

機械システム工学科 選択A群

# 機械力学 講義資料



## 目次

第 1 講	ガイダンス	2	第 9 講	伝達関数	28
1.1	現代の機械 = メカ +	2	第 10 講	伝達関数の使い方	30
1.2	Dynamics の設計?	2	10.1	定常応答の復習	30
1.3	この講義について	2	10.2	伝達関数の作り方	30
1.4	振動現象の実例	2	10.3	伝達関数の使い方	32
1.5	学習方法	3	第 3 回	レポート課題	33
1.6	受講前日までの準備	3	第 11 講	解析力学 (ラグランジュ形式の力学)	34
第 2 講	数値解析ソフト入門	4	第 12 講	倒立ロボットの力学	36
2.1	コンピュータを起動する	4	12.1	数式処理ソフト入門	36
2.2	コマンドを実行する	4	12.2	解析力学 (ラグランジュ形式の力学)	37
2.3	ファイルを作り編集する	5	12.3	倒立ロボットの運動方程式	40
2.4	Octave を使う	5	第 13 講	ロボット制御への応用 II	41
2.5	Octave プログラムを実行する	7	13.1	倒立ロボット	41
2.6	グラフ (表示画面) を印刷する	7	13.2	制御を受けるロボットの挙動	41
第 1 回	レポート課題	7	13.3	倒立ロボットの位置制御	43
第 3 講	自由振動のモデル	8	期末レポート課題	43	
3.1	ニュートンの運動法則	8	第 14 講	まとめ	44
3.2	自由振動のモデル	9			
3.3	自由振動系の実現	10			
3.4	数値解法	10			
第 4 講	振動の固有値	12			
4.1	振動の固有値 (特性値)	12			
4.2	固有値の求め方	12			
4.3	固有値と振動のパターン	12			
4.4	固有値による自由振動の予測	14			
第 5 講	減衰比と固有振動数	15			
5.1	無次元化	15			
5.2	減衰比 $\zeta$ の使い道	16			
5.3	固有振動数 $\omega_n$ の使い道	17			
第 6 講	ロボット制御への応用 I	19			
6.1	倒立ロボットの力学モデル	19			
6.2	制御方式と固有値の関係	19			
6.3	制御系の設計 — ゲイン調整	20			
第 2 回	レポート課題	20			
第 7 講	共振現象のモデル	21			
7.1	強制振動のモデル	21			
7.2	共振現象	21			
7.3	過渡応答と定常応答	23			
第 8 講	周波数応答	25			

# 第 1 講

## ガイダンス

あえて専門外の立場から、機械力学を位置付けておこう。

### 1.1 現代の機械 = メカ +

機械製品の設計というと、製図で学ぶようなメカ(構造や機構)の設計だけを連想しがちだが、現実には、メカだけからなる機械製品は稀である。

例えば、分解すれば分るが、コンピュータのマウスは厳然たるメカ部を有している。しかし、当然、メカだけでマウスの機能は実現しない。マウスからは、数本の信号線が出ており、その電氣的な機能を実現しなければマウスは成立しない。したがって、

マウスの設計 = メカ設計 + 回路設計

ということが分かる。

別の例として、歩行ロボットを挙げよう。むろんメカだけでは歩けない。もちろん制御用の回路設計も必要だが、それだけでは済まない。

実習 1.1 空欄 A を埋めよ。理由を考察せよ。

ロボットの設計 = メカ設計 + 回路設計 + A

### 1.2 Dynamics の設計？

ロボットを歩かせるには、歩き方を教えなければならない。歩き方とは、すなわち動き方である。したがって、ロボット開発者は、メカ、回路のほかに、動き方を設計しなければならない。

現代の工学では、ロボットの動き方を、ロボットの姿勢の時間変動 (dynamics) としてとらえる<sup>1)</sup>。関節角度を一定に保たれたロボットは動けない。ロボットが動き出すためには、関節角度に適切な時間変動を与えてやる必要がある。ようするに、

ロボットの設計 = メカ + 回路 + Dynamics の設計となる。当然、Dynamics の設計 (=動き方の設計) が下手だと、ロボットは歩けない。

機械力学では、機械の Dynamics を設計するための手法を学ぶ。これから作る機械の Dynamics を予見し、自在に設計できれば目標達成である。

<sup>1)</sup> 正確には、幾何学で解ける動き方を運動学 (kinematics)、力学法則まで加味しないと解けない動き方を動力学 (dynamics) という。

### 1.3 この講義について

実際の設計現場を知るためのヒントとして、市販の設計支援ソフトウェアの機能 (CAE) に着目すると、どんなソフトでも、およそ、

機能	対応する分野
熱・流体解析	熱工学, 流体工学 …
変形解析	材料力学, 材料加工 …
運動・振動解析	機械力学, 制御工学 …

のような機能を搭載している。これらはどれも、実際の設計者が求める機能である。

機械力学は、制御工学と並んで、動き方の設計に必要な運動解析と、振動解析の手段を与える。これに対応して、大学の講義科目としての機械力学も、

- 広く機械の運動を扱うもの (運動力学)
- そのなかの振動のみを扱うもの (振動論)

の 2 種類に大別される。多数派は後者であり、本講義でもこれに準じて振動論を扱う。

振動論を重視する理由を列挙しておく。

- 機械は必ず振動する (復元力の存在)
  - 車両, 飛翔体, ロボット, 工作機械, etc.
- Dynamics の典型的な算法が身につく。
  - 固有値, 周波数応答, 伝達関数, 解析力学, etc.
- 制御理論は応用振動論。算法は同じ。
  - 振動論でロボットは立つ! 歩く!

以上をまとめると、可動部を有する機械なら、どんな機械でも、可動部の時間変動を適切に設計する必要に迫られる。そのための基礎知識として、本講義では振動論を学ぶ。

### 1.4 振動現象の実例

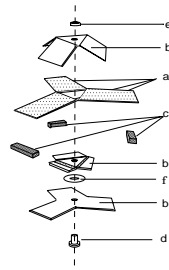
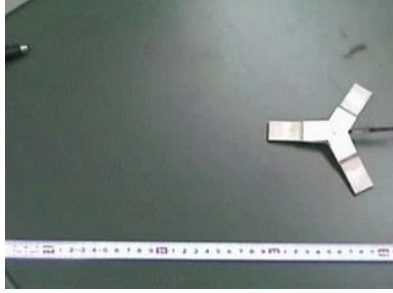
ほんの一例だが、振動現象の実例を紹介しておく。

#### 1.4.1 無電源太陽追尾センサ (吉田研・卒研)



- ずれを、いかに早く収束させるか?
- 収束性を表わす数値とは?  $\implies$  減衰比

#### □ 1.4.2 振動型マイクロ移動ロボット (吉田研・修論)



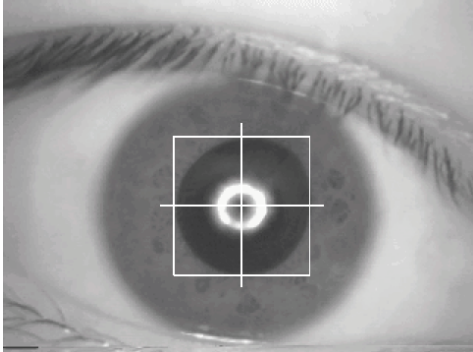
- 入力周波数を変えると，進行方向が変わる．
- その理由は？ ⇒ 共振現象

#### □ 1.4.3 二足歩行ロボット (HONDA ASIMO)



- 倒れまいとすれば，必ず振動する．
- 振動は除去不可能．都合よく振動させるには？  
⇒ ロボット制御工学  
ちなみに自転車も必ず蛇行する．

#### □ 1.4.4 眼反射運動 (吉田研・修論，特許)



- 眼球は，不随意にねじれ運動する．
- ねじれ振動を観察すると，平衡機能の異常が診断できる． ⇒ バイオメカニクス

### 1.5 学習方法

次の2つの作業を，実体験するのが早道である．

- Dynamics(動き方)を，自分で手計算してみる．
- 同じく，自分でコンピュータ計算してみる．

<sup>2)</sup>例えば，他人が書いたプログラムなど繁雑で読めない！

<sup>3)</sup><http://cms1.cc.utsunomiya-u.ac.jp/eng/>

<sup>4)</sup>この授業の登録キーは，7-406

手計算には板書を囲んでわいわいコミュニケーションできる利点があり，学習や検算に適している．反面，人間の手計算では，どんなに長くてもレポート用紙2~3枚が限度だから，実際の機械製品を手計算することは稀である．

コンピュータ計算では，製品レベルの複雑な問題を簡単に解けるが，コミュニケーションには不向きなので，検算に苦労することが多い<sup>2)</sup>．

### 1.6 受講前日までの準備

- 情報処理センターの教育用端末にログインする．
- 授業支援サイト「工学部 Moodle」にアクセスし，機械システム工学科の「機械力学」に受講登録する．聴講届とは連動しない．
- 次回テキストをダウンロード，印刷，予習する．

#### 工学部 Moodle について

- 1) 工学部 Moodle<sup>3)</sup> へは，Firefox, Internet Explorer などのブラウザでアクセスする．ただし，学内からしかアクセスできない．ログインにはセンターのIDとパスワードを使う．
- 2) この授業のコースを見つけてクリックして，登録キー<sup>4)</sup>を入力する．
- 3) プロフィールの入力を求められるので，
  - 「名」に本名．例：吉田勝俊
  - 「姓」に半角学籍番号．例：072100Aおよび「メールアドレス」を正しく入力する．
- 4) この授業の登録キー「7-406」は変更することができるので，注意すること．

## 第 2 講

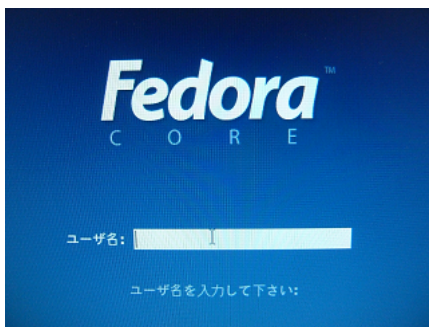
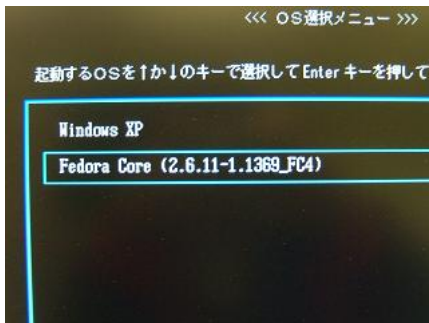
# 数値解析ソフト入門

Dynamics の設計者がよく用いる科学技術ソフトに入門する。高機能版の関数電卓と思えば大したことはない。お勉強というより実務なので、気楽に楽しんでほしい。

Fedora Core という UNIX 互換環境で作業を進めるが、以下の作業に失敗する人は、情報処理センターの管理室に症状を申し出て、システム管理上の不具合があれば解消してもらおうこと。

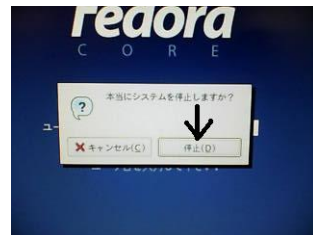
### 2.1 コンピュータを起動する

まず、PC の電源を投入し<sup>1)</sup>、OS 選択メニューで Fedora Core を選び、ユーザ名(Enter)、パスワード(Enter)の順に入力する。



停止するときには、左上のメニューからログアウトした後、マウス操作で停止させ、ディスプレイを切る。

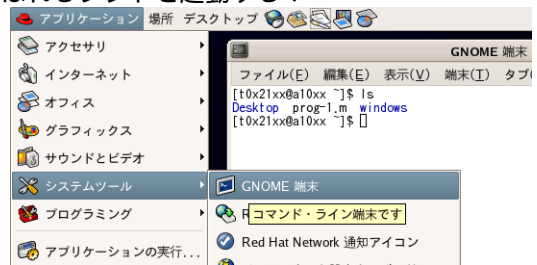
<sup>1)</sup>電源が入らないときは、PCの背面にある黒いマスタースイッチを再投入するとよい。



### 2.2 コマンドを実行する

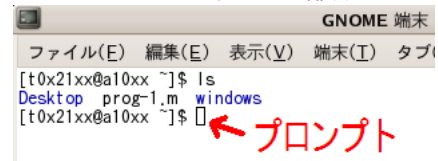
コンピュータが人間の役に立つためには、  
1) 人間が、コンピュータに何らかの指示を与える。  
2) その結果を、コンピュータが人間に返答する。  
という往復のキャッチボールが必要だ。これをマウス操作で行う方式を GUI(Graphical User Interface)、キーボード操作で行う方式を CUI(Character User Interface) という。

端末の起動 CUI を利用するには、まず「端末」と呼ばれるソフトを起動する。



コマンドの実行 次の操作の繰り返しである。

1) 端末上のプロンプトと呼ばれる部分



にコマンドを書いて、(Enter)キーを押す。

表 2.1: 端末上の基本コマンド

exit	利用を終了する .
ls	フォルダにあるファイル一覧を表示する .
cd	フォルダを移動する . cd           自分のフォルダに戻る . cd ..       1つ上のフォルダに移動する . cd abc      abc という名前のフォルダに移動する .
rm	ファイルを削除する . rm abc      abc という名前のファイルを削除する .
cp	ファイルをコピーする . cp abc def   abc というファイルを def というファイルにコピーする .
gedit	ファイルを編集する . gedit abc    abc という名前のファイルを新規作成または修正する .
octave	数値解析ソフトを起動する .

2) コンピュータの返答を読む .

具体例で示す . 例えば, 端末のプロンプト

```
[t0x2lxx@a10xx ~]$
```

に続けて, (1)(s)(Enter) とキーボードを打つと,

```
[t0x2lxx@a10xx ~]$ ls            指示
Desktop windows                 返答
```

という出力を得る . ちなみに, ls というコマンドは, 現在のフォルダにあるファイル一覧を表示するコマンドで, その返答が得たのが上の実行例である .

代表的なコマンドを, 表 2.1 にまとめておく .

## 2.3 ファイルを作り編集する

**実習 2.1** 表 2.1 を見ながら, 次の操作を実行せよ .

以下, プロンプトを略して [... ~]\$ と書く .

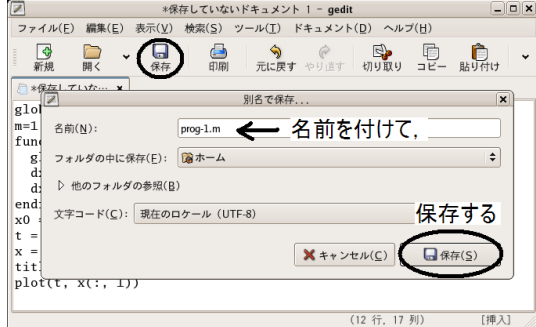
1) フォルダの内容を確認しておく .

```
[... ~]$ ls                    フォルダの内容
Desktop windows               システム用のフォルダ
```

2) ファイル abc を作成するには,

```
[... ~]$ gedit abc    ファイル (abc) の作成
```

とする . 試しに何か書き込んで, 保存しよう .



3) 再度フォルダの内容を確認すると,

```
[... ~]$ ls
Desktop/ abc windows/
```

となり, 確かに abc が出来てる .

4) 既存のファイルを開覧したり, 修正するには,

```
[... ~]$ gedit abc    ファイルの閲覧・修正
```

のように, gedit でもう一度開けばよい .

## 2.4 Octave を使う

数値計算は, Octave というフリーソフト<sup>2)</sup>で行なう .

