Scilab による古典制御の学習

２０１４年４月７日，５月８日
２０１８年４月２日

目次

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計算し，それをグラフに表示することが必要となる．そのような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．ここでは，Scilab を利用した古典制御の解析設計について入門的な内容を説明する．

２章では Scilab 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であるボード線図，ベクトル軌跡，根軌跡の描画例を具体的に示し説明する．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．

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．続編の現代制御による解析設計を[2]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
[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
[4] 佐伯正美、Scilab ボード線図関数プログラム bode2.sci、[2]のホームページに掲載

広島大学　佐伯正美

２．Scilab の導入

数値計算のためのオープンソースプラットホーム Scilab

行列計算などの各種の数値計算やグラフ表示ができ，レポート作成に有用である．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．制御系で用いる基本的な関数は用意されており，たとえば，制御系の時間応答シミュレーション，周波数特性の表示，制御系設計などが行える．このような基本的な関数を用いて自分の目的に合ったプログラムも作れる．

２．１　パソコンにインストールする手順

Scilab 6.0.1 の日本語版をダウンロードしインストールする．

以下の図は Scilab5.4.1 の場合のものであるが，Scilab 6.0.1 も同様である．

２．２　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）Scilab をクリックして起動すると図１のコンソールが表示される．

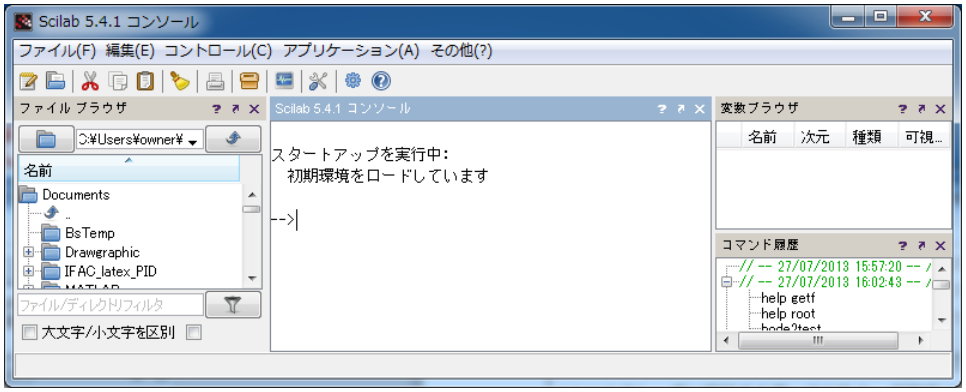

図１　Scilab Console

２）図１の**-->** にコマンドを入力する．

--> help　　**help** コマンドにより使える命令と説明 Window 「ヘルプブラウザ」が開かれる（図２）．コマンドの意味を知るには，「**-->** help コマンド名」とする．

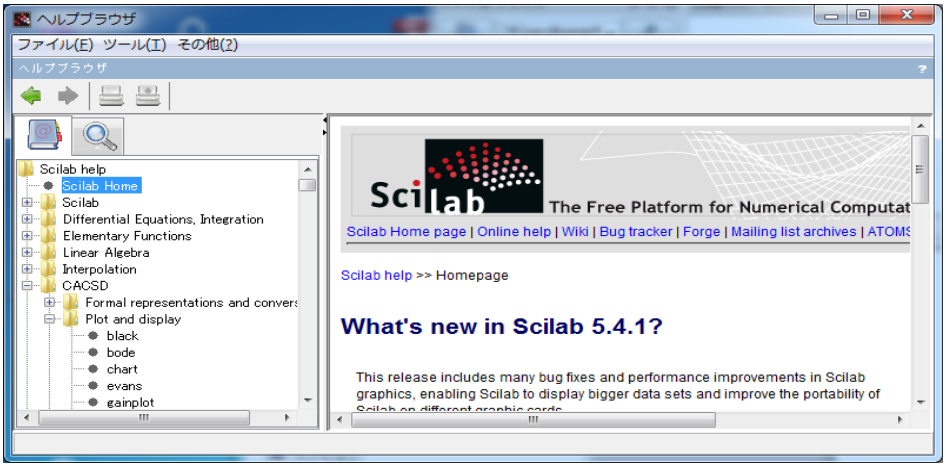

図２　ヘルプブラウザ

図２の Window 内の左に，カテゴリー別に関数が整理されている．関数の使い方の説明が記されている．たとえば，

> **Elementary functions** には，基本的な計算
> たとえば(平方の計算　**sqrt**, 線形システムの定義　**syslin**　など)
> **Linear Algebra** には，行列計算
> たとえば（行列式 **det**, 固有値　**spec** など）
> **CACSD** には，制御系の解析，設計
> たとえば(時間応答シミュレーション　**csim**, 周波数応答計算　**freq** など)

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，行列 A の行列式，および，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]=spec(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
 D  =
    5.    0
```

```
    0    - 2.
T   =
    0.7071068   - 0.8
    0.7071068     0.6
```

2．3　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]=spec(A)
```

1）Scilab をクリックして起動すると図１が表示される．

2）以下で作成するプログラム’test.sce’を保存するディレクトリは，図１のコンソールにおいて，「ファイル」→「現在のディレクトリを変更」で指定できる．自作プログラムを保存したい場合に前もって変更しておくと便利である．

3）図１のコンソールにおいて，メニューの下の左から１番目のメモ用紙をイメージしたボタンをクリックする．SciNotes が画面に表示され(図３)，そこに上記のプログラムを入力する．

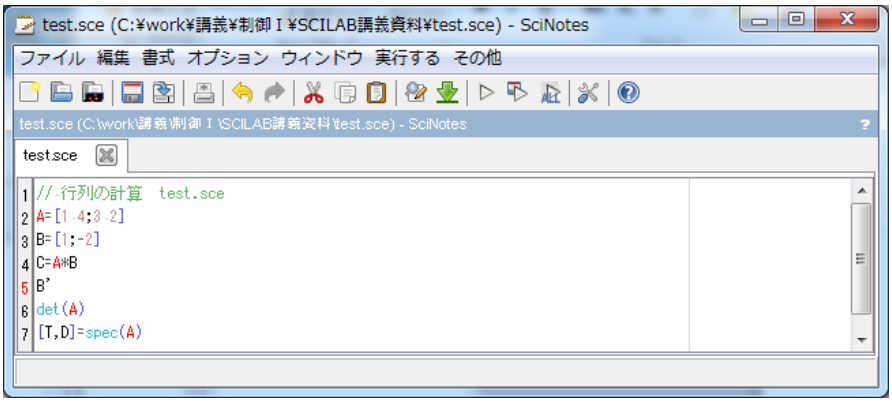

図３　テキストエディタ　SciNotes

4）入力が終了後に，ファイルを保存する（SciNotes の「ファイル」→「保存」を選択し，ファイル名をたとえば，’test.sce’として保存する）．

５）実行する．すなわち，コンソールのメニューから「ファイル」→「実行」で，２．で作成し保存したファイル名（上記の例では'test.sce'）を指定する．これでプログラムが実行され，結果が表示される．

出力結果

```
 A  =
    1.    4.
    3.    2.
 B  =
途中表示は省略
 T  =
    0.7071068   - 0.8
    0.7071068     0.6
 実行は完了しました.
```

