

システム制御理論演習資料

Text for Lab. of Theory and Practice
on

Analysis and Design of Control Systems

平成 27 年 7 月 13 日

レポート課題

課題の 7、8、9、10 をレポートにしてまとめよ。全問できなくとも、出来たところまで提出せよ。また、課題 7 以前のものでも、チェックしてもらってないものも一緒に提出してください。

提出期限 : 8 月 7 日

提出場所 : 大岡山キャンパス 南 5 号館 3 F 305 B 号室

Problems for a report

Solve Ex.7, Ex.8, Ex. 9 and 10 and make a report. If you can not solve all problems, please answer as many as you can and submit them. If you have Ex.s which has not been checked by TA, please submit them also.

Deadline : August 7th

Place of submission : Room 705, Ishikawadai 1st bld., Oh-Okayama campus

授業、演習のテキストは下記 URL からダウンロードできますので、各自ダウンロードしてください。

As the texts for the lecture and exercise can be down loaded from the following URL, please down load the files by yourself.

<http://www.ac.ctrl.titech.ac.jp/member/yamakita/>

MATLAB 入門

このマニュアルではMATLABの簡単な使い方を解説します。MATLABは一般のプログラム言語と同じようにプログラムを組むことが出来ますが、特に行列演算を伴う場合に威力を発揮する言語です。ここでは本演習で必要となる最低の機能を説明します。説明の後に課題問題がありますので解答してください。

1 特殊文字・演算子

MATLAB内では普通のプログラム言語と同様変数が使えますが、C言語のような型はありません。変数名は大文字と小文字を区別し、アルファベットからなる文字列で変数名を表します。しかし、 i は $\sqrt{-1}$ 、 π は π を表し変数名としては使いません。

' :' は等間隔なデータ列を生成する演算子で、 ts, td, te を実数としてつぎのような行ベクトルを生成します。

表現	生成されるベクトル
$ts:dt:te$	$[ts, ts+dt, ts+2dt, \dots, te]$, ($[te]$ は te を越えない $ts+n \cdot dt$ と表現できる実数)
$ts:te$	$[ts, ts+1, ts+2, \dots, [te]]$

この演算子は一定間隔毎の関数値を計算する場合などに非常に便利なものです。例えば関数 $\sin(x)$ の値を x が 0.0 から 1.0 まで 0.1 刻み毎の値が欲しい場合にはつぎのように簡単に計算出来ます。

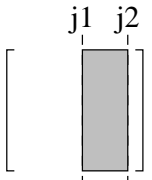

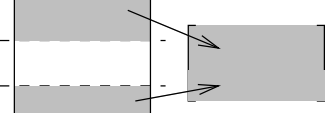
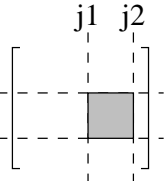
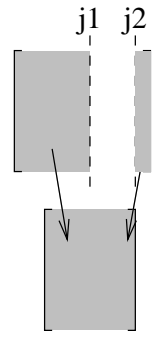
```
x = 0.0:0.1:1.0;
y = sin(x);
```

このように組み込みの 1 変数関数に行列が渡された場合、それぞれの要素に対応した関数値が入った同じサイズの行列が関数の計算結果として返されます。

2 MATLABの定数行列表現

MATLABの定数行列の表現は1のようになっています。行列は基本的にその要素を行ごとに';'で区切って表現します。このとき、要素と要素の間に','を入れます。

表 1: Mathematical 表現と Matlab 表現

Math.	Matlab	Math.	Matlab
$[1 \ 2 \ 3]$ row vector	[1,2,3]	$A = \begin{bmatrix} \\ \\ \end{bmatrix}$ 	$A(:,j1:j2)$
$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ column vector	[1;2;3]	pick up column vectors	
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ matrix	[1,2,3;4,5,6]	$A = \begin{bmatrix} i1 \\ i2 \end{bmatrix}$ 	$A(i1:i2,:)$
$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ block augmentation	[A, B;C, D]	$A = \begin{bmatrix} i1 \\ i2 \end{bmatrix}$ 	$A(i1:i2,:)=[]$
$A = \begin{bmatrix} \\ \end{bmatrix}$  pickup block matrix	$A(i1:i12,j1:j2)$	$A = \begin{bmatrix} \\ \end{bmatrix}$  delete column vectors	$A(:,j1:j2)=[]$

3 行列の保存

MATLABで行列データを保存するには、そのデータを適当な変数に代入します。この際変数の宣言は必要ありません。例えば、

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

という行列を保存したければ、行列の変数名を A として

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

とするだけです。ここで ' ' はリターンキーを表します。また、式に ';' を省略すると、式の値が表示されます。今の場合では行列 A の内容が表示されます。計算結果をすぐに表示する必要がない場合は必ず ';' を付けます。行列の i 行、 j 列要素つまり、 (i, j) 要素を取り出すのは変数名 (i,j) とします。(今の場合 A(i,j) とします)

保存したデータの中味を表示するのは、単に対応する変数名をタイプするだけです。また、行ベクトル、列ベクトル、部分行列の取り出し、拡大行列の定義の仕方を 1 にまとめてあります。行列の値を保持しないで変数だけを定義するには

```
global A;
```

とします。

特殊な行列に対してはつぎのような関数が用意されています。

表 2: 特殊な行列の定義

要素が零の行列	zeros(n) zeros(n,m)	n 行 n 列の正方の零行列 n 行 m 列の零行列
要素が 1 の行列	ones(n) ones(n,m)	要素が全て 1 の n 行 n 列の正方行列 要素が全て 1 の n 行 m 列の行列
対角要素が 1 の行列	eye(n) eye(n,m)	n 行 n 列の単位行列 対角に 1 が、それ以外はゼロの n 行 m 列の行列

[練習]

表 1、表 2 の操作で行列を作成、適当な変数に格納し結果を表示してみなさい。

4 行列に関する演算

A, B を適当なサイズの行列、 a_{ij} は行列 A の (i, j) 要素、 λ_i を固有値のひとつ、 c 適当なスカラー値であるとする。このと、定数倍、行列四則演算は表 3 のように計算される。

表 3: 行列演算の MATLAB 表現

定数倍	$c*A, A*c$
共役転置	A'
和、差、積	$A+B, A-B, A*B$
逆行列	$\text{inv}(A)$
行列式	$\text{det}(A)$
固有値・固有ベクトル	$\text{eig}(A) (= [\lambda_1, \dots, \lambda_n]')$
トレース	$\text{trace}(A)$
ランク	$\text{rank}(A)$
行列のサイズ	$\text{size}(A)$ (結果は [行数, 列数]')
全要素の和	$\text{sum}(A) (= \sum_{ij} a_{ij})$
要素同士の積	$C = A .* B (c_{ij} = a_{ij} * b_{ij})$

[練習]

$c = 4.0$, 行列 A, B を以下のように定義し表 3 の演算を全て実行し結果を確かめなさい。

$$A = \begin{bmatrix} 3 & 4 & 5 \\ 0 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

5 Mファイル

MATLAB はインタプリタ言語であるため、リターンが入力されると即座に計算が行なわれます。しかし、タイプミスなどでデータが正しくないとせっかく入力したデータが失なわれてしまいます。大きな行列などを入れ直すのは大変な作業です。また、データを変えて一連のキー操作を始めから全て手作業で行なうのは大変です。そのために、キーボードから直接打ち込む代わりに、まったく同じものをファイルとして作成しそれをあたかもキーボードから打ち込んだと同じように動作させる機能があります。そのために作ったファイルを M ファイルと呼びます。M ファイルはどのようなテキストエディタ作成してもかまいません。ただし、拡張子は .m とします。この演習

で用いる計算機は能力が十分ないため、ng というシンプルなエディタを使います。

作成した M ファイルを実行するのは、M ファイルのファイル名を filename とすると、MATLAB の動作環境そのファイル名を記述するだけです。例えば、filename という M ファイルのなかに一連のコマンドが入っており、それを実行するのは

```
filename
```

とタイプするだけです。M ファイルの中でコメントを書きたい場合は行の先頭に '%' を書き、その後にコメントを書きます。% のある行は動作に一切影響を与えません。また、ファイル内の一行が長いと自然と改行が行なわれるエディタがありますが、次の行を前の行に続いているように MATLAB に解釈させるためには '...' を前の行の最後にタイプします。

M ファイル実行中にキーボードからデータを入力する場合は input 関数を使います。普通の行列をキーボードから入力する場合は

```
A = input('prompt_string');
```

と記述します。ここで 'prompt_string' は入力要求をする際に表示する文字列です。入力されたデータは A に入ります。

また、プログラムの実行を一時中断するために pause コマンドが用意されています。pause コマンドは引数がない場合はキー入力があるまで実行を停止し、整数の引数 (n) が渡されると n 秒プログラムを停止します。

6 作業環境操作関数

MATLAB で使われる変数は原則的に大域変数で、一度生成されたものは明示的に消去するまで消えることはありません。どのような変数が残っているか、変数を消去するなどの環境を操作する関数には表 4 のようなものがあります。

表 4: 作業環境操作関数

関数名	機能
who	定義されている変数のリストを表示する
whos	定義されている変数のリストを詳しく表示する
clear 変数リスト	変数名の変数を消去する。変数名は';'で区切る。
save filename 変数リスト	現在の作業環境内の変数を filename で指定されるファイルにセーブする セーブされるファイルの拡張子は.dat で、M ファイルと互換性はないので注意。 変数リストが省略されると全ての変数をセーブする。
load filename	save 関数の逆で、ファイルにセーブされた環境を復元する

7 ヘルプ関数

どんな関数が見えるのか、どのように使うのかを調べる関数が表 5 のように用意されています。

表 5: ヘルプ関数

関数名	機能
what	使用可能関数名一覧が表示されます
help function-name	function-name で指定された関数の説明を表示
lookfor string	オンラインマニュアル内の string に一致した部分を表示

8 制御構造

MATLAB は一般のプログラム言語のように十分な、繰り返し、選択などの制御構造を持っています。それらの構文を簡単に列挙します。基本的な動作は C 言語に準じ、式と書かれたところは論理式になります。また、は省略可を示めし、— は or を意味します。

1. if ステートメント

```

if 式
    文
    {elseif 文 | else 文}
end

```

2. while ステートメント

```
while 式
    文
end
```

3. switch ステートメント

```
switch 式
    case 定数1 文1
    {case 定数i 文i}
    {otherwise 文}
end
```

[注意]

MATLABのswitchステートメントでは、一つのcase又はotherwiseの文が実行され、その他のものは実行されません。

4. for ステートメント

```
for ループ変数 = 式1 : 式2 | 式1 : 式2 : 式3 ;
    文
end
```

式1 : 式2 : 式3の場合、式1が初期値、式3が終端値、式2が増分値となる。

5. break

while や for の繰り返しから強制的に抜け出すコマンドがbreakです

[練習]

行列 A_1, A_2, A_3 を次のように定義するとき、以下の解答を連続して表示する M ファイルを作成せよ。

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -3 \end{bmatrix}$$

- (1) $A_2 * A_3$ を計算し結果を表示。
- (2) 次の行列 M を定義し、 M の固有値を計算して表示せよ。ただし、 $0_{3 \times 2}$ は 3 行 2 列の零行列を表す。

$$M = \begin{bmatrix} A_1 & A_2 \\ 0_{3 \times 2} & A_3 \end{bmatrix}$$

[おまけ]

行列演算をさらに効率よく行えるフリーソフト MaTX が以下のサイトから入手可能です。C 言語など型宣言をする言語を使ったことがある人は MATLAB の使い方が分かれば簡単に移行できるでしょう。Windows, Linux 環境など多様な環境で使えますので使ってみてはいかがでしょうか。

www.matx.org

同じサイトからリファレンスマニュアルを入手することができます。さらに親切な解説書が欲しい人は以下のような成書が出版されていますので参考にすると良いでしょう。

古賀雅伸著：MaTX による数値計算 (東京電機大学出版局)

9 関数の定義

MATLAB は普通のプログラム言語と同様ユーザ関数を定義することができます。ただ異なる点は、一つの関数は一つの M ファイルの中に記述し、明示的に宣言しない限りその中で使われる変数はそのファイル内でのみ有効なローカル変数となることです。関数を定義する M ファイルの一般的な内容な次のようになります。(Matlab のコマンドウィンドウのファイルのプルダウンメニューの「新規作成 (N)」を選択して、「ファンクション M ファイル (N)」を選ぶと下記のような画面が表れます。まったく同じではありません。)

```
function y = function_name(x1, {x2,... })
% H1
% Comment
.
.
.

Statements

y = result
```

ここで、function_name は定義したい関数名で、x1, x2,.. は仮変数リスト (関数を呼び出した際に実際の引数と置き換えられる変数のリスト) です。result は Statement の部分で関数の計算結果です。y はどのような名前でも構いませんが、このように定義することで result が関数呼び出しの結果として返されます。注意すべきことは定義した

い関数名とそれを定義したファイルは必ず `function_name.m` とすることです。%は前に説明したうよにコメントを表す行ですが、H1の行は`lookfor`関数の検索の対象となる行です。それ以降の行は`help`関数が呼び出された場合、H1と共に表示されます。

10 大域変数の利用

定義した関数にコマンドラインからパラメータを渡したい場合がありますが、その最も簡単な方法は大域変数を利用することです。大域変数を利用するためには、関数を定義しているMファイル内で使用した大域変数を次のように定義します。例えば、大域変数としてXとYを使いたい場合、

```
function y = function_name(x1, {x2,... })
global X Y
% H1
% Comment
.
.
.

Statements

y = result
```

とします。ここで注意しなければならないことは変数の間を','で区切らず、スペースを入れることです。このように定義した関数に大域変数としてX,Yを渡すには、その関数を呼び出す前に、一度

```
(clear)
global X Y

X = 2.0;
Y = 3.0;
```

として、変数を宣言し値を代入するだけです。そうすることによって、その時代入した値が関数の中で使われます。global 変数として既に X Y が宣言されていて 2 重定義になる心配があるような場合には、global 変数の定義の前に、clear コマンドで変数を消去しておきます。

11 常微分方程式の数値解法

常微分方程式として、次の一階のベクトル常微分方程式 (状態方程式) を考えます。

$$\dot{x}(t) = f(x(t), u(t), t), \quad x(0) = x_0 \quad (1)$$

この常微分方程式を数値的に解くためには、まず $f()$ を表す関数を M ファイルとして定義します。この方法は前節の方法と同じです。ただし、必ず仮引数は 2 つで、最初が実数の時刻 t に対応するもので、2 つ目が $x(t)$ とします。

```

function dx = f(t, x)
% H1
% Comment
% .
% .
% .

Statements

dx = result

```

Statement のところでは、 t と x を使って dx を計算します。今このファイルを `f.m` にセーブしたとします。この関数を定義して、初期状態 x_0 を表す行列を $X0$ とすると、MATLAB によって数値解を得るには次のように記述します。

```
[t, x] = ode45('f', [ts, te], X0);
```

ここで t_s は $X0$ が定義された初期時間で、 t_e は終端時間です。また、 t は x が計算された時刻 (必ずしも一定間隔とは限りません) が縦ベクトルとしてセーブされ、 x にはそれぞれの時刻での状態ベクトル $x(t)$ が時刻毎に横ベクトルとしてセーブされます。(下図参照)

$$t = \begin{bmatrix} t_s \\ t_1 \\ \cdot \\ \cdot \\ \cdot \\ t_e \end{bmatrix}, \quad x = \begin{bmatrix} X'_0 = x'(t_s) \\ x'(t_1) \\ \cdot \\ \cdot \\ \cdot \\ x'(t_e) \end{bmatrix} \quad (2)$$

ここで'は行列(ベクトル)の転置を表します。

12 計算結果のグラフ表示

計算された結果の表示にはplotという関数を使います。上の説明で計算された結果を、横軸を時間、縦軸をi番目の状態としてグラフ表示するには次のように記述します。

```
plot(t, x(:,i),'-',{t, x(:,j),'*',...});  
title('Solution of the ordinal differential equation');  
xlabel('Time [sec]');  
ylabel('State xi');  
legend('xi',{ 'xj', ...});  
grid on;
```

ここで、最後の'-'はグラフを実線にで表示することを示します。また、{}で囲まれた部分は、同じグラフにj番目などのグラフを同時に表示したい場合に記述します。xlabel, ylabelは縦軸、横軸の指定、legendはどのラインが何を表しているかを表示するものです。また、'grid on'はグラフにグリッドを書き入れる場合に指定します。不要の場合には必要ありません。また、グリッドがある状態から、ない状態にするには'grid off'と入力します。また、グラフの表示範囲を陽に指定するには

```
axis([ts te xs xe]);
```

とします。ただし、グラフは横軸の最初がts,最後がte,縦軸の下がxs,上がxeとなります。表示したグラフを消すにはcloseを用います。

最後に、表示されたグラフをレポートなどで使えるようにファイルに保存するには、表示されたウインドウのメニューボタンの[ファ

イル] [エクスポート設定] 「エクスポート」を選択します。データを保存する際の名前とファイルの種類 (eps 形式、jpg 形式) を入れて [保存] ボタンをクリックすると指定した形式でグラフが保存されます。

13 1自由度振動系のシミュレーション

図1に示すような1自由度振動系を考える。ここで、車輪は理想的なもので、摩擦などな一切ないとし、箱の質量を M 、ダンパーの粘性を D 、バネのバネ定数を K とする。また、箱の釣り合いの位置からの変移を y とし、箱に外力 f が働いているものとする。

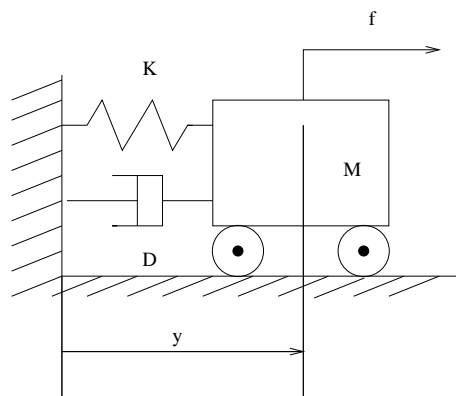


図 1: 1自由度振動系のモデル

この系の運動方程式は

$$M\ddot{y}(t) + D\dot{y}(t) + Ky(t) = f(t), \quad \dot{y}(0) = \dot{y}_0, \quad y(0) = y_0 \quad (3)$$

となる。今、このシステムに対して状態量を

$$x_1 = y, \quad x_2 = \dot{y}, \quad u = f \quad \rightarrow \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4)$$

として状態方程式を求めると

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{D}{M} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} u(t) \quad (5)$$

となる。

この状態方程式を表す M ファイルを記述すると次のようになる。ただし、 u は

$$u(t) = G \sin(\omega t) \quad (6)$$

で与えられるものとし、 G は入力の振幅、 ω は角周波数である。

```
function dx = f(t, x)
global M D K
global omega G

% Differential equation of mass-dumper-spring system
% omega and G should be defined beforehand as global
%variables.

u = G * sin(omega * t);

dx = [x(2,1); inv(M)*(-K*x(1,1) - D*x(2,1) + 1*u)];
```

