

# MATLAB 入門

千葉大学 劉 康志

2006 年 7 月 31 日–8 月 1 日

## 1 MATLAB の立ち上げ

- (1) MATLAB ログをダブルクリック
- (2) コマンドウインド上のダイアログ「file」をクリックし、「新規作成」を選びエディタを立ち上げる。
- (3) エディタで mfile を作成し、名前を付けて保存する。拡張子.m が自動的につけられる。
- (4) コマンドウインドで file 名 (拡張子を省略可) を入力し、リターンを押すと mfile が実行される。例えば、

```
>>filename Return
```

ファイルが大きい場合、メインファイルとサブファイルに分けて編集 作成できる。

## 2 システムデータの与え方

- (1) 伝達関数

$$\begin{aligned} G(s) &= \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_n} \\ &= k \frac{(s - z_0) \dots (s - z_m)}{(s - p_0) \dots (s - p_n)} \end{aligned}$$

については、それぞれ次のように入力する。

```
num=[b0 b1 ... bm]; %分子多項式の係数  
den=[a0 a1 ... an]; %分母多項式の係数  
Gsys=nd2sys(num,den); %分母分子多項式からシステム行列への変換
```

あるいは

```
Z=[z0 z1 ... zm]; %零点  
P=[p0 p1 ... pn]; %極  
k=a;  
Gsys=zp2sys(Z,P,k); %極零点からシステム行列へ  
[A,B,C,D]=unpck(Gsys); %A,B,C,D を取り出す  
G=ss2tf(A,B,C,D,iu); %iu は入力の番号
```

例 1: 伝達関数

$$K(s) = 0.9986 \frac{s + 0.75}{(s + 99.4)(s + 1.42)}$$

はつぎのように入力し, test1.m の名前を付けるとする .

```
Z=[-0.75];  
P=[-99.4 -1.42];  
a=0.9986;  
Ksys=zp2sys(Z,P,a)  
%[A,B,C,D]=unpck(Ksys);  
%[num,den]=ss2tf(A,B,C,D,1)
```

これを実行すると, 状態方程式の係数行列 (MATLAB ではシステム行列という) は Ksys に保存され, 画面上に

```
>> test1  
Ksys =  
  
-100.6336  -11.0632  -0.9997   2.0000  
  11.0632   -0.1864   0.0294     0  
 -0.9997   -0.0294     0         0  
           0         0         0      -Inf
```

が表示される . 係数行列を確認したければ, unpck コマンドを使えばよい .

```
>> [A,B,C,D]=unpck(Ksys)  
A =  
-100.6336  -11.0632  
  11.0632   -0.1864  
B =  
-0.9997  
  0.0294  
C =  
-0.9997  -0.0294  
D =  
0
```

(2) 状態方程式

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

の場合、次のように入力する .

```
A=[a11 a12 ... a1n  
   a21 a22 ... a2n
```

```

    ...
    an11 an2 ... ann];
B=[b11 b12 ... b1m
   b21 b22 ... b2m
   ...
   bn1 bn2 ... bnm];
C=[c11 c12 ... c1n
   ...
   cp1 cp2 ... cpn];
D=[d11 d12 ... d1m
   ...
   dp1 dp2 ... dpm];
Gsys=pck(A,B,C,D);

```

$D = 0$  の場合、省略可 .

## 例 2: 状態方程式モデル

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= [1 \ 2]x \end{aligned}$$

の入力にはつぎの mfile(test2.m) を使えばよい .

```

a=[0 1;-2 -3];
b=[0;1];
c=[1 2];
sys=pck(a,b,c)

```

これを実行すれば , 画面上に次のように表示が出る .

```

>> test2
sys =
    0    1    0    2
   -2   -3    1    0
    1    2    0    0
    0    0    0  -Inf