５）作成したファイル'test.sce'を再度開いて修正したい場合には，図１のコンソールにおいて，「ファイル」→「ファイルを開く」で'test.sce'を指定すればよい．

３．ステップ応答，インパルス応答，一般の応答

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

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

```
s=poly(0,'s');
P=syslin('c',(s+1)/(s^3+2*s^2+3*s+1))
t=0:0.01:20;
y=csim('step',t,P);
plot(t,y)
xtitle('step response','time[s]','y')
```

poly,syslin,csim などのコマンドの意味を知るには，コンソールにおいて，help コマンドを用いる．すなわち，「help poly」，「help syslin」，「help csim」などを入力する．

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．）

```
//  Scilab 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
```

```
s=poly(0,'s');//    s を多項式の変数の表記に用いる
P=syslin('c',(s+1)/(s^3+2*s^2+3*s+1)) //伝達関数を具体的に与える
//時間を 0.0 から 0.01 の増分で 20 秒まで
//最後のセミコロンを省くと t の値を画面に表示する
t=0:0.01:20;
//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=csim('step',t,P);
//横軸 t,縦軸 y として，結果をプロットする
plot(t,y)
//図のタイトルを'step response'，横軸の説明を'time[s]'，縦軸の説明を'y'と書く
xtitle('step response','time[s]','y')
```

２節で述べた方法でプログラムを作成し，ファイル名を’**stepresponse1.sce**’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) Scilab.exe をクリックして起動する．
2) Scilab エディタを用いてプログラムを入力し保存する．

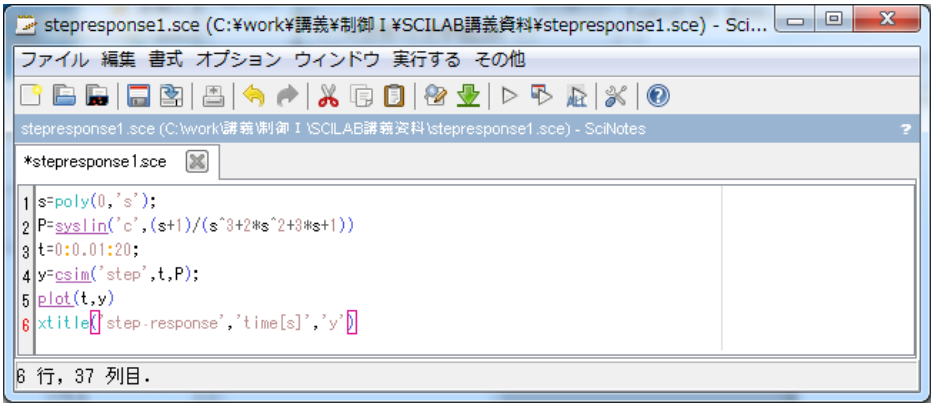

図４　プログラムの入力

3) 実行する．
   コンソールのメニューから「ファイル」→「実行」で，’stepresponse1.sce’を指定する．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

```
 P  =

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

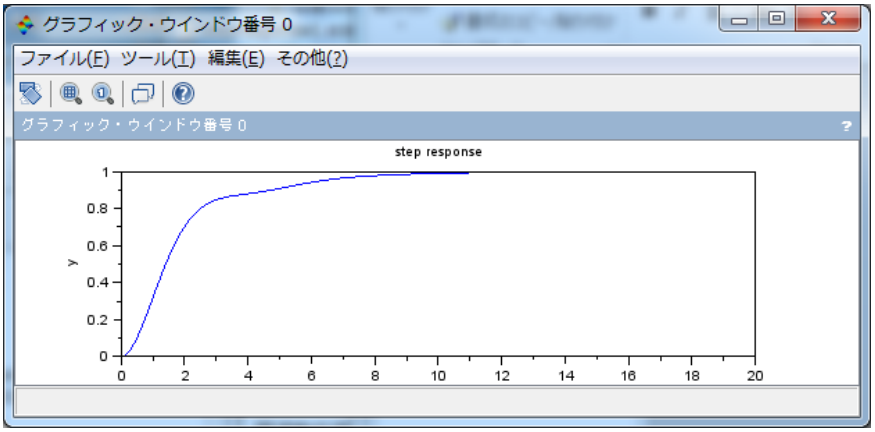

図５　ステップ応答

補足１）関数　csim でインパルス応答を計算するには，y=csim('impulse',t,P);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)を与える．

```
s=poly(0,'s');
P=syslin('c',(s+1)/(s^3+2*s^2+3*s+1))
t=0:0.01:20;
u=sin(t);  //入力信号 u(t)=sin t を与える．
y=csim(u,t,P); //シミュレーションする．
plot(t,y)
xtitle('transient response','time[s]','y')
```

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

Scilab の制御系設計ツール CACSD 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．　Scilab では周波数の単位に基本的に Hz を採用しており，このため bode も横軸が Hz であり横軸を rad/s で描くことができない．ここでは，ボード線図を rad/s で描く自作の関数 bode2.sci を用いる．同様にゲイン特性は Hz で描く gainplot の代わりに rad/s で描く関数 gainplot2.sci を用いる．これらのプログラムは文献[4]に記している．なお，関数のプログラムにはファイル名の拡張子に sci を用い，通常のプログラムには sce を用いることになっている．

プログラム　'bode2test.sce