**実習 2.2** 次の実行例を参考に, 端末上で Octave の起動と終了を何度か繰り返せ .

### Octave の起動と終了

```
[... ~]$ octave    Octave 環境に入る
GNU Octave, version 2.1.73 (i686-pc-cygwi
(中略)
```

<http://www.octave.org/bugs.html> to learn

```
octave:1>    Octave のプロンプト . 指示待ち状態 .
```

Octave のプロンプトに続けて, 例えば (1)(+)(2)

(Enter) とタイプする .

```
octave:1> 1+2
ans = 3                        Octave が出した答え
octave:2> exit                Octave から出る
[... ~]$                        端末に戻った
```

**実習 2.3** 以下の実行例を, そのまま実行せよ .

<sup>2)</sup> 契約条項を遵守すれば無償で使える . 市販の Matlab と互換性を持つ . <http://www.octave.org/>

## □ 2.4.1 四則演算・べき乗

```
octave:1> 1+2-3*4/5      四則演算
ans = 0.60000
octave:2> 2^3           べき乗
ans = 8                 2の3乗
```

## □ 2.4.2 ベクトル・行列

```
octave:1> v=[1,2,3]      横ベクトル
v =
  1  2  3
octave:2> x=[1;2;3]      縦ベクトル
x =
  1
  2
  3
octave:3> x(2)           ベクトルの第2成分
ans = 2
octave:1> A=[-1,2,3; 4,5,6; 7,8,9]  行列
A =
  -1  2  3
   4  5  6
   7  8  9
octave:4> A(2,1)         行列の2行1列成分
ans = 4
octave:5> A(2,:)         2行目の行ベクトル
ans =
  4  5  6
octave:6> A(:,1)         1列目の列ベクトル
ans =
  -1
   4
   7
octave:7> A'             行列の転置
ans =
  -1  4  7
   2  5  8
   3  6  9
```

## □ 2.4.3 線形代数

```
octave:1> A*x           行列 x 縦ベクトル
ans =
  12
  32
  50
octave:2> inv(A)        逆行列
ans =
  -0.50000  1.00000  -0.50000
   1.00000  -5.00000  3.00000
  -0.50000  3.66667  -2.16667
octave:3> eig(A)        行列の固有値
ans =
  15.91419
  -2.77850
  -0.13569
octave:4> x' * x        横 x 縦ベクトル = 内積
ans = 14
octave:5> [1,2,3].*[4,5,6] 成分どうしの積
ans =
  4  10  18
```

## □ 2.4.4 複素数・n次方程式

```
octave:1> i             虚数単位  $i = \sqrt{-1}$ 
```

```
i = 0 + 1i
octave:1> s=3+4i        複素数
s = 3 + 4i
octave:2> real(s)       実部
ans = 3
octave:3> imag(s)       虚部
ans = 4
octave:4> abs(s)        絶対値
ans = 5
octave:5> arg(s)        偏角
ans = 0.92730
```

2次方程式  $s^2 + 2s + 5 = 0$ , 3次方程式  $s^3 + 2s^2 + 5s + 7 = 0$  を解いてみる. Octave では, 多項式を係数ベクトル  $[1, 2, 5]$ ,  $[1, 2, 5, 7]$  で表現する.

```
octave:5> roots([1,2,5])
ans =
  -1.0000 + 2.0000i
  -1.0000 - 2.0000i
octave:6> roots([1,2,5,7])
ans =
  -0.19809 + 2.07975i
  -0.19809 - 2.07975i
  -1.60382 + 0.00000i
```

4次以上も同様に解けるが, 次数を増やすと計算時間が増大し, 精度が落ちる.

## □ 2.4.5 関数のグラフ

```
octave:1> x=[0:0.3:1]   等差数列のベクトル
x =
  0.00000  0.30000  0.60000  0.90000
octave:2> sin(x)
ans =
  0.00000  0.29552  0.56464  0.78333
octave:3> [sin(0),sin(.3),sin(.6),sin(.9)]
ans =
  0.00000  0.29552  0.56464  0.78333
octave:4> x=[0:0.1:8];  行末の; は出力抑制
octave:5> plot(x,sin(x))  2次元プロット
octave:6> plot(x,exp(x))  指数関数も
octave:7> plot(x,exp(-x).*sin(2*x))
```

最後の .\* は成分どうしの積であり,  $\exp(-x) \cdot \sin(2x)$   
=  $[\exp(0) \cdot \sin(0), \exp(-0.1) \cdot \sin(0.2), \dots]$  を意味する.

## □ 2.4.6 ユーザー関数の定義と活用

```
octave:1> function y=f(x)  1変数関数の定義
> y=x + cos(x);           ; を忘れるな!
> endfunction             ここまで
octave:2> plot([0:0.1:5],f([0:0.1:5]));
octave:3> function y=g(m,c,k)  3変数関数
> y=roots([m,c,k]);
> endfunction             ここまで
octave:4> g(1,0.2,2)
ans =
  -0.1000 + 1.4107i
  -0.1000 - 1.4107i
```

2行目の ; を忘れると

```
-- less -- (f)orward. (b)ack, (q)uit
となってしまうが, 端末上で q を打つと復帰する.
```



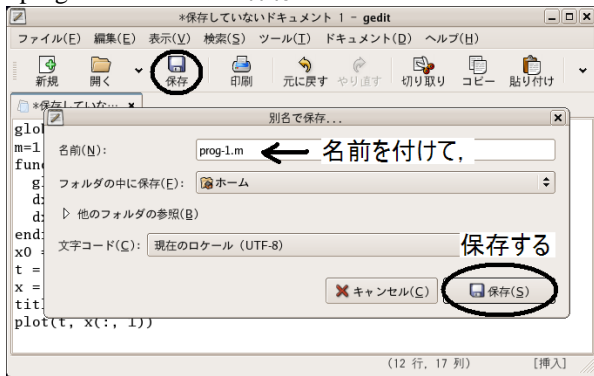
## 2.5 Octave プログラムを実行する

プログラム・ファイルの作成 試しに、第3構で使うプログラム・ファイル「prog-1.m」を作る。そのために、起動した gedit に次の内容を打ち込み、

プログラム「prog-1.m」

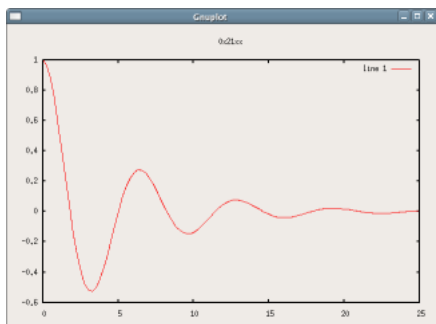
```
global m c k; # この後は無視される
m=1.0; c=0.4; k=1.0;
function dx = f(x, t)
    global m c k;
    dx(1) = x(2);
    dx(2) = -(c/m)*x(2) - (k/m)*x(1);
endfunction
x0 = [1; 0];
t = linspace(0, 25, 100);
x = lsode("f", x0, t);
title("0x21xx"); #学籍番号
plot(t, x(:, 1))
```

0x21xx のところを自分の学籍番号に換えて、名前「prog-1.m」をつけて保存する。



### プログラム・ファイルの実行

```
[... ~]$ ls          プログラムファイルの所在を確認
Desktop/ prog-1.m windows/ 確かにある
[... ~]$ octave          Octave 起動
GNU Octave, ... (略)
octave:1> source "prog-1.m" 実行
```



```
octave:2> exit          Octave 終了
[... ~]$                端末に戻った。
```

## 2.6 グラフ (表示画面) を印刷する

(a) グラフをクリックして「アクティブ」にする。

アクティブな Window



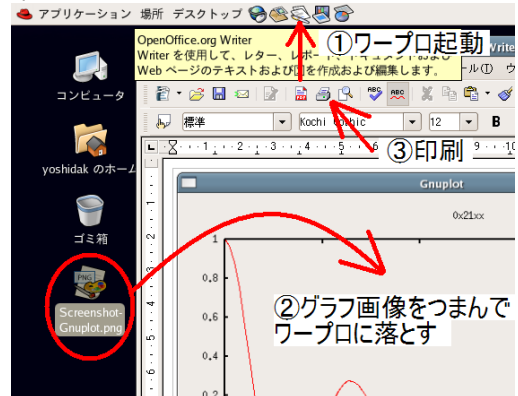
アクティブでない Window



(b) (Alt)を押しながら(Print Screen)をタイプし、グラフの画像をデスクトップに保存する。



(c) グラフの画像ファイルをつまんで、ワープロに落とし、ワープロ文書として印刷する。

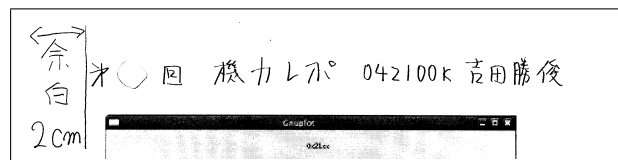


他人のものと区別するため必ず「学籍番号」を記せ。条件や考察を書きこめばレポートで流用可。

第1回レポート課題 プログラム「prog-1.m」のグラフを、以上の要領で印刷し、提出せよ。

### レポート提出方法

- A4 一枚以内 (両面使用可) . 左余白 2cm 以上 .
- 上に「第 X 回機力レポ 学籍番号 氏名」と明記 .
- ✂切: 次回授業日の前日まで . 吉田准教授室 Box . 次々回授業日からは受けとらない .
- 学籍番号なきグラフは盗作と見なす .



注意 レポートは、新入生を読者に想定して書くこと。グラフの軸にラベルが無かったり、解説や考察が無いと、何をどう理解すればよいか分らない。レポートとは、ある出来事の体験者が、未体験者に向けて書く文書。

# 第 3 講

## 自由振動のモデル

大学レベルの高等な力学や数学は一旦忘れてよい。以下の内容は、高校の理科と数学で十分に理解できる。

### 3.1 ニュートンの運動法則

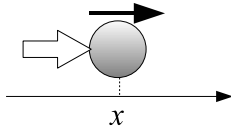
高校物理 高校物理の運動方程式，



$$ma = F \quad (3.1)$$

は知っていると思う<sup>1)</sup>。すなわち、質点の加速度  $a$  に質量  $m$  を乗じて作ったベクトル  $ma$  は、質点を受ける外力のベクトル  $F$  に等しい、これをニュートンの第 2 法則と呼んだ。以上が高校物理。

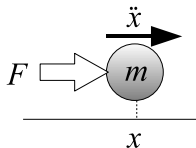
大学物理 以上の加速度  $a$  の代わりに、位置を表わす変数  $x$  を導入すると、大学の物理になる。



ここで 17 世紀の大発見を使う。質点の速度  $v$  と加速度  $a$  は、微分法によって、 $(:=$  は定義の等号)

$$v = \dot{x} := \frac{dx}{dt}, \quad a = \ddot{x} := \frac{d^2x}{dt^2}$$

と書ける ( $t$  は時間)。この大発見は運動方程式 (3.1) を、



$$m\ddot{x} = F \quad (3.2)$$

という形に書き換えた。

運動方程式 (3.1) を、わざわざ (3.2) に書き換えたのは、未知数を位置  $x$  にしたいからである。(3.1) を解いても加速度  $a$  しか分らないが、(3.2) を解けば位置  $x$  が分かる。時々刻々と変化する質点の位置  $x$  は、一般に、時間の関数  $x(t)$  となるが、現代的には、この  $x(t)$  のことを質点の運動 (motion) というわけだ。

以上まとめると、高校物理 (3.1) に、微分法の発見  $a = \ddot{x}$  を代入したのが大学物理 (3.2) である。式 (3.2) を解くと、質点の運動  $x(t)$  が判明する。

<sup>1)</sup>暗記は  $F = ma$  かな。

<sup>2)</sup>少なくとも製品開発の現場では、無用の長物かも知れない。

運動の例 外力  $F$  が簡単なときは、運動  $x(t)$  は積分で解ける。一例として、重力  $F = -mg$  を考えよう。運動方程式は、 $m\ddot{x} = F = -mg$  より、

$$\ddot{x} = -g \quad (g > 0 \text{ は定数}) \quad (*)$$

となる。この式に合う関数  $x(t)$  をを見つけるには、高校で習う次の定理を使えばよい。

算法 1 (微積分学の基本定理) 微分したものを積分すると元に戻る。ただし積分定数が付く。すなわち、

$$\int \frac{dx}{dt} dt = x + C \quad (C \text{ は積分定数})$$

与式 (\*) の両辺を  $t$  で積分して算法 1 を使う。まず、 $d^2x/dt^2$  を積分すると  $dx/dt$  に戻るから、

$$\begin{aligned} \frac{d^2x}{dt^2} = -g &\implies \int \left(\frac{d^2x}{dt^2}\right) dt = \int (-g) dt \\ &\implies \frac{dx}{dt} + C_1 = -gt + C_2 \\ &\implies \frac{dx}{dt} = -gt + C \quad (C := C_2 - C_1) \end{aligned}$$

もう 1 回積分すると、 $dx/dt$  が  $x$  に戻り、

$$\begin{aligned} \implies x + D_1 &= -g\frac{t^2}{2} + Ct + D_2 \\ \therefore x(t) &= -g\frac{t^2}{2} + Ct + D \quad (D := D_2 - D_1) \quad (**) \end{aligned}$$

となる。得られた未知数つきの運動 (\*\*) を、与式 (\*) の一般解という。

未知数を消去するため、初期位置  $x(0) = x_0$  と初期速度  $\dot{x}(0) = v_0$  を代入すると、 $x(0) = -g\frac{0^2}{2} + C \cdot 0 + D = D = x_0$ 、 $\dot{x}(0) = -g \cdot 0 + C = C = v_0$  より、具体的な放物運動、

$$x(t) = -g\frac{t^2}{2} + v_0 t + x_0 \quad (***)$$

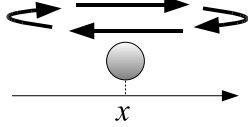
が確定する。

本講義の方針 以上が、微積分を用いた運動  $x(t)$  の解法である。これが現代力学のエッセンスであり、技術者の教養としてはこの程度で十分だ。外力  $F$  が複雑で算法 1 が使えない場合には、手計算は諦めて、数値解析ソフトで数値解を求める。

なぜなら、いくら高級な数学を駆使しても、機械工学的には単純すぎる例題しか解けないことが多いからだ<sup>2)</sup>。というのも、極めて単純な機構でも、非線形性という性質を持つ場合が多く、そうなると原理的に手計算できない。

## 3.2 自由振動のモデル

思考実験によって、振動を起こそう。振動的であるとは、行ったり来たり、が繰り返すことである。

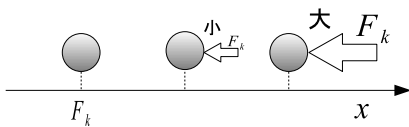


前述の放物運動は、 $t$  の 2 次関数であり、最後に行ったりきたりだから振動ではない。

### 3.2.1 線形復元力

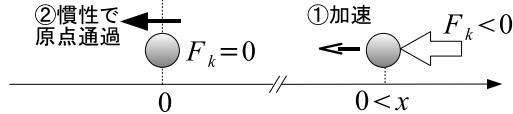
そこで、質点に、質点を原点に引き戻す力、

$$F_k := -kx \quad (k > 0 \text{ は定数}) \quad (3.3)$$

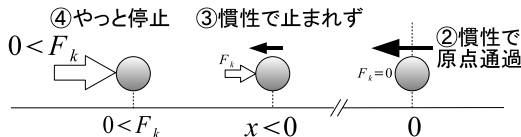


を加える。  $F_k$  を線形復元力<sup>3)</sup>、 $k$  をばね定数という。

このとき、(1) 正の位置  $x > 0$  にある質点は、原点方向  $-kx < 0$  に引かれて加速する。(2) 復元力は原点  $x = 0$  で消えるが、質点は慣性で原点を通過する。



(3) こんどは正の方向  $-kx > 0$  に引かれるが、慣性によってすぐには止まれない。(4) 行きすぎて止る。



そして、逆向きに同じことが繰り返される、はず?!

