

目次

1. はじめに
2. MATLAB の導入
3. ステップ応答, インパルス応答, 一般の応答
4. ボード線図とナイキスト軌跡
5. 根軌跡
6. 設計例
7. 積分器と定常偏差

1. はじめに

制御性能の解析や制御系設計では, 時間応答のシミュレーションや周波数応答などを数値計算し, それをグラフに表示することが必要となる. この目的に適した数値計算ツールに MATLAB/SIMULIK (有料) や類似機能の Scilab (無料) がある. MATLAB は広く世界の大学で用いられており, 企業の開発にも用いられている. SIMULINK はブロック線図で表された制御系のシミュレータである. ここでは, MATLAB を利用した古典制御の解析設計の入門的な内容を説明する.

2章では MATLAB を導入する. 3章から5章では, 古典制御の解析に必要なステップ応答の計算や周波数応答の表示を MATLAB で行えるようにするために, ボード線図, ベクトル軌跡, 根軌跡を描画するプログラム例を示す. 6章ではこれらを用いて古典制御による制御系設計例を示す. 7章では積分器と定常偏差の関係について2次系を例に説明する.

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, 下記の参考図書の11章と12章に記している. 現代制御による解析設計の入門的な内容も下記の URL に掲載している.

[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

広島大学 佐伯正美

2. MATLABの導入

数値計算のためのプラットフォーム MATLAB

行列計算を主体とした数値計算やグラフ表示ができ、レポート作成にも有用である。使い方はコマンド入力による電卓風の使い方と一連のコマンドをファイル（すなわちプログラム）に入力し一度に実行する使い方がある。制御系で用いるプログラムがツールボックスに用意されており、これらのツールを用いることで制御系の時間応答シミュレーション、周波数特性の表示、制御系設計などが効率よく行える。

2.1. 電卓風な使い方（ひとつのコマンドを画面で入力し実行し結果を表示）

1) MATLAB をクリックして起動すると図1のコマンドウィンドウが表示される。

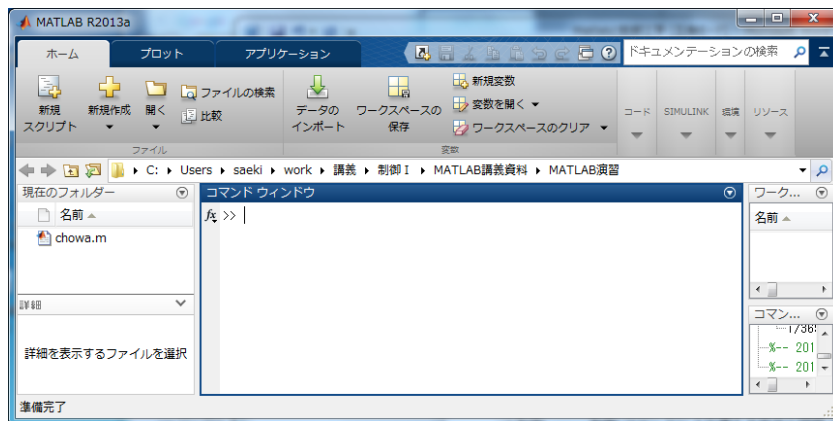


図1 コマンドウィンドウ

2) 図1のコマンドウィンドウのプロンプト `>>` にコマンドを入力する。

「`>> help`」のように入力すれば、`help` コマンドによりツールボックスの一覧表が表示される。制御系設計関連のツールボックスは `control` にあり、「`>> help control`」を入力すれば、制御に用いるコマンド名が表示される。コマンド名がわかっている場合にそのコマンドの意味を知るには、「`>> help コマンド名`」と入力する。たとえば、`help` コマンドの機能を知るには、「`>> help help`」と入力すればよい。

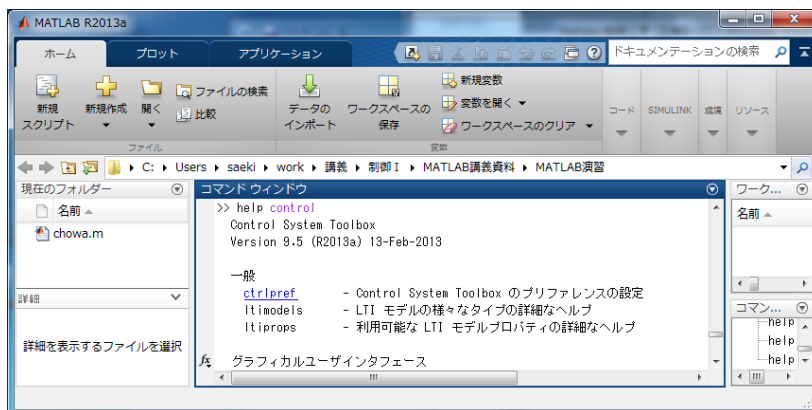


図2 `>> help control` の実行結果

MATLAB は基本的な行列演算を高速かつ高精度に実行できる。基本演算として、行列の加減乗除(+*/)や行列式(det), 逆行列(inv), 固有値(eig)の数値計算などが挙げられる。

3) 実行例 (直接に以下のコマンドを打ち込んでみよう.)

行列 A, ベクトル B を入力し, $C=A*B$, B の転置を計算. 行列 A の行列式, 固有値と固有ベクトルを計算する.

$$A = \begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix}, B = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

出力例

```
>> A=[1 4;3 2]      行列 A の入力, コロン「;」は改行
A =
     1     4
     3     2
>> B=[1;-2]        列ベクトル B の入力
B =
     1
    -2
>> C=A*B           A と B の積を C へ代入
C =
    -7
    -1
>> B'              ベクトル B の転置はダッシュ「'」を使う
ans =
     1    -2
>> det(A)          行列式の計算
ans =
    -10
>> [T,D]=eig(A)    行列 A の固有値と固有ベクトルを計算し, D と T へ代入する
T =
   -0.8000   -0.7071
    0.6000   -0.7071
D =
   -2     0
     0     5
```

2. 2. プログラムの作成と実行による使い方

上記の実行例では、以下の作業を逐次行った。これをファイルに入力し、そのファイル（すなわちプログラム）を実行することで、一度に実行できる。その方法を述べる。

プログラム

```
A=[1 4;3 2]
B=[1;-2]
C=A*B
B'
det(A)
[T,D]=eig(A)
```

- 1) MATLAB を起動すると図 1 が表示される。
- 2) ウィンドウの「新規スクリプト」をクリックすると、「エディター-Untitled」の名前のウィンドウ（図 2）が新たに開かれる。
- 3) このウィンドウに上記のプログラムを入力する。