```
exec('bode2.sci',-1)//  関数を使うのに必要
exec('gainplot2.sci',-1)//関数を使うのに必要
```

```
s=poly(0,'s');
P=syslin('c',(s+1)/(s^3+2*s^2+3*s+1))
omega=logspace(-2,1,100);
scf(1);
bode2(P,omega)
scf(2);
gainplot2(P,omega)
scf(3);
bode(P,omega/(2*%pi))
scf(4);
gainplot(P,omega/(2*%pi))
```

bode2test.sce の bode と bode2 の実行結果

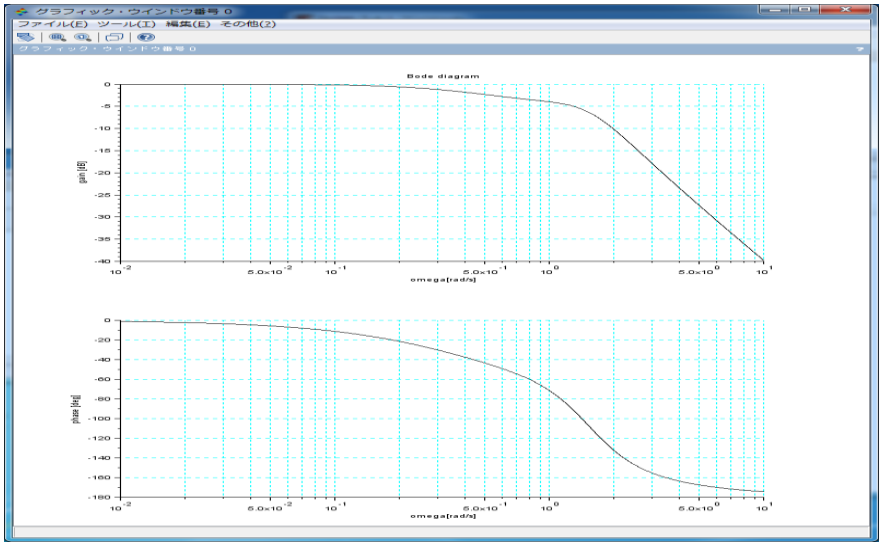

図 6 a　ボード線図（横軸 rad/s）bode2.sci

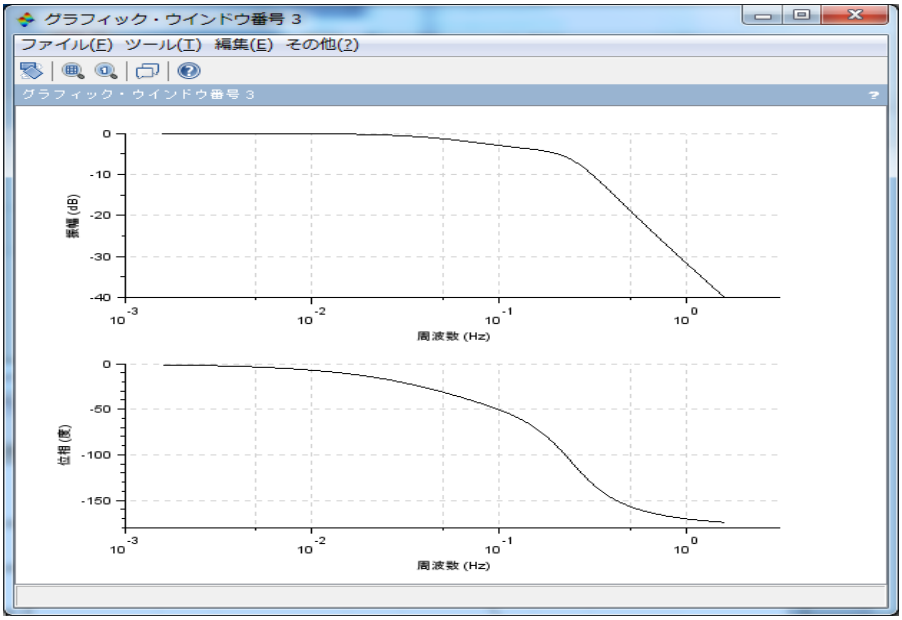

図 6b ボード線図（横軸 Hz）bode

４．２　ナイキスト軌跡の描画

ボード線図をウインドウ１に表示し，ベクトル軌跡（ナイキスト軌跡）を
ウインドウ２に表示せよ．

ナイキスト線図中に示されている周波数の単位は Hz である．rad/s でないので注意が必要である．

プログラム nyquistpolt.sce

```
exec('bode2.sci',-1)
K=1
s=poly(0,'s');
P=syslin('c',(s+1)/(s^3+2*s^2+K*s+1))
scf(1);
omega=logspace(-2,1,100);
bode2(P,omega)
scf(2);
nyquist(P)
```

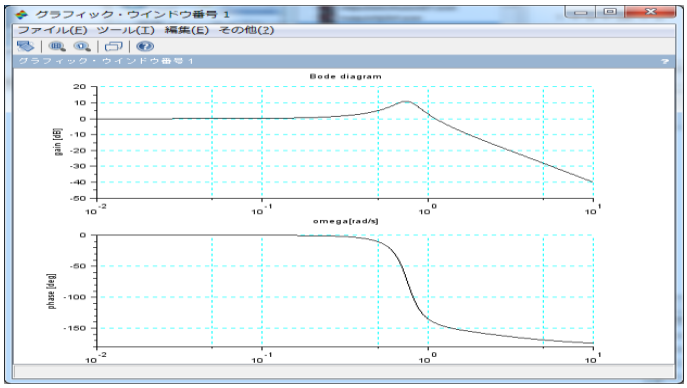

図７－１　ボード線図（ゲイン特性と位相特性）

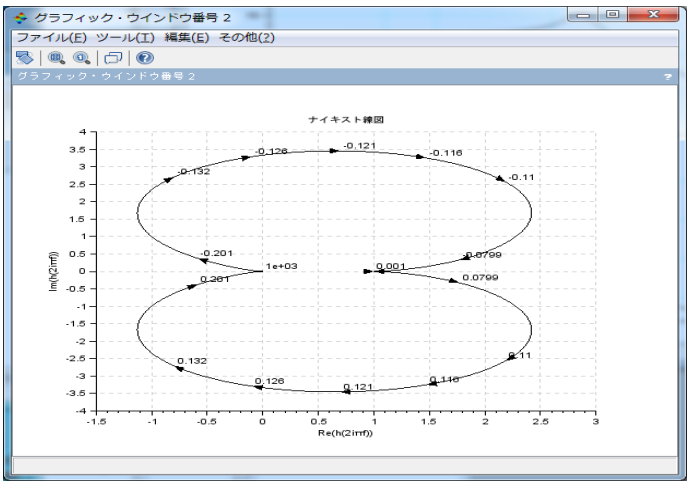

図７－２　ベクトル軌跡（ナイキスト軌跡）

５．時間応答と周波数応答

$$P0(s) = \frac{1}{s^2 + 0.4s + 1}$$

$$P(s) = P0(s)\left(\frac{1}{1+0.05s}\right)^2 = \frac{1}{1+0.5s+1.0425s^2+0.101s^3+0.0025s^4}$$

これら２つの伝達関数の係数を見ても特性の類似度が分からない.

周波数応答や時間応答で類似度が良く分かることを確認する.

プログラム　plant_analysis.sce

```
//plant analysis p0 と p1 の比較
exec('bode2.sci',-1)
//２つのプラント
s=poly(0,'s');
P0=syslin('c',1/(s^2+0.4*s+1))
D=1/(0.05*s+1)^2
P=P0*D

//ステップ応答の比較
t=0:0.01:20;
y0=csim('step',t,P0);
y=csim('step',t,P);
scf(1);
xtitle('step response','time[s]','y')
plot(t,y0,t,y)

//ボード線図の比較
//周波数区間　[0.01 10]
scf(2);
omega=logspace(-2,1,100);
bode2(P0,omega)
bode2(P,omega)

