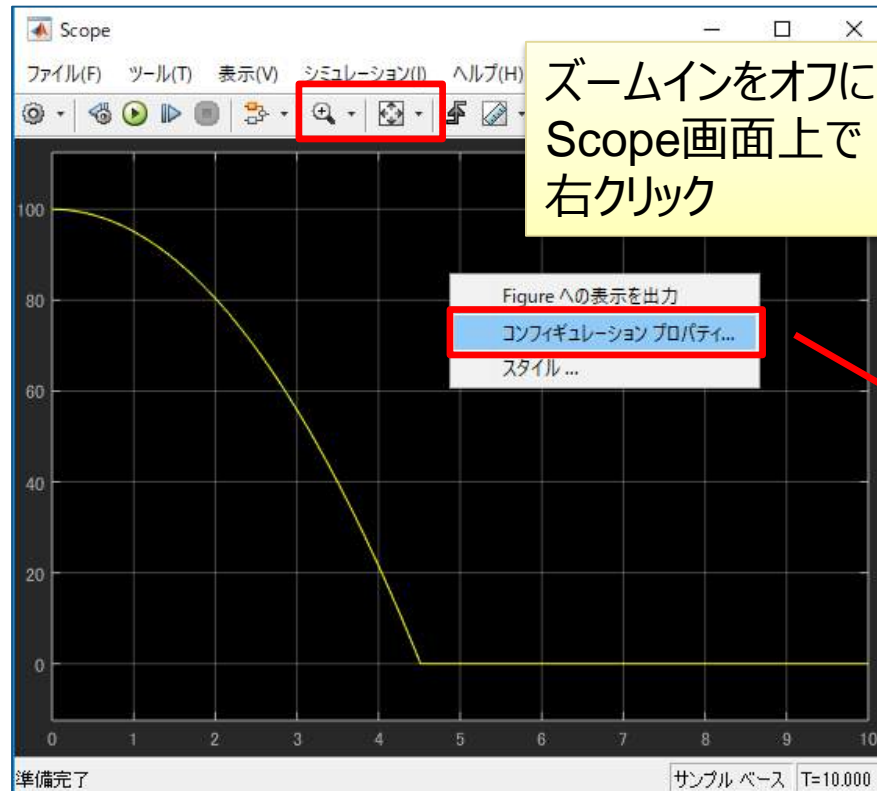
## Scopeブロックのスケール変更 (2)



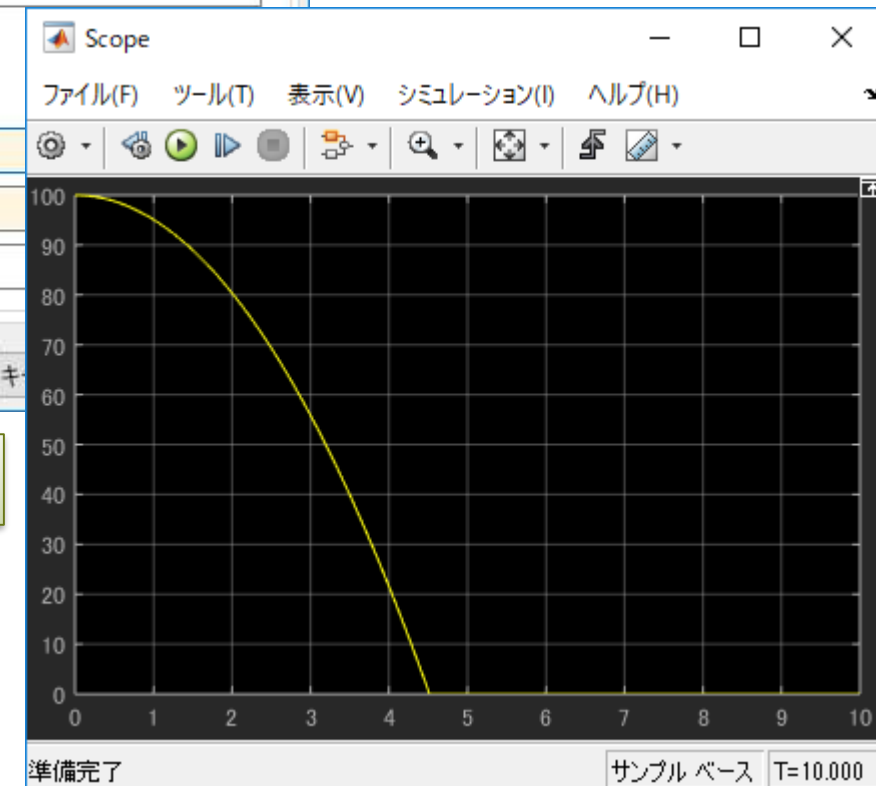
Scope画面の  
X軸とY軸の範囲  
をスケールをクリック



## 補足：Scopeブロックのスケール変更（3）

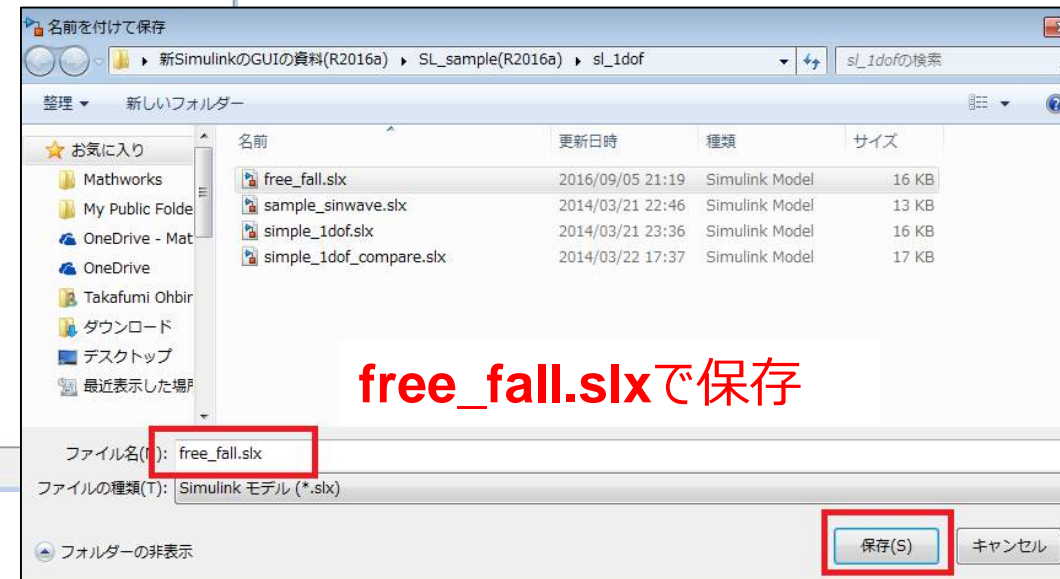
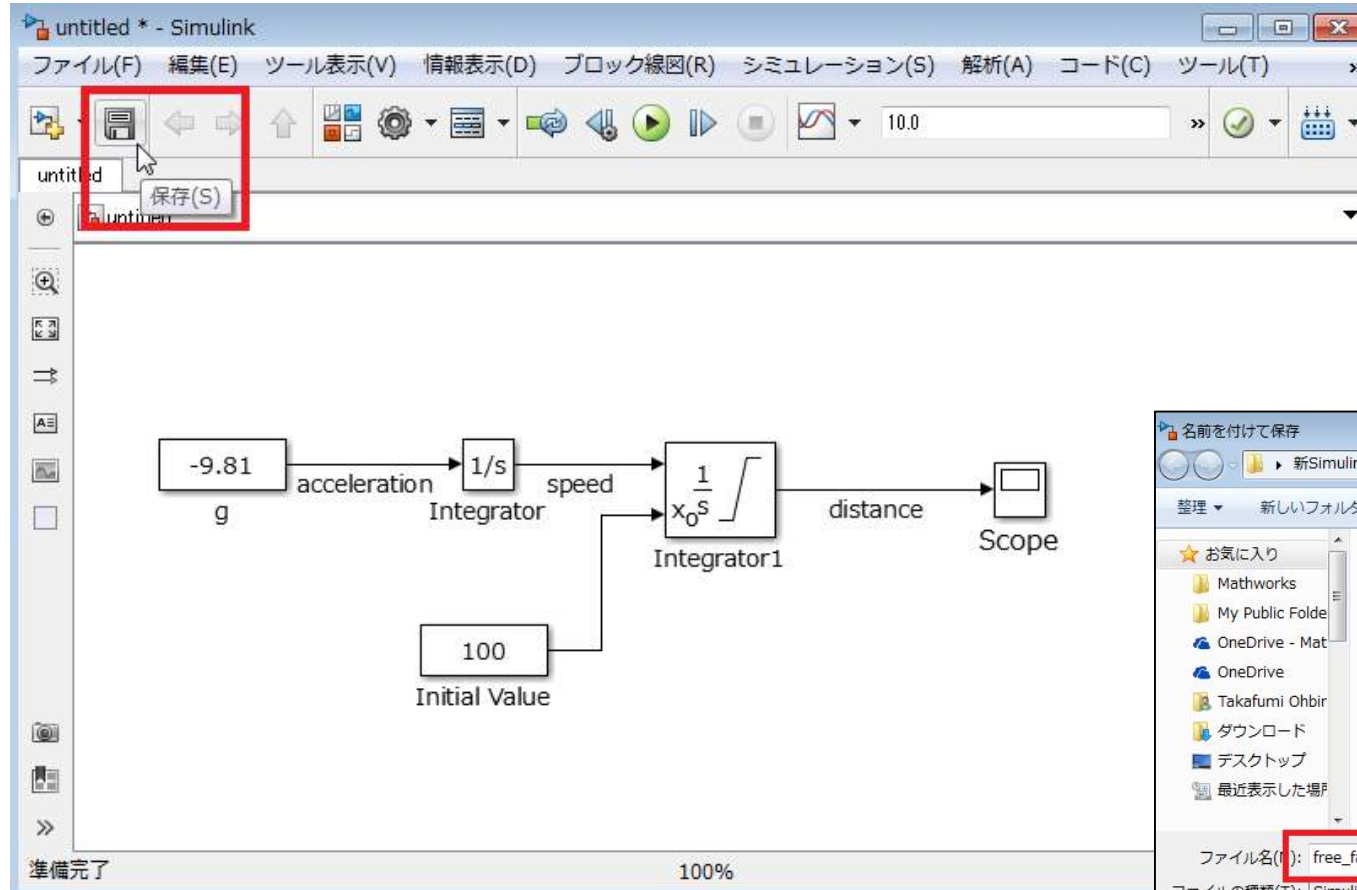


