Simulinkハンズオン資料

Roboone MATLABセミナー　2021/4/24

MathWorks

MATLAB/Simulinkの便利な使い方

番号	場所	内容
1	(MATLAB) コマンドウィンドウ	a***** 頭文字aの候補の呼び出し a→[Tab]キー
2	(MATLAB) mファイル エディタ	選択範囲の一括実行 範囲選択→[F9]キー
3	(MATLAB) 初期化の基本	clc;close all;clear all;bdclose all
4	(MATLAB) グラフの分割	subplot(行,列,番号)
5	(MATLAB) グラフの上書き	hold on hold off
6	(MATLAB) 線種の情報	doc linespec
7	(MATLAB) MATLAB全般の 記号の一覧表	doc 特殊文字

番号	場所	内容
8	サンプル デモ一覧	(MATLAB) demo
9	help マニュアル	(MATLAB) 関数を調べる 例： doc plot
10	Simulinkブロック のコピー/ペースト	(Simulink) ブロック　右クリック　Drug&Drop
11	Simulinkブロック のモデルウィンド ウでの呼び出し	(Simulink) 白い空白部で　ブロック名を入力
12	ブロックの検索	(Simulink) キーワードを検索に入力
13	線上で分岐	(Simulink) 線上で右クリック
14		

微分と積分の関係(並進系)
位置、速度、加速度

$$\frac{d}{dt} \qquad \frac{d^2}{dt^2}$$

$$x \rightarrow \dot{x} = \left(\frac{dx}{dt} \right) \rightarrow \ddot{x} \left(= \frac{d^2x}{dt^2} \right)$$

$$\int_0^t \dot{x}(\tau) d\tau \qquad \int_0^t \ddot{x}(\tau) d\tau$$

$$x \leftarrow \dot{x} = \left(\frac{dx}{dt} \right) \leftarrow \ddot{x} \left(= \frac{d^2x}{dt^2} \right)$$

注: $x(0) = \ddot{x}(0) = \ddot{x}(0) = 0$ とする。

微分と積分の関係(回転系)
角度、角速度、角加速度

$$\frac{d}{dt} \qquad \frac{d^2}{dt^2}$$

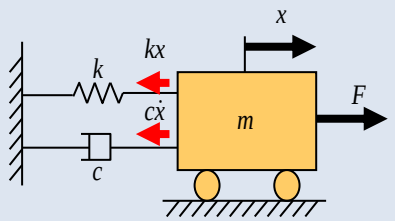
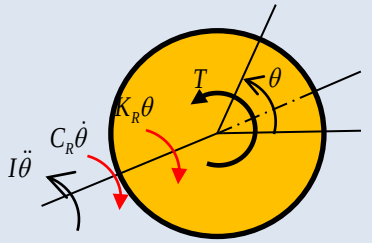
$$\theta \rightarrow \dot{\theta} = \left(\frac{d\theta}{dt} \right) \rightarrow \ddot{\theta} \left(= \frac{d^2\theta}{dt^2} \right)$$

$$\int_0^t \dot{\theta}(\tau) d\tau \qquad \int_0^t \ddot{\theta}(\tau) d\tau$$

$$\theta \leftarrow \dot{\theta} = \left(\frac{d\theta}{dt} \right) \leftarrow \ddot{\theta} \left(= \frac{d^2\theta}{dt^2} \right)$$

注: $\theta(0) = \dot{\theta}(0) = \ddot{\theta}(0) = 0$ とする。

一自由度振動系　機械：回転系と並進系の相似

並進の力学(質点)		回転の力学(剛体)	説明
図式			
方程式	$m\ddot{x}+c\dot{x}+kx=F$	$I\ddot{\theta}+C_R\dot{\theta}+K_R\theta=T$	式の構造は両式共通
慣性項	$m\ddot{x}$	$I\ddot{\theta}$	動き易さ、止め易さ 回り易さ、回りにくさを表す項
バネ項	kx	$K_R\theta$	振動を発生させる項
粘性項	$c\dot{x}$	$C_R\dot{\theta}$	運動エネルギーを減衰させようとする項
外力	力 F	トルク T	

モデリング（直接表現）

運動方程式は

$$m\ddot{x} = -(c\dot{x}) - (kx) + u$$

整理すると

$$m\ddot{x} + c\dot{x} + kx = u$$

初期値は $\ddot{x}(0) = 0, \dot{x}(0) = 0, x(0) = 0$ とする。

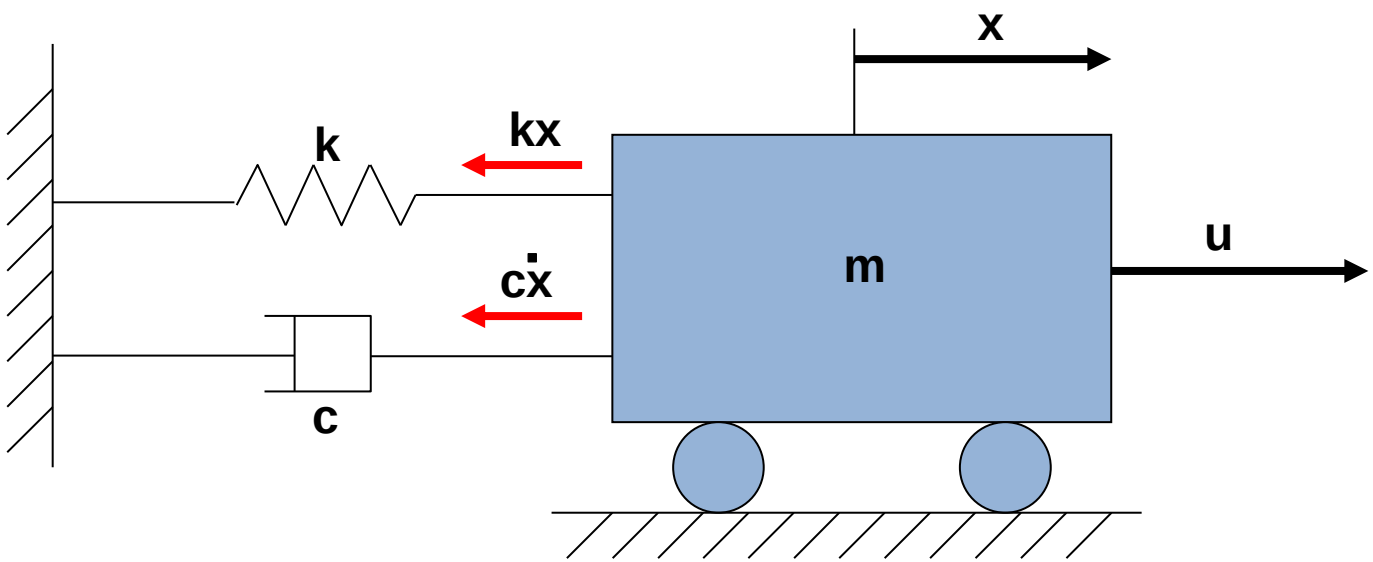


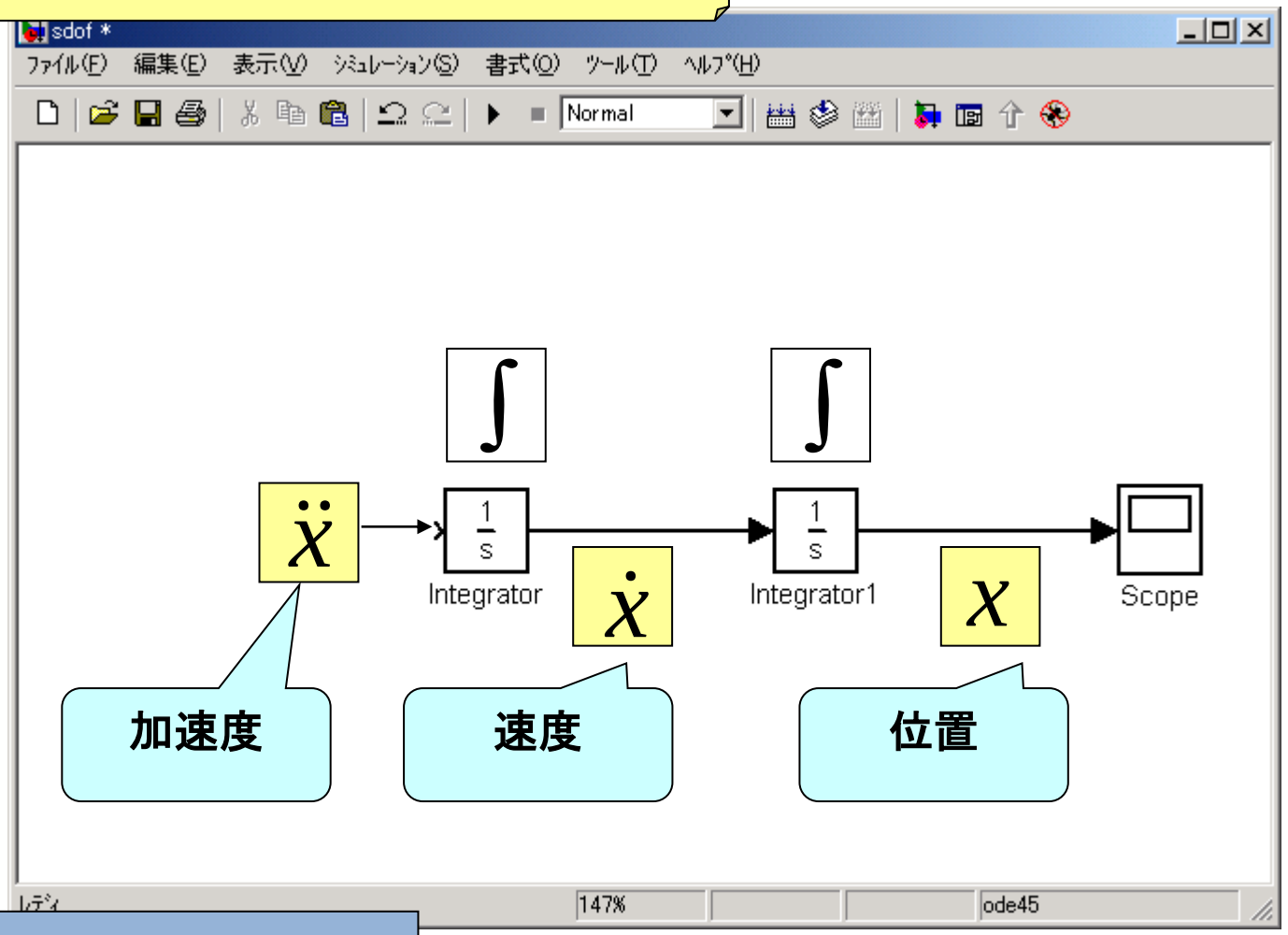
図　機械系ダイナミクスの例（1自由度振動系）

Simulink連続モデルの作成の基本原則

- STEP0　準備
  - 微分回数最大の変数を左辺に移項、残りを右辺に移項
- STEP1　　微積分変数の位置関係
  - 式に登場する微分積分の関係を、積分器(Integrator)を使って表現する。
- STEP2　　右辺　各項の要素　作成
  - 式の右辺に登場する項を、線を分岐させ作成する。
- STEP3　　左辺と右辺の整合性
  - 右辺と左辺の整合性をモデルに表現する。