この M ファイルを `f.m` とすると、この M ファイルを利用して $\dot{y}(0) = 0, y(0) = 1$ からの自由応答を時刻 0 秒から 10 秒まで計算するプログラムは次のようになる。ただし、 $M = 1.0, D = 0.5, K = 4.0$ としてある。


```
global M D K
global omega G

M = 1.0;
D = 0.5;
K = 4.0;

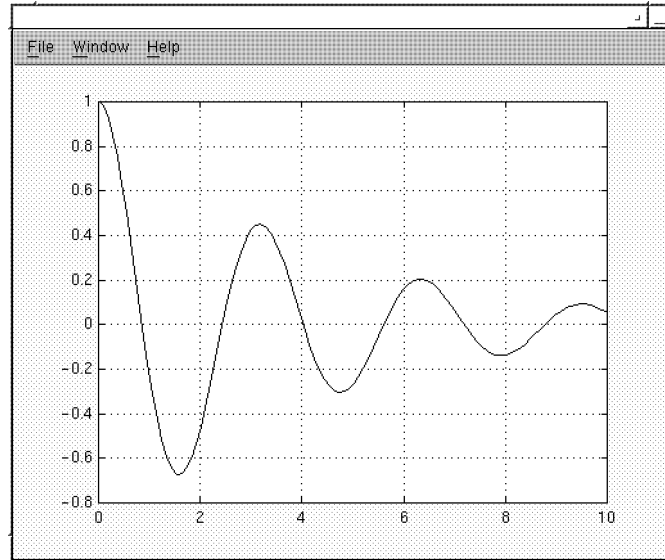
omega = 0.0;
G = 0.0;

[t, x] = ode45('f', [0.0 10.0], [1.0; 0.0]);
```

これにより計算された時刻とそのときの状態ベクトルがそれぞれ行列 t と x に計算される。この結果を横軸を時刻、縦軸を x_1 にしてプロットするには、

```
plot(t, x(:,1), '-');
grid on;
```

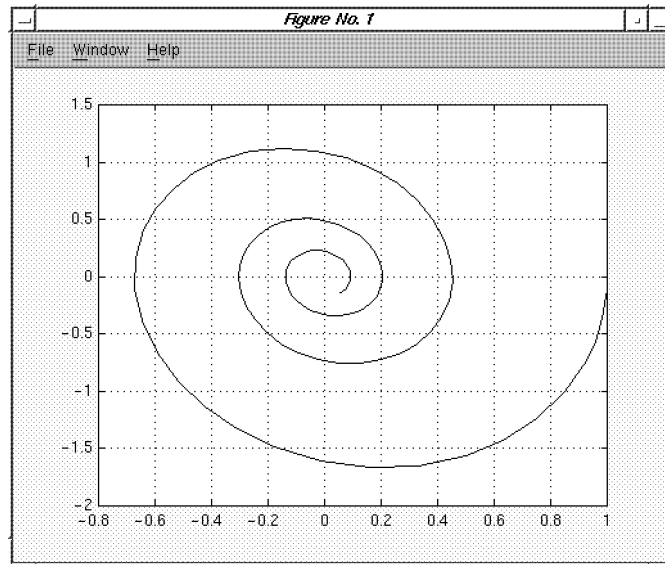
とすればよい。結果は次のようなグラフとなる。



また、状態空間での軌道を横軸を x_1 、縦軸を x_2 として描くには

```
plot(x(:,1), x(:,2), '-');  
grid on;
```

とすればよい。結果は次のようになります。



以上は自由応答の結果ですが、強制振動の応答は、例えば $\omega = 4.0, G = 1.0$ とすると、同じ初期状態からの応答は次のように計算されます。この際、 x_1 と u を同じグラフに表示するプログラムは次のようになります。

```

global M D K
global omega G

M = 1.0;
D = 0.5;
K = 4.0;

omega = 4.0;
G = 1.0;

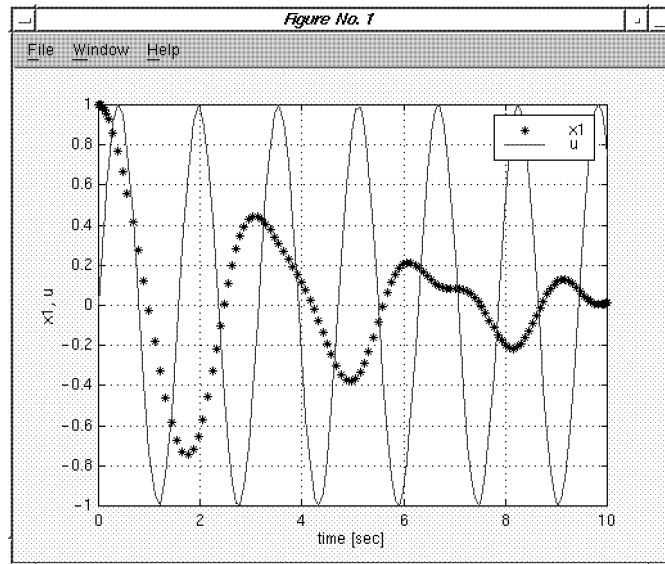
[t, x] = ode45('f', [0.0 10.0], [1.0; 0.0]);

u = G*sin(omega * t);

plot(t, x(:,1), '*', t, u, '-');
xlabel('time [sec]');
ylabel('x1,u');
legend('x1','u');
grid on;

```

このプログラムでは、計算と表示を同時に行わせています。さらに、 t に保存された時刻の入力を計算するために $u = G \cdot \sin(\omega \cdot t)$ を `ode45` の後に計算していることに注意して下さい。また、二つの信号を同一のグラフに表示しているため、 x_1 を '*' で、 u を実線で表示し、その区別を `legend` 関数で表示しています。



14 課題 1-1

図2のような2自由度振動系に関して、次の設問に答えよ。

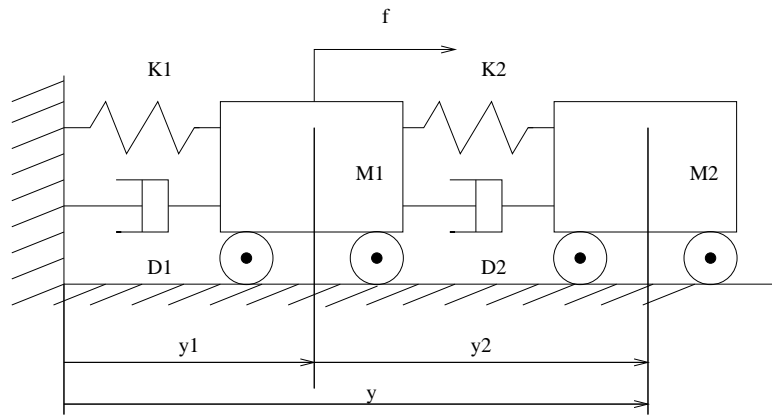


図 2: 2自由度振動系

1. 状態変数を

$$x_1 = y_1, \quad x_2 = y_2, \quad x_3 = \dot{y}_1, \quad x_4 = \dot{y}_2$$

として、このシステムの状態方程式と出力方程式を求めよ。ただし、入力は f で、出力は y とする。

2. $M_1 = 1.0, M_2 = 0.5, D_1 = 0.01, D_2 = 0.001, K_1 = 4.0, K_2 = 10.0$ として次の応答をグラフに示しなさい。

(a) $x_1(0) = 1.0, x_2(0) = 1.0, x_3(0) = 0.0, x_4(0) = 0.0, f(t) = 0$ として時刻 0 秒から 10 秒までの応答を求め、 (x_1, x_3) の位相平面での軌跡を表示しなさい。

(b) $x_1(0) = 0.0, x_2(0) = 0.0, x_3(0) = 0.0, x_4(0) = 0.0, f(t) = 2.0\sin(4.0t)$ とした強制振動の解を時刻 0 から 10 秒まで求め、時刻と y, u の関係を 1 つのグラフに表示しなさい。

[Ex. 1-1]

Let assume a 2 d.o.f.(degree of freedom) system shown in Fig. 2 and answer the following questions.

1. Defining states of the system as

$$x_1 = y1, \quad x_2 = y2, \quad x_3 = \dot{y}1 \quad x_4 = \dot{y}2,$$

express the state equation and output equation of the system using the state variables when we assume that the input for the system is f and the output is y .

2. When $M1 = 1.0, M2 = 0.5, D1 = 0.01, D2 = 0.001, K1 = 4.0, K2 = 10.0$, simulate the system behavior and show the plot of the responses under the following conditions:

- (a) Setting $x_1(0) = 1.0, x_2(0) = 1.0, x_3(0) = 0.0, x_4(0) = 0.0$, and $f(t) = 0$, calculate the solution of the dynamic equation without an external input from 0[sec] to 10 [sec], and plot the trajectory in a phase plain of (x_1, x_3) .
- (b) Setting $x_1(0) = 0.0, x_2(0) = 0.0, x_3(0) = 0.0, x_4(0) = 0.0$, and $f(t) = 2.0\sin(4.0t)$, calculate the solution with the external input from 0[sec] to 10 [sec], and show the responses of y and u in a same graph whose horizontal axis is time, t , and whose veritcal value is $y(t)$ or $u(t)$.

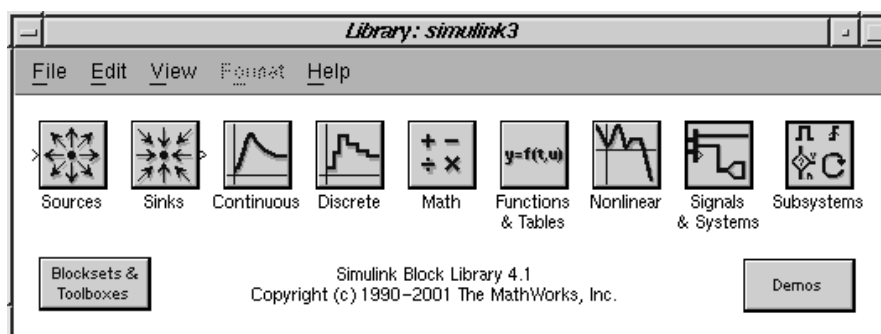
Simulinkによるシミュレーション

MatlabにはGUIを用いて簡単に数値シミュレーションを行う機能が用意されています。しかし、GUIだけを用いて、既存のブロックを用いただけでは複雑なシステムを記述するのは効率がよくありません。ここでは、非線形なシステムのシミュレーションにも容易に利用することができる、Matlab Functionを利用したSimulinkによる数値シミュレーションのやり方について説明します。(通常はS-Functionなどを用いた方法で行いますが、ここで説明をしません。興味のある人は自分で調べてみてください。)

Matlabのコマンドウィンドウで

```
>> simulink
```

とタイプするとSimulinkが使える環境となり、下図のようなLibraryウィンドウが現れます。



ここで、FileメニューからNew-JModelを選択するとUntitledというモデルウィンドウが開き、シミュレーションモデルをGUIで定義できる環境が整います。GUIだけでシミュレーションモデルを構築するためには、LibraryウィンドウにあるSourceやSinksなどのアイコンをダブルクリックして、その中のアイコンで表されるブロックをモデルウィンドウにドラッグして、それぞれのブロックを連結することによりモデルを構築することになります。例えば、LTIシステムのみを結合したシステムのモデルを作成するためには、Continuousライブラリの中のState-Spaceブロックを結合し、ブロックのパラメー

タはブロックをダブルクリックし現れたパラメータ入力ウィンドウで (A, B, C, D) を行列表現で入力するか、Matlab 内で定義された変数で指定します。また、各ブロックの初期状態もここで入力します。正弦波などの入力信号生成ブロックは Sources ライブラリにあり、信号の表示を行うブロックは Sinks ライブラリ内にあります。(一番簡単に利用できるものは Scope ブロックです。)

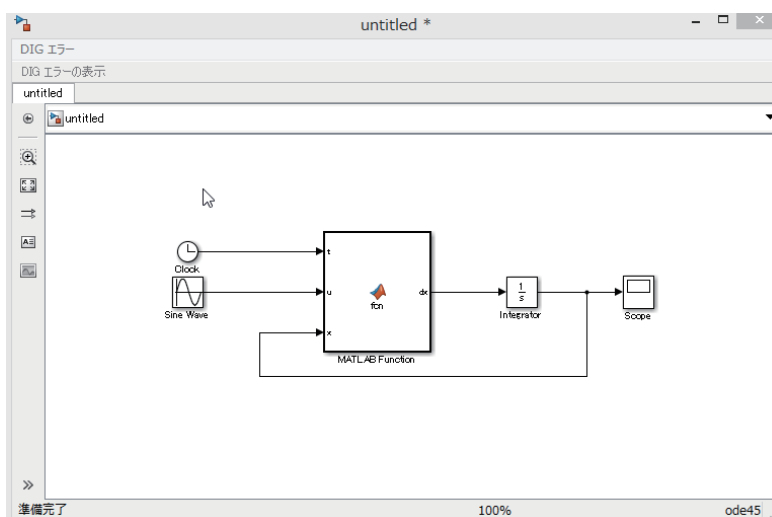
Simulink に用意されたライブラリを結合することで多様なシステムをシミュレーションすることができますが、やはりそれだけでは十分ではありません。以降では、一般のシステムのシミュレーションを行うのに利用できる方法として、Matlab Function を用いたシミュレーションの方法について説明します。

Matlab Function を用いた方法

一般の非線形連続時間系の状態方程式は

$$\dot{x}(t) = f(x(t), u(t), t) \quad (7)$$

として表されます。従って、 $x(t), u(t), t$ に対して、 \dot{x} を計算するブロックを定義し、それを積分するように以下のように定義すればいいです。



ただし、図の中の MATLAB Function の $x(t), u(t), t$ に対して \dot{x} を計算するブロックです。図中の $x(t), u(t), t$ の入力ポートは後に説明

するように、Matlab Function の中身の関数の仮引数を定義することにより自動的に作られます。例えば、

$$\frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} -\sin(x_1(t)) + u(t) \\ -u(t) \end{bmatrix} \quad (8)$$

というシステムを考えます。この時、Matlab Function ブロックをダブルクリックして、ブロック内の関数を以下のように定義します。

```
function dx = f(t,u,x)

dx=[-sin(x(1))+u;-u];
```

ここで注意しなければならないのは、システムの状態 x の次元はモデルブロックの中の積分器の初期状態のベクトルのサイズとして決定されることです。積分器で設定した次数(サイズ)と、関数の中の変数 x の次元は一致しているように用いなければなりません。

15 シミュレーションの実行結果の加工・ファイルへのセーブ

シミュレーションモデルを作成して、単にグラフにして表示させるだけでよければ Sinks フォールドにある Scope を用いることができます。結果を表示させるには、シミュレーションの実行の後に Scope ブロックをダブルクリックすれば結果をみることができます。しかし、多くの場合結果を配列に入れて後でいろいろと加工したり、レポートのためのグラフを作成したい場合があります。そのために、シミュレーション結果をワークスペース(コマンドウィンドウで表示される環境)の変数に配列や構造体として記憶する機能があります。一度変数に記憶することができれば、plot() 関数などを用いて必要な形にグラフを成形することができます。そのためには、[Library フォールド] [Sinks] にある To Workspace というブロックを使うことにより、シミュレーション結果をワークスペースの任意の変数に配列や構造体として格納することができます。

時間応答をプロットするには、横軸の時間データと、縦軸にプロットする状態が通常必要になります。そのために、まず Sorce ライブラリにある Clock ブロックをドラッグして、To Workspace と結合します。また、状態を記録するために State-Space ブロックの出力から分枝を出してもう一つの To Workspace ブロックに結合します。

最後に、表示されたグラフをレポートなどで使えるようにファイルに保存するには、表示されたウインドウのメニューボタンの [ファイル] [エクスポート設定] 「エクスポート」を選択します。データを保存する際の名前とファイルの種類 (eps 形式、jpg 形式) を入れて [保存] ボタンをクリックすると指定した形式でグラフが保存されます。

[課題 1 - 2]

以下の 1 重振子のシミュレータを、Simulink で Matlab Function を用いたモデルとして作成しなさい。ただし、運動方程式は次式で与えられる。

$$(ml^2 + I)\ddot{\theta} + \mu\dot{\theta} + mgl\sin(\theta) = \tau \quad (9)$$

ただし、 $m = 1.0[Kg]$, $l = 0.1[m]$, $I = 0.001[Kgm^2]$, $\mu = 0.001[Kgms]$, $g = 9.8[m/s^2]$ である。この作成したシミュレータを、 $\tau(t) = 0.0$, $\theta(0) = \pi/2$, $\dot{\theta}(0) = 0.0$ の条件の下で、20 秒間シミュレーションした結果の $\theta, \dot{\theta}$ を表示しなさい。ただし、Matlab では π は pi で表される。

[Ex1-2]

We consider the following pendulum system:

$$(ml^2 + I)\ddot{\theta} + \mu\dot{\theta} + mgl\sin(\theta) = \tau, \quad (10)$$

where $m = 1.0[Kg]$, $l = 0.1[m]$, $I = 0.001[Kgm^2]$, $\mu = 0.001[Kgms]$, $g = 9.8[m/s^2]$. Define a model with a Matlab Function using Simulink, and show a simulation result of $\theta, \dot{\theta}$ during 20 sec. when $\tau(t) = 0.0$, $\theta(0) = \pi/2$, $\dot{\theta}(0) = 0.0$

MATLABでのLTI(Linear Time Invariant)システム表現

Bode線図とNyquist線図

16 MATLABでの多項式表現

時不変線形の状態空間(SS:State Space)表現が

$$\dot{x} = Ax + Bu \quad (11)$$

$$y = Cx + Du, \quad x \in R^n, u \in R^m, y \in R^p \quad (12)$$

と表されるとき、このシステムの伝達関数(TF:Transfer Function)行列表現 $H(s)$ は s をラプラシアンとして

$$H(s) = C(sI - A)^{-1}B + D \quad (13)$$

と表される。ここで、 $H(s)$ は s の有理関数を要素とする p 行 m 列の行列である。今、この行列の (i, j) 要素は

$$h_{ij} = \frac{b_n s^n + b_{n-1} s^{n-1} + \dots + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_0} \quad (14)$$

と表され、 s の多項式の比として求められる。このように、LTIシステムを表現するためには多項式を扱う必要がある。MATLABでは、多項式を必要とするところでは、その係数を並べた、または、因数分解した際の根を並べた横ベクトルで表現する。(ただし、根で指定した場合は最高次数の係数が1のモニックなものしか表現しない。) 例えば、多項式として

$$p(s) = s^2 - 2s + 2$$

を表したい場合、

$$[1 \quad -2 \quad 2]$$

と表し、次数の小さくなる順番でその係数を要素とする横ベクトル、又は

$$[1 + i \quad 1 - i]$$

として表現する。これらの表現は、以下に説明するシステムを定義する関数によって使い分けられる。また、離散系を表現する場合、ラプラス変換 s ではなく、遅れに対応する z^{-1} を用いるが、この場合は z^{-1} の次数が大きくなる順番に記述します。例えば、

$$p(z^{-1}) = 3 + 2z^{-1} + 1z^{-2}$$

を表すには

$$[3 \ 2 \ 1]$$

と表す。いずれにしても、係数で表現する場合、横ベクトルの左側の方に '時間的に進んだ' ものに対応した係数が来ると覚えておけばよい。

17 LTIシステムの定義

LTIシステムで同じ入出力特性を表す等価な表現は基本的には状態空間表現と伝達関数表現があります。MATLABではこれらの表現としてシステムを定義し、等価なシステム表現を簡単に得る関数を提供しています。例として次の状態空間表現で表されるシステムを考える。

$$\dot{x} = Ax + bu \quad (15)$$

$$y = cx + du \quad (16)$$

ただし、

$$A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad c = [c_0 \ c_1], \quad d = 0$$

である。このシステムの伝達関数 $H(s)$ は

$$H(s) = \frac{c_1 s + c_0}{s^2 + a_1 s + a_0} \quad (17)$$

で与えられる。また、 $H(s)$ の分子・分母を因数分解して

$$H(s) = k \frac{(s + z_1)}{(s + p_1)(s + p_2)}$$

と表されるとする。この時、A, b, c, d を MATLAB での上記の定数行列を表す変数とすると、状態空間表現でのシステムの定義は

```
sys = ss(A, b, c, d);
```

