

計測制御工学

担当教員: 河合 康典

開講期: 前期 単位数: 2単位

区分: 必修

教科書: 川田昌克「MATLAB/Simulink による現代制御入門」
(森北出版)

成績評価: 中間試験(40%), 期末試験(40%), レポート(20%)

Office Hour: 水曜日 14:40 - 16:10

電気・環境都市工学科棟3F 河合教員室

URL: <https://nobushi-ykawaiab.ssl-lolipop.jp/ykawaiab/>

E-mail: y_kawai@ishikawa-nct.ac.jp

第 1 章 古典制御理論から現代制御理論へ

1.1 高次システムに対する古典制御理論の限界

1.2 現代制御理論における高次システムの取り扱い

キーワード：SISO, MIMO, 状態方程式, 出力方程式

学習目標：古典制御理論で困難な多入力多出力システムを容易にする現代制御理論の重要性を理解する。

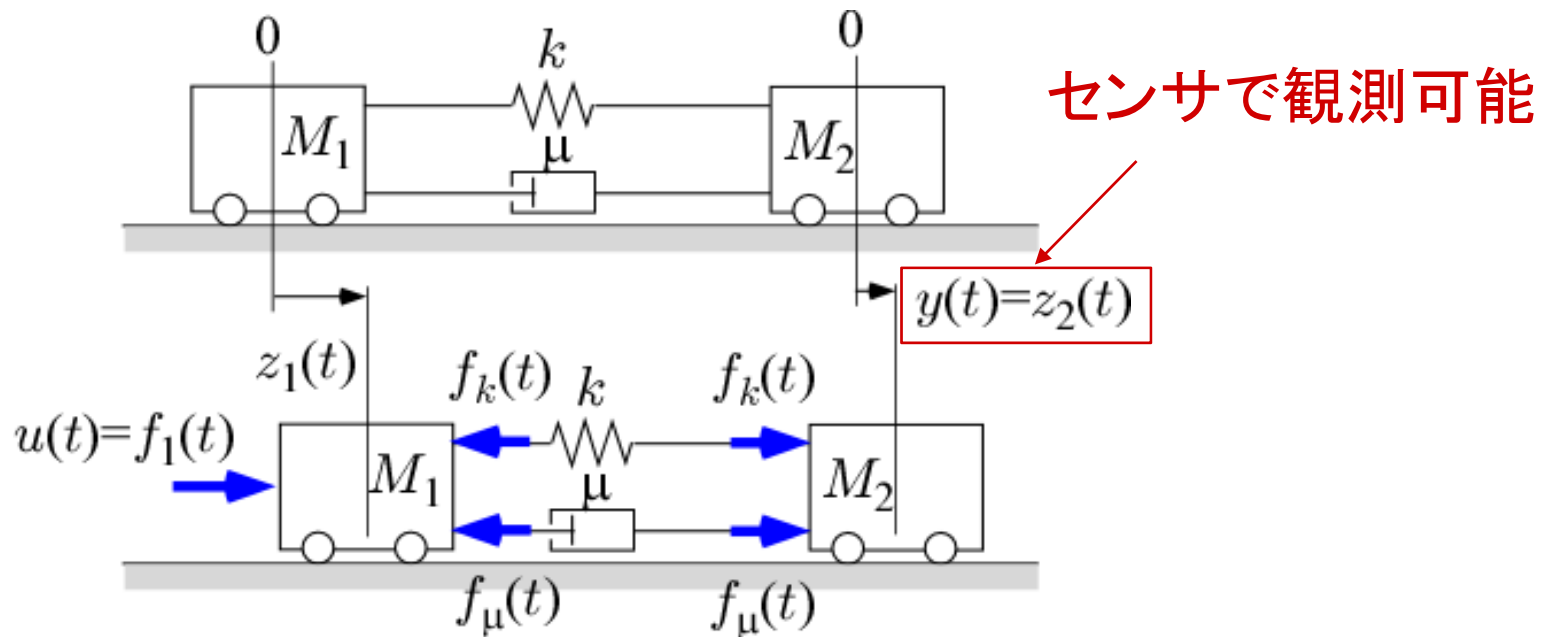
1 古典制御理論から現代制御理論へ

1.1 高次システムに対する古典制御理論の限界

[例 1.1]

台車2の情報のみ利用

$$\begin{cases} M_1 \ddot{z}_1(t) = u(t) - \frac{k(z_1(t) - y(t))}{f_k(t)} - \frac{\mu(\dot{z}_1(t) - \dot{y}(t))}{f_\mu(t)} & (1.1a) \\ M_2 \ddot{y}(t) = \frac{k(z_1(t) - y(t))}{f_k(t)} + \frac{\mu(\dot{z}_1(t) - \dot{y}(t))}{f_\mu(t)} & (1.1b) \end{cases}$$



$$M_2 \ddot{y}(t) = k(z_1(t) - y(t)) + \mu(\dot{z}_1(t) - \dot{y}(t)) \quad (1.1b)$$

(1.1b) 式をラプラス変換

$$M_2 y(s) s^2 = k(z_1(s) - y(s)) + \mu(z_1(s) - y(s)) s$$

$$(M_2 s^2 + \mu s + k) y(s) = (\mu s + k) z_1(s)$$

$$z_1(s) = \frac{M_2 s^2 + \mu s + k}{\mu s + k} y(s) \quad (1.1c)$$

$$M_1 \ddot{z}_1(t) = u(t) - k(z_1(t) - y(t)) - \mu(\dot{z}_1(t) - \dot{y}(t)) \quad (1.1a)$$

(1.1a) 式をラプラス変換

$$M_1 z_1(s) s^2 = u(s) - k(z_1(s) - y(s)) - \mu(z_1(s) - y(s)) s$$

$$(M_1 s^2 + \mu s + k) z_1(s) = u(s) + (\mu s + k) y(s)$$

$$z_1(s) = \frac{M_2 s^2 + \mu s + k}{\mu s + k} y(s) \quad (1.1c)$$

$$(M_1 s^2 + \mu s + k) z_1(s) = u(s) + (\mu s + k) y(s)$$

(1.1c) 式を代入

$$\frac{(M_1 s^2 + \mu s + k) (M_2 s^2 + \mu s + k)}{\mu s + k} y(s) = u(s) + (\mu s + k) y(s)$$

$$\frac{s^2 (M_1 M_2 s^2 + M_1 (\mu s + k) + M_2 (\mu s + k))}{\mu s + k} y(s) = u(s)$$

よって

$$P(s) = \frac{y(s)}{u(s)} = \frac{\mu s + k}{s^2 (M_1 M_2 s^2 + M_1 (\mu s + k) + M_2 (\mu s + k))}$$

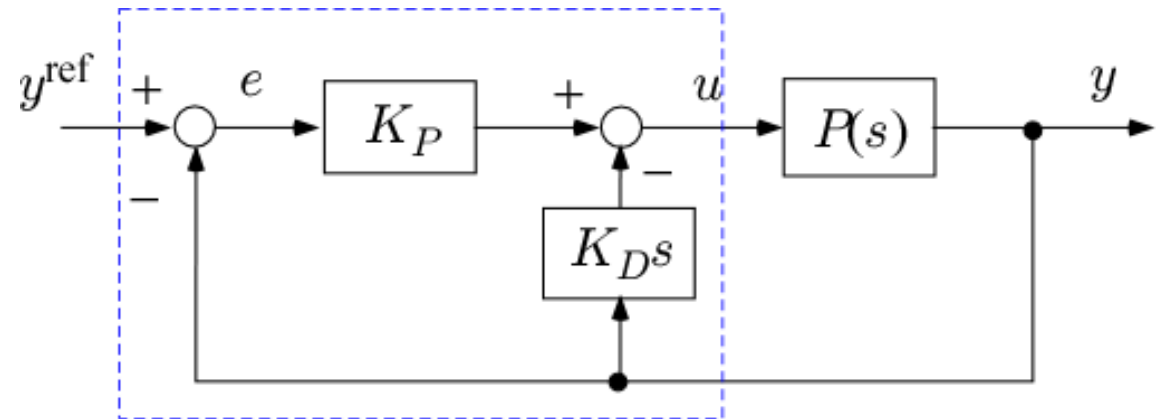
値を代入すると

$$P(s) = \frac{s + 2}{s^2 (0.5s^2 + 0.5(s + 2) + (s + 2))} = \frac{2s + 4}{s^2 (s^2 + 3s + 6)}$$

P-D コントローラ

$$u(t) = k_P e(t) - k_D \dot{y}(t)$$

$$e(t) = y^{\text{ref}}(t) - y(t)$$



[MATLAB/Scilab演習]

[例 1.2] コントローラのゲイン(北森の方法)

$$k_P = \frac{3\omega_n^2}{2}, \quad k_D = 3\zeta\omega_n, \quad \zeta = 0.7, \quad \omega_n = 0.5, \quad 2$$

