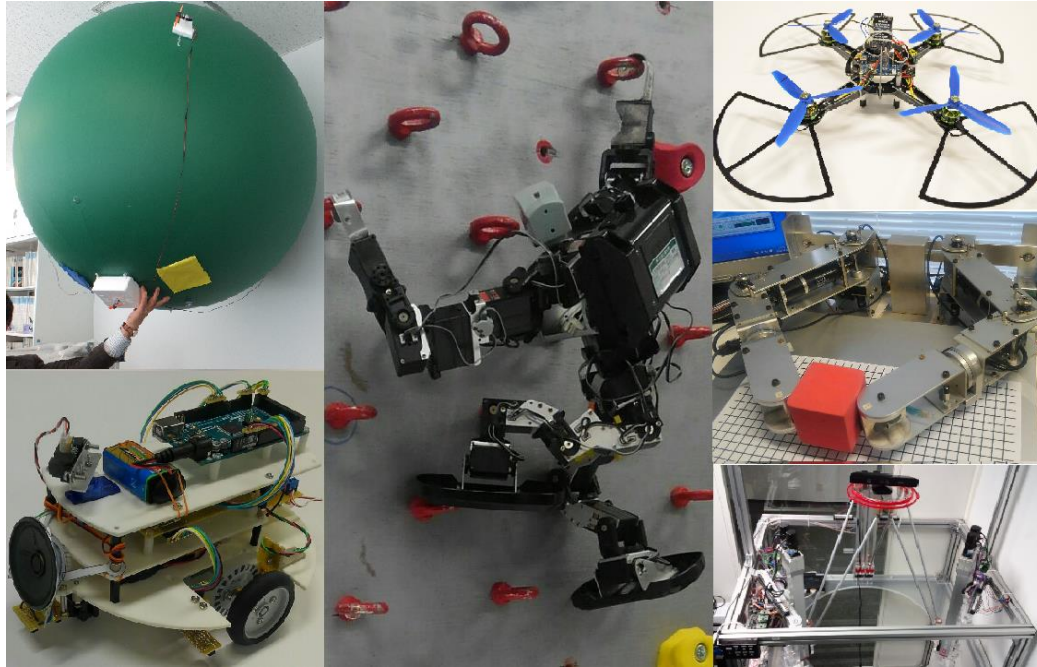


MATLABを用いた実践ロボット制御教育



2017/10/31

Track D3 15:10-15:50

芝浦工業大学 デザイン工学部

島田 明

内容

- はじめに
- デザイン工学部と教育内容
- EEテキスト モーションコントロール
- MATLABを用いたメカトロニクス教育・ロボティクス教育
- MATLABを用いた制御実験
- 研究室で: クライミングロボットと最適化
- おわりに

はじめに

- **芝浦工業大学デザイン工学部**のロボティクス・情報デザイン系では、メカトロニクス、ロボティクス、モーションコントロールの講義科目と連動して、MATLAB®/Simulink®を利用した演習や実習を行っている。
- 制御の基礎理論、ロボティクスの基礎理論には、難解な数学が登場する。それらを理解し、使えるようになり、応用できるまでには、**体系的な講義・シミュレーション・実験**が有効である。どんな予備知識を持った学生が、どのような環境で学んでいるか、何に注意を払い、ロボット制御技術を習得できるようになるかを例を用いて紹介する。
- **講演者**は元は産業用ロボット開発に従事していたエンジニア。メーカーおよび大学での実務経験を活かした講演を行う。

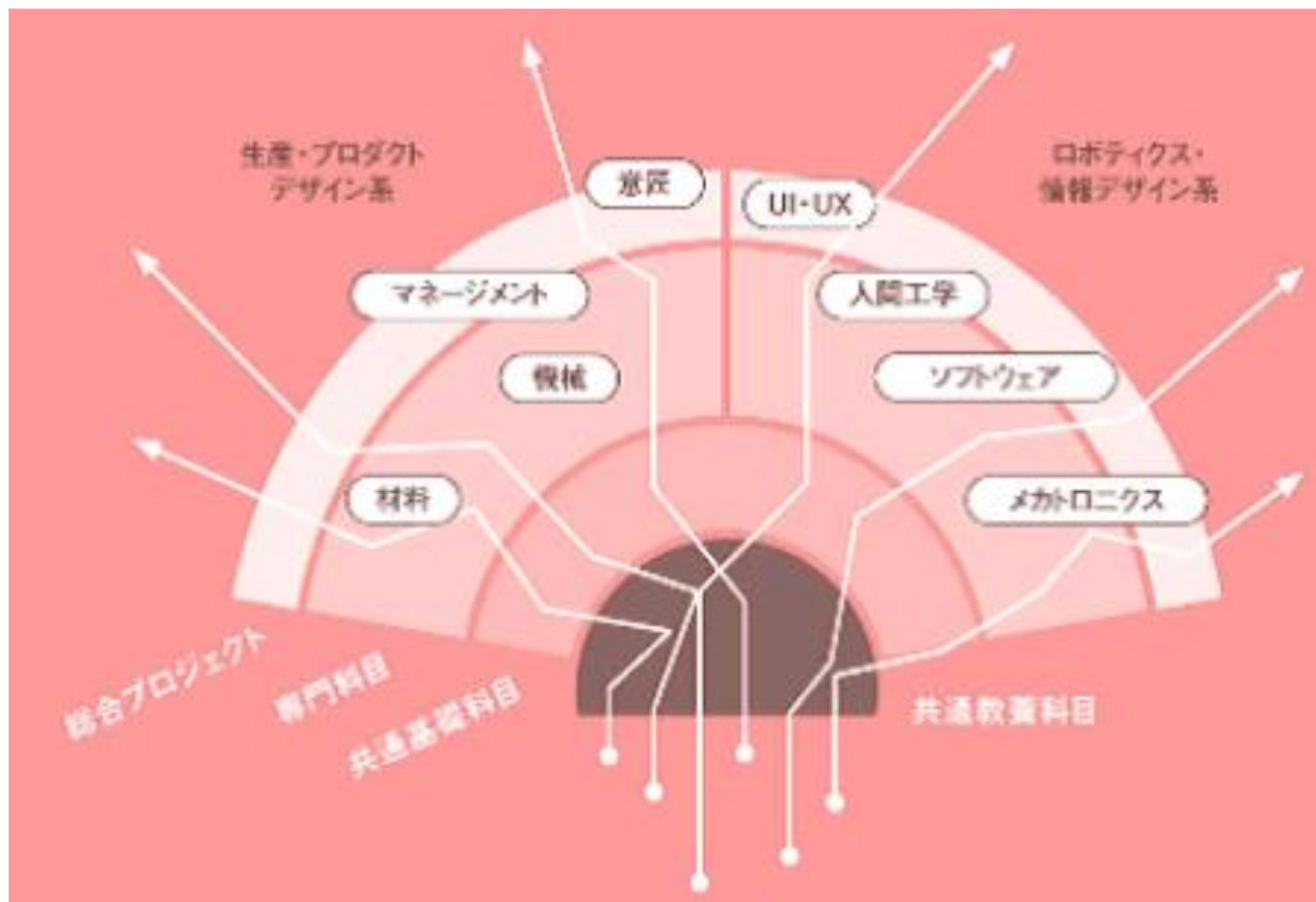
組織と教育内容：デザイン工学部



1-2年：大宮キャンパス
3-4年 & 院生：芝浦キャンパス

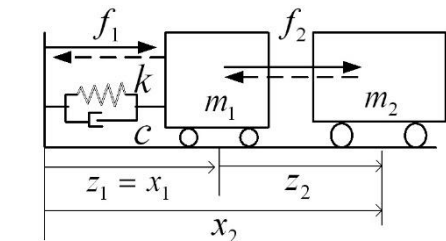
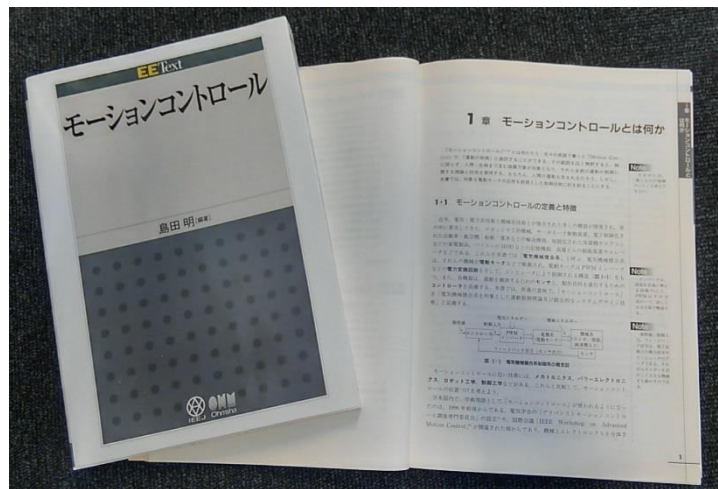
[学部]
工学部
システム理工学部
デザイン工学部
建築学部

[大学院]
理工学研究科

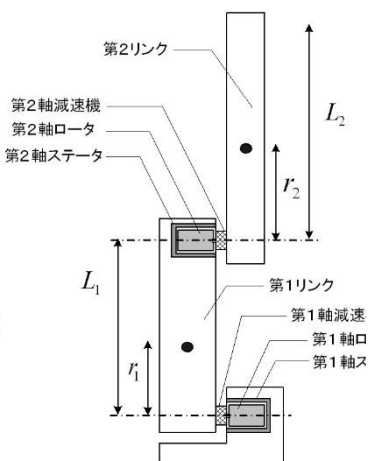


生産・プロダクトデザイン系 & ロボティクス・情報デザイン系

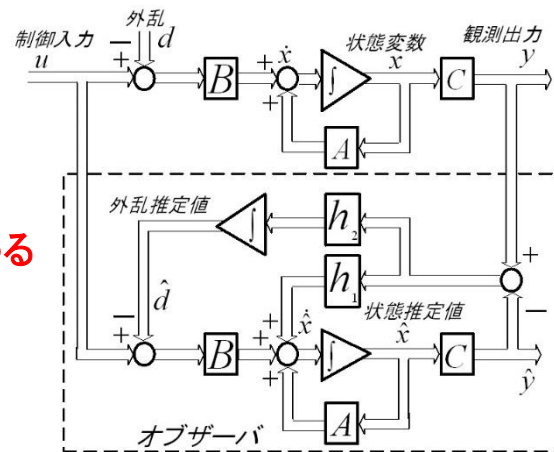
EEテキスト モーションコントロール



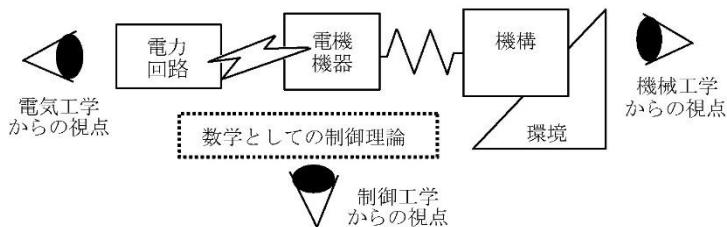
一般化座標を換えると式が換わる



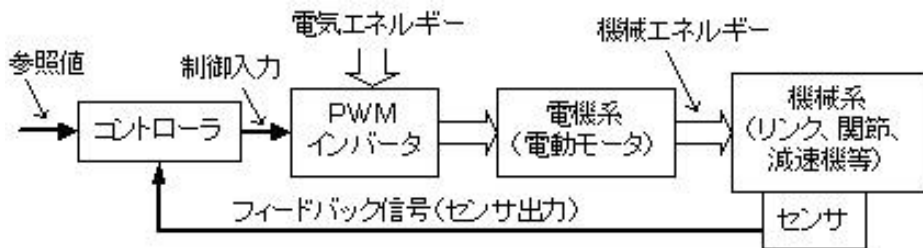
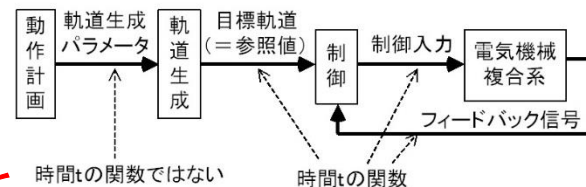
アクチュエータが内蔵されると運動方程式は...?