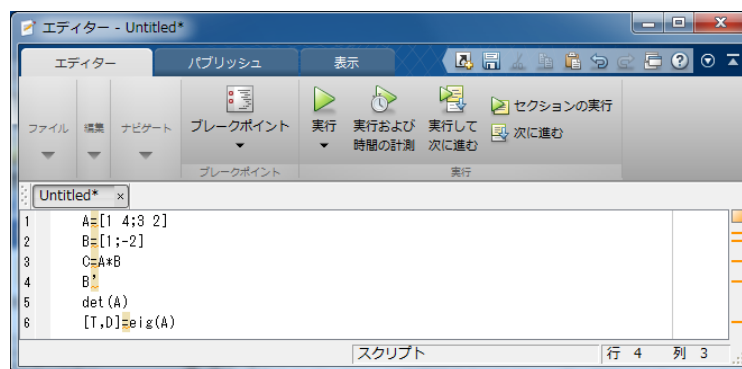


図 3 エディター-Untitled

- 4) 入力終了後に、ファイルを保存する（「ファイル」→「別名で保存」を選択し、ファイル名をたとえば、'test.m'として保存する）。
- 5) 実行する。すなわち、コマンドウィンドウで「>> test」と入力することにより、プログラムが実行され、結果が表示される。

出力結果

```
>> test
A =
     1     4
     3     2
B =
     1
    -2
途中は省略する
```

```
D =
    -2     0
     0     5
>>
```

5) 作成したファイル'test.m'を再度開いて修正したい場合には、コマンドウインドウの傍にある「現在のフォルダー」の test.m をクリックすると、エディターが開く。最初と同様に、ファイルを修正後に保存する。

3. ステップ応答, インパルス応答, 一般の応答

MATLAB の基本演算を組み合わせて各自で目的に合った応用プログラムが作成できる。既に制御系の解析や設計などに用いるプログラム集が種々開発されており、ツールボックスとして有料あるいは無償で提供されている。本稿では control system toolbox(有料)を用いる。

$$y = P(s)u, P(s) = \frac{s+1}{s^3 + 2s^2 + 3s + 1}$$

のステップ応答を計算し、(t, y(t))を図示せよ。ただし、時間 t は[0,20]秒とする。

プログラム 'stepresponse1.m'

```
P=tf([1 1],[1 2 3 1])
```

```
t=linspace(0, 20, 200);
```

```
y=step(P, t);
```

```
figure(1)
```

```
plot(t, y)
```

```
title('step response')
```

```
xlabel('time[s]')
```

```
ylabel('y(t)')
```

```
grid
```

tf, linspace, step, plot などのコマンドの意味を知るには、コマンドウインドウにおいて、「help コマンド」を用いる。すなわち、「>> help tf」, 「>> help step」などと入力する。プログラムの説明(各コマンドの意味をコメント文(%)を先頭に書く)で補足説明している。コメントは不要であれば書かなくても良い。

```
% MATLAB によるステップ応答の数値計算と表示
```

```
% システムの伝達関数 P(s)を与える
```

```
    P=tf([1 1],[1 2 3 1]) % 最初のベクトルが分子係数, つぎのベクトルが分母係数
```

```
%時間を 0.0 から 20 秒までで, 200 点のサンプル数
```

%最後のセミコロンを省くと t の値を画面に表示する.

```
t=linspace(0,20,200);
```

%P(s)のステップ応答を時間 t について数値計算し, y に結果が得られる

```
y=step(P,t);
```

%横軸 t,縦軸 y として, 結果をプロットする

```
figure(1) % 図のウィンドウの番号を指定
```

```
plot(t,y) % グラフを描く
```

```
xlabel('time[s]') % グラフの横軸にtimes[s]を記入
```

```
ylabel('y(t)') % グラフの縦軸にy(t)を記入
```

```
grid % グラフに罫線を描く
```

2節で述べた方法でプログラムを入力し, ファイル名を'steppresponse1.m'として保存する.そして, このファイルを実行する. 以下に復習も兼ねて手順を示す.

- 1) MATLAB を起動する.
- 2) コマンドエディタを用いてプログラムを入力し保存する.



図4 プログラムの入力

- 3) 実行する.

コマンドウィンドウのメニューから '>>stepresponse1.sce' を入力する. これでプログラムが実行され, 結果が表示される.

実行結果

```
>> stepresponse1
```

伝達関数:

```
s + 1
```

```
-----
s^3 + 2 s^2 + 3 s + 1
```

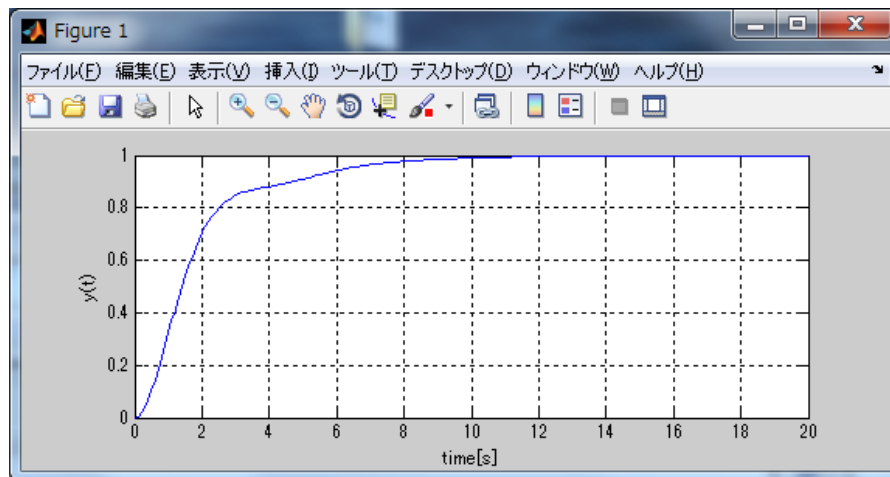


図5 ステップ応答

補足1) インパルス応答を計算するには、`'y=impulse(P,t)'`とする。

補足2) 一般の入力 $u(t)$ に対して $y(t)$ を求めるには、時間ベクトル t に対応した入力ベクトル $u(t)$ を与え、`lsim` を用いる。

```
P=tf([1 1],[1 2 3 1])
t=linspace(0,20,200);
u=sin(t);%入力信号を与える
y=lsim(P,u,t);%シミュレーションを行う
figure(1)
plot(t,y)
xlabel('time[s]')
ylabel('y(t)')
grid
```

