

目次

1. はじめに
2. 行列の演算
3. 時間応答の数値計算
4. システムの特性解析
5. 状態フィードバック制御の設計
6. オブザーバの設計と動的制御器
7. サーボ系の設計
8. 最適制御系の設計
9. LQG 理論によるサーボ系の設計

1. はじめに

現代制御理論では、状態方程式に基づいて制御系設計が行われる。現代制御理論で用いられる計算としては、行列計算の加減乗除、固有値、特異値、線形代数方程式の解法などが挙げられる。Scilab ではこれらを高速に行える。本稿では、現代制御理論で用いられる基本的な計算問題を取り上げて、Scilab 関数を用いたプログラムと実行結果を示す。

参考文献：感度関数と相補感度関数によるフィードバック制御系の特性解析法は、[1]の11章と12章に記している。本稿の説明は[2]に掲載している「Scilab による古典制御の学習」を理解されていることを前提にしている。

[1] 佐伯正美, 制御工学 (古典制御からロバスト制御へ), 朝倉書店, 2013

[2] http://home.hiroshima-u.ac.jp/saeki/index_ja.html

[3] 上坂吉則, MATLAB+Scilab プログラミング事典, ソフトバンククリエイティブ株式会社

広島大学 佐伯正美

2. 行列の演算

つぎの計算を行う.

行列 A, ベクトル B, ベクトル C を入力し, 基本的な行列演算の例を示せ.

プログラム 'MCmatrix_calculation.sce'

//行列の入力

A=[1 4;3 2]

B=[1 4]'

C=[2 0]'

//加減乗

D1=B+C

D2=B-C

D3=A*B

//転置

D4=A'

//行列式

s1=det(A)

//逆行列

D5=inv(A)

//固有値と固有ベクトル

[T, eigA]=spec(A)

A2=T*eigA*inv(T)//確認

//特異値と特異ベクトル

[U, S, V]=svd(A)

A3=U*S*V' //確認

3. 時間応答の数値計算

システムが $P(s) = C(sI - A)^{-1}B + D$ で与えられるとき, ステップ応答, インパルス応答, 指定した初期値と入力 $u(t)$ に対する応答 $y(t)$ を求め, 図示せよ. ただし,

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [2 \ 0], D = 0$$

```

プログラム 'MCtransient_response.sce'
//過渡応答の計算 プログラム
//システムの係数
A=[0 1;-1 -1]
B=[0 1]'
C=[2 0]
D=0
sysP=syslin('c',A,B,C,D)
//ステップ応答とインパルス応答 初期値=0 とする.
t=0:0.01:10;
y1=csim('step',t,sysP);
y2=csim('impulse',t,sysP);
clf();
scf(0);
plot(t,y1,'r',t,y2,'b')
xlabel("Step and impulse responses","time(s)","y1(t), y2(t)")
//初期値 x0, 入力 u(t)に対する応答
x0=[1 0]'
t=0:0.01:10;
u=abs(sin(t));
y3=csim(u,t,sysP,x0);
scf(1);
plot(t,u,'r-',t,y3,'b')
xlabel("Response with x0 and u(t)","time(s)","y(t), u(t)")