この程度の確証を得るには手計算のほうが楽だ。運動方程式は  $m\ddot{x} = F_k = -kx$  より、

$$\ddot{x} = -(k/m)x$$

だが、 $x(t) = \sin(\sqrt{k/m}t)$  や  $x(t) = \cos(\sqrt{k/m}t)$  という解の存在が容易に確かめられる。実際、

$$\begin{aligned} x(t) &= \sin(\sqrt{k/m}t) \\ \Rightarrow \dot{x}(t) &= \sqrt{k/m} \cos(\sqrt{k/m}t) \\ \Rightarrow \ddot{x}(t) &= -(\sqrt{k/m})^2 \sin(\sqrt{k/m}t) = -(k/m)x(t) \end{aligned}$$

同様に  $\cos$  のほうも示せる。このような  $\sin, \cos$  波状の運動を単振動または調和振動という<sup>4)</sup>。

<sup>3)</sup>  $x$  の線形式 (1 次式) であることから、線形という。

<sup>4)</sup> 単に三角関数というが  $\tan$  などが含まれてしまうので、 $\sin, \cos$  だけを取り分けて調和関数と別称する。それが「調和」の由来。

<sup>5)</sup> 関数  $f(X) = X|X|$  は、2 次関数のカーブで第 3 象限から第 1 象限に向かう。

<sup>6)</sup> 単独の 1 次関数で書けない関数を、非線形関数という。

<sup>7)</sup> このように、項を左辺にまとめて表記することが多い。

以上、線形復元力  $F_k = -kx$  を加えることで、単振動という持続振動を引き起こすことに成功した。

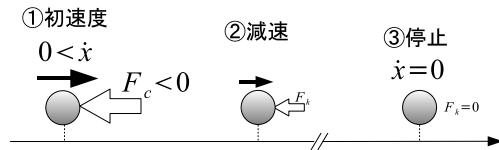
### 3.2.2 線形粘性抵抗

現実の振動は、永遠には持続しない。クーロン摩擦や流体抵抗などのブレーキ力を受けて必ず減衰する。そこで、次のようなブレーキ力を想定しよう。

$$F_c := -c\dot{x} \quad (c > 0 \text{ は定数}) \quad (3.4)$$

$F_c$  を線形粘性抵抗、 $c$  を減衰係数という。

このとき、(1) 正の速度  $\dot{x} > 0$  を持つ質点には、逆向きのブレーキ力  $F_c = -c\dot{x} < 0$  が加わり、(2) 質点は減速する。運動中  $\dot{x} \neq 0$  は常にブレーキが効くので、外部からのエネルギー供給がない限り、(3) 質点はいずれ停止する。



以上、線形粘性抵抗  $F_c = -c\dot{x}$  というブレーキ力を加えることにより、減衰作用を作りだせる。

補足 ちなみに、有名なクーロン摩擦はというと、

$$F_{cl} = \begin{cases} -\mu R & (\dot{x} > 0) \\ 0 & (\dot{x} = 0) \\ \mu R & (\dot{x} < 0) \end{cases} \quad (\mu, R > 0 \text{ は定数})$$

などと書ける。 $\mu$  は摩擦係数、 $R$  は垂直抗力である。また速度の 2 乗に比例する空気抵抗は、

$$F_{ae} = -\alpha\dot{x}|\dot{x}| \quad (\alpha > 0 \text{ は定数})$$

と書ける<sup>5)</sup>。どちらも減衰の仕方が線形粘性抵抗と異なる。いずれも非線形関数<sup>6)</sup>なので、運動  $x(t)$  が手計算できないため、初等理論では無視される。

### 3.2.3 自由振動系

以上の復元力  $F_k$  と抵抗  $F_c$  を同時に与える。運動方程式は、 $m\ddot{x} (= F = F_k + F_c) = -c\dot{x} - kx$  より、

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (3.5)$$

と書ける<sup>7)</sup>。これが、振動現象の最も単純なモデルとなる。これ以上単純化すると、振動が起らない。

式 (3.5) の質量  $m$ 、減衰係数  $c$ 、ばね定数  $k$  を調整すると、現実の様々な振動現象を真似できる。式 (3.5) はフルネームで「1 自由度の線形自由振動系」というが、簡単のため、自由振動系と略称しよう。図 3.1 のように図示する。

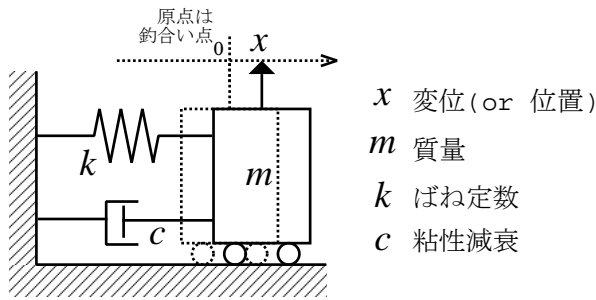


図 3.1: 自由振動系

### 3.3 自由振動系の実現

以上、質点に復元力と抵抗が働く状況を空想することによって、自由振動のモデルを構成した。同様の仕組みを、物理的に実現する方法を述べよう。

#### 3.3.1 機械構造としての実現

我々が住む実空間には、素材の弾性による復元力や、部材間の粘性や摩擦による抵抗が存在するので、ほとんど全ての物体は、自然状態として、図 3.1 のような自由振動系の性質を示す。

このような自然状態から出発して、人間の作為によって、ばね特性や減衰特性を再調整したいときには、次のような機械部品を活用する。

機械式ばね



機械式減衰器 (ダンパー)



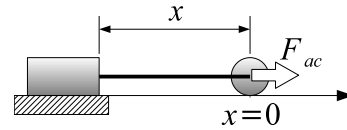
#### 3.3.2 フィードバック制御としての実現

機械式のばねや減衰器を使用すると、ばね定数や減衰係数の調整のたびに部品交換が必要になり、いちいち機械を止めなければならない。

このような手間を避けるために、電気信号を力に変換するアクチュエータというメカトロニクス部品が考案された。



これは、直動アクチュエータというタイプで、シャフトを伸縮させながら、入力電圧に比例した力を発生する。これを質点に装着しよう。アクチュエータの発生力を  $F_{ac}$  とする。



ここで、適当なセンサで質点の位置  $x$  と速度  $\dot{x}$  を測定しながら、それらに比例した電圧を出力する測定回路を組んでやると、アクチュエータに、

$$F_{ac} = -C\dot{x} - Kx$$

のような力を発生させることができる。 $C, K$  はアクチュエータの感度を表わす定数で、ゲインと呼ばれる。このような、測定値  $x, \dot{x}$  に応じた力  $F_{ac}$  を加える操作を、フィードバック制御という。フィードバック制御を受けた質点の運動方程式は、 $m\ddot{x} = F_{ac}$  より、

$$m\ddot{x} + C\dot{x} + Kx = 0$$

となり、ばね定数  $K$ 、減衰係数  $C$  の線形振動系と全く同じ動作が実現する。機械式と違って、 $K, C$  の値は、ボリュームコントローラや、コンピュータにより電氣的に調整できるので、部品交換は不要、機械を止めずに  $K, C$  を変更できる。アクティブ・サスペンションなどは、このように実現されている。同様の発想で、手動操作系の安全性や質感を向上させる試みも始まっている。

#### 3.3.3 数値解としての実現

物理的な実現方法のみならず、数理的な実現方法も役に立つ。実現されるのは、自由振動する数値である。この数値をアニメーション等で可視化し、机上検討の資料とするわけだ。Octave による実現方法はプログラム「prog-1.m」p.7 に示した。

横軸を時間  $t$ 、縦軸を変位  $x(t)$  にとったグラフに、振動を描いたものを振動波形という。 $m, c, k$  を変更すると、3 種類の振動波形が作れる。

実習 3.1 プログラム「prog-1.m」p.7 の 2 行目「m=1.0; c=0.4; k=1.0;」の数値を変更し、3 種類の振動波形に類似するものを作れ。各グラフと対応する  $m, c, k$  の値を、ワープロに並べて示せ。

### 3.4 数値解法

Octave プログラム「prog-1.m」は、自由振動の数値解を求めるプログラムだから、運動方程式、

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (3.5) \text{ 再掲}$$

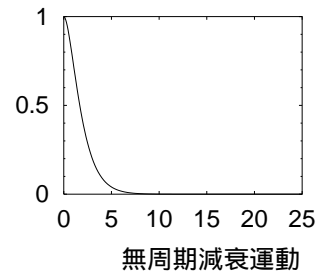
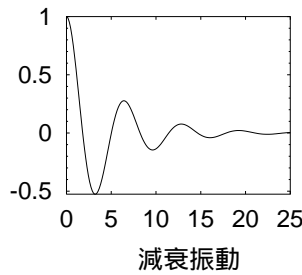
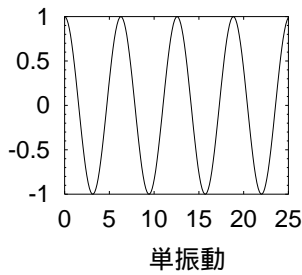


図 3.2: 3 種類の振動波形

が書かれていなければならない。どこに？ — 実は、そのものズバリは書かれておらず、

$$dx(1) = x(2);$$

$$dx(2) = -(c/m)*x(2) - (k/m)*x(1);$$

という別表記が書かれている。これを運動方程式の「1 階化」という。以下に説明する。

### □ 3.4.1 運動方程式の 1 階化

実は、コンピュータは 2 階微分を計算するのが苦手だ。そこで、ダミーの変数を追加することで、2 階微分を無くす方法が編み出された。例えば、

$$\ddot{x} + \dot{x} + x = 0$$

の 2 階微分  $\ddot{x}$  を、見かけ上消去したいときは、1 回微分を別の変数に置いた  $\dot{x} = y$  を連立する。

$$\begin{cases} \dot{x} = y \\ \dot{y} + y + x = 0 \end{cases}$$

$\ddot{x} = \dot{y}$  などに注意して、整理すると、

$$\begin{cases} \dot{x} = y \\ \dot{y} = -y - x \end{cases}$$

とできる。以上を、運動方程式の 1 階化という。

3 階微分  $\ddot{\ddot{x}} + \ddot{x} + \dot{x} + x = 0$  のときも、2 階微分までを別の変数、 $\dot{x} = y$ ,  $\ddot{x} = z$ , において、

$$\begin{cases} \dot{x} = y \\ \dot{y} = z \\ \dot{z} = -z - y - x \end{cases}$$

のように 1 階化できる。

**実習 3.2**  $7\ddot{\ddot{x}} + 5\dot{x} + 3x = 0$  を 1 階化せよ。

### □ 3.4.2 オイラー法

Octave の `lsode` の内部では、これから述べるオイラー法の高精度版が動作している。例題として、

$$\dot{x}(t) = -2x(t)$$

の数値解を求めよう。数値解は電卓で求まる。

電卓は極限を扱えない。ゆえに微分も作れないので、微分の定義にある極限操作をあきらめる。

$$\dot{x}(t) \approx \lim_{h \rightarrow 0} \frac{x(t+h) - x(t)}{h} = -2x(t)$$

$h$  が十分小さければ、上式でいたい良からう。辺々  $h$  を乗じて、左辺を未来、右辺を現在に整理すると、

$$x(t+h) = x(t) - 2x(t)h$$

という漸化式を作る。 $h$  を積分ステップという。

具体例として、積分ステップ  $h = 0.1$  と、初期値  $x(0) = 1$  を代入し、

$$x(0) = 1,$$

$$x(0.1) = x(0) - 2x(0)0.1 = 0.8,$$

$$x(0.2) = x(0.1) - 2x(0.1)0.1 = 0.64,$$

$$x(0.3) = x(0.2) - 2x(0.2)0.1 = 0.512, \dots$$

のように代入を繰り返すと、近似的な数値解が求まっていく。以上を、オイラー法という。

2 階の方程式  $\ddot{x} + \dot{x} + x = 0$  も、1 階化して同様に解く。 $\dot{x} = v$ ,  $\ddot{x} = \dot{v}$  とおいて、

$$\begin{cases} \dot{x} = v \\ \dot{v} = -v - x \end{cases}$$

のように 1 階化すれば、

$$\begin{cases} x(t+h) = x(t) + (v(t))h \\ v(t+h) = v(t) + (-v(t) - x(t))h \end{cases}$$

という漸化式が作れる。初期位置  $x(0) = 1$ , 初速度  $v(0) = 0$ , 積分ステップ  $h = 0.1$  などとすれば、

$$\begin{cases} x(0) = 1 \\ v(0) = 0 \end{cases}$$

$$x(0.1) = x(0) + (v(0))0.1 = 1$$

$$v(0.1) = v(0) + (-v(0) - x(0))0.1 = -0.1$$

$$x(0.2) = x(0.1) + (v(0.1))0.1 = 0.99$$

$$v(0.2) = v(0.1) + (-v(0.1) - x(0.1))0.1 = -0.19$$

などとして、同様に数値解が求まっていく。

**実習 3.3**  $t = 0.3$  における変位  $x(0.3)$  と、速度  $v(0.3)$  を求めよ。

# 第 4 講

## 振動の固有値

例えば、質量  $m = 5.6 \text{ kg}$ 、ばね定数  $k = 7.8 \text{ N/m}$ 、減衰係数  $c = 1.5 \text{ Ns/m}$  の振動系があったとする。この振動系が振動するのか、しないのか、即答できるだろうか？ — 2 次方程式を解けば即答できる。

### 4.1 振動の固有値 (特性値)

$m, c, k$  を構造パラメータと呼ぼう。構造パラメータは、各種設計便覧などをもとに設計図から算定できる。構造パラメータは、機械構造を連想するのは都合がよいが、動作の予測には向かない。

そこで、構造パラメータ  $m, c, k$  を何らかの算法で加工して、動作を表わす特性パラメータを作れないかというアイデアが浮上する。この目的のために、固有方程式という便利な道具が発明された。

冒頭の振動系 ( $m = 5.6, c = 7.8, k = 1.5$ ) の運動方程式は、 $5.6\ddot{x} + 7.8\dot{x} + 1.5x = 0$  と書ける。これと同じ係数をもつ 2 次方程式

$$5.6s^2 + 7.8s + 1.5 = 0$$

を、固有方程式または特性方程式という。その根は、

```
octave:1> roots([5.6, 7.8, 1.5])
ans =
   -1.16243
   -0.23043
```

となるが、これらを固有値または特性値という。

得られた固有値  $\approx -1.2, -0.2$  を、見る人が見れば、

- 冒頭の振動系は、振動せずに減衰する。

ということが瞬時に分かる。以下、その理屈を説明する。固有値から振動波形を連想できれば、目標達成である。固有値は、振動論の大きな 2 本の柱のうちの一つである。(もう一つは「周波数特性」参照)

### 4.2 固有値の求め方

数学は実体を伴わない空想だから、「マニアの発明による奇抜な算法」という側面があり、なぜそう計算すべきなのか、物理的には説明できないことも多い。ただし、文芸や映画などの空想と違って数学の空想は客観的だ。どんなに浮世離れた計算法であっても、それが数学として証明されている限り、誰が計算(空想)しても同じ計算結果(空想結果)になる。

で、面白いことに、こうしたマニアによる計算法が、たまたま物理現象に対応してしまうという奇跡が起こる。固有値はその一例である。

先人が発明した固有値の求め方を述べよう。まず、運動方程式  $m\ddot{x} + c\dot{x} + kx = 0$  に、 $x(t) = e^{st}$  を代入すると、既知の計算法を組み合わせると、

$$ms^2e^{st} + cse^{st} + ke^{st} = (ms^2 + cs + k)e^{st} = 0$$
$$\therefore ms^2 + cs + k = 0$$

までを得る ( $\because e^{st} \neq 0$ )。以上を踏まえて、

算法 2 運動方程式に  $x(t) = e^{st}$  を代入し、両辺を  $e^{st} \neq 0$  で割って得られる代数方程式

$$ms^2 + cs + k = 0$$

を、固有方程式または特性方程式という。その根、

$$s = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m}$$

を、固有値または特性値という。

3 階の常微分方程式  $a\ddot{x} + b\dot{x} + cx + dx = 0$  のときも、全く同様に  $x = e^{st}$  を代入して  $e^{st} \neq 0$  で割ることで、 $as^3 + bs^2 + cs + d = 0$  という固有方程式が得られる。 $n$  階も同様に計算する。

ここまでの話では、物理との対応はまだ何もない。ただ、そういう計算法が発明されただけである。

実習 4.1 振動系  $\ddot{x} + 2\dot{x} + 5x = 0$  の固有値を、Octave で求めよ。2.4.4 節 p.6 を復習せよ。

### 4.3 固有値と振動のパターン

2 次方程式の根は一般に複素数  $\alpha \pm \beta i$  である。ゆえに固有値もまた複素数であり、そのパターンは、

- A. 相異なる実数。判別式  $> 0$
- B. 純虚数の対。  $c = 0$ , 判別式  $< 0$
- C. 共役複素数の対。  $c \neq 0$ , 判別式  $< 0$
- D. 実数の重根。判別式  $= 0$

の 4 種類に分類できる。

実は、こうした固有値の 4 パターンが、4 種類の振動状態にそのまま対応する (D は最終的に省ける)。

固有値と振動波形を具体的に対応づけるには、常微分方程式論で発明された次の算法を使う。

算法 3 運動方程式  $m\ddot{x} + c\dot{x} + kx = 0$  の固有値が  $s_1, s_2$  であるとき、解  $x(t)$  は、

$$x(t) = C_1e^{s_1t} + C_2e^{s_2t}$$

と書ける。(  $C_1, C_2$  は初期条件で定まる定数)

例外的に、固有値が重根  $s_1 = s_2$  の場合は、 $x(t) = (C_1 + C_2t)e^{s_1t}$  という形式になる。

□ A. 相異なる実数

固有値実部の正負を見ると，収束性が分かる．

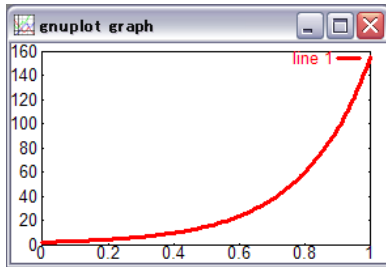
(1) 固有値=(+, +) の場合 例えば  $s_1 = 5, s_2 = 2$  のときの解は，算法 2 p.12 より，

$$x(t) \approx e^{5t} + e^{2t}$$

みたいな格好になる<sup>1)</sup>．Octave でグラフを確認せよ．

#以降は説明用．実行時には書かないこと．

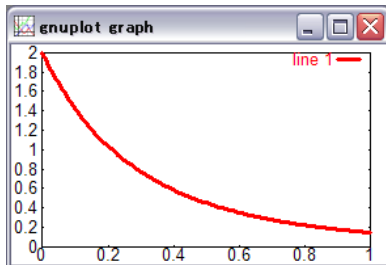
```
octave:1> t=[0:0.01:1]; #公差 0.01 の等差数列
octave:2> s1=5; s2=2;
octave:3> plot(t, exp(s1*t)+exp(s2*t));
```



固有値が正だと， $e^{(\mathbb{E})t} \rightarrow \infty$  より，運動  $x(t)$  は発散する．

(2) 固有値=(-, -) の場合 例えば， $s_1 = -5, s_2 = -2$  とすれば， $x(t) \approx e^{-5t} + e^{-2t}$  のグラフは，