//ベクトル軌跡の比較
//周波数区間　[0.01 10]
scf(3);
nyquist([P0;P],0.01,10)
```

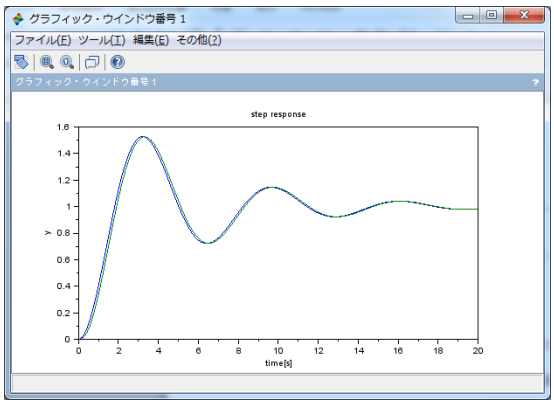

図８－１ステップ応答の比較

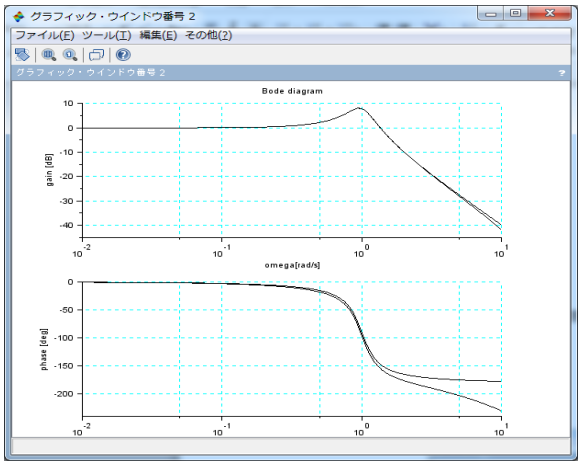

図８－２　ボード線図の比較

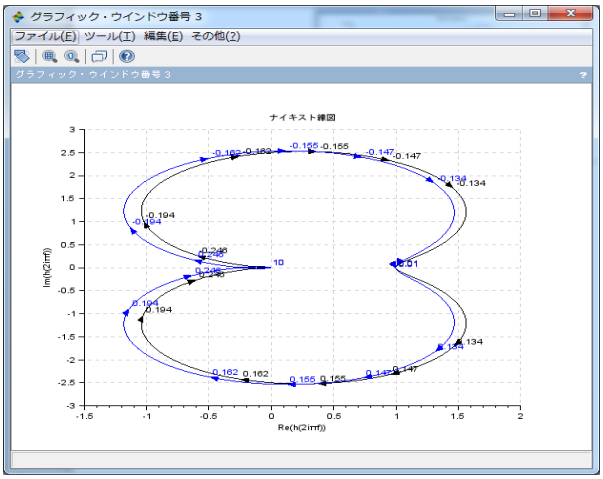

図８－３　ベクトル軌跡の比較

６．根軌跡

根軌跡の計算

例題　$P=\dfrac{1}{s^2+4s+5}, H=\dfrac{1}{s+5}$ で一巡ループ伝達関数が $L=PH$ のときに根軌跡を描け.

Scilab 関数 evans(L,Kmax)を用いる．ゲイン K は 0 から Kmax まで描く.

プログラム　root_locus.sce

