ロバスト PID 制御器のパラメータ平面設計のプログラム

広島大学　佐伯正美

２０１４年4月２４日

１．　　ロバスト制御器の設計問題の概要

制御対象は１入力１出力のスカラー系とし，制御対象のモデル集合が次式の乗法的モデル誤差で表現されているとする．

$$\tilde{P}(s) = (1+\Delta_m(s))P(s) \tag{1.1}$$

ここに，モデル誤差の大きさが

$$\left|\Delta(j\omega)\right| \le \left|W_m(j\omega)\right|, \omega \in R \tag{1.2}$$

により評価されるとし，次式により $\beta(\omega)$ を定義する．

$$\beta(\omega) = \frac{1}{\left|W_m(j\omega)\right|} \tag{1.3}$$

感度関数と相補感度関数は次式で表される．

$$\begin{aligned} S(s) &= \frac{1}{1+P(s)K(s)} \\ T(s) &= \frac{P(s)K(s)}{1+P(s)K(s)} \end{aligned} \tag{1.4}$$

以下に代表的なロバスト制御の設計問題を列挙する．ノミナルのプラントが $K(s)$ で安定化されているとき，各制御問題はそれぞれ以下の条件式を満たす $K(s)$ を求める問題となる．

①　ロバスト安定問題（robust stability problem）

$$\left|T(j\omega)\right| < \beta(\omega), \omega \in R \tag{1.5}$$

②　ノミナル感度問題（nominal sensitivity problem）

$$\left|S(j\omega)\right| < \alpha(\omega), \omega \in R \tag{1.6}$$

③　２ディスク問題（two disc problem）

$$\begin{aligned} \left|S(j\omega)\right| &< \alpha(\omega), \omega \in R \\ \left|T(j\omega)\right| &< \beta(\omega), \omega \in R \end{aligned} \tag{1.7}$$

④　混合感度問題（mixed sensitivity problem）

$$\left(\frac{\left|S(j\omega)\right|}{\alpha(\omega)}\right)^2 + \left(\frac{\left|T(j\omega)\right|}{\beta(\omega)}\right)^2 < 1, \omega \in R \tag{1.8}$$

⑤　ロバスト感度問題（robust sensitivity problem）

$$\frac{|S(j\omega)|}{\alpha(\omega)}+\frac{|T(j\omega)|}{\beta(\omega)}<1,\omega\in R \tag{1.9}$$

（(1.9)式の補足説明をしておく．この式は実プラントに対する感度関数が

$$\tilde{S}(s)=\frac{1}{1+\tilde{P}(s)K(s)}$$

で与えられるとき，モデル集合に対して

$$\left|\tilde{S}(j\omega)\right|<\alpha(\omega),\omega\in R$$

が成り立つための必要十分条件として得られる.）

２．PID 制御器の設計問題とパラメータ平面設計

PID 制御器の伝達関数は

$$K(s)=K_P+K_I\frac{1}{s}+K_D\frac{s}{1+\tau s} \tag{1.10}$$

で表されるが，３つのゲインのうちの一つを固定して，残りの２つのゲインに関してパラメータ平面上に制約を満たす許容領域を描画する問題を考える．この問題はより柔軟に次の伝達関数で表される３項制御器の２つのゲイン p_1, p_2 を扱う問題として表される．

$$K(s)=K_0(s)+p_1K_1(s)+p_2K_2(s) \tag{1.11}$$

これを用いると，PI 制御器のゲイン調整では $(K_P,K_I)=(p_1,p_2)$ と考えて

$$K_0(s)=0,K_1(s)=1,K_2(s)=\frac{1}{s} \tag{1.12}$$

と設定すればよい．また，PID 制御器で積分ゲインを固定して比例ゲインと微分ゲインをパラメータとする場合には，$(K_P,K_D)=(p_1,p_2)$ と考えて，次式で設定すればよい.

$$K_0(s)=\frac{K_I}{s},K_1(s)=1,K_2(s)=\frac{s}{1+\tau s} \tag{1.13}$$

混合感度問題の(1.8)式とロバスト感度問題の(1.9)式を満たすパラメータ (p_1,p_2) の集合を表示するプログラムを作成した．アルゴリズムは周波数を離散化したサンプル周波数に関して各不等式を満たすパラメータを求め，それをパラメータ平面に表示する方法である.

上記の問題の中で，①から③は文献[1]の１３．２節で述べており，MATLAB プログラムも載せている．ここでは，混合感度問題とロバスト感度問題に対する設計法を示す．混合感度問題の領域の計算はノミナル感度問題と同様に計算できる．それに比較してロバスト感度問題の場合の式の導出は面倒であるが，導出結果を文献[2]の付録に記している.

[1] 佐伯，制御工学，朝倉書店，2013

[2]　佐伯，平山，ロバスト感度最小化問題に対する PID 制御器のパラメータ空間設計,

計測自動制御学会論文集，Vol. 32, No. 12, pp. 1612-1619, 1996

３．PID 制御器の設計プログラム例

ひとつのサンプル周波数に対して(1.8)式を満たすパラメータの集合を描画する MATLAB 関数 ”MS_plot_func.m” を作成した. 同様に(1.9)式を満たすパラメータの集合を描画する MATLAB 関数 “RS_plot_func.m”を作成した．これらの使用法を示すために， PI 制御器の設計問題に対するプログラム”design_PI.m”を作成した.