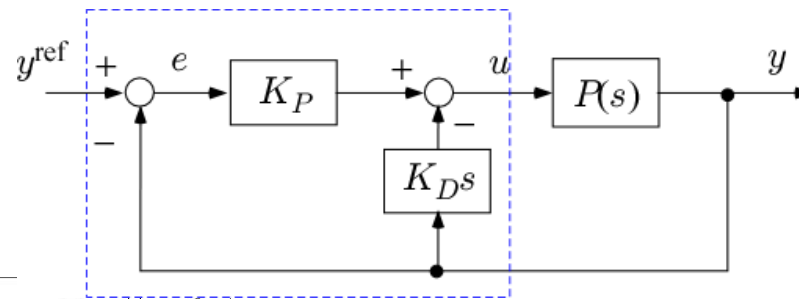
目標値

$$y^{\text{ref}}(t) = \begin{cases} 0 & (t < 0) \\ 1 & (t \geq 0) \end{cases}$$

[MATLAB演習]

example1_2_tf.mdl

$$\omega_n = 0.5, \zeta = 0.7$$

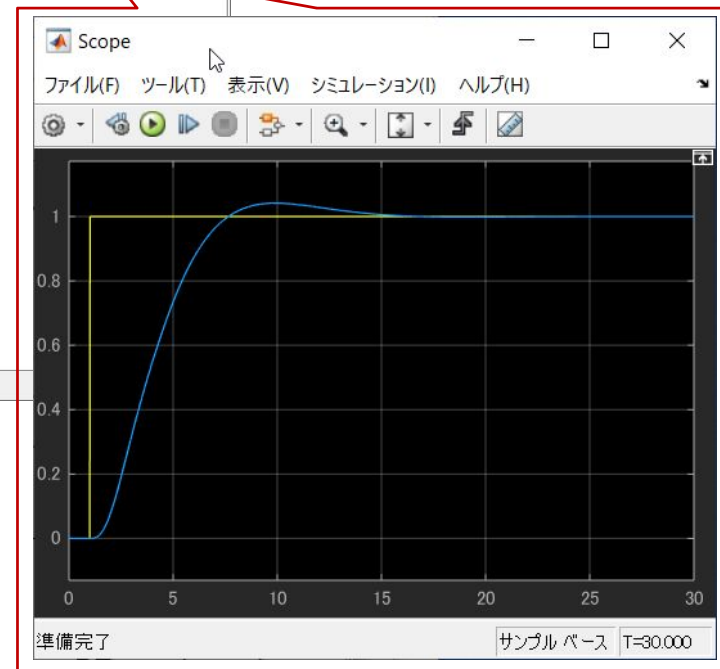
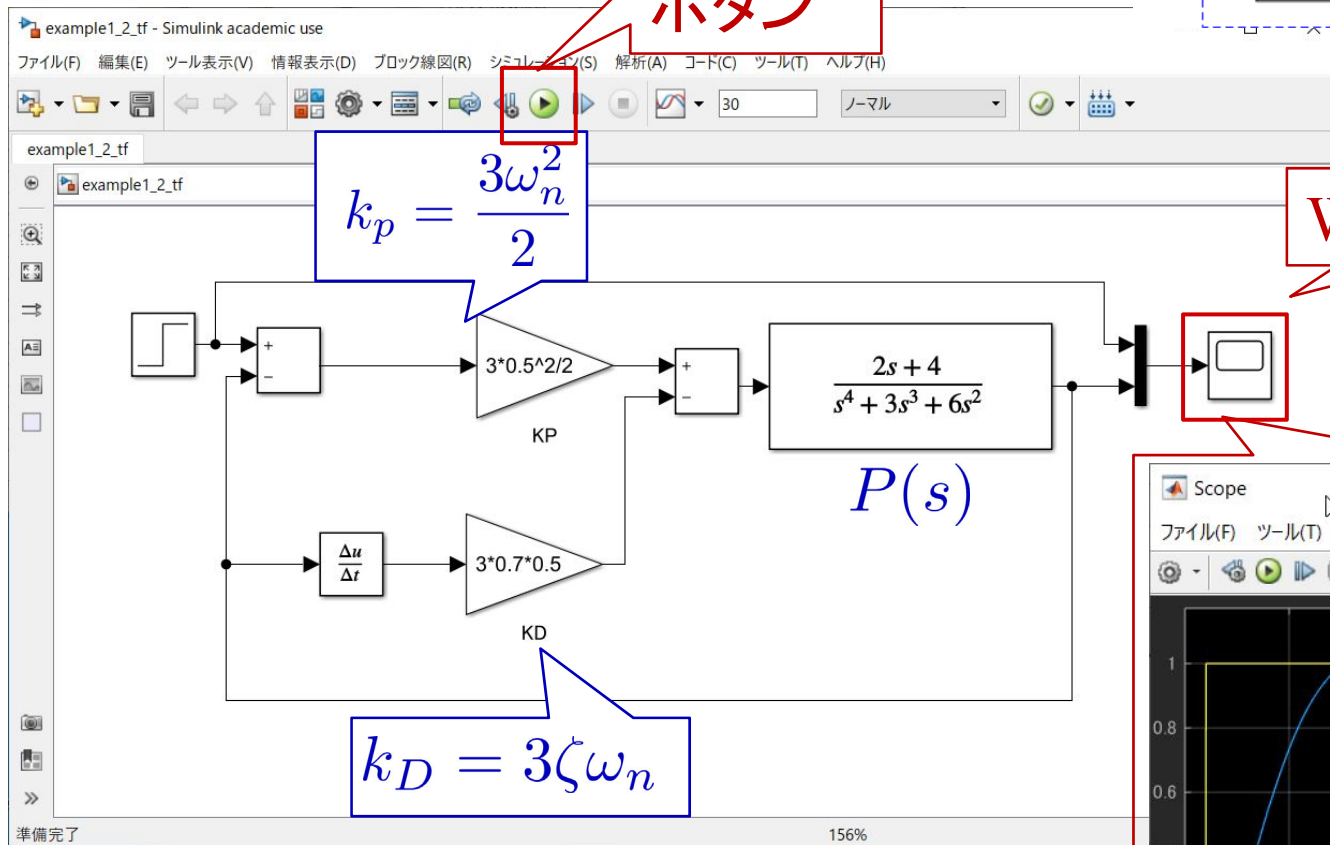


スタート
ボタン

Wクリック

$$k_p = \frac{3\omega_n^2}{2}$$

$$k_D = 3\zeta\omega_n$$



[MATLAB演習]

example1_2_tf.mdl

$$\omega_n = 2, \zeta = 0.7$$

Wクリック

Wクリックして
 $3*0.7*2$
に変更する

ブロックパラメーター: KP

Gain
要素単位のゲイン ($y = K*u$)、または行列ゲイン ($y = K*u$ 、または $y = u*K$)

メイン 信号属性 パラメーター属性

ゲイン:
 $3*0.5^2/2$

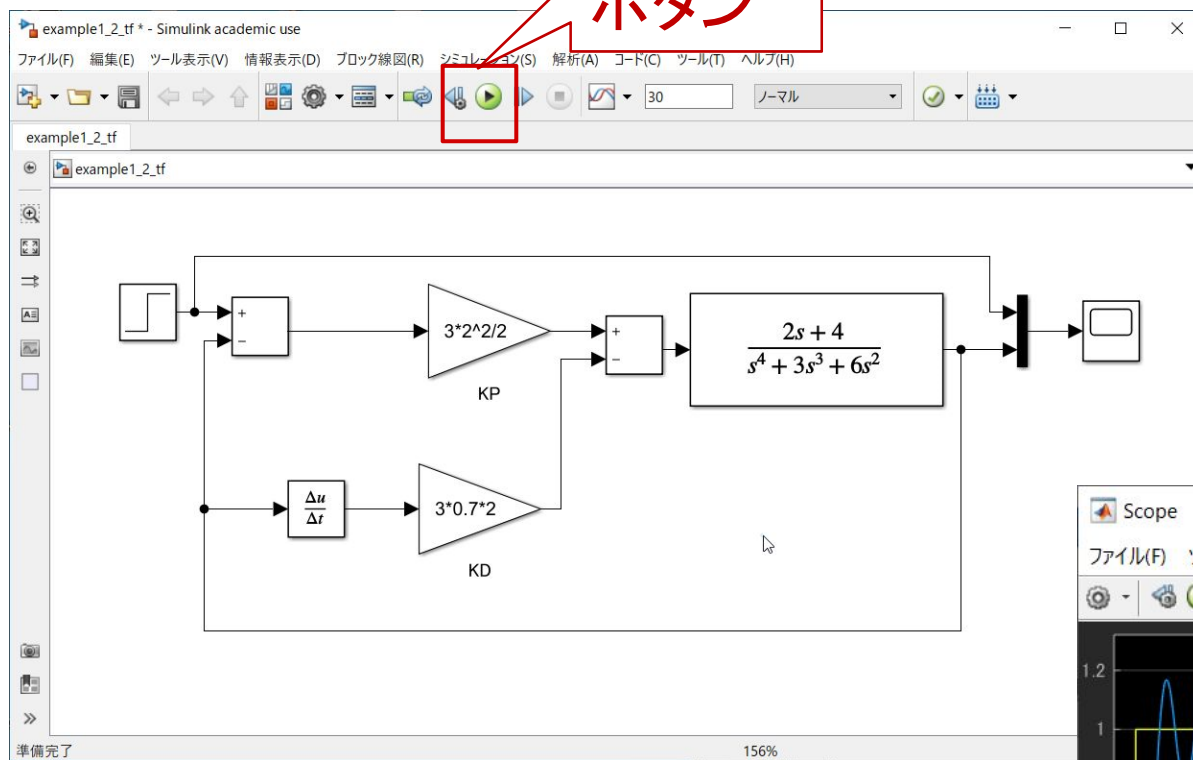
乗算: 要素単元 ($K*u$)

OK(O) キャンセル(C) ヘルプ(H) 適用(A)

準備完了 156% ode4

[MATLAB演習]