```
s=poly(0,'s');
P=syslin('c',1/(s^2+4*s+5));
H=syslin('c',1/(s+5));
L=P*H
evans(L,300)
```

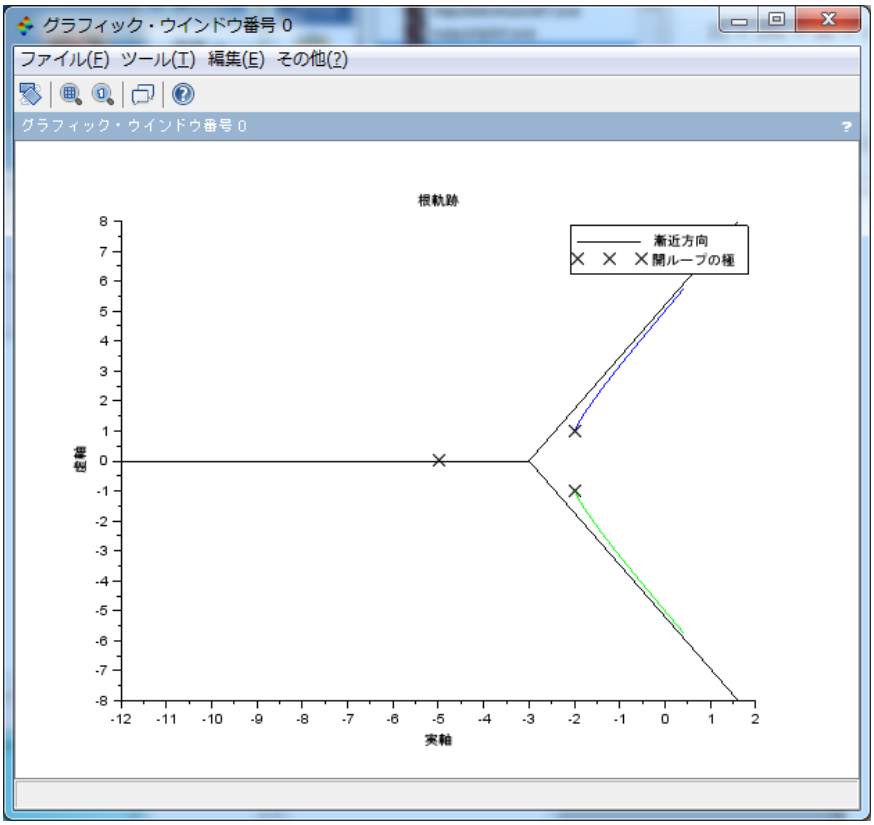

図９　根軌跡と漸近線

７．設計例

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 $L=PK$ 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 $L=PK$ （ただし，a=1）のボード線図を描く．

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

図１０より，位相余裕が３０度で $\omega_c=0.1*2\pi$ [rad/s]である．

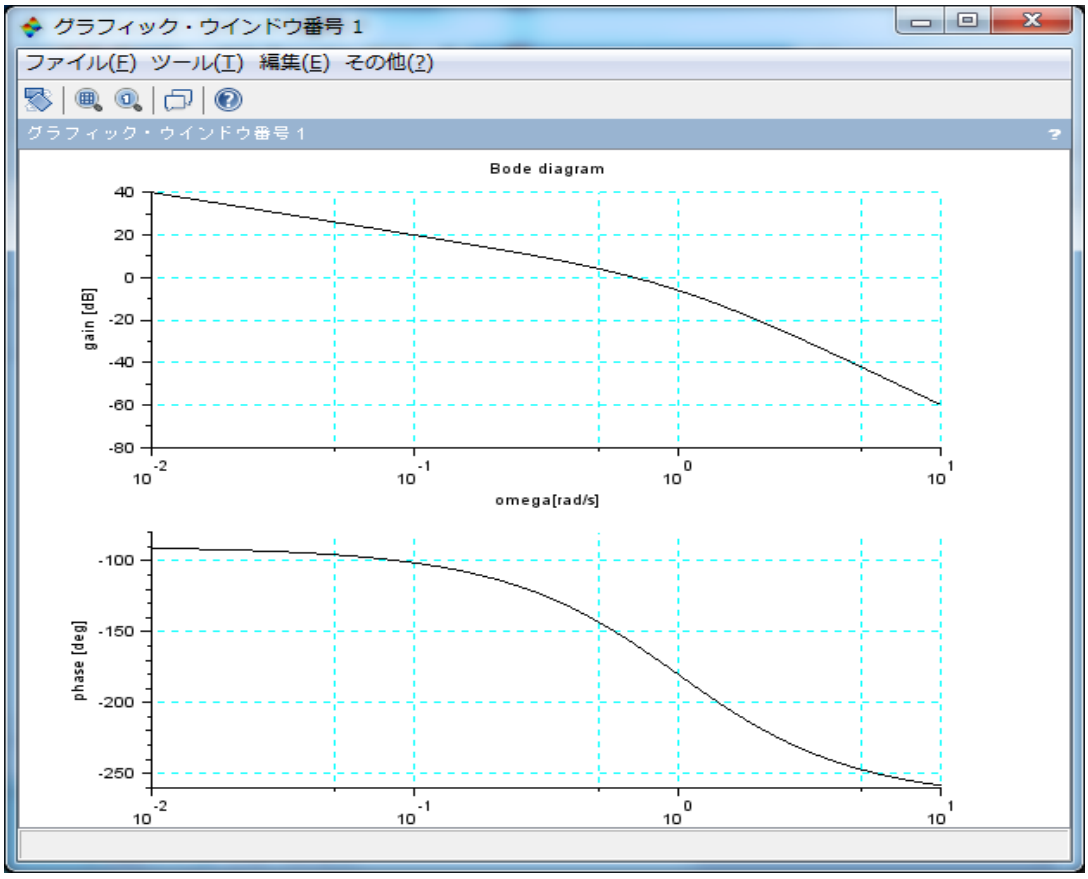

図１０　P(s)と L(s)=P(s)*(1/s)のボード線図

図１１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．

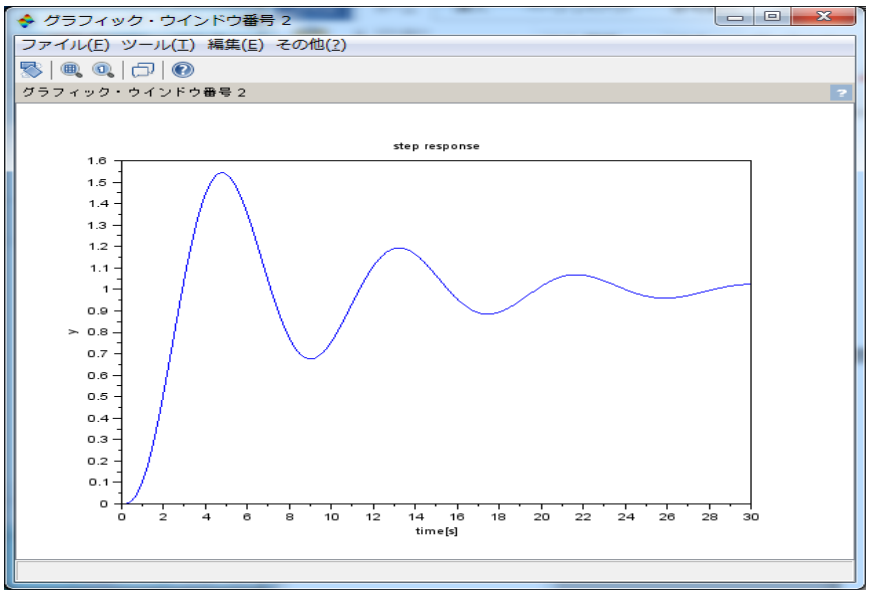

図１１　T=L/(1+L)のステップ応答　（位相余裕３０度なので振動的，定常偏差＝０）