4. ボード線図とナイキスト軌跡

4.1 ボード線図の描画

$P(s)$ のボード線図を $[0.01,10]$ [rad/s] の周波数区間で描け

MATLAB の control system toolbox で用意されているボード線図を描くプログラムは、`bode` である。 `bode(P)` で直ちに描画できるが、つぎのプログラムでは周波数応答の値を求めて描画している。

プログラム `'bode_test.m'`

```
P=tf([1 1],[1 2 3 1])
w=logspace(-2,1,200);% サンプル周波数の設定
[ gain, phase]=bode(P,w);%ゲインと位相の計算
figure(1) % 以下はボード線図の表示
```

```
subplot(2, 1, 1)
semi logx(w, 20*log10(gain(:)), 'LineWidth', 2)
xlabel('omega[rad/s]')
ylabel('gain [dB]')
grid
subplot(2, 1, 2)
semi logx(w, phase(:), 'LineWidth', 2)
xlabel('omega[rad/s]')
ylabel('phase [deg]')
grid
```

結果

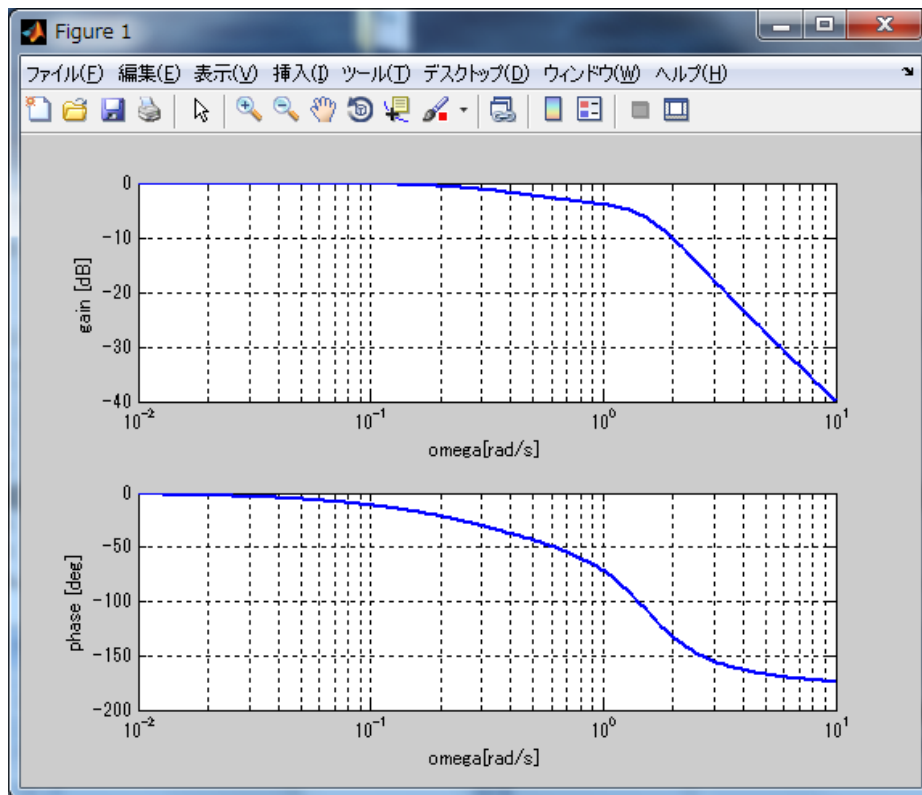


図6 ボード線図

4. 2 ナイキスト線図の描画

ベクトル軌跡 (ナイキスト軌跡) を表示せよ.

nyquist(P)で簡単に描画できるが、つぎのプログラムでは周波数応答を求めて、描画している。

プログラム 'nyquist_test.m'

```
P=tf([1 1], [1 2 3 1])  
w=logspace(-2, 1, 200);  
[realP, imagP]=nyquist(P, w);  
figure(1)  
%plot(realP(:), imagP(:), 'LineWidth', 2)%周波数が正の範囲を描画  
plot(realP(:), imagP(:), 'k', realP(:), -imagP(:), 'k-')%周波数の正と負の範囲を描画  
xlabel('real');  
ylabel('imag');  
grid
```

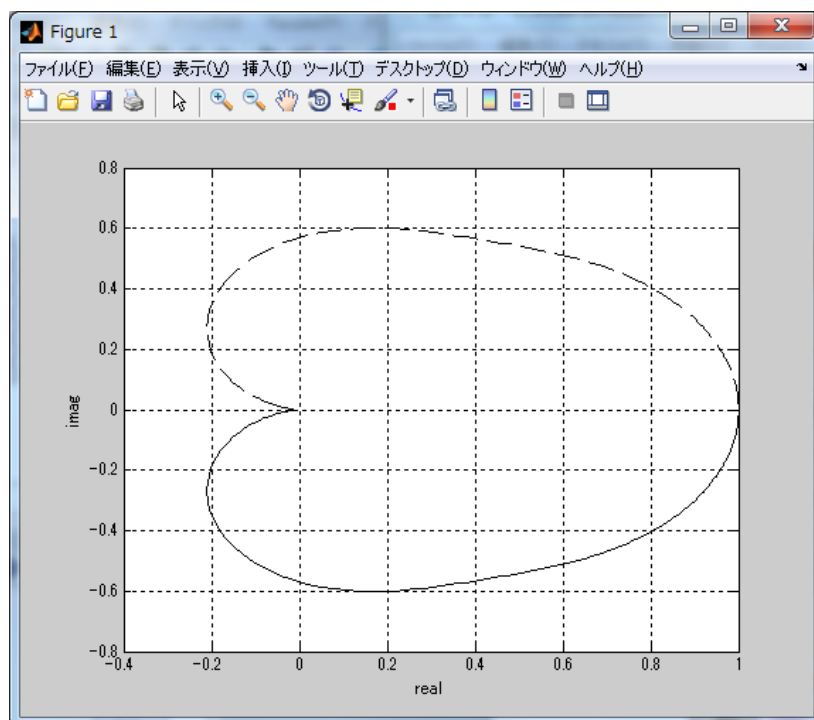


図7 ベクトル軌跡 (ナイキスト軌跡)

5. 根軌跡

根軌跡の計算

例題 $P = \frac{-0.5s+1}{(s+1)(s+3)}, K = \frac{1}{s+5}$ で一巡ループ伝達関数が $L = PK$ のときに根軌跡を描け.

関数 rlocus を用いる. ゲイン K は 0 から Kmax まで描く.