どんな複雑なモデルでも基本原則は変わらない。

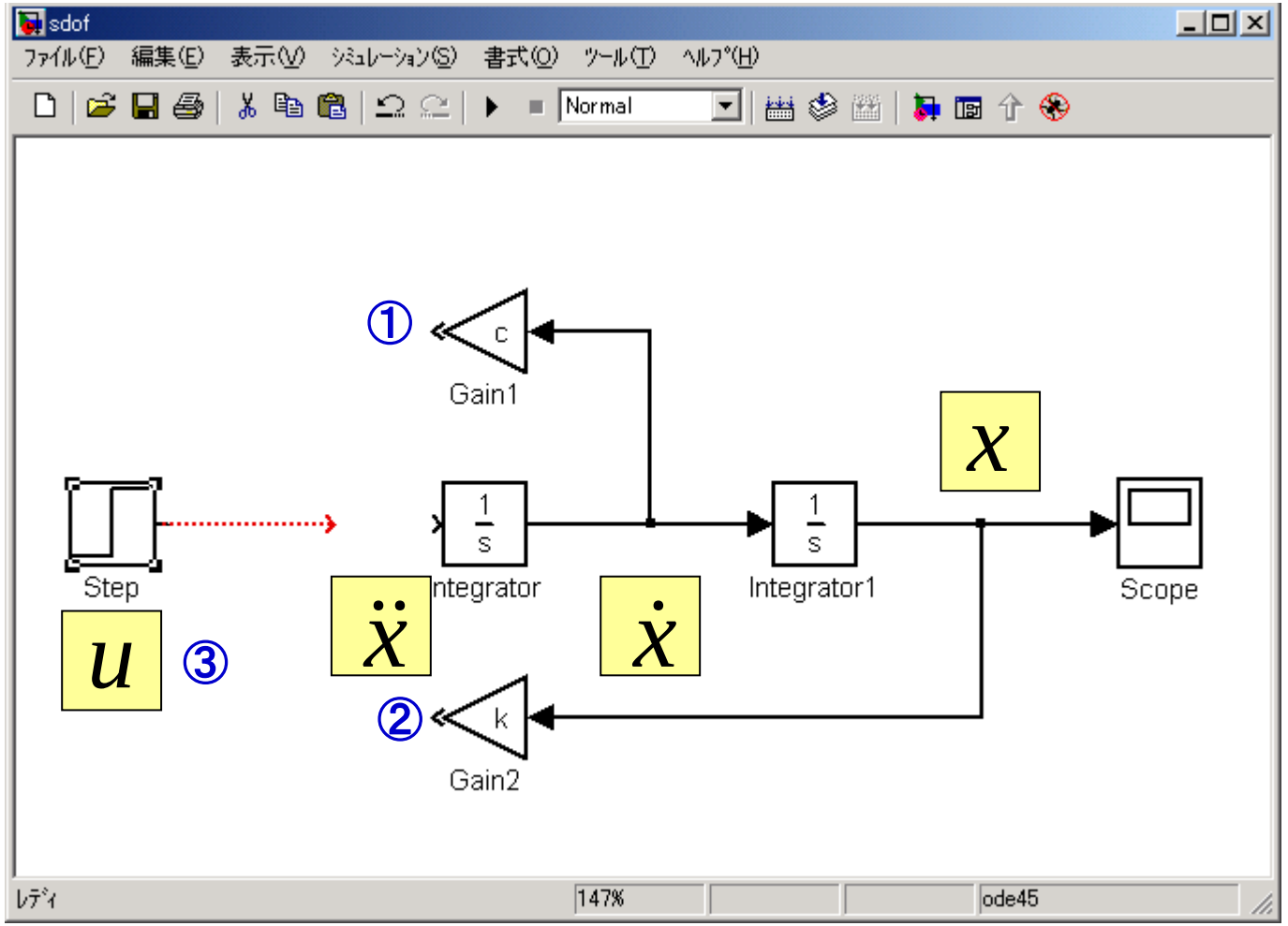
STEP1　微積分変数の位置関係



2階の微分方程式

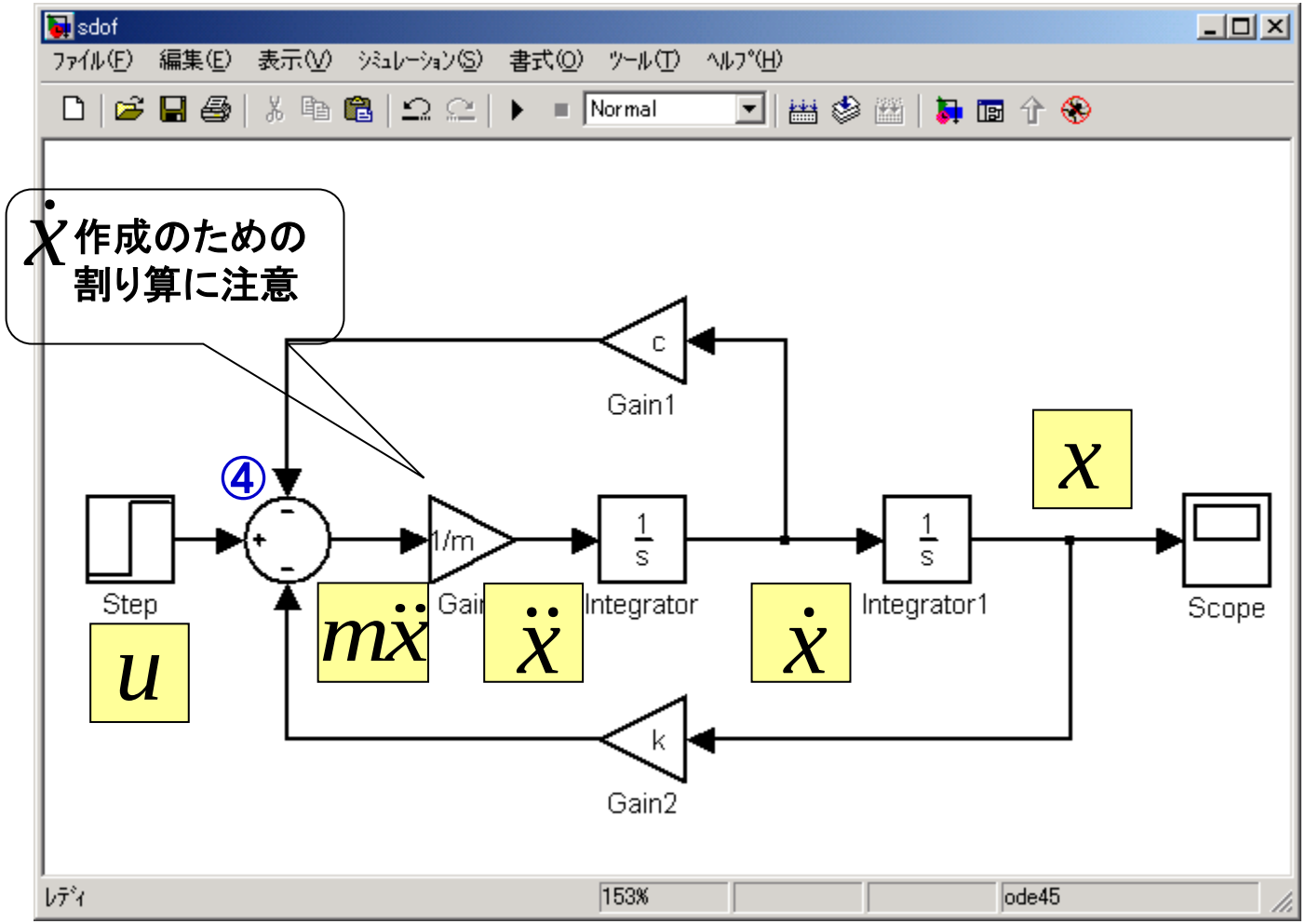
STEP2　　右辺　各項の要素　作成

$$m\ddot{x} = -\underset{①}{(c\dot{x})} - \underset{②}{(kx)} + \underset{③}{u}$$

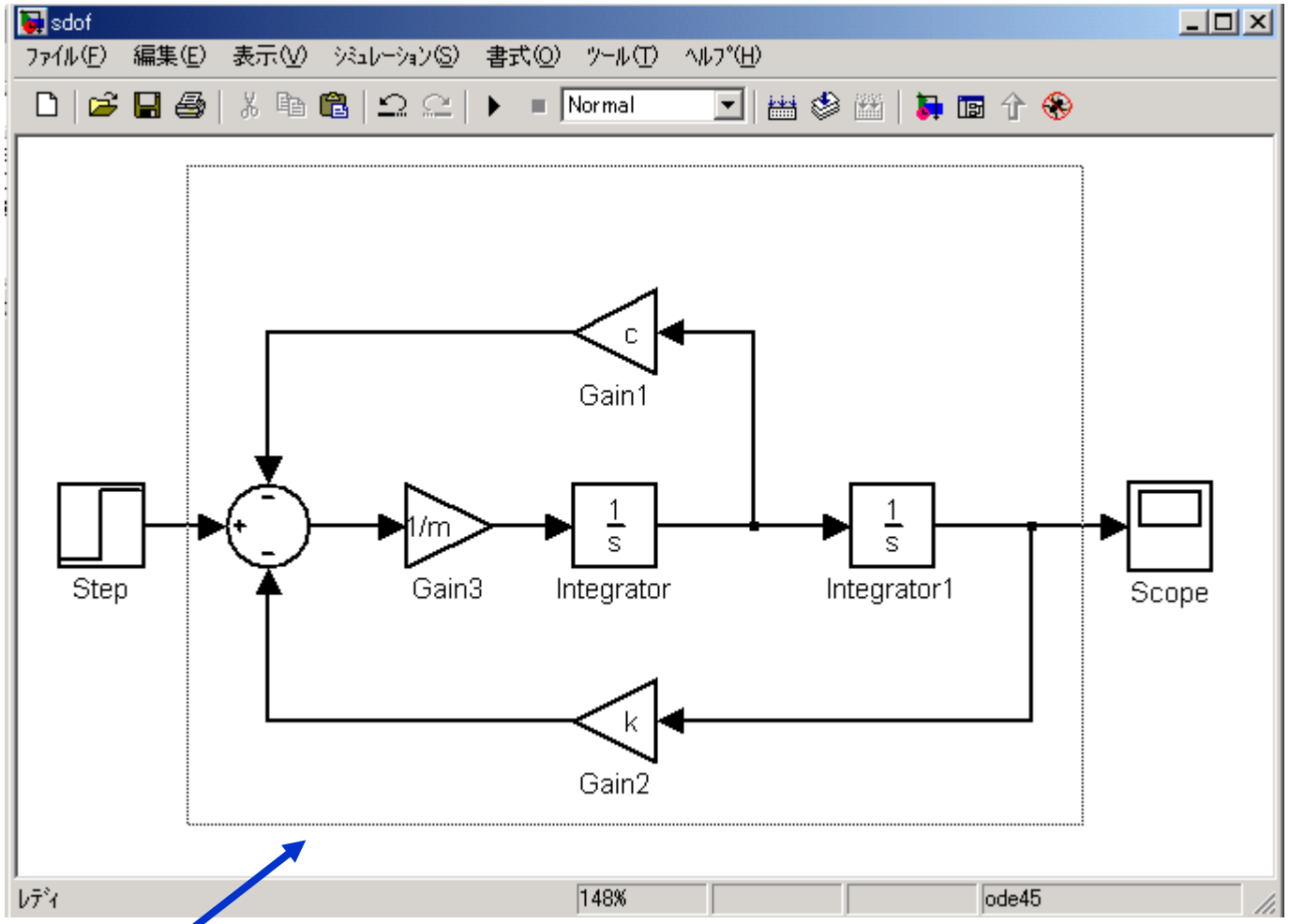


STEP3　　左辺と右辺の整合性

$$\textcircled{4}\quad m\ddot{x} = -(c\dot{x}) - (kx) + u$$

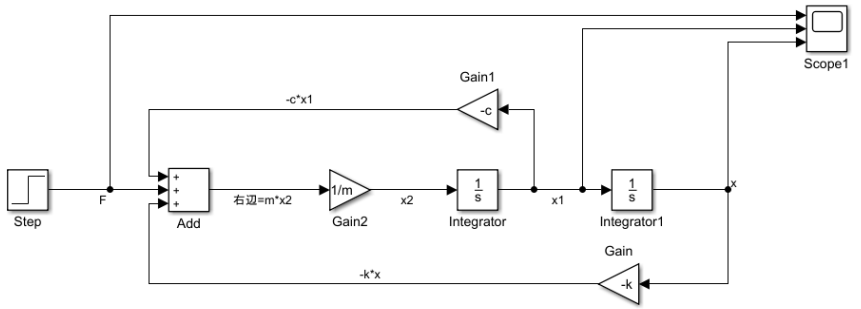


サブシステム化