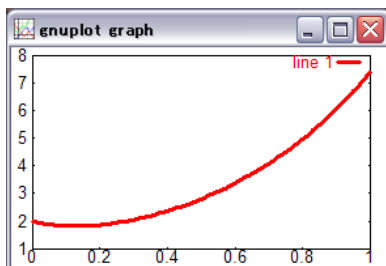
```
octave:1> t=[0:0.01:1];
octave:2> s1=-5; s2=-2;
octave:3> plot(t, exp(s1*t)+exp(s2*t));
```



固有値が負だと， $e^{(\text{負})t} \rightarrow 0$  より，運動  $x(t)$  は収束する．

(3) 固有値=(-, +) の場合 例えば， $s_1 = -5, s_2 = 2$  のとき， $x(t) \approx e^{-5t} + e^{2t}$  のグラフは，

```
octave:1> t=[0:0.01:1];
octave:2> s1=-5; s2=2;
octave:3> plot(t, exp(s1*t)+exp(s2*t));
```



固有値が正，負の複合型だと運動  $x(t)$  は発散する．減少項  $e^{(\text{負})t} \rightarrow 0$  は 0 より小さくならないため，最終的に発散項  $e^{(\mathbb{E})t} \rightarrow \infty$  の効果が勝る．

以上の計算例より，次の法則が判明した．

法則 1 固有値実部  $> 0$  が 1 つでもあると，運動  $x(t)$  は発散する．

運動が発散するとき，運動は不安定であるという．

□ B. 一对の純虚数

例えば， $s_1 = -5i, s_2 = 5i$  を，算法 2 p.12 に代入すると，

$$x(t) \approx e^{(-5i)t} + e^{(5i)t}$$

となる．虚数を肩ののせた指数関数なんて，ありえない空想に見えるかも知れないが，次の公式によって，ありえる空想であることが分かる．

算法 4 (オイラーの公式) 複素数  $x + iy$  の極座標表示  $(r \cos \theta) + i(r \sin \theta)$  は，計算法則が指数関数と同じになる．この性質を利用して，

$$r \cos \theta + ir \sin \theta \equiv re^{i\theta}$$

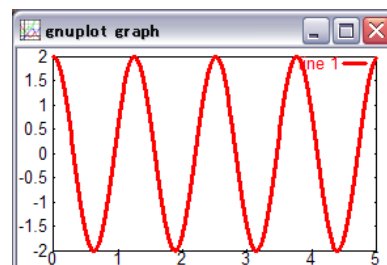
と表記する．

この公式を使うと，

$$\begin{aligned} x(t) &\approx e^{(-5i)t} + e^{(5i)t} = e^{(-5t)i} + e^{(5t)i} \\ &= \{\cos(-5t) + i \sin(-5t)\} + \{\cos(5t) + i \sin(5t)\} \\ &= \{\cos(5t) - i \sin(5t)\} + \{\cos(5t) + i \sin(5t)\} \\ &= 2 \cos(5t) \end{aligned}$$

のように虚部がキャンセルする．このように，意外にも，純虚数の固有値からは，単振動がでる！

```
octave:1> t=[0:0.01:1];
octave:2> s1=-5*i; s2=5*i; # i は純虚数
octave:3> plot(t, exp(s1*t)+exp(s2*t));
octave:4> t=[0:0.01:5]; #時間軸を 5 まで延長
octave:5> plot(t, exp(s1*t)+exp(s2*t));
```



というわけで，2 つめの法則も判明した．

法則 2 固有値虚部  $= 0$  のとき，運動  $x(t)$  は振動的になる．(固有値虚部  $= 0$  のときは振動しない)

<sup>1)</sup> 正確には係数が  $C_1 = C_2 = 1$  の解である．初期条件を逆算すると， $x(0) = e^{5 \cdot 0} + e^{2 \cdot 0} = 2, \dot{x}(0) = 5e^{5 \cdot 0} + 2e^{2 \cdot 0} = 7$ ．

表 4.1: 固有値  $a \pm ib$  による振動状態  $x(t)$  の分類

$a \pm ib$	$b = 0$ (非振動)	$b \neq 0$ (振動)
$a < 0$ 減衰 (安定)	$x(t) =$ 減衰・非振動	$x(t) =$ 減衰・振動
$a = 0$ 一定 (中立安定)	$x(t) =$ 一定値	$x(t) =$ 単振動*
$a > 0$ 発散 (不安定)	$x(t) =$ 発散・非振動	$x(t) =$ 発散・振動

\* 単振動 = 一定振幅で振動

### C. 一対の複素数

法則 1 p.13 と法則 2 は、初等的な指数法則によって組み合わせて使うことができる。例えば、 $s_1 = -1 - 5i$ ,  $s_2 = -1 + 5i$  とすると、

$$\begin{aligned} x(t) &\approx e^{(-1-5i)t} + e^{(-1+5i)t} \\ &= e^{-t-5it} + e^{-t+5it} = e^{-t}e^{(-5t)i} + e^{-t}e^{(5t)i} \\ &= e^{-t}\{e^{(-5t)i} + e^{(5t)i}\} = e^{-t} \cdot 2 \cos 5t \quad (4.1) \end{aligned}$$

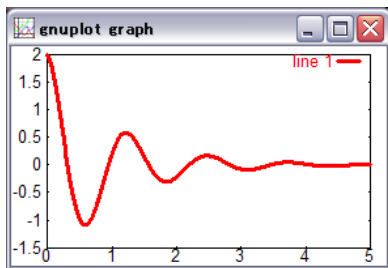
となり、単振動  $2 \cos 5t$  の振幅が指数関数的に減少するような、減衰振動が得られた。

結局のところ、固有値が複素数の場合にも、法則 1, 法則 2 がそのまま成立する。すなわち、

- 「法則 1 で実部  $< 0$ 」より収束  $e^{-t} \rightarrow 0$ 。
- 「法則 2 で虚部  $\neq 0$ 」より振動  $2 \cos 5t$ 。

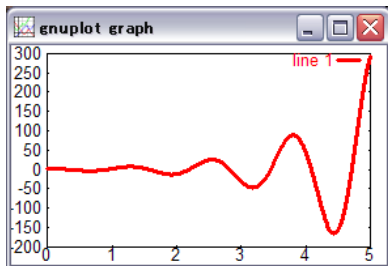
である。

```
octave:1> t=[0:0.01:5];
octave:2> s1=-1-5*i; s2=-1+5*i;
octave:3> plot(t, exp(s1*t)+exp(s2*t));
```



実部  $> 0$  に変更すると、法則 1 より発散するが、虚部  $\neq 0$  だから法則 2 より振動成分が残る。

```
octave:1> t=[0:0.01:5];
octave:2> s1=1-5*i; s2=1+5*i;
octave:3> plot(t, exp(s1*t)+exp(s2*t));
```



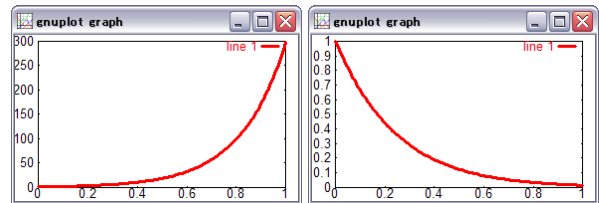
### D. 実数の重根

固有値が重根のときは、例えば  $s_1 = s_2$  に対して、

$$x(t) = (C_1 + C_2 t)e^{s_1 t}$$

という異例の形式になるが、運動のパターンは A のときと区別できない。法則 1 p.13 もそのまま成立して、実部  $> 0$  なら不安定、実部  $< 0$  なら安定である。

```
octave:1> t=[0:0.01:1];
octave:2> s1=5; # 下図の左
octave:3> plot(t, (1+t).*exp(s1*t));
octave:4> s1=-5; # 下図の右
octave:5> plot(t, (1+t).*exp(s1*t));
```



## 4.4 固有値による自由振動の予測

法則 1, 法則 2 によれば、固有値実部の符号、虚部の有無をチェックするだけで、自由振動の様子を予測することができる。こうして数え挙げた、自由振動の全パターンを、表 4.1 に示す。

このように、図 3.1 p.10 の自由振動系には、固有値実部の符号、虚部の有無に応じて、全部で 6 パターンの振動 (or 運動) が起こる。これ以外の可能性はない。

例えば、冒頭の振動系  $5.6\ddot{x} + 7.8\dot{x} + 1.5x = 0$  の固有値  $\approx -1.2, -0.2$  には虚部が無く、実部が全て負だから、表 4.1 より、この振動系の運動  $x(t)$  は非振動的に減衰することが分かる。

実習 4.2 振動系  $\ddot{x} + 2\dot{x} + 5x = 0$  の固有値は  $-1 \pm 2i$  であった。振動波形を表 4.1 から予想しスケッチせよ。

実習 4.3 前問のスケッチを、 $\ddot{x} + 2\dot{x} + 5x = 0$  の実際の振動波形と比較せよ。プログラム「prog-1.m」p.7 の 2 行目「m=1.0; c=0.4; k=1.0;」の数値を変更し、Octave で実行すればよい。



# 第 5 講

## 減衰比と固有振動数

構造パラメータ  $m, c, k$  を固有値  $\alpha \pm i\beta$  に変換し,  $\alpha, \beta$  の値をチェックすると振動状態が分かった. 理論上はこれで必要十分だが, 製造現場には, これとはまた別の特性パラメータが普及している.

### 補足: 振動数の求め方

その前に前回の補足を述べておく. 以下, 複素数  $z = \alpha + \beta i$  の実部と虚部を,

$$\begin{aligned} \text{実部 } \alpha &= \text{real}[\alpha + \beta i] \\ \text{虚部 } \beta &= \text{imag}[\alpha + \beta i] \end{aligned}$$

という演算記号で表わす.

さて, 式 (4.1) p.14 を復習すると, 固有値が,

$$s = \alpha \pm \beta i, \quad i = \sqrt{-1}$$

のときの自由振動解は, 係数を端折ると,

$$\begin{aligned} x(t) &\approx e^{(\alpha - \beta i)t} + e^{(\alpha + \beta i)t} = e^{\alpha t - (\beta t)i} + e^{\alpha t + (\beta t)i} \\ &= e^{\alpha t} e^{(-\beta t)i} + e^{\alpha t} e^{(\beta t)i} = e^{\alpha t} \{e^{(-\beta t)i} + e^{(\beta t)i}\} \\ &= e^{\alpha t} \cdot 2 \cos \beta t \approx e^{\alpha t} \cos \beta t \quad \therefore \text{算法 4 p.13} \end{aligned}$$

より,  $x(t) \approx e^{\alpha t} \cos \beta t$  と書けた.

以上の結果を並べると, 便利な関係が判明する.

$$\begin{aligned} \text{固有値 } s &= \alpha \pm \beta i \\ \text{振動解 } x(t) &\approx e^{\alpha t} \cos \beta t \end{aligned}$$

すなわち, 複素固有値  $s$  の実部  $\text{real}[s]$  は自由振動の指数減衰率に, 虚部の絶対値  $|\text{imag}[s]|$  は振動数に, それぞれ対応している. 以上, 次の算法が判明した.

算法 5 固有値  $s = \alpha \pm \beta i$  に対する自由振動解は, おおよそ

$$x(t) \approx e^{\alpha t} \cos \beta t$$

と書ける. すなわち, 固有値  $s$  の実部  $\text{real}[s]$  は指数減衰率を, 虚部の絶対値  $|\text{imag}[s]|$  は振動数を与える.

### 5.1 無次元化

本題に入る. 自由振動系の数学モデル:

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (3.5) \text{ p.9}$$

に 3 個あるパラメータ  $m, c, k$  を, 2 個まで減らす.

表 5.1: 減衰比と固有振動数

減衰比	$\zeta$	$\frac{c}{2m\omega_n} = \frac{c}{2\sqrt{mk}}$
固有振動数	$\omega_n$	$\sqrt{\frac{k}{m}}$

□ 純数学的に  $m > 0$  で割る:

新しいパラメータ  $C = c/m, K = k/m$  により,

$$\ddot{x} + C\dot{x} + Kx = 0$$

のように 2 個まで減らせる. 固有値は,

$$\frac{-C \pm \sqrt{C^2 - 4K}}{2} \quad (5.1)$$

となる.

□ 固有値の表現を簡略化する:

式 (5.1) の固有値を見ると,  $\sqrt{2}$  個のパラメータとなっているが, これを  $\sqrt{1}$  個のパラメータに集約し, ついでに分母の 2 も消してしまおう.

そのために, 変数変換  $C = 2\zeta\omega_n, K = \omega_n^2$  を導入して,

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = 0 \quad (5.2)$$

のように書きかえる. いわゆるこれが教科書的な標準形である. 固有方程式  $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$  を解くと, 固有値は,

$$s = \omega_n(-\zeta \pm \sqrt{\zeta^2 - 1}) \quad (5.3)$$

となり, 固有値のパターンが  $\zeta$  にしか依存しなくなった.

このように, 算術的にパラメータの数を減らしてしまうことを無次元化という. 無次元化 (= 変数変換) 後の  $\zeta$  を減衰比,  $\omega_n$  を固有振動数と呼ぶ. ここでいう「無次元」は「無単位」とほぼ同義である. 物理パラメータ  $m, c, k$  との関係を表 5.1 に示す.

無次元化の補足 以上の理解で実務上問題ないが, それだけだと大学の先生に叱られるので, 無次元化の意味をもっと正確に解説しておく.

まず, 人為的な代表長さ  $L$  [m] を用意する. 変位を  $x = Lx^*$  [m] と書くと,  $x^*$  は無単位になる.  $x$  から  $L$  を経て  $x^*$  を得る操作を, 変位  $x$  の無次元化といい, 得られた  $x^*$  を無次元量という.

定数  $a$  について  $\frac{d(ax)}{dt} = a \frac{dx}{dt}$  に注意すると,

$$x = Lx^*, \dot{x} = L\dot{x}^*, \ddot{x} = L\ddot{x}^*$$

が分かる. 運動方程式 (3.5) に代入すると,

$$mL\ddot{x}^* + cL\dot{x}^* + kLx^* = 0$$

が得られる．上式の解  $x^*$  は無単位，すなわち無次元となるため，この一連の操作を，運動方程式の無次元化というわけである．

ここで例えば，作為的に  $L = 1/m$  を選べば，

$$\ddot{x}^* + \frac{c}{m} \dot{x}^* + \frac{k}{m} x^* = 0$$

となるが，これが  $m > 0$  で割ることの意味である．

以上は，ズーム撮影を数学的に模したものになっている．ようするに，実際には質量  $m$  の自由振動系の映像を  $1/m$  倍にスケール(拡大・縮小)して，あたかも質量 1 の運動であるかのように見せるわけだ．

もう 1 つのテクニックとして，時間をスケールしてしまう方法，すなわち高 (or 低) 速度撮影の方法があるが，相似な振動波形を重ねる際に実際に用いる．

## 5.2 減衰比 $\zeta$ の使い道

### 5.2.1 振動パターンの整理

式 (5.3) に求めたように，標準形 (5.2) の固有値は，

$$s = \omega_n(-\zeta \pm \sqrt{\zeta^2 - 1})$$

と書けた．減衰比  $\zeta$  を変えると， $\sqrt{\dots}$  の中身が変化して固有値のパターンが変化する．連動して，標準形  $\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = 0$  の振動パターンが変化する．

各  $\zeta$  の値が，表 4.1 p.14 のどのパターンに対応するか，Octave で調べてみよう．Octave に  $\pm$  記号はないので，+ で代用する．

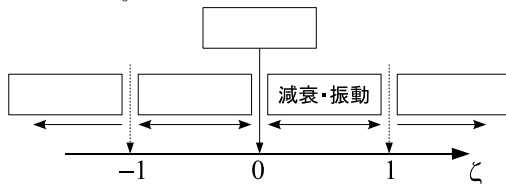
```
octave:1> omn=1;          仮に  $\omega_n = 1$  とおく
octave:2> z=0.5;          $\zeta = 0.5$  の場合を調べる
octave:3> s=omn*(-z+sqrt(z^2-1))  固有値
s = -0.50000 + 0.86603i
```

固有値の実部が負，虚部が  $\neq 0$  なので，この条件は表 4.1 の右上「減衰・振動」に対応している．

表 4.1: 固有値  $a \pm ib$  による振動状態  $x(t)$  の分類

$a \pm ib$	$b = 0$ (非振動)	$b \neq 0$ (振動)
$a < 0$ 減衰 (安定)	$x(t) =$ 減衰・非振動	$x(t) =$ 減衰・振動
$a = 0$ 一定 (中立安定)	$x(t) =$ 一定値	$x(t) =$ 単振動
$a > 0$ 発散 (不安定)	$x(t) =$ 発散・非振動	$x(t) =$ 発散・振動