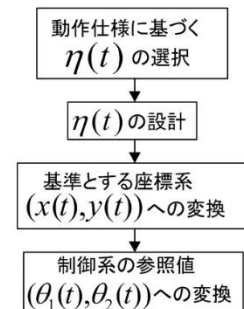
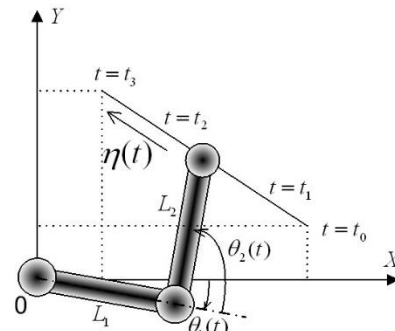
外乱オブザーバ設計



異なる視点からの物の見え方



制御系とハードウェアの関係



ロボットモーションコントロールの仕組み

EEテキスト モーションコントロール

(b) 外乱オブザーバの設計法⁵⁾¹¹⁾¹⁶⁾

入力端への外乱を考慮した制御対象に関する状態方程式と出力方程式を考える。

$$\dot{x} = Ax + Bu - Bd \quad (6.44)$$

$$y = Cx \quad (6.45)$$

ただし、 $x \in R^n$, $u \in R^m$, $d \in R^m$, $y \in R^p$, $A \in R^{n \times n}$, $B \in R^{n \times m}$, $C \in R^{p \times n}$.

簡単な外乱 $d(t)$ は時間微分 $\dot{d}(t) = 0$ のステップ状外乱。拡大系は次式となる。

$$\begin{bmatrix} \dot{x} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} A & -B \\ 0_{m \times n} & 0_{m \times m} \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0_{m \times n} \end{bmatrix} u \quad (6.46)$$

$$y = [C \ 0_{p \times m}] \begin{bmatrix} x \\ d \end{bmatrix} \quad (6.47)$$

上記が可観測ならば、同一次元外乱オブザーバ式(6.48)が得られる。

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{d}} \end{bmatrix} = \begin{bmatrix} A & -B \\ 0_{m \times n} & 0_{m \times m} \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} + \begin{bmatrix} B \\ 0_{m \times n} \end{bmatrix} u + \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} (y - \hat{y}) \quad (6.48)$$

$$\hat{y} = [C \ 0_{p \times m}] \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} \quad (6.49)$$

図 6.33 参照。

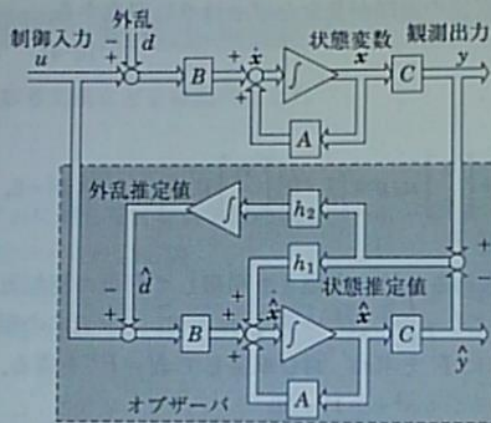


図 6.33 外乱オブザーバの構造

Note

外乱とは外から加わる力だけでなく、摩擦や重力、モデルに含めない弾性力、質量や慣性モーメントの誤差で生じる加速トルク誤差、モータのトルク定数の変動によるトルク誤差などの総和である。

Note

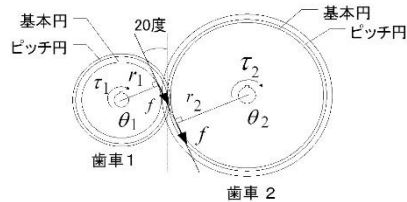
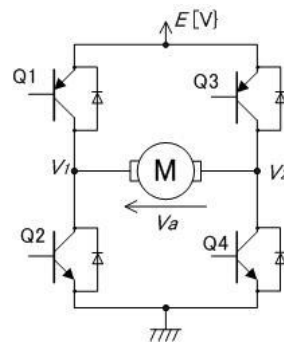
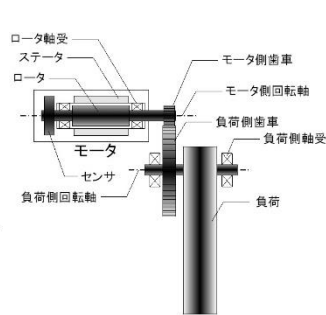
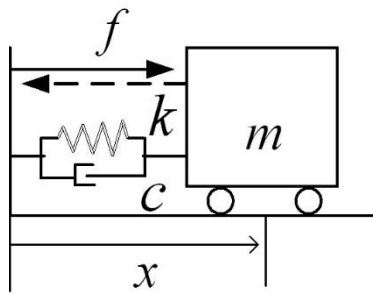
MATLABによる外乱オブザーバの計算プログラムは、
 $A = [0, 1; -1, -2]$;
 $B = [0, 1]$; $C = [1, 0]$; $D = [0]$;
 $OA = [A, -B;$
 $\text{zeros}(1, 2), 0]$;
 $OB = [B; 0]$;
 $OC = [C, 0]$;
 $OD = D$;
 $p1 = [-5, -6, -7]$;
 $F = \text{place}(OA', OC', p1)$; $H = F'$;

Note

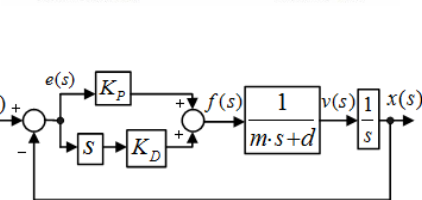
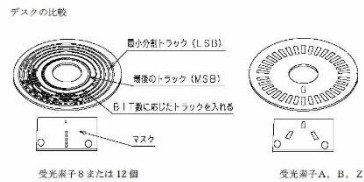
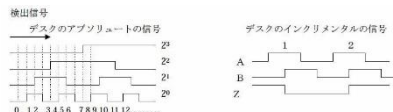
実践的な理論の解説書
 ながら、所々に
 MATLABでの活用法を紹介

外乱オブザーバ設計
 が可能に!

メカトロニクスの講義 (2年後期)



物と理屈の関係を教える



**手計算させる:
応答計算 & Bode線図**

メカトロニクス 課題

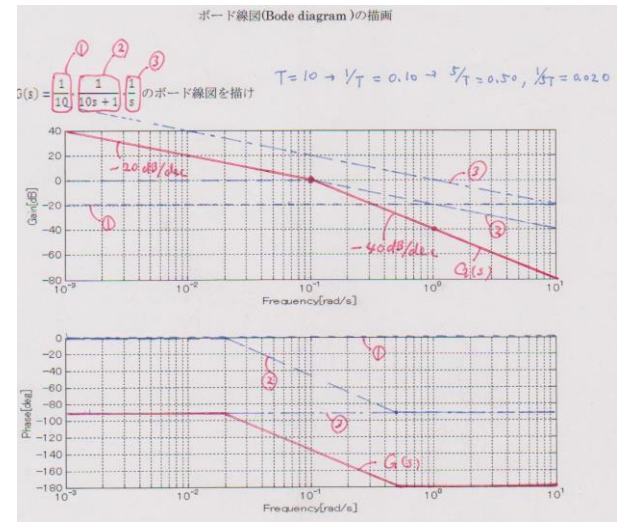
Date 2016.10.10

$$U(s) \rightarrow \frac{Y(s)}{G(s)} \quad G(s) = \frac{12}{s^2 + 5s + 6} \quad \text{に対し}$$

インパルス応答とインテリヤル応答を求めよ。

1) インパルス応答
 $U(t) = \delta(t) \xrightarrow{L} U(s) = 1$
 $Y(s) = G(s) \cdot U(s) = \frac{12}{s^2 + 5s + 6} \cdot 1 = \frac{12}{(s+2)(s+3)} = \frac{a}{s+2} + \frac{b}{s+3}$
 $a = (s+2) \cdot Y(s) |_{s=-2} = \frac{12}{3+2} |_{s=-2} = \frac{12}{5} = 2.4$
 $b = (s+3) \cdot Y(s) |_{s=-3} = \frac{12}{5+2} |_{s=-3} = \frac{12}{7} \approx 1.71$
 $\therefore Y(s) = 2.4 \cdot \frac{1}{s+2} - 0.31 \cdot \frac{1}{s+3} \xrightarrow{L^{-1}} y(t) = 2.4e^{-2t} - 0.31e^{-3t}$

2) インテリヤル応答
 $U(t) = 1 \xrightarrow{L} U(s) = \frac{1}{s}$
 $Y(s) = G(s) \cdot U(s) = \frac{12}{s^2 + 5s + 6} \cdot \frac{1}{s} = \frac{12}{s(s+2)(s+3)}$
 $= \frac{a}{s} + \frac{b}{s+2} + \frac{c}{s+3}$
 $a = (s) Y(s) |_{s=0} = \frac{12}{(5+2)(5+3)} = \frac{12}{49} \approx 0.24$
 $b = (s+2) Y(s) |_{s=-2} = \frac{12}{(-2)(-2+3)} = \frac{12}{-2} = -6$
 $c = (s+3) Y(s) |_{s=-3} = \frac{12}{(-3)(-3+2)} = \frac{12}{3} = 4$
 $\therefore Y(s) = 0.24 \cdot \frac{1}{s} - 6 \cdot \frac{1}{s+2} + 4 \cdot \frac{1}{s+3} \xrightarrow{L^{-1}} y(t) = 0.24 - 6e^{-2t} + 4e^{-3t}$



その上でMATLAB

```

% 講義・実習等々メカトロニクス*exercise_bode_3.m
% Bode diagram exercise
clear all; clc; format short e

%% parameters
num1=77*[ 0.5 8.3]; % revised
den1=[ 1 0.22 89];
num2=1; den2=[ 5 1 0.1]*3; % revised
num3=0.5; den3=[ 0.005 1]; % revised 200161212

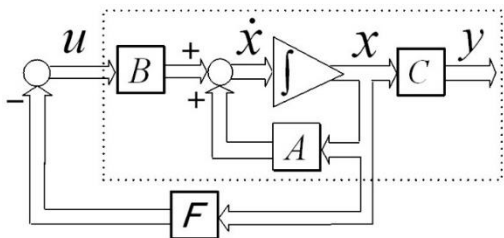
%% transfer function
num=conv(num1,num2); num=conv(num,num3);
den=conv(den1,den2); den=conv(den,den3);

%% Bode diagram
w=logspace(-3,3,200);
[mag,phase]=bode(num,den,w);
figure(1);
subplot(2,1,1); semilogx(w,20*log10(mag)); grid on;
axis([0.1,1000,-100,40]);
xlabel('Frequency[rad/s]'); ylabel('Gain[dB]');
subplot(2,1,2); semilogx(w,phase); grid on;
axis([0.1,1000,-270,90]);
xlabel('Frequency[rad/s]'); ylabel('Phase[deg]');

figure(2); nyquist(num,den); grid on;
axis([-5,10,-10,2]);
    
```



モーションコントロール(3年前期)



1) $\begin{bmatrix} d & f_{dis} \\ m & \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ v_c \\ a_c \end{bmatrix} = \begin{bmatrix} f \\ f_{dis} \\ 0 \end{bmatrix}$ $\dot{x}_c = v_c$ ← 運動方程式 2015.6.11 島田
 $m a_c + d \cdot v_c = f - f_{dis}$
 $\therefore a_c = -\frac{d}{m} v_c + \frac{1}{m} f - \frac{1}{m} f_{dis}$