２－２）位相余裕を６０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，$a=-11[dB]=0.28$ である．このとき，$\omega_c=0.04*2\pi$ [rad/s]である．

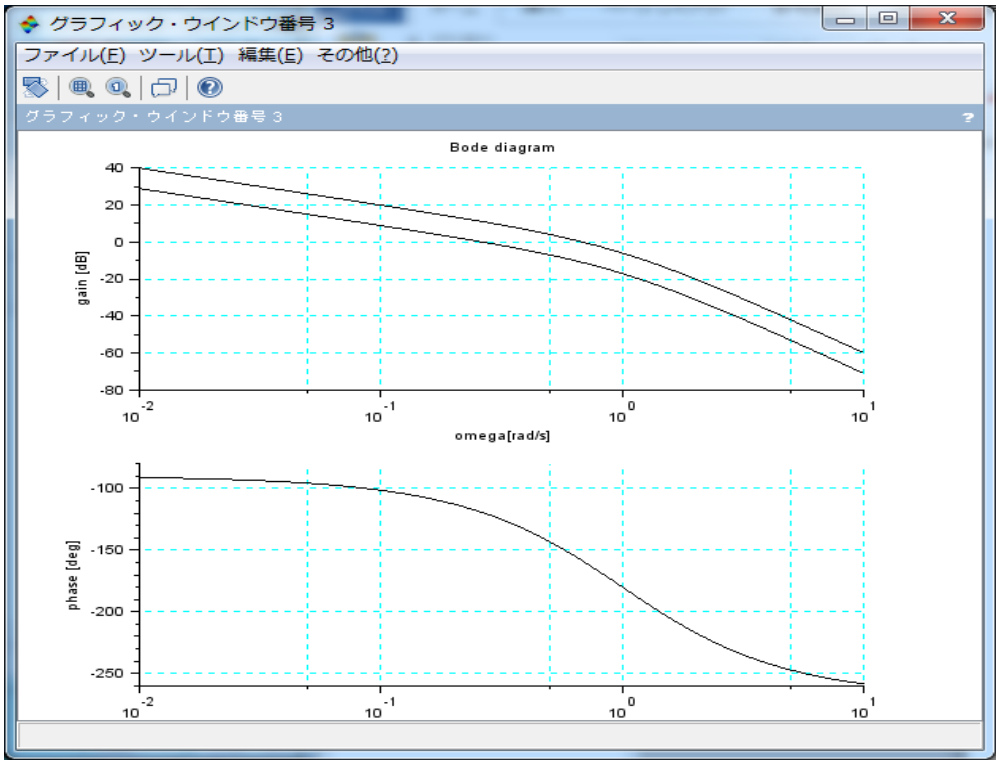

図１２　K=0.28/s の補償により，位相余裕が６０度となる．

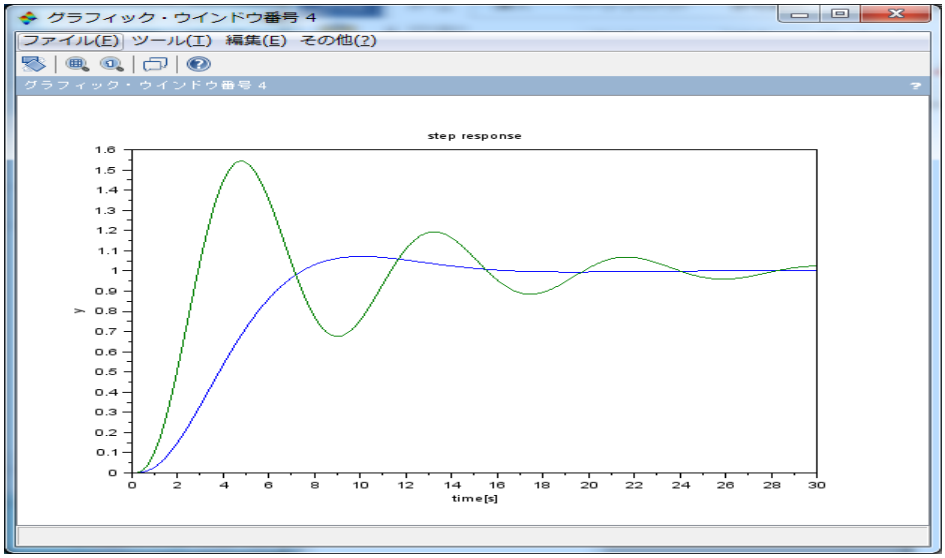

図１３　ステップ応答の比較　（ゲイン余裕が３０度と６０度）

図１３では応答は振動的でない点で良いが，応答を２倍ほど速くするように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１４に示す．$\omega_c = 0.09*2\pi$ [rad/s]となるので，応答が２倍にできたはずである．実際にステップ応答（図１５）より速応性が確認できる．

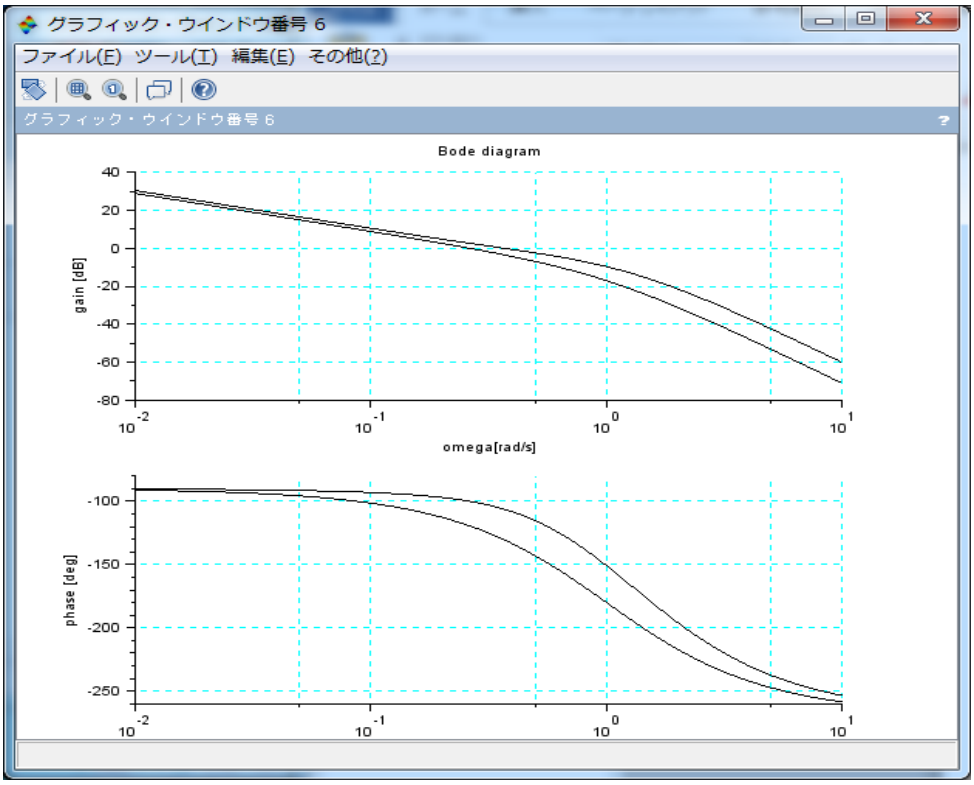

図１４位相進み補償を追加した場合(位相進み補償の追加前後0.1[Hz]で３０度進めた)

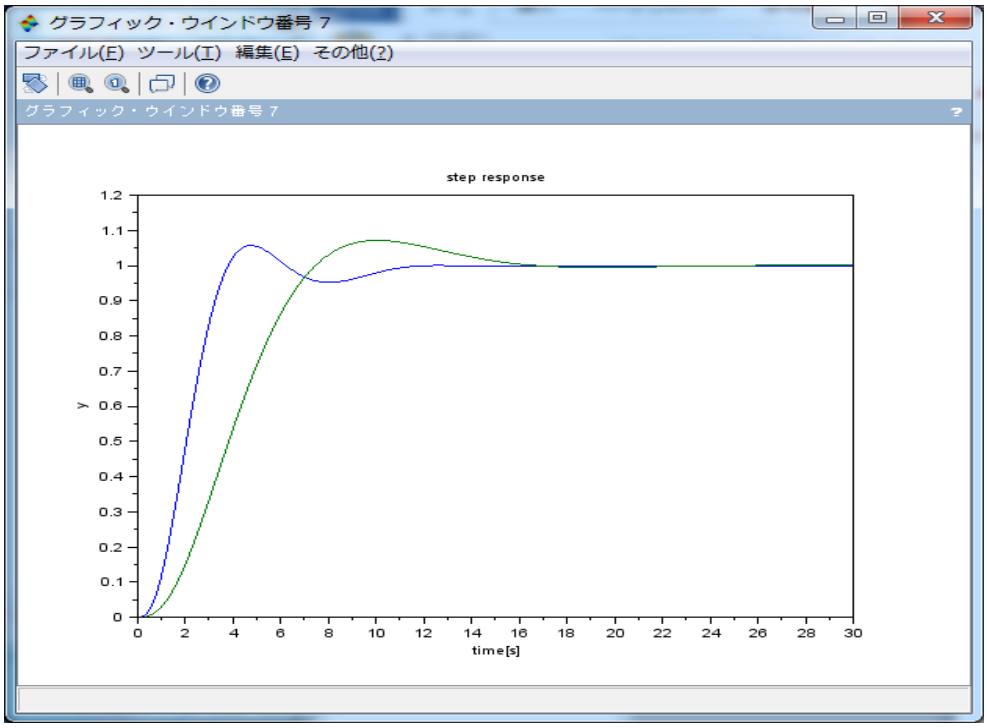

図１５　ステップ応答の速応性の改善

($\omega_c = 0.09*2\pi$ と $\omega_c = 0.04*2\pi$，共に位相余裕は６０度)

プ　ロ　グ　ラ　ム　　control_design.sce