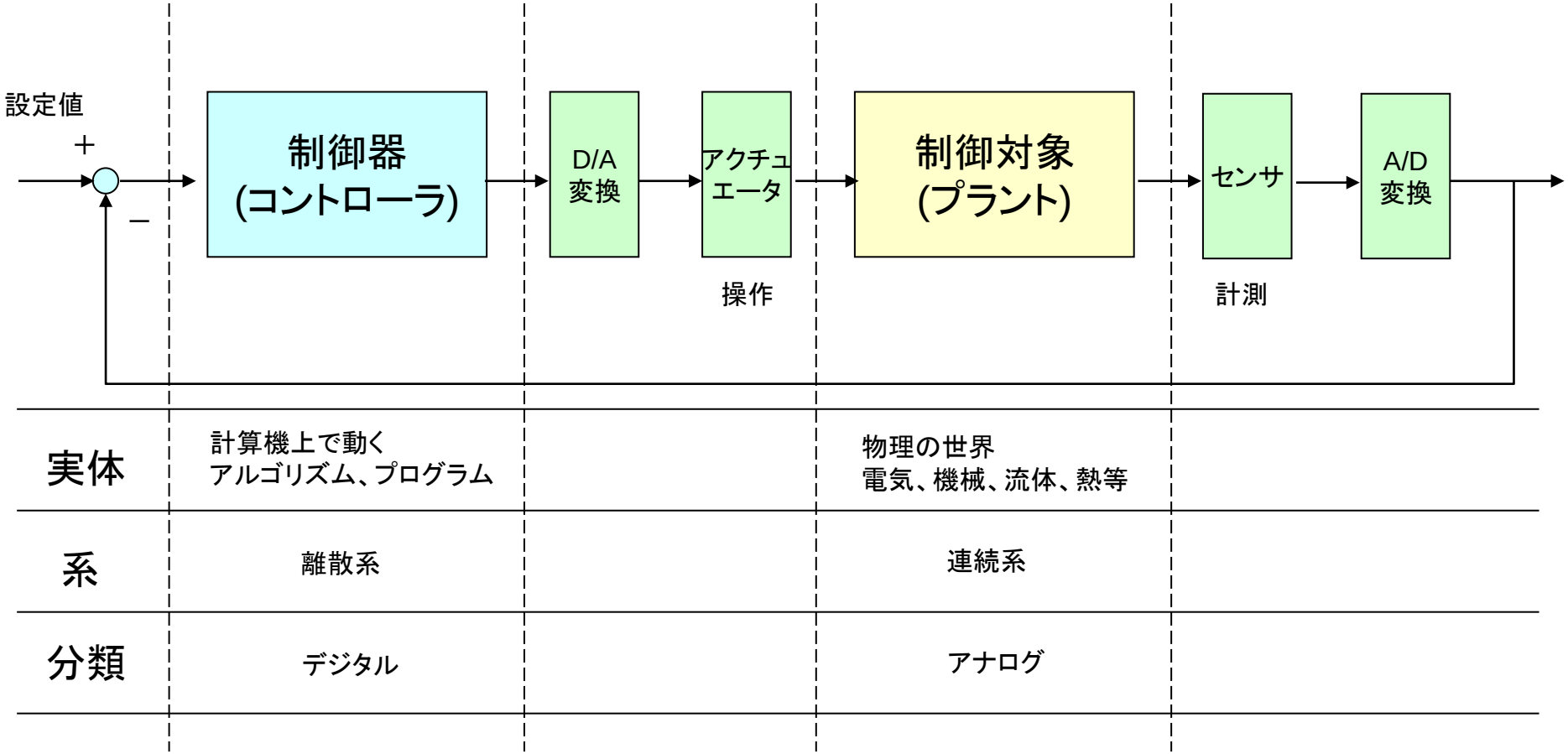
サブシステム化したい箇所を選択（マウス左クリック　範囲指定）

一自由度振動系
パラメータによる変動
演習：波形はどう変わるか？



m	c	k	特徴	波形（位置x）
1	1	10	これを基準とすると？	
1	1	100	ばね固い	
1	0	10	粘性=0	
1	1	0	バネ=0	

フィードバック制御の構造



注：設定値追従の場合

積分の精度　直感的な理解

- 計算機上の積分の精度は、計算刻みで大きく変わります。
- 理論解と、数値計算解を比較しましょう。

- 理論解
- 区間[0 π]のsin(x)の面積は？

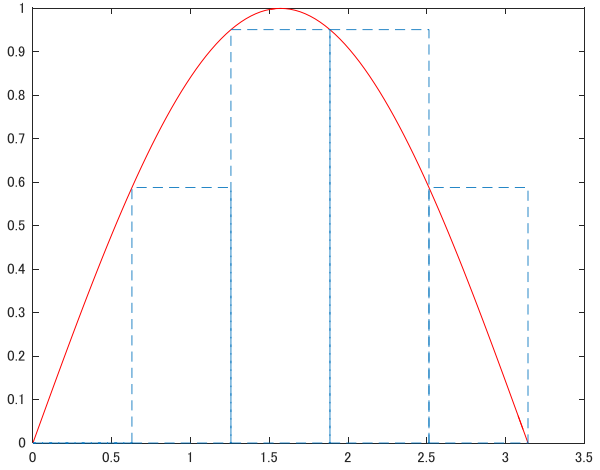
$$\int_0^{\pi} \sin(x)dx = \left[-\cos(x)\right]_0^{\pi}$$
$$= -\left(\cos(\pi) - \cos(0)\right) = 2$$