設計問題

１）制御対象は一次遅れむだ時間系とし，次式で伝達関数が与えられる.

$$P(s)=\frac{k}{1+sb}e^{-Ls} \tag{1.14}$$

ここでは各係数に不確かさがあるとして，以下の区間にパラメータがあるとするモデル集合を考える.

$$0.8\le b\le 1.2, 0.9\le k\le 1.1, 0.8\le L\le 1.2 \tag{1.15}$$

このモデル集合をそのまま扱わずに乗法的モデル誤差のモデル集合で表す．そこで,
ノミナル値を

$$b=1, k=1, L=1 \tag{1.16}$$

に設定し，乗法的モデル誤差の大きさの上界値を求める．上界値は $\beta(\omega)$ とおく.

２）制御器は次式の伝達関数の PI 制御器とし，PI ゲインを設計する.

$$K(s)=K_p+\frac{1}{s}K_i \tag{1.17}$$

制御性能を表す感度制約の重みは次式で与える. 速応性を良くするためにパラメータ q を最小化する．すなわち，q を小さくしていくと制約式を満たす許容集合が縮小するので，集合が消滅するまえのゲインを求める.

$$\alpha(\omega)=\left|\frac{2.5(qs+0.04)}{qs+4}\right|_{s=j\omega} \tag{1.18}$$

計算結果

１）プラント特性やフィードバック系の特性

まず，プラントのステップ応答を多くのパラメータの組に対して求めたものを図１に示す.

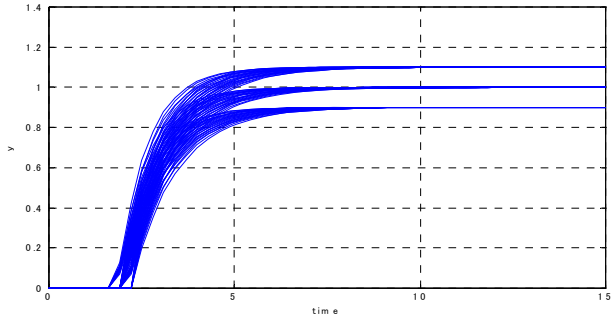

図１　プラントのステップ応答

つぎに，PI ゲインを適当に $K_p = 0.8, K_i = 0.5$ と設定した場合の閉ループ系のステップ目標値応答を図２に示す.

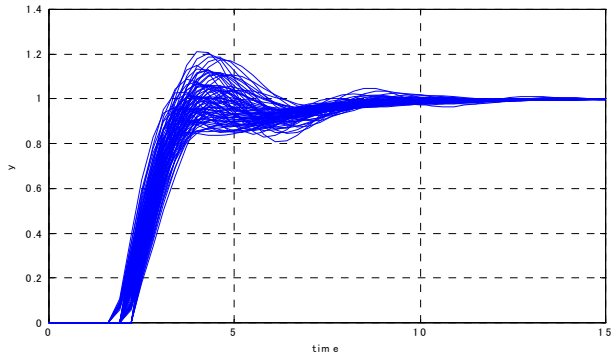

図２　閉ループ系のステップ応答　$K_p = 0.8, K_i = 0.5$

このフィードバック系に対して，感度関数と相補感度関数のゲイン特性を調べる．図３はノミナルプラントに対するグラフであり，図４はモデル集合に対するグラフである.

