

基本は help コマンドで確認すること. help コマンドで確認すると, 関連するコマンドも表示されるので, それらを組み合わせて扱うこと (\*1).

## 1 基本コマンド

(1) 行列の生成と操作: 行列を生成するには,

```
A=[1,2;3,4;5,6]
```

と打てばよい. これで行列  $A$  が生成され,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

が表示される.

- 特殊な行列の生成法: すべて `hogehoge(n,m)` という形で,  $n$  行  $m$  列の行列をつくることができる.
  - 単位行列を生成するには, `eye` を使えばよい. 正方行列ではない場合, 対角要素のみ 1 となり, 他の部分が零になる.
  - 要素が全て 1 の行列を生成するには, `ones` を使えばよい.
  - 要素が全て零の行列を生成するには, `zeros` を使えばよい.

- 行列  $A$  の転置  $A^T$ , および Hermite 共役  $A^*$ , 正方行列の場合の逆行列  $A^{-1}$  は, それぞれ次のように

```
A.';% A の転置
```

```
A';% A の複素共役転置
```

```
eye(size(A))/A; % A の逆行列
```

```
inv(A);% A の逆行列
```

(2) ファイルの保存と読み取り: データの保存には `save` を用い, 読み取りには `load` を用いる. 拡張子を指定しない場合, `mat` ファイルで保存.

```
% 行列を保存する場合
```

```
A=[1,2;3,4];
```

```
save data_A A
```

```
% 異なる長さのベクトルを同時に保存する場合
```

```
b=[0,2,3,4,6];
```

```
A_struct=struct('A',A,'b',b);% 構造体にする
```

```
save data_struct A_struct
```

---

(\*1) 例えば, `arctan(x)` のコマンドを知りたいとき, 「`help tan`」と打てば, 関連コマンドに `atan`, `atan2` が表示される.

c 言語と同様な記述も可能で, fprintf や fscanf も使える. また, Excel で読み込みたい場合は, xlswrite を用いればよい.

```
A=[1,2;3,4];
xlswrite('test.xls', A, 'Matrix', 'A1');
```

これは, ワークシート名 “Matrix”, ワークシート内のセル A1 から 2 行 2 列の行列  $A$  を, “test.xls” という名前で保存する.

(3) シンボリック演算: syms というコマンドを用いる. 詳しくは help で調べるか, WEB(<http://www.mathworks.co.jp/jp/help/symbolic/index.html>) を参照のこと.

- シンボリックな計算:

```
sym s
```

で,  $s$  が記号として扱うという宣言. 宣言した後, 例えば

```
expand((s+1).^3)
```

と打てば,  $(s+1)^3 = s^3 + 3s^2 + 3s + 1$  が表示される. ここで expand は, 展開するという演算 (これを打たないと,  $(s+1)^3 = (s+1)^3$  が出てくる). この逆の演算もあり,

```
simplify(s^3 + 3*s^2 + 3*s + 1)
```

と打つと,  $(s+1)^3$  が表示される.

- シンボリック変数の代入: シンボリックに計算した関数に値を代入するには, subs を使えばよい.

```
syms x;
```

```
f=x.^2;
```

```
a=2;
```

```
subs(f,x,a)
```

と書くと,  $f = 4$  が表示される.

- シンボリックな関数のプロット:

```
syms x
```

```
f=exp(-0.1*x).*sin(3*x);
```

```
x_min=0;
```

```
x_max=10;
```

```
ezplot(f,[x_min,x_max])
```

- シンボリック計算の解: コマンド solve を用いれば, 計算できるものに関して, 解を返す.

(4) 描画

- 2次元プロット: 通常用いる plot, 片対数グラフの semilogx, semilogy, 両対数グラフの loglog, 棒グラフ bar, 等高線 contour などがある. 例えば plot の場合,

```
t=[0:0.1:10];
```

```
y=sin(t).*exp(-t);
```

```
plot(t,y,'r-.','linewidth',3)
```

```
xlabel('time','fontsize',12)
ylabel('y','fontsize',12)
title('time response','fontsize',12)
```

とすれば, 'r-' で赤色鎖線表示で線幅が 3 ポイントの t-y グラフが得られる. 縦軸と横軸のラベルをそれぞれ 12 ポイントで time, y と表示させ, 描画タイトルを time response とする. 他の描画コマンドは, help を調べること.

- 3次元プロット: mesh, plot3, surf などを用いる.

```
t=[0:0.1:10];
x=cos(2*t).*exp(-0.1*t);
y=sin(2*t).*exp(-0.1*t);
z=exp(-0.1*t);
plot3(x,y,z,'r-','linewidth',3)
xlabel('x','fontsize',12)
ylabel('y','fontsize',12)
zlabel('z','fontsize',12)
title('time response','fontsize',12)
grid on;
```

- (5) function-m ファイル:  $y = f(x)$  のように, 引数  $x$  を与えて  $y$  を返す関数を定義する方法.

## 2 伝達関数と周波数応答

- (1) 有理伝達関数:

```
num=[1 2 3];
den=[1 9 3 2];
G=tf(num,den)
```

と打つと,  $G = \frac{s^2+2s+3}{s^3+9s^2+3s+2}$  という伝達関数が生成される. この伝達関数は構造体として得られる. 逆に伝達関数  $G$  が与えられたとき,

```
[num,den] = tfdata(G);
num = cell2mat(num);
den = cell2mat(den);
```

で分子多項式の係数ベクトル num と分母多項式の係数ベクトル den が得られる. cell2mat は, セル配列データを数値配列データに変換するコマンド.