となる。また、伝達関数行列で定義する場合は

```
sys = tf([1 a1 a0], [c1 c0]);
```

とします。また、伝達関数を因数分解して、分子多項式の根 (零) と分母多項式の根 (極) とゲインで定義するには

```
sys = zpk([z1], [p1 p2], k);
```

とします。もし、離散時間システムを定義するのであれば、最後にサンプリング時間 T_s を引数として渡します。

伝達関数行列でシステムを定義する場合、伝達関数行列の要素毎に、その伝達関数の分子多項式、分母多項式、必要によってゲインを要素とする、横ベクトル又は定数を要素とする配列を引数として関数を呼び出します。配列の定義は、行列とほぼ同じですが、`[]`ではなく、`{}`によって始まりと終りを示します。(既約分解表現のように、配列が分子行列、分母行列のように多項式を要素とする行列として意味のあるものではないことに注意して下さい。あくまでも、対応する要素毎に意味のあるものです。) 例えば、2入力1出力で伝達関数行列が

$$H(s) = \begin{bmatrix} 1 & s+1 \\ s^2+3s+2 & 2s^2+3s+1 \end{bmatrix}$$

のシステムを定義するには、

```
sys = tf({[1] [1 1]}, {[1 3 2] [2 3 1]});
```

とします。

これらのシステムの定義では、定義されるシステムの変数は全て `sys` で表していますが、内部的には SS, TF, ZPK タイプのデータとし

て保存されます。しかし、これらの違いを陽に区別する必要は普通ありません。伝達関数表現などから状態空間表現を求める必要がある場合の内、特に特殊な例を除いて、必要なデータタイプは自動的に計算されます。(ただし、どの表現形式を用いるかによって、数値的計算精度が大きく異なる場合があります。特に、伝達関数表現を経由する場合に問題になる場合があります。)

これに関連して、複数のサブシステムを結合して新しいシステムを作る場合、それぞれのサブシステムがSS, TF, ZPKタイプのデータを混在して持っていた場合、結果のシステムのタイプは次の優先順位の順に決定されます。

$$SS > ZPK > TF$$

つまり、サブシステムが複数のタイプのデータによって定義されている場合、最も左側にあるタイプのデータによって結果のシステムが定義されるということです。これは、SSタイプのデータが最も数値的に安定であることから来ています。

18 各種表現パラメータの取得

前節では、LTIシステムの定義を説明しましたが、定義されたシステムから、それぞれの表現形式のパラメータは以下のようにして取得します。

```
[A,B, C, D, Ts] = ssdata(sys);  
[N,D, Ts ] = tfdata(sys);  
[Z, P, T_s] = zpkdata(sys);
```

ここで、Tsは離散系のサンプリング時間で、連続系の場合ゼロが返されます。

sysがm入力p出力の場合、その部分伝達特性を持つLTIシステムを指定するのは行列のblock行列を操作するのと同様に、システムを表す変数に()を付けてその部分を指定すればよい。例えば、sysの最初の2つの入力部分から、3番目の出力までを表すシステムを取り出したければ、


```
subsys = sys(3,1:2);
```

とすれば良い。この指定は、伝達関数行列の部分伝達関数行列を取り出すことに対応している。

19 Bode 線図

伝達関数の $j\omega$ 軸上での特性を横軸を角周波数 [rad/sec]、縦軸にゲインと位相を片対数表にプロットしたものが Bode 線図である。システムが安定である場合、この特性は周波数伝達関数に一致する。(不安定な場合は単なる $j\omega$ 軸上での特性である) 一般に、ゲインはデシベル [dB]、位相は度 [deg] で表され、上下別々の図として表されます。ゲインを表す図をゲイン線図、位相を表す図を位相先途と呼びます。横軸の対数軸で等間隔なデータを表示するために、そのような角周波数データを生成する関数として `logspace` 関数が用意されています。基本的な使い方は

```
w = logspace(d1,d2,n);
```

で、これによって 10^{d1} から 10^{d2} の間を対数間隔で n 当分した点からなるベクトルを作り、`w` に代入します。この `w` を用いて、Bode 線図を描くためのデータは `bode` 関数によりつぎのように計算することができます。

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

`gain`, `phase` は `w` の要素の周波数での絶対値と位相を計算した 3 次元配列で、多入出力伝達関数行列で、 (i,j) 要素の k 番目周波数でのゲイン、位相を取り出すのはそれぞれ、`gain(i,j,k)`、`phase(i,j,k)` となります。`gain` は絶対値であるので、ゲインの単位を dB に変換するには

```
gain = 20*log10(gain);
```

とします。これらのデータから Bode 線図を描くには、1つの画面に2つのグラフを書くために `subplot` 関数と `semilogx` 関数を組み合わせます。`subplot` 関数は画面を分割して、その分割されたどの部分に `plot` をするかを指定する関数で、

```
subplot(nmp);
```

で指定します。ここで、 n,m は画面を n 行、 m 列に分割し、 p はどの画面に plot するかを表しています。 p は左から右に、上から下に番号を数えます。

伝達関数行列の (1,1) 要素の Bode 線図を描くには次のようにします。

```
Gain = squeeze(gain(1,1,:));  
Phase = squeeze(phase(1,1,:));  
subplot(211);semilogx(w, Gain);grid on;  
subplot(212);semilogx(w, phase);grid on;
```

ここで、`squeeze` 関数を用いているのは、3次元配列から2次元部分を取り出しても、そのデータのタイプは3次元配列で、それを2次元の行列 (ベクトル) にするためです。

以上は基本の関数を定義通りに組み合わせて表示した場合で、`bode`の左辺を指定せずに

```
bode(sys, w);
```

とタイプすれば上記と同様のグラフが表示されます。この際 w を省略すると、 w に対応するデータはシステムの零と極によって自動的に計算されます。

20 Nyquist 線図

Bode 線図は伝達関数の $j\omega$ 軸上での特性を各 ω についてゲインと位相に分けて表示するのであったが、この特性を複素平面上にプロットするのが Nyquist 線図である。Nyquist 線図を描く関数は `nyquist` で、使い方はほぼ `bode` と同じである。基本的な使い方は下記のいずれかである。

```
nyquist(sys);  
or  
[re, im, w] = nyquist(sys);
```

ここで、re と im はそれぞれ w 要素に対応する伝達関数の実部と虚部を表したベクトルである。

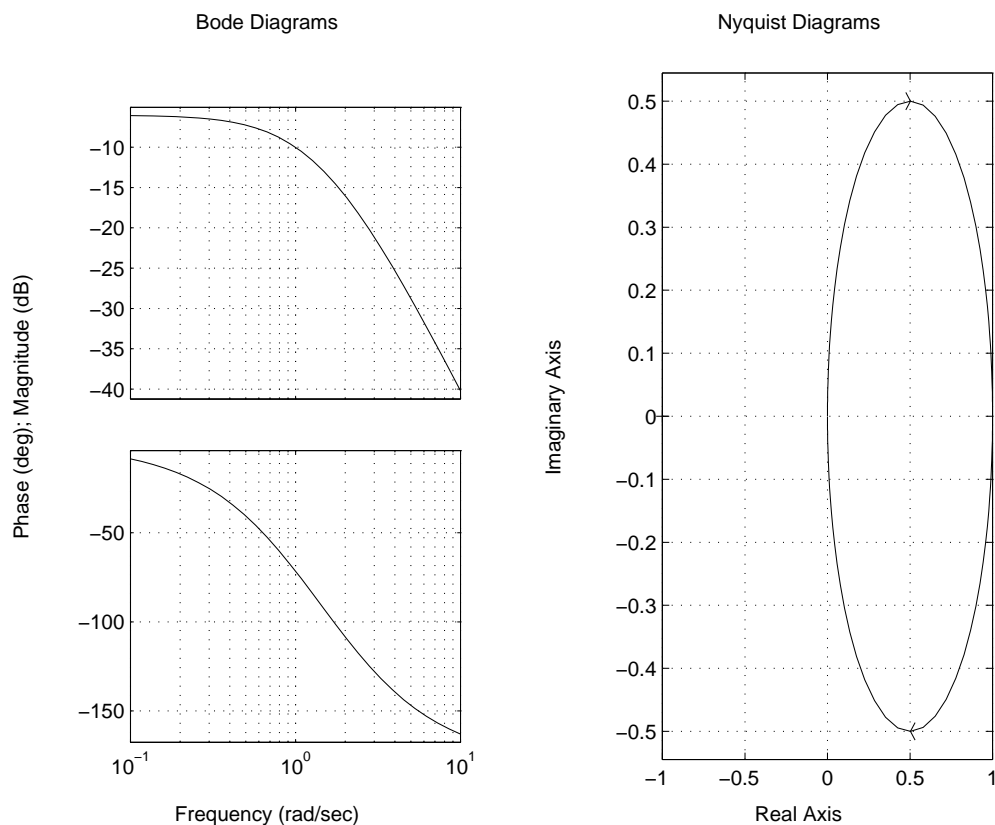
[例] 次の伝達関数を持つシステムを定義し、

$$H(s) = \left[\frac{1}{s^2 + 3s + 2} \quad \frac{s + 1}{2s^2 + 3s + 1} \right]$$

最初の入力から出力までの伝達関数の Bode 線図と、2 番目の入力から出力までの伝達関数の Nyquist 線図を同じ画面に横に並べて描画するには次のように入力する。

```
sys = tf([1] [1 1]],[1 3 2] [2 3 1]);
subplot(121);bode(sys(1,1));grid on;
subplot(122);nyquist(sys(1,2));grid on;
```

結果は次のようになる。



21 課題 2

前の課題(課題 1 - 1)のシステムに対して、出力が y 以外に y_2 も出力とする 1 入力 2 出力のシステムを考える。このとき次の設問に答えなさい。

1. システムを状態空間モデルにより定義し、対応する伝達関数モデルを求めなさい。
2. 入力から y までの Bode 線図と、入力から y_2 までの伝達関数の Nyquist 線図を横に並べて表示しなさい。
3. 入力から y までのゲイン線図でもっとも大きなゲインを持つ角周波数(共振角周波数)をグラフより求め、シミュレーションによりそのことを確認しなさい。(定常状態での共振周波数の正弦波入力とその出力を同じグラフの中に描きなさい。)

[Ex.2]

Consider again the system in Ex.1-1 but now we consider that y_2 is also the output of the system in addition to y , i.e., we consider an 1 input and 2 output system. Then answer the following question.

1. Define a state space model of the 1 input and 2 output system, and calculate the transfer function matrix of the system.
2. Plot a Bode diagram of a transfer function from u to y , and plot a Nyquist plot of a transfer function from u to y_2 , then display the both plots in a row in one figure.
3. Determine a circular frequency whose amplitude of the transfer function from u to y is maximum, i.e., resonance circular frequency, from the figure, and make sure the relationship in time domain by a simulation. (Plot a sinusoidal input with the frequency and the corresponding output in the same graph when the time is large enough.)

22 システムの極

次の状態空間表現で表される LTI システムを考える。

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (18)$$

このとき、次の多項式:

$$\det(sI - A)$$

を特性多項式と呼び、この根 (A の固有値) をシステムの極と呼ぶ。MATLAB では、特性多項式の係数は

```
p = poly(A);
```

により求めることができる。ただし、p は特性多項式の係数が入った横ベクトルである。固有値は計算された多項式ベクトルより roots() 関数を用いて、

```
roots(p)
```

により計算できるが、A 行列より eig() 関数によりつぎのように

```
v = eig(A);
```

直接計算できることは既に説明した。eig() 関数の使い方として、固有値だけではなくて、固有ベクトルを同時に求めるためには、

```
[v, d] = eig(A);
```

とすることにより、d の対角に固有値を、また対応する固有ベクトルをの対応する位置に列ベクトルとして求めることができる。この際、固有ベクトルはユークリッドノルムが 1 となるように規格化される。このように、関数を呼び出す左辺の変数の数により、関数が返す値の数が変わる関数が MATLAB では多くある。

23 不変零点

式(18)で与えられるシステムに対して、次の行列をシステム行列と呼ぶ。

$$\begin{bmatrix} A - sI & B \\ C & D \end{bmatrix} \quad (19)$$

このシステム行列がフルランク (Normal Full Rank) であるとき、そのランクが変化する s のことを不変零点と呼ぶ。MATLABではこの零点の集合を

```
d = tzero(sys);  
or  
d = tzero(A, B, C, D);
```

により計算することができる。

この s といわゆる伝達関数の零との関係を探るため、1入力1出力の次の状態空間表現で与えられるシステムを考える。

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -3 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y &= [6 \ 5 \ 0]x + 0u \end{aligned} \quad (20)$$

このシステム行列は正方行列となるため、そのランクが変化する s は行列式を計算し、それがゼロとなるもので与えられる。このシステム行列の行列式は $5s + 6$ で与えられ、このシステムの伝達関数 $H(s)$ は

$$H(s) = \frac{5s + 6}{s^3 + 2s^2 + 3s + 4}$$

となり、不変零点が伝達関数の零点 $s = -1.2$ と一致していることが分かる。

[演習]

上記のことをシステム行列の行列式を手計算し(ヒント:システム行列の一番右の列に対して行列式の展開公式を適用)、また `tf()`, `tzero()` を用いて確認せよ。

MATLAB では、システムの極と不変零の配置を同時に図にプロットする関数 `pzmap()` 関数が用意されており、使い方は

```
pzmap(sys);  
or  
[p, z] = pzmap(sys);
```

で、左辺を定義した場合は `p` に極が、`z` に零が代入される。

24 可制御性・可観測性

制御系設計をする際、システムの可制御性・可観測性(可安定性・可検出性)をチェックすることは非常に重要である。そのチェックを怠ると、解の存在しない問題を、ただ時間を無駄にして探しているという状況に陥る可能性もある。(MATLAB のようなツールを用いていけば、解が存在しない場合はそのような警告がでるが、理由を理解するには少なくとも必要)

式(18)で与えられるシステムの可制御性・可観測性はいわゆる PBH(Popov-Belvitch-Hautus) テストにより判定することができる。その判定法は次のように与えられる。

[可制御性]

システムが完全可制御であるための必要かつ十分条件は以下のいずれかが成り立つことである。

1. $W(0, t_1) := \int_0^{t_1} e^{At} B B^T e^{A^T t} dt$ が正則である。(この $W(0, t_1)$ を可制御グラミアンという)
2. 可制御行列を $V := [B, AB, \dots, A^{n-1}B]$ とするとき、 $\text{rank } V = n$ である。
3. λ を A の任意の固有値とするとき、 $\text{rank}[\lambda I - A, B] = n$ である。

また、可観測性はつぎの双対なもので与えられる。

[可観測性]

システムが完全可観測であるための必要かつ十分条件は以下のいずれかが成り立つことである。

1. $M(0, t_1) := \int_0^{t_1} e^{A^T t} C^T C e^{A t} dt$ が正則。($M(0, t_1)$ を可観測グラムリアンという。)
2. 可観測行列を $N^T := [C^T, A^T C^T, \dots, (A^T)^{n-1} C^T]$ とするとき $\text{rank } N = n$ である。
3. λ を A の固有値とすとき、 $\text{rank } [\lambda I - A^T, C^T] = n$

MATLAB では可制御行列・可観測行列 V, N は次ぎのコマンドにより求めることができる。

```
V = ctrb(sys);
N = obsv(sys);
```

これらの行列のランクを `rank()` 関数により計算すれば、可制御性・可観測性をチェックすることができる。

また、可制御グラムリアン $W_c = W(0, \infty)$ 、可観測グラムリアン $W_o = M(0, \infty)$ の定常状態での値は次ぎのコマンドにより求めることができる。

```
Wc = gram(sys, 'c');
Wo = gram(sys, 'o');
```

これは、

$$\int_0^\infty dt \frac{d}{dt} W(0, t) = W(0, \infty) - W(0, 0) = A \int_0^\infty e^{A t} B B^T e^{A^T t} dt + \int_0^\infty e^{A t} B B^T e^{A^T t} dt A^T \quad (21)$$

となり、 A が安定として、可制御グラムリアンの定義を用いると、

$$-B B^T = A W_c + W_c A^T \quad (22)$$

となり、この行列方程式を解くことにより可制御グラムリアンが求められている。このような行列方程式のことをリアプノフ方程式といい、一般形は X を変数行列、 A, Q を既知の定数行列として

$$A X + X A^T + Q = 0 \quad (23)$$

として与えられる。

MATLAB ではこのリアプノフ方程式は次ぎのコマンドで解くことができる。

```
X = lyap(A, Q);
```

[課題 3]

図3のような2自由度振動系に関して、次の設問に答えよ。

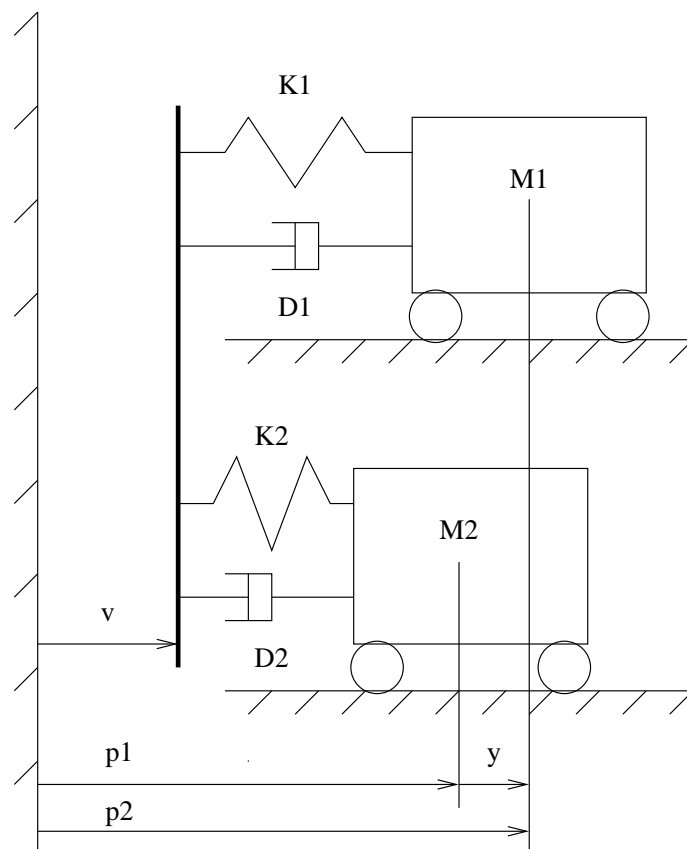


図 3: 並列 2 自由度振動系

1. 状態変数を

$$x_1 = p1, \quad x_2 = p2, \quad x_3 = \dot{p}1 \quad x_4 = \dot{p}2 \quad x_5 = v$$

として、このシステムの状態方程式と出力方程式を求めよ。ただし、システムへの入力は \dot{v} 、出力は $v, p1 - p2$ とする。

2. $M1 = 1.0, M2 = 0.5, D1 = 0.01, D2 = 0.001, K1 = 4.0, K2 = 10.0$ としてシステムの極、不変零点を求め、可制御性・可観測性を判定せよ。

3. $M1 = 1.0, M2 = 1.0, D1 = 0.01, D2 = 0.01, K1 = 4.0, K2 = 4.0$

としてシステムの極、不変零点を求め、可制御性・可観測性を判定せよ。

[Ex.3]

Let consider a 2 dof system in Fig 3 , and answer the following questions:

1. Let define state variables of the systems as

$$x_1 = p1, \quad x_2 = p2, \quad x_3 = \dot{p}1 \quad x_4 = \dot{p}2 \quad x_5 = v,$$

express the state space equation and output equation of the system using the state variables where input for the system is \dot{v} and the output are $v, p1 - p2$.