区間の分割数が多いほど
数値積分の精度は高まります。
但し、その分計算回数が増え、計算時間はかかります。

トレードオフ

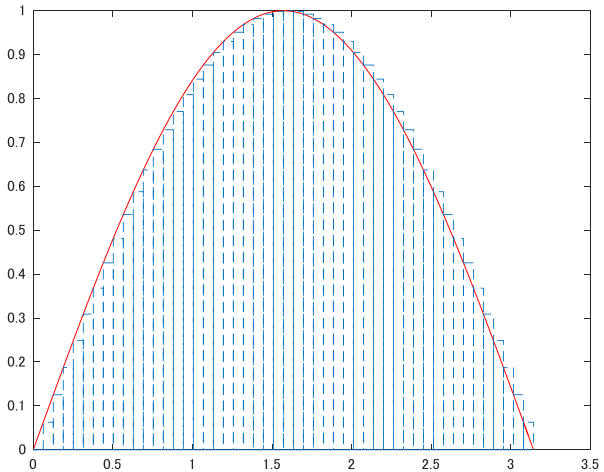
area_calc.m

区間を5分割した場合



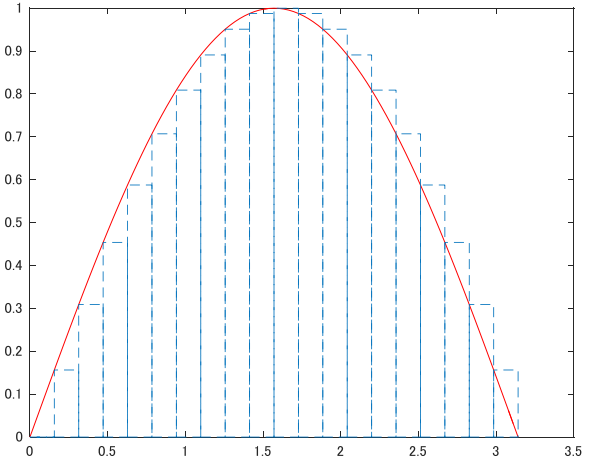
面積は1.9338　誤差3.3%

区間を50分割した場合



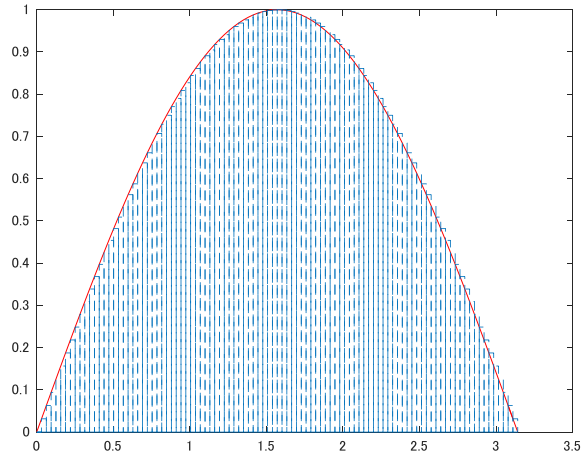
面積は1.9993　誤差0.032%

区間を20分割した場合



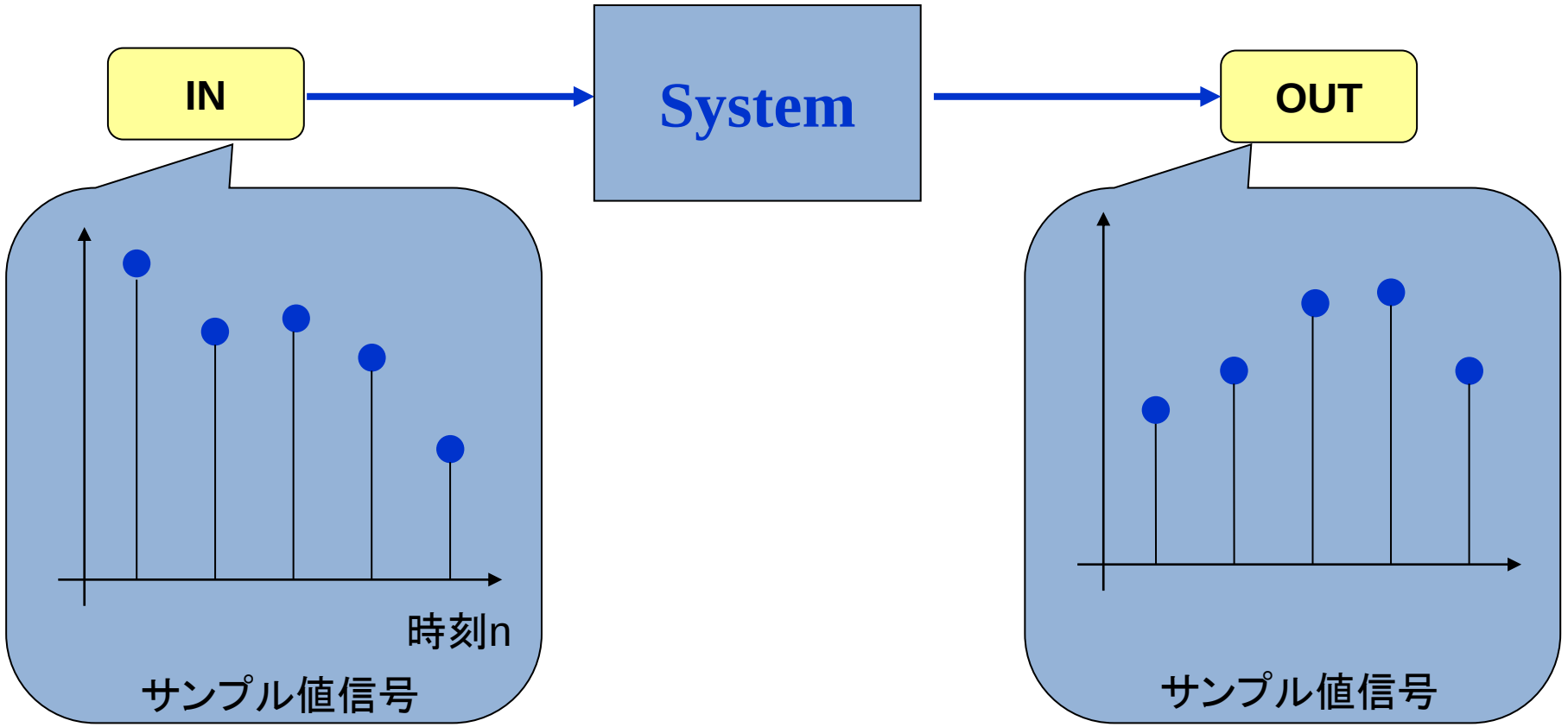
面積は1.9959 誤差0.2%

区間を100分割した場合

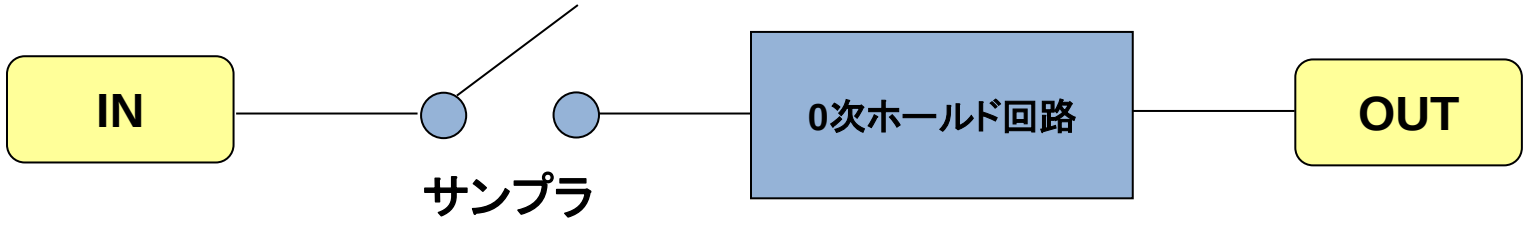


面積は1.9998　誤差0.008%

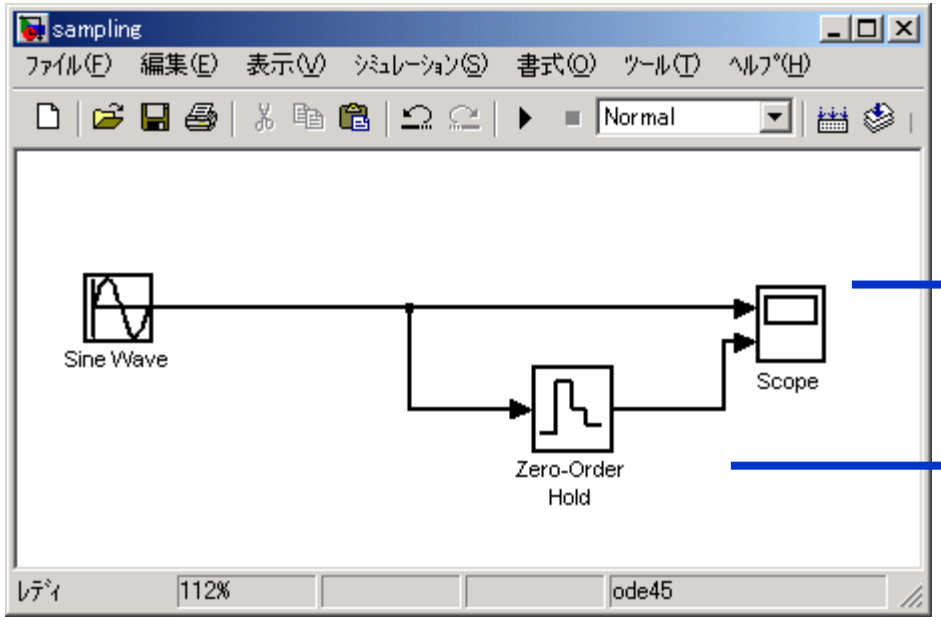
離散システム

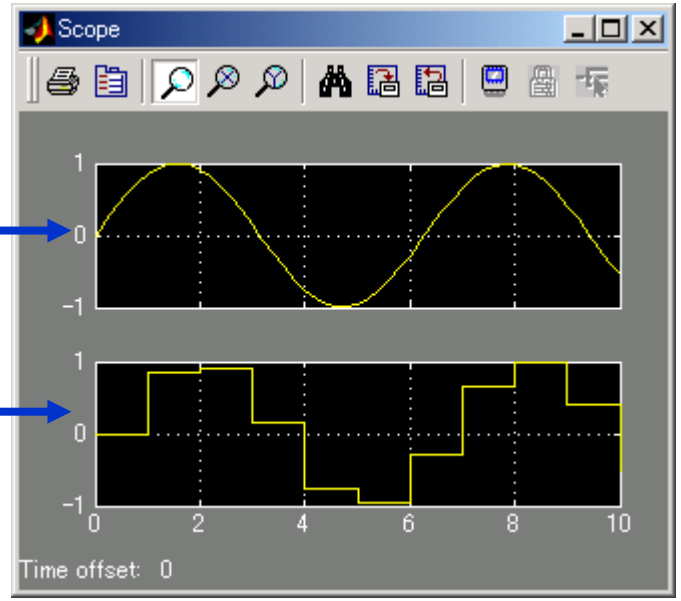


離散システム
～サンプラと0次ホールド



例





連続系と離散系の伝達関数モデリング
ローパスフィルター

- 連続系の一次遅れ伝達関数を定義します。
- >>sys=tf(1,[1 1])　　(A)
- sysをサンプリング周期0.1[秒]で離散化します。
- >>sysd=c2d(sys,0.1)　(B)
- $sysd = \frac{0.09516}{z-0.9048} = \frac{0.09516z^{-1}}{1-0.9048z^{-1}}$
- 伝達関数は入力と出力の比です。
- 離散系では,n番目の入力u[n]と出力y[n]の関係は
- $y[n] = sysd * u[n]$
- $y[n] = \frac{0.09516z^{-1}}{1-0.9048z^{-1}}u[n]$
- $(1 - 0.9048z^{-1})y[n] = (0.09516z^{-1})u[n]$
- z^{-1}は信号を1サンプル前に戻す遅延演算子です。
- $y[n] - 0.9048\,y[n-1] = 0.09516\,u[n-1]$
- $y[n] = 0.9048\,y[n-1] + 0.09516\,u[n-1] \quad (C)$

(A)

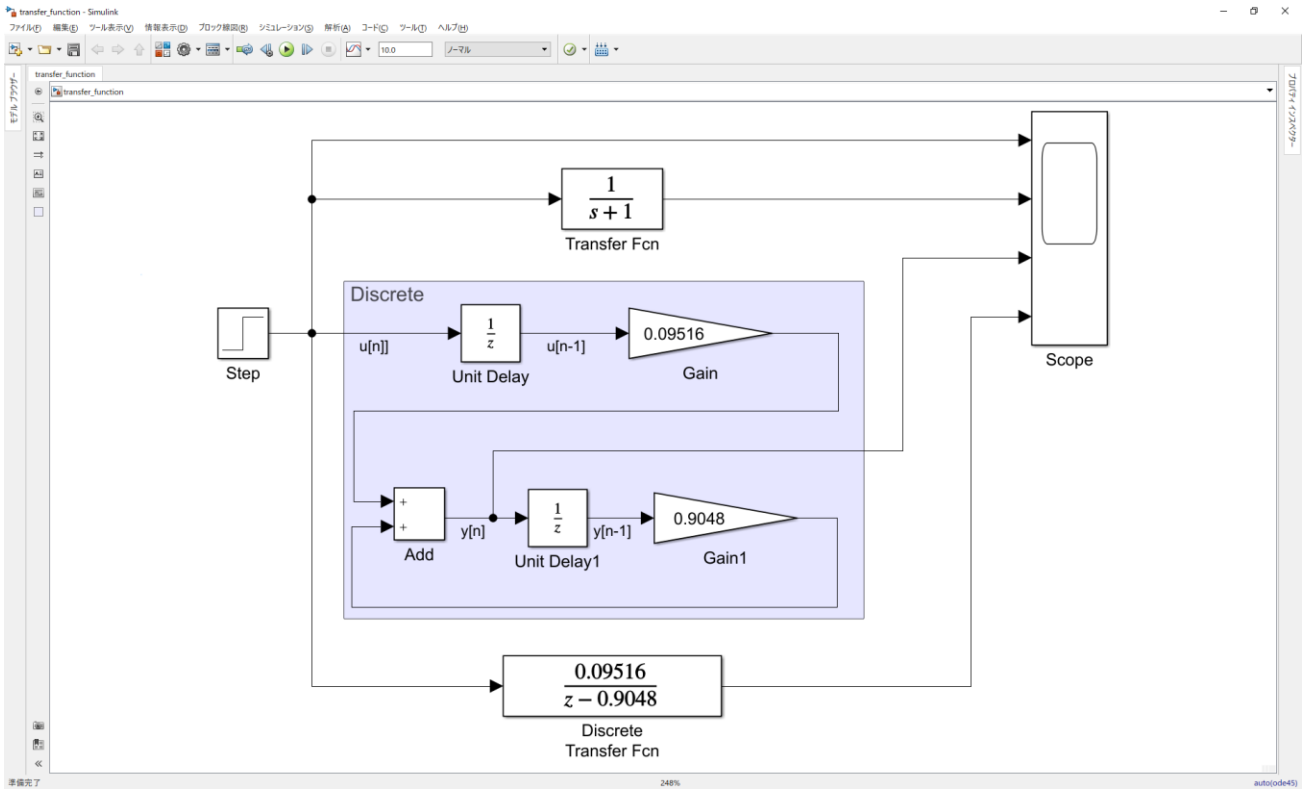
```
>> sys=tf(1,[1 1])

sys =

    1
  -----
  s + 1
```

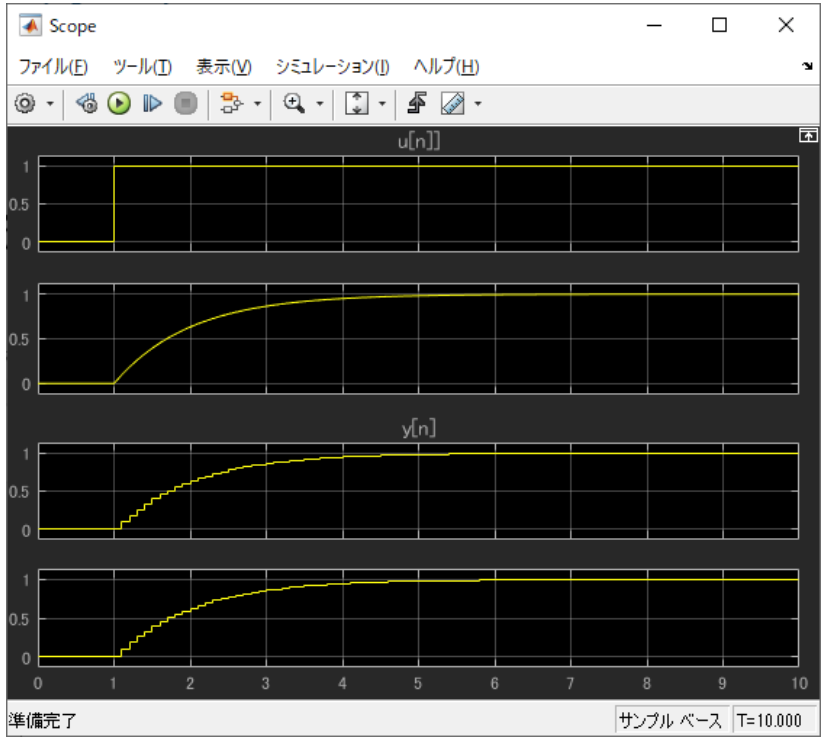
(B)　連続時間の伝達関数です。

```
sysd =

   0.09516
  ----------
  z - 0.9048

サンプル時間:  0.1 seconds
離散時間の伝達関数です。
```

Unit Delay Blockを使ったSimulinkの離散システム

- $y[n] = 0.9048\, y[n-1] + 0.09516\, u[n-1] \quad (C)$
- (C)には、y[n]の1サンプル前y[n-1],u[n]の1サンプル前u[n-1]があるので、Unit Delay Blockは2つ必要です。