スタート
ボタン

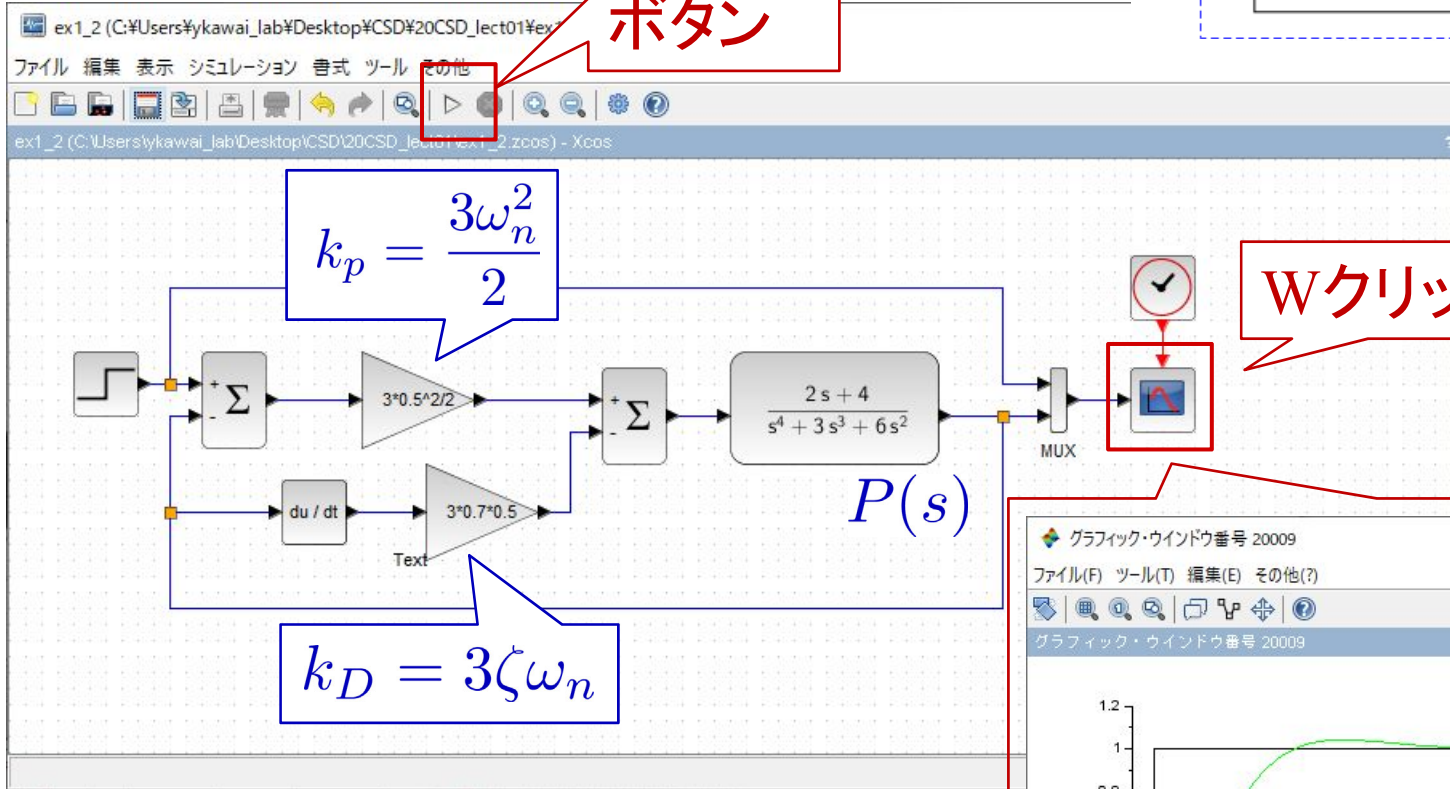
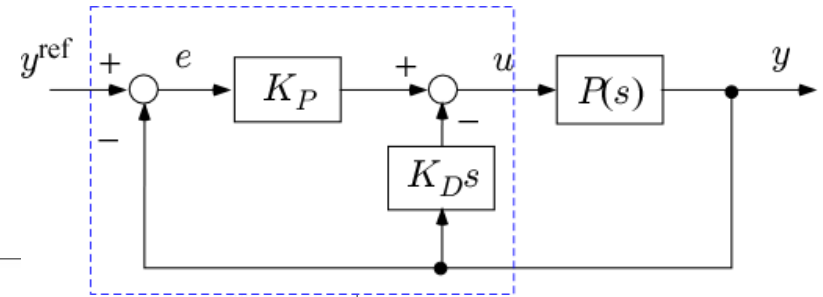


[Scilab演習]

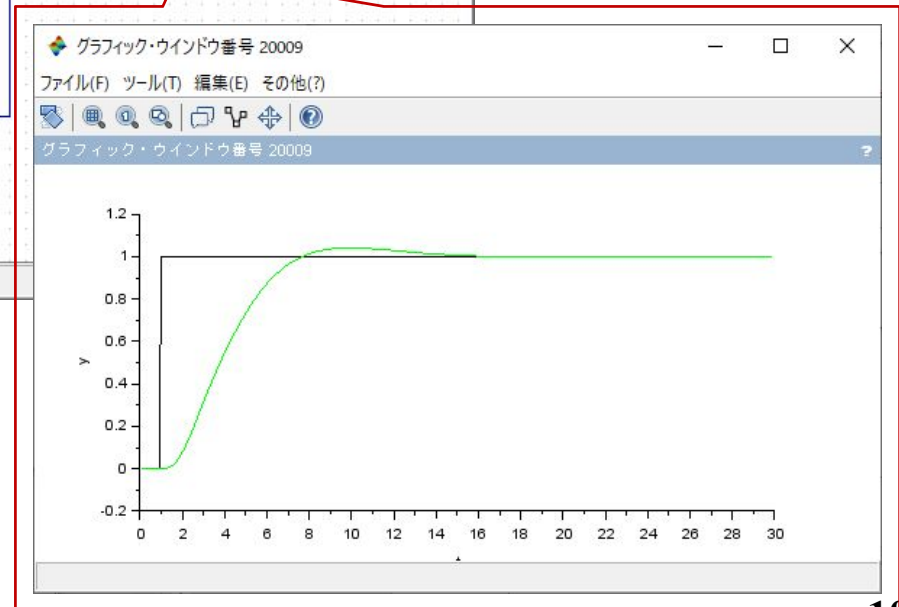
example1_2_tf.zcos

$$\omega_n = 0.5, \zeta = 0.7$$

スタート
ボタン



Wクリック



[Scilab演習]

example1_2_tf.zcos

$$\omega_n = 2, \zeta = 0.7$$

The image shows a Scilab simulation environment with a control system model and a dialog box for setting parameters. The model consists of a step input, a summing junction, a gain block, a derivative block, another summing junction, a transfer function block, and a multiplexer (MUX) leading to a plot. The dialog box is titled "Scilab複数値リクエスト" and "GAIN_f ブロックパラメータを設定". It shows a "Gain" field with the value "3*0.5^2/2".

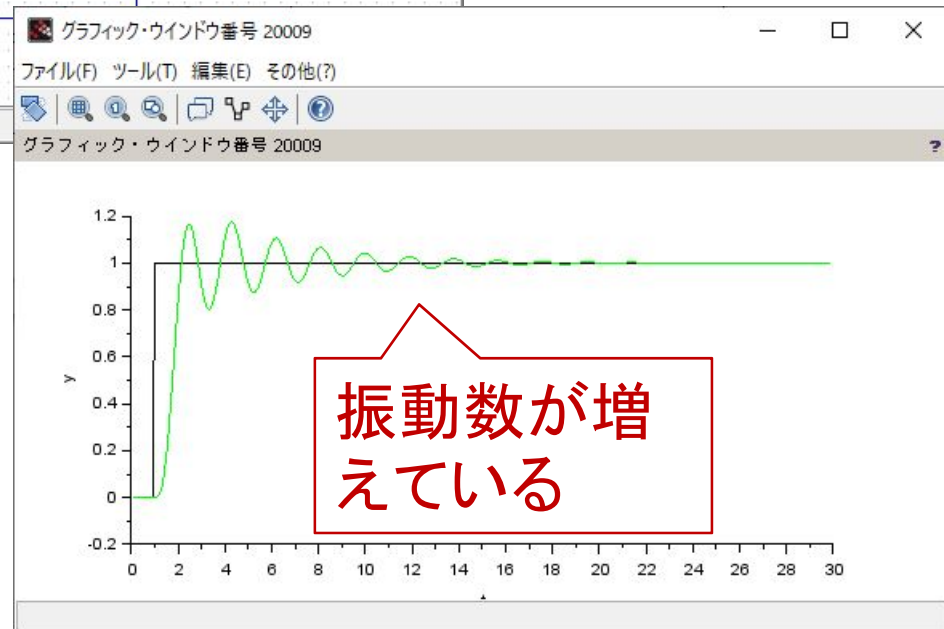
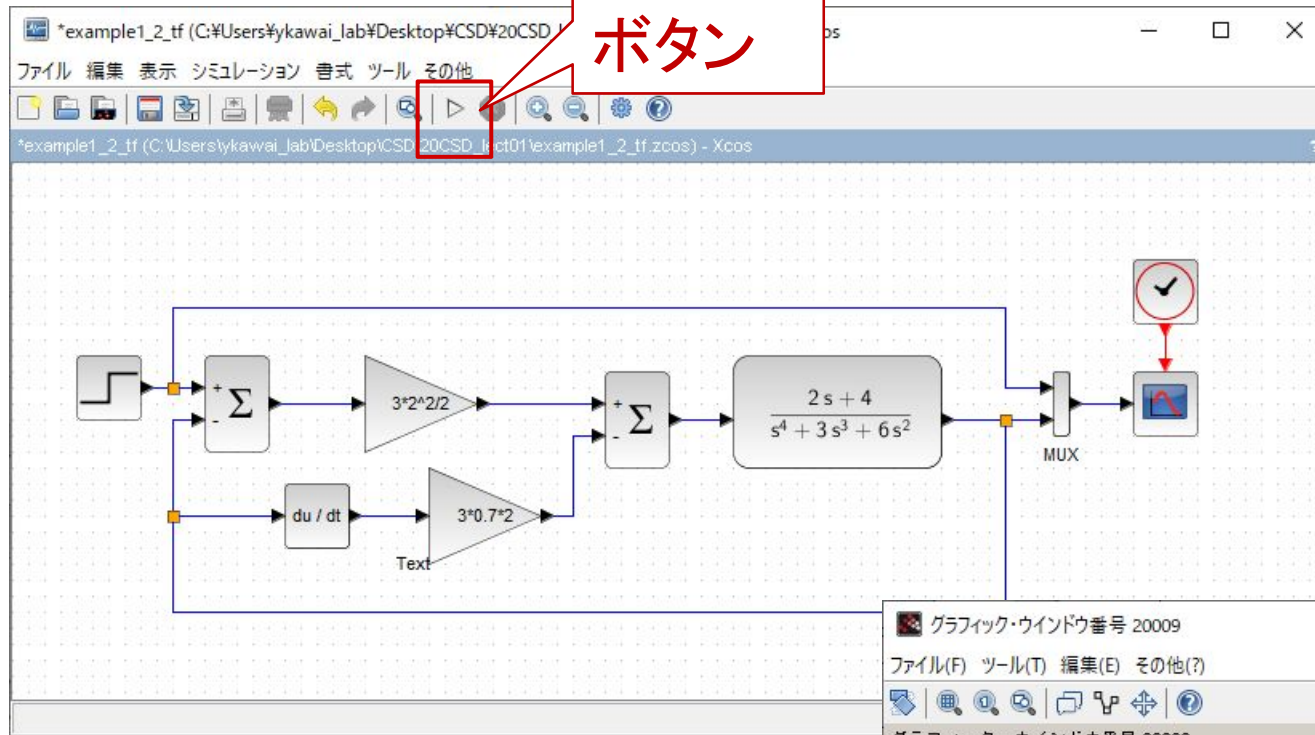
Wクリック