縦軸の範囲を設定





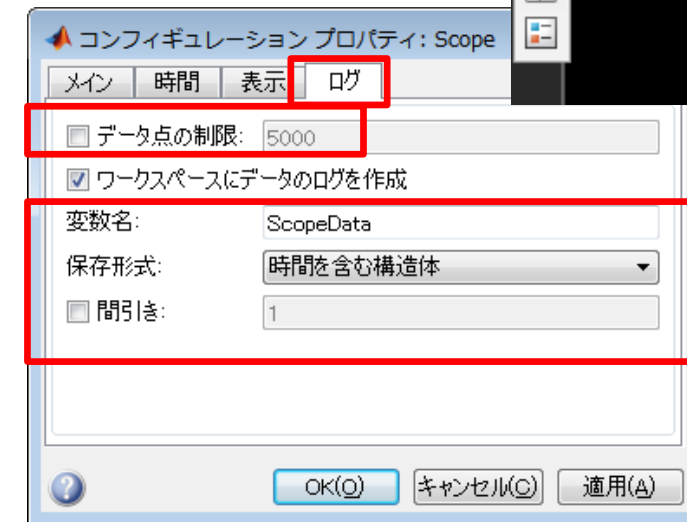
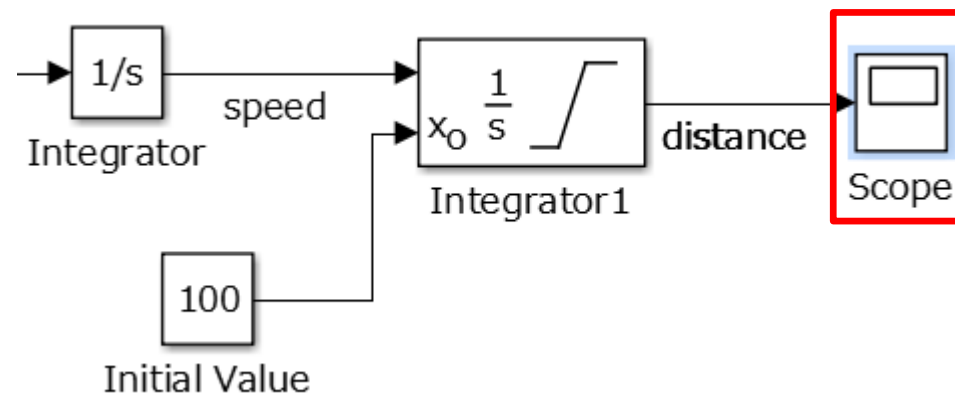
# ファイルの保存



free\_fall.slxで保存

# MATLABワークスペースにデータ保存.

**Simulink**のシミュレーション結果を, **MATLAB**ワークスペースに保存する設定を行う.



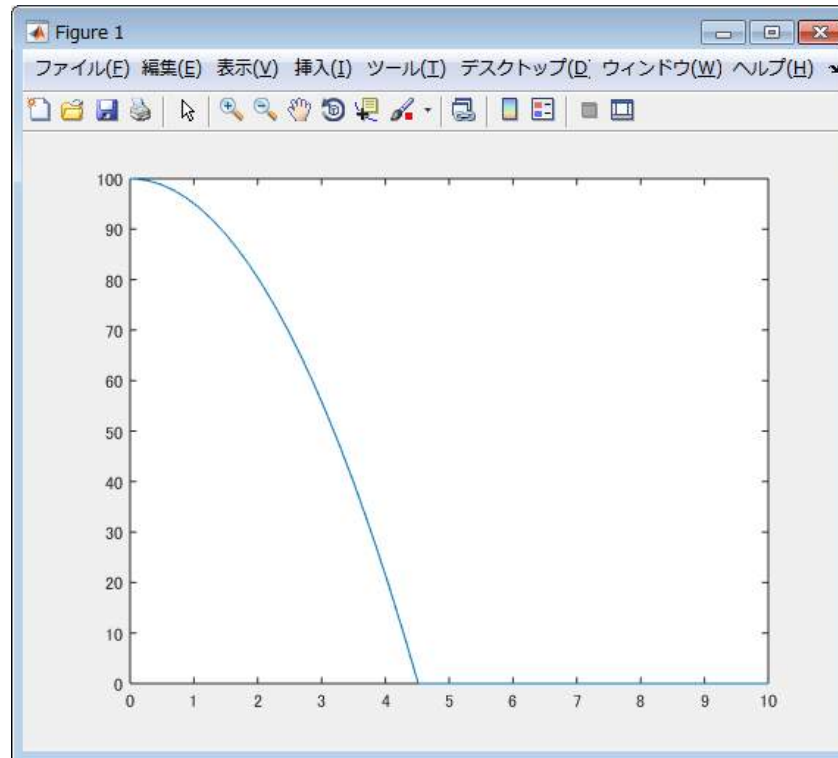
この設定を行った後に, シミュレーションを実行すると,  
**MATLAB**ワークスペース上に, シミュレーション結果が保存される..

ワークスペース	
名前	値
ScopeData	1x1 struct
tout	59x1 dou...