2. Let assume that $M1 = 1.0, M2 = 0.5, D1 = 0.01, D2 = 0.001, K1 = 4.0, K2 = 10.0$, calculate system poles and invariant zeros of the system, and also check the controllability and observability of the system.
3. Let assume that $M1 = 1.0, M2 = 1.0, D1 = 0.01, D2 = 0.01, K1 = 4.0, K2 = 4.0$, calculate system poles and invariant zeros of the system, and also check the controllability and observability of the system.

状態フィードバックによる極配置とオブザーバ併合系

25 乱数の発生

システムの応答を調べるために数値シミュレーションする際、実際の場合に即して、観測外乱などを考慮したシミュレーションを行うことは重要である。MATLAB では確率的な信号を発生するために、2つの関数 `rand()` と `randn()` が用意されており、それぞれ機能は次のようになっている。

`rand(n,m,l)` 0 から 1 の間の一様乱数を要素とする 3次元配列として生成する。l を省略すると、n 行 m 列の各要素が一様乱数となった行列となる。

`randn(n,m,l)` 平均値 0 分散 1 の正規分布に従う乱数を要素とする 3次元配列を生成する。l を省略すると、n 行 m 列の各要素が正規分布に従う行列となる。

これより、平均値が m で、 $m-L$ から $m+L$ の間の一様乱数の実数を発生させるには、

$$m - L + 2 * L * rand(1, 1) \quad (24)$$

とすれば良い。また、平均値 m 、分散 σ の正規分布に従う実数乱数を発生させるためには、

$$m + \sqrt{\sigma} * randn(1, 1) \quad (25)$$

とすればよい。

[演習] 上記のことを証明しなさい。

MATLAB の数値積分関数 `ode45()` などでは、内部で収束演算を行っているため、微分方程式の中で確率変数を用いると積分の増分が収束しないためシミュレーションが終らなくなってしまう。そこ

で、この演習では、予め確率外乱を一定周期で行列に保存しておき用いることとする。例えば、1 入力のシステム

$$\dot{x} = Ax + Bw, \quad x(0) = X0$$

で、 w が分散 Q の白色性の確率外乱であるシミュレーションをすることを考えるとき次のようにプログラミングすることとする。ただし、 t_s , t_e はシミュレーションの開始時刻と終端時刻であり、 h は確率外乱を記憶する（更新する）インターバルである。

```
global w N

N = round((te-ts)/h);
w = randn(N+1,1);

[t, x] = ode45('f', [ts, te], X0);
```

```
function dx = f(t, x)
% Simulation with stochastic state disturbance

i = round(t/h) + 1;
if i > N
    i = N
end

u = w(i,:)*sqrt(Q)/sqrt(h);

dx = A*x + B*u
```

ここで `round()` は実数から切捨てにより整数を求める関数で、`if` の部分は `t` が `te` より大きな値になる可能性への対処である。MATLAB の数値積分関数では、積分の時間刻みを適応的に最適化するため、このような処理が必要となる。また、`u` の計算の際、 \sqrt{h} で割っているのは、連続時間の確率外乱に等価な離散時間での確率外乱とするためである。(ode 内では `f` に `h` が掛算されるため、`w` は $w * h$ となり、 $Cov(wh) = 1 * Q/h * h^2 = Qh$ となる)

26 状態フィードバックによる極配置

次の状態空間表現で表される LTI システムを考える。

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (26)$$

このとき、 F を適当な行列として、制御入力を、

$$u(t) = Fx(t)$$

とすることを状態フィードバックという。また、この制御入力によって閉ループの状態方程式は次の自律系

$$\dot{x} = (A + BF)x \quad (27)$$

となり、閉ループ系のシステムの極は $A + BF$ の固有値となる。この固有値を状態フィードバックにより配置することを状態フィードバックによる極配置という。この状態フィードバックによって、閉ループ系の特性多項式の係数が実数という制約の下で、閉ループ系の極を任意の場所に配置できるための必要十分条件はシステムが完全可制御であることである。

システムが完全可制御であるとき、MATLAB で状態フィードバック行列 F を計算する関数は `place()` で、次のように用いる。

```
F = place(A,B,p);
```

ただし、 p は閉ループ系の極を並べたベクトルである。ここで注意することは、MATLAB で計算されるものは $(A - BF)$ の極配置問題を解くものであるので、 $u = Fx$ とする場合には計算された F の逆符号 $-F$ を用いなければならない。また、place での極配置のアルゴリズムでは、極の重複が入力の数よりも大きい時は解けないので注意する。

27 オブザーバ (状態推定器)

27.1 双対システム

式 (26) で与えられるシステムに対して、次ぎの式で与えら得るものを元のシステムに対する双対 (共役) システムと呼ぶ。

$$\frac{d}{dt}x^* = -A^T x^*(t) - C^T y(t) \quad (28)$$

$$u(t) = B^T x^*(t) + D^T y(t) \quad (29)$$

ここで、このシステムの入力の次元が元の出力の次元と同じに、出力の次元が元のものの入力の次元と同じであることに注意する。また、伝達関数を考えた場合、伝達関数行列を転置して、 s を $-s$ に換えたものに等しくなっている。(このような双対性の概念はそれぞれ内積の定義されている 2 つの空間での写像を考え、その双対写像をいうものものを考えることによりシステムティックに導出されるがここでは詳しいことは省略する。) このままでは、元のシステムが安定である場合、双対システムは不安定となっているため、適当な終端時間 T を仮定し、

$$\tau = T - t \quad (30)$$

という逆時間 τ を導入し、微分方程式を τ で書き直し

$$\frac{d}{d\tau}x^*(\tau) = A^T x^*(\tau) + C^T y(\tau) \quad (31)$$

$$u(\tau) = B^T x^*(\tau) + D^T y(\tau) \quad (32)$$

を双対システムということもある。このような双対システムを考える利便性は、可観測性という概念が双対システムの可制御性に置き

換えることができるなど、可観測性の条件を双対システムの可制御性の条件に機械的に置き換えてその条件を導くことができるためである。

27.2 全次元オブザーバ

元のシステムと同じ次数のオブザーバを全次元オブザーバという。(確率的な外乱を考えた際の最適状態推定器である Kalman フィルターはこの全次元オブザーバの一種) 全次元オブザーバは次の LTI システムで表される：

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + K(\hat{y} - y) \\ \hat{y} &= C\hat{x} + Du\end{aligned}\quad (33)$$

ただし、 K はオブザーバゲインである。オブザーバの状態と実際のシステムの状態の偏差ベクトルを $\xi := \hat{x} - x$ を定義すると、この偏差系の状態方程式は

$$\dot{\xi} = (A + KC)\xi \quad (34)$$

となる。従って、 $A + KC$ の固有値を安定にできれば、任意の初期状態 $\xi(0)$ から $\xi(t)$ は零に収束し、 \hat{x} は x はその固有値に対応して収束する。ここでの問題は、適当なオブザーバゲインによって $A + KC$ の固有値を望ましいものに配置できるかということである。 $A + KC$ の特性多項式は

$$\det(sI - A - KC) = \det(sI - A^T - C^T K^T) \quad (35)$$

となるので、この問題は (A^T, C^T) の極配置問題と等価となる。これが出るための条件は式 (29) を見るとちょうど双対システムの完全可制御と等価であることが分かる。つまり、状態フィードバックアルゴリズムにおいて A を A^T に、 B を C^T に置き換え、計算された状態フィードバックゲイン F の転置をとって K とすれば求めるオブザーバゲインが得られる。従って MATLAB には特にオブザーバを設計する関数は用意されていない。

[課題 4]

課題 1 の 2 自由度振動系に対して以下の設問に答えよ。

1. 状態フィードバックによって、閉ループ系の極を $-0.5+j\sqrt{3}$, $-0.5-j\sqrt{3}$, $-1+j\sqrt{3}$, $-1-j\sqrt{3}$ とする状態フィードバックゲイン F を求めよ。
2. 状態推定偏差系の極が $-10, -11, -12, -13$ となるような全次元オブザーバのオブザーバゲインを求めよ。
3. システムとオブザーバの併合系を考えると、以下の設問に答えよ。
 - (a) 状態フィードバックとオブザーバを用いた拡大系のシステムを定義しなさい。ただし、観測出力 y には外乱 v が加算されて観測されるものとする。
 - (b) 上で求めた拡大系のシステムの極を計算せよ。
 - (c) 外乱 v が平均値 0、分散 0.1 の正規確率外乱とし、またシステムの初期状態を $(1.0, 0.1, 0.0, 0.0)$ 、オブザーバの初期状態を零状態として、0 秒から 10 秒までシミュレーションしなさい。ただし、確率外乱のインターバルは 0.01 秒としなさい。
 - (d) オブザーバの極を $-20, -21, -22, -23$ と変更し、上記と同じ条件でシミュレーションし、結果を比較しなさい。

オプション 上記のシミュレーションを Simulink を用いて実行しなさい。ただし、正規性外乱は Random Number ブロックにより生成でき、インターバルはサンプル時間により設定できる。また、信号の足し算は Sum ブロックにより実現できる。

[Ex.4]

Let suppose a system considered in Ex.1, and answer the following questions.

1. Determine a state feedback gain F so that poles of the closed loop system with the feedback are assigned at $-0.5+j\sqrt{3}$, $-0.5-j\sqrt{3}$, $-1+j\sqrt{3}$, $-1-j\sqrt{3}$.
2. Determine an observer gain K of a full state observer whose poles are assigned at -10 , -11 , -12 , -13 .
3. Let consider an augment system composed of the controlled system and the observer determined in the previous question, and answer the following questions:
 - (a) Define the augmented system as a state space representation where we assume that the output of the controlled system is contaminated by a disturbance v .
 - (b) Calculate the poles of the augmented system defined in the previous question.
 - (c) Let assume that the disturbance is a Gaussian stochastic noise whose mean value, covariance are 0 and 0.1 respectively. Simulate the response of the closed loop system from $t=0$ [sec] to $t=10$ [sec] when the initial state of the controlled system is $(1.0, 0.1, 0.0, 0.0)$ and that of the observer is zero where the holding time of the disturbance is 0.01 [sec].
 - (d) Changing the poles of the observer to -20 , -21 , -22 , -23 , and simulate the response of the closed loop system under the same condition of the previous question.

Option Execute the above simulations using Simulink where Gaussian random signal can be generated by a Random Number block and holding time can be specified by the sampling time

in the block. Addition of signals can be realized by a Sum block.

特異値分解とデジタル制御系のシミュレーション

28 特異値分解と誘導行列ノルム

X, Y を実数値の空間として、 X から Y への線形変換を考えると

$$y = mx, \quad x, y \in R$$

と書くことができる。ただし、 m は適当な実数である。ここで、 x の絶対値と y の絶対値の比をゲインと定義すると、それは $\|m\|$ で与えられる。つまり、ゲインは入力信号 x の大きさが何倍されて出力の信号 y として観測されるかを表している。

次に R^n から R^m への線形変換を考えると形式的には上記と同様に

$$y = Mx \quad x \in R^n, y \in R^m \quad (36)$$

と書くことができる。ただし、 M は n 行 m 列の実数を要素とする行列である。 x をベクトルと考えてその要素を x_i ($i = 1, \dots, m$)、その長さをユークリッドノルム

$$\|x\| := \sqrt{\sum_{i=1}^m x_i^2} \quad (37)$$

で定義すると、これは x の大きさを測る一つの指標となる。また、同様に y の方もユークリッドノルムで大きさを測るとすると、そのノルムの比

$$\frac{\|y\|}{\|x\|} \quad (38)$$

を考えることができる。しかし、この比は一般に $\|x\|$ の大きさが一定でも、その'方向'によって異なるため、実数の場合のように一意に定めることができない。そこで、この比が最も大きくなるものを考えて、行列 M のゲインと考えるのである。このゲインのことをユークリッドノルムから誘導される行列ノルム (または、単に行列ノルム) といい、次の式で定義される。

$$\|M\| := \max_{\|x\|=1} \frac{\|y\|}{\|x\|} = \max_{x \neq 0} \frac{\|y\|}{\|x\|} \quad (39)$$

2つめの等号は行列を線形写像と考えた際の線形性から導出される。

この誘導ノルムと関係が深いのが行列の特異値分解であり、その分解は次のように与えられる。

[特異値分解]

任意の行列 $M \in R^{n \times m}$ は次のような形に分解することができる。

$$M = U \Sigma V^* \quad (40)$$

ただし、 $U \in R^{n \times n}, V \in R^{m \times m}$ はユニタリ行列で、

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & \cdot & \cdot & \cdots \\ 0 & \sigma_2 & \cdots & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdots & \sigma_r & \cdot & \cdots \\ \cdot & \cdot & \cdots & \cdot & 0 & \cdots \\ \cdot & \cdot & \cdots & \cdot & \cdot & \mathbf{0} \end{bmatrix}, \sigma_1 \geq \sigma_2 \cdots \geq \sigma_r > 0$$

である。

ここで、 σ_1 を最大特異値、 σ_r を最小特異値という。この分解より、 M の行列ノルムは σ_1 であり、 $\|M\| = \|M^T\|$ であることが簡単に示される。

[演習]

次のことを行列の特異値分解の性質を用いて示せ。

1. $\|M\| = \sigma_1$
2. $\|M\| = \|M^T\|$

29 システムの離散化・連続化

計算機を用いて連続時間の物理システムを制御する際の制御系の構成は一般に次のようになる。

図4で、D/A, A/D はそれぞれ D/A 変換器、A/D 変換器で、スイッチのように書かれているものはサンプラと呼ばれるもので、一般には

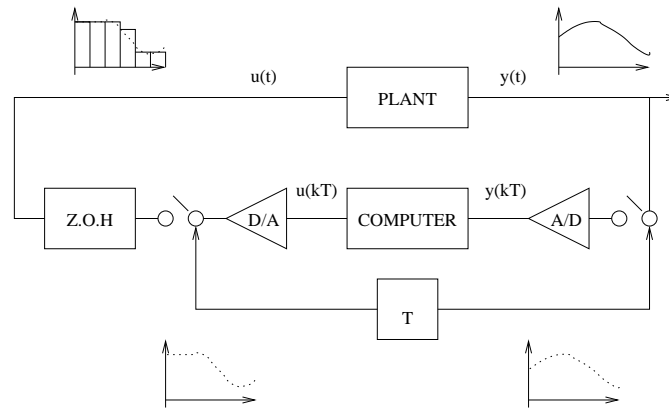


図 4: 取り扱うデジタル制御系

図のように入力と出力がサンプリング時刻 T の間隔で閉じられ、その瞬間の値が出力される。更に、図の Z.O.H. は Zero Order Holder (0 次ホールダ) の略で、入力された値を次の入力が入力されるまで一定にホールドするものである。(ホールダには 0 次ホールダ以外にも 1 次ホールダなど任意の次数のホールダが考えられるが、0 次ホールダが一般的に用いられている。) 回路的には D/A 変換と Z.O.H. が一緒に実現されているものが多い。また、計算機で実現する場合、完全に同時刻にサンプラを駆動することはできない場合が多いが、近似的に同時とみなすことが多い。(厳密にその遅れを考慮した設計法もあるが。) このような、連続時間の出力信号 $y(t)$ をサンプルし、ホールダの値ををサンプリング周期に同期して設定する制御系をデジタル制御系という。

この図で、PLANT の部分が次の状態方程式

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad x(0) = x_0 \\ y &= Cx + Du \end{aligned}$$

で表されるとし、対象のシステムを数列 $\{u(k)\}$ から $\{y(k)\}$ への関係を与えるものであると考え、その関係は次のように簡単に計算することができる。

ゼロ次ホールダの出力を入力した場合の出力列 $\{y(kT)\}$ は微分方程式の解より簡単に計算できる。入力 $u(t)$ がゼロ次ホールダにより

生成されているので、プラントへの入力は次式で与えられる。

$$u(t) = u(kT) \quad (kT \leq t < (k+1)T) \quad (41)$$

また、状態方程式の一般解より $u(t)$ に対するプラントの出力は $y(t)$ は

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (42)$$

$$y(t) = Cx(t) + Du(t) \quad (43)$$

で与えられる。従って、時刻 $t = kT$ での出力は上式の t に kT を代入して

$$x(kT) = e^{AkT}x_0 + \int_0^{kT} e^{A(kT-\tau)}Bu(\tau)d\tau \quad (44)$$

$$y(kT) = Cx(kT) + Du(kT) \quad (45)$$

と与えられる。次に、 $t = (k+1)T$ の出力も同様に

$$x((k+1)T) = e^{A(k+1)T}x_0 + \int_0^{(k+1)T} e^{A((k+1)T-\tau)}Bu(\tau)d\tau \quad (46)$$

$$y((k+1)T) = Cx((k+1)T) + Du((k+1)T) \quad (47)$$

として与えられるが、(41) の条件を用いると、システムを簡単な差分方程式に書き直すことができる。実際、

$$\begin{aligned} x((k+1)T) &= e^{AT}e^{AkT}x_0 + e^{AT}\int_0^{kT} e^{A(kT-\tau)}Bu(\tau)d\tau \\ &\quad + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)}Bu(\tau)d\tau \end{aligned} \quad (48)$$

となり、 $u(t)$ が $[kT, (k+1)T)$ で一定値 $u(kT)$ を取ることを考慮すると、

$$x((k+1)T) = e^{AT}x(kT) + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)}Bu(\tau)d\tau \quad (49)$$

となる。更に、上式右辺第二項は、

$$\int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)}Bu(\tau)d\tau = \int_0^T e^{A\tau}d\tau Bu(kT) \quad (50)$$

と変形できる。ここで、

$$\Phi := e^{AT} \quad \Gamma := \int_0^T e^{A\tau}d\tau B \quad (51)$$

と定義すると、 $y(kT)$ は $(x(kT), u(kT))$ を用いて

$$x((k+1)T) = \Phi x(kT) + \Gamma u(kT) \quad (52)$$

$$y(kT) = Cx(kT) + Du(kT) \quad (53)$$

と差分方程式により表される。ここで、 (Φ, Γ) はサンプリング・インターバル T が決まれば定数行列として計算できることに注意する。

このように、 (A, B, T) から (Φ, Γ) を求めることをシステムの離散化と呼ぶ。MATLAB では `c2d()` 関数で次のように計算することができる。

```
sysd = c2d(sys, T);
```

ただし、`sys` は連続系の LTI システム表現で、`sysd` は離散時間での LTI システム表現である。また、逆に (Φ, Γ, T) から (A, B) を求めることをシステムの連続化と呼び、MATLAB では次のように計算される。

```
sys = d2c(sysd);
```

ここで `d2c()` にサンプリング周期 T を渡す必要がないのは、`sysd` がその情報を持っているからである。

30 デジタル制御系のシミュレーション

MATLAB では簡単にデジタル制御系のシミュレーションを行なう関数が用意されていないので、シミュレーションを各サンプリング周期に分割して、その間を `ode45()` などの関数を用いてシミュレーションする。例として $T = 0.1$ として 0 秒から 10 秒間、 (A, B, C, D) で表されるシステムに、 $u(k) = \sin(\omega T k)$ ($k = 0, \dots, 1000 = (10/0.01)$) の入力をいれた場合のシミュレーションプログラムを示す。

```

global u

T = 0.1;
tspan = 0:T:(10.0+T);
v = sin(tspan* )';

Newx = X0';
x = Newx;

Newt = 0.0 ;
t = Newt;

n = 1;

for i=1:round(10.0/T)

    u = v(i,:);

    [Newt, Newx] = ode45('f', [(i-1)*T+ts, i*T], Newx(n,:))';

    [n,m] = size(Newx);

    x = [x;Newx(n,:)];
    t = [x;Newt(n,:)];

end

```

```
function dx = f(t, x)
% u is constant over the sampling period

dx = A*x + B*u
```

ここで、`size()` は行列のサイズを計算する関数 (実際は配列) で、引数が行列の場合は、戻り値は最初が行の数、2つ目が列の数となる。また、`tspan` のところで `10.0+T` としているのは、計算される状態の数が終端時刻での状態も含まれるために、入力のステップ数と状態のステップ数をそろえるためである。これは入力と状態を同じグラフに表示する際に必要となる。

[課題 5]

課題 1 の 2 自由度振動系に対して以下の設問に答えよ。

1. システムの可制御行列 V を計算し、それを特異値分解して Σ のランクが 4 であることを確認しなさい。
2. 入力として、-1 から +1 の一様乱数を $u(k)$ として入力した時の応答を、 $t=0$ から 10 秒シミュレーションし出力列 $\{y(k)\}$ を計算しなさい。ただし、システムの初期状態はゼロとし、サンプリング時間 T は 0.1 秒とする。
3. システムを $T=0.1$ 秒で離散化し、差分の状態方程式を求めよ。また、このシステムに前の問題と全く同じ入力列 $\{u(k)\}$ を入れた際の出力列 $\{y(k)\}$ を漸化式を解くことによりもとめ、前問と同じになることを確認しなさい。

[Ex.5]

Let assume a system considered in Ex.1, and answer the following questions.