Wクリックして
 $3*0.7*2$
に変更する

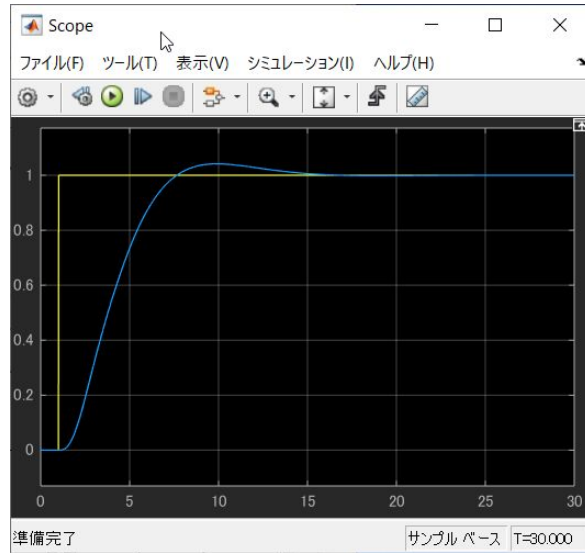
3*2^2/2
に変更する

[Scilab演習]

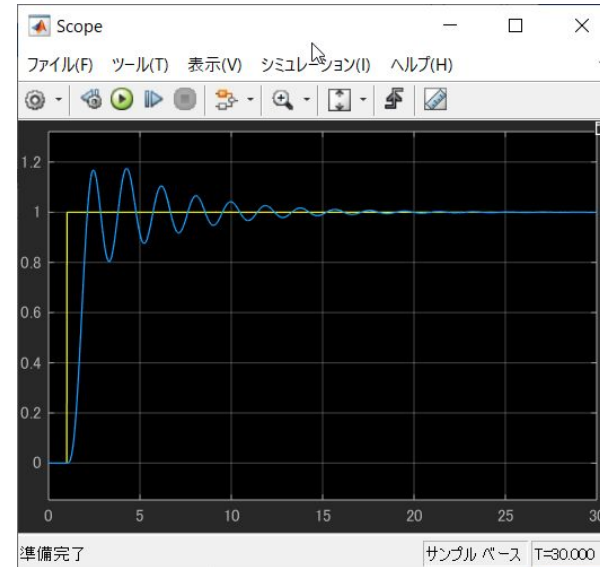
スタート
ボタン



$$\omega_n = 0.5, \zeta = 0.7$$



$$\omega_n = 2, \zeta = 0.7$$



ω_n を大きくすると不安定になる。

制御対象

$$P(s) = \frac{2s + 4}{s^2 (s^2 + 3s + 6)} = \frac{2s + 4}{s^4 + 3s^3 + 6s^2}$$

4次

P-D コントローラ

$$u(t) = k_P e(t) - k_D \dot{y}(t)$$

2つしかパラメータない
(極を任意に設定できない)

1 古典制御理論から現代制御理論へ

1.2 現代制御理論における高次システムへの取り扱い

[例 1.3]

台車2に加えて, 台車1の状態を利用

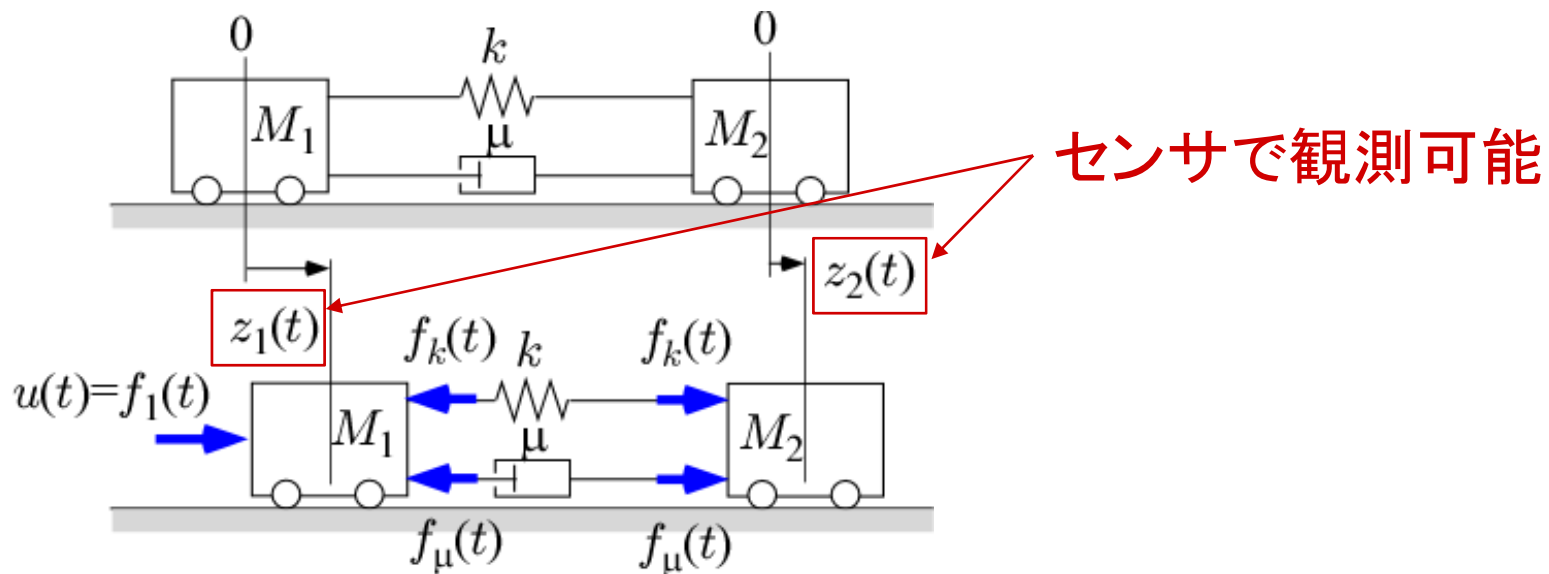
P-D コントローラ

$$u(t) = \underbrace{k_{P1}e_1(t) - k_{D1}\dot{z}_1(t)}_{\text{台車1のP-D制御}} + \underbrace{k_{P2}e_2(t) - k_{D2}\dot{z}_2(t)}_{\text{台車2のP-D制御}}$$

台車1のP-D制御

台車2のP-D制御

$$e_1(t) = y^{\text{ref}}(t) - z_1(t), \quad e_2(t) = y^{\text{ref}}(t) - z_2(t)$$



$$u(t) = \underbrace{k_{P1}(y^{\text{ref}}(t) - z_1(t)) - k_{D1}\dot{z}_1(t)}_{\text{台車1のP-D制御}} + \underbrace{k_{P2}(y^{\text{ref}}(t) - z_2(t)) - k_{D2}\dot{z}_2(t)}_{\text{台車2のP-D制御}}$$

台車1のP-D制御

台車2のP-D制御

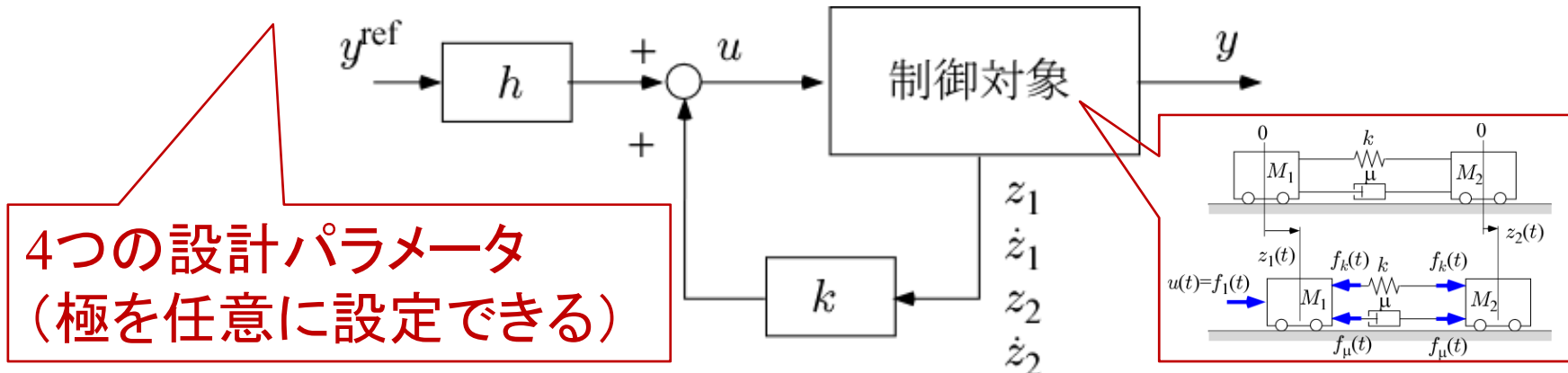
$$= \underbrace{\begin{bmatrix} -k_{P1} & -k_{D1} & -k_{P2} & -k_{D2} \end{bmatrix}}_k \underbrace{\begin{bmatrix} z_1(t) \\ \dot{z}_1(t) \\ z_2(t) \\ \dot{z}_2(t) \end{bmatrix}}_{x(t)} + \underbrace{(k_{P1} + k_{P2})}_{h} y^{\text{ref}}(t)$$

台車1の状態
台車2の状態

$$= kx(t) + hy^{\text{ref}}(t)$$

コントローラのゲイン(最適レギュレータ)

$$k = \begin{bmatrix} -1.90 & -1.71 & -5.84 & -4.45 \end{bmatrix}, \quad h = 7.75$$