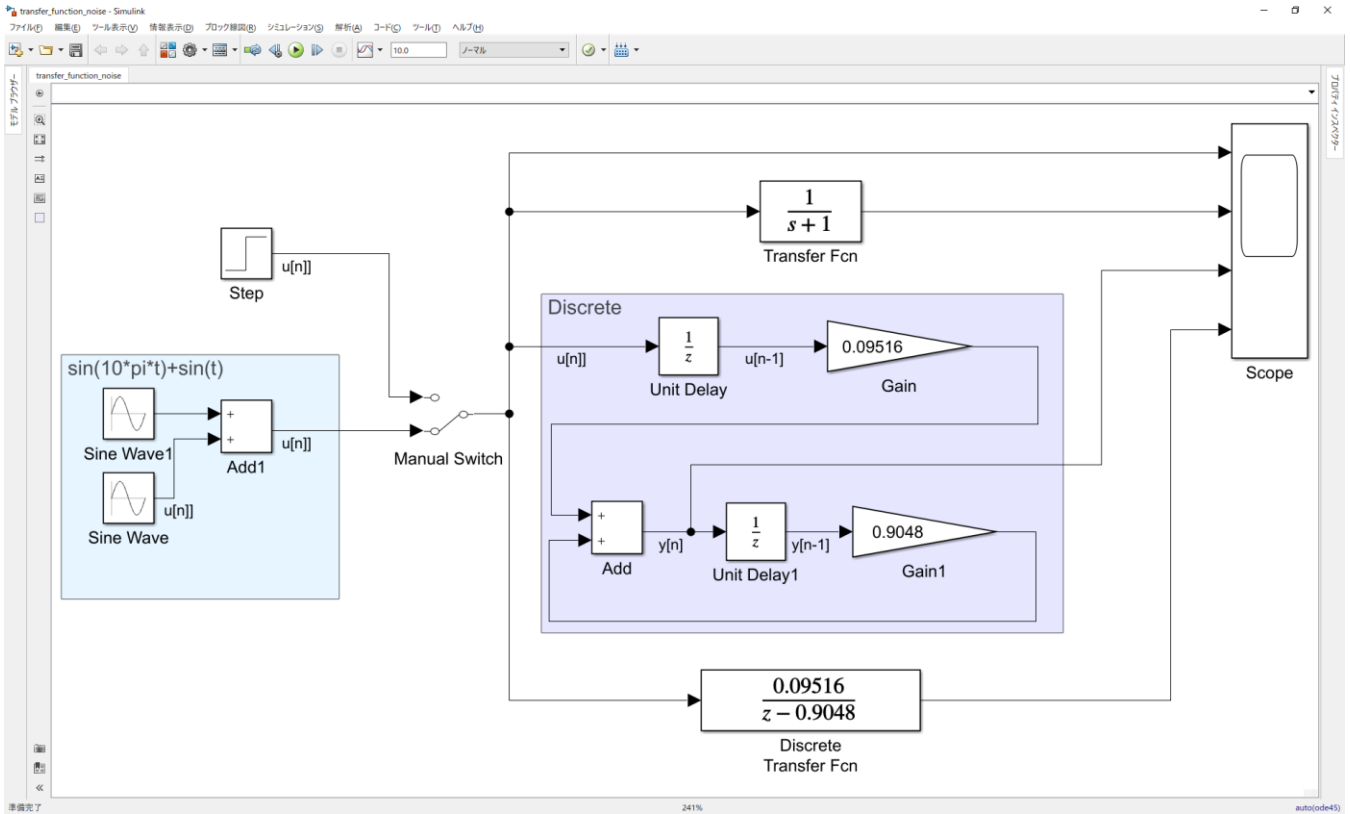


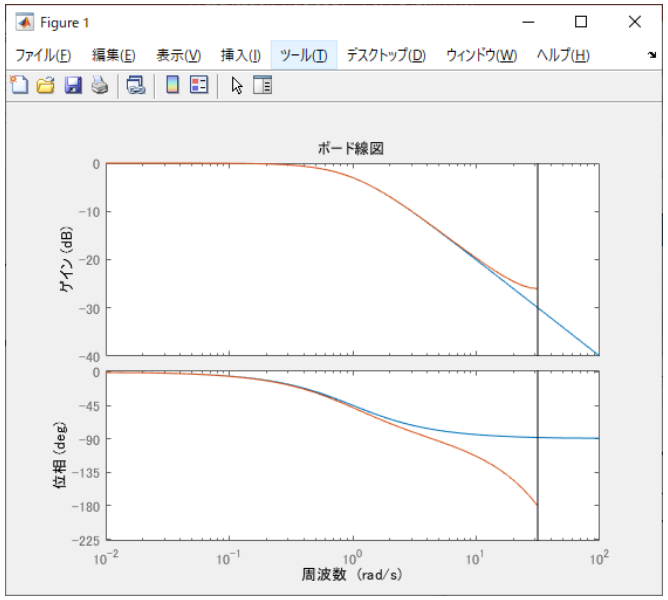


サンプル時間が十分小さければ似た波形

ローパスフィルター

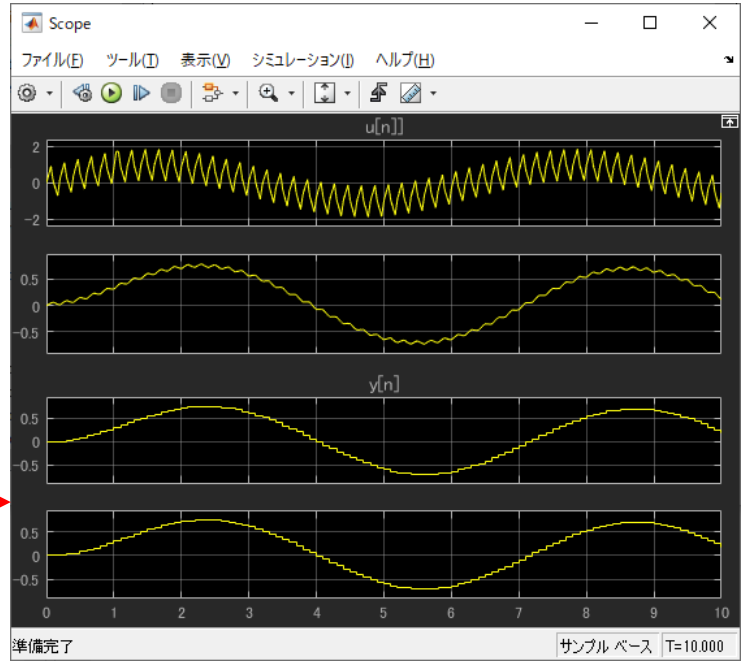
- >>figure;bode(sys,sysd)
- 高周波成分はゲインが小さくなります。
- ⇒入力信号の高周波成分はカットされます。
- シミュレーションで確かめましょう(transfer_function_noise.slx)





ノイズ含む

ノイズカット



離散系PID制御のモデリング

参照先:
トピック: 連続系のPID制御と離散系のPID制御の理解

演習
サンプルモデル(PID_start.slx)は、設定値追従制御（サーボ系）のブロック線図です。
制御対象は、伝達関数G= $\frac{1}{s+1}$ です。
上の四角内は、設定値追従制御（連続）のブロック線図です。
下の四角内は、設定値追従制御（離散）のブロック線図です。
上のPID制御は、次の式を実装したものです。パラメータKp,Ki,Kdは事前に調整済みです。

$$Y(s)=P(s)+I(s)+D(s)$$

比例項: $P(s)=Kp=10$

積分項: $I(s)=\frac{Ki}{s}=\frac{10}{s}$

微分項: $D(s)=\frac{Kd*s}{0.05s+1}=0.1*\frac{du}{dt}*\frac{1}{0.05s+1}$

離散系のPID制御は、上の式を離散化します。
サンプリング時間Ts=0.001[s]とします。

$$Y(k)=P(k)+I(k)+D(k)$$

比例項: $P(k)=Kp=10$

積分項: $I(k)=10*(I(k-1)+Ts*u(k))=10*(I(k-1)+0.01*u(k))$

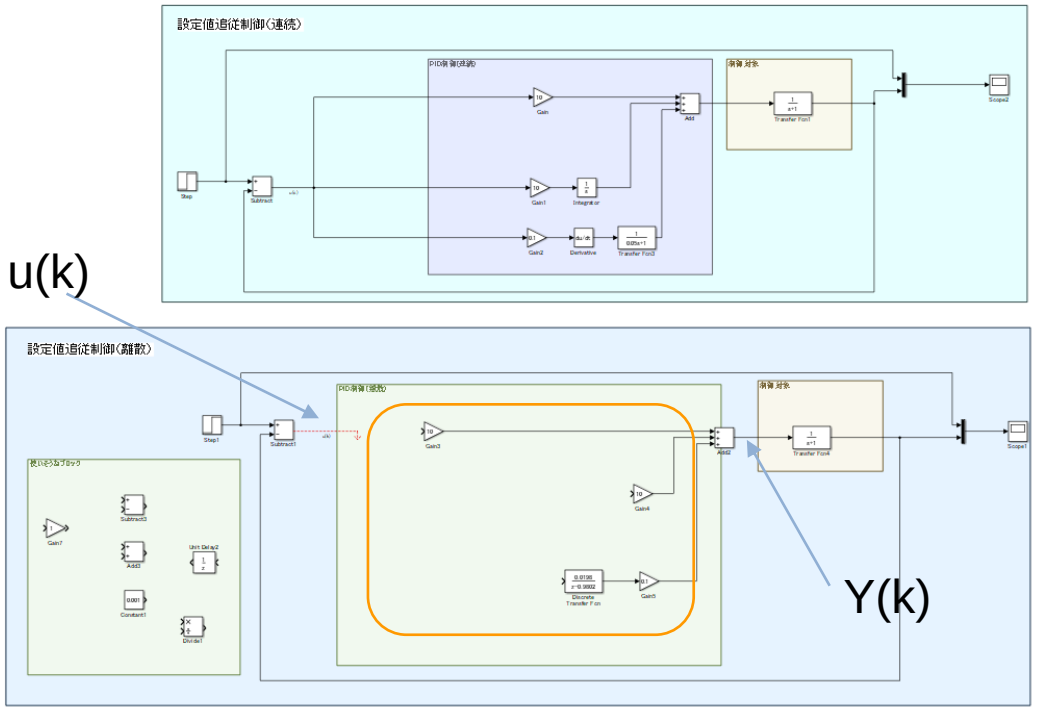
微分項: $D(k)=\frac{u(k)-u(k-1)}{Ts}*\frac{0.0198}{z-0.9802}=\frac{u(k)-u(k-1)}{0.001}*\frac{0.0198}{z-0.9802}$