```

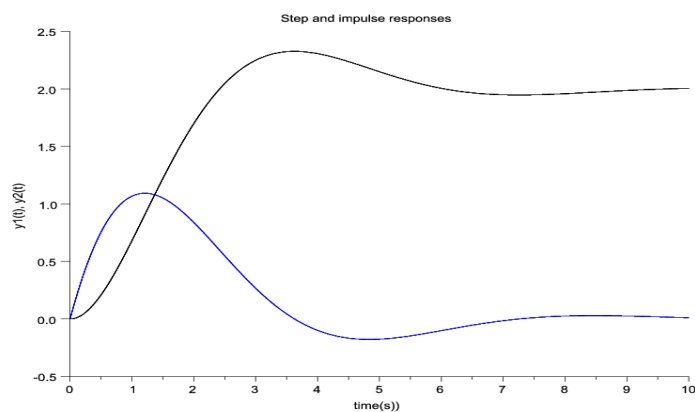


図1 ステップ応答とインパルス応答

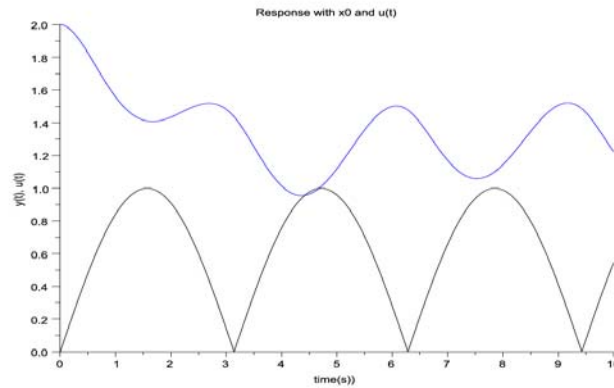


図2 初期値 x_0 と入力 $u(t)$ に対する応答

4. システムの特性解析

システムが $P(s) = C(sI - A)^{-1}B + D$ で与えられるとき，伝達関数，極，ゼロ点，可制御性，可観測性，周波数特性を調べよ．ただし，

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1 \quad 0], D = 0$$

プログラム ‘MCsystem_analysis.sce’

// システムの特性解析

exec('bode2.sci',-1)//自作の関数 bode2.sci を使うのに必要

//システムの状態方程式

A=[0 1;-1 -1]

B=[0 1]'

C=[1 0]

D=0

sysP=syslin('c',A,B,C,D)

//極、ゼロ点、伝達関数

eigP=spec(A)

zeroP=trzeros(sysP)

P=ss2tf(sysP)

//可制御性

Uc=cont_mat(A,B)//可制御性行列 Uc

```

n=contr(A,B) //n は可制御部分空間の次元
[na,na]=size(A)
if na==n then
    disp('可制御')
else
    disp('不可制御',na)
end

//可観測性
Uo=obsv_mat(A,C)//可観測性行列 Uo
nbar=unobs(A,C) //nbar は不可観測部分空間の次元
[na,na]=size(A)
if nbar==0 then
    disp('可観測')
else
    disp('不可観測',na)
end

//周波数特性
nw=200;
w=logspace(-2,1,nw);//単位は rad/s
Pw=freq(A,B,C,D,w*%i);
rePw=real(Pw);
imPw=imag(Pw);
gPw=abs(Pw);
//ベクトル軌跡
scf(0);
plot2d(rePw,imPw,rect=[-0.5, -1.2, 1.1, 0.1]);
plot2d(rePw,-imPw);
xlabel('vector locus','real','imag')
xgrid();
//
scf(1);
nyquist(sysP)// パラメータ周波数は Hz

//ゲイン図

```

```

scf(2);
plot2d('ln',w,20*log10(gPw));
xlabel('Gain plot','frequency[rad/s]','magnitude[dB]')
xgrid();
//ボード線図
scf(3);
bode2(P,w) //横軸の周波数の単位は rad/s

```

5. 状態フィードバック制御の設計 (極配置)

システムが $P(s) = C(sI - A)^{-1}B + D$ で与えられるとき、極を $\lambda_1, \lambda_2, \dots, \lambda_n$ に配置する状態フィードバックゲイン F を求めよ。ただし、

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1 \quad 0], D = 0$$

$$\lambda_1 = -1 + 2j, \lambda_2 = -1 - 2j$$

```

プログラム 'MCstate_feedback_design.sce'
// 状態フィードバックゲインの設計 (極配置)
//システムの状態方程式
A=[0 1; 0 -1]
B=[0 1]'
C=[1 0]
D=0
sysP=syslin('c',A,B,C,D)
//状態フィードバック系の極を指定
poles=[-1+2*i,-1-2*i]

```

```

//プラントの極
opoles=spec(A);