# MATLAB plot関数によるグラフ表示

時間付き構造体データScopeDataを, plot関数を用いてプロットする.

```
>> plot(ScopeData.time, ScopeData.signals.values)
```



R2019a をインストールしている場合

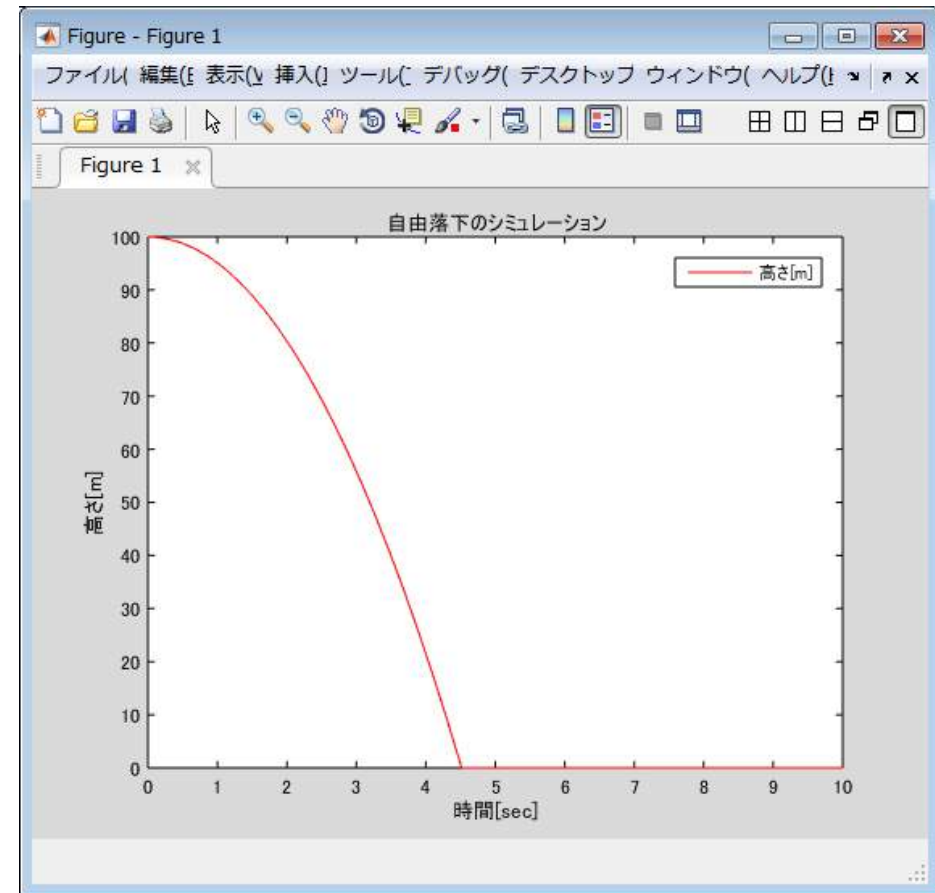
```
plot(out.ScopeData.time,out.ScopeData.signals.values)
```

# MATLABのグラフィック機能の利用

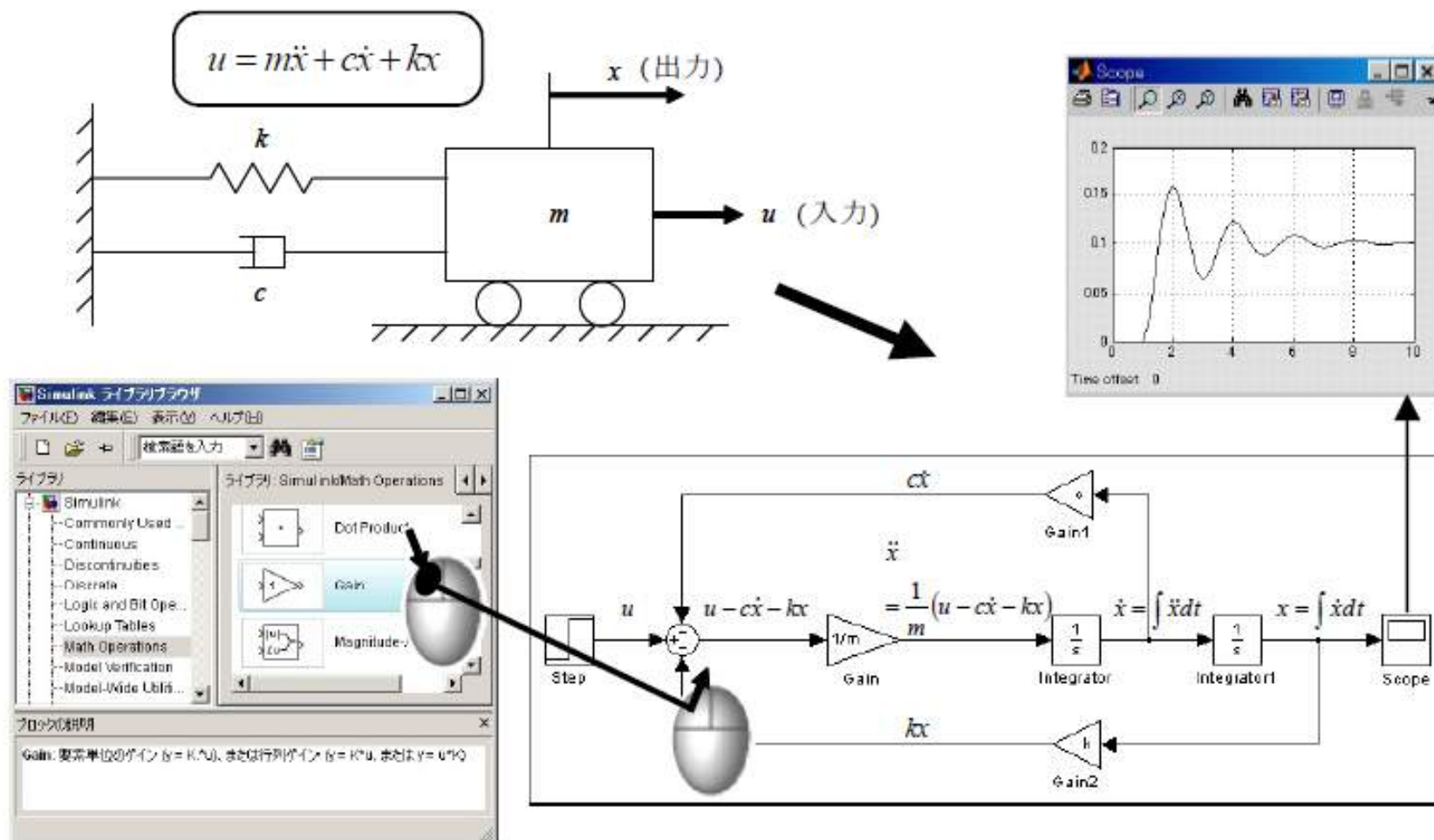
## 軸ラベル, 凡例, ラインスタイルの変更など

**plot\_free\_fall.m**

```
%% Scopeのデータ表示
plot(ScopeData.time, ScopeData.signals.values)
%% 図のハンドル番号取得
H=get(gca,'parent');
%% 図の背景を白色に設定
whitebg(H, 'white')
%% 速度,変位の図のハンドル番号取得
h=get(gca,'child');
set(h(1),'Color','r')
%% 凡例
legend('高さ[m]')
%% タイトル
title('自由落下のシミュレーション')
%% X軸, Y軸ラベル
xlabel('時間[sec]')
ylabel('高さ[m]')
```



## 演習4: 1自由度振動系のモデリングとシミュレーション

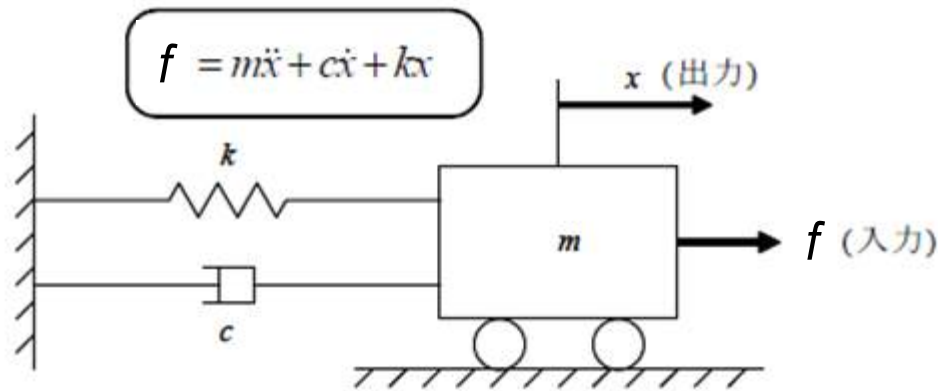


## **Simulinkによるシミュレーション (時間応答解析)**

- (1) 1自由度振動系の**Simulink**モデルの作成.
- (2) **Simulink**モデル内のパラメータ（変数）の設定.
- (3) コンフィギュレーションパラメータの設定.
- (4) シミュレーション実行し, 出力信号の観測.



# 1自由度振動系のモデリング手順 (1)



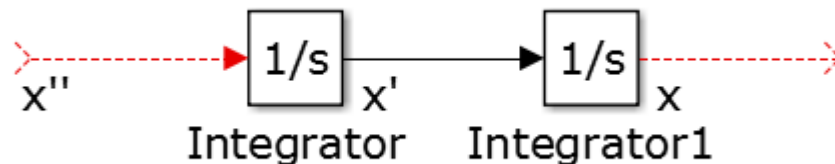
物体の質量	$m$ [kg]
ダンパの減衰係数	$c$ [Ns/m]
バネ定数	$k$ [N/m]
外力 (入力)	$f$ [N]
平衡状態からの変位(出力)	$x$ [m]
(注) 摩擦, 空気抵抗などは無視.	