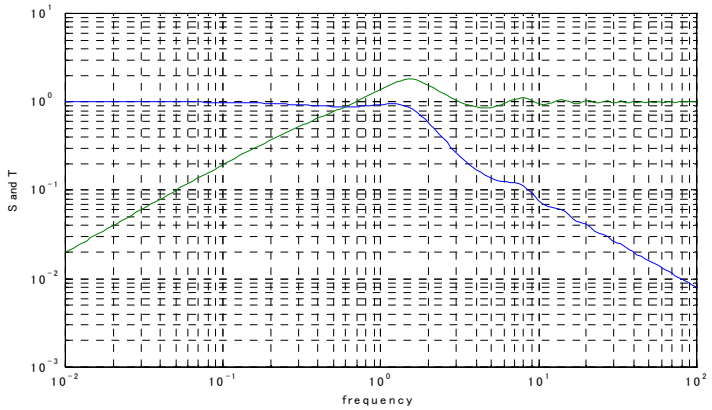

図３　感度関数Ｓ、相補感度関数Ｔのゲイン図　（ノミナル制御対象，$K_p = 0.8, K_i = 0.5$）

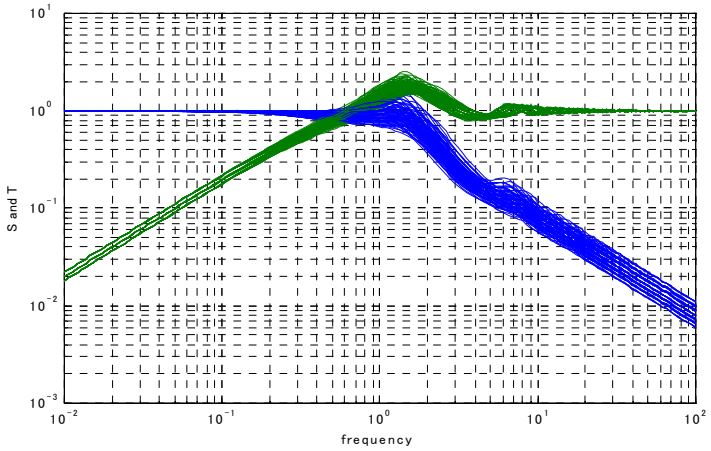

図４　感度関数Ｓ、相補感度関数Ｔのゲイン図　（変動有り制御対象、$K_p = 0.8, K_i = 0.5$）

２）モデル誤差の評価

加法的モデル誤差の大きさは次式で評価される．図５にモデル集合に対するグラフを描いているので，各周波数における最大値がモデル誤差の大きさの評価となる．

$$\left|\Delta_a\right| = \left|P(s,1,1,1) - P(s,k,b,L)\right| \tag{1.19}$$

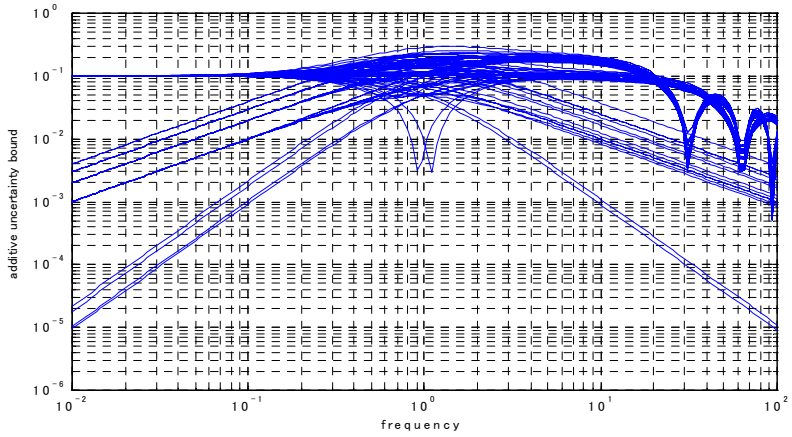

図５　加法的モデル誤差の大きさ

乗法的モデル誤差の大きさは次式で評価される．図６にモデル集合に対するグラフを描いているので，各周波数における最大値がモデル誤差の大きさの評価となる．

$$\left|\Delta_m\right| = \frac{\left|P(s,1,1,1) - P(s,k,b,L)\right|}{\left|P(s,1,1,1)\right|} \tag{1.20}$$

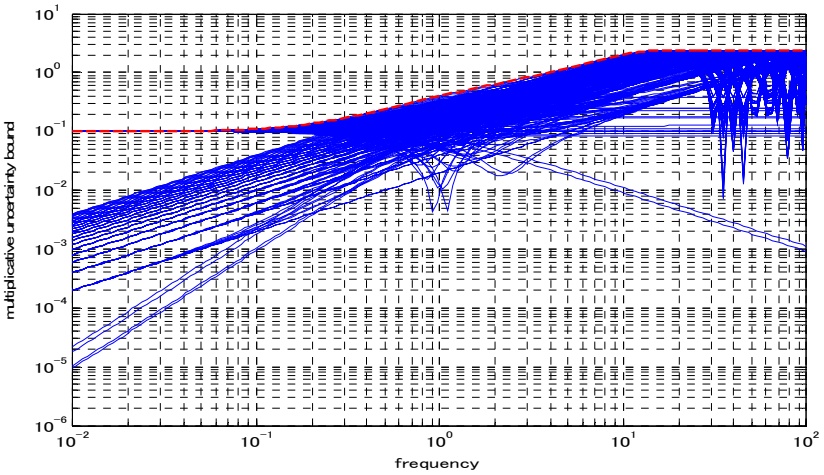

図６　乗法的モデル誤差の大きさ

３）ＰＩ制御系の設計

図６でグラフの最大値は赤い破線で示している．これを取り出し図７に示す．

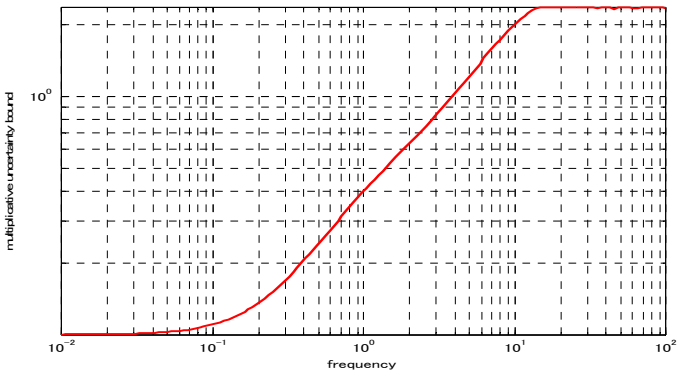

図７　乗法的モデル誤差の大きさ　$|W_m(j\omega)| = 1/\beta(\omega)$

感度関数の重み $\alpha(\omega)$ は(1.18)式で与えられる．$\beta(\omega)$ と $q = 4.7$ の場合の $\alpha(\omega)$ のグラフを図８に示す．