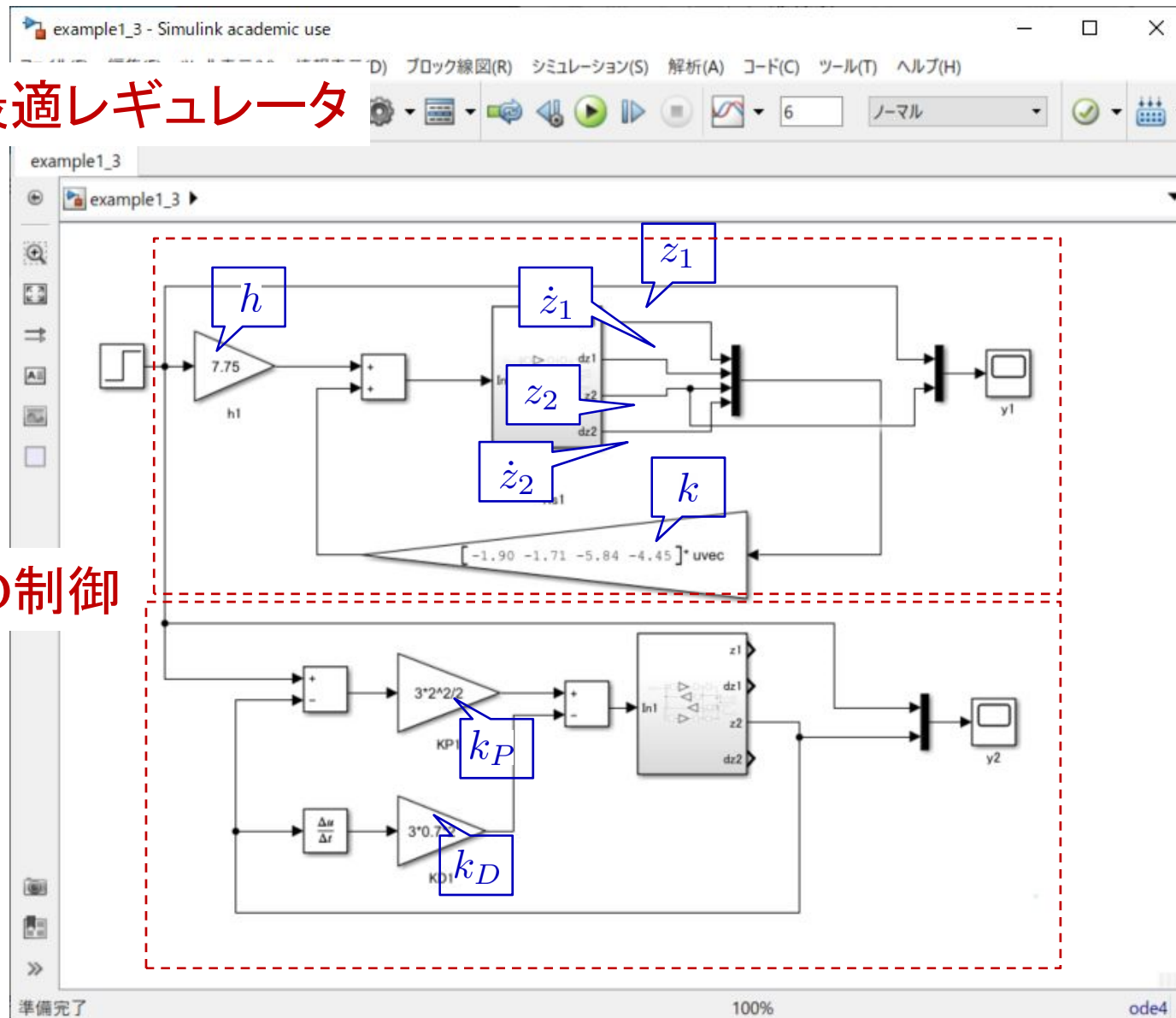
4つの設計パラメータ
(極を任意に設定できる)

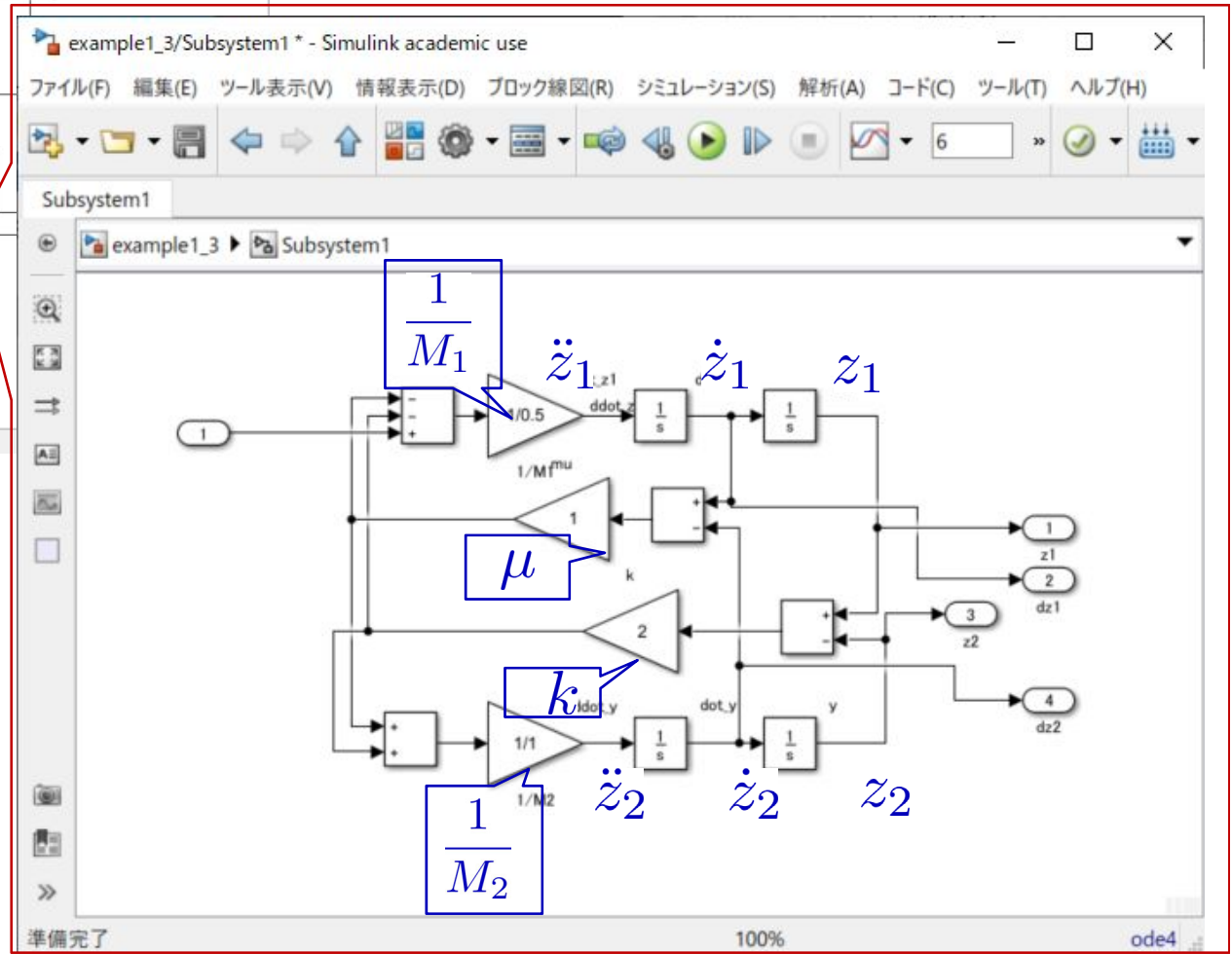
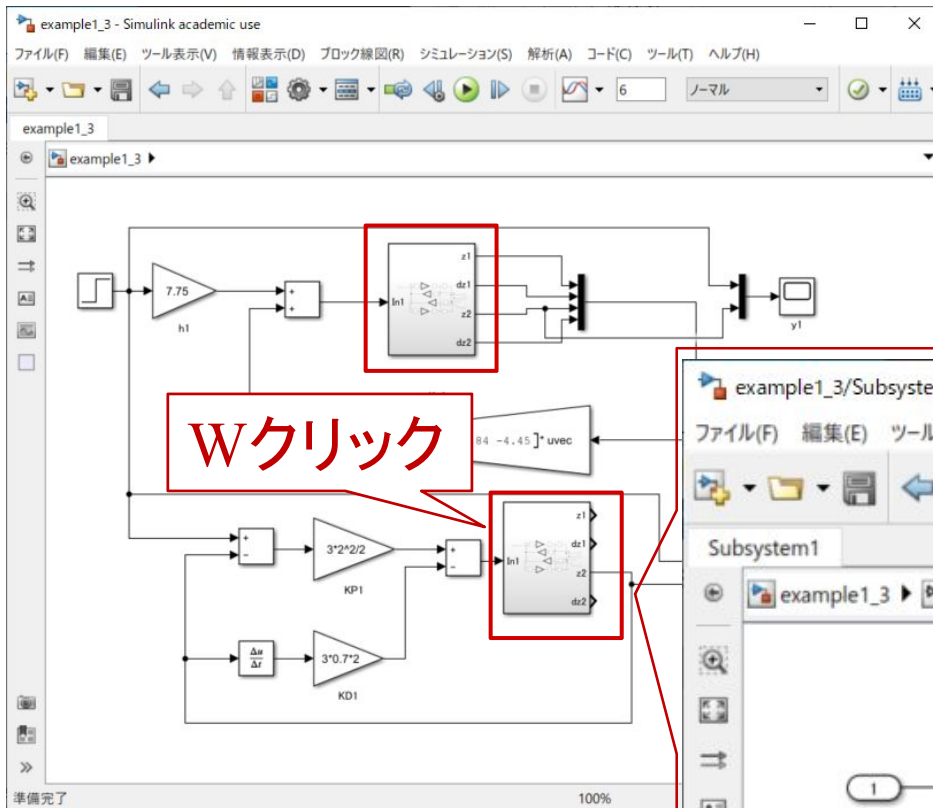
[MATLAB演習]