プログラム 'rlocus_test.m'

```
P=tf([-0.5 1],[1 4 3])
```

```
K=tf(1,[1 5])
```

```
L=P*K
```

```
figure(1)
```

```
rlocus(L)
```

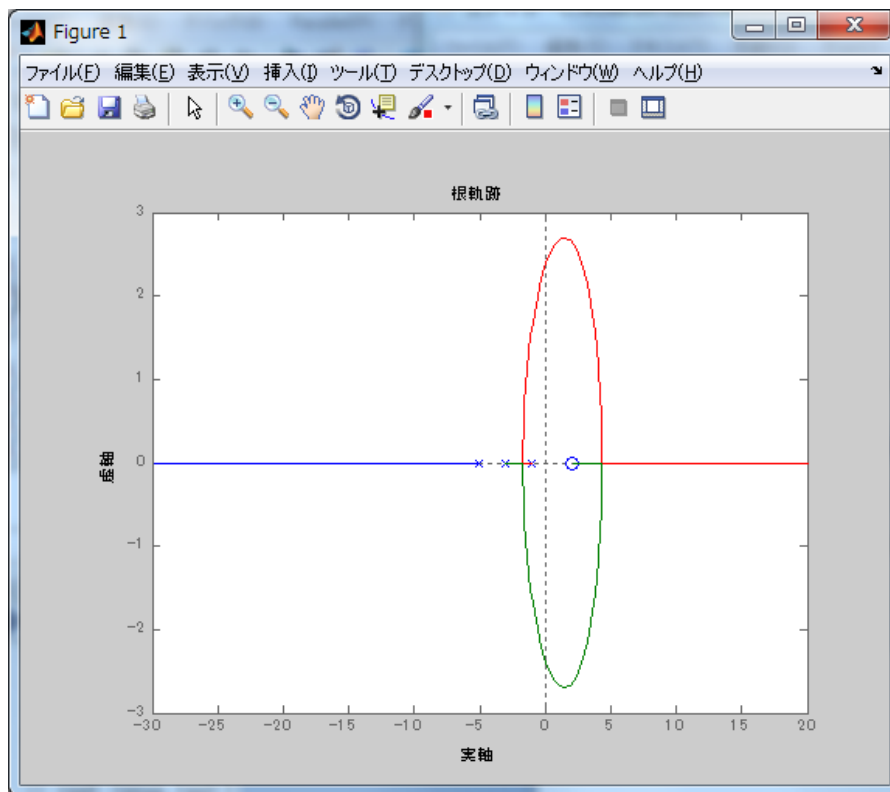


図8 根軌跡と漸近線

6. 設計例

つぎのプラントに対して，ステップ目標値に対して定常偏差がゼロで位相余裕が60度となるように $K(s)$ を設計せよ．

$$P(s) = \frac{1}{(s+1)^2}$$

1) 定常偏差がゼロとなるにはPKが1型であることが必要．そこで，最も簡単な積分補償とする．

$$K(s) = \frac{a}{s}$$

2) 位相余裕が60度になるように，ゲイン調整し a を決める．

2-1) 一巡ループ伝達関数 L (ただし， $a=1$) のボード線図を描く．

$$L = \frac{1}{(s+1)^2 s}$$

図9より，位相余裕が30度で $\omega_c = 0.1 * 2\pi$ [rad/s]である．

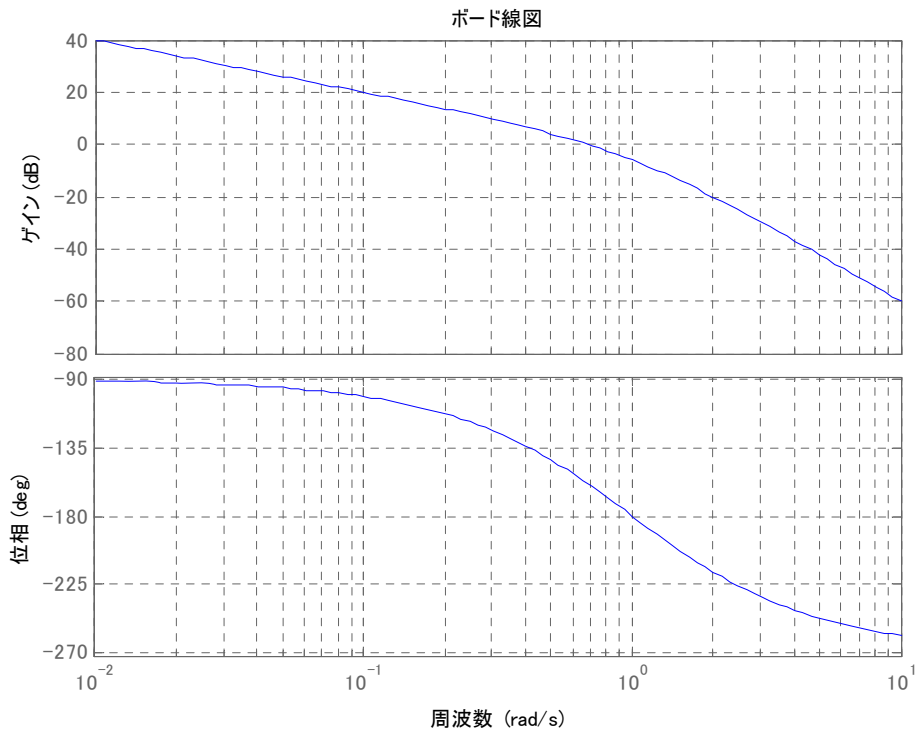


図9 $P(s)$ と $L(s)=P(s)*(1/s)$ のボード線図

図 1 0 に閉ループ系のステップ応答を示す．定常偏差がゼロであるが振動的である．

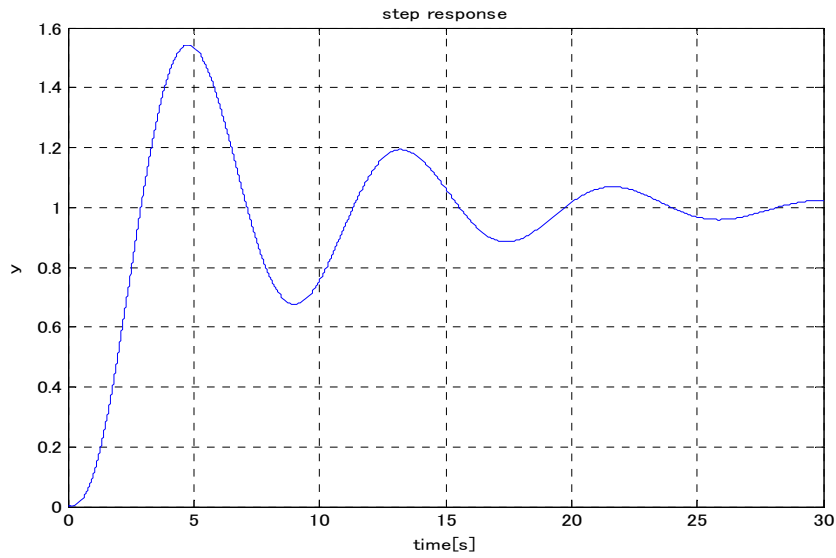


図 1 0 $W=L/(1+L)$ のステップ応答（位相余裕 3 0 度なので振動的，定常偏差= 0）

2-2) 位相余裕を 6 0 度とするには， L のゲインを 11[dB]下に移動するとよい．すなわち， $a = -11[\text{dB}] = 0.28$ である．このとき， $\omega_c = 0.04 * 2\pi [\text{rad/s}]$ である．