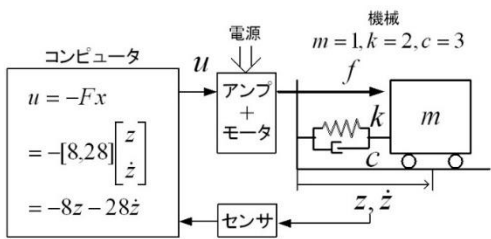
2) $\dot{\hat{x}} = \begin{bmatrix} \dot{x}_c \\ v_c \end{bmatrix}$, $u = f$, $y = x_c$
 $\begin{bmatrix} \dot{x}_c \\ v_c \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{d}{m} \end{bmatrix} \begin{bmatrix} x_c \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f - \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f_{dis}$

$y = x_c = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ v_c \end{bmatrix}$

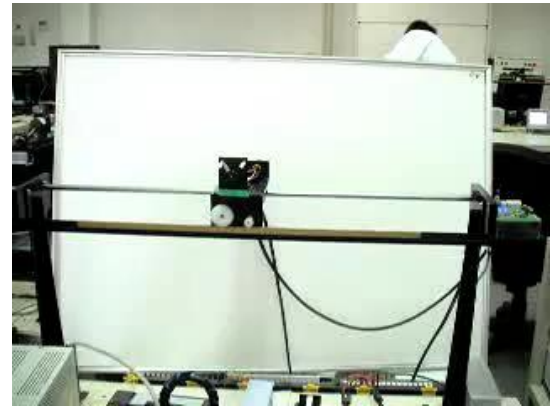
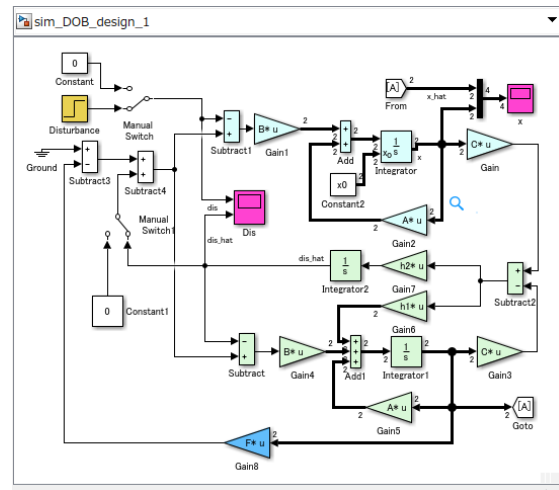
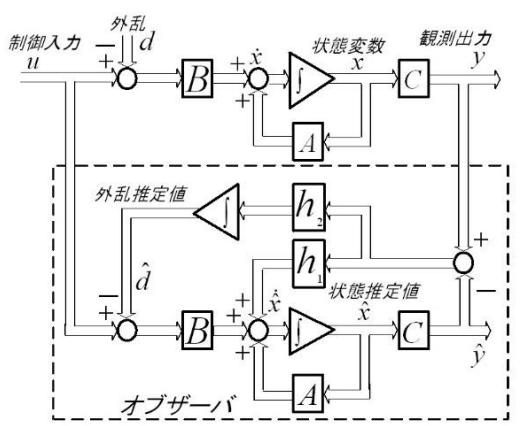
3) $f_{dis} = 0$ と仮定し.
 $\hat{x} = \begin{bmatrix} x \\ f_{dis} \end{bmatrix}$ とおくと $\begin{bmatrix} \dot{x}_c \\ v_c \\ f_{dis} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{d}{m} & -\frac{1}{m} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ v_c \\ f_{dis} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \end{bmatrix} f$
 $Y = x_c = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ v_c \\ f_{dis} \end{bmatrix}$

DOB の設計例

4) オブザーバ設計.
 $\begin{bmatrix} \dot{\hat{x}}_c \\ \dot{\hat{v}}_c \\ \dot{\hat{f}}_{dis} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{d}{m} & -\frac{1}{m} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_c \\ \hat{v}_c \\ \hat{f}_{dis} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \end{bmatrix} f + \begin{bmatrix} h_{11} \\ h_{12} \\ h_2 \end{bmatrix} (x_c - \hat{x}_c)$
 $\dot{\hat{x}} = \bar{A} \cdot \hat{x} + \bar{B} \cdot u + \bar{H} (y - \bar{C} \hat{x})$
 $\bar{A} - \bar{H} \bar{C} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{d}{m} & -\frac{1}{m} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} h_{11} \\ h_{12} \\ h_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -h_{11} & 1-h_{11} & 0 \\ -h_{12} & -\frac{d}{m}-h_{12} & -\frac{1}{m} \\ -h_2 & 0 & 0 \end{bmatrix}$
 よて、オブザーバの極は、
 $\det(\lambda - (\bar{A} - \bar{H} \bar{C})) = \begin{vmatrix} \lambda + h_{11} & 1-h_{11} & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{vmatrix} = \begin{vmatrix} \lambda + h_{11} & -1 & 0 \\ h_{12} & \lambda + \frac{d}{m} & \frac{1}{m} \\ h_2 & 0 & \lambda \end{vmatrix}$
 $= \lambda(\lambda + h_{11})(\lambda + \frac{d}{m}) - h_2 \frac{1}{m} - (-1) h_{12} \lambda = \lambda^3 + (h_{11} + \frac{d}{m}) \lambda^2 + (h_{11} \frac{d}{m} + h_{12}) \lambda - \frac{h_2}{m}$
 $m = 2.0, d = 6.0$ 極 $\lambda = -3, -4, -5$ より、
 $(\lambda + 3)(\lambda + 4)(\lambda + 5) = \lambda^3 + (3+4+5)\lambda^2 + (3 \cdot 4 + 4 \cdot 5 + 5 \cdot 3)\lambda + 3 \cdot 4 \cdot 5$
 $h_{11} + \frac{6.0}{2.0} = h_{11} + 3.0 = 12, \quad h_{11} \cdot \frac{6.0}{2.0} + h_{12} = \frac{-h_2}{2.0} = 60$
 $\therefore h_{11} = 9.0 = 9.0 \cdot 3.0 + h_{12} = 47, \quad \therefore h_{12} = 20$
 $\therefore h_2 = -120, \quad \bar{H} = \begin{bmatrix} 9.0 \\ 20 \\ -120 \end{bmatrix}$

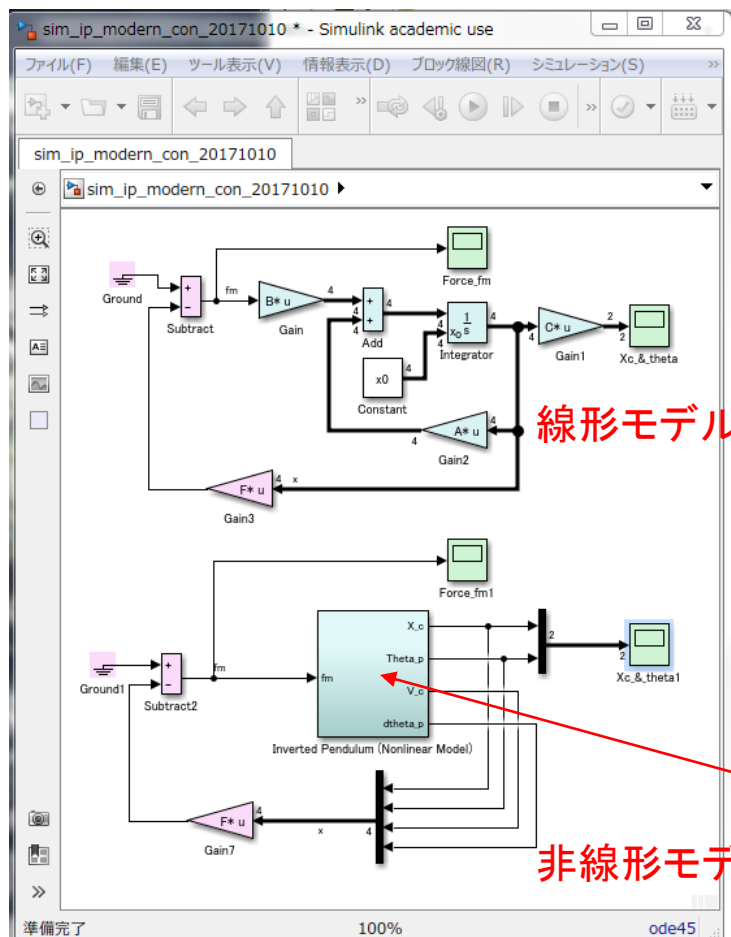


理論と実機の対応



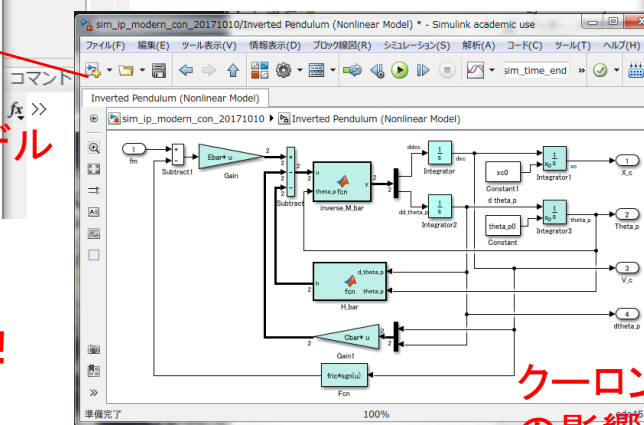
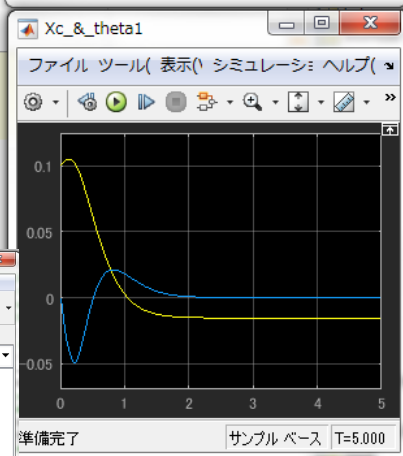
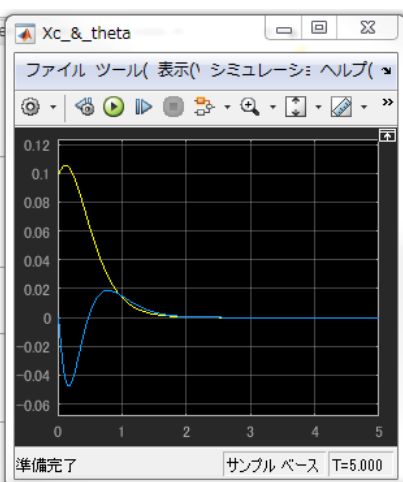
手計算させる: 制御系、外乱オブザーバ設計、... → シミュレーション → 実験

モーションコントロール(線形/非線形)