この結果を， $\zeta$  の数直線に書き込もう．



実習 5.1 残りの空欄を埋めよ．各区間の  $\zeta$  の代表値を選び，固有値を求めて表 4.1 と照合すればよい．

このように，減衰比  $\zeta$  を変化させるだけで，振動の全パターンを網羅できる．以上，減衰比  $\zeta$  には，振動のパターンを一系列に整理させる機能がある．

<sup>1)</sup> 「減衰」を「制動」や「抵抗」で言いかえた本もある．

減衰比の各区間の名称 減衰比の区間によって振動パターンが変わるが，各区間には名前がついている<sup>1)</sup>．

- $\zeta = 0 \dots$  無減衰．
- $0 < \zeta < 1 \dots$  不足減衰．
- $\zeta = 1 \dots$  臨界減衰．
- $1 < \zeta \dots$  過減衰．

実習 5.2 以上の区間名を  $\zeta$  の数直線に書き入れよ．(マイナス部分  $\zeta < 0$  を表わす術語は無さそう)

### 5.2.2 振動波形の相似判定

表 5.1 を復習すると，減衰比は，

$$\zeta = \frac{c}{2\sqrt{mk}}$$

のように定義されている．ということは， $\zeta$  の値が同じでも，対応する構造  $(m, c, k)$  は無数にとれる．

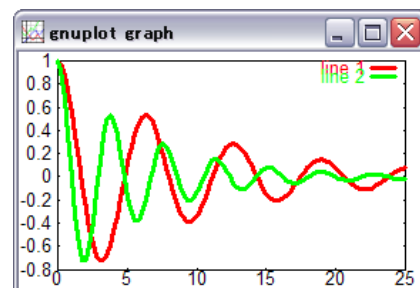
例えば， $(m, c, k) = (3.6, 1.2, 10)$  と  $(1, 0.2, 1)$  の減衰比はともに  $\zeta = 0.1$  で一致するが，構造が異なり振動波形も異なる．

比較用のプログラムを用意した．各自実行せよ．

プログラム「zeta-1.m」

```
global m c k; # この後は無視される
function dx = f(x, t)
    global m c k;
    dx(1) = x(2);
    dx(2) = -(c/m)*x(2) - (k/m)*x(1);
endfunction
x0 = [1; 0];
m1=3.6; c1=1.2; k1=10.0; #1 つ目の条件
m2=1.0; c2=0.2; k2=1.0; #2 つ目の条件
t = linspace(0, 25, 100);
title("0x21xx");
m=m1; c=c1; k=k1;
x1 = lsode("f", x0, t);
m=m2; c=c2; k=k2;
x2 = lsode("f", x0, t);
plot(t, x1(:, 1), t, x2(:, 1))
```

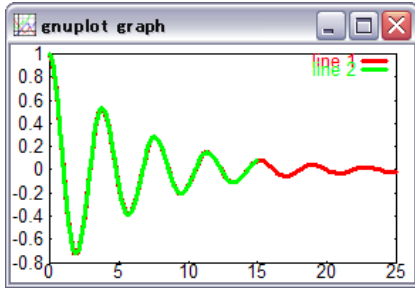
```
octave:1> source "zeta-1.m"
```



となり，確かに振動波形は異なる．

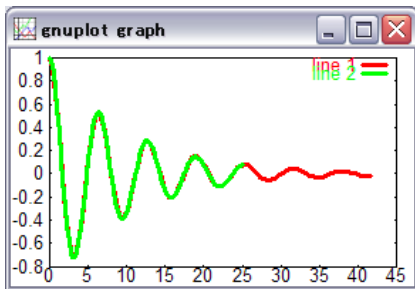
では， $\zeta$  が同じとき何が同じなのか？ — この疑問を解くヒントを実行例で示す．

```
octave:1> source "zeta-1.m"
octave:2> plot(t,x1(:,1), 0.6*t,x2(:,1))
```



このように、時間軸を片方だけ縮めてやると、振動波形を重ねることができる。逆側を同じだけ伸ばすことでも、当然重なる。(時間をスケールするという)

```
octave:1> source "zeta-1.m"
octave:2> plot(1.6667*t,x1(:,1), t,x2(:,1))
```



一般に、次の法則が成立する<sup>2)</sup>。

**法則 3** 減衰比  $\zeta$  が同じとき、振動波形は相似になる。

ちなみに、時間軸の伸縮の倍数 0.6 の出処が不思議だが、実はこれは  $(m, c, k) = (3.6, 1.2, 10), (1, 0.2, 1)$  それぞれの固有振動数 (表 5.1 p.15),

$$\omega_n = \sqrt{\frac{10}{3.6}} = \frac{1}{0.6}, \quad \omega_n = \sqrt{\frac{1}{1}} = 1,$$

の比になっている。

**減衰比  $\zeta$  の補足** 減衰比は、例えば SF 映画を作るのに役立つ。本物の鉄塔と同じように揺れて見える 1/100 モデルを作りたいとする。どうすれば同じに見えるかという、減衰比を同じにすればいいわけだ。このとき振動波形は相似だから、あとは撮影/再生スピードを適切に調整すれば、どちらも同じ振動波形で揺れてくれる。

このようなモデル実験は、なにも SF 映画の専売特許ではなくて、むしろ工業製品の開発コスト削減のために活用するのが本道である。

### 5.3 固有振動数 $\omega_n$ の使い道

ここでも、標準形の固有値 (5.3) p.15,

$$s = (-\zeta \pm \sqrt{\zeta^2 - 1})\omega_n$$

<sup>2)</sup>厳密にいうと、減衰比は自然法則というより、人為的な発明である。現象の説明に便利な、算法上の空想物である。

が出发点になる。

#### □ 5.3.1 実際の振動数 $\omega_d \neq \omega_n$

振動的な条件  $0 < \zeta < 1$  (不足減衰) で考える。このときルート部分は  $\sqrt{\text{負}}$  であるから、

$$\sqrt{\zeta^2 - 1} = i\sqrt{|\zeta^2 - 1|} = i\sqrt{1 - \zeta^2}, \quad i = \sqrt{-1}$$

となり、ゆえに固有値は複素数、

$$s = -(\zeta\omega_n) \pm i(\sqrt{1 - \zeta^2}\omega_n)$$

になる。この減衰振動の振動数は、算法 5 p.15 より、

$$\begin{aligned} \omega_d &= |\text{imag}[-(\zeta\omega_n) \pm i(\sqrt{1 - \zeta^2}\omega_n)]| \\ &= \sqrt{1 - \zeta^2}\omega_n \end{aligned} \quad (5.4)$$

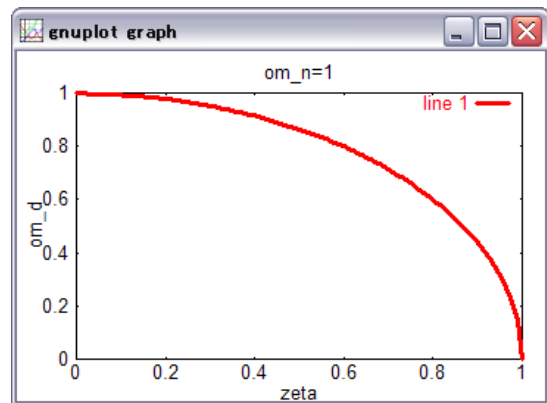
となる。この  $\omega_d$  が減衰振動の実際の速さを表わす。

減衰振動  $0 < \zeta < 1$  の条件下では、したがって、減衰振動の振動数  $\omega_d$  と、固有振動数  $\omega_n$  の大小は、

$$\omega_d = \sqrt{1 - \zeta^2}\omega_n < \omega_n$$

のようになる。実際、固有振動数  $\omega_n = 1$  を基準に、自由振動の振動数を  $\zeta$  の関数  $\omega_d = \omega_d(\zeta)$  としてプロットすると、

```
octave:1> zeta=[0:0.01:1];
octave:2> om_n=1;
octave:3> s=(-zeta+sqrt(zeta.^2-1))*om_n;
octave:4> om_d=imag(s); #虚部を取り出す
octave:5> title("om_n=1");
octave:6> xlabel("zeta");
octave:7> ylabel("om_d");
octave:8> plot(zeta,om_d);
```



のようなグラフが得られる。 $\zeta$  に対して  $\omega_d$  は単調減少する (どんどん振動が遅くなる)、臨界減衰  $\zeta = 1$  で  $\omega_d = 0$  になるが、これは固有値虚部が消えて振動が無くなることの表れである。

$\omega_d = \omega_n$  となるのは無減衰  $\zeta = 0$  のときだが、摩擦抵抗 0 の物体は、まず有り得ないので、次の法則を認めてよからう。

**法則 4** 現実の自由振動系は、おおむね固有振動数  $\omega_n$  付近で揺れる。ただし  $\omega_n$  より確実に遅い。

現場用語 実測した振動波形から測れるのは  $\omega_d$  の方なので、厳密にいうと、固有振動数  $\omega_n$  は測定できない(減衰比と組み合わせて算定する)。ただし、事前に  $\zeta$  が十分小さいと分っているとき、 $\zeta \ll 1$  ならば  $\omega_d \approx \omega_n$ 、であることを利用して、実際に測れる  $\omega_d$  を固有振動数と呼んでしまう場合がある。文脈から判断せよ。

### □ 5.3.2 相似倍率 $\omega_n$

振動論を実用するときには、実際の振動数  $\omega_d = \sqrt{1 - \zeta^2} \omega_n$  さえ分ればよくて、固有振動数  $\omega_n$  は不要である、というのは大きな誤解である。まさに実用上の問題として、本物とモデルを比較する道具として、固有振動数  $\omega_n$  は欠かせないのである。

標準形  $\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = 0$  の解は、固有値実部が  $-\zeta\omega_n$ 、虚部が  $\sqrt{1 - \zeta^2}\omega_n$  であるから、算法 5 より、おおよそ、

$$x(t) \approx e^{-\zeta\omega_n t} \cos(\sqrt{1 - \zeta^2}\omega_n t)$$

のような格好になる。

ここで、高速(or 低速)度撮影を使って、実際の 1 秒 ( $t = 1$ ) が、 $\omega_n$  秒 ( $\tau = \omega_n$ ) かけて再生されるように、時間軸をスケール(伸縮)してみよう。すると、再生時の時間軸は、

$$\tau = \omega_n t$$

と書けるから、再生時の映像のなかで、解は、

$$x(\tau) \approx e^{-\zeta\tau} \cos(\sqrt{1 - \zeta^2}\tau)$$

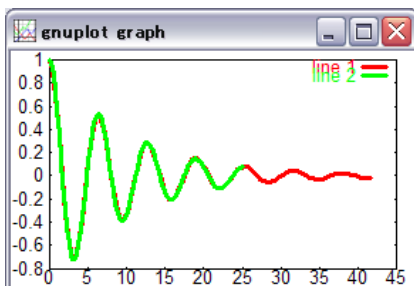
のように見えることになる。撮影速度のトリックを知らない視聴者にとっては、この映像は、固有振動数が  $\omega_n = 1$  のときの振動そのものに映るだろう。

一般に次の法則が成立する。

法則 5 振動波形の時間軸を  $\tau = \omega_n t$  にスケールすると、 $\omega_n = 1$  の振動波形に重なる。(  $\zeta$  は共通とする)

先の例題で確かめると、

```
octave:1> source "zeta-1.m"
octave:2> om_n1=sqrt(k1/m1) #条件 1
om_n1 = 1.6667
octave:3> om_n2=sqrt(k2/m2) #条件 2
om_n2 = 1
octave:4> tau=om_n1*t; #条件 1 をスケール
octave:5> plot(tau,x1(:,1),t,x2(:,1))
```



のように、確かに振動波形が重なる。

以上の操作を運動方程式で表現すると、

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2x = 0$$

$$\downarrow \tau = \omega_n t$$

$$\frac{d^2x}{d\tau^2} + 2\zeta \frac{dx}{d\tau} + x = 0$$

となっている。上の式から下の式を導く方法は、詳細は付録に後述するが、大雑把にいうと、 $\tau = \omega_n t \implies d\tau = \omega_n dt$  を前提に、

$$\frac{d}{dt} = \frac{d}{d\tau} \frac{d\tau}{dt} = \omega_n \frac{d}{d\tau}, \quad \frac{d^2}{dt^2} = \left(\frac{d}{dt}\right)^2 = \omega_n^2 \frac{d^2}{d\tau^2}$$

を作り、これらを代入して、両辺を  $\omega_n^2$  で割ればよい。

このように、振動波形を相似に保ちながらグラフ用紙の縮尺を変える方法を、スケール変換または尺度変換という。

### まとめ

固有値とはまた異なる使い心地の特性パラメータとして、減衰比と固有振動数を導入した。減衰比は振動波形の本質的な形状を定める。同じ減衰比をもつ振動波形は、時間軸のスケール(伸縮)によって互いに重なる。重ねるときの倍率は、固有振動数が与える。

注意すべき点として、減衰比と固有振動数には適用限界がある。機械でいうところの線形振動系、電気でいうところの線形共振回路などは、支配方程式が 2 階(やその連成)なので適用可能である。ところが、3 階以上の一般の支配方程式(例えば、多入力・多出力の制御系など)については、減衰比と固有振動数を定義できない状況が出てくる。

そのようなときでも固有値は有効なので、固有値に戻ればよい。固有値ならどんな階数の系にでも適用できる。

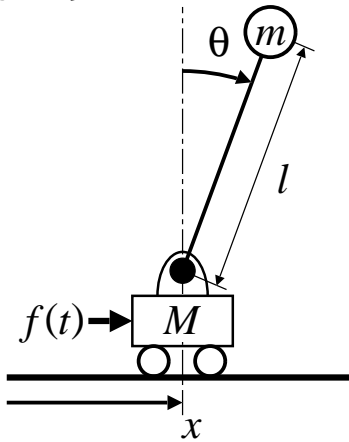
# 第 6 講

## ロボット制御への応用 I

これまでの学習内容を応用すると、自律型倒立ロボットの制御方法を設計できる。ロボットの振動波形(?)を予測しながら、制御パラメータを調整できれば目標達成。

### 6.1 倒立ロボットの力学モデル

自律型倒立ロボットの原型として、次のようなカラクリを考えよう。



質量  $M$  の台車に質量  $m$  の単振り子を取付ける。制御力  $f(t)$  は台車にのみ作用させることができる。倒れ角  $\theta$  をセンサで監視し、それに応じた制御力  $f(t)$  を自動発生できれば、倒立状態を自動的に維持する、自律型倒立ロボットが実現されるはずだ。

詳細は 12 講で学ぶが、このカラクリの運動方程式は、 $\theta$  が小さいと仮定すると、

$$(Ml)\ddot{\theta} - g(M + m)\theta = -f(t) \quad (6.1)$$

と書ける。 $g$  は重力加速度、 $g \approx 9.8 \text{ m/s}$  とする。

簡単のため  $l = 3 \text{ m}$ ,  $M = 1/3 \text{ kg}$ ,  $m = 2/3 \text{ kg}$  の場合を考えると、運動方程式は、

$$\ddot{\theta} - g\theta = -f(t) \quad (6.2)$$

となる。この式は、

- 外力  $-f(t)$  が作用する、無減衰で、ばね定数が負  $-g$  の自由振動系の運動方程式と区別できない。ゆえに、このカラクリは、そのような自由振動系と全く同じように動く。

### 6.2 制御方式と固有値の関係

自由振動系と同じなのだから、このカラクリの動作は、固有値で検討できる。表 4.1 p.14 と照合しながら、

制御方式を机上検討してみよう。結論からいうと、3.3.2 節 p.10 と同じ考え方でロボットは立つ。

#### 6.2.1 制御なしの場合

$f(t) = 0$  より、固有方程式は次式となる。

$$s^2 - g = 0 \quad (6.3)$$

これより固有値は、 $s = \sqrt{g}, -\sqrt{g}$  となり、片方が正なので、振子の角度  $\theta$  は発散する。つまり何もしないと倒れる。支えないで逆立ちさせたのだから、あたりまえ。

#### 6.2.2 角度のフィードバック

何らかの測定値に応じて制御力を調整することをフィードバック制御というが、とりあえず、角度  $\theta$  をフィードバックしてみよう。ようするに、角度  $\theta$  に比例した力で台車を押してやる。この制御方式を、比例制御または P 制御という。

そのために、 $K$  をゲイン (調整パラメータ) として  $f(t) = K\theta$  という外力を台車に作用させる。

$$\ddot{\theta} - g\theta = -f(t) = -K\theta \quad (6.4)$$

移項して整理すると次のようになる。

$$\ddot{\theta} + (-g + K)\theta = 0 \quad (6.5)$$

$K$  は人為的に設定できる定数だから、P 制御すると、ばね定数を人為的に変更できる。すなわち、P 制御は人工的なばねとして働く。固有方程式は、

$$s^2 + (-g + K) = 0 \quad (6.6)$$

となり、 $-g + K > 0$  となるようにゲイン  $K$  を調整すると、固有値を純虚数の対  $s = i\sqrt{-g + K}, -i\sqrt{-g + K}$  に調整できる。このとき振子は単振動する。以上、発散は抑制されたが、振子は静止しない。