微分からブロック線図に置き換える方法

- (1) 最高次の微係数について整理する.

$$\ddot{x} = \frac{1}{m}(f - c\dot{x} - kx)$$

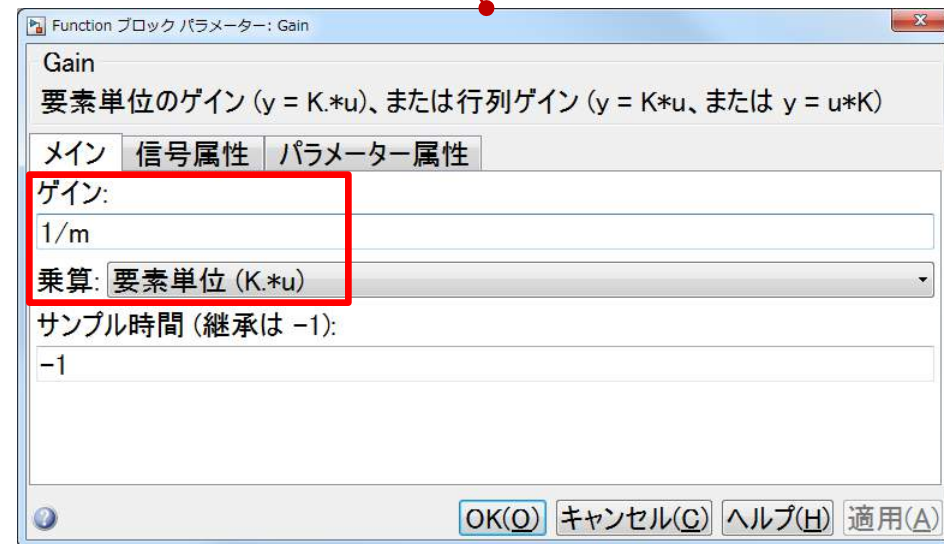
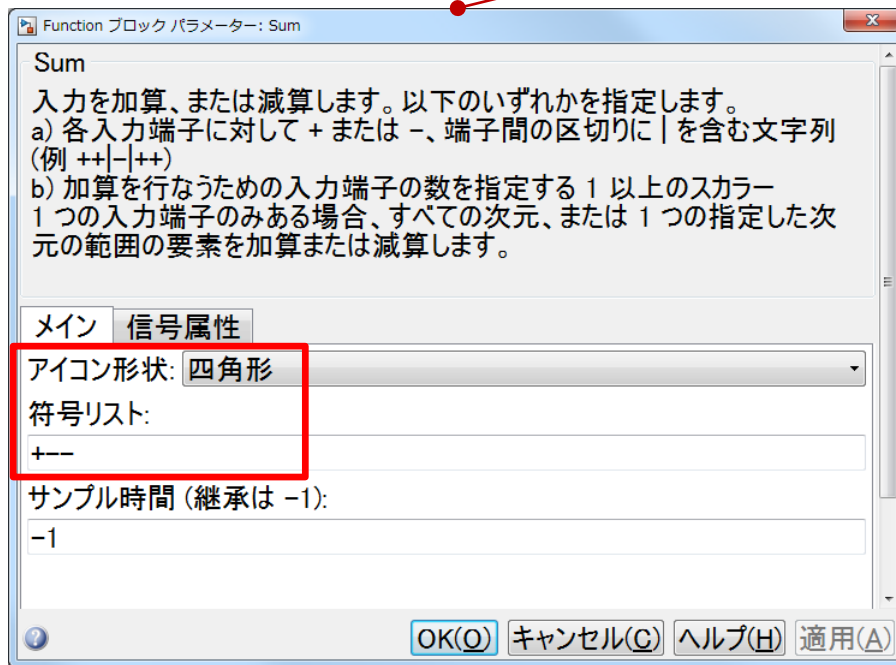
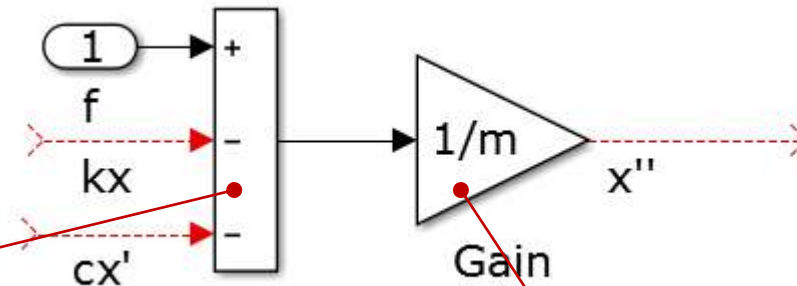
- (2) 積分器を中心にモデリングする.



# 1自由度振動系のモデリング手順 (2)

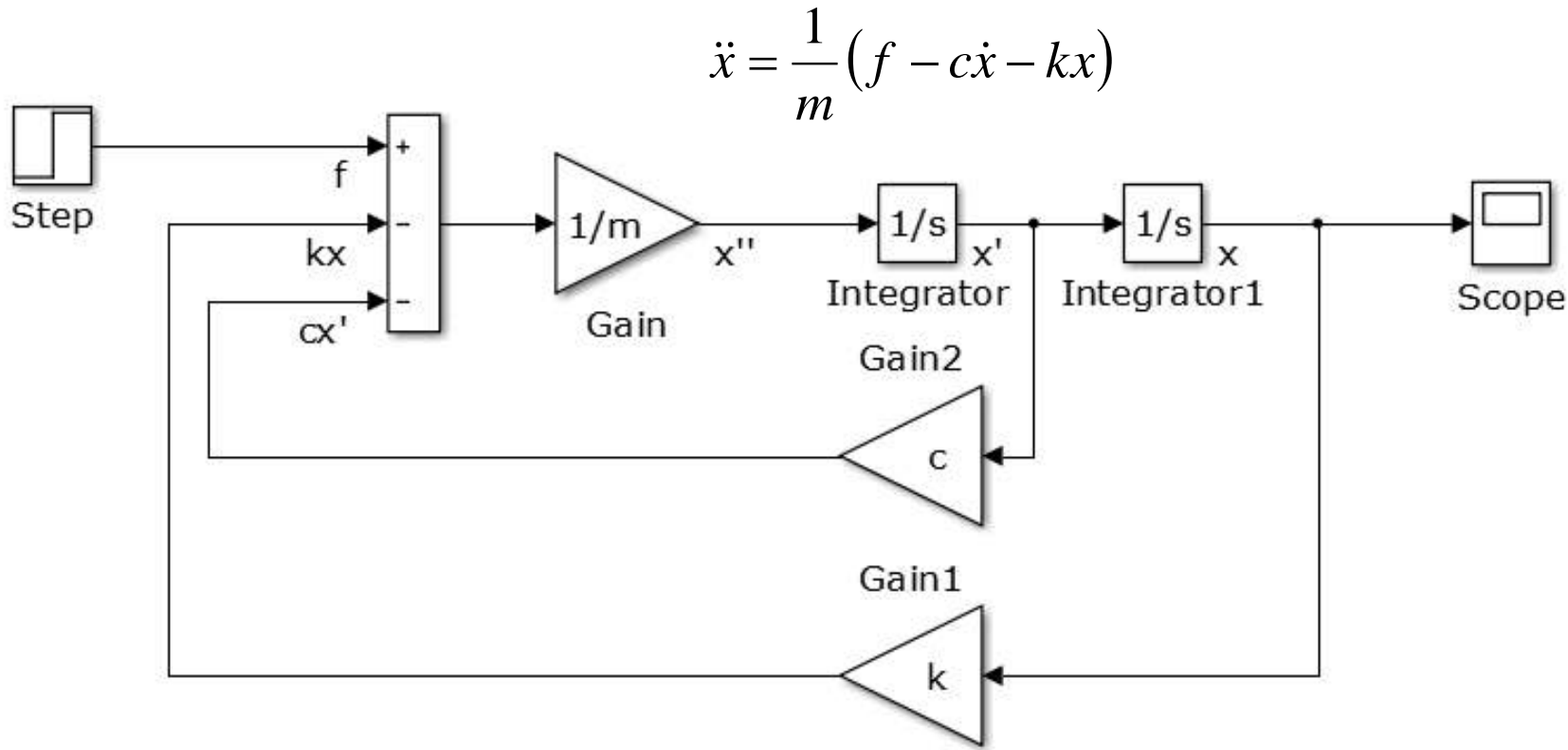
- (3) 次式のように、左辺の $x$ の2階微分が、右辺の式になるように、ブロック線図を作成する。