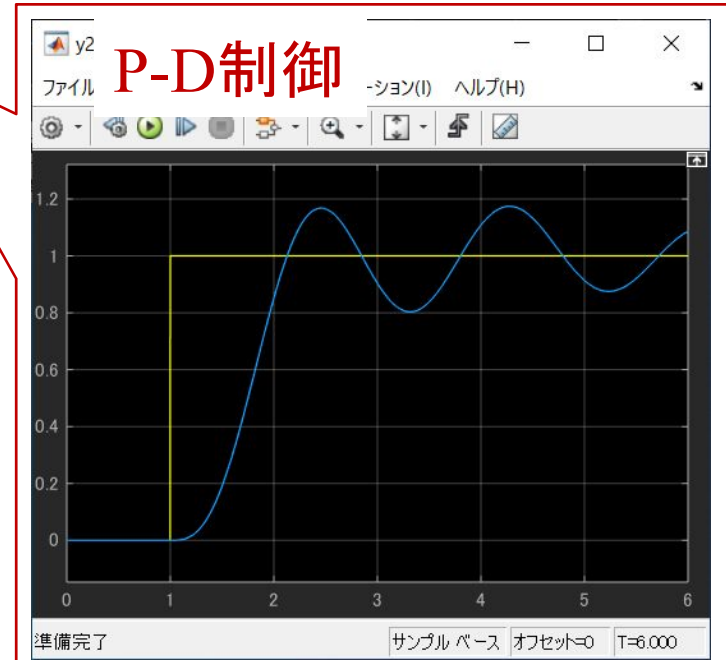
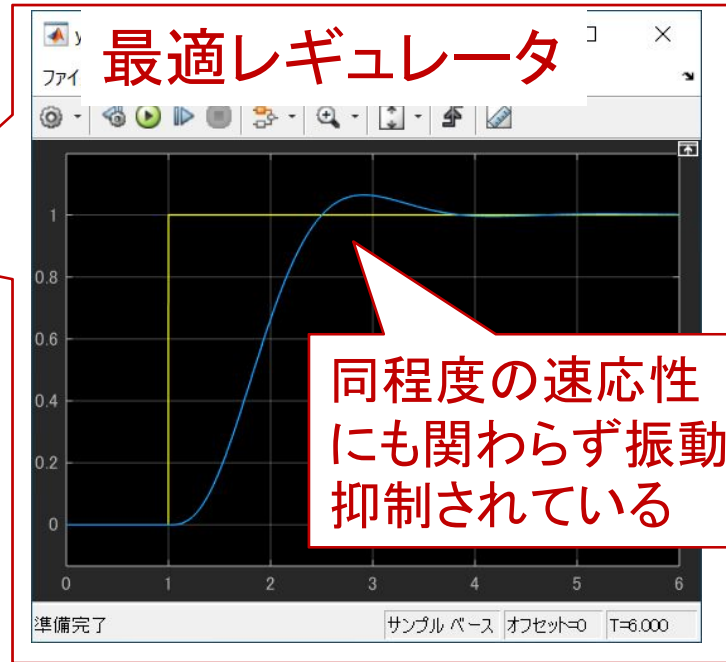
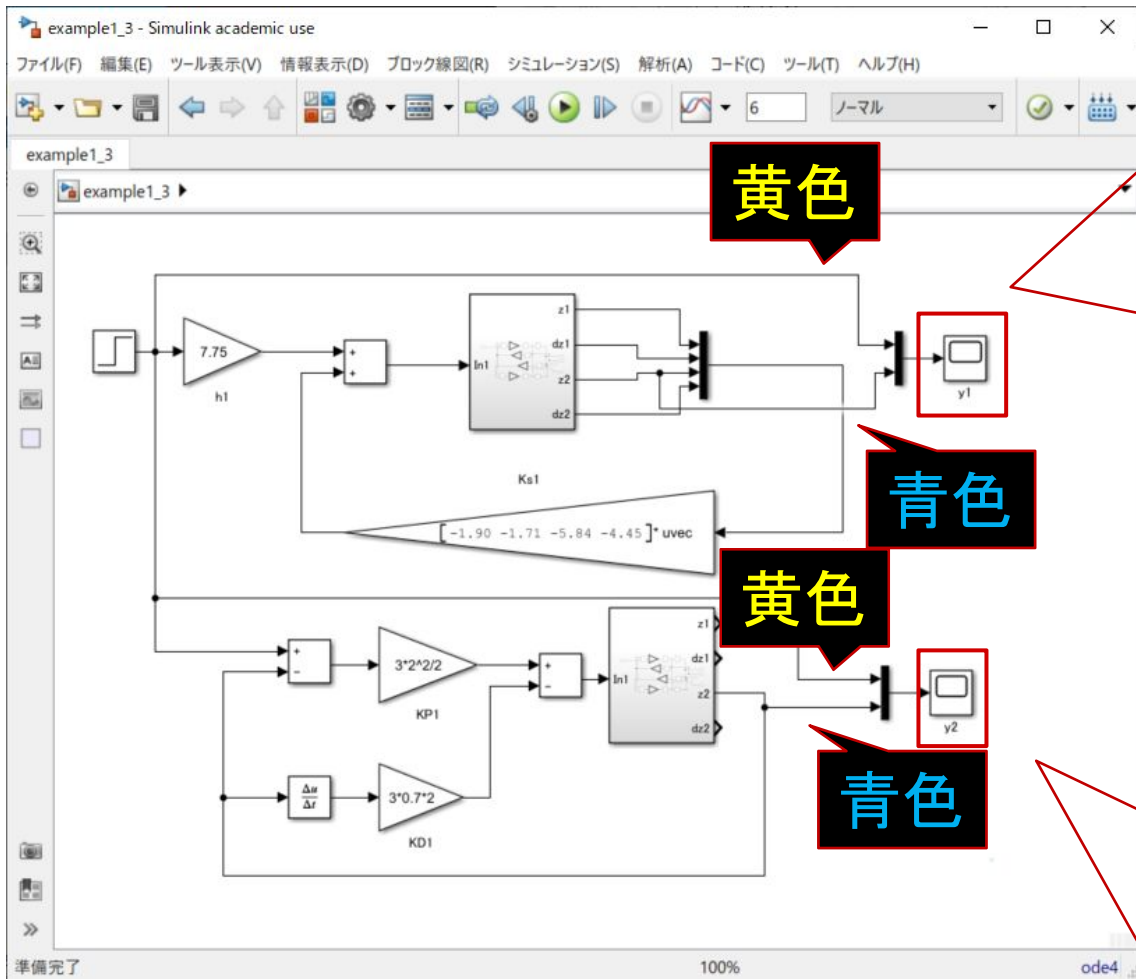
example1_3.mdl