線形モデル

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
2 %% Inverted Pendulum Control program  
3 %% May.12.2016  
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
5 clear all;clc;  
6 %% Physical parameters  
7 g=9.8;r1=1.6e-2;r2=0.63e-2;m=0.62;J=1.7e-6;M=m+J/r2^2;  
8 mp=0.022;l=0.11;Io=mp*l^2/3;Cx=0.75;Ctheta=0; %% Pendulum  
9 fric=50e-3*9.8; %% クーロン摩擦  
10 Kt=0.023; %% Torque constant  
11 %% Complementary gains  
12 Gamp=1.0/0.10;Gpot=-12/5.0;Gf=r2/Gamp/Kt;Gz=2*pi*r1/Gpot;  
13 Genc=500*4;Ge=2*pi/Genc; %% Pendulum  
14 %% Initial value & Reference  
15 xc0=0.1;theta_p0=0;vc0=0;dtheta_p0=0;  
16 x0=[xc0;theta_p0;vc0;dtheta_p0];  
17 ts=0.001;St=0; %% 実験用時間データ  
18 %% State Space Model  
19 Mbar=[M*mp,mp*l;mp*l,Io+mp*l^2];invMbar=inv(Mbar);  
20 Cbar=diag([Cx,Ctheta]);Kbar=[0,0,0,-mp*l*g];Ebar=[1;0];  
21 A=[zeros(2,2),eye(2);-invMbar*Kbar,-invMbar*Cbar];  
22 B=[zeros(2,1);invMbar*Ebar];C=[eye(2),zeros(2,2)];  
23 %% Control gains  
24 Q=diag([1e3,1e2,1e2,1e1]);R=1;F=lqr(A,B,Q,R)  
25 %% Simulation  
26 sim_time_end=5.0;sim('sim_ip_modern_con_20171010');
```



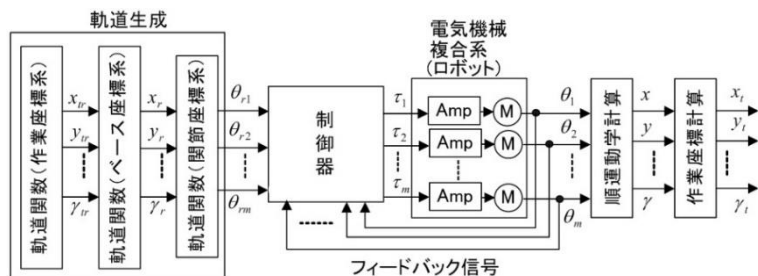
非線形モデル

線形モデルを仮定して制御系設計
⇒非線形モデルでも評価してみよう！

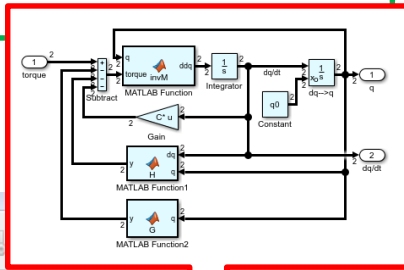
クーロン摩擦、非線形特性などの影響を調べる

ロボティクス (3年後期)

[2リンクマニピュレータの軌道制御]

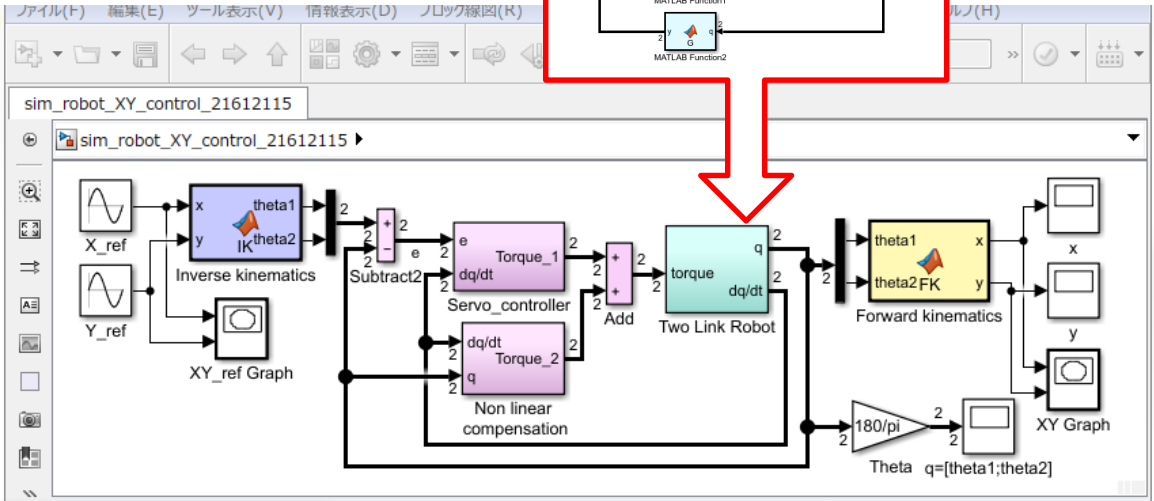


講義内容
 運動学: 回転行列、同次変換行列、...
 力学: ラグランジュの運動方程式、...
 制御: 多入力多出力、非線形、リアプノフ関数、...
 軌道計画: ...



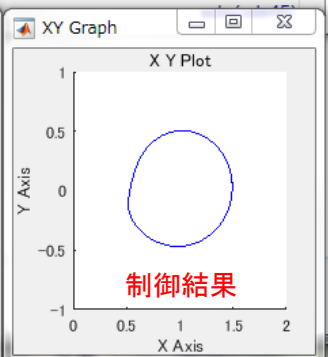
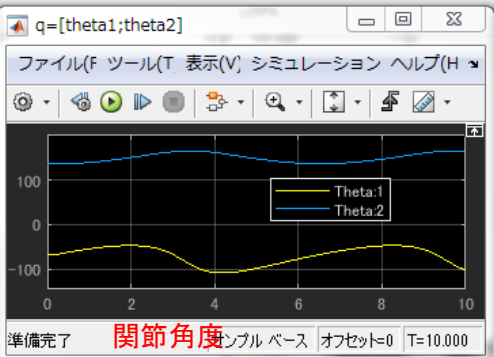
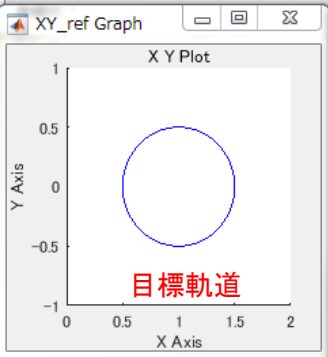
```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% Robot XY control for 2 Link Manipulator
3  %% Dec.15.2016 A.Shimada
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  clear all;clc;
6  %% Physical Parameters
7  global L1 L2
8  L1=2.0;L2=2.0; % Length of Links
9  r1=1.0;r2=1.0; % Distance between joint and COM
10 m1=10.0;m2=5.0;% mass of links
11 I1=m1*r1^2+I2=m2*r2^2;c1=5;c2=5;
12 C=diag([c1,c2]);% Co-efficient of viscous friction
13 g=9.8;
14
15 %% Controller
16 Kp1=3000;Kd1=500;Kp2=800;Kd2=200;
17 Kp=[Kp1,0;0,Kp2];Kv=[Kd1,0;0,Kd2];
18
19 %% Initial values & Reference
20 bias=1.0;radius=0.5;omega=1;x0=1.5;y0=0;q0=func_IK(x0,y0);
21 theta_1_ref=30*(pi/180);theta_2_ref=90*(pi/180);
22 step_time_1=2; step_time_2=4;
    
```



```

1 function q = IK(x,y)
2 %%codegen
3 global L1 L2
4 L=sqrt(x^2+y^2);
5 cos_alpha=(L1^2+L2^2-L^2)/(2*L1*L2); % 余弦定理
6 sin_alpha=sqrt(1-cos_alpha^2);
7 alpha=atan2(sin_alpha,cos_alpha);
8 cos_ep=(L^2+L1^2-L^2)/(2*L*L1); % 余弦定理
9 sin_ep=sqrt(1-cos_ep^2);
10 ep=atan2(sin_ep,cos_ep);
11 delta=atan2(y,x);
12 theta1=delta-ep;theta2=pi-alpha;q=[theta1;theta2];
    