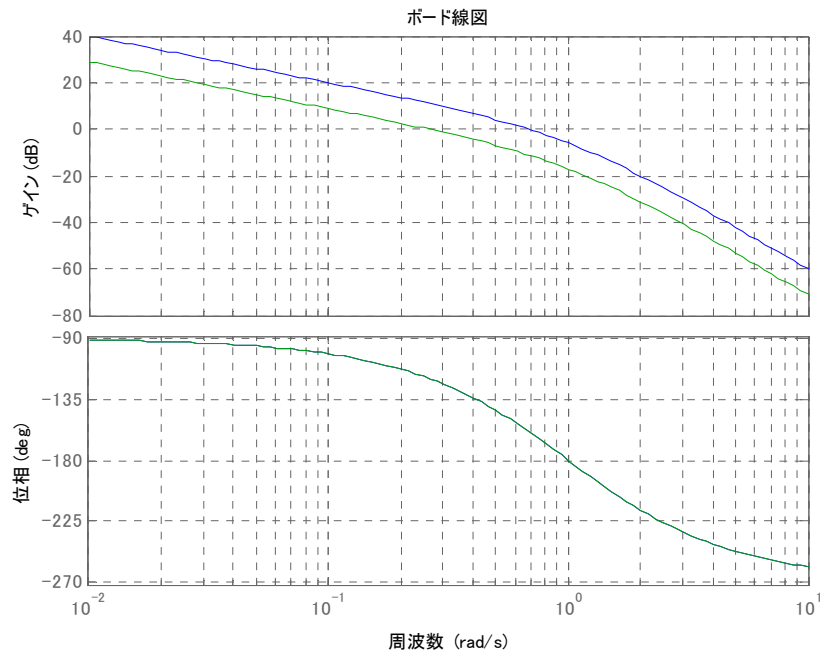


図 1 1 $K=0.28/\text{s}$ の補償により，位相余裕が 6 0 度となる．

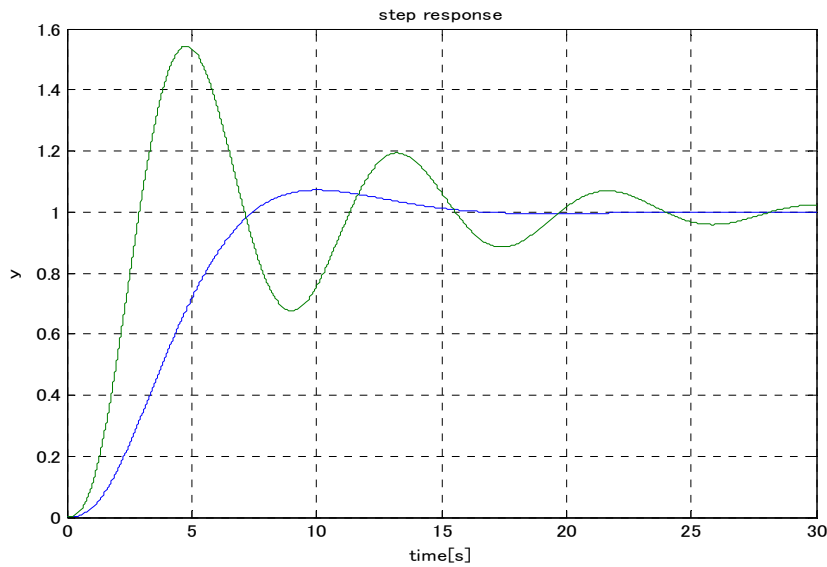


図 1.2 ステップ応答の比較 (ゲイン余裕が 30 度と 60 度の違い)

図 1.2 では応答は振動的でないのが良いが、もう少し応答を速くするように $K(s)$ を設計せよ。ただし、位相余裕が 60 度で定常偏差がゼロとする。

1) $K(s)$ は積分補償に位相進み補償器を追加する。

位相進み補償器は $\omega_{\max} = 0.1 * 2\pi$ [rad/s] で 30 度進めるものを求めると $\alpha = 0.333, T = 2.25$ となり、

$$H(s) = \frac{1 + 2.25s}{3 + 2.25s}$$

となる。よって、制御器は

$$K(s) = \frac{1 + 2.25s}{3 + 2.25s} \times \frac{a}{s}$$

であり、この PK による位相余裕が 60 度になるようにゲイン調整すると $a=1.4$ が得られた。よって、

$$K(s) = \frac{1 + 2.25s}{3 + 2.25s} \times \frac{1.4}{s}$$

このときのボード線図を図 1.3 に示す。 $\omega_c = 0.09 * 2\pi$ から $\omega_c = 0.09 * 2\pi$ [rad/s] となり、ゲイン交差角周波数が約 2 倍になった。このことから、過渡応答の速さが約 2 倍になったと期待できる。実際、これはステップ応答 (図 1.4) により確認できる。