```

```

//可制御性のテスト
n=contr(A,B) //n は可制御部分空間の次元
[na,na]=size(A);
if na==n then

```

```

disp('可制御');
else
disp('不可制御',na);
break;
end
//状態フィードバックゲインの決定
F=ppol(A,B,poles)
cpoles=spec(A-B*F) //極配置の確認

```

6. オブザーバの設計と併合系の特性解析

システムが $P(s) = C(sI - A)^{-1}B + D$ で与えられるとき、オブザーバの極を $\lambda_1, \lambda_2, \dots, \lambda_n$ に配置するフルオーダーオブザーバを設計せよ。オブザーバゲインは K とする。状態フィードバック則 $u = -Fx$ とオブザーバから構成される動的制御器 $H(s)$ を求めよ。

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1 \ 0], D = 0$$

$$\lambda_1 = -2, \lambda_2 = -3$$

$$F = [5 \ 1]$$

```

プログラム 'MCobserver_design.sce'
// オブザーバの設計 (極配置)
//システムの状態方程式
A=[0 1; 0 -1];
B=[0 1]';
C=[1 0];
D=0;
sysP=syslin('c',A,B,C,D)
F=[5 1]//状態フィードバックゲインが与えられるとする
//オブザーバの極を指定
observer_poles=[-2,-3]
//可観測性のテスト (可制御性の双対)
n=contr(A',C) //n は可制御部分空間の次元
[na,na]=size(A);

```

```

if na==n then
    disp('可観測');
else
    disp('不可観測',na);
    break;
end
//オブザーバゲインの決定 (状態フィードバックの双対)
Ktemp=ppol(A',C',observer_poles);
K=Ktemp'
check_poles=spec(A-K*C)
// 動的制御器 H(s)= (Ah,Bh,Ch,Dh)の状態方程式
Ah=A-B*F-K*C;Bh=K;Ch=F;Dh=0;
//制御器の伝達関数
sysH=syslin('c',Ah,Bh,Ch,Dh);
H=ss2tf(sysH)

```

7. サーボ系の設計

プラント $P(s) = C_p (sI - A_p)^{-1} B_p$ に対して、ステップ関数に対して定常偏差がゼロとなる動的制御器 $H(s)$ を設計せよ。さらに、構成した系の目標値応答と外乱応答をステップ関数に対して数値計算で求めよ。制御器の伝達関数を求め、感度関数 S と相補感度関数 T のゲイン特性を描け。ただし、プラントのモデルを次式で与える。

$$A_m = \begin{bmatrix} 0 & 1 \\ -1 & -0.3 \end{bmatrix}, B_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_m = [1 \quad 0]$$

制御対象がゼロ型であるので、積分器を追加した拡大系に対して状態フィードバックゲインを極配置法で設計し、制御対象の状態はフルオーダーオブザーバで推定する。これらを用いて動的制御器を構成する。

制御対象のモデルの状態方程式

$$\begin{aligned} \dot{x}_m &= A_m x_m + B_m u \\ y &= C_m x_m \end{aligned}$$

出力 y の積分の状態方程式

$$\begin{aligned} \dot{x}_I &= y \\ y_I &= x_I \end{aligned}$$

上式をまとめると拡大系の状態方程式が得られる.

$$\dot{x} = Ax + Bu$$

ここに,

$$x = \begin{bmatrix} x_m \\ x_l \end{bmatrix}, A = \begin{bmatrix} A_m & 0 \\ C_m & 0 \end{bmatrix}, B = \begin{bmatrix} B_m \\ 0 \end{bmatrix}$$

状態フィードバックゲイン $F = [F_x, F_l]$ は $A - BF$ の極配置で設計する. これにより制御則として

$$u = -Fx = -[F_x \quad F_l]x$$

が得られた. つぎにプラントの状態を推定するオブザーバゲイン K は $A_m - KC_m$ の極配置で設計する. よってオブザーバの状態方程式は次式で与えられる.

$$\dot{x}_m = A_m x_m + B_m u - K(C_m x_m - y)$$

外乱と目標値に対する制御量と制御入力 of 応答のシミュレーションを考える. 外乱はプラント入力に加わるとすると, プラントの状態方程式が $\dot{x}_p = A_p x_p + B_p(u + d)$ となる.