```



モデル作成の作法 (良い例)

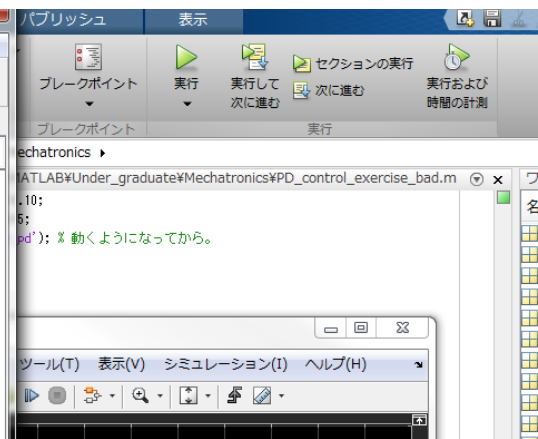
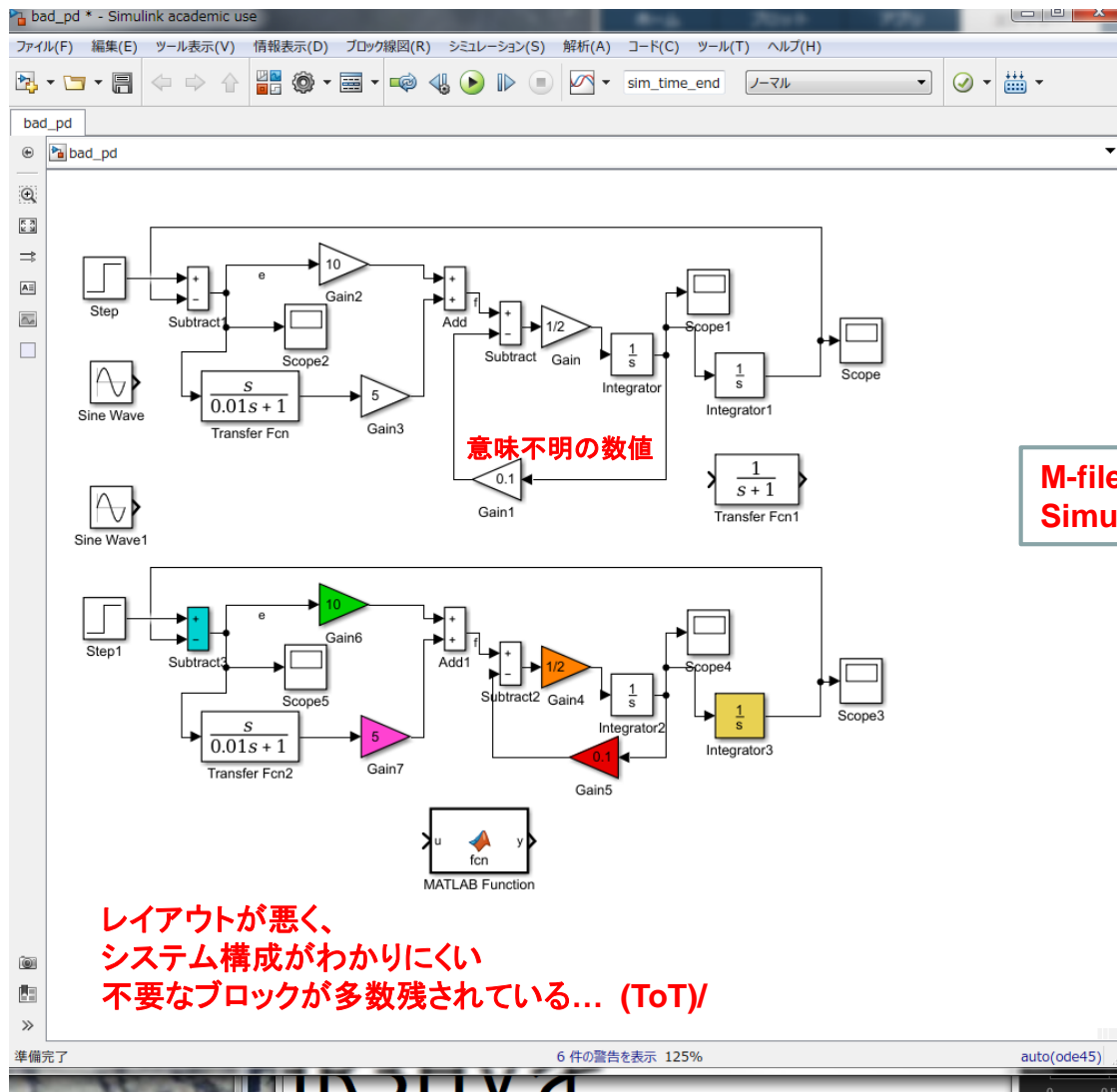
The image displays a Simulink model and its corresponding MATLAB code. The Simulink model, titled 'sim_pd_control_exercise', shows a control loop. A step input is fed into a summing junction. The error signal e is processed by a transfer function block $\frac{s}{0.01s+1}$ (Kd) and a gain block (Gain3). The output is summed with a feedback signal from a position sensor (Gain1) and a constant block (Xc_0). The resulting signal f is summed with a feedforward path (Gain2) and then passed through a gain block (Gain), an integrator ($\frac{1}{s}$), and a mass block ($\frac{1}{m}$). The final output is the position X_c .

The MATLAB code in the editor window defines the physical parameters, controller design, and simulation parameters:

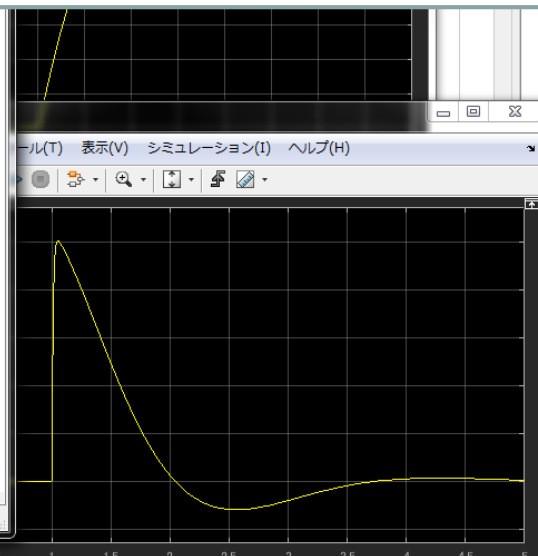
```
1 ***** PD Control Exercise *****
2 % PD Control Exercise
3 % Oct.10.2017 A.Shimada
4 *****
5 clear all;clc;
6 %% Physical parameters
7 % 位置:Xc[m], 速度:Vc[m/s], 力:f[N]
8 m=2.0; % 質量[kg]
9 c=0.10; % 粘性摩擦係数 [N/(m/s)]
10
11 %% Controller design
12 % G(s)=(Kd*s+Kp)/(m*s^2+(c+Kd)s+Kp)
13 Kp=10;Kd=5;
14
15 %% Initial values
16 Xc_0=0.0; % 初期位置[m]
17 Xc_ref=1.0; % 目標位置[m]
18
19 %% Simulation
20 sim_time_end=5.0;
21 sim('sim_pd_control_exercise'); % 動くようになってから。
22 %
23 time=Xc_data.time;
24 Xc_ref=Xc_data.signals(1).values;
25 Xc=Xc_data.signals(2).values;
26 plot(time,Xc_ref,'-b',time,Xc,'r','LineWidth',1.2);grid on;
27 set(gcf,'color','w'); %ウィンドウを白に変更
28 set(gca,'Fontname','Times New Roman','FontSize',14); %文字設定
29 xlabel('Time [s]','FontSize',16); %ラベル設定
30 ylabel('Position[m]','FontSize',16); %
31 axis([0,sim_time_end,0,1.5]); %x,y軸max&min
32 legend('Position Reference: Xc_ref','Position: Xc','Location','Southeast')
33
```

可能な限りの情報が見えている。変数値はm-fileで定義。可視化にも気配り。

モデル作成の作法 (悪い例)

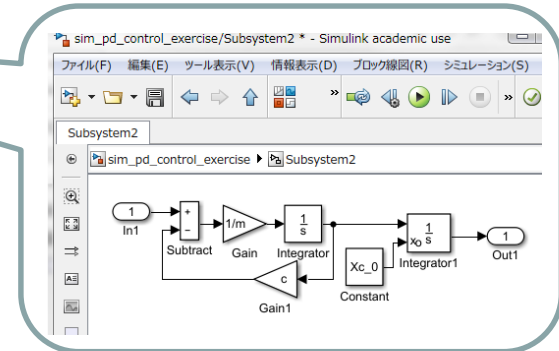
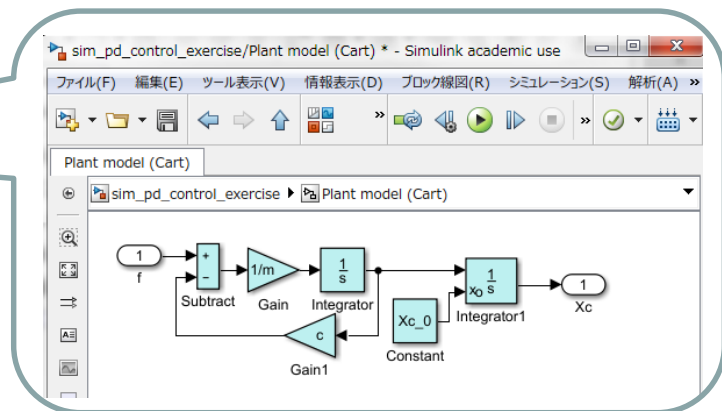
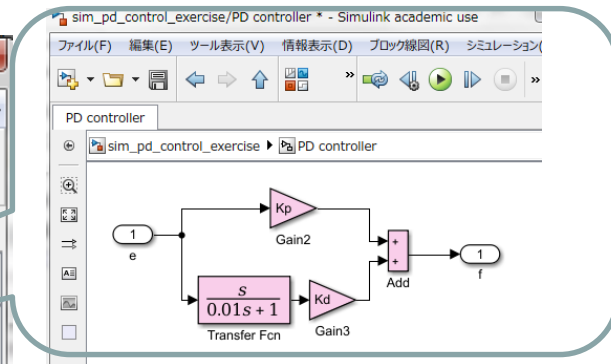
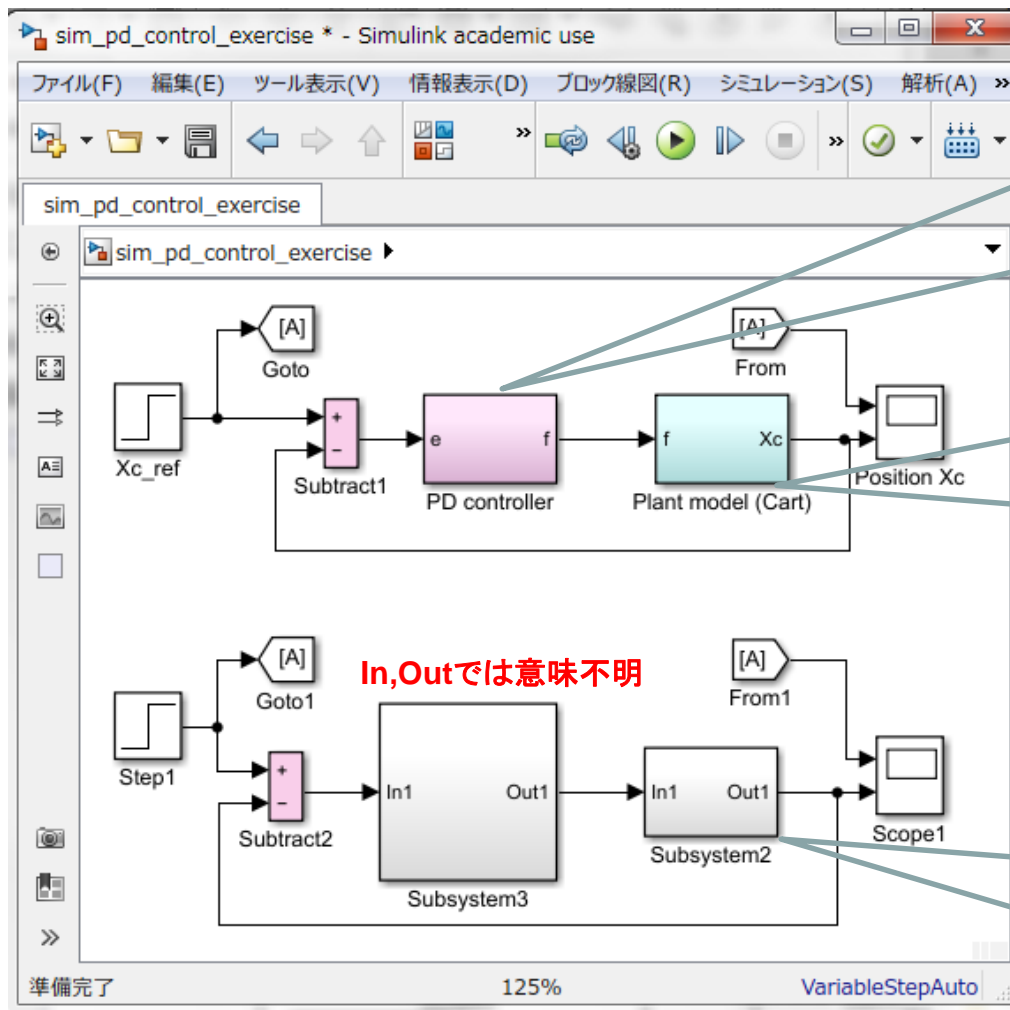


M-fileを開ける度にSimulinkモデルが消え、
Simulinkモデルをクリックする度にm-fileが消える



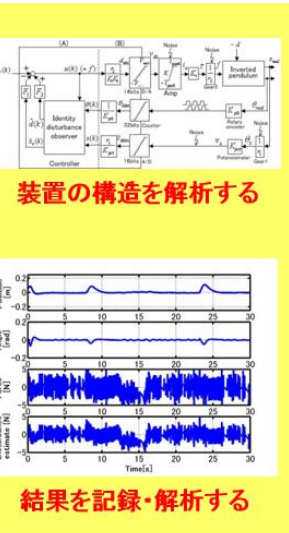
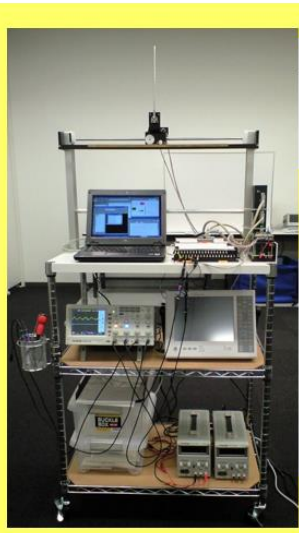
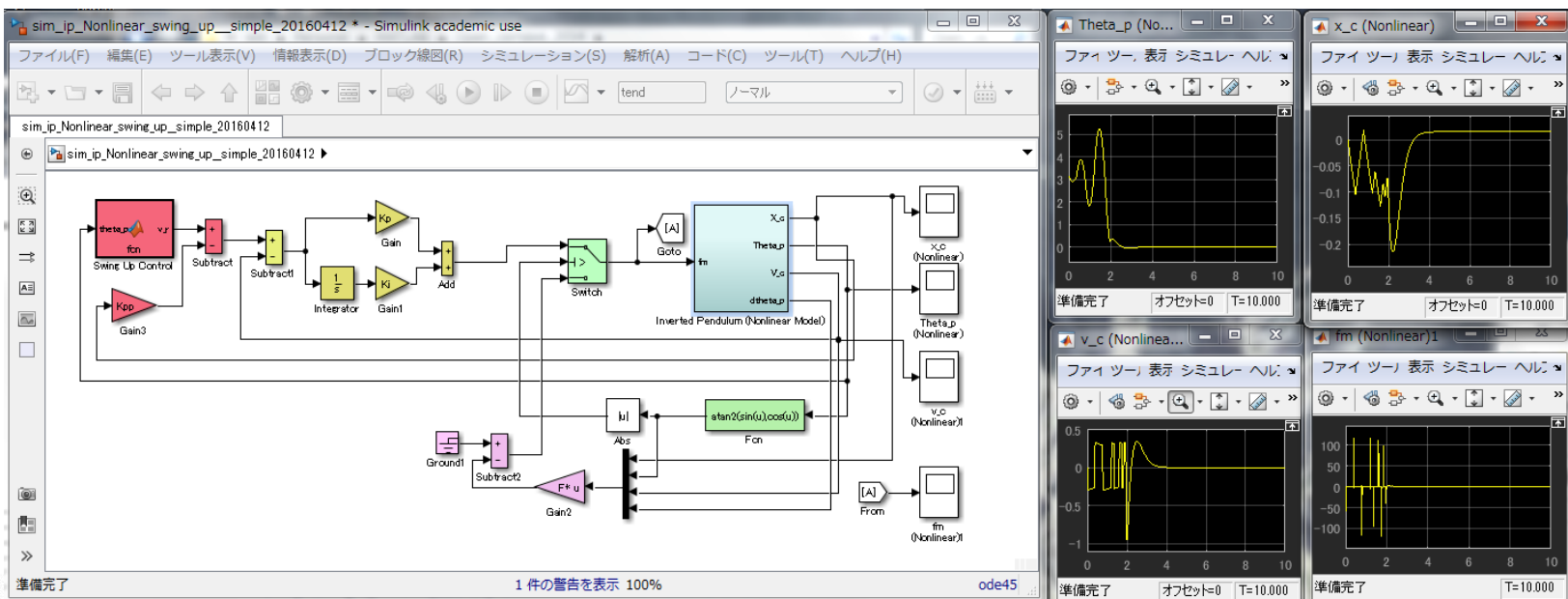
無駄な空間 & 意味不明の数値 & 意味不明のScope & 意味不明の色使い

モデル作成の作法 (比較)



意味不明のブロック & 意味不明の入出力があり、
可読性が悪い。作者しか理解できない。(ToT)/

プロジェクト演習：MATLAB/Simulinkを用いた実習



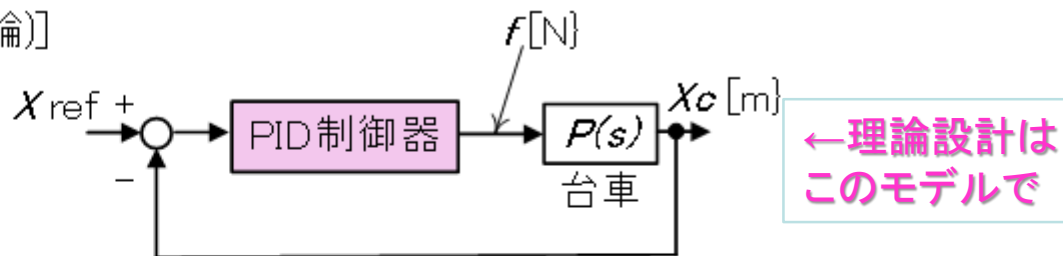
```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Inverted Pendulum Swing Up Control
3 % May.10.2016 A.Shinada
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 clear all;clc;
6 %% Physical parameters
7 g=9.8;r1=1.6e-2;r2=0.03e-2;m=0.62;J=1.7e-6;M=m+J/r2^2;
8 mp=0.022;l=0.11;Ic=mp*l^2/3;
9 Cx=0.75;Ctheta=0; % Pendulum
10 fric=50e-3*49.8; % クーロン摩擦
11
12 %% State Space Model
13 Mbar=[Mmp,mpl;mpl,Ic+mp*l^2]; % A, B行列を作るために線形化したMbar
14 invMbar=inv(Mbar);Cbar=diag([Cx,Ctheta]);
15 Kbar=[0,0,-mp*Icg]; % A行列を作るために線形化したKbar
16 Ebar=[1;0];
17 A=[zeros(2,2),eye(2);-invMbar*Kbar,-invMbar*Cbar];
18 B=[zeros(2,1);invMbar*Ebar];C=[eye(2),zeros(2,2)];
19
20 %% State Feedback Control gains
21 Q=diag([1e3,1e2,1e2,1e1]);R=1;F=diag(A,B,0,R)
22
23 %% Velocity and Swing Up Control gains
24 Kp=200;Kl=10.0;
25 Kz=0.3;Kp=0.35;
26
27 %% Threshold
28 threshold=pi/8;
29
30 %% Initial value & Time specification
31 xc0=0;theta_p0=pi; % Initial datas of xc and theta_p
32 vc0=0;dtheta_p0=0; % Initial datas of vc and dtheta_p/dt
33 x0=[xc0;theta_p0;vc0;dtheta_p0];
34 ts=0.001;St=0; % 実験用時間データ % Initial values
35 tend=10.0 % シミュレーション時間
    