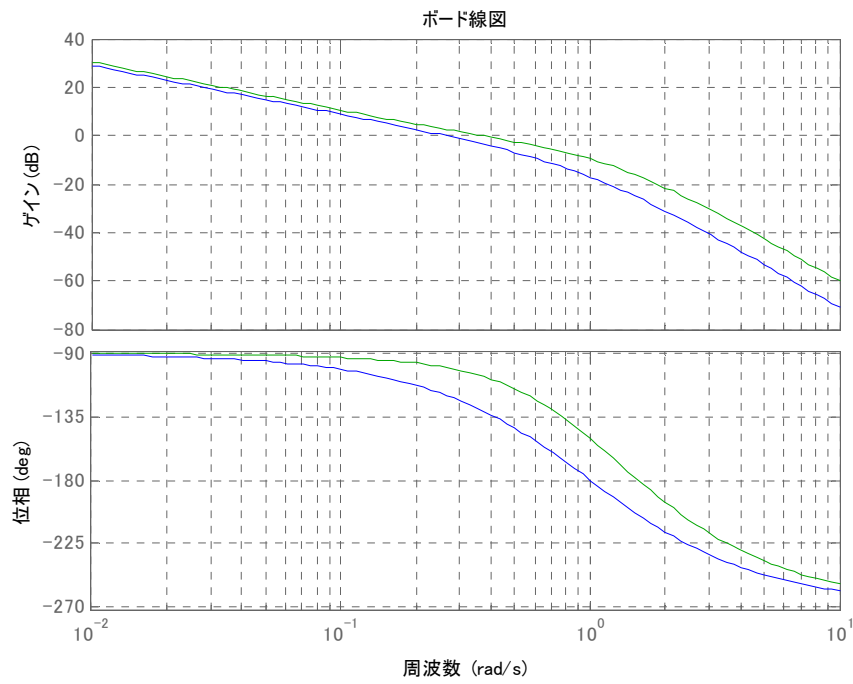


図 1 3 位相進み補償を追加した場合（位相進み補償の追加前後 0.1[Hz]で 30 度進めた）

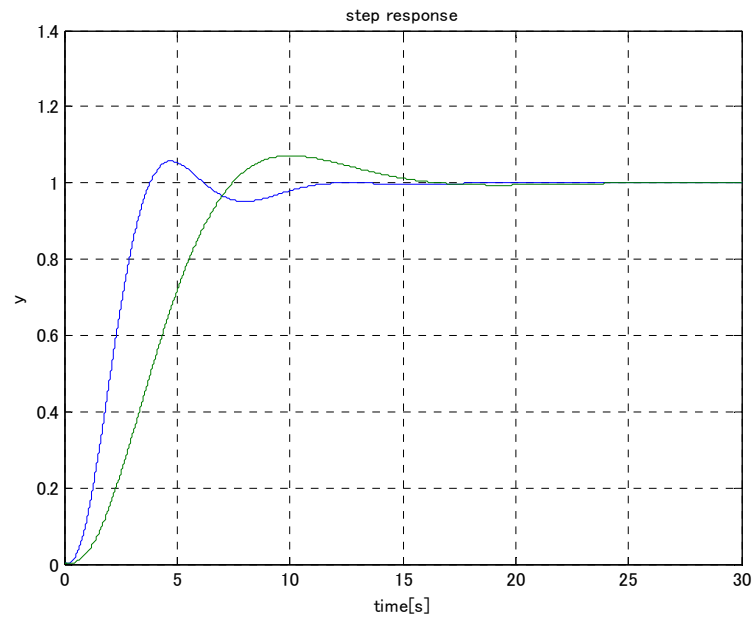


図 1 4 ステップ応答の速応性の改善
 $(\omega_c = 0.09 * 2\pi$ と $\omega_c = 0.04 * 2\pi$, 共に位相余裕は 60 度)

```

プログラム 'control_design.m'
%制御系設計
% プラント
P=zpk([], [-1 -1], 1)
% 制御器
K=tf(1, [1 0])
L=P*K
% L=PKのボード線図 (a=1)
% 周波数区間 [0.01 10]
figure(1)
omega=logspace(-2, 1, 100);
bode(L, omega)
grid
%ステップ応答
T=feedback(L, 1)
t=0:0.01:30;
y=step(T, t);
figure(2)
plot(t, y)
title('step response')
xlabel('time[s]')
ylabel('y')
grid
%La=PKaのボード線図 (a=0.28)
%周波数区間 [0.01 10]
a=0.28
La=L*a;
figure(3)
bode(L, La, omega)
grid
%ステップ応答の比較
t=0:0.01:30;
Ta=feedback(La, 1);
ya=step(Ta, t);
figure(4)

```

```

plot(t, ya, t, y)
title('step response')
xlabel('time[s]')
ylabel('y')
grid
%Lb=PKbボード線図
% 周波数区間 [0.01 10]
Kb=tf([2.25 1], [2.25 3 0])
Lb=P*Kb
figure(5)
bode(La, Lb, omega)
grid
% Lb2=1.4PKbボード線図
% 周波数区間 [0.01 10]
Lb2=P*Kb*1.4
figure(6)
bode(Lb, Lb2, omega)
grid
% ステップ応答の比較
t=0:0.01:30;
Tb2=feedback(Lb2, 1)
yb2=step(Tb2, t);
figure(7);
plot(t, yb2, t, ya)
title('step response')
xlabel('time[s]')
ylabel('y')
grid

```


7. 積分器と定常偏差

システムの型と周波数応答およびステップ応答の関係を調べる。

図15のフィードバック系について考える。外乱 $d=0$ とする。

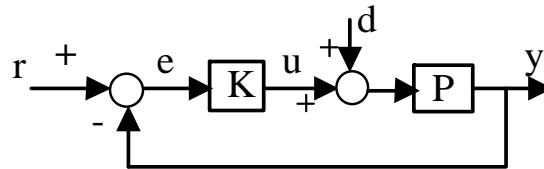


図15 フィードバック制御系

目的：システムの型として0型，1型，2型があり，時間応答，周波数応答の特徴を理解する。

方法：特性方程式が

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

であり，0型，1型，2型となる一巡伝達関数 PK として以下を取り上げる。

$$0 \text{ 型 } PK = \frac{ap_0}{s^2 + 2\zeta\omega_n s + a}$$

$$a = \frac{\omega_n^2}{p_0 + 1}, p_0 > 1$$

$$1 \text{ 型 } PK = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)}$$

$$2 \text{ 型 } PK = \frac{2\zeta\omega_n s + \omega_n^2}{s^2}$$

これらに対して

1) PK のボード線図とベクトル軌跡

2) $T = \frac{PK}{1+PK}$ のステップ応答とランプ応答

3) 相穂感度関数 T と感度関数 $S = \frac{1}{1+PK}$ のゲイン特性 ($T = 1 - S$ の関係あり)