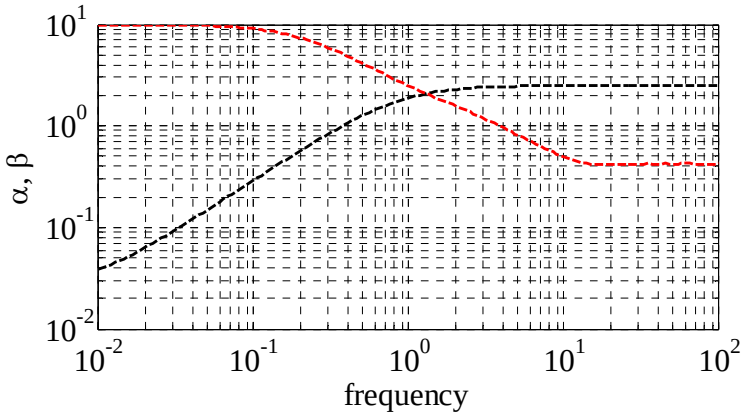

図８　重み関数の特性

a）混合感度問題の解領域

q を減少させていくと，パラメータ平面上の許容集合が縮小する．$q = 3.32$ の場合のパラメータ平面を図９に示す.

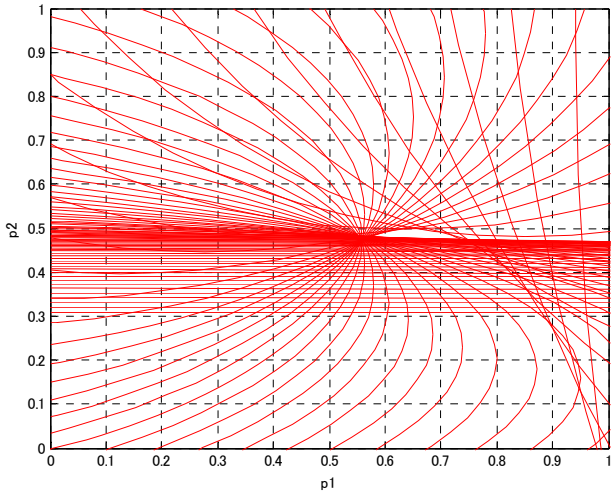

図９　解の領域（混合感度問題, $q = 3.32$ ）

許容集合は図９の中央付近の小さい領域である.

この空白の部分から，たとえば　$K_p = 0.65, K_i = 0.48$ を選ぶ．この場合のモデル集合に対する感度関数と相補感度関数を図１０に示す．混合感度問題ではモデル集合は制約を厳密には満たさず，少し制約を越える場合が起こる．これは図１０にも現れている．図１１にモデル集合に対する閉ループ系のステップ目標値応答を示す.

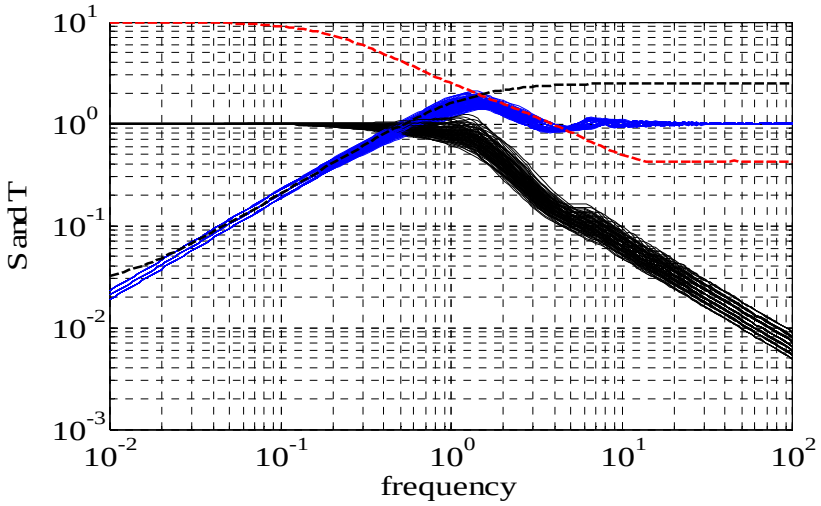

図１０　モデル集合に対する感度関数と相補感度関数（$K_p = 0.65, K_i = 0.48$）

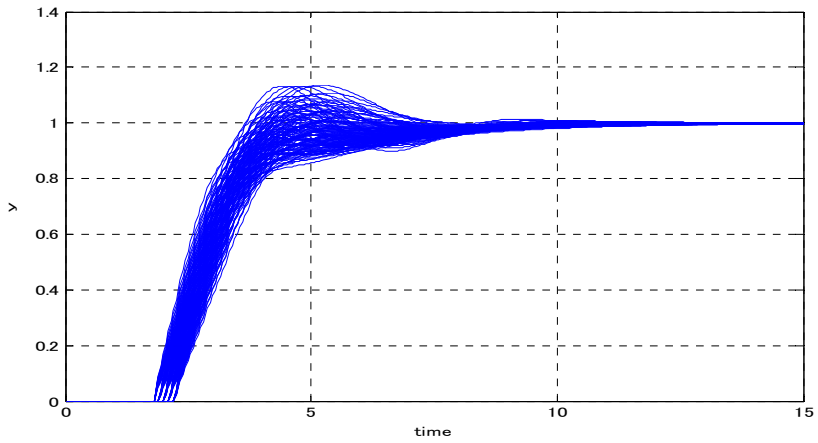

図１１　閉ループ系のステップ応答（$K_p = 0.65, K_i = 0.48$）

b) ロバスト感度問題の解領域

q を減少させていくと，パラメータ平面上の許容集合が縮小する．$q = 4.7$ の場合のパラメータ平面を図１２に示す.