- 状態空間モデルが与えられたときの伝達関数: 例えば 1 入力 1 出力系が次の状態空間表

現を持つとする.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, C = [1 \ 0 \ 0 \ 1], D = 0$$

このとき, コマンド `ss2tf` を用いて, 伝達関数  $G$  の分子多項式係数 `num` と分母多項式係数 `den` を得ることができる.

```
[num,den]=ss2tf(A,B,C,D);
```

```
G=tf(num,den)
```

多入力多出力系の場合は, 引数が追加で必要になるので, `help` を参照すること.

- 伝達関数の分母多項式の根を計算するには, `roots` を使えばよい.

```
den=[1 9 3 2];
```

```
roots(den)
```

で,  $x^3 + 9x^2 + 3x + 2 = 0$  の解が計算される. 分子多項式も同様である.

- 伝達関数で極を直接計算するには, `pole` を用いればよい.

```
num=[1 2 3];
```

```
den=[1 9 3 2];
```

```
G=tf(num,den);
```

```
pole(G)
```

零点の場合, `pole` の代わりに `zero` を用いればよい. **zeros ではないことに注意!**

- (2) 極が与えられたときに, 分母多項式の係数を計算するには, `poly` を用いればよい. 例えば  $s = -1, -3$  の極が与えられたとき,

```
num = [1];
```

```
den = poly([-1,-3]);
```

```
G = tf(num,den);
```

とすれば,  $G(s) = \frac{1}{s^2+4s+3}$  が得られる.

- (3) 無駄時間: 無駄時間  $T$  を伝達関数で表現すると,

```
T=3;
```

```
s = zpk('s');
```

```
D = exp(-T*s)
```

で  $T = 3$  のときの無駄時間の伝達関数  $e^{-3s}$  が生成される.

- (4) 伝達関数の結合: 直列結合は `series`, 並列結合は `parallel`, フィードバック結合は `feedback` で結合できる. 例えば, 伝達関数  $G_1, G_2$  が与えられているとき,

```
series(G1,G2)
```

```
parallel(G1,G2)
```

`feedback(G1,G2)`

のようにすれば, それぞれの結合による伝達関数が得られる. フィードバック結合は, ネガティブフィードバックであり,

$$\begin{aligned}\text{feedback}(G1, G2) &= \frac{G_1}{1 + G_1 G_2}, \\ \text{feedback}(\text{tf}(1, 1), \text{series}(G1, G2)) &= \frac{1}{1 + G_1 G_2}, \\ \text{feedback}(\text{series}(G1, G2), \text{tf}(1, 1)) &= \frac{G_1 G_2}{1 + G_1 G_2},\end{aligned}$$

となる. これらのコマンドには少し注意が必要で, 例えば  $G_1(s) = \frac{1}{s+1}$ ,  $G_2(s) = \frac{s+1}{s^2+s+1}$  を `series` で結合すると,

$$\text{series}(G1, G2) = \frac{s+1}{(s+1)(s^2+s+1)},$$

となる. 極零相殺を行いたい場合は, コマンド `minreal` を用いればよい.

$$\text{minreal}(\text{series}(G1, G2)) = \frac{1}{s^2+s+1}.$$

#### (5) 周波数応答

- Bode 線図: 伝達関数を生成しておき,

`bode(G)`

で Bode ゲイン線図と位相線図が得られる. 生成する伝達関数は, 有理伝達関数でなくともよい.

`w=10.^[-1:0.1:10];`

`bode(G,w)`

で, 角周波数  $w = 10^{-1}, 10^{-0.9}, \dots, 10^{10}$  の Bode ゲイン線図と位相線図を表示させる. この角周波数におけるゲインの値と位相の値が必要な場合,

`w=10.^[-1:0.1:10];`

`[gain,phase]=bode(G,w);`

`gain = squeeze(gain);`

`phase = squeeze(phase);`

によって得られる. ここで関数 `bode` から得られるデータは 3 次元配列のため, それを 1 次元に落とすために `squeeze` を用いている.

- Nyquist 線図: 伝達関数を生成しておき,

`nyquist(G)`

で Nyquist 線図が得られる. 生成する伝達関数は, 有理伝達関数でなくてもよい.

- ゲイン余裕および位相余裕: `margin` を使えばよい.

`margin(G)`

- ゲイン余裕, 位相余裕, むだ時間の余裕や単一フィードバック系の安定性は, `allmargin(G)` で調べることができる.

#### (6) 性能評価

安定な伝達関数  $G$  が与えられたとき, その性能評価として  $H^\infty$  ノルムと  $H^2$  ノルムがよく用いられる.

- $H^\infty$  ノルム: `norm(G,inf);`
- $H^2$  ノルム: `norm(G,2);`

SISO 系の場合,  $H^\infty$  ノルムは, 伝達関数  $G$  の Bode ゲイン線図の一番大きな値を意味し,  $H^2$  ノルムは, 厳密にプロパーな伝達関数  $G$  を, 虚軸上で絶対二乗積分した値を意味する.

## 3 パラメータ推定

入出力データが与えられているものとする.

- (1) 周波数データの推定: `spa` を用いる.  $y$  を出力データ,  $u$  を入力データとし, とともに縦ベクトルとする. このとき, 周波数応答は, 次のように求められる.

```
Ts=0.002;% サンプリング周期
data=iddata(y,u,Ts);% システム同定用のデータオブジェクトに変換
G=spa(data);
w=10.^[-1:0.01:1.5];
[gain,phase]=bode(G,w);
```

- (2) パラメータ最適化: `fminsearch` を用いる.