を計算し図示する。

計算結果

0型の場合

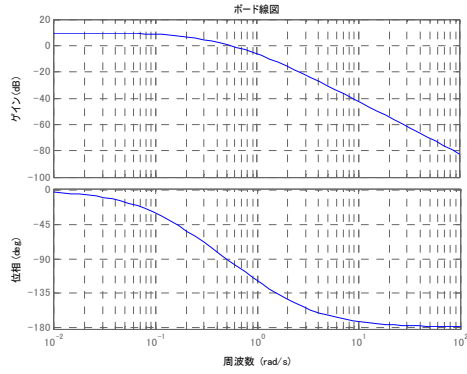


図 1 6 - 1 PK のボード線図 0 型 p0=3

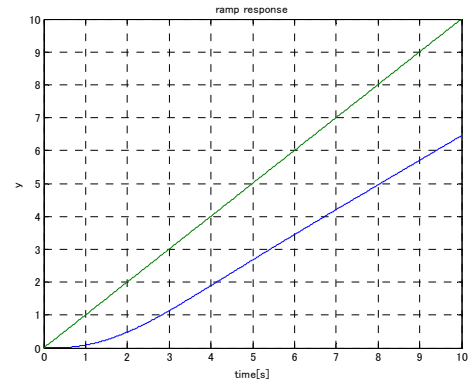


図 1 6 - 4 T のランプ応答 0 型 p0=3

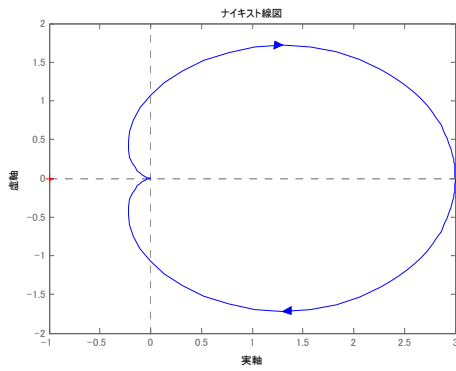


図 1 6 - 2 PK のベクトル軌跡 0 型 p0=3

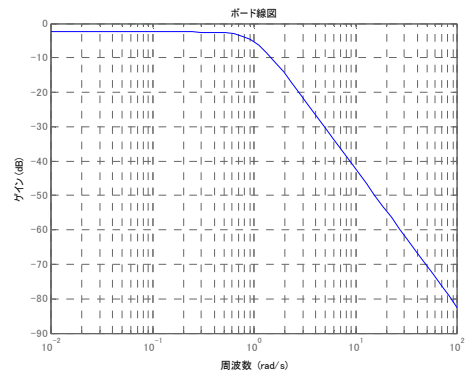


図 1 6 - 5 T のボード線図 0 型 p0=3

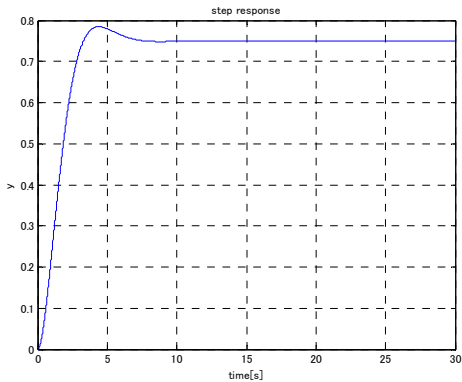


図 1 6 - 3 T のステップ応答 0 型 p0=3

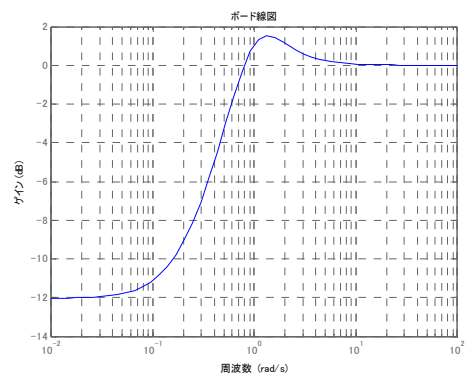


図 1 6 - 6 感度関数 S のゲイン特性 0 型

1型の場合

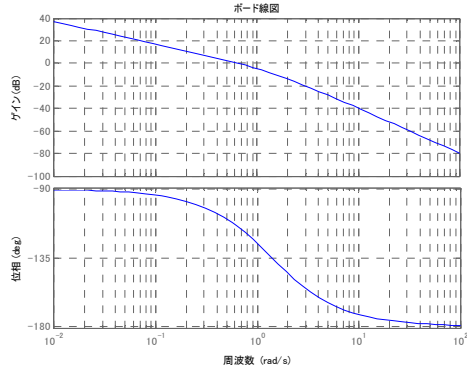


図 1 7 - 1 PK のボード線図 1 型

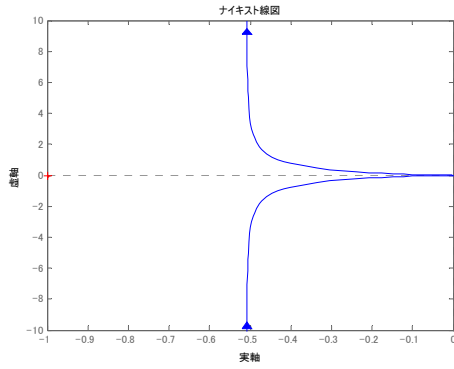


図 1 7 - 2 PK のベクトル軌跡 1 型

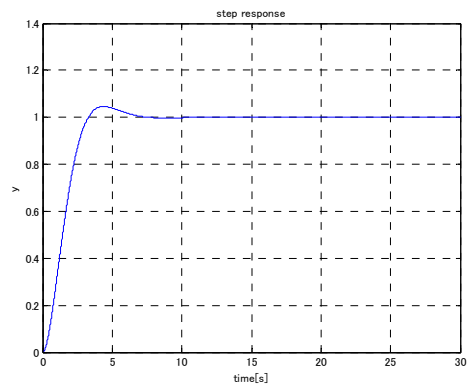


図 1 7 - 3 T のステップ応答 1 型

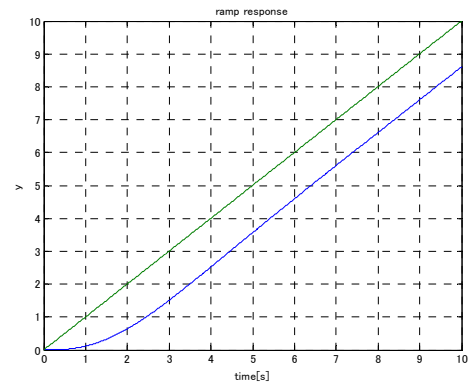


図 1 7 - 4 T のランプ応答 1 型

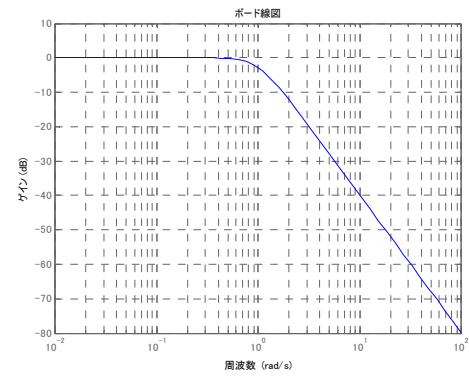


図 1 7 - 5 相補感度 T のゲイン特性 1 型

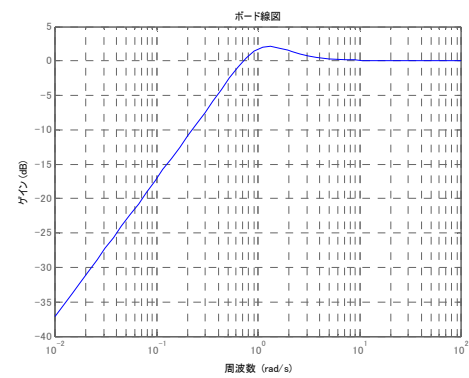


図 1 7 - 6 感度関数 S のゲイン特性 1 型

2型の場合

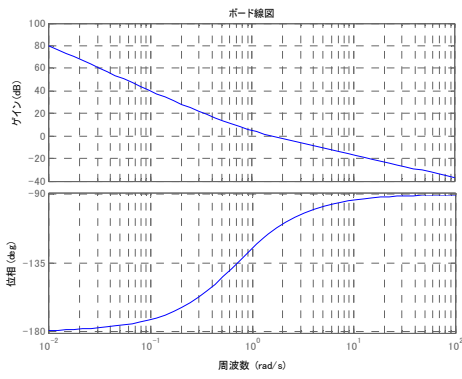


図 1 8 - 1 PK のボード線図 2型

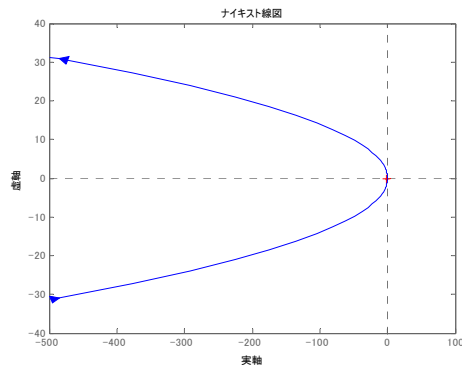


図 1 8 - 2 PK のベクトル軌跡 2型

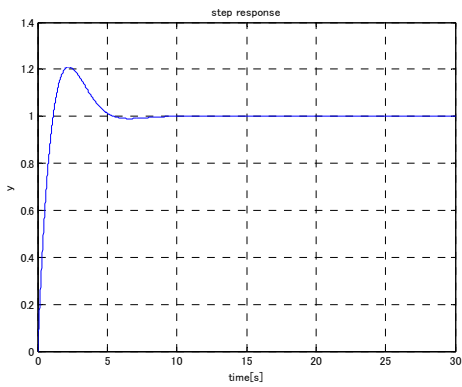


図 1 8 - 3 T のステップ応答 2型

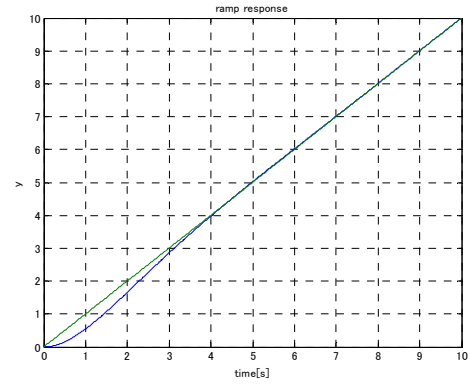


図 1 8 - 4 T のランプ応答 2型