最適レギュレータ

P-D制御





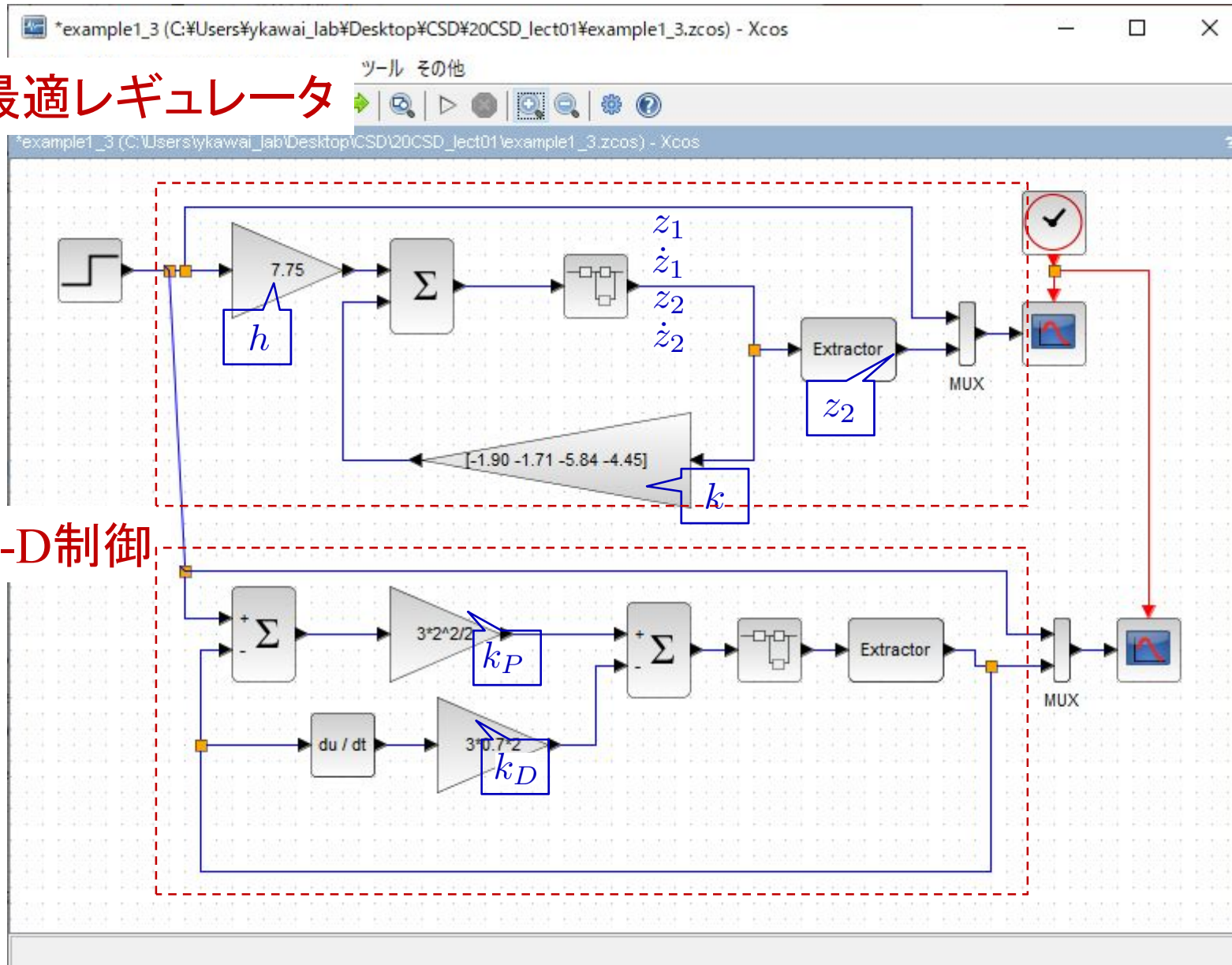


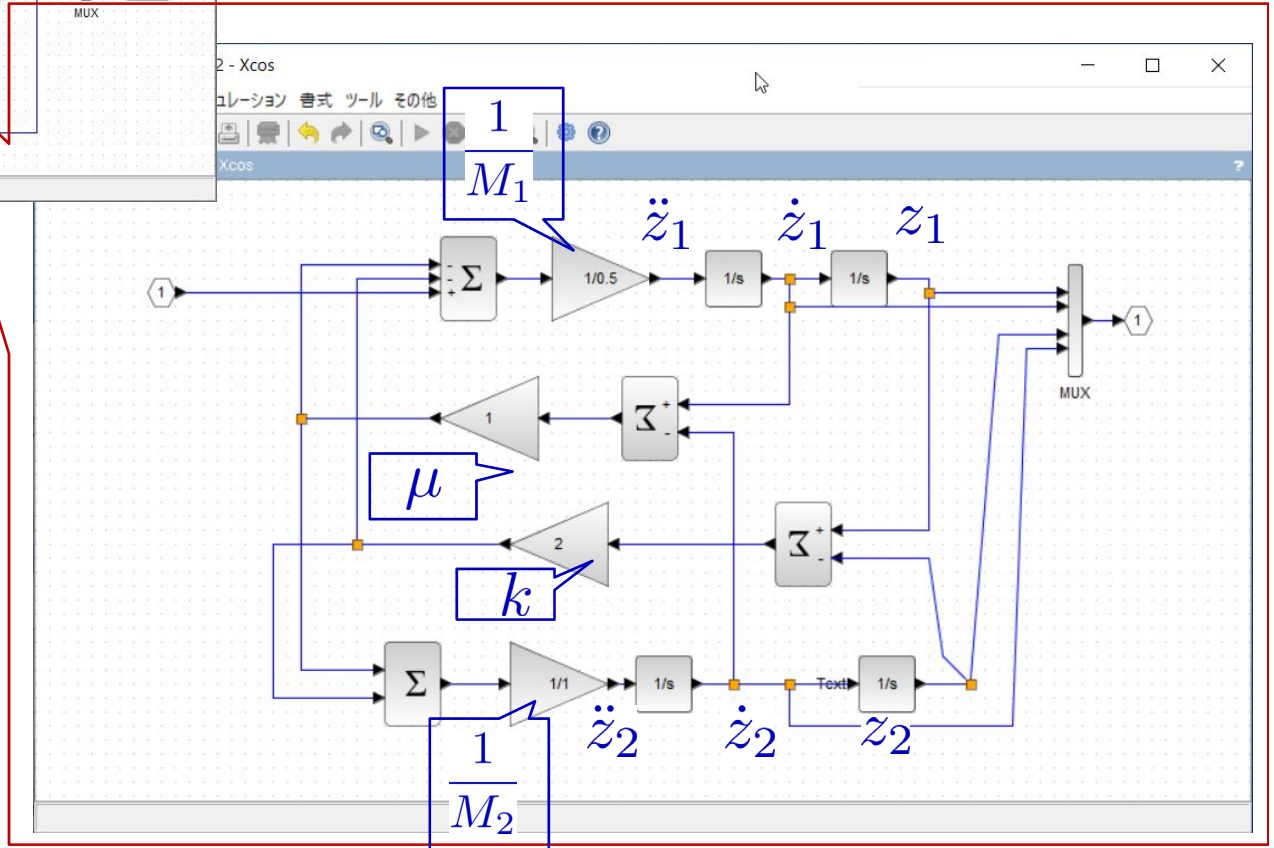
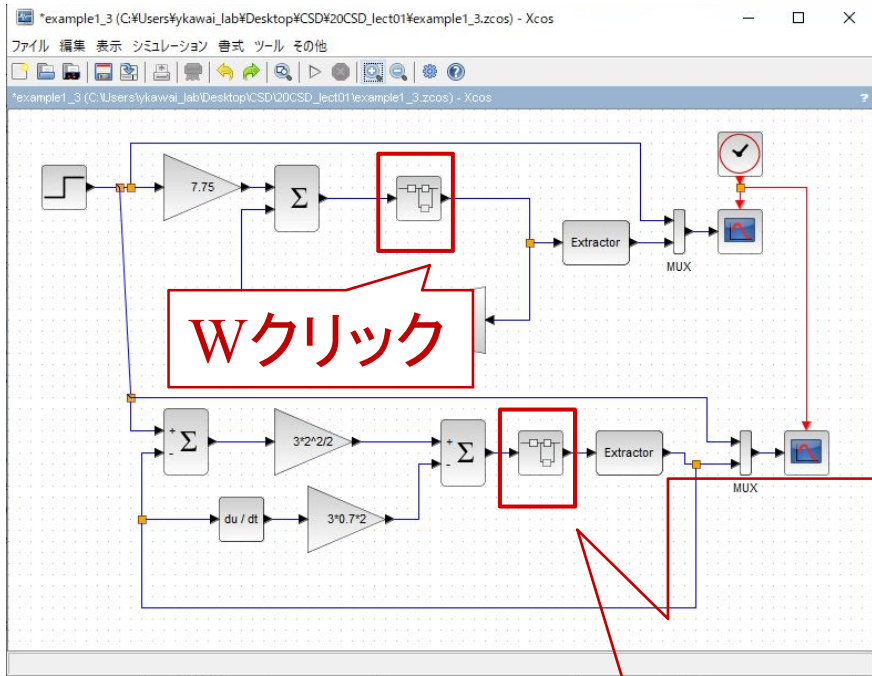
[Scilab演習]

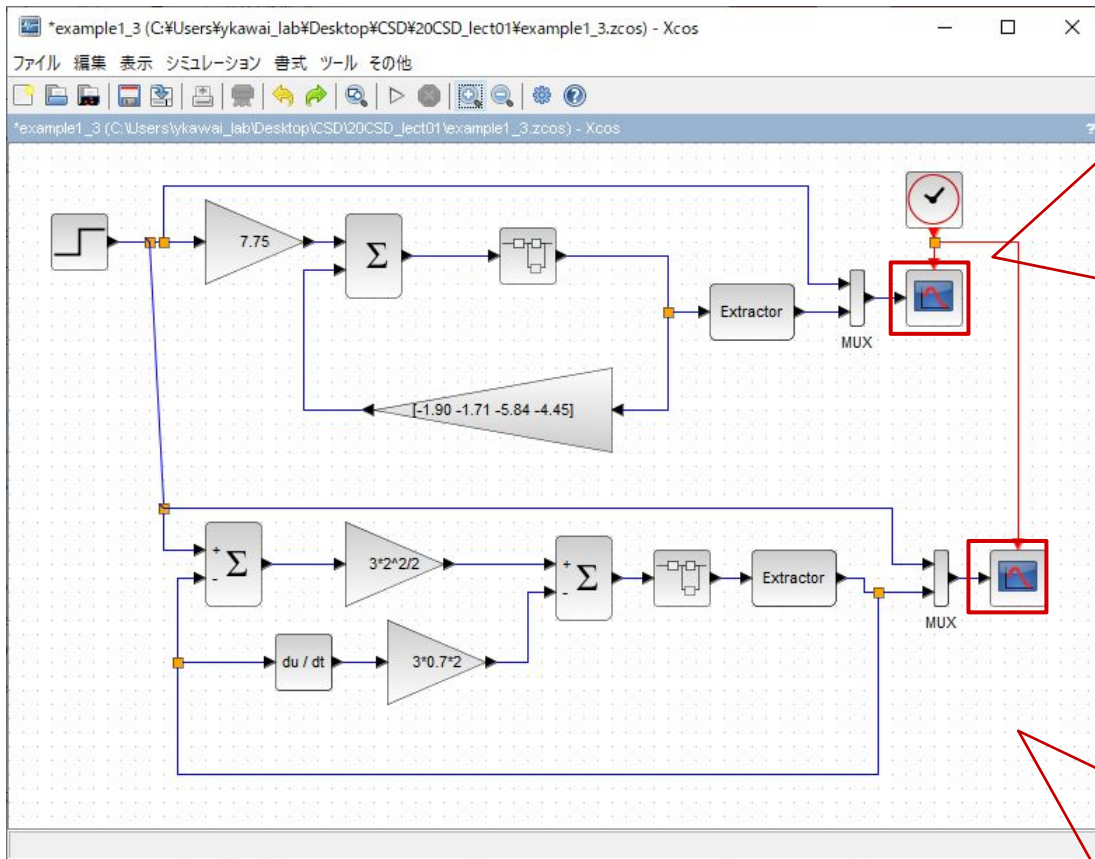
example1_3.zcos

最適レギュレータ

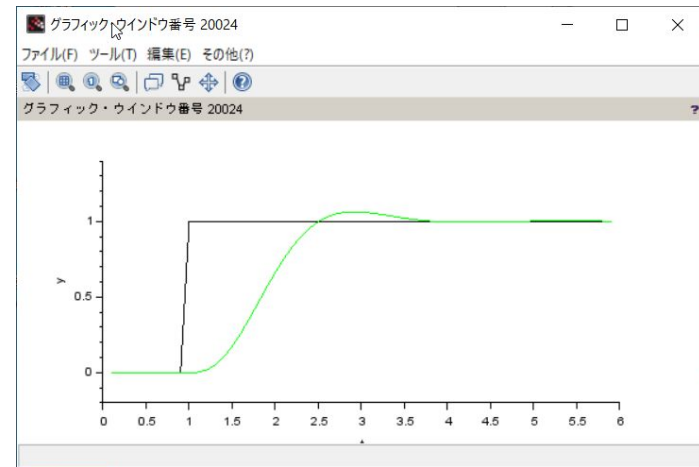
P-D制御



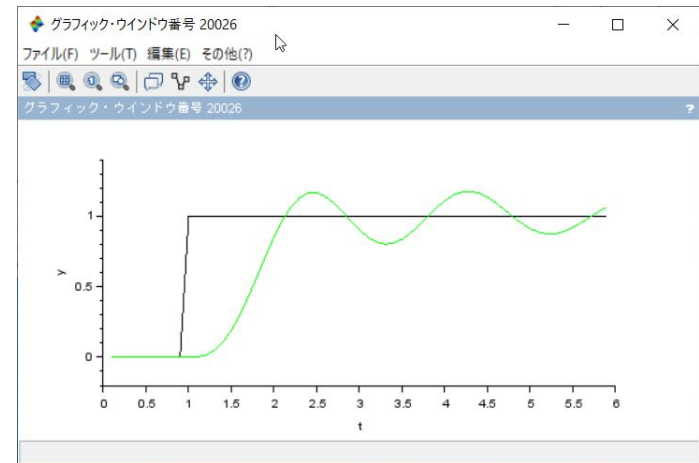




最適レギュレータ



P-D制御



制御対象

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) && \text{状態方程式} \\ y(t) &= Cx(t) && \text{出力方程式} \end{aligned}$$