```



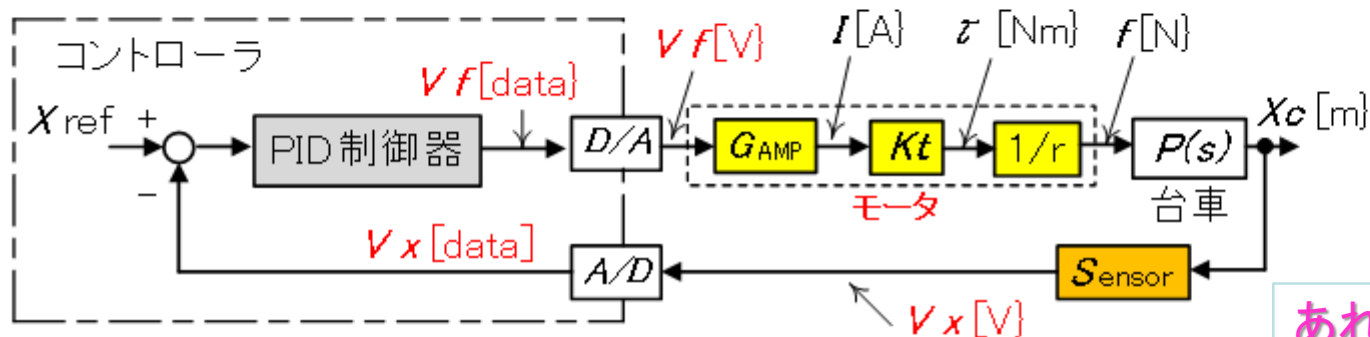
理論と実装のギャップ

[制御系設計(理論)]



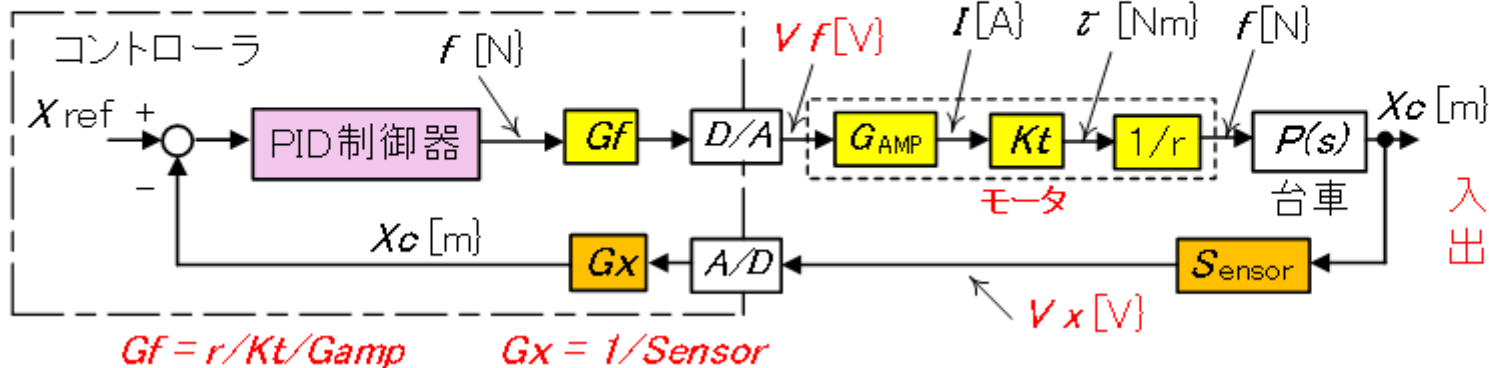
入力: 力 $f[N]$
出力: 位置 $x_d[m]$

[実装設計 1]



入力: 電圧 $Vf[V]$
出力: 電圧 $Vx[V]$

[実装設計 2]



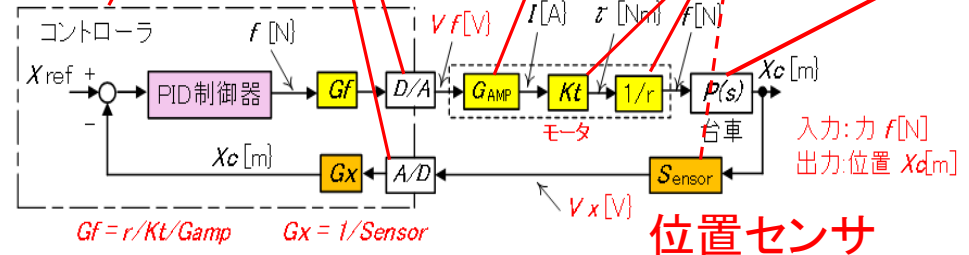
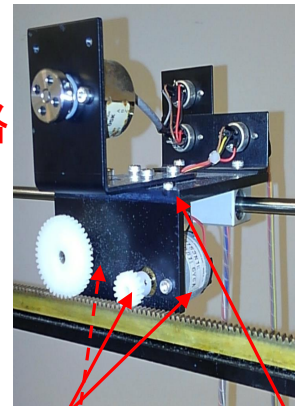
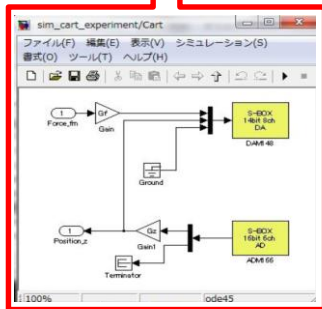
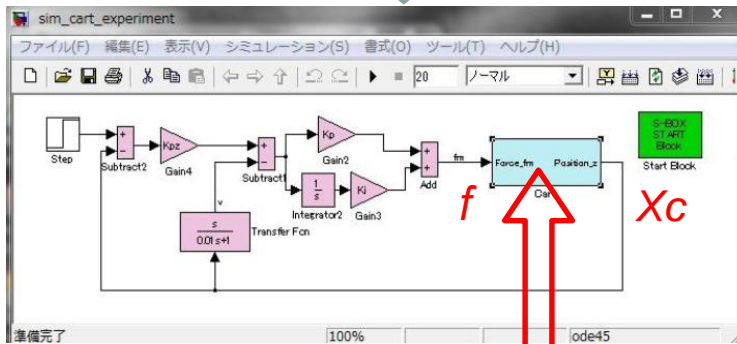
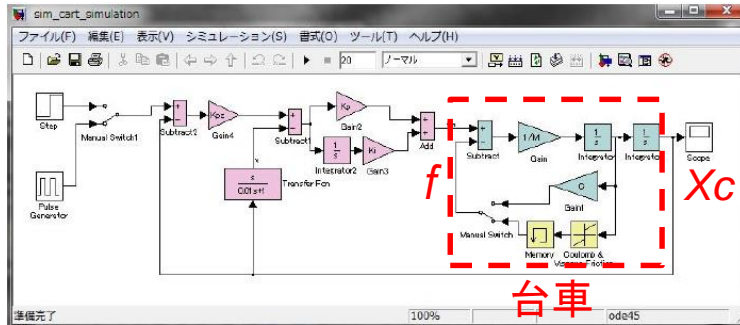
入力: 力 $f[N]$
出力: 位置 $x_d[m]$

こうすれば (等価的に) 入出力が合う!!

シミュレーションと実機の対応

課題: 力 f を入力、位置 X_c を出力として設計したのに、
実機の入力と出力は電圧だった...???

実機とのギャップをどう埋める?



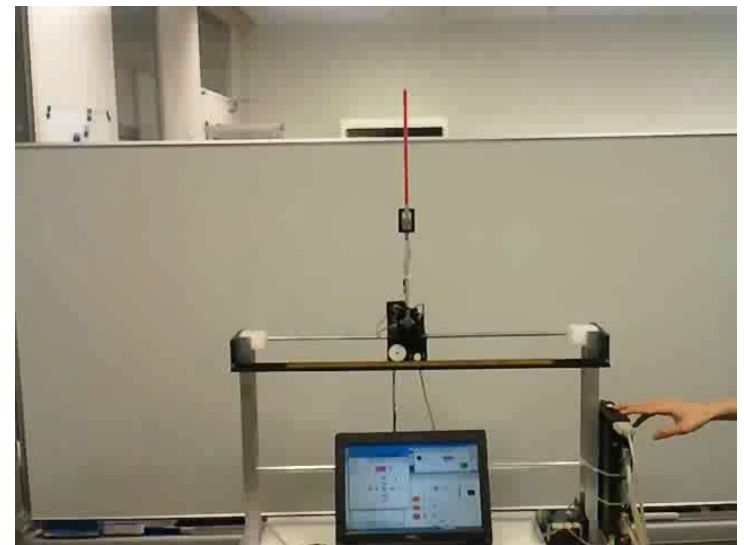
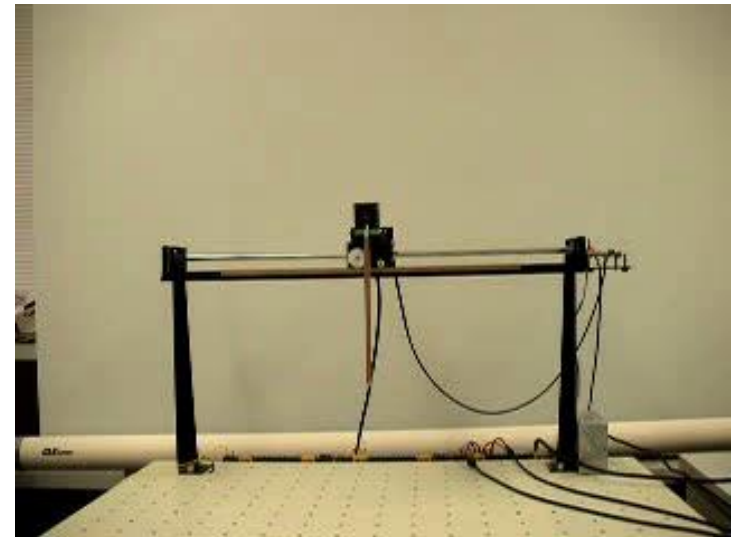
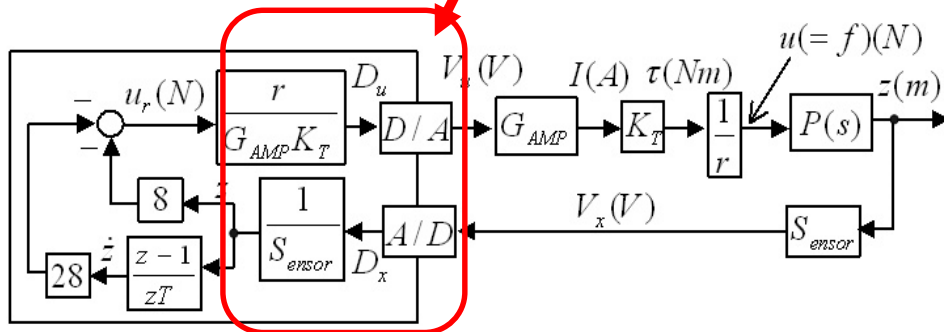
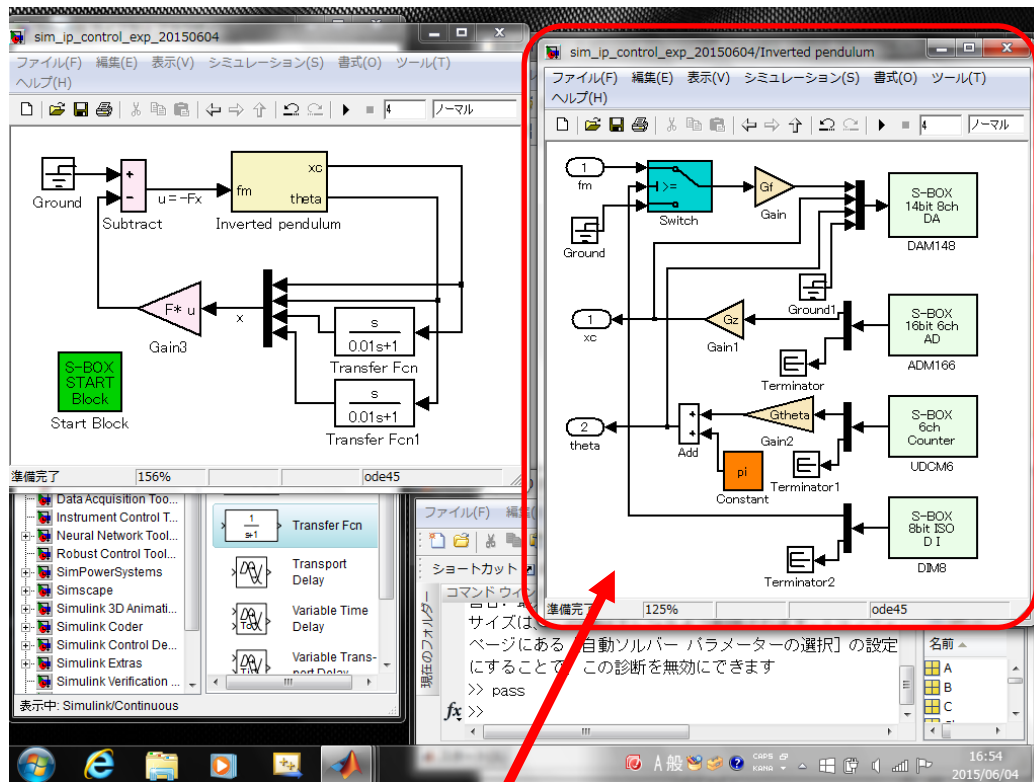
台車

AMP回路

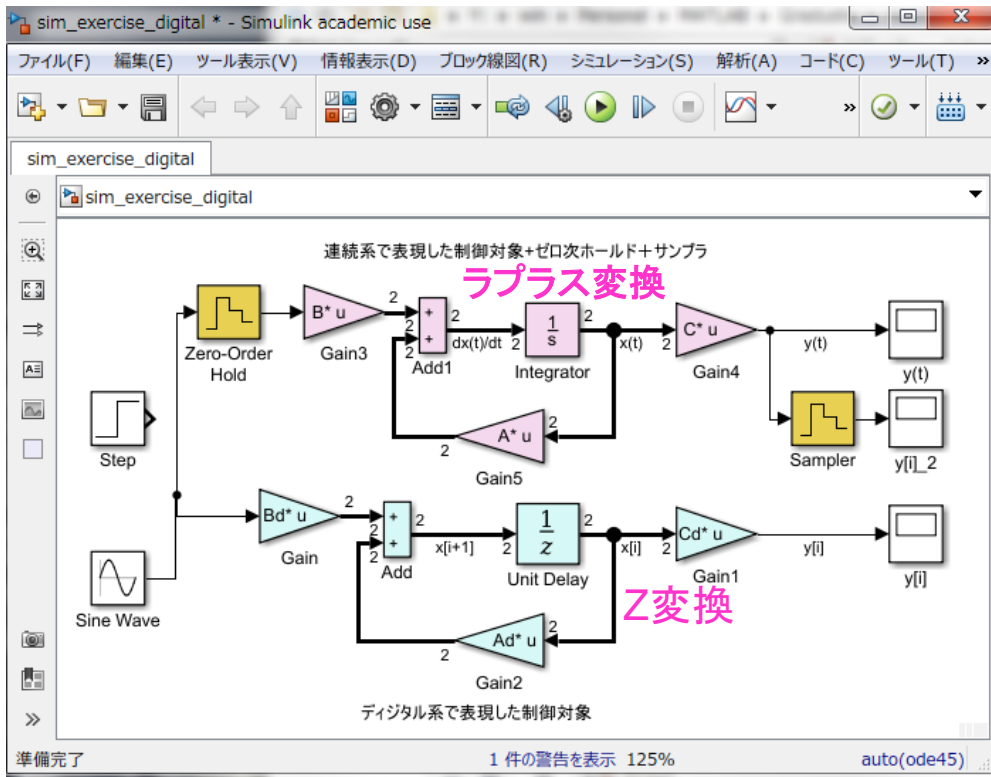
台車

位置センサ

倒立振子の場合も同様に

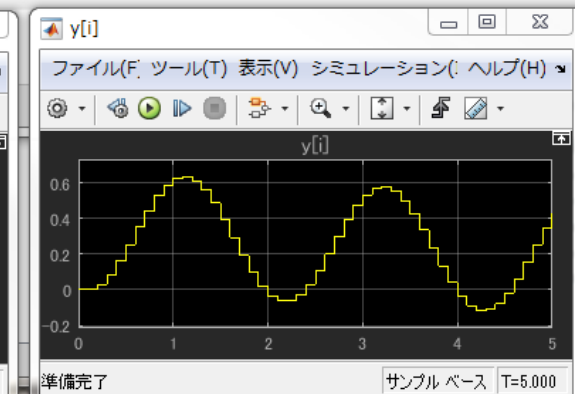
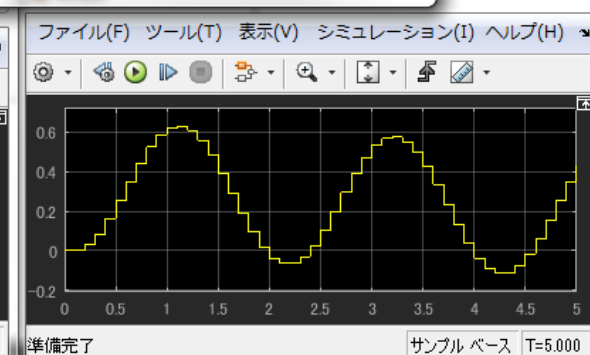
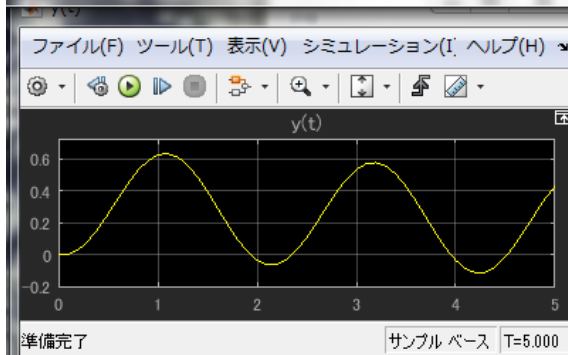


連続系とデジタル系 (大学院)