また, 目標値と出力の偏差信号を積分器に加えることで定常偏差をゼロにするので, 積分器の状態方程式が $\dot{x}_l = y - r$ となる. これらにオブザーバの状態方程式と状態フィードバックの代数式を組み合わせれば, 次式のフィードバック系の状態方程式が得られる.

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c w \\ z &= C_c x_c \end{aligned}$$

ここに,

$$\begin{aligned} x_c &= \begin{bmatrix} x_p \\ x_m \\ x_l \end{bmatrix}, A_c = \begin{bmatrix} A_p & -B_p F_x & -B_p F_l \\ KC_p & A_m - B_m F_x - KC_m & -B_m F_l \\ C_p & 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} B_p & 0 \\ 0 & 0 \\ 0 & -I \end{bmatrix} \\ z &= \begin{bmatrix} y \\ u \end{bmatrix}, w = \begin{bmatrix} d \\ r \end{bmatrix}, C_c = \begin{bmatrix} C_p & 0 & 0 \\ 0 & -F_x & -F_l \end{bmatrix} \end{aligned}$$

感度や相補感度の計算のために, 次式のフィードバック系を考える.

$$\begin{aligned} y &= P(s)u \\ u &= H(s)(r - y) \end{aligned}$$

このとき, $H(s)$ の状態方程式は次式となる.

$$\begin{aligned} \dot{x}_h &= A_h x_h + B_h y \\ u &= -C_h x_h \end{aligned}$$

ここに

$$x_h = \begin{bmatrix} x_m \\ x_I \end{bmatrix}, A_h = \begin{bmatrix} A_m - B_m F_x - K C_m & -B_m F_I \\ 0 & 0 \end{bmatrix}$$
$$B_h = \begin{bmatrix} K \\ I \end{bmatrix}, C_h = F$$

プログラム 'MCservo_feedback_design.sce'

// サーボ系の設計

// 極配置, オブザーバと拡大系に対する状態フィードバック制御

clear all

exec('bode2.sci',-1)

//プラントのモデルの状態方程式 (Ap,Bp,Cp,Dp)

Am=[0 1; -1 -0.3]

Bm=[0 1]'

Cm=[1 0]

Dm=0

/**/

//積分器を含む拡大系(A,B)

[ny,nx]=size(Cm)

[nx,nu]=size(Bm)

A=[Am zeros(nx,ny)

Cm zeros(ny,ny)]

B=[Bm

zeros(ny,nu)]

//状態フィードバック系の極を指定

poles=[-1+2*%i,-1-2*%i, -3]

//プラントの極

opoles=spec(A)

//可制御性のテスト

n=contr(A,B) //n は可制御部分空間の次元

[na,na]=size(A);

```

if na==n then
    disp('可制御');
else
    disp('不可制御',na);
    break;
end

//状態フィードバックゲインの決定
F=ppol(A,B,poles)
cpoles=spec(A-B*F)

//*****//
// オブザーバの設計 ; モデル(Am,Bm,Cm,Dm)に対する設計 (極配置)
//オブザーバの極を指定
observer_poles=[-2,-3]
//可観測性のテスト (可制御性の双対)
n=contr(Am',Cm')//n は可制御部分空間の次元
if nx==n then
    disp('可観測');
else
    disp('不可観測',nx);
    break;
end

//オブザーバゲインの決定 (状態フィードバックの双対)
Ktemp=ppol(Am',Cm',observer_poles);
K=Ktemp'
observer_poles=spec(Am-K*Cm)

//*****

//シミュレーション(外乱, 目標値応答)
//プラントの状態方程式 (Ap,Bp,Cp,Dp)
Ap=[0 1; -1 -0.3]
Bp=[0 1]'
Cp=[1 0]