1. Calculate a controllability matrix V and check that the rank of V is 4 to see Σ by using singular decomposition (SVD).
2. Assume that a holding interval of input is 0.1 [sec] and zero initial state. Simulate the response of the system from $t=0$ to $t=10$ when $u(k)$ is injected to the system where $u(k)$ is a uniform random signal from -1 to 1.
3. Discretize the system by the holding interval 0.1 [sec] and obtain a difference state equation. By recursively applying the difference equation check if the response of the system to the same input $\{u(k)\}$ in the last question is equal to the response in the last question.

31 Kalman フィルターとイノベーションモデル

モデルに基づく制御系解析・設計では、システムのモデルが必要である。ここでは、状態空間モデルを直接推定する部分空間同定法について演習を行う。部分空間同定法以外には、従来から良く用いられている入出力モデルのパラメータを推定してから、そこから状態空間モデルを決定する方法があるが、ここでは省略する。MATLAB に用意されている部分空間同定法の考え方に基づく `n4sid()` 関数では、外乱を陽な形で扱えるようなもので、次のようなシステムの表現を基本とし、そそれを変形して推定モデルとする。

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) + v(k), \quad x(0) = x_0 \\ y(k) &= Cx(k) + Du(k) + w(k) \end{aligned} \quad (54)$$

ただし、 $x_0, v(k), u(k)$ は平均ゼロの正規確率変数、正規白色確率過程で共分散行列は

$$E\{x_0 x_0^T\} = \Sigma_0, \quad E\left\{ \begin{bmatrix} v(k) \\ w(k) \end{bmatrix} [v(l) \ w(l)] \right\} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta(k-l) \quad (55)$$

で与えられる。ただし、 $\delta()$ はクロネッカのデルタ関数である。

このシステムのパラメータ (Φ, Γ, C, D) 並びに統計的パラメータである $\Sigma_0, (Q, S, R)$ が既知の場合、ここでは詳しく述べないがシステムの常態 $x(k)$ の最小分散状態推定器 (全状態オブザーバの一種) は次の Kalman Filter で与えられることが知られている。

$$\begin{aligned} \hat{x}(k+1) &= \Phi \hat{x}(k) + \Gamma u(k) + K(k)(\hat{y}(k) - y(k)) \\ \hat{y}(k) &= C \hat{x} + Du(k) \end{aligned} \quad (56)$$

ただし、 $K(k)$ はカルマンフィルターゲインと呼ばれるもので、次の式により計算される。

$$K(k) = (\Phi\Sigma(k)C^T + S)(C\Sigma(k)C^T + R)^{-1} \quad (57)$$

$$\Sigma(k+1) = \Phi\Sigma(k)\Phi^T + Q - K(k)(C\Sigma(k)C^T + R)K^T(k), \quad \Sigma(0) = \Sigma_0 \quad (58)$$

このフィルタに対して、 $\nu(k)$ を次の式で定義するとき、

$$\nu(k) := \hat{y}(k) - y(k) \quad (59)$$

$\nu(k)$ は正規白色過程となることが知られている。この $\{\nu(k)\}$ をイノベーションプロセスという。この $\nu(k)$ を使ってカルマンフィルタを書き直すと

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma u(k) + K\nu(k) \\ y(k) &= C\hat{x} + Du(k) + \nu(k) \end{aligned} \quad (60)$$

となり、確率外乱 ν だけによって y が生成生成されている形に書き直すことができる。このモデルのことをイノベーションモデルという。(ただし、カルマンフィルタゲインは定常のものを用いるとして定数としてある。)

MATLAB の部分空間同定法である `n4sid()` は上記のモデルを仮定してそのパラメータを推定するものであり、 (Φ, Γ, C, D, K) と \hat{x}_0 を推定結果として計算する。

32 n4sid の使い方

今、システムの入力の数が m 、出力の数が p とする。また、予想されるシステムの次数の最大のものを n とする。この時、システムを同定するために入力した入力列とそれに対応した出力列を下の図のように並べた行列 yu を作成する。ただし、 N はデータサンプル数であり、 T はサンプリング周期である。

$$\begin{pmatrix} y1(0) & y2(0) & \dots & yp(0) & | & u1(0) & \dots & um(0) \\ y1(T) & y2(T) & & yp(T) & | & u1(T) & & um(T) \\ \cdot & \cdot & & \cdot & | & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & | & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & | & \cdot & & \cdot \\ y1((N-1)T) & y2((N-1)T) & & yp((N-1)T) & | & u1((N-1)T) & & um((N-1)T) \end{pmatrix}$$

このデータから `n4sid()` によってシステムのパラメータを同定するには次のように記述する。

```
TH = n4sid(yu,1:n,p);
```

ここで n が完全に未知の場合は `1:n` の代わりに `'best'` と記述しても良い。この場合はデフォルトの `1:10` と同じ意味になる。また、 n を適当に小さく設定すると、システムの同定と次数の低次元化を同時に行なうことができる。この場合、 n が小さすぎると、特異値のグラフが出て、どの位その低次元化が妥当かを表示し、 n をその場で変更できるようになる。

`n4sid()` で得られる結果 `TH` は Theta フォーマットと呼ばれる ARX などのモデルを統一的に表現できる同定パッケージ特有のものであるので、これから通常の状態空間表現のパラメータを取り出し、離散の LTI システムを定義するには、次のように `th2ss()` 関数と `ss()` 関数を用いる。

```
[Phi, Gamma, C, D, K, x0] = th2ss(TH);
sysd = ss(Phi, Gamma, C, D, T);
```

また、これより等価な連続系を求めるのは前に説明したように次のようにすればよい。

```
sys = d2c(sysd);  
sys.A  
sys.B  
sys.C  
sys.D
```

ここで、 $sys.A, sys.B, sys.C, sys.D$ は連続系の状態空間表現のそれぞれ A, B, C, D 行列を取り出す演算である。(離散系の LTI システムから Φ, Γ を取り出すのも $sysd.A, sysd.B$ と $.A, .B$ を用いる。本演習では連続系と離散系を明確に区別するためい A, B と Φ, Γ を用いているが、離散系のみを考えているような場合には、 Φ, Γ の代わりに A, B を用いる場合も多いため、そのような演算になっている。)

[課題 6]

課題 1 のシステムに対して以下の問題に答えよ。

1. 課題 5 の (2) で求めた入出力データから `n4sid()` によって離散のシステムを同定し、連続化することによって連続系の状態空間表現を求め実際のシステムと比較しなさい。
2. 前の問題の出力データに平均値 0、標準偏差 0.1 の正規確率外乱を加算して、同様に `n4sid()` によって連続系の常態空間表現を求めよ。更に、推定されたシステムの極、零点、ボード線図を実際のものと比較しなさい。(極、零点の表示には `pzmap()` 関数を用いよ。)
3. 前述の問題で $n = 2$ として `n4sid()` 用いて低次元化したモデルを同定し、ボード線図を比較せよ。

[Ex.6]

1. Identify a system using `n4sid()` based on an input-output data obtained in (2) of Ex.5, and determine a continuous state space model. Compare the identified continuous system with the actual continuous model.
2. Add a noise sequence whose mean value is 0 and whose standard deviation is 0.1 to the input-output data used in the last question, and determine a continuous state space model using the contaminated data by the `n4sid()`. Furthermore, calculate poles, zeros and Bode digram and compare those with the actual ones. (Plot the zeros and poles by `pzmap()`.)
3. Assuming $n = 2$ in the last question, identify a continuous system by `n4sid()` and compare the system with the system obtained in the last question by plotting Bode diagrams.

システムの性質 (システムゲイン)

33 写像としてのシステムとその空間

実数値の空間 R から実数値の空間 R への線形変換 (写像、オペレータ) は

$$y = mu, \quad u, y \in R \quad (61)$$

と表現できる。ここで m は適当な実数値であるが、 m は R の要素であると同時に、1つの線形変換を表している。このように考えると、実数全体の集合は実数値の空間 R から実数値の空間 R への線形変換全体の集合を表していることになる。

今、実数直線上の2点の距離はその差の絶対値で表される。例えば、 $u_1, u_2 \in R$ の間の距離は

$$|u_1 - u_2| \quad (62)$$

で表される。



一方、 m も実数空間の要素であるから、その絶対値

$$|m| \quad (63)$$

を考えることができる。これは0という線形変換からの距離

$$|m - 0| \quad (64)$$

を表しているとも考えることもできる。ここで $|y|$ と $|u|$ の比を考えると、

$$\frac{|y|}{|u|} = \frac{|mu|}{|u|} = |m|, \quad |u| \neq 0 \quad (65)$$

となり、これは線形変換 m のゲインと考えられるが、これは m の空間での原点からの距離と一致している。



このように、一般にある空間からある空間への (線形) 変換または写像が与えられた時、写像からなる空間を考えて、そこでの距離を定義することによって写像のゲインを定義することができる。一般の空間での距離は数学的にはノルムによって定義されるため、写像のゲインを考えることは、写像の空間に適切なノルムを定義することとなる。特に、実数から実数への線形変換の例のように、入力空間のノルムと出力空間のノルムの比から定義される値と、写像の空間で適当に定義したノルムの値とが一致する時、写像の空間に定義されたノルムは誘導ノルム (Induced Norm) と呼ばれる。

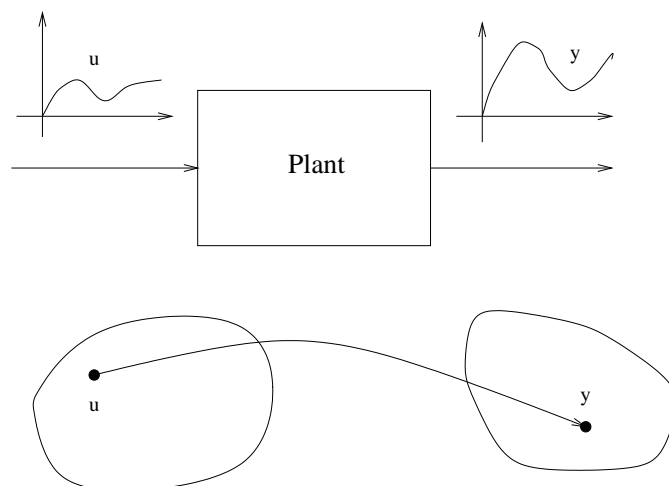
写像の空間へのノルムの導入は一意ではないため、そこで定義されたノルムは必ずしも誘導ノルムではない。また、誘導ノルムを考える場合、写像の定義域と値域でのノルムの定義にも依存することに注意する。写像のノルムを誘導ノルムで定義する場合でも、そうでない場合でも、写像の空間へ数学的に導入できるノルムは多様であるが、工学的に意味のあるものを使う必要があることは言うまでもないことである。

我々の考えているシステムは、状態空間表現すると

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad u(t) \in R^m, y(t) \in R^p, x(t) \in R^n \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (66)$$

となる。特に、初期状態を零と考えると、 y は u のみの関数となり、下の図のように、時間関数 u を時間関数 y に変換しているものとみなせる。ここで、時間 t の関数 $f(t)$ を考えるとしたとき、 $f(t)$ は時

刻 t を指定したときの $f(\cdot)$ の値であり、 (t) を省略した時は値ではなくて、時間関数そのものの、等価であるが時間 t に対する $f(t)$ グラフの全体を表していると考ええる。



このシステムの零初期状態からの解は

$$y(t) = \int_{-\infty}^{\infty} H(t - \tau)u(\tau)d\tau \quad (67)$$

ただし、

$$H(t) = \begin{cases} Ce^{At}B + D\delta(t) & (t \geq 0) \\ 0 & (t < 0) \end{cases} \quad (68)$$

と表すことができる。ここで、 δ は Dirac のデルタ関数である。通常の因果的なシステムで、 $u(t) = 0 (t < 0)$ の場合には

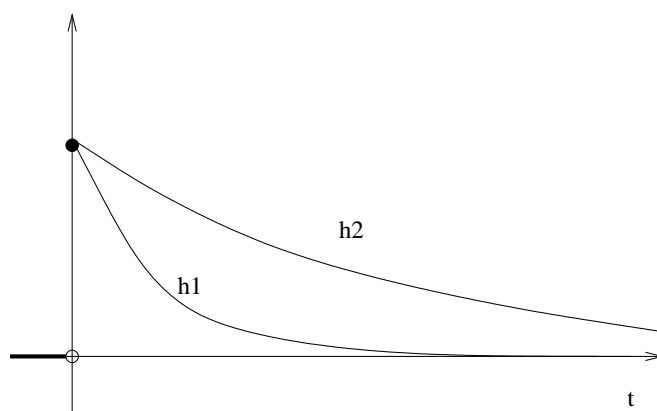
$$y(t) = \int_0^t H(t - \tau)u(\tau)d\tau \quad (69)$$

となる。このように、LTIシステムを入力関数から出力関数への変換であると考えた場合、その特性は重み関数行列 H にのみ依存する。この重み関数の Laplace 変換が伝達関数行列であるから、伝達関数行列のみに依存すると考えても良い。LTIシステムの入出力特性は重み関数行列または伝達関数行列で決まるため、システムを写像と考えた際のゲイン (これをシステムゲインと呼ぶ) は、重み関数からな

る空間でのノルム、又は伝達関数行列からなる空間でのノルムを定めることと等価となる。

34 H_2 空間

今、下図のような2つの重み関数 h_1 と h_2 と考えた時、



各時刻での振幅を自乗して $-\infty$ から ∞ まで積分した量を考えると、 h_1 より h_2 の方が大きな値となる。(これは信号のパワーを表していると考えることができる。もちろん、恒等的に零の重み関数に対応するものが最小となる。) このような負の時刻で零の値を取る因果的な重み関数で、 $-\infty$ から ∞ まで積分した値:

$$\|h\|_2 := \sqrt{\int_{-\infty}^{\infty} |h(t)|^2 dt} \quad (70)$$

が有界な値となる関数の空間を H_2 空間と呼ぶ。

多入力多出力なシステムの場合は、重み関数が重み関数行列となるので被積分値が関数値の自乗で定義できないので、同様の考え方から、関数行列の全ての要素の自乗を足し合わせた、各時刻での Frobenius ノルムの自乗を積分することとする。

つまり、 H_2 ノルム、 $\|H\|_2$ を次の式により定義する。

$$\|H\|_2^2 = \int_0^\infty \|H(t)\|_F^2 dt = \int_0^\infty \text{trace}\{H(t)H^T(t)\}dt = \int_0^\infty \text{trace}\{H^T(t)H(t)\}dt \quad (71)$$

また、具体的な(A,B,C)を用いてH(t)を表現すると、明らかにD = 0が必要であり、教科書のように

$$\|H\|_2^2 = \int_0^\infty \|H(t)\|_F^2 dt = \int_0^\infty \text{trace}\{Ce^{At}BB^Te^{A^Tt}C^T\}dt = \int_0^\infty \text{trace}\{B^Te^{A^Tt}C^T Ce^{At}\}dt \quad (72)$$

となり、可制御グラミアンQ、可観測グラミアンPを次の式で定義すると、

$$P := \int_0^\infty e^{At}BB^Te^{A^Tt}dt$$

$$Q := \int_0^\infty e^{A^Tt}C^T Ce^{At}dt$$

H₂ ノルムは

$$\text{trace}\{CPC^T\} = \text{trace}\{B^TQB\} \quad (73)$$

となる。従って、H₂ ノルムの計算は可制御グラミアン、可観測グラミアンの計算に帰着される。

重み関数行列を定義する際、わざわざ負の時間まで考えるのは、負の時間でのみ値があり、負の時間での同様な積分が有界な関数のクラスを考え、関数行列の内積を

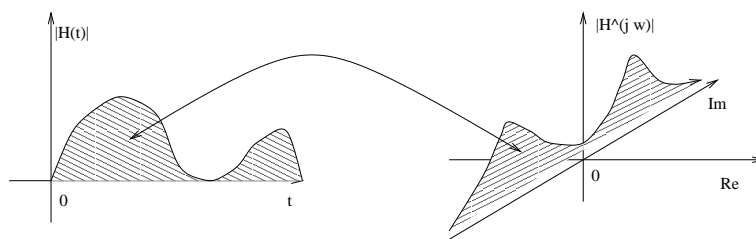
$$\langle H_1, H_2 \rangle := \int_{-\infty}^\infty \text{trace}\{H_2(t)H_1^T(t)\}dt \quad (74)$$

と定義することによって、2つの関数空間が、 $-\infty$ から ∞ までの Frobenious ノルムの自乗の積分が有界な関数の集合 $L_2^{m \times p}(-\infty, \infty)$ の直交空間になっているからである。また、両サイド Laplace 変換によって H_1, H_2 を変換したものを \hat{H}_1, \hat{H}_2 とすると、これら2つの空間も直交空間になっており、関数空間の直交性を保存した変換先の空間を考えることができるのである。これは、 $\hat{H}(\infty) = 0$ で、複素開右半面で解析的な関数と、複素開左半面で解析的な関数とにちょうど

対応する。もともと H_2 空間は複素関数を $\hat{H}(s)$ (必ずしも伝達関数行列でなくともよい) とすると、これが複素開右半面で解析的で、

$$\|\hat{H}\|_2^2 := \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}\{\hat{H}(j\omega)\hat{H}^*(j\omega)\}d\omega < \infty \quad (75)$$

となる複素関数の空間として定義されている。ただし、 $*$ は共役転置である。しかし、Parseval の定理によってこの値は時間領域で定義した値と定数倍を除いて一致するのである。(下図参照)



また、複素関数の空間での直交性を考える場合は、内積を

$$\langle \hat{H}_1, \hat{H}_2 \rangle := \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}\{\hat{H}(j\omega)\tilde{\hat{H}}(j\omega)\}d\omega \quad (76)$$

により定義する。ただし、 $\tilde{\sim}$ は

$$\tilde{\hat{H}}(s) = \hat{H}^T(-s) \quad (77)$$

である。この $\tilde{\sim}$ によって、右半面で解析的な関数は左半面で解析的になり、 $H(\infty) = 0$ と合わせてコーシーの積分定理より、右半平面で解析的な関数と左半平面で解析的な関数の内積が零になるのである。