```
control_2017 ►
Personal\MATLAB\Graduate\Mechanics_control_2017\exercise_digital.m
1 *****
2 % デジタル制御基礎
3 % Oct.20.2017 A.Shimada
4 *****
5 clear all;clc;
6 %% Physical parameters
7 m=1; % 質量 [kg]
8 c=0.1; % 粘性摩擦係数 [N/(m/s)]
9
10 %% 状態方程式、出力方程式
11 % x_dot(t)=Ax(t)+Bu(t), y(t)=Cx(t)+Du(t),
12 % 位置xc(t), 速度vc(t), 力f(t), x(t)=[xc(t),vc(t)]', u(t)=f(t), y(t)=xc(t)
13 A=[0,1;0,-c/m];B=[0;1/m];C=[0,1];D=0;
14 sys_cont=ss(A,B,C,D); %% ss命令でワンパックにまとめておく
15
16 %% 離散化
17 T=0.1; % 制御周期 1ms
18 sys_digital=c2d(sys_cont,T); % c2dはcontinuous to digitalの変換関数
19 % Ad=Exp(AT),Bd=...を計算してくれる。
20 [Ad,Bd,Cd,Dd]=ssdata(sys_digital); % その中身はAd,Bd,Cd,Dd
21
22 %% 初期値
23 xc_0=0.0; vc_0=0.0; x0=[xc_0;vc_0];
24
25 %%
26 disp('end')
27
```

離散系を理解するには...

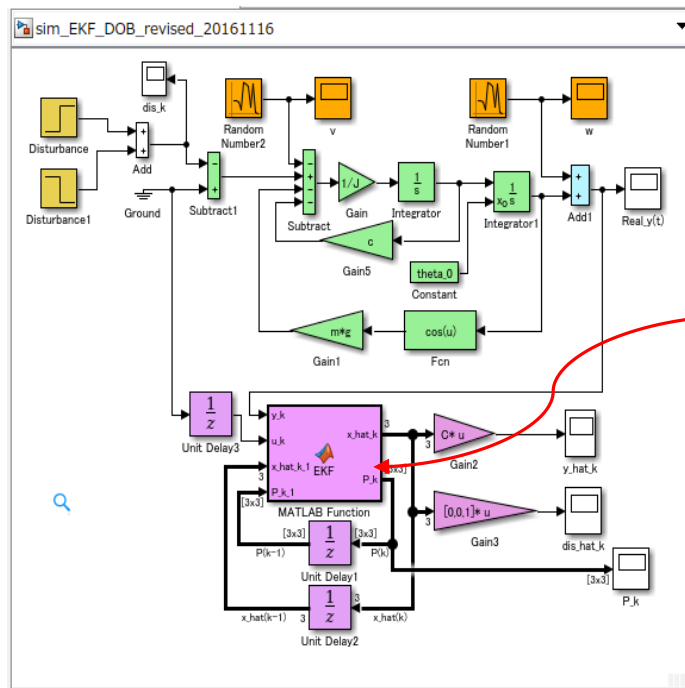
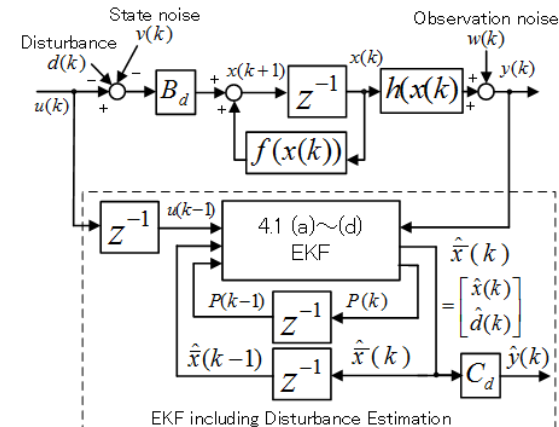


拡張カルマンフィルタの実装例(大学院)

- (a) 事前状態推定 $\hat{x}^-(k) = f(\hat{x}(k-1)) + B_d \cdot \tau(k-1)$
- (b) 事前誤差共分散行列

$$P^-(k) = A(k-1)P(k-1)A^T(k-1) + Qbb^T$$
- (c) カルマンゲイン

$$g(k) = P^-(k)C(k)(C^T(k)P^-(k)C(k) + R)^{-1}$$
- (d) 状態推定値 $\hat{x}(k) = \hat{x}^- + g(k)\{y(k) - h(\hat{x}^-(k))\}$
 事後誤差共分散行列 $P(k) = \{I - g(k)C^T(k)\}P^-(k)$

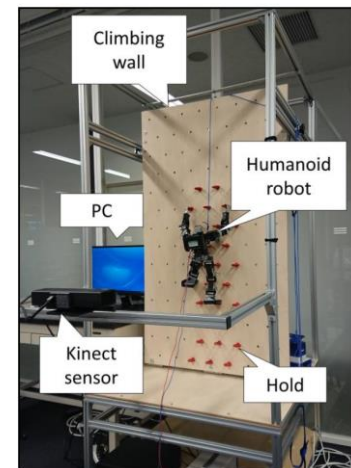
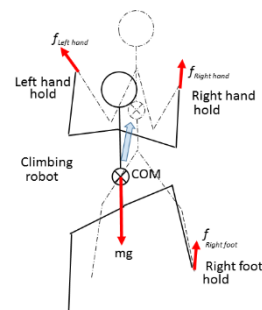
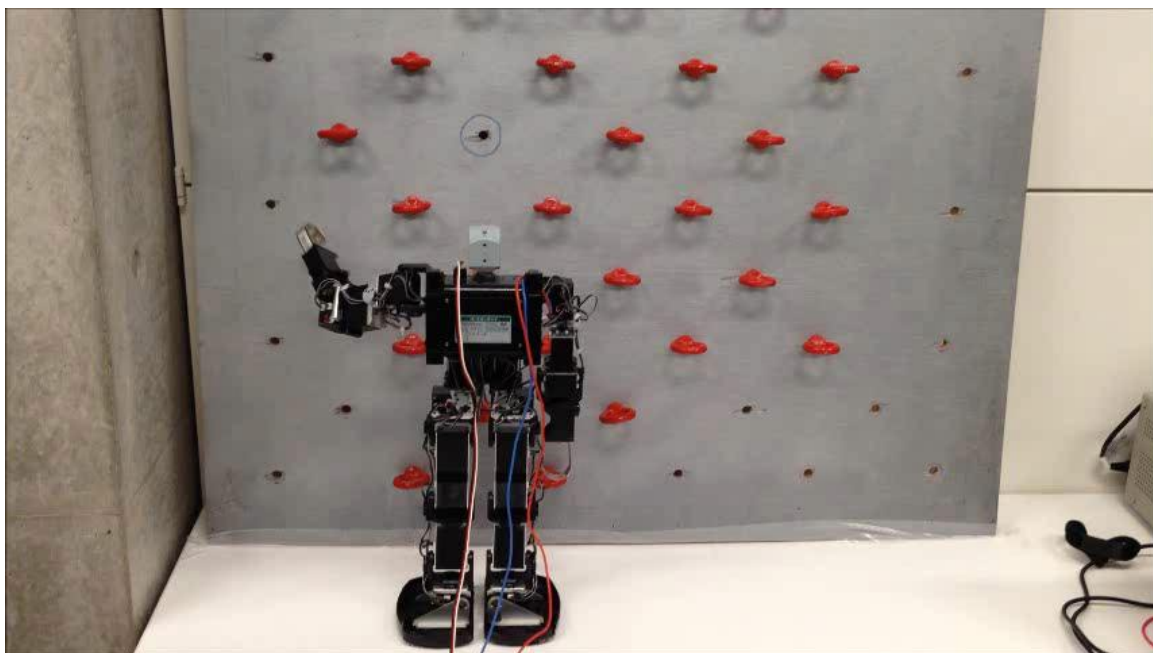


