

教科書 13 章 175～178 ページ

13.2.3 設計例の例 13.1 と例 13.2 のプログラムを Scilab で作成した。これらを以下に示す。

佐伯正美 広島大学
2014 年 5 月 10 日

1) 例 13.1

```
// パラメータ平面法 ; 周波数応答に基づく PID 制御器の設計
// sec13_2_example1.sce by M. Saeki May 2014
// 数値例 1 KI を与え, (KP,KD)平面に領域を描画する
// プラントの特性, alpha, beta などの設定

clear all
KI=0.65//積分ゲインの設定
axis_data=[-20 -10 20 20];
s=poly(0,'s');
sysD=syslin('c',s/(1+0.01*s))//近似微分器
N=300;
w=logspace(-2,2,N)//サンプル周波数
M=100;
th_vec=linspace(0,2*pi,M)//楕円を描くパラメータ theta
for i=1:N
    alpha(i)=1.5//重み alpha の設定
    if w(i)<3 then
        betaw(i)=9;
    else
        betaw(i)=0.1;
    end //重み betaw の設定
end
sysP0=syslin('c',(12*s+8)/(20*s^4+113*s^3+146*s^2+62*s+8))//制御対象の有理関数部
[P0_vec]=repfreq(sysP0,w/(2*pi));
Pinv=zeros(N,1);
for i=1:N
    P_vec(i)=P0_vec(i)*exp(-sqrt(-1)*w(i));
    Pinv(i)=1/P_vec(i)// Pinv(s)=1/P(s)
end
```

```

// 安定領域の表示
scf(1);
for i=1:N
    KP_stb(i)=-real(Pinv(i));
    KD_stb(i)=(1/w(i))*(KI/w(i)-imag(Pinv(i)));
end
plot2d(KP_stb,KD_stb,rect=axis_data)
xlabel('stability region','KP','KD')
xgrid()
// 感度制約を満たす領域の表示 // |S| < alpha
scf(2);
for k=1:M
    th=th_vec(k);
    for i=1:N
        ejth=exp(sqrt(-1)*th);
        KP_S(i)=real((-1+ejth/alpha(i))*Pinv(i));
        KD_S(i)=(1/w(i))*(KI/w(i)+imag((-1+ejth/alpha(i))*Pinv(i)));
    end
    plot2d(KP_S,KD_S,5,rect=axis_data)
end
xlabel('sensitivity region','KP','KD')
xgrid()
// 相補感度制約を満たす領域の表示 // |T| < beta
scf(3);
for i=1:N
    for k=1:M
        th=th_vec(k);
        ejth=exp(sqrt(-1)*th);
        KP_T(k)=real(betaw(i)/(ejth-betaw(i))*Pinv(i));
        KD_T(k)=(1/w(i))*(KI/w(i)+imag(betaw(i)/(ejth-betaw(i))*Pinv(i)));
    end
    if betaw(i)>1
        plot2d(KP_T,KD_T,5,rect=axis_data)
    else
        plot2d(KP_T,KD_T,2,rect=axis_data)
    end
end

```

```

end
xtitle('complementary sensitivity region','KP','KD')
xgrid()
// 設計結果 ; S と T のゲイン特性の表示
scf(4);
sysK=syslin('c',2+0.65/s+2.8*s/(1+0.01*s));// 上記の設計で得られた PID 制御器
[K_vec]=repefreq(sysK,w/(2*%pi));
for i=1:N
    L=P_vec(i)*K_vec(i);
    gS(i)=1/abs(1+L);//感度関数
    gT(i)=abs(L/(1+L));//相補感度関数
end
axis_data2=[0.01,-40,10,10];
plot2d('ln',w,20*log10(gS),2,rect=axis_data2);
plot2d('ln',w,20*log10(gT),5,rect=axis_data2);
xtitle('Gain plots','omega[rad/s]','S and T')
xgrid()

```

2) 例 1 3. 2

```

// 数値例 2 KD を与え, (KP,KI)平面に領域を描画する
// sec13_2_example2.sce M. Saeki, May 2014
// プラントの特性, alpha, betaw などの設定
clear
p=0.20
KD=0 //この例では KD=0 に固定
axis_data=[-3 -0.1 7 1];
s=poly(0,'s');
sysD=syslin('c',s/(1+0.01*s));
N=200;
w=logspace(-3,2,N);
M=100;
th_vec=linspace(0,2*%pi,M);
Ms=2;
A=0.1;
sysalpha=syslin('c',Ms*(s+p*A)/(s+Ms*p));//重み関数 alpha(s)の設定