```

モデル間の変換には Robust Toolbox のコマンドが利用できる .

```

[num,den]=ss2tf(a,b,c,d,iu) %状態方程式から伝達関数へ , iu は入力の番号
[z,p,k]=ss2zp(a,b,c,d,iu)  %状態方程式から極零点へ
[a,b,c,d]=tf2ss(num,den)   %伝達関数から状態方程式へ
[a,b,c,d]=zp2ss(z,p,k)    %極零点から状態方程式へ
[z,p,k]=tf2zp(num,den)    %伝達関数から極零点へ
[num,den]=zp2tf(z,p,k)    %極零点から伝達関数へ

```

### 3 時間応答

#### (1) インパルス応答

```
y=impulse(num,den); %自動表示
or
y=impulse(A,B,C,D,iu)
```

あるいは時間と座標のスケールを指定する .

```
t=0:0.1:10 %0-10秒でシミュレーション,刻みは0.1秒
[y,x,t]=impulse(num,den,t); %指定した時間帯のインパルス応答を計算
plot(t,y); %応答をプロット
title('Impulse Response'); %題をつける
xlabel('Time[sec]'); %x座標のラベル
ylabel('Amplitude') %y座標のラベル
```

例1の系のステップ応答を次の mfile で求められる (test3.m) .

```
Z=[-0.75];
P=[-99.4 -1.42];
k=0.9986;
[num,den]=zp2tf(Z,P,k);
t=0:0.1:10;
[y,x,t]=step(num,den,t);
plot(t,y);
title('Step Response');
xlabel('Time[sec]');
ylabel('Amplitude')
```

(2) ステップ応答インパルス応答と同様, コマンド `impulse` `step` に変えればよい .

### 4 周波数応答

```
w=logspace(-2,4,100); % $10^{-2}$ から  $10^4$ [rad/sec] までの周波数を 100 等分
Gfr=frsp(Gsys,w); %指定した周波数大域で周波数応答を計算, Gsys はシステム行列である .
vplot('bode',Gfr); %周波数応答をボード線図形式で描く
title('Bode Plot');
xlabel('Frequency[rad/sec]');
ylabel('Gain[dB]')
```

ゲインだけを描きたい場合, オプション 'liv,lm' を使う

```
vplot('liv,lm',Gfr)
```

それに, システムツールボックスのコマンドとして

```
[mag, phase, w]=bode(num, den, w)
```

がある .

例 2 の系の周波数応答を計算する mfile : test4.m

```
a=[0 1;-2 -3];
b=[0;1];
c=[1 2];
sys=pck(a,b,c);
w=logspace(-2,3,100);
Gfr=frsp(sys,w);
vplot('bode',Gfr);
title('Bode Plot');
xlabel('Frequency[rad/sec]');
ylabel('Gain[dB]')
```

実習問題 1 : 次の二つの伝達関数について , mfile を書きそのボード線図およびステップ応答を MATLAB で求めよ . また , 両者の応答を比較せよ . なお , 周波数の範囲は  $10^{-2} - 10^3$  [rad/sec] , 時間の範囲は  $0 - 3$  [sec] とする .

$$P_1(s) = 0.9986 \frac{s + 0.75}{(s + 99.4)(s + 1.42)}$$
$$P_2(s) = 1.24 \times 10^{-4} \frac{(s + 0.8)(s + 7725.8)}{(s + 94.3)(s + 1.58)}$$

実習問題 2 : 時間遅れを持つプラント

$$\tilde{P}(s) = e^{-Ts} \frac{1}{s + 10}, \quad 0 \leq T \leq 0.1$$

を乗法変動を持つプラント集合

$$\tilde{P} = P(1 + \delta W), \quad P(s) = \frac{1}{s + 10}, \quad \|\delta\|_\infty \leq 1$$

で表したい . 配布された mfile 「delay」を使って MATLAB で変動重み  $W(s)$  を一つ求めよ . (解答のひとつは  $W = 0.21s / (0.1s + 1)$ )