```

1 function [x_hat_k,P_k] = EKF(y_k,u_k,x_hat_k_1,P_k_1,Qk1,Rk1,T,m,g, J,Bv,c)
2 %%codegen
3
4 %% Priori Estimation
5 theta_hat_k_1=x_hat_k_1(1);
6 dtheta_hat_k_1=x_hat_k_1(2);
7 tau_dis_hat_k_1=x_hat_k_1(3);
8
9 theta_hat_minus_k=theta_hat_k_1+T*dtheta_hat_k_1; % a priori state estimate 事前状態推定値
10 dtheta_hat_minus_k=dtheta_hat_k_1-T/J*m*cos(theta_hat_k_1)-T*c/J*dtheta_hat_k_1-T/J*tau_dis_hat_k_1;
11 tau_dis_hat_minus_k=tau_dis_hat_k_1;
12 x_hat_minus_k=[theta_hat_minus_k;dtheta_hat_minus_k;tau_dis_hat_minus_k];
13
14 %%
15 A_k_1=[1,T,0;
16        -T/J*m*cos(theta_hat_k_1),1-T*c/J,-T/J;
17        0,0,1];
18 C_k_t=[1,0,0];C_k=C_k_t';
19 P_minus_k=A_k_1*P_k_1*A_k_1'+Qk1*Bv*Bv'; % a priori covariance matrix 事前共分散行列
20
21 %% Kalman Gain matrix
22 G_k=P_minus_k*C_k_t/(C_k_t*P_minus_k*C_k_t+Rk1); % カルマンゲイン行列
23
24 %% Posteriori Estimation
25 x_hat_k=x_hat_minus_k+G_k*(y_k-theta_hat_k_1);
26 % State estimate 状態推定値
27 P_k=(eye(3)-G_k*C_k_t)*P_minus_k; % a posteriori covariance matrix
    % 事後共分散行列
    
```

島田明: 外乱推定カルマンフィルタ設計, MEC-16-027, pp. 1-6, 電気学会メカトロニクス制御研究会, 2016-12-11

研究室: クライミングロボットと最適化問題

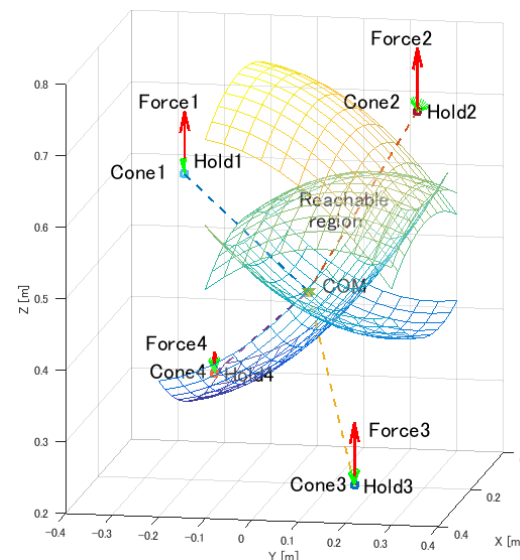


minimize $f(X)$ subject to $g_i(X) \leq 0, i = 1, 2, \dots, m$

```

50 %% Linear equation constraints 線形等式拘束条件
51 Aeq=[cos(theta1)*cos(phi1),cos(theta2)*cos(phi2),cos(theta3)*cos(phi3),cos(theta4)*cos(phi4),0,0,0;
52      cos(theta1)*sin(phi1),cos(theta2)*sin(phi2),cos(theta3)*sin(phi3),cos(theta4)*sin(phi4),0,0,0;
53      sin(theta1),sin(theta2),sin(theta3),sin(theta4),0,0,0];
54 beq=[0;0;m*g];
55
56 %% Initial values & min-max 初期値と最小値最大値
57 f0=[1;1;1];f_min=[0;0;0];f_max=[50;50;90;90];
58 x0=(x01+x02+x03+x04)/4;y0=(y01+y02+y03+y04)/4;z0=(z01+z02+z03+z04)/4;
59 xyz_0=[x0;y0;z0];xyz_min=[-0.1;-0.25;0.0];xyz_max=[0.20;0.25;0.7];
60 xx0=[f0;xyz_0];xx_min=[f_min;xyz_min];xx_max=[f_max;xyz_max];
61
62 %% Run fmincon 非線形計画法の実施
63 [xx,fval]=fmincon(@climbing_fun3_4limbs,xx0,[],[],Aeq,beq,xx_min,xx_max,@climbing_nonlcon3_4limbs);
64 f=xx(1:4);x=xx(5);y=xx(6);z=xx(7); f1_f2_f3_f4_x_y_z=[f',x,y,z] % Force and x,y,z
65

```



島田明: クライミングロボットの平衡状態を維持するための最適化法, 第59回自動制御連合講演会, ThC7-5, pp.536-541北九州国際会議場, 2016-11-10

使用製品例

- 基本環境 : MATLAB、Simulink
- 制御設計 : Control System Toolbox
- 数式処理 : Symbolic Math Toolbox、
- コード生成 : MATLAB Coder、Simulink Coder

まとめ

- 芝浦工業大学 デザイン工学部の組織・教育について紹介した
- **手書き計算**の重要性を示すと共に、**シミュレーション**ソフトウェアを駆使した**実践教育**の重要性を示した
- ロボット教育・制御教育を中心としたカリキュラムとMATLABを活かした内容・特徴を紹介した
- **理論と実システム**の間のギャップの埋め方について紹介した
- 最近の研究上のトピックス(**クライミングロボット**)を紹介した