$$\ddot{x} = \frac{1}{m}(f - c\dot{x} - kx)$$



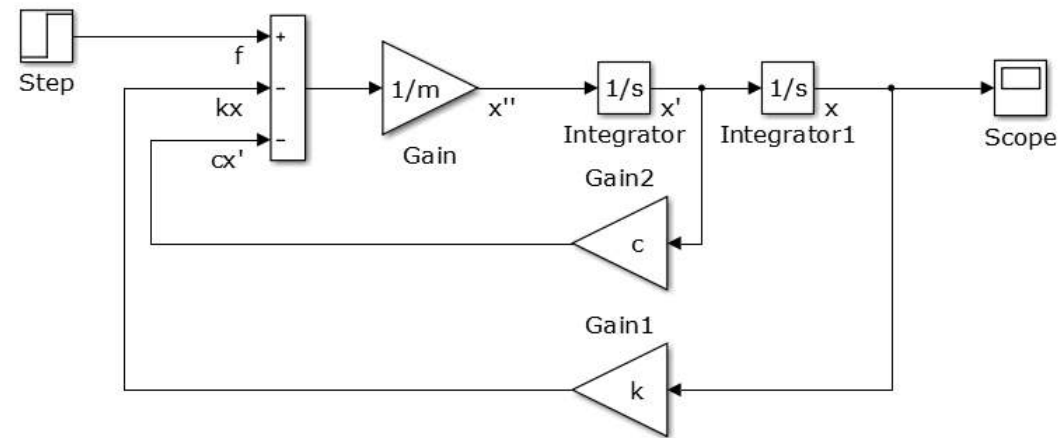
## 1自由度振動系のモデリング手順 (3)

- (4)  $x$ の一階微分を $c$ 倍,  $x$ を $k$ 倍したものを, 下図のように結線する.  
 $f$ はステップ信号とする.



# 1自由度振動系のモデリングで使用するブロック

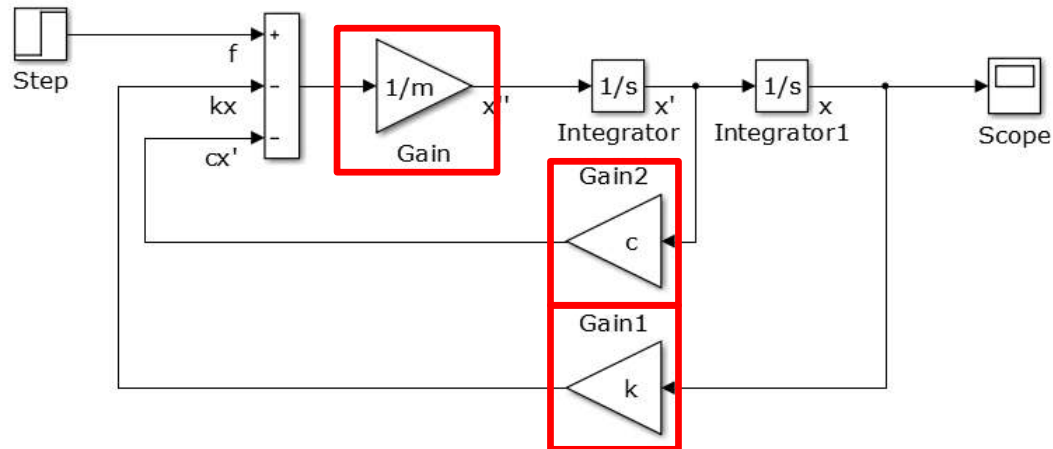
<i>Simulink</i> ライブラリの名前	ブロック名	機能
Continuous	Integrator	積分
Math Operations	Sum	加算/減算
Math Operations	Gain	ゲイン
Sources	Step	ステップ信号
Sinks	Scope	波形観測



## **Simulinkによるシミュレーション (時間応答解析)**

- (1) 1自由度振動系の**Simulink**モデルの作成.
- (2) **Simulink**モデル内のパラメータ（変数）の設定.
- (3) コンフィギュレーションパラメータの設定.
- (4) シミュレーション実行し, 出力信号の観測.

# 1自由度振動系のパラメータ設定



コマンドウィンドウ

```
MATLAB をはじめて
>> m=1;
>> c=1;
fx >> k=10;
```

**Simulink**モデル内のパラメータは、モデル内に数値を直接入力するよりも、MATLABスクリプトなどで管理する方が便利.

**para\_1dof.m**

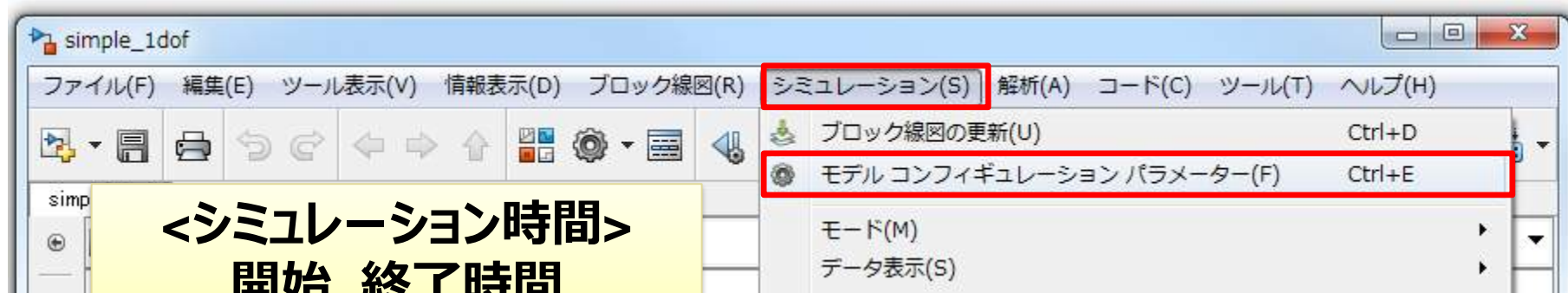
```
1 %% 1自由度振動系のパラメータ
2
3 m=1 % 物体の質量[kg]
4 c=1 % ダンパの減衰係数[Ns/m]
5 k=10 % バネ定数[N/m]
```



## **Simulinkによるシミュレーション (時間応答解析)**

- (1) 1自由度振動系の**Simulink**モデルの作成.
- (2) **Simulink**モデル内のパラメータ（変数）の設定.
- (3) コンフィギュレーションパラメータの設定.
- (4) シミュレーション実行し, 出力信号の観測.