#### 6.2.3 角度および角速度のフィードバック

そこで、角速度  $\dot{\theta}$  に比例した力でも台車を押してやる。これを微分制御または D 制御という。P 制御と D 制御を併せて PD 制御という。実際、 $K, L$  をゲインとして  $f(t) = K\theta + L\dot{\theta}$  という外力を台車に作用させると、

$$\ddot{\theta} - g\theta = -f(t) = -K\theta - L\dot{\theta} \quad (6.7)$$

移項して整理すると次のようになる。

$$\ddot{\theta} + L\dot{\theta} + (-g + K)\theta = 0 \quad (6.8)$$

$L$  は人為的に設定できる定数だから、D 制御すると減衰係数を人為的に変更できる。すなわち、D 制御

は人工的なダンパー（減衰器）として働く．固有方程式は，

$$s^2 + Ls + (-g + K) = 0 \quad (6.9)$$

となり，固有値は，

$$s = \frac{-L \pm \sqrt{L^2 - 4(-g + K)}}{2} \quad (6.10)$$

だから， $L > 0, L^2 - 4(-g + K) < 0$  となるように  $K, L$  を選べば，固有値を実部が負の共役複素数

$$s = \frac{-L \pm i\sqrt{|L^2 - 4(-g + K)|}}{2} \quad (6.11)$$

に調整できる．このときの振子は，減衰振動の条件にあるため，ちょっとずらしても減衰振動しながら上死点に向う．

以上，PD 制御法によって，ロボットは安定に倒立する．このようなフィードバック制御法を基礎に，ロボットの打ち上げや 2 足歩行が実現されている．

### 6.3 制御系の設計 — ゲイン調整

これまでの机上検討で，ロボットを安定に倒立させる方式は確定したが，具体的な試作に移るには，ゲイン  $K, L$  の具体値を決める必要がある．

第 2 回レポート課題 適当な  $K, L$  の値を式 (6.9) に代入し，そのときの固有値を Octave で求めよ．

- 1) 固有値を見ながら  $K, L$  を調整し，安定に倒立する  $K, L$  の組合せの中で，a) 振動的なもの，b) 非振動的なもの，を 1 つずつ探せ．
- 2) a), b) の固有値から予想されるロボットの動きを，それぞれ考察し，スケッチせよ．
- 3) プログラム「prog-1.m」を改造して，式 (6.9) の実際の振動波形を求め，スケッチと照合せよ．

$K = 1, L = 1$  のときの実行例を示す．まず，1) で固有値は，

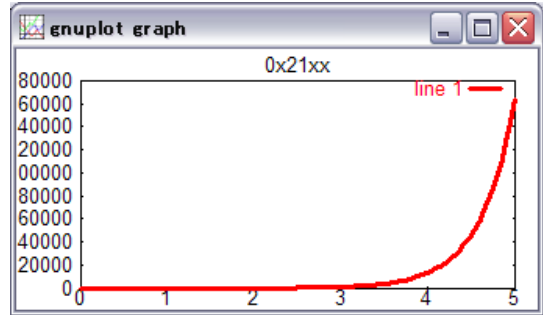
```
octave:1> g=9.8;
octave:2> K=1; L=1;
octave:3> roots([1,L,(-g+K)])
ans =
   -3.5083
    2.5083
```

となる．実部に正 2.5084 があり，虚部が無い．表 4.1 p.14 と照合すると「発散・非振動」なので選定失敗．倒れ角  $\theta$  が発散するから，ロボットは倒立状態  $\theta = 0$  を維持できない．

2) のスケッチは，各自分りやすく工夫せよ．3) を実行するには， $m, c, k$  の代りに  $g, K, L$  が使えるように，プログラム「prog-1.m」の 1, 2, 4 行目を書き換え，6 行目の運動方程式を書き換えればよい．

参考まで，不適切なゲインではあるが  $K = 1, L = 1$  のときの振動波形を示そう．見やすさのた

め，時間軸は 5 秒まで  $t = \text{linspace}(0, 5, 100)$  ; とした．

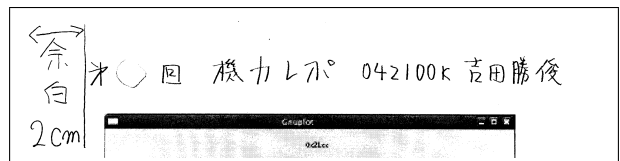


固有値による予測の通り，ロボットの倒れ角  $\theta$  が指数関数的に増大している．ちなみに， $\theta = 60000 \text{ rad} \approx 9549$  回転など， $\pi/2$  を超える倒れ角は，このロボットの運動としては非現実的だ．なぜなら，振子が水平を超えると，台車を横から押しても，振子を立てる力にならない．このような計算上の大きな倒れ角は，運動方程式に本来ある三角関数  $\sin \theta, \cos \theta$  を  $\theta$  や 1 で近似したための幻の解である．近似しない運動方程式は，12 講で導く．

このレポート課題は，制御系設計の最も原始的な一例になっている．より高度な制御理論を学ぶと，このようなゲイン調整を試行錯誤なしにできるようになるが，結果オーライの開発現場では，この課題のように試行錯誤で調整することも少なくない．

### レポート提出方法

- A4 一枚以内（両面使用可）．左余白 2cm 以上．
- 上に「第 X 回機力レポ 学籍番号 氏名」と明記．
- ✂切：次回授業日の前日まで．吉田准教授室 Box．次々回授業日からは受けとらない．
- 学籍番号なきグラフは盗作と見なす．



注意 レポートは，新入生を読者に想定して書くこと．グラフの軸にラベルが無かったり，解説や考察が無いと，何をどう理解すればよいか分らない．レポートとは，ある出来事の体験者が，未体験者に向けて書く文書．

# 第7講

## 共振現象のモデル

外力のない振動を自由振動といった。外力を加えたときの振動を強制振動という。強制振動特有の現象に共振現象がある。強制振動の理論=共振現象の理論である。本日は共振現象を観察する。

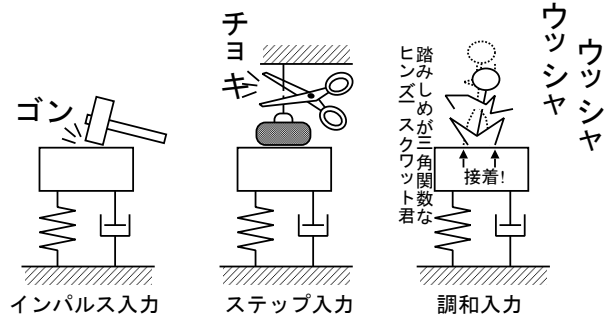


図 7.2: 外力  $f(t)$  の種類

### 7.1 強制振動のモデル

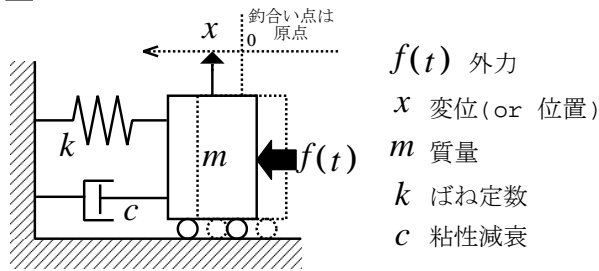


図 7.1: 強制振動系

#### 7.1.1 強制振動系

図 3.1 p.10 の自由振動系に、外力  $f(t)$  を付け加えたものを、強制振動系という。

図 7.1 のモデルにおいて、外力  $f(t)$  の発生要因は問われない。風だろうが、地震だろうが、それこそ超能力だろうが、単位 [N] で測れさえすれば問題ない。強制振動系の運動方程式は、

$$m\ddot{x} + c\dot{x} + kx = f(t) \quad (7.1)$$

となる。力の等式である運動方程式の右辺に  $f(t)$  を付加した、ただそれだけである。

#### 7.1.2 外力 $f(t)$ の種類

現実の外力はもっと多様だが、図 7.2 はテスト入力と総称され、振動試験で条件を揃えるのに使われる。外力を入力、振動系の動きを応答と呼ぶ場合がある<sup>1)</sup>。以後区別せずに用いる。

**インパルス入力(衝撃力)** 「ハンマーでゴン」のモデル化。インパルスとは衝撃力のことである。インパルスの数学モデルは、ちょっと乱暴に書くと

$$f(t) = \delta(t) = \begin{cases} 0 & (t \neq 0) \\ \infty & (t = 0) \end{cases} \quad (7.2)$$

である。 $\delta(t)$  をディラックのデルタ関数または単位インパルスと呼ぶ。 $\delta(t)$  とは、力積 1Ns の衝撃力の理想化である。力積(面積)を 1 に固定したまま、外力の作用時間を 0 に絞ったものである<sup>2)</sup>。

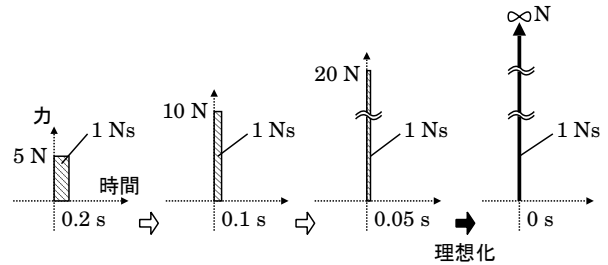


図 7.3: 単位インパルス関数

**ステップ入力** 「チョコキ、ズシッ」のモデル化。ステップとは階段のことである。これの数学モデルは

$$f(t) = u(t) = \begin{cases} 0 & (t < 0) \\ 1 & (t \geq 0) \end{cases} \quad (7.3)$$

である。 $u(t)$  を単位ステップ関数と呼ぶ。0 だった外力が、突然 1 まで立ち上がる。

**調和入力** 図 7.2 の調和入力は周期外力のつもりである<sup>3)</sup>。調和励振ともいう。式で書くと

$$f(t) = A \cos \omega t \quad (7.4)$$

である。ようは三角関数の外力である。正確には

$$f(t) = A \cos \omega t + B \sin \omega t \quad \text{or} \quad = A \cos(\omega t + \phi)$$

と書く。さもないと例えば  $f(t) = \sin \omega t$  を調和入力と呼べなくなる。

## 7.2 共振現象

強制振動に特有な、共振現象を観察しよう。

<sup>1)</sup>もともと制御の言葉だが、振動屋と制御屋は 1 人 2 役なことも多く、なにげに区別されない。教科書等でもそうである。

<sup>2)</sup>太さ 0 の線が物理的に実現不可能なのと同様、単位インパルスも実現不可能。その反面、計算が大幅に単純化される。

<sup>3)</sup>sin, cos を調和関数と呼ぶ。これが語源。

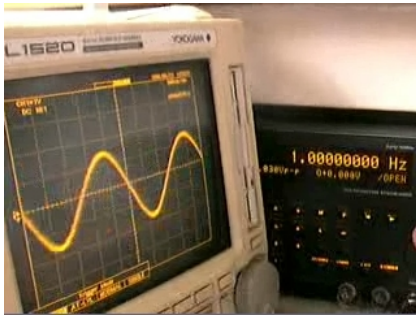
□ 7.2.1 モータ-振り子系

- 1) モータに振り子を接続する .
- 2) 交流電圧をかける .
  - (a) 電圧一定 0.2 V
  - (b) 周波数 1 ~ 2.5 Hz
- 3) 振り子の振れ幅 (振動振幅) を観察する .



[再生]

電圧一定で周波数を変化



[再生]

自由振動 電源を切っておくと, 振り子は減衰振動する .



[再生]

強制振動 (前半) 周波数を, 1Hz から徐々に増加させる .



[再生]

実習 7.1 さらに周波数を増加させると, 振り子の動き方はどう変化していくか? 絵, 言葉, グラフなどを使って机上検討せよ .

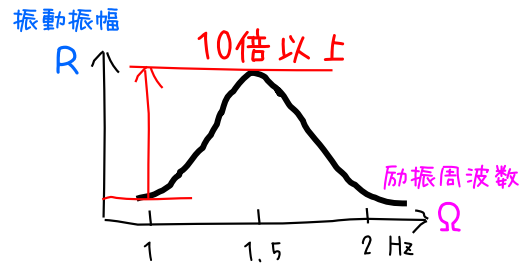
強制振動 (後半) さらに周波数を増加させる .

<sup>4)</sup> 振動波形  $x(t)$  の高さに相当する  $\max_t x(t) - \min_t x(t)$  を, **peak to peak**(ピーク・トゥ・ピーク) という . P-P などと略記する .  
<sup>5)</sup> 片振り振幅という .

実習 7.2 1Hz から 2.5Hz までの, 振り子の動き方を詳細に観察し, グラフで記録せよ . [再生]

□ 7.2.2 共振曲線 — 共振の図示法

振り子の振動振幅は  $\Omega = 1.5$  付近で最大となり . その前後で減少していた . このような振幅の増減は, 横軸が  $\Omega$ , 縦軸が振れ幅のグラフによって, 簡潔に図示できる . このようなグラフを共振曲線という .



このように, 入力する交流電圧が一定であるにもかかわらず, 交流の周波数によって, 振り子の振れ角は 10 倍以上変化してしまう .

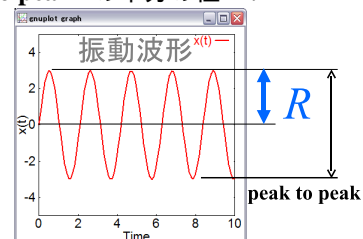
このような, 入力周波数によって振動振幅が変化する現象を, 共振現象という . 詳細な対応関係は省くが, 以上の振り子の共振現象は, 調和入力 ( $\sin$  や  $\cos$  の外力) を受ける強制振動系,

$$m\ddot{x} + c\dot{x} + kx = f(t) = A \cos \omega t$$

によって, 上手に真似することができる .

□ 7.2.3 共振曲線の数値計算

振動波形をたくさん描いて, そこから共振曲線を測定する Octave プログラムを示す . 調和入力  $f(t) = \cos \omega t$  の  $\omega$  の値を変えながら, そのときの振動波形の **peak to peak**<sup>4)</sup> の半分<sup>5)</sup> :





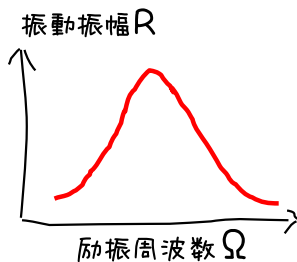
を測り、 $\omega$  の関数としてプロットしている。

プログラム「reso-1.m」

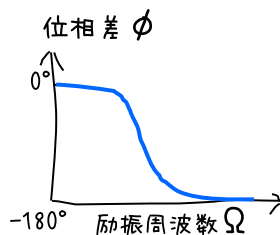
```
global om;
function dx = model(x, t)
    global om;
    m=1.0; c=0.2; k=1.0;
    dx(1)=x(2);
    dx(2)=- (c/m)*x(2)-(k/m)*x(1)+cos(om*t);
endfunction
freq = linspace(0.2,1.6,29);
__gnuplot_set__ yrange[-5:5];
__gnuplot_set__ xlabel "Time t";
__gnuplot_set__ ylabel "Amplitude x"
x0 = [0; 0.1];
t = linspace(0, 100, 300);
for i = 1:29
    om = freq(i); f = cos(om*t);
    x = lsode("model", x0, t);
    title(sprintf("(om = %.3f)", om));
    plot(t,f,";f(t);",t,x(:,1),";x(t);")
    sleep(0.1);
    xmax(i) = max(x(250:300,1));
    # 振動波形が落ち着いたあとの P-P
endfor
sleep(2);
__gnuplot_set__ yrange[0:5];
__gnuplot_set__ xlabel "Frequency om"
title("Response Curve");
plot(freq, xmax);
```

実習 7.3  $x(t)$  の振幅の増減をスケッチせよ。また、 $f(t)$  と  $x(t)$  のタイミングのずれもスケッチせよ。

共振曲線と位相差 振幅の増減のグラフを、共振曲線と呼んだ。



これに加えて、外力  $f(t)$  と振動  $x(t)$  のタイミングのずれを位相差という。



結論からいうと、「外力による振動」の特徴は、(共振曲線, 位相差) のペアで完全に表わせる。次講で詳

<sup>6)</sup> 唯一の例外として、減衰 0 のときは共振点で「外力による振動」が発散する。

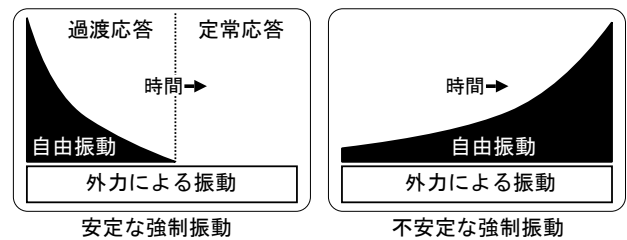
しく学ぶ。ちなみに、自由振動の特徴は固有値の(実部, 虚部)のペアで表わせた。ペアって本質的かも。