練習

```
>>PID_start
```

演習1
離散系のPID制御の式をPID_startモデルに実装しましょう。

演習2
シミュレーションして、連続系の波形と離散系の波形を比較します。



解答:離散系PID制御のモデリング

参照先: 第 6 章
トピック: 連続系のPID制御と離散系のPID制御の理解

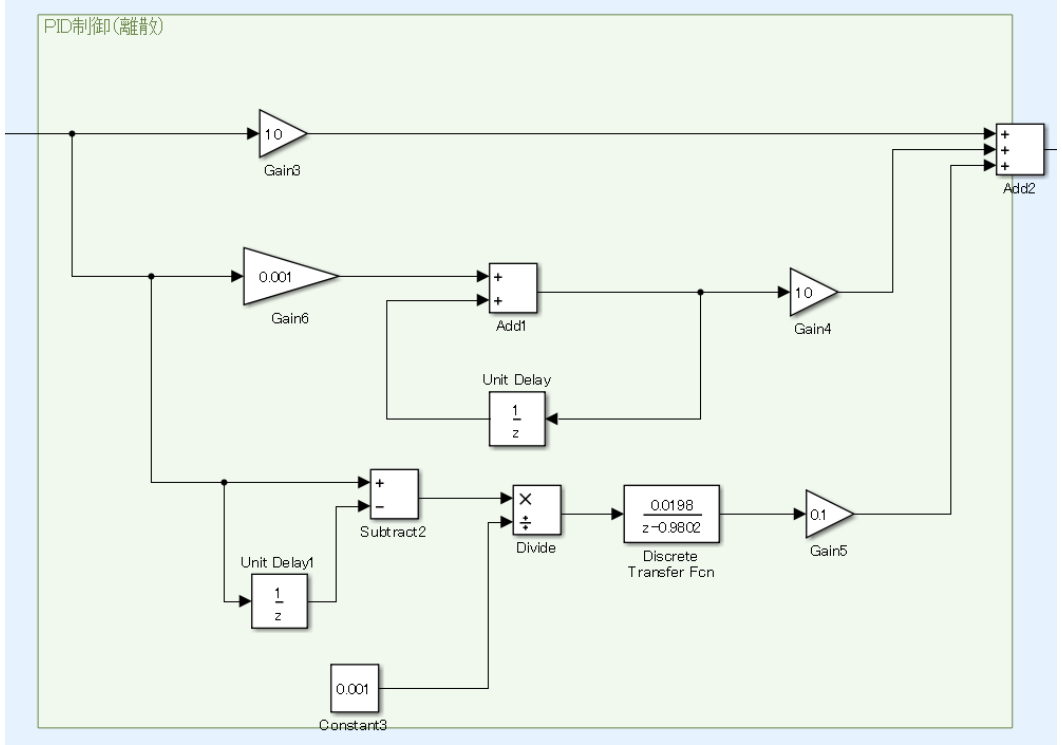
練習

```
>>PID_soln
```

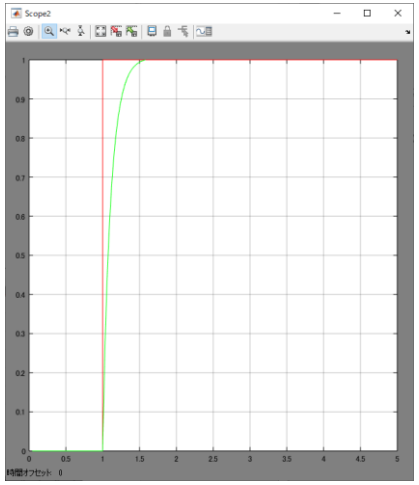
演習1
離散系のPID制御のブロック線図をPID_soln.slxに示します。

演習2
シミュレーション結果を図に示します。

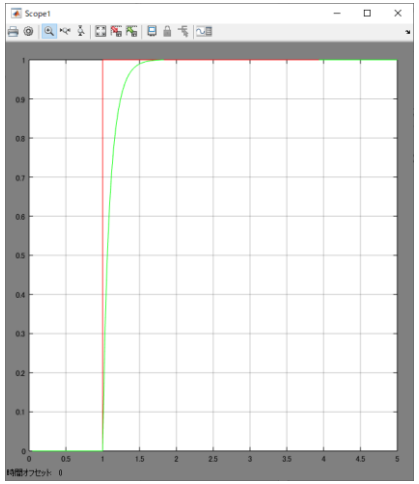
(PID_soln.slx)



連続系のPID制御 Scope



離散系のPID制御 Scope



５点移動平均フィルタ

５点移動平均フィルタ　$$y_n = \frac{1}{5}\left(x_n + x_{n-1} + x_{n-2} + x_{n-3} + x_{n-4}\right)$$

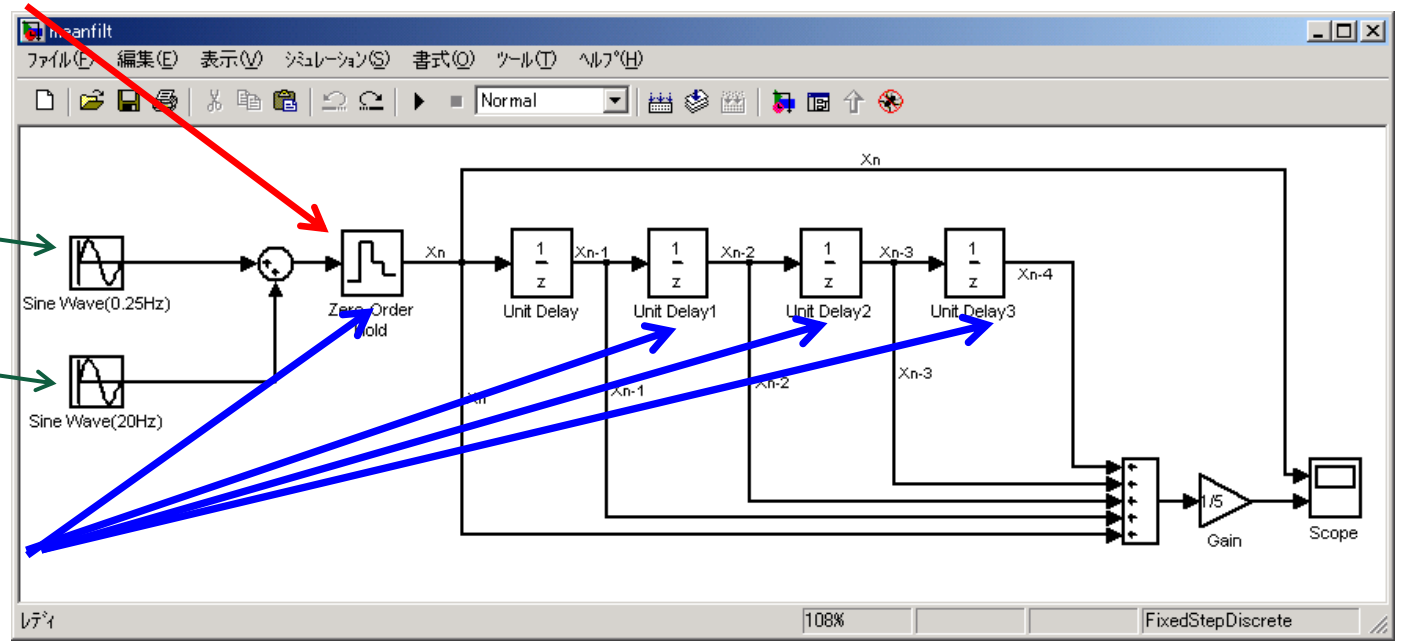
遅延演算子を 1/Z とすると　$$y_n = \frac{1}{5}\left(x_n + x_n z^{-1} + x_n z^{-2} + x_n z^{-3} + x_n z^{-4}\right)$$

サンプル時間
0.01[s]

振幅10
周波数 0.25*2*pi

振幅1
周波数 20*2*pi

サンプル時間
-1[s]　%継承



(meanfilt.mdl)

便利な機能　Tips

Data Type Conversion
数値型変換(キャスト)

サンプル時間の表示

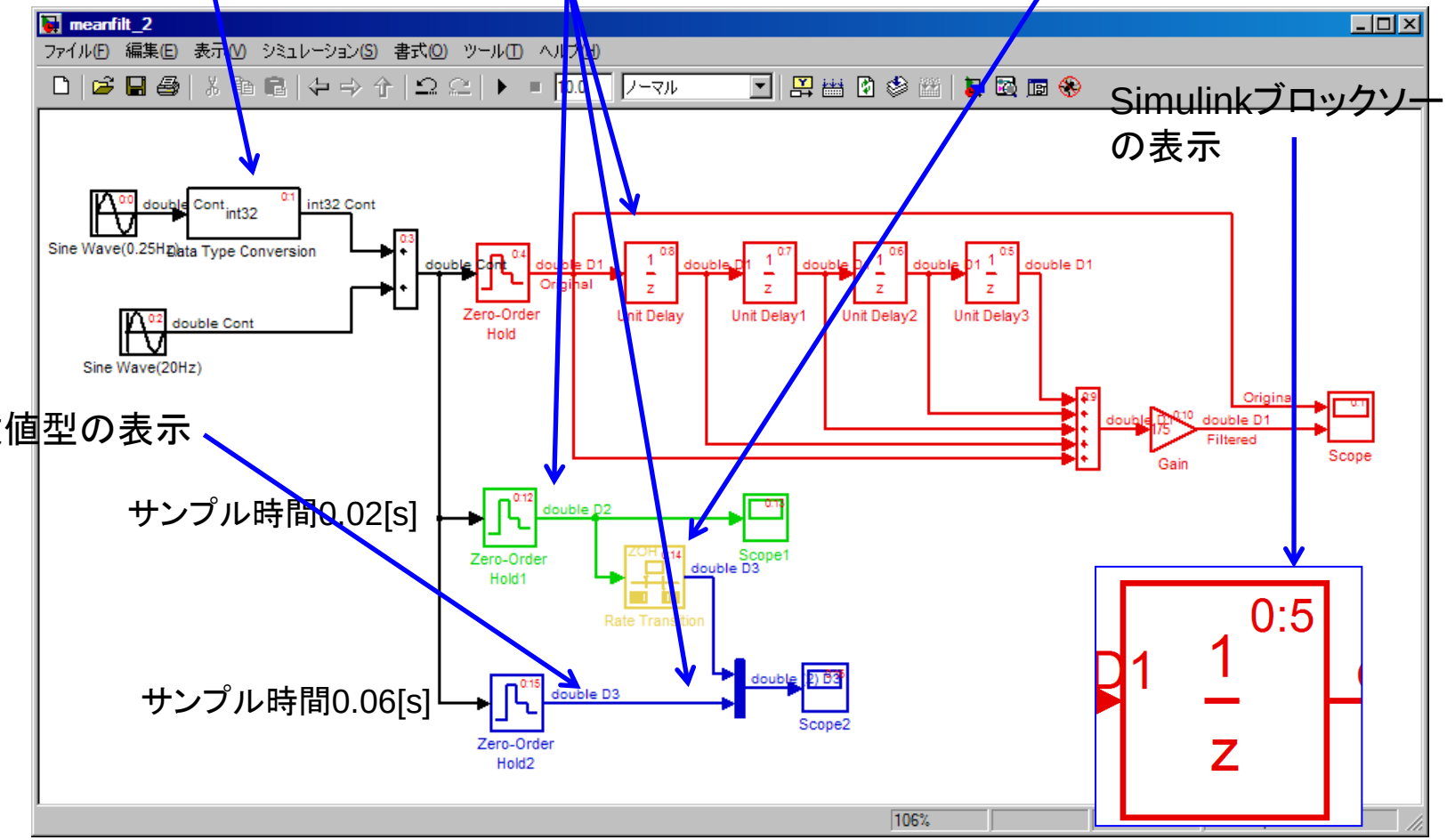
Rate Transition
サンプル時間の変換

Simulinkブロックソート(計算)順序の表示

数値型の表示

サンプル時間0.02[s]

サンプル時間0.06[s]



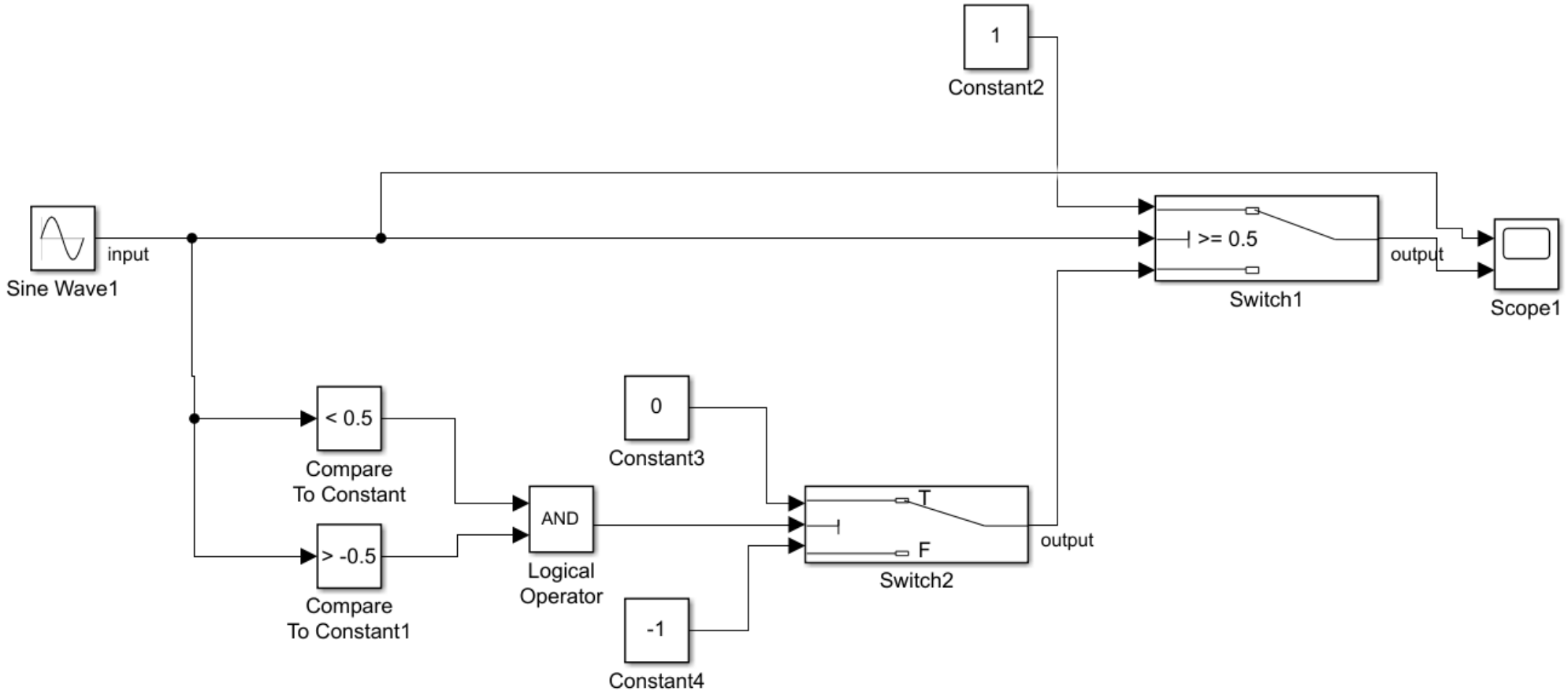
(meanfilt_2.mdl)