# シミュレーション時間, ソルバの設定



**<ソルバーオプション>  
積分手法の選択, ステップサイズ**

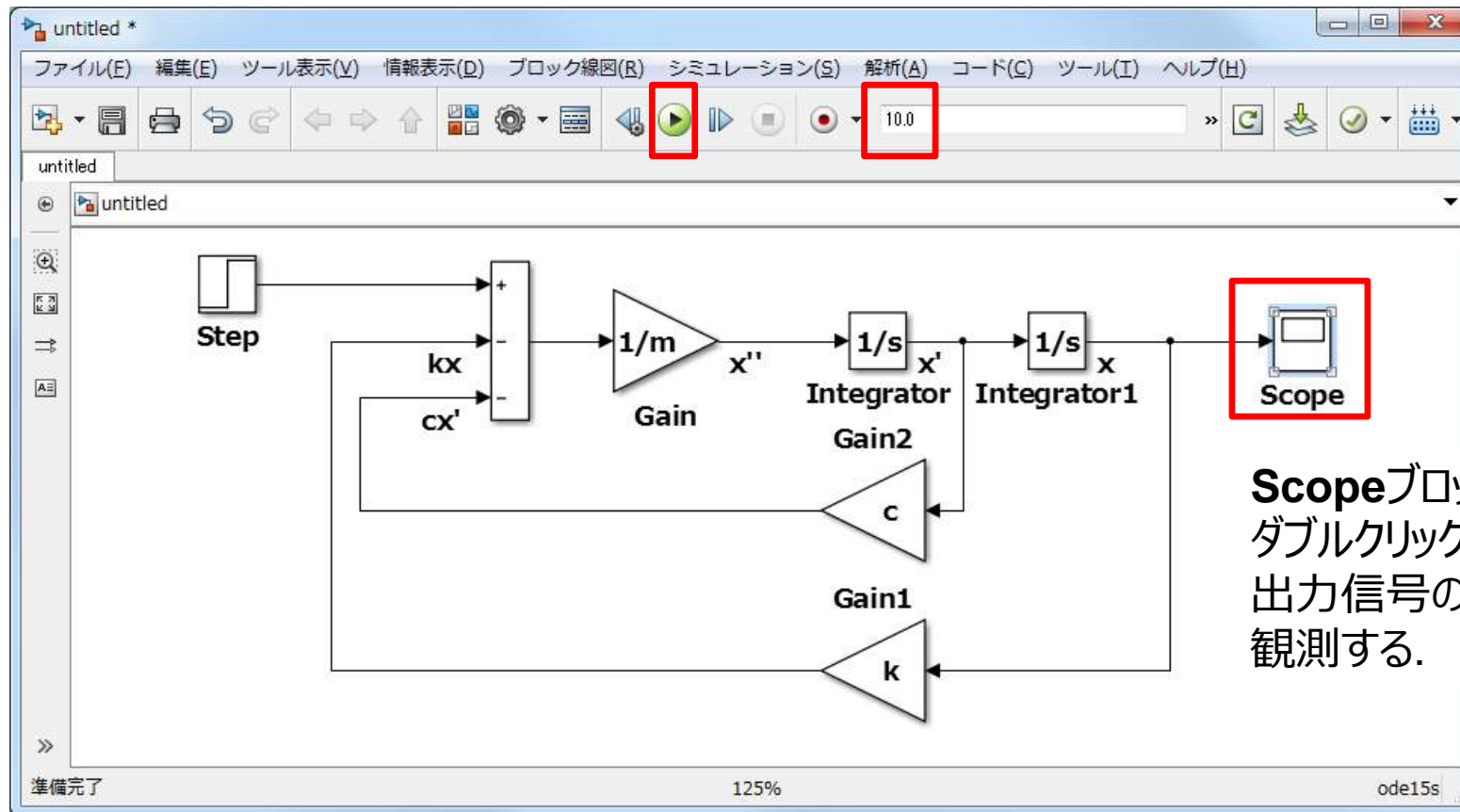
(注) **[最大ステップサイズ]**は, デフォルトの**[auto]**に設定すると,  
上記の例では, 終了時間の10.0[sec]を50で割った0.2[sec]に自動で設定される.  
今回は, **[最大ステップサイズ]**を**0.01[sec]**としてシミュレーションする.

## **Simulinkによるシミュレーション (時間応答解析)**

- (1) 1自由度振動系の**Simulink**モデルの作成.
- (2) **Simulink**モデル内のパラメータ（変数）の設定.
- (3) コンフィギュレーションパラメータの設定.
- (4) シミュレーション実行し, 出力信号の観測.

# シミュレーションの実行

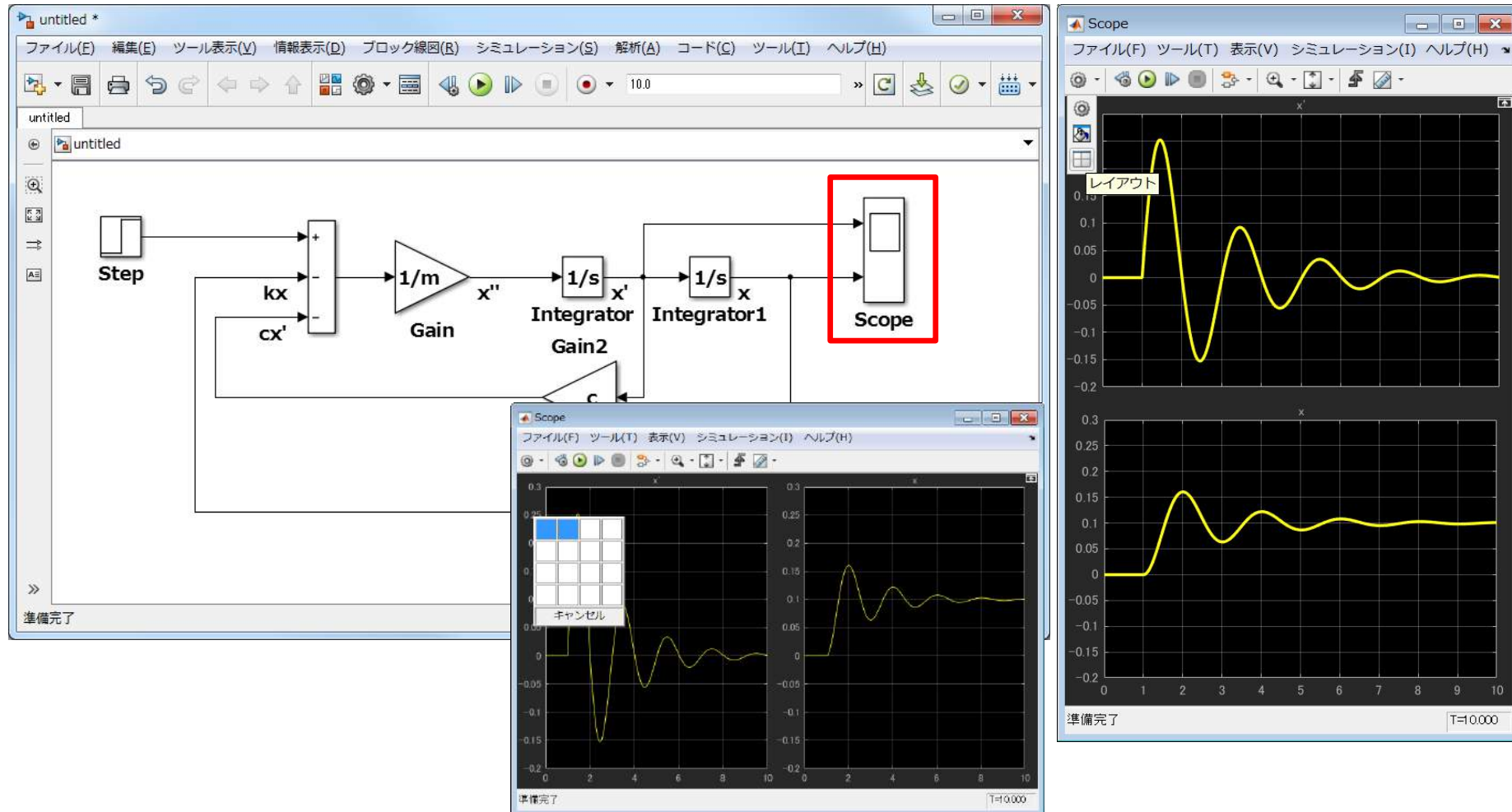
シミュレーション時間を設定し, 1自由度振動系の**Simulink**モデルのシミュレーションを実行する.