実習問題 3 : モータと負荷を軸でつないだ 2 慣性系の状態モデルは

$$\dot{x} = \begin{bmatrix} 0 & -k/J_M & 0 \\ 1 & 0 & -1 \\ 0 & k/J_L & 0 \end{bmatrix} x + \begin{bmatrix} 1/J_M \\ 0 \\ 0 \end{bmatrix} u$$
$$y = [1 \ 0 \ 0]x$$

で与えられる . その中 ,  $J_M, J_L, k$  はそれぞれモータの慣性モーメント , 負荷の慣性モーメントと軸のバネ定数を表す . その値の範囲は

$$J_M = 1 \pm 0.2, \quad J_L = 0.8(1 \pm 0.2), \quad k = 1000(1 \pm 0.2)$$

である . 配布された mfile で各パラメータが変化したときの伝達関数ボード線図を描き , 影響の大きいものを特定せよ .

実習問題 4 : 配布された mfile でハードディスクの  $\mathcal{H}_\infty$  設計を行え .

実習問題 5 : 配布された mfile でハードディスクの  $\mu$  設計を行え .

## 5 2 慣性系の位置制御

物理モデル： $J_M$ 、 $J_L$  はモータ及び負荷の慣性モーメント、 $\theta_M$ 、 $\theta_L$  はそれぞれの回転角、 $\omega_M$ 、 $\omega_L$  はそれぞれの角速度で、 $k$  は軸のバネ定数である。

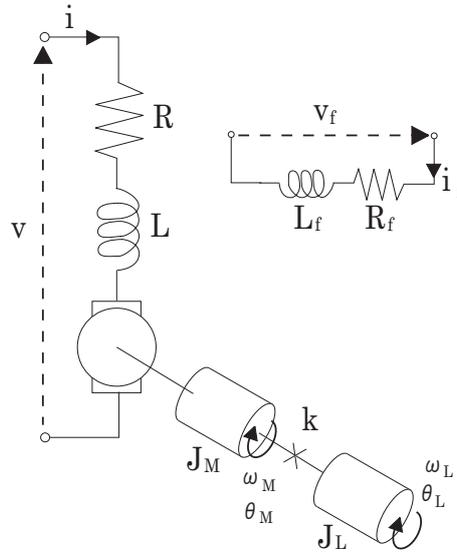


図 1: 電気 - 2 慣性システム

状態方程式：

$$\frac{d}{dt} \begin{bmatrix} \omega_M \\ \omega_L \\ \theta_M \\ \theta_L \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{k}{J_M} & \frac{k}{J_M} \\ 0 & 0 & \frac{k}{J_L} & -\frac{k}{J_L} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_M \\ \omega_L \\ \theta_M \\ \theta_L \end{bmatrix} + \begin{bmatrix} \frac{1}{J_M} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tau \quad (1)$$

$r$  は目標角度である。観測できる値はモータの角速度  $\omega_M$  及びモータの角度  $\theta_M$  であるため、出力方程式は以下ようになる。

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_M \\ \omega_L \\ \theta_M \\ \theta_L \end{bmatrix} \quad (2)$$

一般化プラント： $w_1$  は入力外乱、 $W_3$  はその重み、 $w_2$  は目標値、 $z_1$  は追従誤差  $r - \theta_L$  に関する評価出力、 $W_1$  はステップ状目標値と入力外乱の道徳性を表す重み、 $z_2$  は入力トルク  $\tau$  の評価出力、 $W_2$  はその重み、 $z_3$  は負荷の角速度  $\omega_L$  の評価出力、 $W_4$  はその重みである。

一般化プラント  $G$  は Fig.3 のようになる。 $\theta_L$  と  $\omega_M$  も評価したいため、プラント  $P$  の出力方程式を以下のように書き換えておく。

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_M \\ \omega_L \\ \theta_M \\ \theta_L \end{bmatrix} \quad (3)$$



$$W_2 = \frac{s^2 + 2.0 \times 10^2 s + 2.0 \times 10^5}{s^2 + 7.5 \times 10^3 s + 5.4 \times 10^7} \quad (8)$$

$$W_3 = 0.0001 \quad (9)$$

$$W_4 = \frac{s^2 + 2.0 \times 10^2 s + 2.0 \times 10^5}{s^2 + 7.5 \times 10^3 s + 5.4 \times 10^7} \quad (10)$$

実習問題 6 : 配布された mfile で 2 慣性系の  $\mathcal{H}_\infty$  設計を行え .