35 H_∞ ノルム

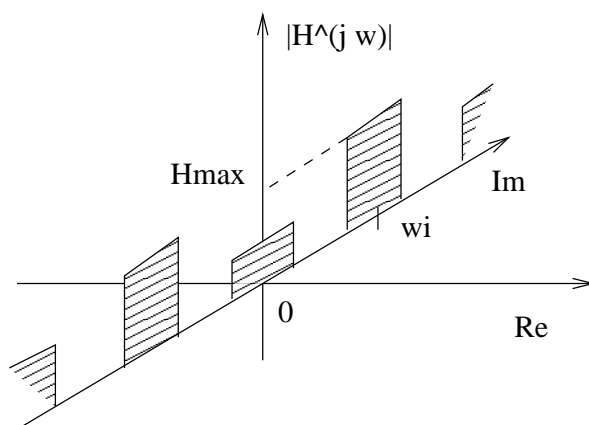
前節の H_2 のノルムは時間領域では信号のパワーを表しているが、同時に式 (75) より各周波数でのパワーを全周波数に渡って積分したものであることが分かる。これより、 H_2 ノルムは全周波数に渡る平均的な指標と考えることができる。従って、機械系などの共振が問題になるような全周波数に渡る特性よりも、特定の周波数でどのよ

うな特性を持っているかを評価したい場合には H_2 ノルムは不向きであることが分かる。

スカラのシステムに対して、伝達関数が安定である時、正弦波入力に対する定常応答は

$$y(t) = |\hat{H}(j\omega)| \sin(\omega t + \text{Arg}\hat{H}(j\omega)) \quad (t \gg 0) \quad (78)$$

で与えられた。従って、出力の振幅が最大になるものは周波数伝達関数の絶対値が最大になる周波数 ω を探せば良いことが分かる。このピークを解析的なノルムとして定義するために、次の有限次元ベクトル空間での問題を考えよう。今、下図のように \hat{H} が n 個の周波数帯域でのみ値を持ちその幅が $2\pi\Delta\omega$ で、その周波数帯域でノルムが一定値 $|\hat{H}(j\omega_i)|$ ($i = 1, \dots, n$) を持つとしよう。



そうすると、 H_2 ノルムは

$$\|\hat{H}\|_2 = \left\{ \sum_{i=1}^n |\hat{H}(j\omega_i)|^2 \Delta\omega \right\}^{1/2} \quad (79)$$

となる。この式は各周波数帯での絶対値を 2 乗して和を取り、それを $1/2$ 乗していると考えることができる。これを拡張して次のノルムを考えよう。

$$\|\hat{H}\|_p = \left\{ \sum_{i=1}^n |\hat{H}(j\omega_i)|^p \Delta\omega \right\}^{1/p} \quad (80)$$

ここである周波数帯 $i_k (k = 1, \dots, m)$ でその絶対値が最大で \hat{H}_{max} でそのそのピークが N 点あると仮定して上の式を書き直す。

$$\|\hat{H}\|_p = \left\{ \hat{H}_{max}^p \sum_{i=1}^n |\hat{H}(j\omega_i)/\hat{H}_{max}|^p \right\}^{1/p} \quad (81)$$

ここで $p \rightarrow \infty$ とすると、 $|\hat{H}(j\omega_i)/\hat{H}_{max}|$ が 1 未満の所では p が大きくなるとその指数乗は零に近付くので

$$\lim_{p \rightarrow \infty} \|\hat{H}\|_p \approx \lim_{p \rightarrow \infty} \left\{ \hat{H}_{max}^p N \Delta\omega \right\}^{1/p} = \hat{H}_{max} \quad (82)$$

となり、最大の絶対値を取り出すことができることが分かる。この考え方を一般の連続な関数 $\hat{H}(s)$ に対して定義したものが H_∞ ノルムである。

一般の伝達関数行列においては、周波数 ω を固定した場合のゲインは絶対値では考えられないので、各周波数での $H(j\omega)$ を複素行列と見て行列ノルムを計算し、周波数領域でその行列ノルムの最大を H_∞ ノルムと定義する。式で表現すると

$$\|\hat{H}\|_\infty = (ess.) \sup_{-\infty \leq \omega \leq \infty} \|\hat{H}(j\omega)\| = (ess.) \sup_{-\infty \leq \omega \leq \infty} \bar{\sigma}(\hat{H}(j\omega)) \quad (83)$$

ここで、 $\bar{\sigma}(\cdot)$ は最大特異値である。注意すべきことは、 $\hat{H}(s)$ は閉複素右半平面で解析的あるという条件は付いていることに注意する。この付帯条件を言葉で表現する代わりに式で表現する場合には、教科書のような定義になる。

この H_∞ ノルムは入出力信号の誘導ノルムとして考えることができ、入力空間、出力空間をそれぞれ

$$L_2^m(0, \infty) := \left\{ u(t) \in R^m \mid \|u\|_2 := \sqrt{\int_0^\infty u^T(t)u(t)dt} \right\} \quad (84)$$

$$L_2^p(0, \infty) := \left\{ u(t) \in R^p \mid \|y\|_2 := \sqrt{\int_0^\infty y^T(t)y(t)dt} \right\} \quad (85)$$

とすると、 H_∞ ノルムは $L_m(0, \infty)$ から $L_p(0, \infty)$ への誘導ノルムとなる。つまり、

$$\|H\|_\infty = \sup_{u \in L_2(0, \infty)} \frac{\|y\|_2}{\|u\|_2}, \quad \|u\|_2 \neq 0 \quad (86)$$

が成り立つ。

[演習]

H_∞ ノルムが

$$\frac{\|y\|_2}{\|u\|_2}, \|u\|_2 \neq 0$$

の上界 (upper bound) になっていることを Parseval の定理と特異値の性質を使って示せ。(上限 (sup) になっていることは、適当な実数値の関数列を作る必要があるため、教科書のようにちょっと面倒である。)

教科書の二分探索による H_∞ ノルムの値は MATLAB では `hinfnorm()` によって次のように計算することができる。

```
hinfnorm(P, epsilon)
```

ただし、 P はシステム行列で、`epsilon` は探索の精度である。このシステム行列表現は今後ロバスト制御を MATLAB で扱う上で基本となるデータ構造で、情報としては、状態空間表現 (A, B, C, D) に対して、

$$P := \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (87)$$

により与えられ、MATLAB での LTI システムの状態空間表現の変数が `sys` であるとき、次のコマンドにより変換される。

```
P = pck(sys.A, sys.B, sys.C, sys.D)
```

逆に P から状態空間表現の (A, B, C, D) を取り出すのは

```
[A, B, C, D] = unpck(P)
```

となる。また、 H_2 ノルムは次のコマンドにより計算される。内部では、実は可制御グラミアンを計算しているだけである。

`h2norm(P)`

[課題 7]

課題 1 のシステムに対して以下の問題に答えよ。

1. 課題 1 で考えたシステムの H_2 ノルムを計算せよ。
2. 課題 1 で考えたシステムの H_∞ ノルムを計算し、課題 2 で求めたボード線図のピークとほぼ一致していることを確認せよ。

[Ex.7]

Let consider a system considered in Ex.1 and answer the following questions.

1. Calculate the H_2 norm of the system considered in Ex.1.
2. Calculate the H_∞ norm of the system and compare the norm with the peak value of the gain plot of the Bode diagram drawn in Ex. 2.

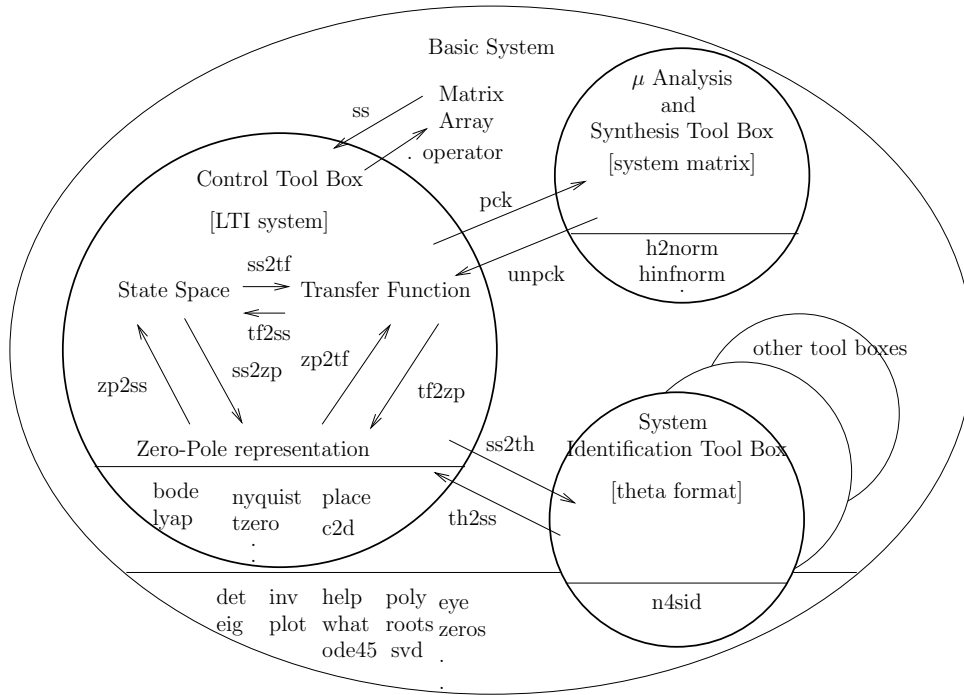
36 MATLABでのシステム表現

今まで MATLAB の演習を行なう上で種々のコマンドを用いてきたが、関数によってシステムの表現が異なっていた。これは、実は MATLAB が基本のシステムと機能的に分割されたツールボックスと呼ばれる関数の集まりから構成されており、そのツールボックスに適した、又は開発者の好みの表現形式が用いられているからである。本演習で用いているツールボックスは

コントロールツールボックス μ 解析・設計ツールボックス システム同定ツールボックス	古典的な制御系の基本解析ツール ロバスト制御系解析・設計ツール システム同定ツール
------------------------------------------------------	-------------------------------------------------

である。(従って、基本の MATLAB システムを用意しただけでは上記のツールボックスによって提供されている関数は使えないので注意する。) これらツールボックスとそこで用いられているシステム表現並びに変換関数をまとめて図示すると次のようになる。□内はシステムの表現形式名で、矢印のところに書かれているのは、各システム表現に変換する関数名である。また、各ツールボックスの横線より下の部分には本演習で用いてきた関数がどのツールボックスで提供されているかを示している。

MATLAB



[課題 8]

課題 6 の 3. で、システムの次元を $n = 2$ として同定した。このように高次の振動モードを定性的にないものと思ってシステムを同定することは良く行なわれる。このような定性的なモデリングが制御系設計に対してどのように影響するか見るために、次の問題を考える。

1. 課題 6 の 3. と同じように $n = 2$ と仮定してシステムを同定しなさい。ただし、真のシステムのバネ定数、粘性係数は $K_2 = 16.0, D_2 = 0.03$ とし、今回は観測ノイズはゼロとする。この同定されたシステムに対して、閉ループの極が $-1 + j\sqrt{3}, -1 - j\sqrt{3}$ とする状態フィードバックゲインとオブザーバの極が $-10, -11$ となるオブザーバを設計し、同定対象のシステムに適用しその応答を示しなさい。(今後この修正されたパラメータを持つシステムを課題 8 のシステムと呼ぶ)
2. 前問の応答が不安定になった場合、再同定はせず、閉ループ系の極や、オブザーバの極のみを調整し、応答が開ループ系のものよりも良くなるようなコントローラを 1 つ見つけなさい。また、この時の調整過程を説明しなさい。

[Ex. 8]

In the problem 3 of Ex. 6, we identified the plant assuming that the order of the system is 2, i.e., $n = 2$. This kind of assumption is made to identify a system when we simply assume that there are no high frequency modes. In order to see how such modeling affects the controller design, we consider the following questions.

1. As in the problem 3 of Ex. 6, identify a system assuming $n = 2$ where in this case we assume that true parameters of the system are given as $K_2 = 16.0, D_2 = 0.03$ and that there is no observation noise this time. For the identified plant, design a state feedback gain so that poles of the closed loop system are allocated at $-1 + j\sqrt{3}, -1 - j\sqrt{3}$ and design a full state observer

whose poles are $-10, -11$. Then show the response when the designed controller is applied for the true system. (Hereafter, the plant with the new parameters is referred as the plant in Ex. 8.)

2. If the closed loop system becomes unstable in the last question, find a controller which stabilizes the plant by just modifying the feedback gain or/and observer poles without re-identification. Explain the procedure how you found the controller.

37 不確定性の表現

制御系を設計する際、対象のシステムの動特性をモデル化し、その特性を仮定して制御系を設計することが良く行なわれる。このときその仮定した特性をノミナルな (nominal, 公称の) 特性という。特に、特性を伝達関数で表現した場合には、その伝達関数をノミナルな伝達関数という。そして、ノミナルな特性と実際の (真) の特性との差を不確定性または変動という。

実際のシステムでこの変動が生じる原因は色々があるが、主として次の3つに大別される。

1. 操作条件の変動
2. モデルの簡略化
3. パラメータ推定の誤差

例えば、図5のように、ロボットマニピュレータのハンドが物体を把持していない場合をノミナルな特性と考え、物体を把持している場合は変動した特性と言える。また、図6のような鉄鋼の圧延システムのように、操作部と検出部が離れている場合、圧延された鉄板の搬送速度 v の変動が不確定性となる。

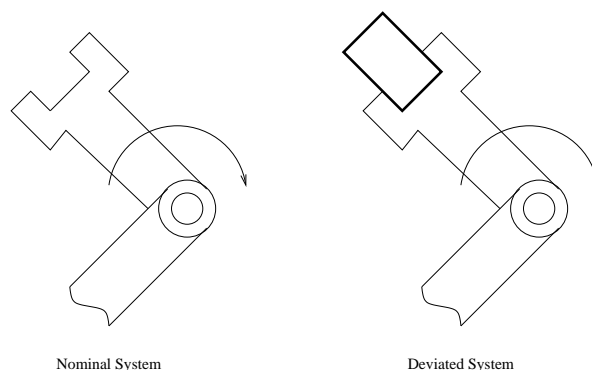


図 5: マニピュレータの物体把持に伴う特性変動

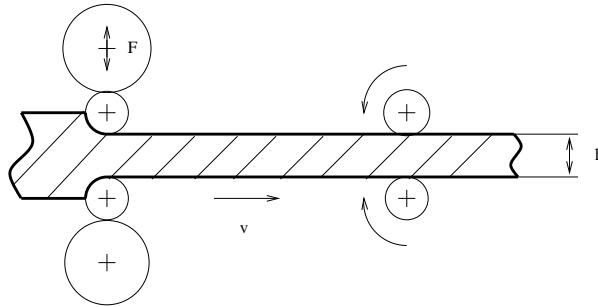


図 6: 圧延システムの運搬速度変化に伴う特性変動

また、平面内のダンピング係数の小さなフレキシブルな梁（アーム）の回転トルクから先端変移までの伝達特性は

$$H(s) = \frac{k_0}{s^2} + \sum_{i=1}^{\infty} \frac{k_i}{s^2 + \omega_i^2} \quad (88)$$

のような無限次元のシステムとなるが、普通は適当なモード N までを考えて、ノミナルな伝達関数 $\hat{H}(s)$ を

$$\hat{H}(s) = \frac{k_0}{s^2} + \sum_{i=1}^N \frac{k_i}{s^2 + \omega_i^2} \quad (89)$$

として、残りの部分を不確定性 $\Delta_a(s)$:

$$\Delta_a(s) = \sum_{i=N+1}^{\infty} \frac{k_i}{s^2 + \omega_i^2} \quad (90)$$

と考える。この考え方は課題 6-3 でも用いたものである。今、この不確定性と、真の伝達関数の関係は

$$H(s) = \hat{H}(s) + \Delta_a(s) \quad (91)$$

となっている。このように、不確定性が、真の伝達関数とノミナルな伝達関数の差で表されるような不確定性を加法的な不確定性または加法的変動と呼ぶ。

また、モータ系をサーボアンプと機械系と分けて考えて、サーボアンプ部を一次遅れ系で表されるとした場合、電圧からモータの回転

角までの伝達関数は図7のようになる。しかし、普通電気系の時定数 T_a は機械系の時定数 T_m より十分小さいので、その部分の動特性は無視して矢印下のような伝達関数として考える場合も多い。この場合、真の伝達関数 $H(s)$ は

$$H(s) = \frac{AK}{s(T_ms + 1)(T_as + 1)} \quad (92)$$

であるのに対して、ノミナルな伝達関数 $\hat{H}(s)$ は

$$\hat{H}(s) = \frac{AK}{s(T_ms + 1)} \quad (93)$$

で与えられる。この場合、一つの不確定性の表現は、 $H(s)$ と $\hat{H}(s)$ の比

$$\Delta_0(s) = H(s)/\hat{H}(s) = \frac{1}{T_as + 1} \quad (94)$$

である。しかし、不確定性の大きさをオペレータのノルムで表現しようとした場合、上記のように単純な比で不確定性を表現すると、不確しかさがない場合の不確定性の伝達特性が1となり都合が悪い。そこで、比で表された不確定性から1を引いたものを改めて不確定性であると考えるのである。これを $\Delta_m(s)$:

$$\Delta_m(s) := \Delta_0(s) - 1 \rightarrow \Delta_0(s) = 1 + \Delta_m(s) \quad (95)$$

として定義する。これを用いて $H(s)$ と $\hat{H}(s)$ を表現すると

$$H(s) = \hat{H}(1 + \Delta_m) \quad (96)$$

となる。このように真の伝達関数とノミナルな伝達関数を与える不確定性 $\Delta_m(s)$ を、乗法的な不確しかさまたは乗法的変動という。このように定義すると、不確定性がない場合、 Δ_m の H_∞ のノルムなどが零となり都合が良いのである。多入出力系の乗法的変動の場合、伝達関数の積がに交換則がなりたたないため、入力部分の不確定性と出力部分の不確定性とはそれぞれ別々に考えなければならない点に注意する。

加法的変動と乗法的変動を伝達関数で表現すると図8のようになる。

また、システム同定のように、構造は正しく仮定しても、同定のために用いる入力特性や、外乱などによって同定されるモデルが変化してしまうような場合も、不確定性であると考えることができる。

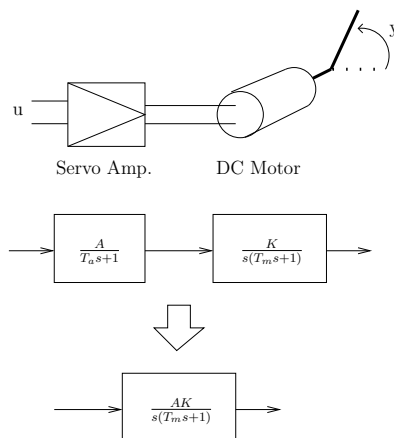


図 7: アンプ-モーター系の伝達関数の近似

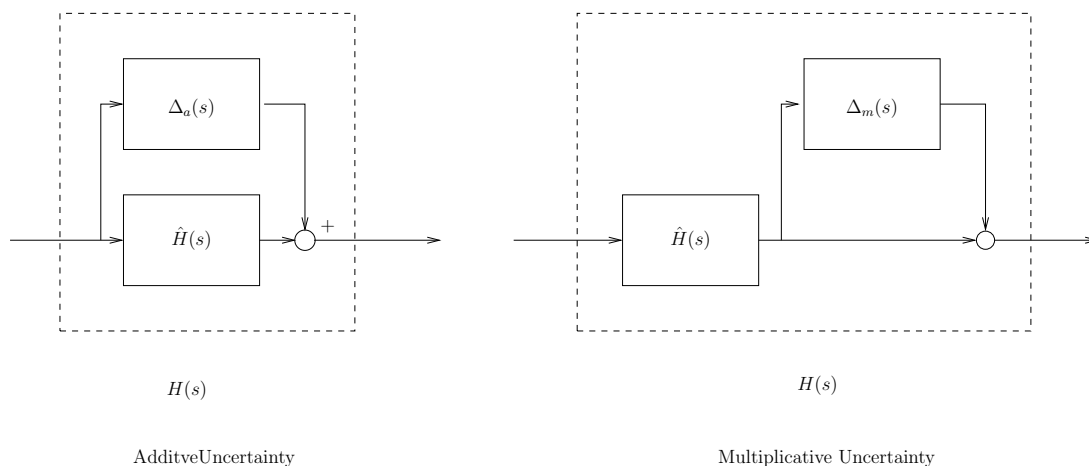


図 8: 加法的な不確かさと乗法的な不確かさ

38 ロバスト安定性とスモールゲイン定理

今、真のシステムに対してフィードバックコントローラ C で閉ループ系を組むことを考える。図 9 のように、不確定性ブロックの出力

信号を w , 入力信号を z とし、 w から z までの伝達特性を $M(s)$ とすると、いずれの不確定性表現を用いても、不確定ブロックに対して $M(s)$ でフィードバックした形になる。