```
//制御系設計
exec('bode2.sci',-1)
// プラント
s=poly(0,'s');
P=syslin('c',1/(s+1)^2)

//制御器
K=syslin('c',1/s)
L=P*K

// L=PK のボード線図(a=1)
//　周波数区間　[0.01 10]
scf(1);
omega=logspace(-2,1,100);
bode2(L,omega)

// ステップ応答
W=L/(1+L)
Wa=La/(1+La)
ya=csim('step',t,Wa);
scf(4);
xtitle('step response','time[s]','y')
plot(t,ya,t,y)

// Lb=PHa/s ボード線図
//　周波数区間　[0.01 10]
Kb=syslin('c',(1+2.25*s)/(s*(3+2.25*s)))
Lb=P*Kb
scf(5);
bode2(La,omega)
bode2(Lb,omega)

// Lb=PHa/s ボード線図
t=0:0.01:30;
y=csim('step',t,W);
scf(2);
xtitle('step response','time[s]','y')
plot(t,y)

// 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;
scf(3);
bode2(L,omega)
bode2(La,omega)


// ステップ応答の比較
t=0:0.01:30;

//　周波数区間　[0.01 10]
Lb2=P*Kb*1.4
scf(6);
bode2(La,omega)
bode2(Lb,omega)

// ステップ応答の比較
t=0:0.01:30;
Wb2=Lb2/(1+Lb2)
yb2=csim('step',t,Wb2);
scf(7);
xtitle('step response','time[s]','y')
plot(t,yb2,t,ya)
```

８．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１６のフィードバック系について考える．外乱 d=0 とする．

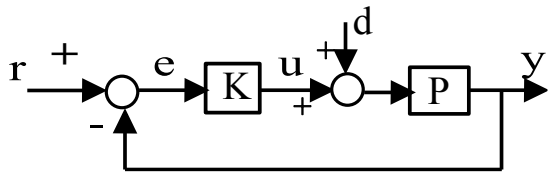

図１６　フィードバック制御系

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$1+PK=0$ が

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

であり，かつ，一巡伝達関数PKが０型，１型，２型となる以下の場合を検討する．

０型　$PK=\dfrac{ap_0}{s^2+2\zeta\omega_n s+a}$，ここに　$a=\dfrac{\omega_n^2}{p_0+1}, p_0>1$

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

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

これらに対して

１）PKのボード線図とベクトル軌跡

２）$T=\dfrac{PK}{1+PK}$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$S=\dfrac{1}{1+PK}$のゲイン特性（T=1-S の関係あり）

を計算し図示せよ．

計算結果０型の場合

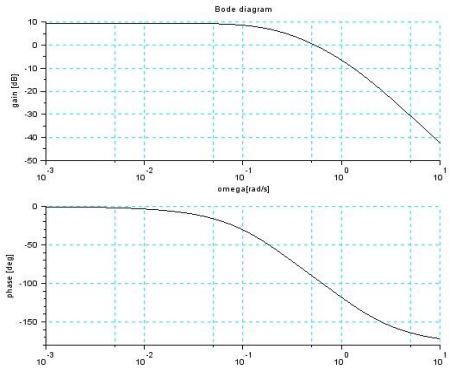

図１７－１　*PK* のボード線図０型 p0=3

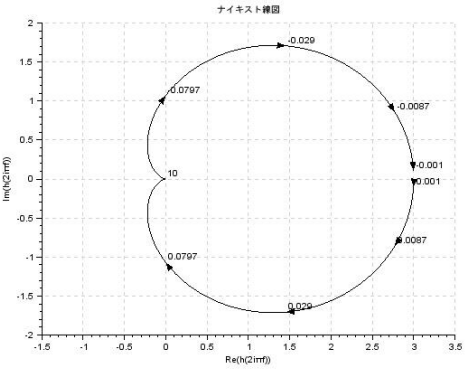

図１７－２　*PK* のベクトル軌跡０型 p0=3

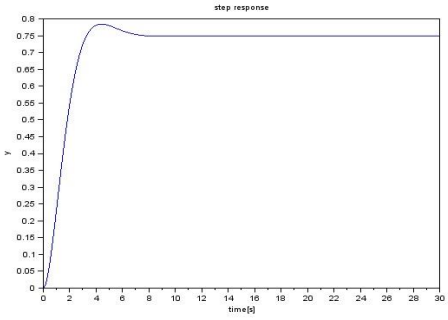

図１７－３ *T* のステップ応答０型 p0=3

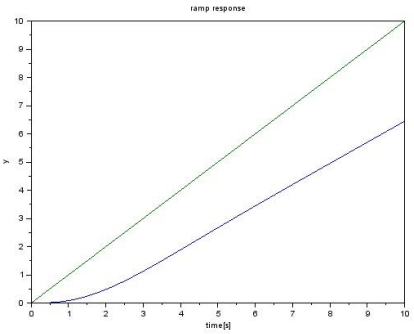

図１７－４　*T* のランプ応答　０型 p0=3

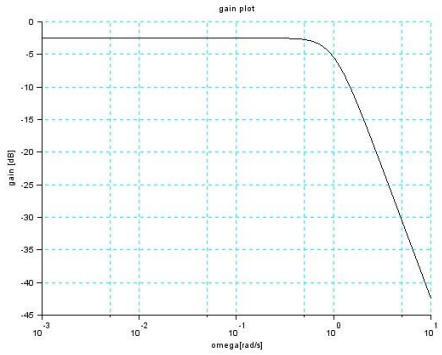

図１７－５　*T* のボード線図　０型 p0=3

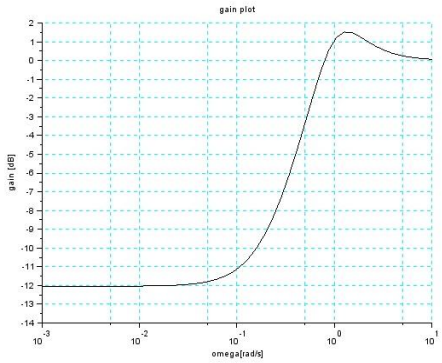

図１７－６　感度関数 *S* のゲイン特性０型

１型の場合

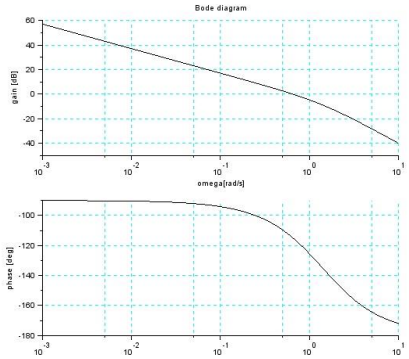

図１８－１ PK のボード線図　１型

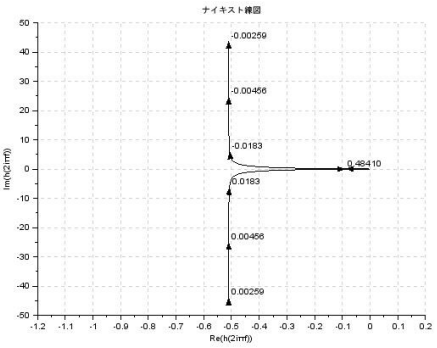

図１８－２ PK のベクトル軌跡　１型

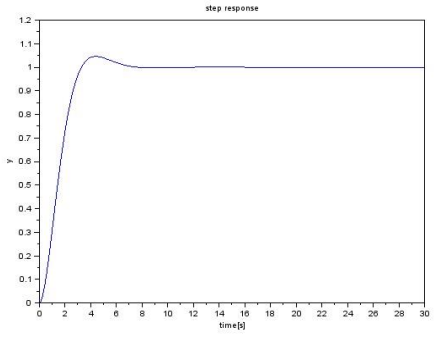

図１８－３ T　のステップ応答　１型

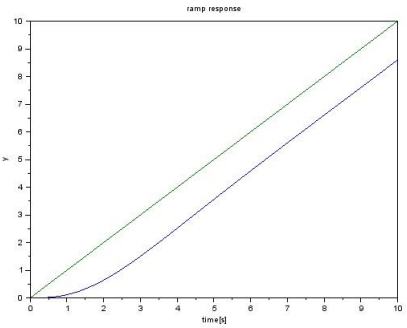

図１８－４ T のランプ応答　１型

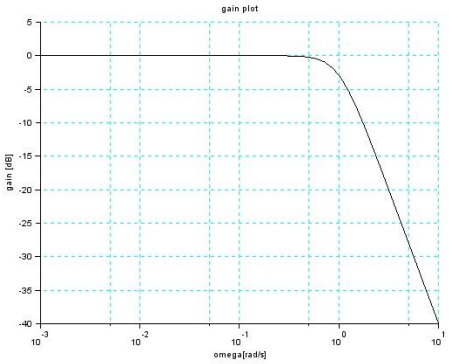

図１８－５相補感度 T のゲイン特性　１型

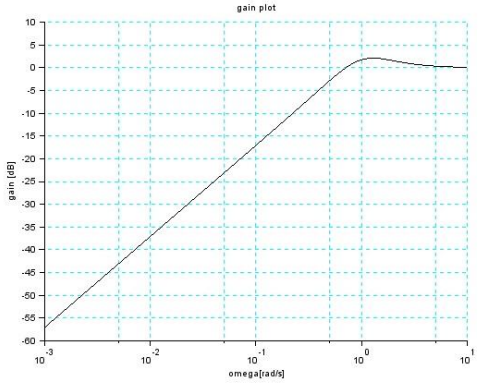

図１８－６　感度関数 S のゲイン特性　１型

２型の場合

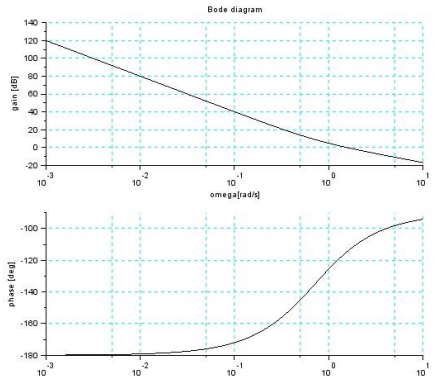

図１９－１ PK のボード線図　２型

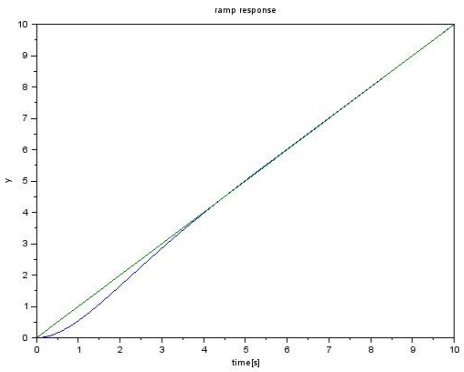

図１９－４ T のランプ応答　２型

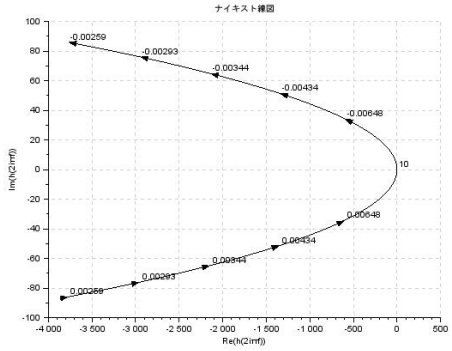

図１９－２ PK のベクトル軌跡　２型

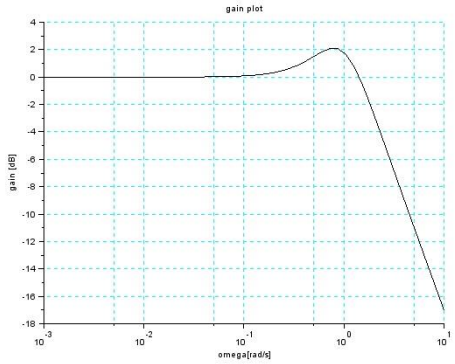

図１９－５　相補感度 T のゲイン線図　２型

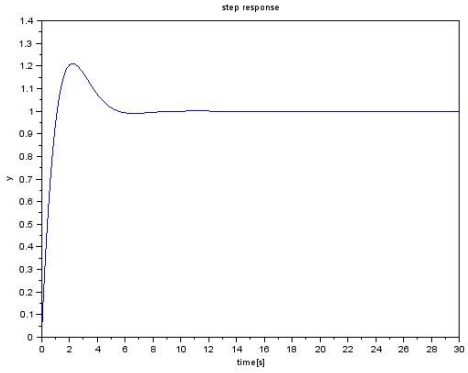

図１９－３　T のステップ応答　２型

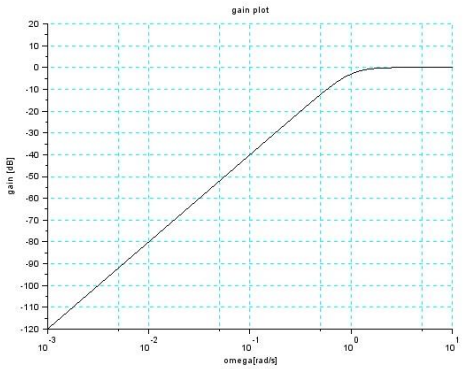

図１９－６　感度関数 S のゲイン線図　２型

プログラム名　contol_design_2ji.sce