## 6 ハードディスクの $\mathcal{H}_\infty$ 制御設計プログラム

### 1. $\mathcal{H}_\infty$ 制御設計メインファイル : design.m

```
clear
format short e
w=logspace(-2,6,100);
%一般化プラントの作成
gplant                %サブプログラムを読み込む
%H 制御器の設計
disp('Hoo Controller Design with hinfsys')
K=[];
controller;          %サブプログラムを読み込む
disp('Hit any key to plot Bode diagram of Hoo controller')
pause
if isempty(K)==1
    break;
end;
%制御器ボード線図の表示
contplot;           %サブプログラムを読み込む
disp('Hit any key to start the simulation')
pause
%シミュレーション
simu                %サブプログラムを読み込む
disp('Hit any key to end the program')
pause
close all
clc
```

### 2. データの指定と一般化プラントの設定 : gplant.m

```
%プラントと重みの定義
Psys=nd2sys([1],[1 0 0],3.87e7);
W1sys=nd2sys([1 125.7],[1 1e-4],5e-1);
W2num1=[1 1e+4 5.7e+7];
W2dencof=1;
W2den1=[1 1.2e+4 4.04e+8];
```

```

W2sys1=nd2sys(W2num1,W2den1);
W2sys=mmult(W2sys1,W2sys1,23.9);
W3sys=0.1;
W4sys=nd2sys([10 10*0.5*5e4],[1 10*5e4]);
%重み関数のボード線図
Pfr=frsp(Psys,w);
W1fr=frsp(W1sys,w);
W2fr=frsp(W2sys,w);
W3fr=frsp(W3sys,w);
W4fr=frsp(W4sys,w);
figure(1);
vplot('liv,lm',W1fr,W2fr,W3fr,W4fr);
title('Weighting Functions')
xlabel('Frequency [rad/s]')
ylabel('Gain')
grid on
    display('Hit any key to start the generation of Gp')
    pause
%一般化プラントを作成
systemnames='Psys W1sys W2sys W3sys W4sys';
inputvar='[w1; w2; u]';
outputvar='[W1sys; W2sys; W4sys; w2+Psys]';
input_to_W3sys='[w1]';
input_to_Psys='[u+W3sys]';
input_to_W1sys='[Psys+w2]';
input_to_W2sys='[Psys]';
input_to_W4sys='[u]';
sysoutname='Gpsys';
cleanupysic='yes';
sysic;

```

### 3. $\mathcal{H}_\infty$ 制御器の設計 : controller.m

```

%プログラム'hinfsyn'を使う
    glow=0;ghigh=10;tol=1e-2; % 反復時のノルム下界 ,
                                上界と許容誤差
while isempty(K)==1 & ghigh<1e6;
    [K,CL,gopt]=hinfsyn(Gpsys,1,1,glow,ghigh,tol);
    if isempty(K)==1 & ghigh<1e6;
        ghigh=ghigh*10;          %ghigh を 10 倍に設定
        tol=ghigh*1e-3;
    end
    clear CL ghigh glow tol
else
    disp(['No solution is found'])

```

```
end;
```

#### 4. 制御器のボード線図を描く : contplot.m

```
Kfr=frsp(K,w);  
figure(2);      %2番目の図ウインドを開く  
subplot(2,1,1)  
vplot('liv,lm',Kfr);  
title('Bode plot of Controller')  
xlabel('Frequency [rad/s]')  
ylabel('Gain')  
grid;  
subplot(2,1,2)  
vplot('liv,p',Kfr);  
xlabel('Frequency [rad/s]')  
ylabel('Phase(degrees)')  
grid;
```