Switch Blockで作成してみましょう。

```
if (input  >= 0.5) %条件A
        out put = 1;
        else if ( (input < 0.5) & (input > -0.5) ) %条件B
        output = 0;

        else  %条件C
        output = -1;
end
```

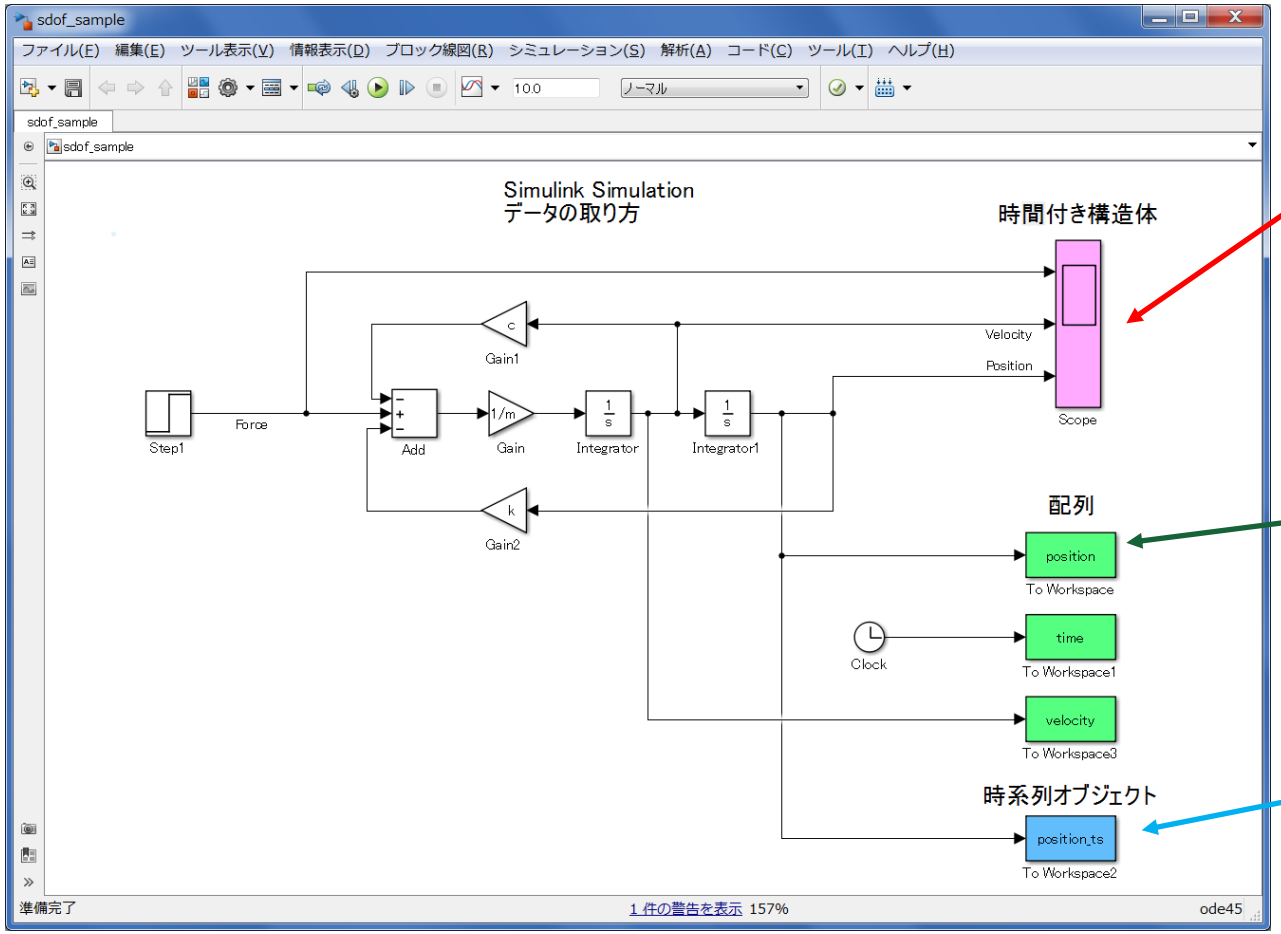
ヒント：Switch blockは2個使います。

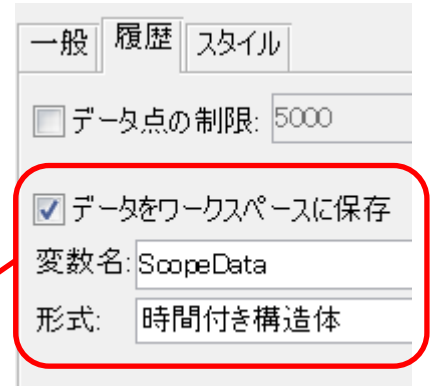
回答例

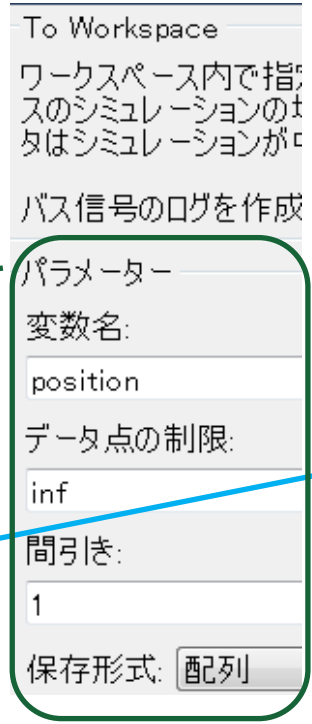

1
Constant2
input
Sine Wave1
>= 0.5
output
Switch1
Scope1
< 0.5
Compare
To Constant
0
Constant3
AND
Logical
Operator
T
F
output
Switch2
> -0.5
Compare
To Constant1
-1
Constant4

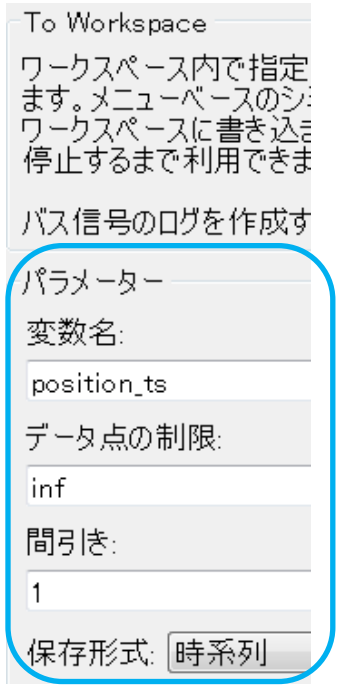
Simulinkのシミュレーション結果を、(1)MATLABで利用、(2)グラフに描画する

- モデル(sdof_sample.slx)







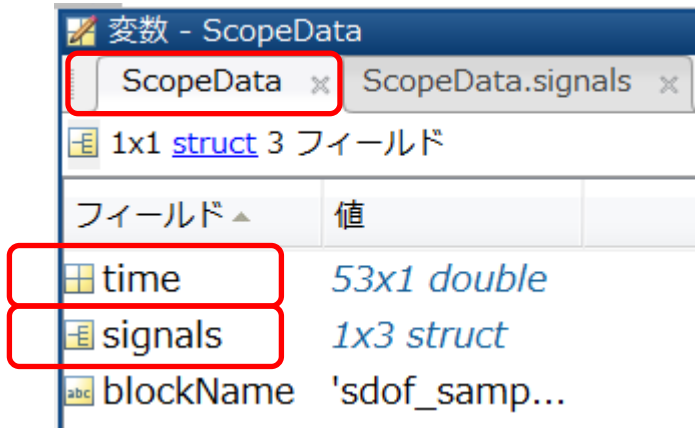


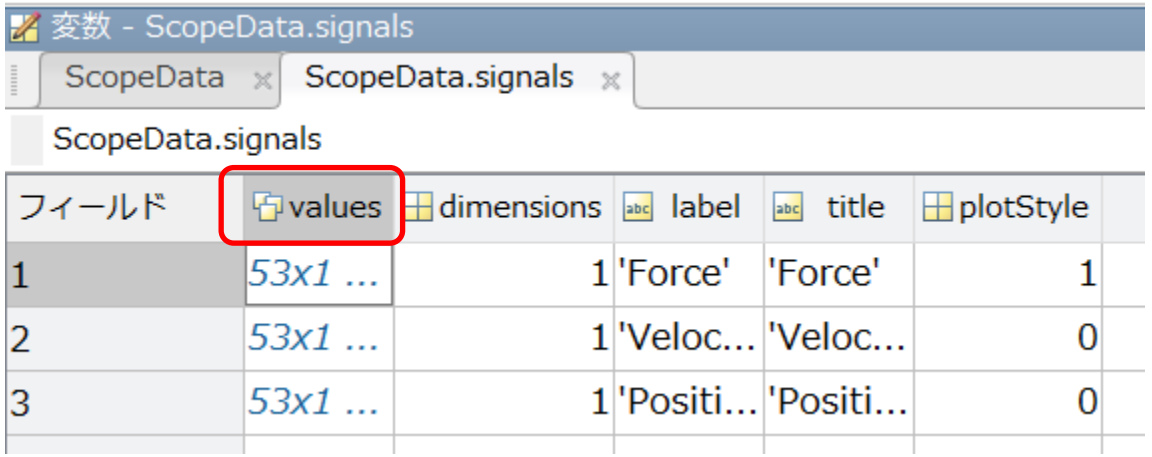
目的

- Simulinkのシミュレーション結果を、(1)MATLABで利用、(2)グラフに描画する等の作業は頻繁にあります。
- また、MATLABのグラフを見栄えよくしたいことは多々あります。(発表、論文紙面、技報等)
- この資料はその参考です。

【時間付き構造体】

- 参照方法
- 構造体名.メンバ名
- 時間データ:　構造体名.time
- 数値データ:　構造体名.signals.values





配列

- 数値のベクトルデータ
- 通常のMATLABのワークスペース変数

- 注意：行ベクトルか列ベクトルか？
- size関数で調べる。
- >> size(time)

- ans =

- 53　1(53行1列の意味)

変数 - position

position

53x1 double

	1
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	1.0114e-28
9	0.0181
10	0.0617
11	0.1108
12	0.1472
13	0.1604
14	0.1501
15	0.1241
16	0.0943
17	0.0720