```

Dp=0

//シミュレーションの閉ループ系の状態方程式(Ac,Bc,Cc,Dc)

Fx=F(1:nx);

FI=F(nx+1:nx+ny);

Ac=[Ap -Bp*Fx -Bp*FI

K*Cp Am-Bm*Fx-K*Cm -Bm*FI

Cp zeros(ny,nx) zeros(ny,ny)];

//reference input

Bcr=[zeros(nx,ny)

zeros(nx,ny)

-eye(ny)];

//disturbance input

Bcd=[Bp

zeros(nx,nu)

zeros(ny,nu)];

Cc=[Cp zeros(ny,nx) zeros(ny,ny)

zeros(nu,nx) -Fx -FI];

Dc=zeros(ny+nu,ny);

sys_closed_r=sslin('c',Ac,Bcr,Cc,Dc);

sys_closed_d=sslin('c',Ac,Bcd,Cc,Dc);

//

t=0:0.01:5;

yr=csim('step',t,sys_closed_r);

yd=csim('step',t,sys_closed_d);

clf();

scf(0);

plot(t,yd);

plot(t,yr);

xtitle("Step response","time(s)","yr(t), ys(td)")

xgrid

//制御器の伝達関数

Ah=[Am-Bm*Fx-K*Cm,-Bm*FI

zeros(ny,nx), zeros(ny,ny)];

```

Bh=[K
      eye(ny,ny)];
Ch=F;
Dh=zeros(nu,ny);
sysH=syslin('c',Ah,Bh,Ch,Dh);
//プラントの伝達関数
sysP=syslin('c',Ap,Bp,Cp,Dp);

//感度関数と相補感度関数
sysS=1/(1+sysP*sysH);
sysT=1-sysS;

//周波数特性
nw=200;
omega=logspace(-2,2,nw);
[As,Bs,Cs,Ds]=abcd(sysS);
[At,Bt,Ct,Dt]=abcd(1-sysS);
Sw=freq(As,Bs,Cs,Ds,omega*%i);
Tw=freq(At,Bt,Ct,Dt,omega*%i);
gSw=abs(Sw);
gTw=abs(Tw);
//ゲイン図
scf(1);
plot2d('ln',omega,20*log10(gSw));
plot2d('ln',omega,20*log10(gTw));
xgrid();

```

8. 最適制御系の設計

プラント $P(s) = C(sI - A)^{-1}B + D$ に対し、つぎの LQ 評価関数を最小にする制御器を設計せよ。

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

一巡伝達関数 $L = F(sI - A)^{-1}B$ のベクトル軌跡を描き、円条件を満たすことを確認せよ。入力重み $R = rI$ を $r=0.1, 1, 10$ に増加させるときの状態フィードバック系の極配置, および, 同様に入力から見た感度関数 S と相補感度関数 T のゲイン特性を描け。ただし, プラントのモデルを次式で与える。