状態空間表現

$$x(t) = \begin{bmatrix} z_1(t) & \dot{z}_1(t) & z_2(t) & \dot{z}_2(t) \end{bmatrix}^T$$

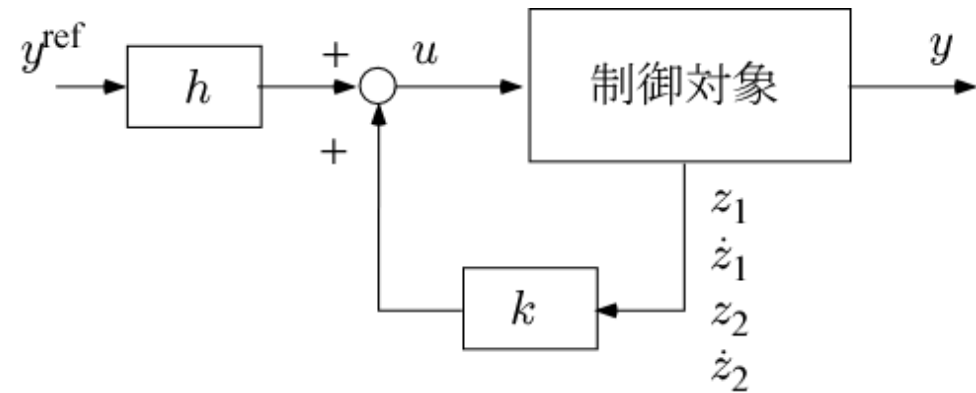
コントローラ

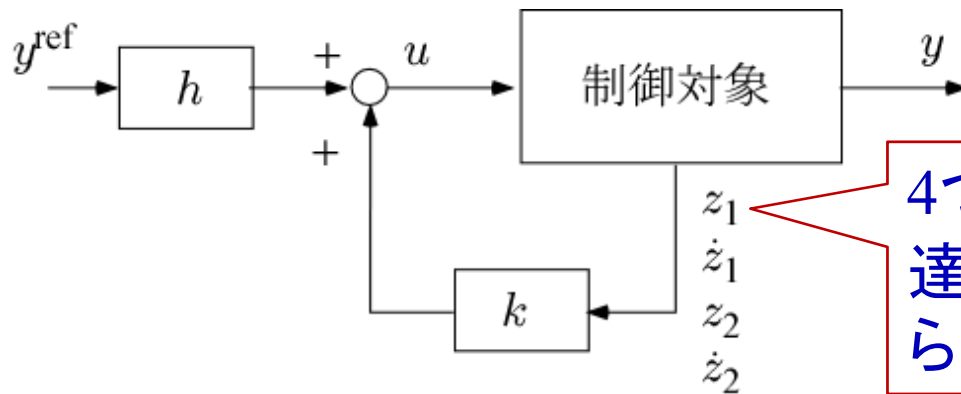
$$u(t) = Kx(t) + Hy^{\text{ref}}(t)$$

$$y^{\text{ref}}(t) = y_1^{\text{ref}}(t)$$

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \end{bmatrix},$$

$$H = h_{11}$$

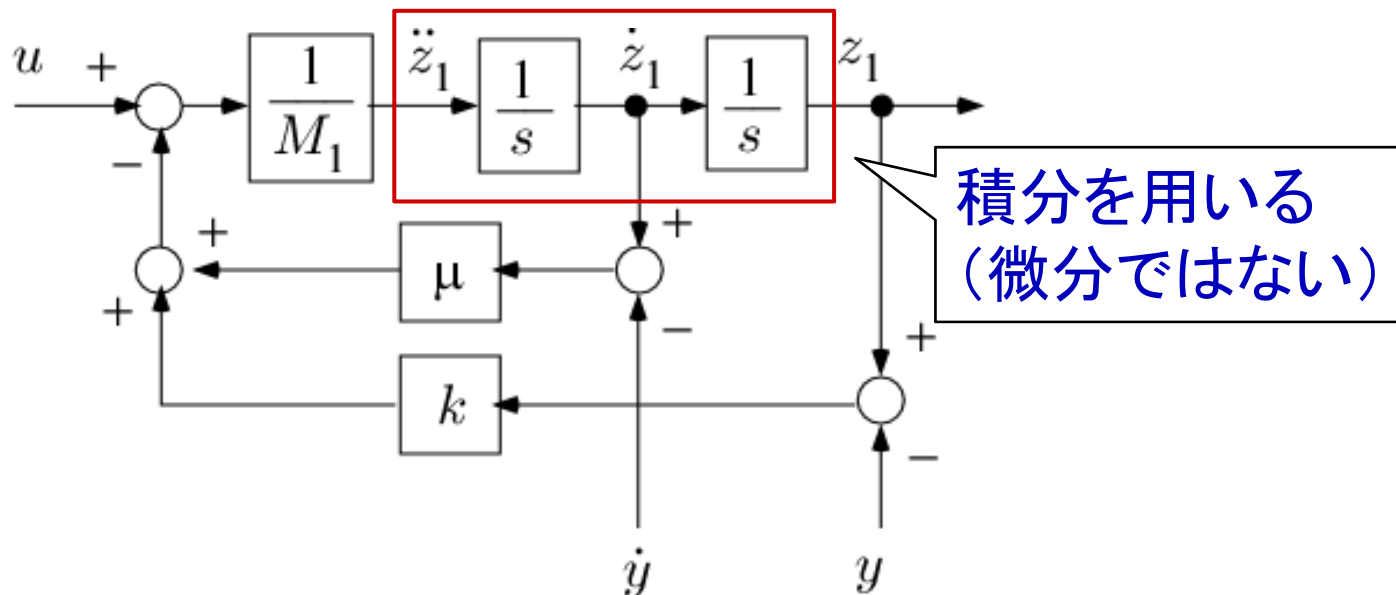




4つの情報を観測するために伝達関数ではなく、微分方程式から考える

$$M_1 \ddot{z}_1(t) = u(t) - k(z_1(t) - y(t)) - \mu(\dot{z}_1(t) - \dot{y}(t))$$

$$\ddot{z}_1(t) = \frac{1}{M_1} (u(t) - k(z_1(t) - y(t)) - \mu(\dot{z}_1(t) - \dot{y}(t)))$$

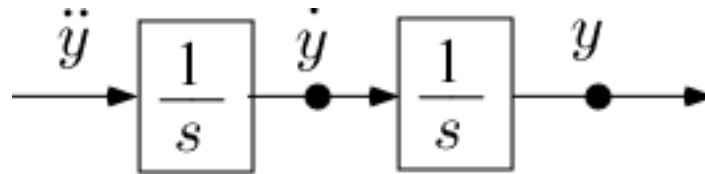


積分を用いる
(微分ではない)

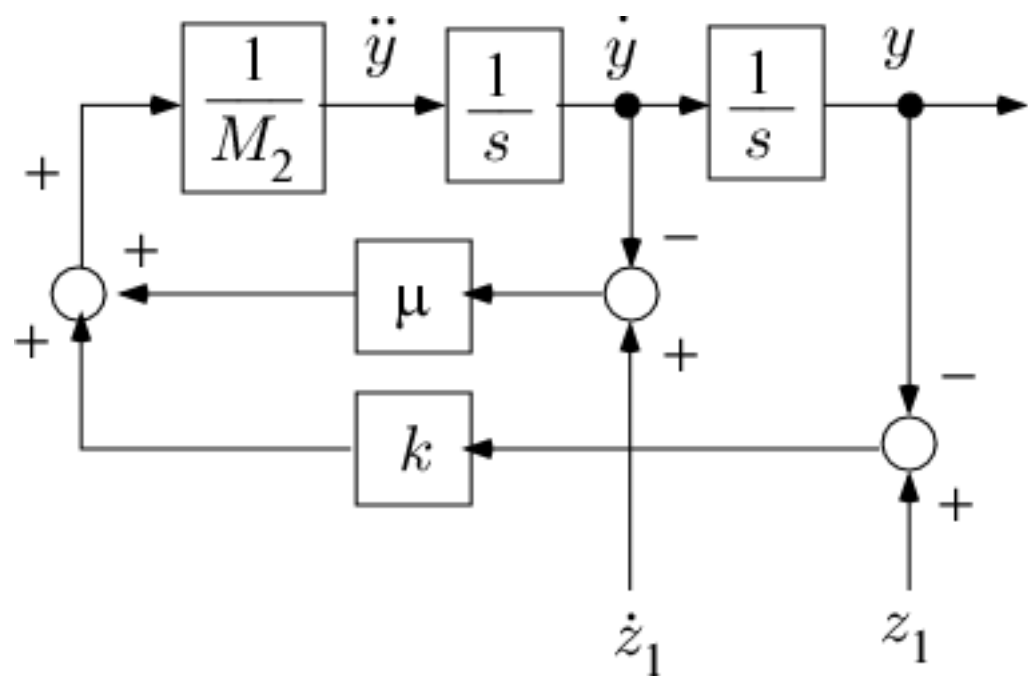
[計算演習]

ブロック線図で描け

$$M_2 \ddot{y}(t) = k(z_1(t) - y(t)) + \mu(\dot{z}_1(t) - \dot{y}(t))$$



[模範解答]



第1章 古典制御理論から現代制御理論へ

1.1 高次システムに対する古典制御理論の限界

1.2 現代制御理論における高次システムの取り扱い

キーワード：SISO, MIMO, 状態方程式, 出力方程式

学習目標：古典制御理論で困難な多入力多出力システムを容易にする現代制御理論の重要性を理解する。