**Scope**ブロックを  
ダブルクリックし,  
出力信号の波形を  
観測する.

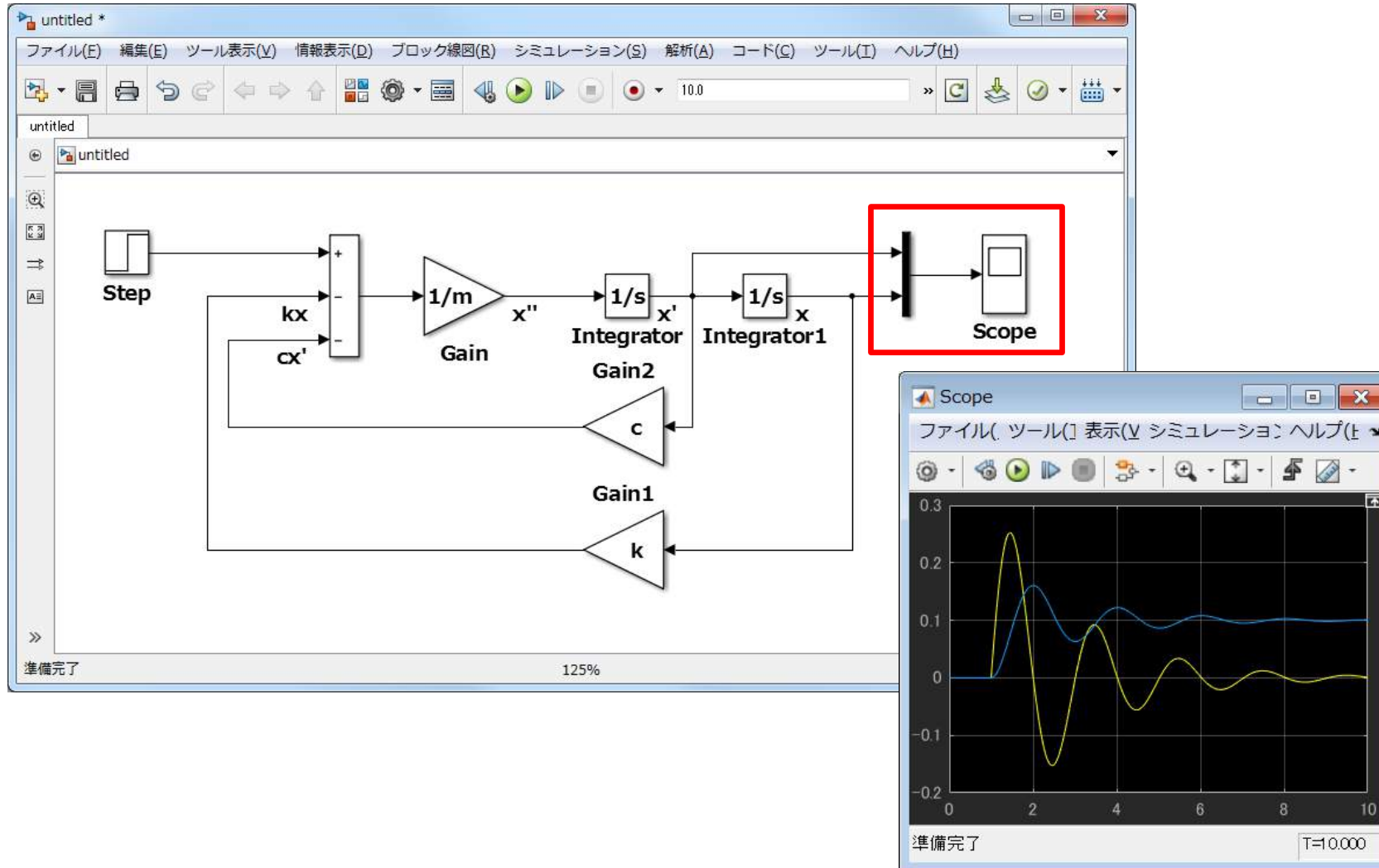
## 2つの座標軸に, それぞれの応答を表示する.

1自由度振動系の結果. (上段:速度 $x'$ [m/s], 下段:変位 $x$ [m])



# 1つの座標軸に, 2つの応答を表示する.

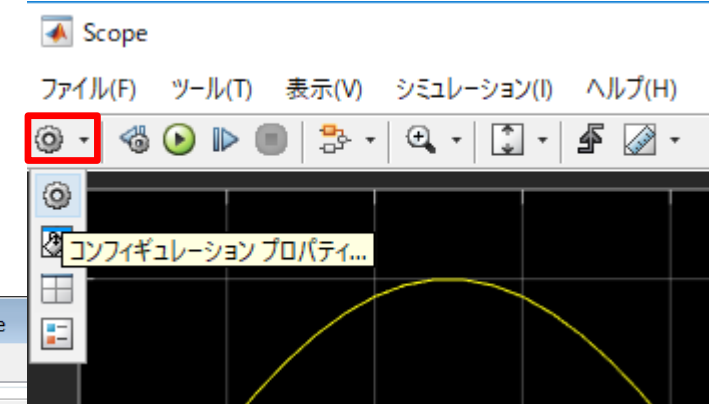
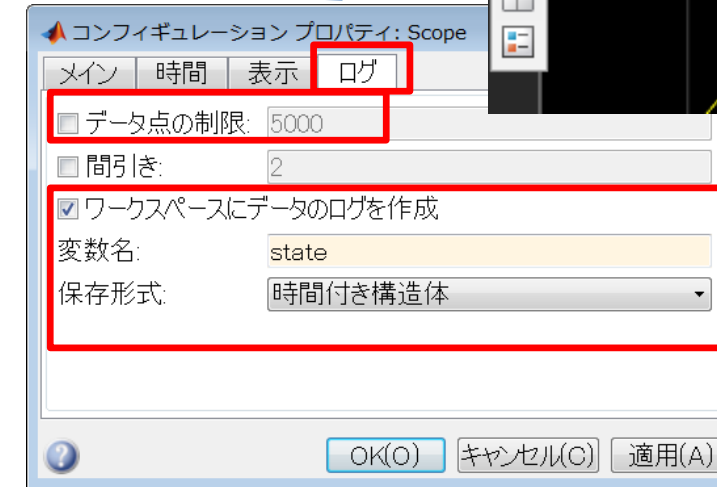
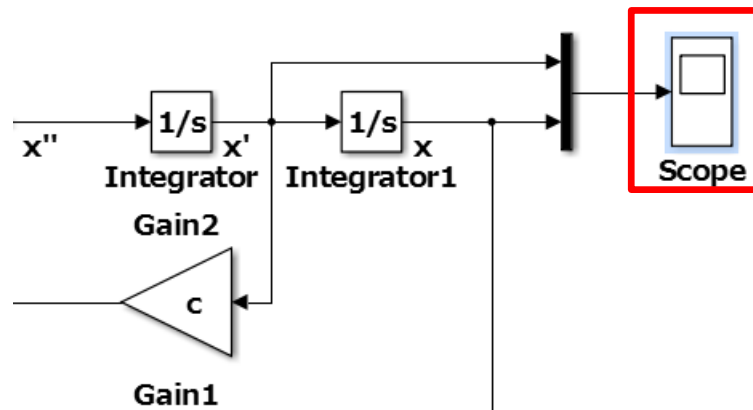
1自由度振動系の結果. (黄色:速度 $x'$ [m/s], 青:変位 $x$ [m])





# MATLABワークスペースにデータ保存.

**Simulink**のシミュレーション結果を, **MATLAB**ワークスペースに保存する設定を行う.



この設定を行った後に, シミュレーションを実行すると,  
**MATLAB**ワークスペース上に, シミュレーション結果が保存される..

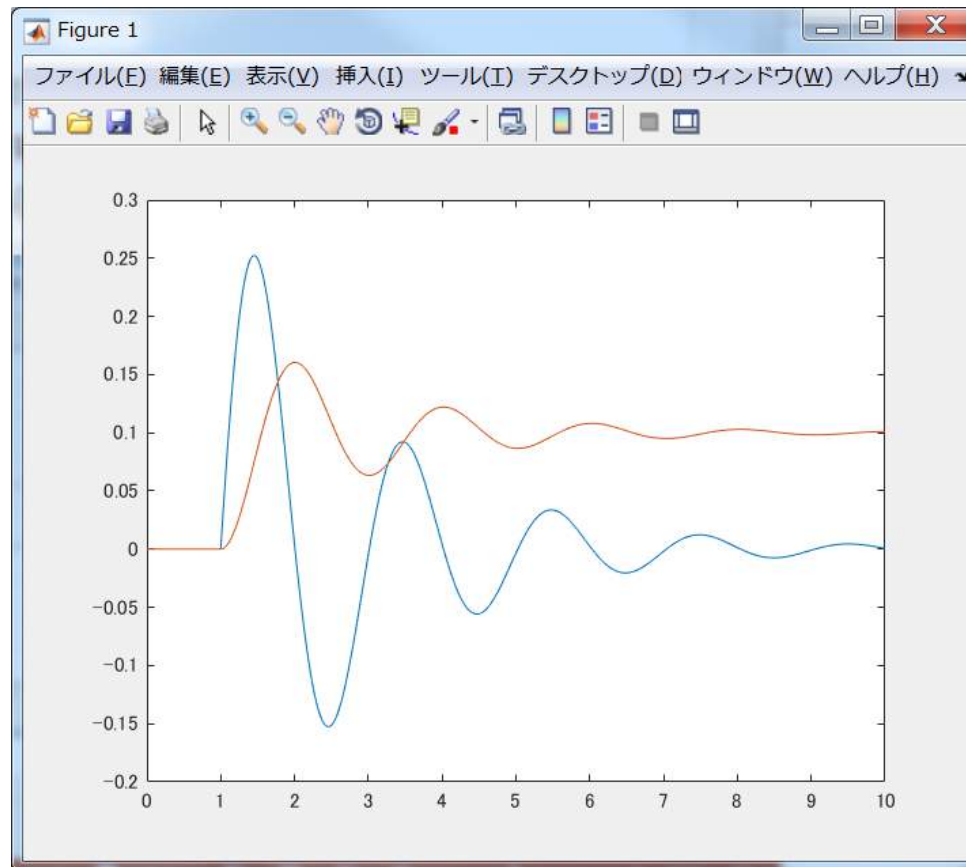
ワークスペース	
名前	値
c	1
k	10
m	1
state	<1x1 struct>

# MATLAB plot関数によるグラフ表示

時間付き構造体データ`state`を, `plot`関数を用いてプロットする.