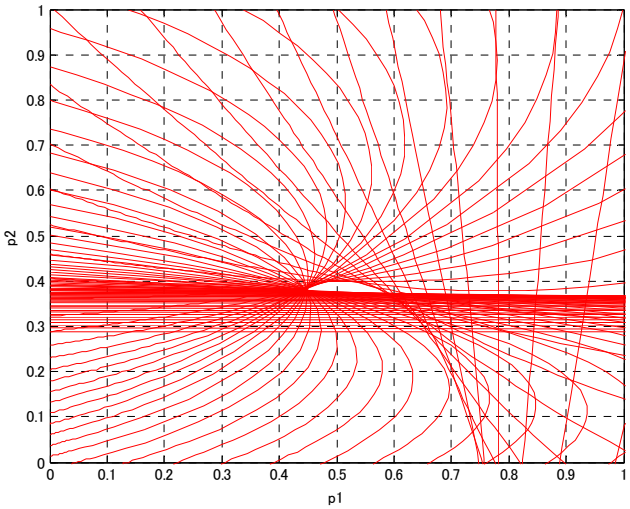

図１２　解の領域（ロバスト感度問題、$q = 4.7$ ）

許容集合は図１２の中央付近の小さい領域である.

この空白の部分から，たとえば　$K_p = 0.5, K_i = 0.4$ を選ぶ．この場合のモデル集合に対する感度関数と相補感度関数を図１３に示す．ロバスト感度問題ではモデル集合は制約を厳密に満たす．これは図１３にも現れている．図１４にモデル集合に対する閉ループ系のステップ目標値応答を示す.