#### 5. シミュレーション用閉ループ系の作成 : closedlp.m

```
systemnames='Psys K';  
inputvar='[w; r]';  
outputvar='[Psys+r; K]';  
input_to_Psys='[K+w]';  
input_to_K='[Psys+r]';  
sysoutname='Clsys';  
cleanupysic='yes';  
sysic;
```

#### 6. $\mathcal{H}_\infty$ 制御器を用いるシミュレーション : simu.m

```
closedlp;  
[pa,pb,pc,pd]=unpck(Psys);  
[ka,kb,kc,kd]=unpck(K);  
[da,db,dc,dd]=unpck(sel(Clsys,1,1));  
[ua,ub,uc,ud]=unpck(sel(Clsys,2,1));  
%外乱応答  
t=[0:0.0001:0.1];  
[y,x,t]=step(da,db,dc,dd,1,t);  
[u,x,t]=step(ua,ub,uc,ud,1,t);  
figure(3)  
subplot(2,1,1)  
plot(t,y)  
axis([0 0.1 0 16])  
xlabel('Time[s]')
```

```

ylabel('Amplitude')
title('Step disturbance response')
grid on;
%制御入力
subplot(2,1,2)
plot(t,u)
axis([0 0.01 -1.7 0])
xlabel('Time[s]')
ylabel('u[V]')
title('Input')
grid on

```

## 7 ハードディスクの $\mu$ 設計プログラム

### 1. $\mu$ 制御設計メインファイル : musyn.m

```

clear
format short e
DK_DEF_NAME='dk_defin';
w=logspace(-2,6,100);
%一般化プラントの作成
gplant
%D-K 反復による mu 設計
disp('Start D-K iteration')
dkit %D-K 反復設計を開始
K=k_dk2suffix; %2 回目の制御器をプロット
pause
disp('Hit any key to plot Bode diagram of mu controller')
contplot;
%シミュレーション
pause
disp('Hit any key to start simulation 1')
K=k_dk1suffix; %1 回目の制御器でシミュレーション
simu2_1
pause
disp('Hit any key to start simulation 2')
K=k_dk2suffix; %2 回目の制御器でシミュレーション
simu2_2
disp('Hit any key to end the program')
pause
close all
clc

```

### 2. D-K 反復の設定 : dk\_defin.m

```

%mu 設計を自動的に行うツール dkit のパラメータ設定
% dk_defin
%公称開ループ系結合構造
NOMINAL_DK = Gpsys;
%測定出力の数
NMEAS_DK = 1;
%制御入力の数
NCONT_DK = 1;
%mu 計算のブロック構造
BLK_DK = [1 2;1 0]; %性能変動ブロックは 2 入力 (z1,z3)1 出力 (w1)
%周波数応答の範囲
OMEGA_DK = logspace(2,6,50);
%連続系として設定
DISCRETE_DK = 0;
% AUTOINFO_DK(自動反復用)
% 1 回目から開始, 2 回目で終了, 結果を表示, スケーリング行列の最大次数は 3 とする
%AUTOINFO_DK=[1 2 1 3*ones(1,size(BLK_DK,1))];
AUTOINFO_DK=[1 2 1 3*ones(1,size(BLK_DK,1))];
%ワークスペースへ出力するすべての変数に添え字をつける
NAME_DK = 'suffix';

```

### 3. データの指定と一般化プラントの設定 : gplant.m

```

%プラントと重みの定義
Psys=nd2sys([1],[1 0 0],3.87e7);
W1sys=nd2sys([1 125.7],[1 1e-4],1.7*5e-1);
W2num1=[1 1e+4 5.7e+7];
W2den1=1;
W2den1=[1 1.2e+4 4.04e+8];
W2sys1=nd2sys(W2num1,W2den1);
W2sys=mmult(W2sys1,W2sys1,23.9);
W3sys=0.1;
%W4sys=nd2sys([10 10*0.5*5e4],[1 10*5e4]);
W4sys1=nd2sys(10*[1 0.5*5e4],[1e-2 5e5]);
W4sys=mmult(W4sys1,W4sys1,2);
%重みのボード線図
Pfr=frsp(Psys,w);
W1fr=frsp(W1sys,w);
W2fr=frsp(W2sys,w);
W3fr=frsp(W3sys,w);
W4fr=frsp(W4sys,w);
figure(1);
vplot('liv,lm',W1fr,W2fr,W3fr,W4fr);
title('Weighting Functions')
xlabel('Frequency [rad/s]')