変数 - time

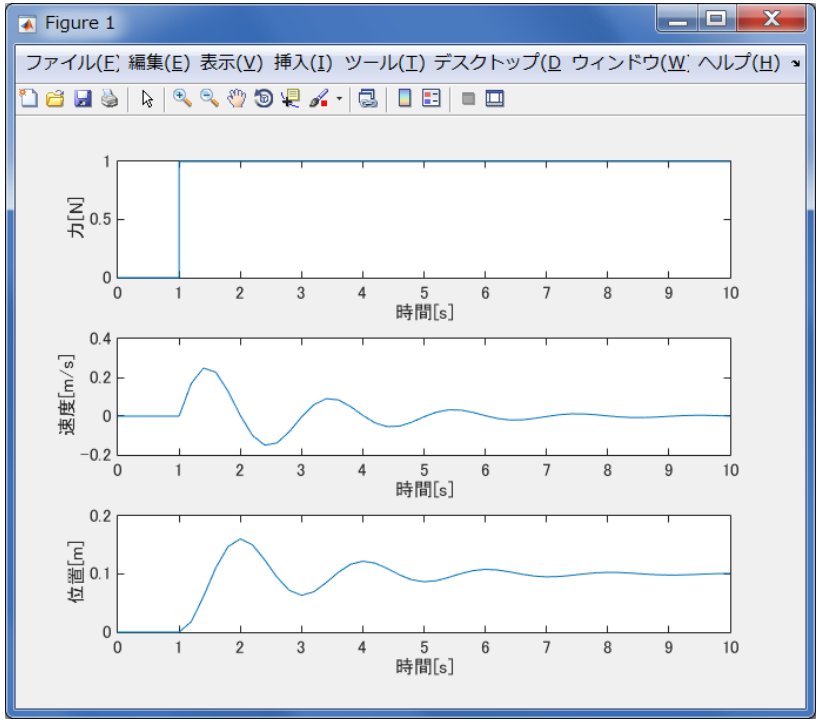
time

53x1 double

	1
1	0
2	0.2000
3	0.4000
4	0.6000
5	0.8000
6	1.0000
7	1
8	1.0000
9	1.2000
10	1.4000
11	1.6000
12	1.8000
13	2.0000
14	2.2000
15	2.4000
16	2.6000
17	2.8000

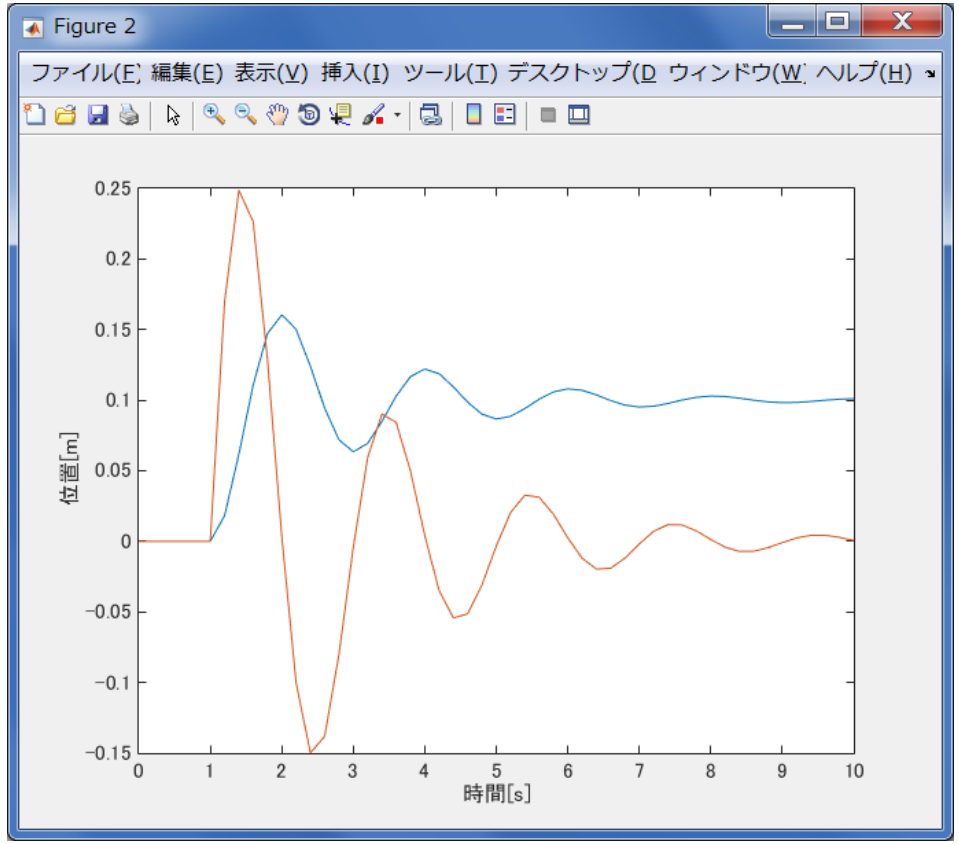
グラフに描画
構造体データ

```
% sample_plot.m
%% 構造体データ
%構造体のデータをワークスペース変数に変換
Time_sc = ScopeData.time;
U_sc = ScopeData.signals(1).values;
V_sc = ScopeData.signals(2).values;
P_sc = ScopeData.signals(3).values;

figure(1) %figureの枠を用意
subplot(311)%グラフ3行の1行目
plot(Time_sc,U_sc)%グラフ描画
% 注：各plot毎に加工
xlabel('時間[s]')
ylabel('力[N]')

subplot(312)%グラフ3行の2行目
plot(Time_sc,V_sc)%グラフ描画
% 注：各plot毎に加工
xlabel('時間[s]')
ylabel('速度[m/s]')

subplot(313)%グラフ3行の3行目
plot(Time_sc,P_sc)%グラフ描画
% 注：各plot毎に加工
xlabel('時間[s]')
ylabel('位置[m]')
```



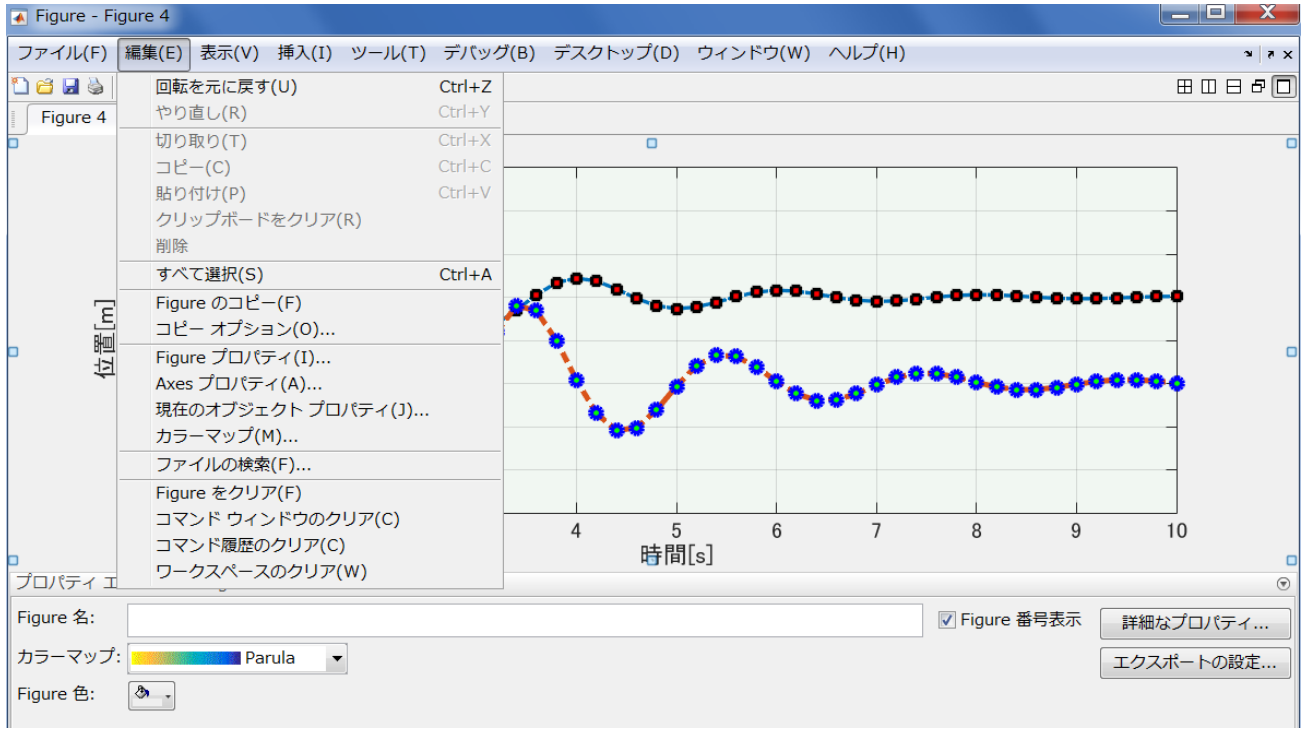
グラフに描画
配列

```
%% 配列データ
figure(2) %figureの枠を用意
plot(time,position)%グラフ描画 x:time,y:position
hold on%上書きモード開始
plot(time,velocity)%グラフ描画 x:time,y:velocity
hold off%上書きモード終了
xlabel('時間[s]')
ylabel('位置[m]')
```

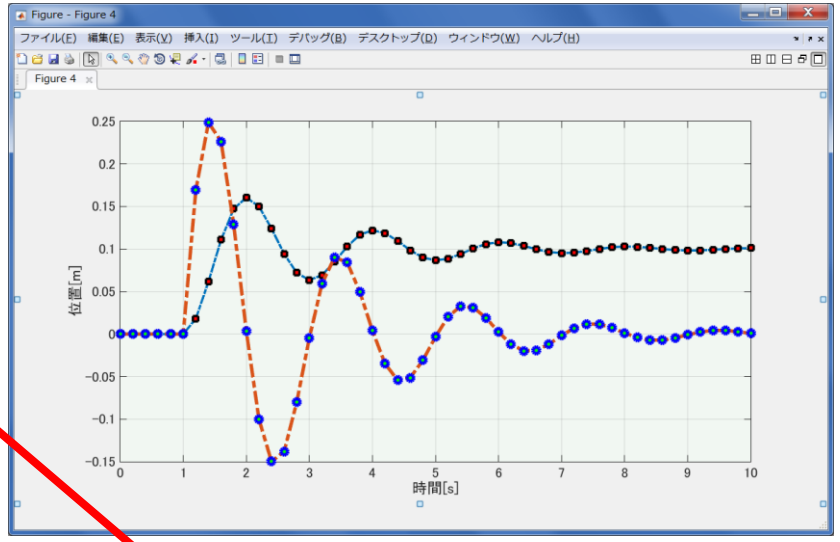


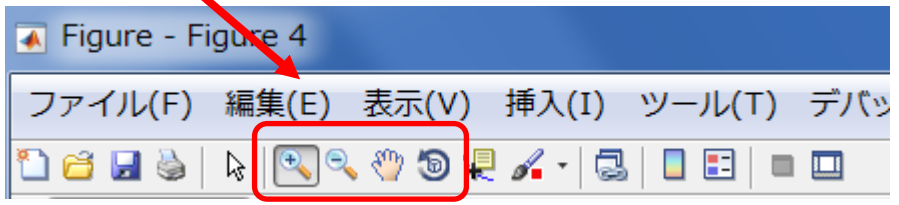
GUIから変更

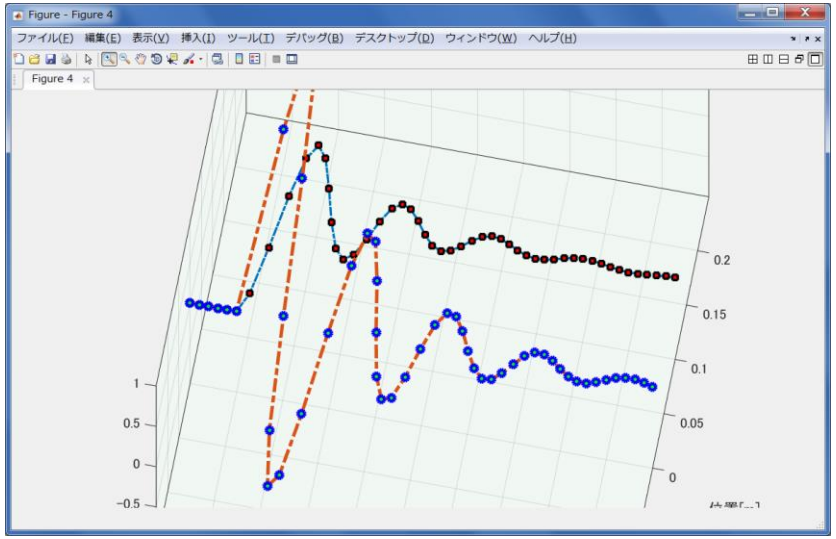
Figureから直接変更



拡大、縮小、移動、回転







編集→Figureプロパティ→当該箇所をマウスで指示
(線、背景、フォント等)