### 7.3 過渡応答と定常応答

強制振動の振動波形はどうなるのか？

#### 7.3.1 強制振動=自由振動+外力による振動

強制振動系は、外力  $f(t)$  の存在以外は自由振動系と同じである。ならば自由振動系のなごりが残りそうだが、実際そうである。模式図で示す。



外力による振動は振幅一定なので、強制振動の安定性は、自由振動成分の発散/収束で決まる。

- 1) 安定な強制振動 = 自由振動成分 ( $\rightarrow 0$ ) + 外力による振動成分 (一定)
- 2) 不安定な強制振動 = 自由振動成分 ( $\rightarrow \infty$ ) + 外力による振動成分 (一定)

ということは、興味深い算法が判明する。結局のところ、強制振動系の安定判別に、外力  $f(t)$  の情報は必要ないのである<sup>6)</sup>。

算法 6 (強制振動系の安定判別) 強制振動系の安定性は、 $f(t) = 0$  とおいて得られる自由振動系、

$$m\ddot{x} + c\dot{x} + kx = 0$$

の固有値で判別できる。

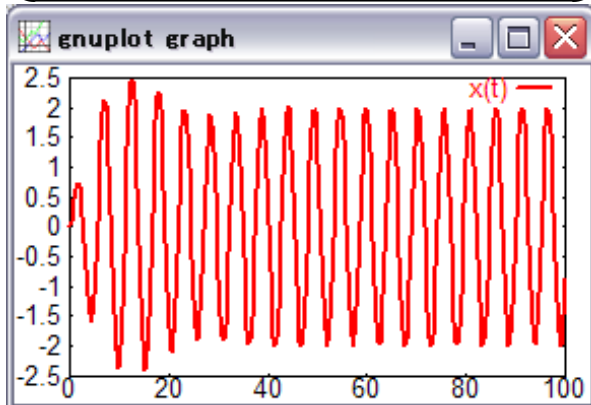
以上、自由振動成分は固有値で理解できるので、外力による振動成分をこれから解明すれば、強制振動の全貌が明らかになる。この「外力による振動成分」を、定常応答と呼ぶ。これに対して、自由振動成分が減衰しきる前の強制振動を過渡応答と呼ぶ。

#### 7.3.2 定常応答 = 落ち着いたあとの振動

定常応答を実験的に観察するには、自由振動成分が減衰しきった後の振動波形を見ればよい。プログラム「reso-2.m」を用いて、調和励振  $f(t) = \cos \omega t$  を受ける強制振動系の定常応答を観察せよ。

プログラム「reso-2.m」

```
global om;
function dx = model(x, t)
    global om;
    m=1.0; c=0.2; k=1.0;
    dx(1)=x(2);
    dx(2)=-(c/m)*x(2)-(k/m)*x(1)+cos(om*t);
endfunction
om=1.2;
x0 = [0; 0.1];
t = linspace(0, 100, 300);
x = lsode("model", x0, t);
plot(t,x(:,1),"ix(t);")
```



実習 7.4 得られたグラフについて、過渡応答の部分と定常応答の部分を違う色で塗り分けよ。

実習 7.5 もういちどプログラム「reso-1.m」を実行し、過渡応答と定常応答を観察せよ。

# 第 8 講

## 周波数応答

**(共振現象の性質)**

時間  $t$  の経過  $\rightarrow$  過渡応答  $\rightarrow$  定常応答  
 外力周波数  $\Omega$  の増加  $\rightarrow$  振幅の増減, 位相の遅れ.

$f(t) = \cos \Omega t$   
 $\Omega = 0.3$  (低)  
 振幅小  
 遅れ小 ( $\approx$  同相)

Mechanical Dynamics: 第 8 講 周波数応答 - 1 - 2

時間  $t$  の経過  $\rightarrow$  過渡応答  $\rightarrow$  定常応答  
 外力周波数  $\Omega$  の増加  $\rightarrow$  振幅の増減, 位相の遅れ.

$f(t) = \cos \Omega t$   
 $\Omega = 1.0$  (中)  
 振幅大  
 位相遅れ中

Mechanical Dynamics: 第 8 講 周波数応答 - 2 - 3

時間  $t$  の経過  $\rightarrow$  過渡応答  $\rightarrow$  定常応答  
 外力周波数  $\Omega$  の増加  $\rightarrow$  振幅の増減, 位相の遅れ.

$f(t) = \cos \Omega t$   
 $\Omega = 1.2$  (高)  
 振幅小  
 遅れ大 ( $\approx$  逆相)

Mechanical Dynamics: 第 8 講 周波数応答 - 3 - 4

**(共振現象のグラフ)**

- 振動振幅の増減を, 共振曲線で表わす.
- 外力と振動のずれを, 位相差で表わす.

振動振幅  $R$

共振周波数  $\Omega$

位相差  $\phi$

共振周波数  $\Omega$

$\rightarrow$  共振曲線はどんな式? 位相差はどんな式?

Mechanical Dynamics: 第 8 講 周波数応答 - 4 - 5

**本日の学習目標**

- 過渡応答・定常応答の再定義 (理論計算用)
  - 過渡応答  $\alpha(t) \equiv$  自由振動成分.  $\rightarrow$  固有値  
 ※実験的には  $\alpha(t) + \beta(t)$  を過渡応答と言う
  - 定常応答  $\beta(t) \equiv$  外力による振動.  $\rightarrow$  周波数応答
- 定常応答の手計算
  - 強制振動系:  $\ddot{x} + 2\zeta\dot{x} + x = A \cos \omega t$  (無次元化  $t \mapsto \omega_n t$  した)
  - 定常応答の解  $x(t) = \beta(t)$  をハーモニックバランスで手計算.
- 振幅比  $K(\omega)$  と位相差  $\phi(\omega)$  ※ペアを周波数応答という.
  - どちらも, 定常応答の式  $\beta(t)$  を整理すると求まる.
  - 共振曲線  $\equiv$  定数  $\times K(\omega)$ .

Mechanical Dynamics: 第 8 講 周波数応答 - 5 - 6

**過渡応答と定常応答**

**法則** 強制振動  $x(t)$  は, 2つの部分に分解できる.

$$x(t) = \underbrace{\alpha(t)}_{\text{自由振動の解}} + \underbrace{\beta(t)}_{f(t)\text{の影響}}$$

■ 2通りの定義:

- 理論:  $x(t) = \alpha(t) + \beta(t)$  過渡応答 定常応答 定常応答は共通!
- 実験:  $x(t) = \alpha(t) + \beta(t)$  収束 過渡応答 定常応答  $\beta(t)$  ( $t \rightarrow \infty$ )  
 $\rightarrow \alpha(t)$  は固有値で分かる.  $\beta(t)$  はどうする?

Mechanical Dynamics: 第 8 講 周波数応答 - 6 - 7

**強制振動解析の目標**

■ 強制振動  $x(t) = \alpha(t) + \beta(t)$  過渡応答 定常応答

- 過渡応答の新しい理論は不要. 固有値でOK.
- 定常応答には, 新しい理論が必要!  
 $\rightarrow$  定常応答が分かる = 外力の影響が分かる = 強制振動が分かる!

☆ 目標: 外力の影響  $\beta(t)$  の理論を作るため,

1.  $\beta(t)$  を数式として求める.
2.  $\beta(t)$  の性質を数式で表わす.

Mechanical Dynamics: 第 8 講 周波数応答 - 7 - 8

**算法上の作戦**

**算法** 定常応答を調べるには, 調和入力を受ける振動系,

$$\ddot{x} + 2\zeta\dot{x} + x = A \cos \omega t$$

の公式だけ作ればよい.

- 定常な外力  $f(t)$  は, 周波数の異なる三角関数の和で書ける.
 
$$f(t) = A_1 \cos \omega_1 t + A_2 \cos \omega_2 t + \dots + A_n \cos \omega_n t$$
 $\rightarrow$  線形常微分方程式の線形性
- $\ddot{x} + 2\zeta\dot{x} + x = f(t)$  の定常応答  $\beta(t)$  は, 個別の定常応答の和,
 
$$\beta(t) = \beta_1(t) + \beta_2(t) + \dots + \beta_n(t)$$
 で書ける.  $\beta_1(t)$  は  $\ddot{x} + 2\zeta\dot{x} + x = A_1 \cos \omega_1 t$  の定常応答.

Mechanical Dynamics: 第 8 講 周波数応答 - 8 - 9

### ハーモニックバランス法

■ 目標:  $\ddot{x} + 2\zeta\dot{x} + x = A \cos \omega t$  の定常応答  $\beta(t)$  を求める.

**法則** 入力が周波数  $\omega$  の三角関数なら, 出力も周波数  $\omega$  の三角関数となる!

■ 作戦:

1.  $\beta(t) = a_1 \cos \omega t + a_2 \sin \omega t$  と仮定  
 $\hookrightarrow$  周波数  $\omega$  の調和関数 (sin, cos) を網羅.
2. 係数  $a_1, a_2$  を調整して,  $\ddot{x} + 2\zeta\dot{x} + x = A \cos \omega t$  の解にする.

### 手順 1

仮定  $x(t) = \beta(t) = a_1 \cos \omega t + a_2 \sin \omega t$  を, 運動方程式,

$$\ddot{x} + 2\zeta\dot{x} + x = A \cos \omega t$$

へ代入し,

$$(\dots) \times \cos \omega t + (\dots) \times \sin \omega t = 0$$

の形式へ整理せよ.

### 計算例 1

$$x(t) = a_1 \cos \omega t + a_2 \sin \omega t \quad \text{仮定}$$

$$\Rightarrow \dot{x}(t) = -a_1\omega \sin \omega t + a_2\omega \cos \omega t \quad \text{sin, cos の微分}$$

$$\Rightarrow \ddot{x}(t) = -a_1\omega^2 \cos \omega t - a_2\omega^2 \sin \omega t // \quad \text{sin, cos の微分}$$

運動方程式に代入すると,

$$\underbrace{-a_1\omega^2 \cos \omega t - a_2\omega^2 \sin \omega t}_{\ddot{x}(t)} + 2\zeta \underbrace{-a_1\omega \sin \omega t + a_2\omega \cos \omega t}_{\dot{x}(t)} + \underbrace{a_1 \cos \omega t + a_2 \sin \omega t}_{x(t)} = A \cos \omega t$$

$$\therefore (-a_1\omega^2 + 2\zeta a_2\omega + a_1 - A) \cos \omega t + (-a_2\omega^2 - 2\zeta a_1\omega + a_2) \sin \omega t = 0$$

$$\Leftrightarrow \{(1 - \omega^2)a_1 + 2\zeta\omega a_2 - A\} \cos \omega t + \{-2\zeta\omega a_1 + (1 - \omega^2)a_2\} \sin \omega t = 0 //$$

### 手順 2

任意の時刻  $t$  について,

$$(\dots) \times \cos \omega t + (\dots) \times \sin \omega t = 0$$

となるような,  $a_1, a_2$  を求めよ.

ヒント: 2連立方程式になる. 行列の形式,

$$\begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} ? \\ ? \end{pmatrix}$$

で整理するとよい.

### 計算例 2

恒等的に0になるのは,  $\sin \omega t$  と  $\cos \omega t$  の係数が0のとき.

$$\begin{cases} (1 - \omega^2)a_1 + 2\zeta\omega a_2 - A = 0 \\ -2\zeta\omega a_1 + (1 - \omega^2)a_2 = 0 \end{cases}$$

行列にして計算すると,

$$\begin{pmatrix} (1 - \omega^2) & 2\zeta\omega \\ -2\zeta\omega & (1 - \omega^2) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} A \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} (1 - \omega^2) & -2\zeta\omega \\ 2\zeta\omega & (1 - \omega^2) \end{pmatrix} \begin{pmatrix} A \\ 0 \end{pmatrix}$$

$$= \frac{A}{\Delta} \begin{pmatrix} 1 - \omega^2 \\ 2\zeta\omega \end{pmatrix} //$$

ただし,  $\Delta = (1 - \omega^2)^2 + (2\zeta\omega)^2$  は行列式.

### 手順 3

最初に仮定した  $\beta(t)$  に  $a_1, a_2$  を代入し, 定常応答  $\beta(t)$  を求めよ.

### 計算例 3

$$x(t) = \beta(t) = a_1 \cos \omega t + a_2 \sin \omega t$$

$$= \frac{A}{\Delta} (1 - \omega^2) \cos \omega t + \frac{A}{\Delta} 2\zeta\omega \sin \omega t$$

$$\therefore \beta(t) = A \frac{(1 - \omega^2) \cos \omega t + 2\zeta\omega \sin \omega t}{\Delta}$$

$$\ast \Delta = (1 - \omega^2)^2 + (2\zeta\omega)^2$$

### 手順 4

**算法**  $X \cos \theta + Y \sin \theta = \sqrt{X^2 + Y^2} \cos(\theta - \tan^{-1} \frac{Y}{X})$

上記の算法により, 得られた定常応答:

$$\beta(t) = A \cdot \frac{(1 - \omega^2) \cos \omega t + 2\zeta\omega \sin \omega t}{\Delta}$$

は次の形式に整理できる.

$$\beta(t) = A \cdot K \cos(\omega t + \phi)$$

具体的に  $K$  と  $\phi$  を求めよ.

$\ast K$  を振幅比,  $\phi$  を位相差という.

### 計算例4

「A」以降の部分に**算法**を使うと、

$$X = \frac{(1 - \omega^2)}{\Delta}, \quad Y = \frac{2\zeta\omega}{\Delta}$$

より、

$$K = \sqrt{X^2 + Y^2} = \sqrt{\frac{(1 - \omega^2)^2 + (2\zeta\omega)^2}{\Delta^2}} = \sqrt{\frac{\Delta}{\Delta^2}} = \sqrt{\frac{1}{\Delta}}$$

$$= \frac{1}{\sqrt{(1 - \omega^2)^2 + (2\zeta\omega)^2}}$$

$$\phi = -\tan^{-1} \frac{Y}{X} = -\tan^{-1} \frac{2\zeta\omega}{1 - \omega^2}$$

※  $K$  が共振曲線の式。  $\phi$  が位相差の式！

### 定常応答 $\beta(t)$ の正体

#### ■ 計算結果の意味：

■  $f(t) = A \cos(\omega t)$  という入力波形が、振動系の作用で

$\beta(t) = K \times A \cos(\omega t + \phi)$  という波形に加工された。

$$\begin{cases} K = K(\omega) = \frac{1}{\sqrt{(1 - \omega^2)^2 + (2\zeta\omega)^2}} \\ \phi = \phi(\omega) = -\tan^{-1} \frac{2\zeta\omega}{1 - \omega^2} \end{cases}$$

**法則** 定常応答  $\beta(t)$  の振動波形は、入力波形  $A \cos \omega t$  を  $K$  倍し、時間方向に  $-\phi$  平行移動したものに等しい。

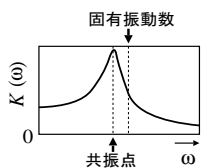
$K$  を**振幅比**、 $\phi$  を**位相差**と呼ぶ<sup>†)</sup>。

<sup>†)</sup>制御の分野では、振幅比を**振幅特性**、位相差を**位相特性**と呼ぶ。

### 振幅比 (別名:振幅特性)

**誤解に注意!**

- 「外力の振動数」 = 「自由振動の振動数」のとき振幅最大
- 「外力の振動数」 = 「固有振動数」のとき振幅最大



- 外力:  $f(t) = A \cdot \cos \omega t$  に対して、共振曲線  $= A \cdot K(\omega)$ .
- 共振点 < 固有振動数.
- 共振点 < 自由振動数.
- ※減衰0のときは全て一致.

### 共振振動数

#### ■ 共振点 = $K(\omega)$ のピーク：

$$\Leftrightarrow \frac{dK(\omega)}{d\omega} = 0 \text{ より, } \omega_p = \sqrt{1 - 2\zeta^2}$$

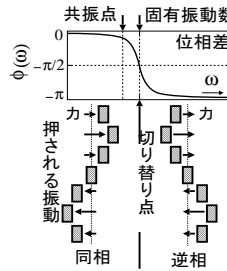
$1 - 2\zeta^2 < 0 \Rightarrow \omega_p$  は虚数 ∴ 共振は起きない。

#### ■ 実際の大小関係：(スケール変換 $\omega_n = 1$ に注意)

$$\omega_n \sqrt{1 - 2\zeta^2} < \omega_n \sqrt{1 - \zeta^2} < \omega_n$$

※ 減衰を十分大きくすれば共振現象を消せる。こんなところまで計算できる。つまり設計できる。

### 位相差 (別名:位相特性)



※ 振動波形を90度回転し、下向きに描いた。

- 固有振動数の左では押される振動(同相)
- 右では引っ張られる振動(逆相)
- 固有振動数で同相と逆相が切り替わる。

### 周波数応答

#### ■ 振幅比 $K$ と、位相差 $\phi$ を2つ並べて周波数応答という。

☆ 振動系は、入力  $A \cos \omega t$  を  $K \times A \cos(\omega t + \phi)$  へ加工した。

$$f(t) = A \cos(\omega t) \xrightarrow{\text{振動系}} \beta(t) = K \times A \cos(\omega t + \phi)$$

振動系がしたことは、 $K$  と  $\phi$  以外にない。

⇨  $K$  と  $\phi$  に、強制振動系の特徴が全て出る。

#### ■ ペア ( $K, \phi$ ) で一人前。片方だけでは意味がない。

- $K$  を無視すると、振幅の増減が分らない。
- $\phi$  を無視すると、タイミングのずれが分らない。

### ボード線図

周波数応答(振幅比と位相差)を**対数グラフ**に描いたものを**ボード線図**という。

1.  $\omega = \Omega/\omega_n$  の目盛は  $\log_{10} \omega$ .
2. 振幅比  $K(\omega)$  の目盛は  $20 \log_{10} K(\omega)$ . 単位はdB (デシベル).
3. 位相差  $\phi(\omega)$  はそのまま.