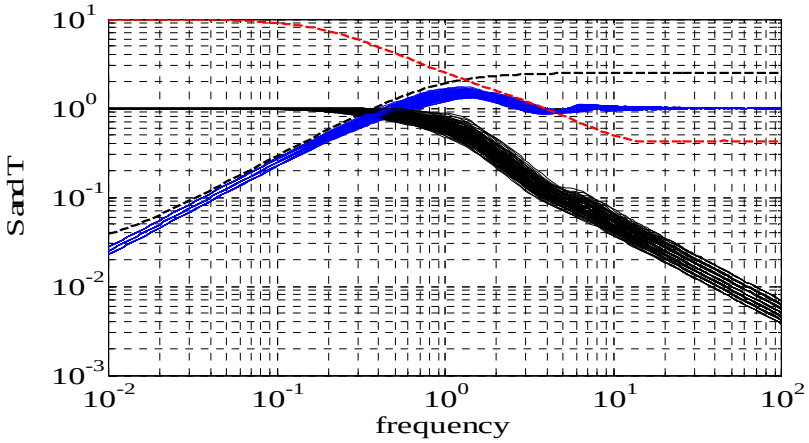

図１３モデル集合に対する感度関数と相補感度関数（$K_p = 0.5, K_i = 0.4$）

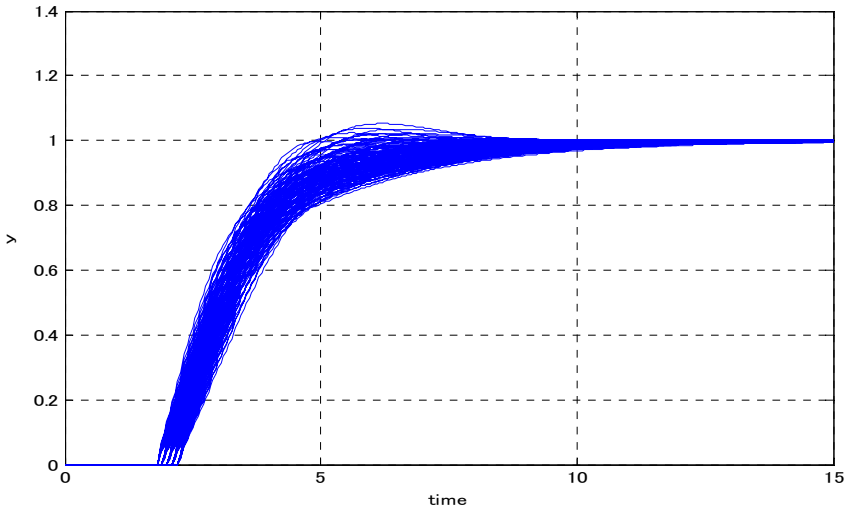

図１４　閉ループ系のステップ応答（$K_p = 0.5, K_i = 0.4$）

付録　プログラム

```
% ------------------------------------------------------------------
% プログラム名　design_PI.m
% ３項制御器のパラメータ空間設計法のPI制御への適用例
% 次式の３項制御器のパラメータを (p1, p2)平面に描画する関数を用いる.
%              K=K0(s)+K1(s)*p1+K2(s)*p2
% つぎの２つの問題が扱える.
% 1)　混合感度問題
%            {|1/(1+GK)|(1/alph)}^2+{|GK/(1+GK)|(1/beta)}^2 < 1
% 2)　ロバスト感度問題
%            |1/(1+GK)|(1/alph)+|GK/(1+GK)|(1/beta) < 1
% ------------------------------------------------------------------
%%
clear all
close all
% 混合感度法か, ロバスト感度法を指定
   %design_method='mixedsensitivity'
   design_method='robustperformanc'
%  描画する領域は矩形領域[p1min,p1max,p2min,p2max]で指定する.
% rctn=[p1min,p1max,p2min,p2max]
     rctn=[-1 20 -200 1000];
%  CP:矩形領域[p1min,p1max,p2min,p2max]に含まれる点の数である
%  ひとつの曲線を描く際の点数CPが指定値numCP以下であれば, その曲線は描かない.
    numCP=1;% if CP is greater than or equal to numCP, plot the set.
% サンプル周波数を指定
       wnum=150;
       w=logspace(-2,2,wnum);
% 制御対象の周波数応答データ　g=reg(i)+j*img(i), i=1,2,...,wnumを計算
 K=1;
 b=1;
 L=1;
  for i=1:wnum
   jomega=w(i)*sqrt(-1);
   g=(K/(1+jomega*b))*(cos(w(i)*L)-sqrt(-1)*sin(w(i)*L));
   reg(i)=real(g);
   img(i)=imag(g);
```

```
  end
% プラントのベクトル軌跡の描画
figure(1)
plot(reg,img)
xlabel('real part of g(jw)')
ylabel('imaginal part of g(jw)')
grid

% --------------------------------
  % ３項制御器のK1,K2,K3をPID制御器として用いるための設定
  %  K=K0(s)+K1(s)p1+K2(s)p2
    % PI制御器 K=p1+(1/s)p2 の場合の設定
    % これより, p1=> Kp, p2=>Kiに対応
      % K0(s) =0
        numK0=[0]; denK0=[1];
      % K1(s)=1
        numK1=[1]; denK1=[1];
      % K2(s)={1/s}
        numK2=[1]; denK2=[1,0];
    % 各要素の周波数応答の計算
      [a0,b0]=nyquist(numK0,denK0,w);
      [a1,b1]=nyquist(numK1,denK1,w);
      [a2,b2]=nyquist(numK2,denK2,w);
% ----------------------------------------
 % 制約式の重み alphとbetaの設定

 % beta  乗法的モデル誤差の逆数
 %    maxdel(w)  乗法的モデル誤差の大きさの上界の評価値
 %    beta(w) =1/maxdel(w)

 % 数値例に対する乗法的モデル誤差の大きさ maxdel(w)の計算
 %   プラントパラメータ b,K,Lの値を振って上界を計算
 for i=1:wnum
    maxdel(i)=0;
 end
 figure(2)
```

```
for b=0.8:0.1:1.2
for K=0.9:0.1:1.1
for L=0.8:0.02:1.2
for i=1:wnum
   jomega=w(i)*sqrt(-1);
   p=(K/(1+jomega*b))*(cos(w(i)*L)-sqrt(-1)*sin(w(i)*L));
   p0=(1/(1+jomega*1))*(cos(w(i))-sqrt(-1)*sin(w(i)));
 % 加法的モデル誤差は deltaadd(i)=abs(p-p0);
 deltamul(i)=abs((p-p0)/p0);
 if deltamul(i)>maxdel(i)
    maxdel(i)=deltamul(i);
 end
end
% 各パラメータ値に対する乗法的モデル誤差のゲインを青線で図２にプロット
loglog(w,deltamul)
hold on
end
end
end
% 乗法的モデル誤差のゲインの上界を赤線で図２にプロット
figure(2)
loglog(w,maxdel,'r--','LineWidth',2)
grid
xlabel('frequency')
ylabel('multiplicative uncertainty bound')
% betaを設定
for i=1:wnum
   beta(i)=1/maxdel(i);
end

% alpha 感度関数の制約
% alphaの帯域を変えるパラメータ　qが小さいほど感度を抑える帯域が増加
  %q=3.32 % 混合感度
  q=4.7 % 感度最小化
       numAlph=[1*q 0.04]*2.5;% alphaの分子
       denAlph=[1*q 4];  % alphaの分母
```

```
    gam=1;  % gammaの値、通常１に固定
   Talph=tf(numAlph,denAlph);
    [magal,phaseal]=bode(Talph,w);
   for i=1:wnum
      alph(i)=magal(i);
   end
% alpha(w)とbeta(w)の描画
figure(3)
loglog(w,alph,'k--',w,beta,'r--','LineWidth',2)
xlabel('frequency','FontSize',16,'FontName','Times New Roman')
ylabel('¥alpha, ¥beta','FontSize',16,'FontName','Times New Roman')
set(gca,'FontSize',16)
set(gca,'FontName','Times New Roman')
grid

%% ----------------------------------------
%  領域を描画するための準備
%  cos(th)とsin(th)を計算する.
% thnumは楕円を現す点数を指定するので,
% thnumが大きいほどに描画精度が高くなるが計算時間が長くなる.
    thnum=101;
    for k=1:thnum
      costh(k)=cos(2*pi*(k-1)/(thnum-1));
      sinth(k)=sin(2*pi*(k-1)/(thnum-1));