```
//制御系設計
exec("bode2.sci",-1)
exec("gainplot2.sci",-1)

// PK の積分器の数＝０，１，２の指定
typeselect=2

//　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;

////////////////////////////////
s=poly(0,'s');
if typeselect==0
////////////////////////////////
// ０型プラント
p0=3
a=wn*wn/(p0+1);
PK=syslin('c',(p0*a)/(s^2+2*zeta*wn*s+a))
elseif typeselect==1
////////////////////////////////////
// 1 型プラント
PK=syslin('c',wn^2/(s*(s+2*zeta*wn)))
elseif typeselect==2
// 2 型プラント
PK=syslin('c',(2*zeta*wn*s+wn^2)/s^2)
end

// PK のボード線図
//　周波数区間　[wmin wmax]
scf(1);
omega=logspace(log10(wmin),log10(wmax));
bode2([PK],omega)

// ナイキスト軌跡
scf(2);
nyquist([PK],wmin,wmax)

// ステップ応答
S=1/(1+PK)
T=1-S
t=0:dt:tmax;
y=csim('step',t,T);
scf(3);
xtitle('step response','time[s]','y')
plot(t,y)
//　ランプ応答
integral=syslin('c',1/s)
T2=T*integral
t=0:dt:tmax/3;
y=csim('step',t,T2);
scf(4);
xtitle('ramp response','time[s]','y')
plot(t,y,t,t)

//　S,T のゲイン線図
//　周波数区間　[wmin wmax]
scf(5);
gainplot2(T,omega)
scf(6);
gainplot2(S,omega)
```

９．付録

古典制御で用いる Scilab 関数の例

モデル P の作成	s=poly(0,'s')
	P=syslin('c',(s+1)/(s^2+s+3))
P1 と P2 の並列結合	P3=P1+P2
P1 と P2 の直列結合	P3=P1*P2
時間 t の指定	t=0:0.01:20
ステップ応答の計算	y=csim('step',t,P)
インパルス応答の計算	y=csim('impulse',t,P)
入力 u に対する応答の計算	y=csim(u,t,P)
グラフ表示	plot(t,y)
描画の winndow No. 3 を開く	xset('window',3)
P(s)のボード線図の描画　（横軸 Hz）	bode(P,wmin,wmax)
P(s)のゲイン線図の描画　（横軸 Hz）	gainplot(P,wmin,wmax)
P(s)のナイキスト線図の描画　（Hz）	Nyquist(P)
一巡関数が L(s)のフィードバック系の根軌跡	Evans(L,Kmax)
自作関数 bode2.sci (横軸 rad/s)を使う場合	exec('bode2.sci',-1);
	w=logspace(-1,2,200);
	bode2(P,w)
自作関数 gainplot2.sci (横軸 rad/s)を使う場合	exec('gainplot2.sci',-1);
	gainplot2(P,w)