```

```

[alpha]=abs(repfreq(sysalpha,w/(2*pi)));
for i=1:N //重み betaw の設定 (数値例 2 では betaw=10 により緩い制約)
    if w(i) < 10 betaw(i)=10; else betaw(i)=10; end
end
sysP10=syslin('c',(12*s+8)/(20*s^4+113*s^3+146*s^2+62*s+8));//プラント P1
sysP20=syslin('c',(7*s+8)/(20*s^4+113*s^3+130*s^2+62*s+6));//プラント P2
sysP30=syslin('c',(10*s+8)/(10*s^4+113*s^3+146*s^2+55*s+10));//プラント P3

sysP0_vec=[sysP10,sysP20,sysP30];
timedelay_vec=[0.2 0.3 0.1];

for sys_no=1:max(size((sysP0_vec)))
    [P0_vec]=repfreq(sysP0_vec(1,sys_no),w/(2*pi));
    timedelay=timedelay_vec(sys_no);
    Pinv=zeros(N,1);
    for i=1:N
        P_vec(i)=P0_vec(i)*exp(-sqrt(-1)*w(i)*timedelay);
        Pinv(i)=1/P_vec(i);// Pinv(s)=1/P(s)
    end
    // 安定領域の表示
    scf(1);
    for i=1:N
        KP_stb(i)=-real(Pinv(i));
        KI_stb(i)=w(i)*(KD*w(i)+imag(Pinv(i)));
    end
    plot2d(KP_stb,KI_stb,rect=axis_data)
    xtitle('stability region','KP','KD')
    xgrid()
    // 感度制約を満たす領域の表示 // |S| < alpha
    scf(2);
    for k=1:M
        th=th_vec(k);
        for i=1:N
            ejth=exp(sqrt(-1)*th);
            KP_S(i)=real((-1+ejth/alpha(i))*Pinv(i));
            KI_S(i)=w(i)*(KD-imag((-1+ejth/alpha(i))*Pinv(i)));
        end
    end
end

```

```

        end
        plot2d(KP_S,KI_S,5,rect=axis_data)
    end
xtitle('sensitivity region','KP','KI')
xgrid()
// 相補感度制約を満たす領域の表示 // |T| < betaw
scf(3);
for i=1:N
    for k=1:M
        th=th_vec(k);
        ejth=exp(sqrt(-1)*th);
        KP_T(k)=real(betaw(i)/(ejth-betaw(i))*Pinv(i));
        KI_T(k)=w(i)*(KD-imag(betaw(i)/(ejth-betaw(i))*Pinv(i)));
    end
    if betaw(i)>1
        plot2d(KP_T,KI_T,5,rect=axis_data)
    else
        plot2d(KP_T,KI_T,2,rect=axis_data)
    end
end
xtitle('complementary sensitivity region','KP','KI')
xgrid()
end

// 設計結果 ; S と T のゲイン特性の表示
scf(4);
for sys_no=1:max(size((sysP0_vec)))
    [P0_vec]=repfreq(sysP0_vec(1,sys_no),w/(2*%pi));
    timedelay=timedelay_vec(sys_no);
    for i=1:N
        P_vec(i)=P0_vec(i)*exp(-sqrt(-1)*w(i)*timedelay);
    end
    sysK=syslin('c',1.8+0.21/s);// 上記の設計で得られた PID 制御器
    [K_vec]=repfreq(sysK,w/(2*%pi));
    for i=1:N
        L=P_vec(i)*K_vec(i);
    end
end

```

```
gS(i)=1/abs(1+L);//感度関数
gT(i)=abs(L/(1+L));//相補感度関数
end
axis_data2=[0.001,-40,10,20];
plot2d('ln',w,20*log10(gS),2,rect=axis_data2);
plot2d('ln',w,20*log10(gT),5,rect=axis_data2);
xlabel('Gain plots','omega[rad/s]','S and T')
xgrid()
end
```