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 各サンプル周波数で (p1, p2)の許容集合を描画し, パラメータ平面に重ね描きする.
    figure(5)
   for i=1:wnum
      g=[reg(i),img(i)];
      K0=[a0(i),b0(i)];
      K1=[a1(i),b1(i)];
      K2=[a2(i),b2(i)];
      if design_method=='mixedsensitivity'
         % 混合感度法による描画の関数を用いる
         CP=MS_plot_func(g,alph(i),beta(i),K0,K1,K2,costh,sinth,rctn,numCP);
```

```
      elseif design_method=='robustperformanc'
         % ロバスト感度法による描画の関数を用いる
         CP=RS_plot_func(g,alph(i),beta(i),K0,K1,K2,costh,sinth,rctn,numCP);
      else
          break
      end
  end
  % 矩形領域を表示
    axis(rctn)
    xlabel('p1')
    ylabel('p2')
    grid

 % 解領域の存在する範囲にズームして表示
    axis([0 1 0 1])
% 図５に示された解領域から適当な点を選定する
   Kp=input('Kp= ')
   Ki=input('Ki= ')

%% 選定したKp,Kiに対してパラメータを振って感度関数と相補感度関数の
% ゲイン特性を 図４に重ね描きする.
figure(4)
for b=0.8:0.1:1.2
for K=0.9:0.1:1.1
for L=0.8:0.1:1.2
for i=1:wnum
   jomega=w(i)*sqrt(-1);
   h=Kp+Ki/jomega;
   p=(K/(1+jomega*b))*(cos(w(i)*L)-sqrt(-1)*sin(w(i)*L));
   T(i)=abs(p*h/(1+p*h));
   S(i)=abs(1/(1+p*h));
end
loglog(w,T,'k',w,S,'b')
hold on
end
end
```

```
end
figure(4)
xlabel('frequency','FontSize',16,'FontName','Times New Roman')
ylabel('S and T','FontSize',16,'FontName','Times New Roman')
set(gca,'FontSize',16)
set(gca,'FontName','Times New Roman')
loglog(w,alph,'k--',w,beta,'r--','LineWidth',2)
grid
```

```
% ------------------------------------------------------------------------
% ファイル名　MS_plot_func.m
% 混合感度問題のパラメータ平面への描画のための関数
% 次式の３項制御器に対して，ひとつの周波数s=jwにおけるパラメータ(p1, p2)の
% 許容集合を描く.
%              K=K0(s)+K1(s)*p1+K2(s)*p2
% 次式の混合感度制約を満たすパラメータ集合を描く
%            {|1/(1+GK)|(1/alph)}^2+{|GK/(1+GK)|(1/beta)}^2 < 1
%ここに，制御器の周波数応答のひとつの周波数における値が
%                       K0=K0(1)+j*K0(2)=a0+j*b0
%                       K1=K1(1)+j*K1(2)=a1+j*b1
%                       K2=K2(1)+j*K2(2)=a2+j*b2
%で与えられ，制御対象の周波数応答の値が G=G(1)+j*G(2)=reg+j*imgで与えられる.
% ノート１）領域の形状は楕円である．許容領域が楕円の外部であれば，赤線で曲線を
% 描き，楕円の内部であれば青線で描く.
% ノート２）rctn=[p1min,p1max,p2min,p2max]で指定された矩形領域ないに曲線を描く
% ためのサンプル点数がnumCP以下であれば曲線は描画しない.
% ノート３）変数 sinth と costhはそれぞれ sin(theta)と cos(theta)をあらわし，
% thetaは0 から 2*piまでとする．これらは楕円を描くサンプル点に対応している.
% ------------------------------------------------------------------------
function CP=MS_plot_func(G,alph,beta,K0,K1,K2,costh,sinth,rctn,numCP)
        reg=G(1);img=G(2);
        a0=K0(1);b0=K0(2);
        a1=K1(1);b1=K1(2);
        a2=K2(1);b2=K2(2);
        [temp,thnum]=size(costh);
        p1min=rctn(1);p1max=rctn(2);p2min=rctn(3);p2max=rctn(4);
% ----------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 点を計算し領域の境界を描く
    absg2=reg*reg+img*img;
    A=beta*beta-1;
    invTH=inv([a1 a2;b1 b2]);
    c=-reg*beta*beta/absg2/A;
    d=img*beta*beta/absg2/A;
    if A>0
```

```
    CP=0;
    r=sqrt((alph*alph+A)/absg2)*beta/alph/abs(A);
      for k=1:thnum
        x=c+r*costh(k);
        y=d+r*sinth(k);
        % 領域の外部
        p=invTH*([x, y]'-[a0,b0]');
        px(k)=p(1);py(k)=p(2);
        % 領域に含まれる点数を数える
        if p(1)>p1min & p(1)<p1max & p(2)>p2min & p(2)<p2max
          CP=CP+1;
        end
      end
      if CP >= numCP
        plot(px, py,'r')
        drawnow
        hold on
       end
elseif A<0 & alph*alph+A>0
    CP=0;
    r=sqrt((alph*alph+A)/absg2)*beta/alph/abs(A);
    for k=1:thnum
      x=c+r*costh(k);
      y=d+r*sinth(k);
      % 領域の内部
      p=invTH*([x, y]'-[a0,b0]');
      px(k)=p(1);py(k)=p(2);
      % 領域に含まれる点数を数える
      if p(1)>p1min & p(1)<p1max & p(2)>p2min & p(2)<p2max
          CP=CP+1;
      end
    end
    if CP >= numCP