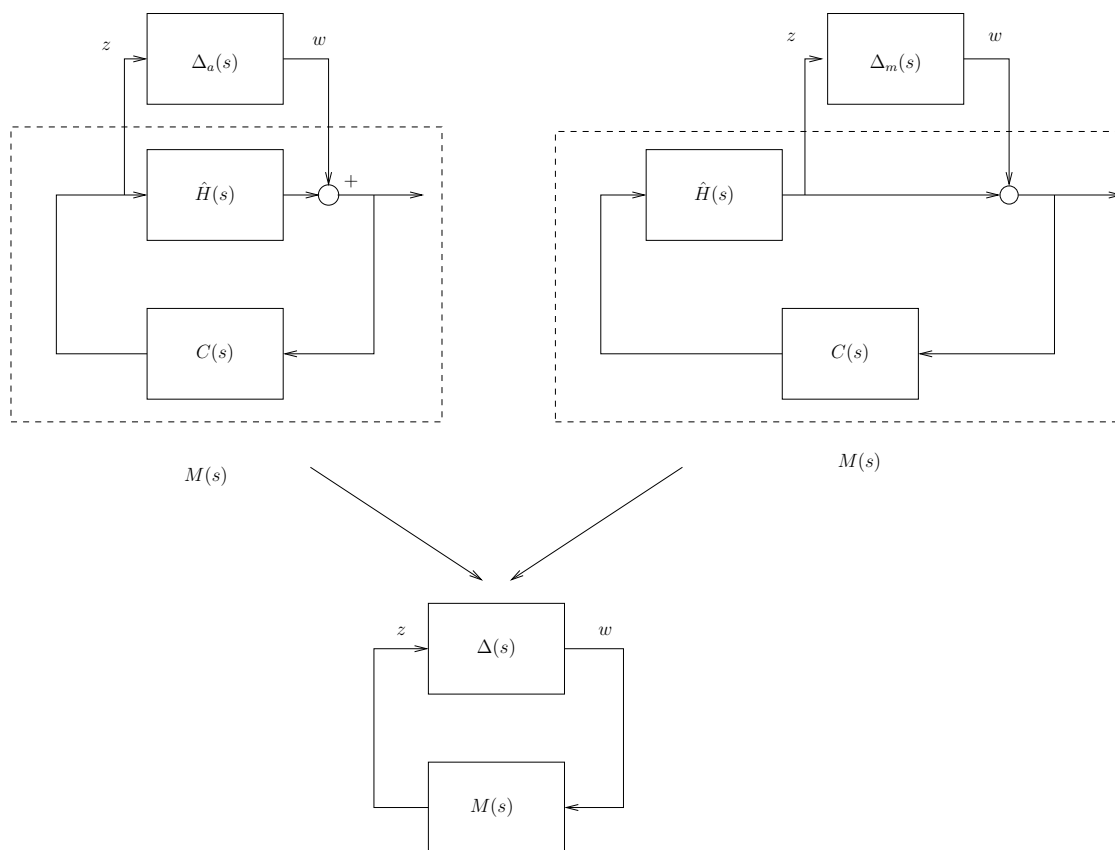


図 9: 不確定性を含むフィードバック系

この不確定性を含むシステムの安定性については、以下の定理が成り立つ

[定理](スモールゲイン定理)

$M(s) \in \mathcal{RH}_\infty$ で $\gamma > 0$ とする。このとき、全ての条件を満たす不確定性 $\Delta(s)$ に対して、図 9 の閉ループ系が well-posed で内部安定であるのは、

- (a) $\|\Delta(s)\|_\infty \leq 1/\gamma$ if and only if $\|M(s)\|_\infty < \gamma$.
- (b) $\|\Delta(s)\|_\infty < 1/\gamma$ if and only if $\|M(s)\|_\infty \leq \gamma$.

が成り立つときである。

[注] 上記の定理で必要十分条件になっているのは、条件を満たす全ての $\Delta(s)$ を考えているからであり、特定の不確定性を考えたときは十分条件であることに注意する。

したがって、図9のような系の安定性を判別するのは、それぞれのブロックの H_∞ ノルムを計算すれば安定性の十分性はチェックできる。

上述の定理では $\Delta(s)$ のゲインの周波数依存性は考慮されていない。例えば、 $\Delta(s)$ が次のいずれの伝達関数で与えられても

$$\Delta(s) = \frac{a}{s+a} \quad \text{or} \quad \Delta(s) = e^{-sT}, \quad a > 0, T > 0 \quad (97)$$

$\|\Delta(s)\|_\infty = 1$ となる。

$\Delta(s)$ のゲインの周波数依存性を考慮するために、 $\Delta(s)$ の各周波数に対するゲインの上界が

$$\bar{\sigma}(\Delta(j\omega)) \leq |w(j\omega)| \quad (98)$$

与えられているとする。ただし、 $w(s)$ は安定なスカラ関数 (重み関数)。このときは、上記 $M(s)$ を $w(s)M(s)$ と修正して、 $\|w(s)M(s)\|_\infty < 1$ となるようなロバスト安定化コントローラを設計すればよい。なぜなら、ロバスト安定化の十分条件は

$$\|\Delta(s)M(s)\|_\infty < 1 \quad (99)$$

であるが、 $\Delta(s)$ のゲイン条件より

$$\|\Delta(s)/w(s)w(s)M(s)\|_\infty \leq \|\Delta(s)/w(s)\|_\infty \|w(s)M(s)\|_\infty \leq \|w(s)M(s)\|_\infty \quad (100)$$

である。(最後の不等式にゲイン条件を使った。) 従って、 $\|w(s)M(s)\|_\infty < 1$ であればロバスト安定性の条件満たすことが分かる。

[課題 9]

課題 8 の対象のシステムの伝達関数を $H(s)$, 課題 8 の 1. で $n = 2$ として同定したシステムを $\hat{H}(s)$ とする。この時、下記の問題に答えなさい。

1. 加法的変動の伝達関数を計算し、その伝達関数の Bode 線図を描きなさい。
2. 課題 8 の 2. で求めた安定化コントローラと対象のシステムの閉ループ系に対して、それぞれの不確定性ブロックの出力 w から入力 z までの特性を LTI モデルとして求めよ。
3. それぞれ 1., 2. の H_∞ ノルムを計算し、スモールゲイン定理の条件を満たしているか確かめなさい。もし、満たしていない場合は、なぜ安定化できたのか考察しなさい。

[Ex.9]

Let assume that $H(s)$ is a true transfer function of the system in Ex. 8 and that $\hat{H}(s)$ is the identified model of the system whose degree n is 2. Answer the following questions.

1. Calculate a transfer function of the additive disturbance and plot a Bode diagram of it.
2. Let consider the closed loop system which consists of the stabilizing controller and the plant used in 2. of Ex. 8. Determine a LTI model whose input port is the output port of the disturbance w and whose output port is the input port of the disturbance z .
3. Calculate H_∞ norms of the systems in the above questions 1. and 2., and check if the small gain condition is satisfied or not. If the condition is not satisfied, explain why the closed loop system becomes stable.

ただし、二つの LTI モデルのシステム sys1 , sys2 があった場合、これらのシステムの並列結合、直列結合、ネガティブフィードバック

結合 (前向きブロックが sys1, 後ろ向きブロックが sys2) の LTI システムは次の関数で計算できる。

結合	関数
並列	parallel(sys1, sys2)
直列	series(sys1, sys2)
ネガティブフィードバック	feedback(sys1, sys2)

また、オブザーバはプラントの入力と出力を入力とするシステムであるので、オブザーバの状態空間表現を

$$\dot{\hat{x}} = A\hat{x} + Bu + K(\hat{y} - y) = (A + KC)\hat{x} + Bu - Ky \quad (101)$$

とすると、オブザーバの併合系のコントローラを出力フィードバックコントローラとするためには、 $u = F\hat{x}$ という事実を利用して、状態方程式を、

$$\dot{\hat{x}} = (A + KC + BF)\hat{x} - Ky \quad (102)$$

として、プラントの出力のみを入力に持つ出力フィードバックコントローラの形に変形してからブロック線図を考える。

あるいは、入力の性質を陽に考慮しない場合は次のように考える。これは、構造を残したままシミュレーションをする場合や、部分的なループ特性を考える場合には有用となる。オブザーバ併合系をブロック線図で表現すると図 10 のようになる。

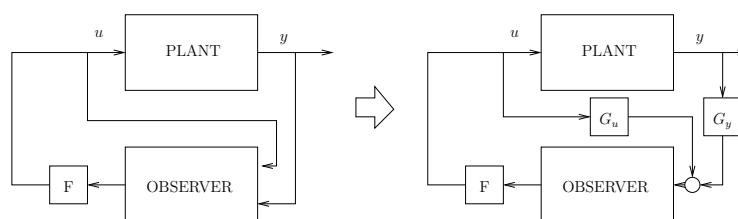


図 10: 不確定性を含むフィードバック系

そこで、オブザーバの状態方程式を次のように変形する。

$$\dot{\hat{x}} = (A+KC)\hat{x}+[B, -K] \left(\begin{bmatrix} u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ y \end{bmatrix} \right) = (A+KC)\hat{x}+[B, -K] \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} y \right) \quad (103)$$

ここで

$$G_u := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad G_y := \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

と定義すると図 10 右図のような表現となるなり、形式的な変換が可能になる。

39 極配置とLQR

状態が観測できる時、システムが完全可制御なら極配置によって閉ループ系の極を任意に配置することによってシステムの初期値応答を調整することができる。しかし、収束の速度を早くするように、閉ループ系の極を複素左半面の遠いところに配置すると、フィードバックゲインが大きくなる(いわゆるハイゲインになり、入力の振幅が非常に大きくなってしまふ。このような大きな入力は、ハードウェアの制約により実際には使えない場合が多い。また、多入力システムの場合、極の配置だけでは、フィードバックゲインが唯一に定まらないという問題もある。

これらの問題に対して、応答と入力のトレードオフを取りながら、フィードバックゲインを唯一に定める方法として、次の評価関数を最小化する、2次形式最適状態フィードバック制御(LQR: Linear Quadratic optimal Regulator)がある。

$$\begin{aligned} J &= \int_0^{\infty} x^T(t)Qx(t) + 2x^T(t)Su(t) + u^T(t)Ru(t)dt \\ &= \int_0^{\infty} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \end{aligned} \quad (104)$$

ただし、対象のシステムの状態空間表現は

$$\dot{x}(t) = Ax(t) + B_2u(t) \quad (105)$$

$$(106)$$

である。この評価関数に対して、 $Q = Q^T \geq 0, R = R^T > 0$ を仮定すると、行列：

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \quad (107)$$

は半正定行列となるので、

$$W := \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \quad (108)$$

は次のように分解することができる。

$$W := \begin{bmatrix} C_1^T \\ D_{12}^T \end{bmatrix} \begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \quad (109)$$

従って、上の評価関数はノルムを用いて

$$J = \|C_1 x + D_{12} u\|_2^2 \quad (110)$$

と表現することができる。ただし、 W の分解は、 W が半正定対称行列であるので、特異値分解を行うと、

$$W = U \Sigma U^T \quad (111)$$

と分解できるので、 Σ の各要素の平方根を取った行列を $\Sigma^{1/2}$ とすると、次のように分解することもできる。

$$W = U \Sigma^{1/2} \Sigma^{1/2} U^T \quad (112)$$

従って、 C_1, D_{12} は次のように求めることができる。

$$\begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \Sigma^{1/2} U^T \quad (113)$$

(この行列の平方根は MATLAB では `sqrtm()` によっても求めることができる。)

特に、 $S = 0$ の時は、

$$C_1 = \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix}, \quad D_{12} = \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix}$$

となる。

このような問題設定の下、標準の 2 次形式最適制御問題は次のように定式化される。

次のシステム：

$$\dot{x} = Ax + Bu, \quad x(0) = x_0 \quad (114)$$

$$z = C_1x + D_{12}u \quad (115)$$

ただし、仮定として次の条件を満たすものとする。

1. (A, B_2) は可安定
2. D_{12} は列フルランクで $[D_{12}D_{\perp}]$ はユニタリ行列
3. (C_1, A) は可検出
4. $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ は全ての ω に対してフル列ランクを持つ。

に対して、 $u \in L_2[0, \infty)$ で $\|z\|_2^2$ を最小化するものを求める。この問題に対する解は、次の定理によって与えられる。

[定理]

状態が観測できるとき、標準 2 次形式最適制御問題の解は次の状態フィードバック：

$$u(t) = Fx(t) \quad (116)$$

で与えられる。ただし、

$$F := -(B_2^T X + D_{12}^T C_1) \quad (117)$$

で、 X は次の Riccati 方程式の安定化解である。

$$(A - B_2 D_{12}^T C_1)^T X + X(A - B_2 D_{12}^T C_1) - X B_2 B_2^T X = -C_1^T D_{\perp} D_{\perp}^T C_1 \quad (118)$$

また、評価関数の最小値は次式で与えられる。

$$J^* = x_0^T X x_0 \quad (119)$$

この LQR と極配置については次のようなことが知られている。簡単のために 1 入力、1 出力のシステムを考えて、対象のシステムを

$$\dot{x} = Ax + bu \quad (120)$$

$$z = \begin{bmatrix} c \\ 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \sqrt{r} \end{bmatrix} u \quad (121)$$

とする。ただし、 (A, b, c) は可制御、可観測とする。今、

$$c(sI - A)^{-1}b = \frac{b(s)}{a(s)} \quad (122)$$

とし、閉ループ系の特性多項式を

$$\det(sI - A - bK) = \prod_{i=1}^n (s - z_i) \quad (123)$$

とする。この時、 z_i は次の多項式 $\Delta(s)$ の根にもなっている。

$$\Delta(s) = a(s)a(-s) + r^{-1}b(s)b(-s) \quad (124)$$

また、根の対称性より次のようにも表現できる。

$$\Delta(s) = a_K(s)a_K(-s) \quad (125)$$

この関係より次のことが解る。

[High Cost of Control ($r \rightarrow \infty$)] 評価関数の r が大きい場合、

$$a_K(s)a_K(-s) \rightarrow a(s)a(-s) \quad (126)$$

より次のことがわかる。

1. 安定な極は動かさない
2. 不安定な極は、虚軸対称の点に近づく

[Low Cost of Control ($r \rightarrow 0$)] 評価関数 r が小さい、つまり入力
が'Cheap' で大きな入力を許容する場合は、

$$a_K(s)a_K(-s) \rightarrow r^{-1}b(s)b(s) \quad (127)$$

となる。

1. 有限の極は $b(s)b(-s)$ に近づく
2. 残りの根は

$$(-1)^n s^{2n} + r^{-1}(-1)^m b_0^2 s^{2m} = 0 \quad (128)$$

ただし、

$$b(s) = b_0 s^m + \dots + b_m \quad (129)$$

とする。この $2(n-m)$ 個の解は半径 $(b_0^2/r)^{1/2(n-m)}$ の'Butterworth' 配置と呼ばれる特別なものとなっている。

40 LQR と H_2 ノルム

LQR 制御の最適解は

$$\dot{x} = (A + B_2 F)x, \quad x(0) = x_0 \quad (130)$$

$$z = (C_1 + D_{12} F)x \quad (131)$$

によって与えられる。 z の Laplace 変換は

$$Z(s) = C_F (sI - A_F)^{-1} x_0 \quad (132)$$

ただし、

$$A_F := A + BF, \quad C_F := C_1 + D_{12} F$$

となる。従って、 $\|z\|_2^2$ は Parseval の定理より

$$\|z\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace} C_F (j\omega I - A_F)^{-1} x_0 x_0^T (-j\omega I - A_F)^{-T} C_F^T d\omega$$

となる。ここで、 $x_0 x_0^T$ は半正定行列であるので、 B_1 を次の等号を満たす $n \times n$ の行列とすると、

$$B_1 B_1^T = x_0 x_0^T \quad (133)$$

$\|z\|_2^2$ は次のように表現できる

$$\|z\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace} C_F (j\omega I - A_F)^{-1} B_1 B_1^T (-j\omega I - A_F)^{-T} C_F^T d\omega$$

このノルムは次のシステム $G_c(s)$ の H_2 ノルムに等しい。

$$G_c(s) = \begin{bmatrix} A_F & B_1 \\ C_F & 0 \end{bmatrix} \quad (134)$$

この関係式は、任意の安定化状態フィードバックを用いた閉ループ系の H_2 ノルムを求めるためにも用いることができる。

また、一般化プラントを用いた LQR 問題のための表現は

$$\left[\begin{array}{c|c|c} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ \hline I & 0 & 0 \end{array} \right] \quad (135)$$

となる。(確かめてみよ)

41 標準 H_2 制御と簡単化 H_∞ 制御

標準 H_2 、簡単化 H_∞ 最適化問題は、一般化プラント表現されたシステム：

$$\left[\begin{array}{c|c|c} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ \hline C_2 & D_{21} & 0 \end{array} \right] \quad (136)$$

に対して、 w から z までの与えられたノルムを最小化する問題として定式化することができる。ただし、ここで注意しなければならない点は、 H_2 ノルム最適化問題が可解の場合は唯一解が存在するのに対して、 H_∞ 最適化問題では、解が唯一でないことである。

H_2 最適制御と前節の LQR 制御の違いは、LQR の場合は最小化したい評価関数に対して、最適な状態フィードバックゲインを求め、実際には状態が利用できない場合が多いので、これに対してオブザーバを構成するという 2 段構えでコントローラを設計する。この時、オブザーバの設計によって、結果として出来上がるコントローラは異なり、本来評価したい評価関数の H_2 ノルムの意味での最適性は一般に失われる。 H_2 最適制御の場合は、唯一な最適なコントローラが出力フィードバック系として一度に設計される点である。

システム行列表現されたプラント P に対して、 H_2 最適制御系を求めるコマンドは MATLAB では次のようになる。

```
[K,G,norms,kfi,gfi,hamx,hamy] =
    h2syn(P, nmeas, ncon, ricmethod);
```

ここで、 P は対象のシステムのシステム行列表現、 $nmeas$ は観測出力数、 $ncon$ は制御入力数、 $ricmethod$ は Riccati 方程式を解くための方法番号 (デフォルトは 2) である。また、 K はシステム行列されたコントローラで、 G は結果として得られる閉ループ系、観測条件が異なる場合の各ノルム、 FI 問題の場合の状態フィードバック、 FI 問