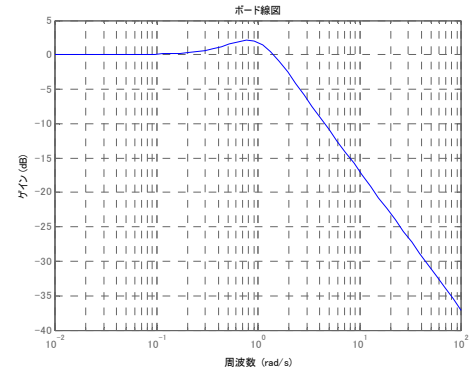


図 1 8 - 5 相補感度 T のゲイン線図 2型

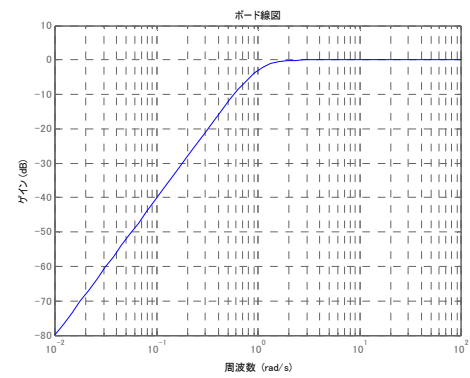


図 1 8 - 6 感度関数 S のゲイン線図 2型

```

プログラム 'control_design_2ji.m'
%制御系設計
% PKの積分器の数= 0, 1, 2 の設定 %%%
typeselect=0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%プラントデータ 2次系
wn=1
zeta=0.7
%計算区間
wmin=0.01;wmax=100;
tmax=30;dt=0.01;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if typeselect==0
% 0型プラント
p0=3
a=wn*wn/(p0+1);
PK=tf(p0*a, [1 2*zeta*wn a])
elseif typeselect==1
% 1型プラント
PK=tf(wn^2, [1 2*zeta*wn 0])
elseif typeselect==2
% 2型プラント
PK=tf([2*zeta*wn wn^2], [1 0 0])
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PKのボード線図
%周波数区間 [wmin wmax]
figure(1)
omega=logspace(log10(wmin), log10(wmax));
bode(PK, omega)
grid
% ナイキスト軌跡
figure(2)
nyquist(PK)
% ステップ応答

```

```

S=1/(1+PK)
T=1-S
t=0:dt:tmax;
y=step(T, t);
figure(3)
plot(t, y)
title('step response')
xlabel('time[s]')
ylabel('y')
grid
%ランプ応答
integral=tf(1, [1 0])
T2=T*integral
t=0:dt:tmax/3;
y=step(T2, t);
figure(4)
plot(t, y, t, t)
title('ramp response')
xlabel('time[s]')
ylabel('y')
grid
% S, Tのゲイン線図
% 周波数区間 [wmin wmax]
figure(5)
bodemag(T, omega)
grid
figure(6)
bodemag(S, omega)
grid

```