      plot(px, py,'b')
      drawnow
      hold on
```

```
    end
else
    % A=0の場合はまず起こらないので，描いていない．
end
```

```
% ------------------------------------------------------------------------
% ファイル名　RS_plot_func.m
% ロバスト感度問題のパラメータ平面への描画のための関数
% 次式の３項制御器に対して，ひとつの周波数s=jwにおけるパラメータ(p1, p2)の
% 許容集合を描く.
%              K=K0(s)+K1(s)*p1+K2(s)*p2
% 次式のロバスト感度制約を満たすパラメータ集合を描く
%             |1/(1+GK)|(1/alph)+|GK/(1+GK)|(1/beta) < 1
%ここに，制御器の周波数応答のひとつの周波数における値が
%                       K0=K0(1)+j*K0(2)=a0+j*b0
%                       K1=K1(1)+j*K1(2)=a1+j*b1
%                       K2=K2(1)+j*K2(2)=a2+j*b2
%で与えられ，制御対象の周波数応答の値が G=G(1)+j*G(2)=reg+j*imgで与えられる.
% ノート１）領域の形状は楕円に近い．許容領域が領域の外部であれば，赤線で曲線を
% 描き，領域の内部であれば青線で描く.
% ノート２）rctn=[p1min,p1max,p2min,p2max]で指定された矩形領域ないに曲線を描く
% ためのサンプル点数がnumCP以下であれば曲線は描画しない.
% ------------------------------------------------------------------------

function CP=RS_plot_func(G,alph,beta,K0,K1,K2,costh,sinth,rctn,numCP)
        reg=G(1);img=G(2);
        a0=K0(1);b0=K0(2);
        a1=K1(1);b1=K1(2);
        a2=K2(1);b2=K2(2);
        [temp,thnum]=size(costh);
        p1min=rctn(1);p1max=rctn(2);p2min=rctn(3);p2max=rctn(4);
% ------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 点を計算し領域の境界を描く
    absg2=reg*reg+img*img;
    A=absg2*(1-1/beta/beta);
    C=1-1/alph/alph;
    invTH=inv([a1 a2;b1 b2]);
    if A>0 & C<0
         CP=0;
        for k=1:thnum
```

```
        B=reg*costh(k)-img*sinth(k)-sqrt(absg2)/alph/beta;
        r=(-B+sqrt(B*B-A*C))/A;
        x=r*costh(k);
        y=r*sinth(k);
        % 領域の外部
        p=invTH*([x, y]'-[a0,b0]');
        px(k)=p(1);py(k)=p(2);
        % 領域に含まれる点数を数える
         if p(1)>p1min & p(1)<p1max & p(2)>p2min & p(2)<p2max
          CP=CP+1;
         end
      end
      if CP >= numCP
        plot(px, py,'r')
        drawnow
        hold on
      end
elseif A<0 & C>0
     CP=0;
      for k=1:thnum
        B=reg*costh(k)-img*sinth(k)-sqrt(absg2)/alph/beta;
        r=(-B-sqrt(B*B-A*C))/A;
        x=r*costh(k);
        y=r*sinth(k);
        % 領域の内部
        p=invTH*([x, y]'-[a0,b0]');
        px(k)=p(1);py(k)=p(2);
        % 領域に含まれる点数を数える
        if p(1)>p1min & p(1)<p1max & p(2)>p2min & p(2)<p2max
          CP=CP+1;
         end
      end
      if CP >= numCP
         plot(px,py,'b')
         drawnow
         hold on
```

```
    end
elseif A>0 & C>0
      CP=0;
    fai=atan2(reg,-img);
    Dsign=1/alph/beta-sqrt((1-1/alph/alph)*(1-1/beta/beta));
    if Dsign>-1
       thb=asin(Dsign);
       th2=linspace(pi-thb-fai,2*pi+thb-fai,thnum);
       for k=1:thnum
         B=reg*cos(th2(k))-img*sin(th2(k))-sqrt(absg2)/alph/beta;
         D=B*B-A*C;
         if D <0  % 数値計算誤差で負になる場合があるのでマイナー修正
           D=D+0.0000001;
         end
         rmax=(-B+sqrt(D))/A;
         rmin=(-B-sqrt(D))/A;
         % 領域の外部
         xmax=rmax*cos(th2(k));
         ymax=rmax*sin(th2(k));
         xmin=rmin*cos(th2(k));
         ymin=rmin*sin(th2(k));
         % 領域の外部
         pmax=invTH*([xmax, ymax]'-[a0,b0]');
         pxmax(k)=pmax(1);pymax(k)=pmax(2);
         pmin=invTH*([xmin, ymin]'-[a0,b0]');
         pxmin(k)=pmin(1);pymin(k)=pmin(2);
         % 領域に含まれる点数を数える
         if pmin(1)>p1min & pmin(1)<p1max & pmin(2)>p2min & pmin(2)<p2max
          if pmax(1)>p1min & pmax(1)<p1max & pmax(2)>p2min & pmax(2)<p2max
           CP=CP+1;
          end
         end
       end
       if CP >=numCP
         plot(pxmax, pymax,'r',pxmin, pymin,'r')
         drawnow
```

```
            hold on
        end
    end
elseif A<0 & C<0
    display('A<0 & C<0 が起こったので，評価関数の重みが適切でない．')
end
```