題の場合の閉ループ系、 $hamx$ はリカッチ方程式の解 X , $hamy$ はもう一方のリカッチ方程式の解である。詳しくは `help` を参照のこと。

42 H_∞ 制御制御とロバスト安定化

ここでは、 H_∞ 最適制御の応用として、ロバスト安定化問題を考える。特にここでは、下図に示されるような系を安定化する、加法的変動に対するロバスト安定化問題を考える。

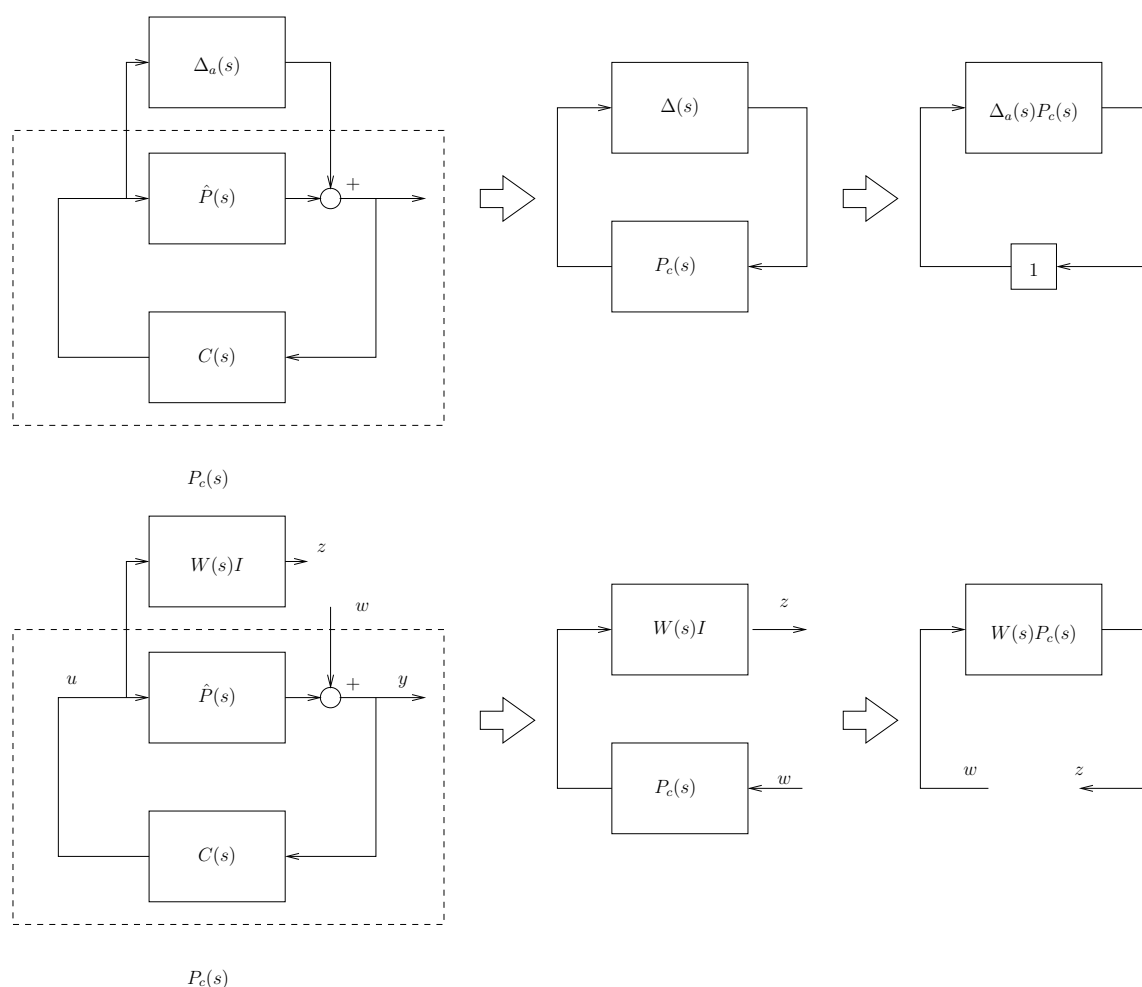


図 11: ロバスト安定化と周波数重みを含む一般化プラント表現

図 11 左上のような系を安定化する場合、上の行の変換の用に $\|\Delta_a(s)P_c(s)\|_\infty$ が 1 未満であれば、小ゲイン定理により安定性が保証される。そこ

で、この問題に H_∞ ノルム最小化問題を適用するため、図下段のように $\Delta_a(s)$ を安定な重み伝達関数 $W(s)$ に置き代える。ただし、 $W(s)$ は次のゲイン条件を満たす $\Delta(s)_a$ の上界関数 (bounding function) となっているものを選ぶ。

$$\bar{\sigma}(\Delta_a(j\omega)) \leq |W(j\omega)| \quad (137)$$

このシステムに対して、 w から z の H_∞ のノルムが 1 未満になるようなコントローラ C を設計すれば、 $\|W(s)P_c(s)\|_\infty < 1$ となり、もとの閉ループ系を安定化するコントローラ $C(s)$ が設計できる。この関係は次のように示される。

$\|W(s)P_c(s)\|_\infty < 1$ を満たす $C(s)$ が設計されたとすると、

$$\bar{\sigma}(W(s)(j\omega)P_c(j\omega)) < 1 \quad (138)$$

ところが、

$$|W(j\omega)|\bar{\sigma}(P_c(j\omega)) = \bar{\sigma}(W(s)(j\omega)P_c(j\omega)) \quad (139)$$

であるから、

$$\bar{\sigma}(\Delta_a(j\omega))\bar{\sigma}(P_c(j\omega)) \leq |W(j\omega)|\bar{\sigma}(P_c(j\omega)) \quad (140)$$

となり、結局

$$\bar{\sigma}(\Delta_a(j\omega)P_c(j\omega)) \leq \bar{\sigma}(\Delta_a(j\omega))\bar{\sigma}(P_c(j\omega)) \leq 1 \quad (141)$$

となって、 $\|\Delta_a(s)P_c(s)\|_\infty < 1$ となり安定性が保証される。

このように、 H_∞ ノルム最小化問題を制御系設計に応用するためには、考えている問題のどの部分からどの部分への H_∞ ノルム最小化問題にすればよいか、また、どのような重み関数を導入するかを考えることがキーになる。また、実際に問題を解くためには、解くべき一般化プラントの状態空間表現を得ることが必要になる。

上で考えた加法的変動に対する制御すべき一般化プラント表現は次のように求められる。今、 $\hat{P}(s), W(s)$ の伝達関数をそれぞれ

$$\begin{aligned} \hat{P}(s) &= C(sI - A)^{-1}B \\ W(s) &= C_w(sI - A_w)^{-1}B_w + D_w \end{aligned} \quad (142)$$

とすと、図 11 の左下図より、

$$\begin{aligned} z &= Wu \\ y &= Pu + w \end{aligned} \quad (143)$$

となる。 $\hat{P}(s), W(s)$ を状態空間実現した場合の状態を $x(t), x_w(t)$ とすると、上式の関係は

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{x}_w &= Ax_w + B_w u \\ z &= C_w x_w + D_w u \\ y &= Cx + w \end{aligned} \quad (144)$$

となる。つまり、

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} x \\ x_w \end{bmatrix} &= \begin{bmatrix} A & \\ & A_w \end{bmatrix} \begin{bmatrix} x \\ x_w \end{bmatrix} + \begin{bmatrix} 0 & B \\ 0 & B_w \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \\ \begin{bmatrix} w \\ y \end{bmatrix} &= \begin{bmatrix} & C_w \\ C & \end{bmatrix} \begin{bmatrix} x \\ x_w \end{bmatrix} + \begin{bmatrix} 0 & D_w \\ I & 0 \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \end{aligned} \quad (145)$$

となる。これより、一般化プラント表現は

$$\left[\begin{array}{c|c|c} A & 0 & B \\ & A_w & B_w \\ \hline & C_w & D_w \\ \hline C & I & 0 \end{array} \right] \quad (146)$$

となる。

このように最適化問題を実際の応用に適用しようとした場合、実際の信号に適切な周波数重み (考えている信号に対してフィルタを通したような信号を考えること) が必要となる、このような信号は元のシステムとフィルタを直列したシステムを考えることにより得られ、MATLAB では `series(sys1, sys2)` というコマンドを用いることが出来る。(課題 9 参照)

H_∞ ノルム最小化問題を制御系設計に応用するためには、重み関数

の導入と結果の一般化プラント表現を得ることに熟練する必要がある。(逆に、それだけで良い。)

H_∞ 最適制御系を求めるコマンドはMATLABでは次のようになる。

```
[K,G,gfin,ax,ay,hamx,hamy] =  
hinfsyn(P, nmeas, ncon, gmin,gmax,tol,ricmethod,epr,
```

ここで、 P は対象のシステムのシステム行列表現、 $nmeas$ は観測出力数、 $ncon$ は制御入力数、 $gmin$ は γ の下限、 $gmax$ は γ の上限、 tol は最終の γ と 1 つ前の γ の違いの許容範囲、 $ricmethod$ は Riccati 方程式を解くための方法の番号 (デフォルトは 2) である番号 (デフォルトは 2)、 epr は Hamiltonian の固有値の実部の零判定、 epp は X_∞, Y_∞ 解は正定であるかの判定である。また、 K はシステム行列表現されたコントローラで、 G は結果として得られる閉ループ系、 $gfin$ は達成された最終 γ 値、 ax, ay は最終的に得られた X_∞, Y_∞ の行列、 $hamx, hamy$ は計算途中の γ に対する X_∞, Y_∞ 行列を格納した行列である。詳しくは `help` を参照のこと。

[課題 10]

1. 課題 8 の対象のシステムに対する二次形式評価関数 :

$$J = \int_0^{\infty} x^T(t)Qx(t) + u^T(t)Ru(t)dt$$

ただし、

$$Q := \begin{bmatrix} 10 & & & \\ & 10 & & \\ & & 10 & \\ & & & 10 \end{bmatrix}, \quad R := [1]$$

を考え、課題 10 の B_1 から次の z までの H_2 ノルムを最小化するコントローラを求めなさい。ここで、 H_2 ノルム最適化問題にするためには、 z として次のものを定義する。

$$z := \begin{bmatrix} Q^{1/2}x \\ R^{1/2}u \end{bmatrix}$$

ただし、観測値は

$$y = [1, 1, 0, 0]x$$

とし、状態の全ては観測できないものとする。また、このときの H_2 ノルムと課題 4 で用いたオブザーバ併合系との H_2 ノルムを比較しなさい。

2. 一次遅れ系で表されるリニアサーボモータを用いて、1 自由度振動系を制御することを考える。ただし、サーボモータの入力電圧から出力の力までの伝達関数は

$$H_e(s) = \frac{1}{T_e s + 1}$$

と表され、振動系の力から出力の変位までの伝達関数は

$$H_m(s) = \frac{1}{ms^2 + \mu s + k}$$

とする。ただし、それぞれのパラメータは次の表に与えられているものとする。(制御対象はこれらの伝達関数が単純に直列に結合されたものとする)

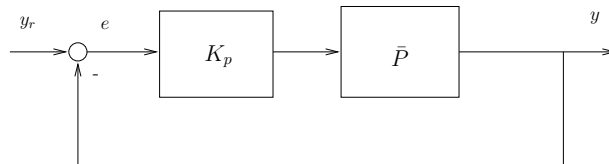
記号	値
T_e	0.1
m	1.0
μ	4.0
k	10.0

次の設問に答えなさい。

- (a) 電流から変位までの伝達関数を、サーボモータの遅れ特性を無視して、ノミナルプラントの伝達関数 \bar{P} を

$$\bar{P}(s) = \frac{1}{ms^2 + \mu s + k}$$

とする。このノミナルなシステムに対して、次の図のようなP(比例)制御を考え、定常偏差が1/10以下となるようにP制御のゲイン K_p を定めなさい。



- (b) 前の問題の制御系を実際のシステム(サーボモータの特性も入れた系)に適用した場合の応答を求めなさい。
- (c) 無視したサーボモータの特性を不確定性と考え、それを乗法的変動と考えたときに乗法的変動の伝達関数を求めよ。
- (d) 前の問題で求めた乗法的変動の伝達関数を不確定性の上限関数と考え、ロバスト安定性を保証した制御系を設計したい。ただし、定常偏差は1/10以下となるように設計する。(出来るだけ高い周波数まで1/10のとしたい) つぎのステップに従ってコントローラを設計しなさい。

- i. 不確定性を持つ制御対象を図 12 のように偏差 e を入力として制御入力を出力するコントローラ C を設計する。このために、その図の下のような一般化プラントを考える。 $S(s) = 1/(1 + CP), T(s) = CP/(1 + CP)$ とした時、 w から z_1, z_2 それぞれの伝達関数を求めよ。この結果から何が分かるか説明しなさい。

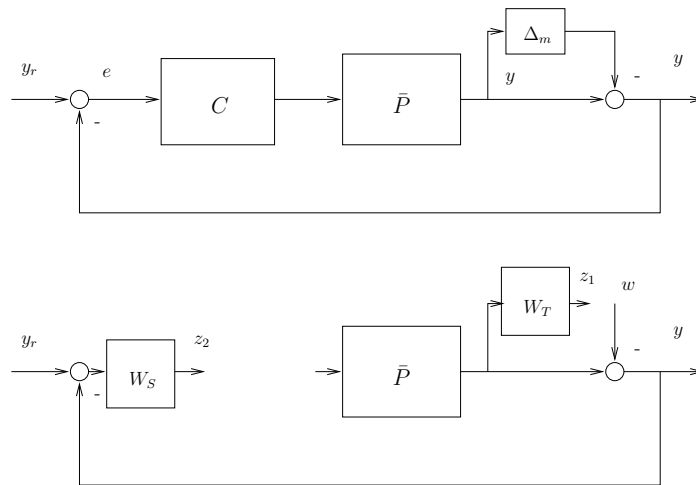


図 12: ロバスト安定化と周波数重みを含む一般化プラント表現

- ii. ロバスト安定性を保証する不確定性の上限関数 $W_T(s)$ をここでは簡単のため、 $W_T(s) = \Delta_m(s)$ とし、追従性能を保証するため $W_S(s) = \frac{K_S}{T_S s + 1}$ ($K_S > 10$) とする。 w から $z = [z_1, z_2]^T$ までの H_∞ ノルムが 1 未満となるコントローラ $C(s)$ を設計せよ。注意! このままの一般化プラントは H_∞ 標準問題の仮定を満たさない。よって、3 番目の評価出力 z_3 を $z_3 = \epsilon u$ とせよ。
- iii. いくつかの T_s に対して、得られたコントローラを用いてステップ目標値に対する応答をプロットせよ。
- iv. w から y, e までのボード線図をプロットして、それぞれのゲインが $|W_T^{-1}(j\omega)|, |W_S^{-1}(j\omega)|$ 以下となっていることを確

かめよ。

[Ex.10]

1. Let consider a quadratic criterion function below for a system in Ex. 8:

$$J = \int_0^{\infty} x^T(t)Qx(t) + u^T(t)Ru(t)dt$$

where

$$Q := \begin{bmatrix} 10 & & & \\ & 10 & & \\ & & 10 & \\ & & & 10 \end{bmatrix}, \quad R := [1].$$

Then calculate an H_2 optimal controller when an external signal w is injected through B_1 and the evaluated signal z is defined as

$$z := \begin{bmatrix} Q^{1/2}x \\ R^{1/2}u \end{bmatrix}$$

In this case we assume that we can only observe an output signal y defined as

$$y = [1, 1, 0, 0]x.$$

Compare the obtained H_2 norm with that obtained in Ex.4 which consisted of an optimal state feedback and an observer.

2. A mechanical system of one degree of freedom is controlled by a linear servo motor where the transfer function of the amplifier from voltage to force is given by

$$H_e(s) = \frac{1}{T_e s + 1},$$

and the transfer function from the force to the position is given by

$$H_m(s) = \frac{1}{ms^2 + \mu s + k}.$$

The parameters in the transfer functions are given in the following table. (The controlled plant can be expressed as a transfer function which is a serial connection of both transfer functions.)

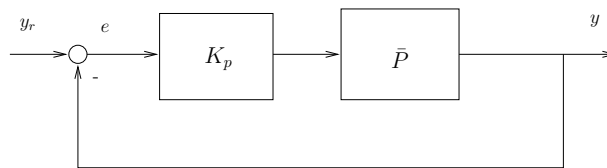
notation	value
T_e	0.1
m	1.0
μ	4.0
k	10.0

Answer the following questions.

- (a) By ignoring the dynamics of the servo motor, let us assume that the transfer function of a nominal plant, \bar{P} , can be expressed as

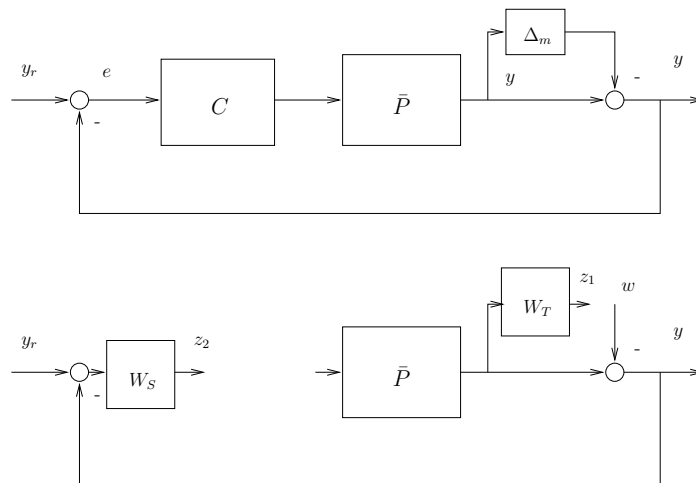
$$\bar{P}(s) = \frac{1}{ms^2 + \mu s + k}.$$

Determine a proportional feedback gain, K_p , in the following figure so that the steady state error becomes 1/10 to the reference signal.



- (b) Show a response of the true closed loop system including the dynamics of the servo motor when the proportional feedback is applied for.

- (c) Let us assume that the ignored dynamics of the servo motor as an uncertainty and determine the multiplicative uncertainty Δ_m according to the uncertainty.
- (d) An upper bound function for the uncertainty in the previous question is given by the multiplicative uncertainty itself, and robust stable controlled system should be designed where the steady state error is suppressed less than or equal to 1/10 to the reference signal. (Error should be suppressed less than 1/10 to as high frequency as possible.) Design a controller in the following steps.
- i. Controller C is design for the plant with the uncertainty in Fig.13so that the input to the controller is error signal and the output is the input to the plant. For this design a generalized plant is considered for the block diagram below in the figure. If we set $S(s) = 1/(1 + CP)$, $T(s) = CP/(1+CP)$, compute transfer functions from w to z_1, z_2 . Explain what we can see from the relationship.



⊗ 13: Robust stability and generalized plant with frequency weights.

- ii. Let us assume that the bounding function for the uncertainty, $W_T(s)$ is given as $W_T(s) = \Delta_m(s)$, and the weight function for the tracking performance is given as $W_S(s) = \frac{K_S}{T_S s + 1}$ ($K_S > 10$). Design a controller so that H_∞ norm $z = [z_1, z_2]^T$ is less than 1. **Note! Since this generalized plant does not satisfy assumptions for standard H_∞ control problems, define a third signal z_3 as $z_3 = \epsilon u$.**
- iii. Show the step response of the designed closed loop system for several T_s .
- iv. Plot bode diagrams from w to y , e and show that the gain plots are lower than $|W_T^{-1}(j\omega)|$, $|W_S^{-1}(j\omega)|$, respectively.