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

次式のリカッチ方程式の正定対称行列の解を P とする。

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

最適解の状態フィードバックゲインは

$$F = R^{-1}B^T P$$

で与えられる。よって, フィードバック系は次式で表される。

$$\dot{x} = Ax + Bu$$

$$u = -Fx$$

プラントの入力で閉ループを開いた場合の一巡伝達関数は

$$L = F(sI - A)^{-1}B$$

で与えられる。1 入力系の場合には感度関数と相補感度関数は

$$S = \frac{1}{1+L}, T = 1 - S = \frac{L}{1+L}$$

で与えられる。フィードバック系の極は $A - BF$ の固有値で与えられる。

プログラム ‘MCLQcontrol_design.sce’

```
// 最適制御系の設計
```

```
// LQ 状態フィードバック制御
```

```
//プラントのモデルの状態方程式 (A,B)
```

```
A=[0 1; 0 0]
```

```
B=[0 1]'
```

```
//行列のサイズ
```

```
[nx,nu]=size(B)
```

```
// 重み関数
```

```
Q=[1 0; 0 0]
```

```

// R を変えて特性を調べる。
Rvec=[0.1,1,10]
for i=1:length(Rvec)
    R=Rvec(i)
//状態フィードバックゲインの決定
X=riccati(A,B*inv(R)*B',Q,'c','eigen');
F=inv(R)*B'*X;
// norm(A'*X+X*A-X*B*inv(R)*B'*X+Q,1) //Riccati check
//極配置
LQroot=spec(A-B*F);
scf(0);
plot(real(LQroot),imag(LQroot),'*')
xgrid();
//ベクトル軌跡
scf(1);
sysL=syslin('c',A,B,F,0);
nyquist(sysL);
xgrid();
//感度関数と相補感度関数
sysS=1/(1+sysL);
sysT=1-sysS;
//周波数特性
nw=200;
omega=logspace(-2,2,nw);
[As,Bs,Cs,Ds]=abcd(sysS);
[At,Bt,Ct,Dt]=abcd(1-sysS);
Sw=freq(As,Bs,Cs,Ds,omega*%i);
Tw=freq(At,Bt,Ct,Dt,omega*%i);
gSw=abs(Sw);
gTw=abs(Tw);
//ゲイン図
scf(2);
plot2d('ln',omega,20*log10(gSw));
plot2d('ln',omega,20*log10(gTw));
xgrid();
end

```

9. LQG 理論によるサーボ系の設計

プラント $P(s) = (A_p, B_p, C_p, D_p)$ に対して、ステップ関数に対して定常偏差がゼロとなる動的制御器 $H(s)$ を LQG 理論に基づき設計せよ。さらに、構成した系の目標値応答と外乱応答をステップ関数に対して数値計算で求めよ。制御器の伝達関数を求め、感度関数 S と相補感度関数 T のゲイン特性を描け。ただし、プラントのモデルを次式で与える。

$$A_m = \begin{bmatrix} 0 & 1 \\ -1 & -0.3 \end{bmatrix}, B_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_m = [1 \quad 0], D_m = 0$$

制御対象がゼロ型であるので、積分器を追加した拡大系に対して状態フィードバックゲインを LQ 理論で設計し、制御対象の状態はオブザーバで推定する。これらを用いて動的制御器を構成する。ここに、LQ 理論の重みを

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

とする。オブザーバはカルマンフィルタにより設計するので、プラントを次式で表し、

$$\dot{x}(t) = A_p x + B_p u(t) + v(t)$$

$$y(t) = Cx(t) + w(t)$$

外乱と測定雑音は平均がゼロで共分散行列が次式で与えられるとする。

$$E[v(t)v(\tau)^T] = V\delta(t-\tau)$$

$$E[w(t)w(\tau)^T] = W\delta(t-\tau)$$

プログラム ‘MCLQcontrol_design.sce’

// 最適制御系+オブザーバの設計

// LQI 状態フィードバック制御+オブザーバ

//プラントのモデルの状態方程式 (Ap,Bp,Cp,Dp)

Am=[0 1; -1 -0.3]

Bm=[0 1]'

Cm=[1 0]

Dm=0

//***//**

//積分器を含む拡大系(A,B)

[ny,nx]=size(Cm)

[nx,nu]=size(Bm)

A=[Am zeros(nx,ny)


```

Cm zeros(ny,ny)
B=[Bm
zeros(ny,nu)]
//*****//
// LQ 制御による設計
q=1
Q=q*[1 0 0
      0 1 0
      0 0 1];
R=1;
//状態フィードバックゲインの決定
X=riccati(A,B*inv(R)*B',Q,'c','eigen');
F=inv(R)*B'*X;
// norm(A'*X+X*A-X*B*inv(R)*B'*X+Q,1) //Riccati check
//状態フィードバック系の解析
//極配置
LQroot=spec(A-B*F);
scf(0);
plot(real(LQroot),imag(LQroot),'*')
xgrid();
//ベクトル軌跡
scf(1);
sysL=syslin('c',A,B,F,0);
nyquist(sysL);
xgrid();
//感度関数と相補感度関数
sysS=1/(1+sysL);
sysT=1-sysS;
//周波数特性
nw=200;
omega=logspace(-2,2,nw);
[As,Bs,Cs,Ds]=abcd(sysS);
[At,Bt,Ct,Dt]=abcd(1-sysS);
Sw=freq(As,Bs,Cs,Ds,omega*%i);
Tw=freq(At,Bt,Ct,Dt,omega*%i);
gSw=abs(Sw);

