

## 制御工学 講義内容

制御工学 で学んだ連続系の伝達関数によるフィードバック制御を基礎とする。動的システムを状態変数による連立微分方程式（状態方程式）で表現する方法を学び、その動的システムの解析を行う。また、状態方程式と伝達関数の関係も学ぶ。次に状態空間表現に基づく各種設計法（極配置，オブザーバ，最適レギュレータ）を習得する。数値計算用ソフトウェア MATLAB(Control System Tool Box)/Simulink を活用し，演習問題を解くことで状態空間手法の一層の理解を図る。

### 0．ガイダンス

#### 1．線形性、運動方程式の線形化

#### 2．運動方程式の状態表現 > 状態方程式

##### 2．1 状態空間法の理解に必要な行列、行列式を復習する。

MATLAB による演算

##### 2．2 状態方程式、状態方程式と伝達関数の関係

MATLAB による演算

#### 3．可制御性（Matlab による演算確認）

#### 4．可観測性

#### 5．状態フィードバック制御

##### 5．1 極配置の方法による状態フィードバック制御則

MATLAB による設計とシミュレーション

##### 5．2 2次形式評価関数を最小にするように状態フィードバック制御則

MATLAB による設計とシミュレーション

#### 6．状態推定（観測）法

Matlab によるシステム的设计

## 0 . ガイダンス

制御工学 の授業で、なにを学んでいこうか？

制御工学 では、制御システムの基本構成、制御システムに関わる幾つかの概念：安定性、定常特性、応答特性に触れました。また、制御手法として、PID 制御則について説明しました。こんな沢山のことを勉強して、まだ足りないかと疑問に感じますが、一番重要なところ、制御則の設計法については、まだ不十分である。言い換えれば、PID 制御法ですべての問題は解決できないので、制御についての研究はさらに進められ、その続きをこの制御工学 で学びます。

### PID 制御

一つの信号をフィードバック  
試行錯誤  
1 入力 1 出力に適用

### 状態フィードバック制御

沢山の信号をフィードバック  
論理的  
多入力多出力適用

制御工学 で学ぶ制御の設計法は、状態フィードバック制御という制御則の決め方である。PID との根本的な違いは、前者はフィードバックする信号を一つに対して、後者は沢山の信号をフィードバックする。また、前者は試行錯誤的に制御パラメータを決めるのに対して、後者はある程度理論的な方法でフィードバックパラメータが決められる。さらに、前者が 1 入力 1 出力のシステムに対して、後者は多入力多出力系にも対応できます。

この授業では、なるべく自分達で制御設計を行ったり、システムの応答をシミュレーションするように、教育用計算機にある Matlab というソフトウェアを使った演習も授業の一環として行います。すでに実験が始まっています。実験室によっては、Matlab を用いて制御実験を行う場合もあります。そのとき、注意して学んでください。

MATLAB はその名前の由来が **MAT**rix **LAB**oratory(行列実験室) であることからわかるように、行列操作を得意とするプログラミング言語です。したがって、MATLAB を理解することは、制御系の解析や設計を行うための一つの有効なツールを手に入れるという意味で重要なことであると思います。

MATLAB はプログラミング言語の一つです。プログラミング言語は、**インタプリタ**(逐次命令を解釈しながら実行させるタイプの言語)とコンパイラ(C 言語)に大別しますが、MATLAB は前者に属します。

## Matlab による制御工学の演習

### 1 回目： Matlab の基本を知る

#### 1、 電卓として

```
>> 2+3
>> 2+4;
>> a=1; b=2; (変数 a と b に数値を代入する)
>> c=a+b ;
>> d=c/a
>> y=3^2 ; (y = 32)
```

#### 2. ベクトルと行列の扱い

$A = \begin{bmatrix} 1 & 2 & 6 \\ 2 & 0 & 5 \\ 4 & 7 & 8 \end{bmatrix}$ 、  $B = \begin{Bmatrix} 1 \\ 1 \\ 2 \end{Bmatrix}$  の入力

```
>> A=[1 2 6; 2 0 5; 4 7 8] ; (「 ; 」は改行マーク )
>> B=[ 1; 1; 2];
>> C=A*B ;
>> A1=inv(A); (A1 = A-1)
>> eig(A) ; (A の固有値を求める )
>> A' (行列 A の転置)
```

#### 3 . 関数の計算とグラフ表現

```
>> t = 0 : 0.5 : 360 ; (ベクトル t = [0 0.5 1 ... 360])
>> pi=3.1416;
>> y=sin(pi/180*t); ( t の各要素に、 sin 関数を計算する )
>> plot(t , y) (関数 plot による 2 次元グラフ表現 )
>> xlabel('degree');
>> ylabel('y=sin()');
```

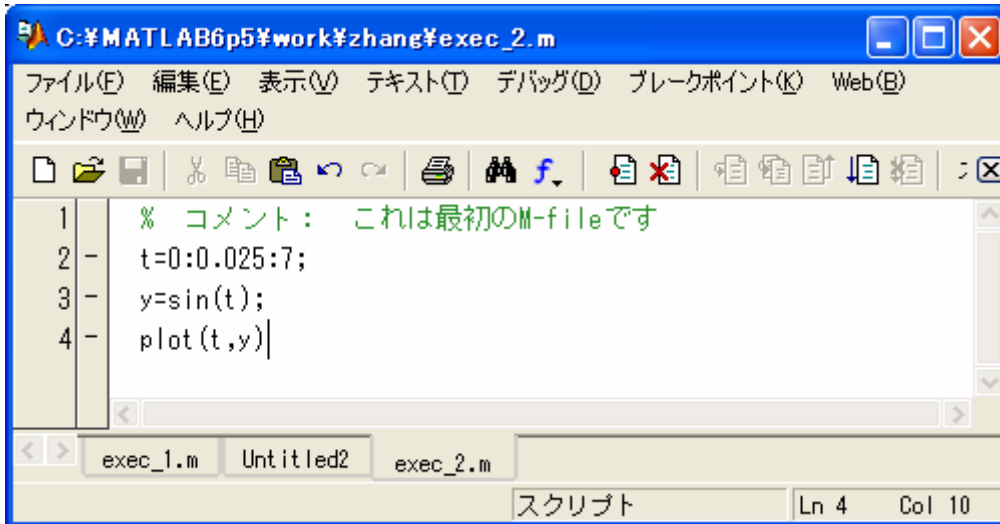
#### 4 . コマンドウエンドウでよく使う関数

```
>> who (作業領域のデータを確認)
>> ls (作業ディレクトリにあるファイルを表示)
>> clear (作業領域のすべてのデータをクリアする)
>> clear A (作業領域のデータ A をクリアする)
>> help
```

>>demo (例題を見る)

## 5. エディタによる M ファイルの作成 (コマンドの編集) と実行

Matlab ウィンドウの左上をクリックし、ファイル編集するウィンドウを開く  
( >> edit )



このファイルを example\_1.m で保存

このとき、example\_1.m というM - ファイルが出来る。

>> example\_1 (ファイルの実行)