**>> plot(state.time, state.signals.values)**

1自由度振動系の結果. (青:速度 $x'$ [rad/s], 赤:変位 $x$ [rad])



# MATLABのグラフィック機能の利用

## 軸ラベル, 凡例, ラインスタイルの変更など

**plot\_1dof.m**

%% Scopeのデータ表示

```
plot(state.time, state.signals.values)
```

%% 図のハンドル番号取得

```
H=get(gca,'parent');
```

%% 図の背景を白色に設定

```
whitebg(H, 'white')
```

%% 速度,変位の図のハンドル番号取得

```
h=get(gca,'child');
```

```
set(h(1),'Color','r')
```

```
set(h(2),'Color','b')
```

%% 凡例

```
legend('速度[m/s]', '変位[m]')
```

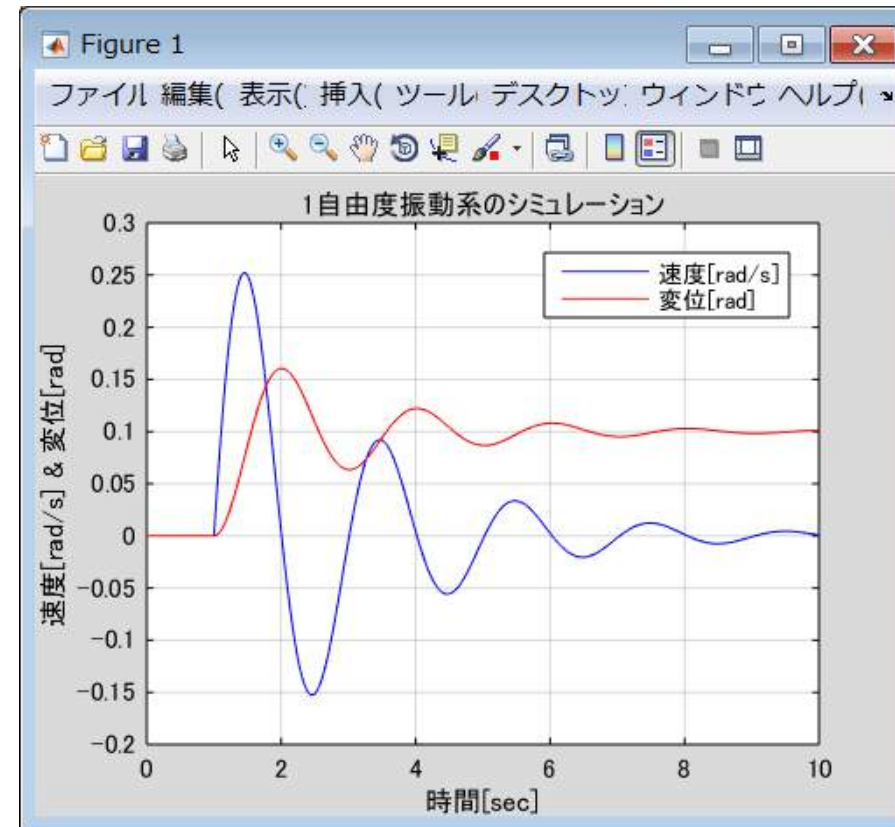
%% タイトル

```
title('1自由度振動系のシミュレーション')
```

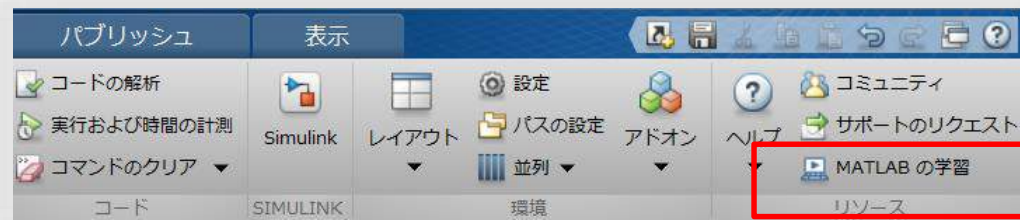
%% X軸, Y軸ラベル

```
xlabel('時間[sec]')
```

```
ylabel('速度[rad/s] & 変位[rad]')
```



# オンライン自習システムへのアクセス（無償2時間コース）



Simulink入門  
2時間コース無料

MATLABのツールストリップ(画面上部)から「MATLAB の学習」をクリック  
または「[matlab academy](#)」で検索してURLを見つけログイン

MathWorks アカウントを作成し、Simulinkのインストールが必要

初心者の方におすすめ

**MATLAB 入門 (日本語)**  
最短でMATLAB の基礎を学びましょう。  
無料で  
コースを開始    コース詳細

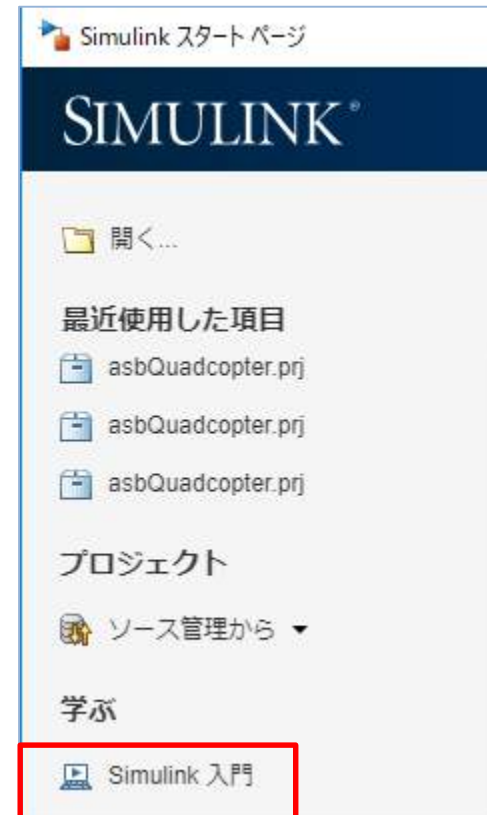
**Simulink 入門 (日本語)**  
最短でSimulinkの基礎を学びましょう。本コースはSimulinkをインストールすると受講できます。  
無料で  
コースを開始    コース詳細

**ディープラーニング入門 (日本語)**  
ディープラーニング手法を使用した画像認識を行う方法を学びましょう  
無料で  
コースを開始    コース詳細

受講するためには

R2019a 以降:

1. MATLAB®を起動して Simulink を開く。
2. スタートページで [Simulink 入門] を選択する。



# Simulink 入門オンラインコース

1.	コースの概要 コースの概要を理解する。	5 分
2.	Simulink のグラフィカル環境 Simulink のブロックと信号について学ぶ。	15 分
3.	信号の確認 シミュレーション信号を可視化する。	10 分
4.	基本アルゴリズム 数学と論理演算子を使用してアルゴリズムを記述する	15 分
5.	ヘルプの取得 Simulink からドキュメンテーションにアクセスする。	10 分
6.	プロジェクト:自動車の燃費のモデリング 数学と論理演算子を使用した実践作業。	15 分

7.	Simulink および MATLAB Simulink で MATLAB の変数と関数を使用する。	10 分
8.	Simulink での動的システム 動的システムを確認し、Simulink でのモデリングについて学ぶ。	5 分
9.	離散システム 離散時間システムをモデリングする。	10 分
10	連続システム ・ 連続時間システムをモデリングする。	10 分
11	シミュレーション時間 ・ シミュレーション時間を設定する。	10 分
12	プロジェクト:サーモスタットのモデリング ・ 離散動的システムの理解を深める。	15 分
13	プロジェクト:ハヤブサのモデリング ・ 連続動的システムの理解を深める。	20 分



© 2019 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://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.