```

```

gTw=abs(Tw);
//ゲイン図
scf(2);
plot2d('ln',omega,20*log10(gSw));
plot2d('ln',omega,20*log10(gTw));
xgrid();
//*****//
// オブザーバの設計 ; モデル(Am,Bm,Cm,Dm)に対する設計 (カルマンフィルタ)
//共分散行列
V=[1 0
   0 1]
W=1
//オブザーバゲインの決定
Y=riccati(Am,Cm'*inv(W)*Cm,V,'c','eigen');
K=Y*Cm'*inv(W);
observer_poles=spec(Am-K*Cm)
//*****//
//シミュレーション(外乱, 目標値応答)
//プラントの状態方程式 (Ap,Bp,Cp,Dp)
Ap=[0 1; -1 -0.3]
Bp=[0 1]'
Cp=[1 0]
Dp=0

//シミュレーションの閉ループ系の状態方程式(Ac,Bc,Cc,Dc)
Fx=F(1:nx);
FI=F(nx+1:nx+ny);
Ac=[Ap -Bp*Fx -Bp*FI
    K*Cp Am-Bm*Fx-K*Cm -Bm*FI
    Cp zeros(ny,nx) zeros(ny,ny)];
//reference input
Bcr=[zeros(nx,ny)
     zeros(nx,ny)
     -eye(ny)];
//disturbance input
Bcd=[Bp

```

```

        zeros(nx,nu)
        zeros(ny,nu)];
Cc=[Cp zeros(ny,nx) zeros(ny,ny)
     zeros(nu,nx) -Fx -FI];
Dc=zeros(ny+nu,ny);
sys_closed_r=syslin('c',Ac,Bcr,Cc,Dc);
sys_closed_d=syslin('c',Ac,Bcd,Cc,Dc);
//シミュレーション（目標値応答、外乱応答）
t=0:0.01:20;
yr=csim('step',t,sys_closed_r);
yd=csim('step',t,sys_closed_d);
scf(3);
subplot(2,1,1)
plot(t,yr);
xlabel("Step reference response","time(s)","yr(t)")
xgrid
subplot(2,1,2)
plot(t,yd);
xlabel("Step disturbance response","time(s)","yd(t)")
xgrid
//制御器の伝達関数
Ah=[Am-Bm*Fx-K*Cm,-Bm*FI
     zeros(ny,nx), zeros(ny,ny)];
Bh=[K
     eye(ny,ny)];
Ch=F;
Dh=zeros(nu,ny);
sysH=syslin('c',Ah,Bh,Ch,Dh);
//プラントの伝達関数
sysP=syslin('c',Ap,Bp,Cp,Dp);
//感度関数と相補感度関数
sysS=1/(1+sysP*sysH);
sysT=1-sysS;
//周波数特性
nw=200;
omega=logspace(-2,2,nw);

```

```
[As,Bs,Cs,Ds]=abcd(sysS);  
[At,Bt,Ct,Dt]=abcd(1-sysS);  
Sw=freq(As,Bs,Cs,Ds,omega*%i);  
Tw=freq(At,Bt,Ct,Dt,omega*%i);  
gSw=abs(Sw);  
gTw=abs(Tw);  
//ゲイン図  
scf(4);  
plot2d('ln',omega,20*log10(gSw));  
plot2d('ln',omega,20*log10(gTw));  
xgrid();
```