```

```

ylabel('Gain')
grid on
pause
disp('Hit any key to start generation of Gp')
%一般化プラントを作成
systemnames='Psys W1sys W2sys W3sys W4sys';
inputvar='[w1; w2; u]';
%outputvar='[W1sys; W2sys; W4sys; w2+Psys]';
outputvar='[W1sys; W4sys; W2sys; w2+Psys]'; %zの順番をDK反復の性能
                                             変動ブロックに合わせる

input_to_W3sys='[w1]';
input_to_Psys='[u+W3sys]';
input_to_W1sys='[Psys+w2]';
input_to_W2sys='[Psys]';
input_to_W4sys='[u]';
sysoutname='Gpsys';
cleanupysic='yes';
sysic;

```

#### 4. シミュレーション用閉ループ系の作成 : closedlp.m

```

%閉ループ形の作成
systemnames='Psys K';
inputvar='[w; r]';
outputvar='[Psys+r; K]';
input_to_Psys='[K+w]';
input_to_K='[Psys+r]';
sysoutname='Clsys';
cleanupysic='yes';
sysic;

```

#### 5. $\mathcal{H}_\infty$ 制御器を用いるシミュレーション : simu2\_1.m=simu2\_2.m

```

closedlp;
[pa,pb,pc,pd]=unpck(Psys);
[ka,kb,kc,kd]=unpck(K);
[da,db,dc,dd]=unpck(sel(Clsys,1,1));
[ua,ub,uc,ud]=unpck(sel(Clsys,2,1));
%外乱応答をプロット
t=[0:0.0001:0.1];
[y,x,t]=step(da,db,dc,dd,1,t);
[u,x,t]=step(ua,ub,uc,ud,1,t);
figure(3)
subplot(2,1,1)
plot(t,y)

```

```

axis([0 0.1 0 16])
xlabel('Time[s]')
ylabel('Amplitude')
title('Step disturbance response (Hoo controller)')
grid on;
%制御入力をプロット
subplot(2,1,2)
plot(t,u)
axis([0 0.01 -1.7 0])
xlabel('Time[s]')
ylabel('u[V]')
title('Input')
grid on

```

## 8 2 慣性系の位置制御プログラム

### 1. メインファイル: mass2.m

```

% 2mass システムの位置制御
variable;           %パラメータの設定
hinf2mass;         %H 設計
simu                %シミュレーション

```

### 2. パラメータの設定: variable.m

```

% 各種変数の設定
a = 2.5;           % モータ定数
k = 39.21;         % 軸のばね定数
Jm = 4.016e-3;    % モータの慣性モーメント
Jl = 2.921e-3;    % 負荷の慣性モーメント

```

### 3. $\mathcal{H}_\infty$ 設計: hinf2mass.m

```

% 重みの設定
numw11 = [0 0 1];
denw11 = [1/100 1 0];
w1gain = 3;
W1sys = nd2sys(numw11, denw11, w1gain);
numw2 = [1 2.0e2 2.0e5];
denw2 = [1 7.5e3 5.4e7];
w2gain = 1;
W2sys = nd2sys(numw2, denw2, w2gain);
W3sys = 0.0001;
numw4 = [1 2.0e2 2.0e5];
denw4 = [1 7.5e3 5.4e7];

```

```

w4gain = 1;
W4sys = nd2sys(numw4, denw4, w4gain);
% 状態方程式
Ap = [ 0    0 -k/Jm  k/Jm;
       0    0  k/Jl -k/Jl;
       1    0    0    0;
       0    1    0    0];
Bp = [ 1/Jm  0    0    0]';
Cp = [ 1    0    0    0;
       0    0    1    0];
Dp = [ 0; 0];

Cp2 = [ 1    0    0    0;
        0    1    0    0;
        0    0    1    0;
        0    0    0    1];
Dp2 = [ 0; 0; 0; 0];

MM = [ 1    0    0    0;
       0    0    1    0];
MLo = [ 0    1    0    0];
MLt = [ 0    0    0    1];
Psys = pck(Ap, Bp, Cp2, Dp2);
% 行フルランクにするための重み
ep = 1.0e-6;
V = [eye(2)];
eV = ep * V;
% 一般化プラントの組み立て
systemnames = 'Psys W1sys W2sys W3sys W4sys MM MLo MLt eV';
inputvar = '[w1; w2; v1; v2; u]';
outputvar = '[W1sys; W2sys; W4sys; MM + eV; w2]';
input_to_Psys = '[u + W3sys]';
input_to_W3sys = '[w1]';
input_to_W4sys = '[MLo]';
input_to_MM = '[Psys]';
input_to_MLo = '[Psys]';
input_to_MLt = '[Psys]';
input_to_W1sys = '[- MLt + w2]';
input_to_W2sys = '[u]';
input_to_eV = '[v1; v2]';
sysoutname = 'Gpsys';
cleanupsysic = 'yes';
sysic;
% 積分器の調整

```

```

[A, B, C, D] = unpck(Gpsys);
An = A;
An(6,6) = -10e-3;
Bn = B;
Cn = C;
Dn = D;
Gp2sys = pck(An, Bn, Cn, Dn);
% プラントのボード線図のプロット
Pr = pck(Ap, Bp, MM, [0 0]');
w = logspace(1,4,100);
Pr_G = frsp(Pr, w);
vplot('bode', Pr_G)
pause
figure;
% 重みのプロット
W1_G = frsp(W1sys, w);
W2_G = frsp(W2sys, w);
vplot('liv,lm', W1_G, '-', W2_G, '--');
ylabel('Log Magnitude')
xlabel('Frequency (rad/s)')
legend('W1', 'W2,W4')
pause
% H 無限大制御器の計算
gmax = 1;
gmin = 0.0001;
K = hinfsyn(Gp2sys, 3, 1, gmin, gmax, 0.1);
[Ak Bk1 Ck Dk] = unpck(K);
Br = Bk1(:,3);
Bk = Bk1(:,1:2);
% 制御器のボード線図のプロット
Kr_G = frsp(K, w);
figure;
vplot('bode', Kr_G)
pause

```

#### 4. シミュレーション：simu.m

```

%閉ループ系を作成
systemnames = 'K Psys MM MLt';
inputvar = '[w]';
outputvar = '[MLt;K]';
input_to_MM = '[Psys]';
input_to_MLl = '[Psys]';
input_to_K = '[MM;w]';
input_to_Psys = '[K]';

```

```

sysoutname = 'Clsys';
cleanupsysic = 'yes';
sysic;
[pa,pb,pc,pd]=unpck(Psys);
[ka,kb,kc,kd]=unpck(K);
[da,db,dc,dd]=unpck(sel(Clsys,1,1));
[ua,ub,uc,ud]=unpck(sel(Clsys,2,1));
%目標值応答
t=[0:0.0005:0.4];
[y,x,t]=step(da,db,dc,dd,1,t);
[u,x,t]=step(ua,ub,uc,ud,1,t);
figure(3)
subplot(2,1,1)
plot(t,y)
axis([0 0.4 -0.1 1.1])
xlabel('Time[s]')
ylabel('Amplitude')
title('Step disturbance response')
grid on;
%制御入力
subplot(2,1,2)
plot(t,u)
axis([0 0.4 -2 6])
xlabel('Time[s]')
ylabel('u[Nm]')
title('Input')
grid on

```