※ 計測器メーカーなどにボード線図を請求すると、この規格のものが送られてくる。

# 第 9 講

## 伝達関数

### 学習目標

- 「オイラーの公式」が使える。
  - 複素数の極座標表示は、指数関数と同じ性質を持つ。
- 複素数によるハーモニックバランス(複素HB)ができる。
  - 運動方程式に複素数を代入できる。
  - 計算結果から、振幅比 $K(\omega)$ 、位相差 $\phi(\omega)$ が抽出できる。
- 伝達関数の作られ方を説明できる。
  - 複素HBの右辺を、周波数伝達関数という。
  - その関数形を、伝達関数という。

☆ 伝達関数は、自動制御理論のエッセンス！

### オイラーの公式

**算法** 複素数の極座標表示は、指数関数と同じ計算法則を持つ。  

$$e^{i\theta} = \cos \theta + i \sin \theta$$

(証明)

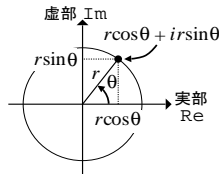
- 微分法則:  $(e^{i\theta})' = -\sin \theta + i \cos \theta = i(\cos \theta + i \sin \theta) = i e^{i\theta} //$
- 指数の加法:  $e^{i(\theta+\phi)} = \cos(\theta+\phi) + i \sin(\theta+\phi)$   
 $= (\cos \theta \cos \phi - \sin \theta \sin \phi) + i(\cos \theta \sin \phi + \sin \theta \cos \phi)$   
 $= (\cos \theta + i \sin \theta)(\cos \phi + i \sin \phi) = e^{i\theta} e^{i\phi} //$
- 算法が共通だから同一視して、 $r e^{i\theta}$ と書いてしまう。  
 ※ 複素数の計算を指数関数の計算にできる。

### 大学レベルの複素数

- 四則演算, 微分積分
    - 虚数 $i$ を定数と見なして、四則演算, 微分積分する。
    - 最後に、 $i^2 = -1$ を代入する。
- 例)  $x^3 + i \sin ix$ を $x$ で微分すると、 $3x^2 + i^2 \cos ix = 3x^2 - \cos ix$ .

#### 複素数の極座標表示:

- 平面の点 $x$ の極座標表示  
 $x = (r \cos \theta, r \sin \theta)$ .
  - 複素数 $z$ の極座標表示  
 $z = r \cos \theta + i r \sin \theta$ .
- (実, 虚)を $(x, y)$ と見て極座標表示!



### まとめ — 表示形式の変更 —

$$r e^{i\theta} \Rightarrow a + i b$$

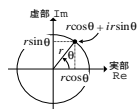
$$r e^{i\theta} = r \cos \theta + i r \sin \theta \quad (\text{オイラーの公式})$$

$$a + i b \Rightarrow r e^{i\theta}$$

$$a + i b = \sqrt{a^2 + b^2} e^{i \tan^{-1} \frac{b}{a}}$$

例えば、 $1 + 2i = |1 + 2i| e^{i \angle(1+2i)} = \sqrt{5} e^{i \tan^{-1} 2}$ .

### 絶対値と偏角



- $r = |r \cos \theta + i r \sin \theta|$ を絶対値という。
- $\theta = \angle(r \cos \theta + i r \sin \theta)$ を偏角という。

**算法** 複素数の絶対値と偏角:

$$|a + i b| = \sqrt{a^2 + b^2}, \quad \angle(a + i b) = \tan^{-1} \frac{b}{a}$$

※  $\tan^{-1}$ は $\tan$ の逆関数。  $y = \tan x \Rightarrow x = \tan^{-1} y$ .

### 運動方程式への複素数の代入?

**算法** 線形振動系に複素数を代入しても、  
**実部と虚部は混じらない**

(証明)

- $x = x_R + i x_I, f = f_R + i f_I$ を $\ddot{x} + a \dot{x} + b x = f$ へ代入。(iは定数)  
 $(\ddot{x}_R + i \ddot{x}_I) + a(\dot{x}_R + i \dot{x}_I) + b(x_R + i x_I) = f_R + i f_I$   
 $\Leftrightarrow (\ddot{x}_R + a \dot{x}_R + b x_R) + i(\ddot{x}_I + a \dot{x}_I + b x_I) = f_R + i f_I$   
 左辺と右辺は同じ複素数だから、座標が一致する。  
 $\Rightarrow$  独立な2連立方程式を得る。  $\begin{cases} \ddot{x}_R + a \dot{x}_R + b x_R = f_R \\ \ddot{x}_I + a \dot{x}_I + b x_I = f_I \end{cases}$

### 例題

複素数  $z = 1 + 2i$  の絶対値  $|z|$  と、偏角  $\angle z$  を求めよ。

三平方の定理より、 $|z| = r = \sqrt{1^2 + 2^2} = \sqrt{5}$ .  
 $\tan \theta = \frac{2}{1}$  より、 $\angle z = \theta = \tan^{-1} 2 \approx 1.1071$  (値は関数電卓)  
 $\therefore |1 + 2i| = \sqrt{5}, \quad \angle[1 + 2i] = \tan^{-1} 2$ .

### 実習

複素数  $z = 2 + i$  の絶対値  $|z|$  と、偏角  $\angle z$  を求めよ。

※ 複素平面に描くと混乱しない。

### 複素数によるハーモニックバランス

#### 画期的なアイデア:

1. 励振力を  $f(t) = e^{i\omega t} = \cos \omega t + i \sin \omega t$  とおく。
  2. 定常解を  $\beta(t) = K e^{i(\omega t + \phi)}$   
 $= K \cos(\omega t + \phi) + i K \sin(\omega t + \phi)$  と仮定する。
  3. こうしても、虚部と実部は混じらない!
- $\Rightarrow$  同時に、2つのハーモニックバランス

	与える入力	仮定する応答
実部では	$\cos \omega t$	$K \cos(\omega t + \phi)$
虚部では	$\sin \omega t$	$K \sin(\omega t + \phi)$

を解くのと同一ことになる。

### 手順1

運動方程式:  $\ddot{x} + 2\zeta\dot{x} + x = f(t)$

に,  $x(t) = \beta(t) := Ke^{i(\omega t + \phi)}$ ,  $f(t) := e^{i\omega t}$  を代入し,

$$\frac{Ke^{i(\omega t + \phi)}}{e^{i\omega t}} = \frac{1}{(\dots) + i(\dots)}$$

という形式へ整理せよ.

### 計算例1

運動方程式:  $\ddot{x} + 2\zeta\dot{x} + x = f(t)$  に,  $f(t) = e^{i\omega t}$  と,

$$x(t) = Ke^{i(\omega t + \phi)}$$

$$\Rightarrow \dot{x}(t) = K(i\omega)e^{i(\omega t + \phi)} \quad \text{変数は } t \text{ のみ}$$

$$\Rightarrow \ddot{x}(t) = K(i\omega)^2 e^{i(\omega t + \phi)} \quad \text{変数は } t \text{ のみ}$$

を代入すると,

$$K(i\omega)^2 e^{i(\omega t + \phi)} + 2\zeta K(i\omega)e^{i(\omega t + \phi)} + Ke^{i(\omega t + \phi)} = e^{i\omega t}$$

$$\Rightarrow Ke^{i(\omega t + \phi)} \{ (i\omega)^2 + 2\zeta(i\omega) + 1 \} = e^{i\omega t}$$

$$\Rightarrow \frac{Ke^{i(\omega t + \phi)}}{e^{i\omega t}} = \frac{1}{(i\omega)^2 + 2\zeta(i\omega) + 1} = \frac{1}{(1 - \omega^2) + i(2\zeta\omega)} //$$

### 手順2

1. 指数関数の算法を用いて, 左辺を通分せよ.

2. 右辺分母の表示形式を変更して,  $\frac{1}{(\dots)e^{i(\dots)}}$  とせよ.

### 計算例2

$$\frac{Ke^{i(\omega t + \phi)}}{e^{i\omega t}} = \frac{K e^{i\omega t} e^{i\phi}}{e^{i\omega t}} = K e^{i\phi} //$$

$$(\text{分母}) = \underbrace{(1 - \omega^2)}_a + i \underbrace{(2\zeta\omega)}_b \quad \left( = \sqrt{a^2 + b^2} e^{i \tan^{-1} \frac{b}{a}} \right)$$

$$= \sqrt{(1 - \omega^2)^2 + (2\zeta\omega)^2} e^{i \tan^{-1} \frac{2\zeta\omega}{1 - \omega^2}}$$

$$\therefore K e^{i\phi} = \frac{1}{(\text{分母})}$$

$$= \frac{1}{\sqrt{(1 - \omega^2)^2 + (2\zeta\omega)^2}} e^{i \left( -\tan^{-1} \frac{2\zeta\omega}{1 - \omega^2} \right)} //$$

### 手順3

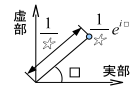
このとき両辺は,

$$Ke^{i\phi} = \frac{1}{\sqrt{a^2 + b^2}} e^{i\phi}$$

のような形で等値できる. 両辺を比較して,

$$K = \frac{1}{\sqrt{a^2 + b^2}} = |\text{右辺}|, \quad \phi = \phi = \angle \text{右辺}$$

を取り出せ.



■ 周波数応答が得られる.

$$\begin{cases} K(\omega) = \frac{1}{\sqrt{(1 - \omega^2)^2 + (2\zeta\omega)^2}} \\ \phi(\omega) = -\tan^{-1} \frac{2\zeta\omega}{1 - \omega^2} \end{cases}$$

※ 振幅比  $K$  は, 文字通り, 入力振幅 1 と出力振幅  $K$  の比!

### 周波数伝達関数

■ 三角関数を, 虚数  $i$  を含む指数関数で表示したら,

⇨ ハーモニックバランスと同じ結果が得られた.

$$\underbrace{K(\omega)}_{\text{振幅比}} e^{i\phi(\omega)} = \frac{1}{\underbrace{(\omega^2 + i2\zeta\omega + 1)}_{\text{振動系の情報}}}$$

右辺の複素数を **周波数伝達関数** という.

⇨ 振幅比  $K(\omega) = |\text{周波数伝達関数}| = \frac{1}{\sqrt{(1 - \omega^2)^2 + (2\zeta\omega)^2}}$

⇨ 位相差  $\phi(\omega) = \angle \text{周波数伝達関数} = -\tan^{-1} \frac{2\zeta\omega}{1 - \omega^2}$

### 伝達関数

※  $\ddot{x} + 2\zeta\dot{x} + x = f(t)$  に関する  $\underbrace{s^2 + 2\zeta s + 1 = 0}_{\text{固有方程式}}$   
固有方程式

■ 人類の発明 — 固有方程式を分母とする新たな関数!

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

⇨ 複素ハーモニックバランスの右辺は, この関数で,

$$G(i\omega) \equiv \frac{1}{(i\omega)^2 + 2\zeta(i\omega) + 1} = \frac{1}{-\omega^2 + i2\zeta\omega + 1}$$

と書いてしまう.

■ 前ページで  $G(i\omega)$  を, **周波数伝達関数** と呼んだ.

■ その関数形  $G(s)$  を, **伝達関数** という.

### まとめ

複素ハーモニックバランスの右辺の複素数:

$$Ke^{i\phi} = \boxed{\text{複素数}} \quad \left( \equiv G(i\omega) \right)$$

を, **周波数伝達関数** と呼び, これを  $G(i\omega)$  と書く. その関数形  $G(\cdot)$  を **伝達関数** という.

**算法** 定常応答の振幅比と位相差は, **周波数伝達関数**  $G(i\omega)$  の絶対値と偏角として計算できる.

$$K(\omega) = |G(i\omega)|, \quad \phi(\omega) = \angle G(i\omega)$$

$$\text{外力} \begin{matrix} \cos \omega t \\ \sin \omega t \end{matrix} \Rightarrow \boxed{\text{振動系}} \Rightarrow \begin{matrix} K \cos(\omega t + \phi) \\ K \sin(\omega t + \phi) \end{matrix}$$

# 第 10 講

## 伝達関数の使い方

運動方程式の固有値から自由振動が分かり、周波数応答から強制振動が分かった。運動方程式の代わりに伝達関数を出発点にしても、全く同様の解析ができる。

### 10.1 定常応答の復習

$$\ddot{x} + 2\zeta\dot{x} + x = f(t)$$

法則 6 強制振動  $x(t)$  は、2 つの部分に分解できる。

$$x(t) = \underbrace{\alpha(t)}_{\text{自由振動の解}} + \underbrace{\beta(t)}_{f(t) \text{ の影響}}$$

法則 7 外力  $f(t)$  が周波数  $\omega$  の三角関数なら、定常応答  $\beta(t)$  も周波数  $\omega$  の三角関数となる！

法則 8 調和入力  $f(t) = A \cos \omega t$  に対する定常応答  $\beta(t)$  の振動波形は、入力波形  $A \cos \omega t$  を  $K$  倍し、時間方向に  $-\phi$  平行移動したものに等しい。

$K$  を振幅比、 $\phi$  を位相差と呼ぶ<sup>1)</sup>。これらのペア  $(K, \phi)$  を周波数応答という。

算法 7 (絶対値と偏角)

$$|a + ib| = \sqrt{a^2 + b^2}, \quad \angle(a + ib) = \tan^{-1} \frac{b}{a}$$

$\tan^{-1}$  は  $\tan$  の逆関数。  $y = \tan x \Rightarrow x = \tan^{-1} y$  .

算法 8 (オイラーの公式) 複素数の極座標表示は、指数関数と同じ計算則を持つ。

$$e^{i\theta} = \cos \theta + i \sin \theta$$

算法 9 線形振動系に複素数を代入しても、実部と虚部は混じらない

運動方程式に  $f(t) = e^{i\omega t}$ ,  $x(t) = K e^{i(\omega t + \phi)}$  を代入して整理すると出てくる、右辺の複素数：

$$K e^{i\phi} = \overbrace{\frac{1}{(1 - \omega^2) + (2\zeta\omega)i}}^{\text{複素数}} \quad (\equiv G(i\omega))$$

を、周波数伝達関数と呼び、これを  $G(i\omega)$  と書く。その関数形  $G(\cdot)$  を伝達関数という。

<sup>1)</sup>制御分野では、振幅比を振幅特性、位相差を位相特性と呼ぶ。

算法 10 定常応答の振幅比と位相差は、周波数伝達関数  $G(i\omega)$  の絶対値と偏角として計算できる。

$$K(\omega) = |G(i\omega)|, \quad \phi(\omega) = \angle G(i\omega)$$

強制振動系  $\ddot{x} + 2\zeta\dot{x} + x = f$  について、

$$\overbrace{(\cdot)^2 + 2\zeta(\cdot) + 1}^{\text{固有方程式}} = 0 \quad \text{固有方程式}$$

固有多項式

↓ 固有多項式を分母へ

$$G(\cdot) = \frac{1}{(\cdot)^2 + 2\zeta(\cdot) + 1} \quad \text{伝達関数}$$

↓  $i\omega$  を代入

$$G(i\omega) = \frac{1}{(i\omega)^2 + 2\zeta(i\omega) + 1} \quad \text{周波数伝達関数}$$

$$= \frac{1}{(1 - \omega^2) + (2\zeta\omega)i}$$

という関係がある。

このように、伝達関数  $G(s)$  を介して、自由振動を表わす固有方程式 (算法 2 p.12) と、強制振動を表わす周波数伝達関数 (算法 10) が結ばれる。

### 10.2 伝達関数の作り方

以上は空想上の発見であるから、理由を述べるのは難しい。その代わりに、ある運動方程式が与えられたときに、誰がやっても同じ伝達関数にたどり着く算法を紹介しよう。ラプラス変換という。

#### 10.2.1 固有方程式の限界

自由振動系 ( $f(t) = 0$ ) の理論では、

$$\text{運動方程式: } m\ddot{x} + c\dot{x} + kx = 0$$

$$\downarrow x(t) = e^{st} \text{ の代入}$$

$$\text{固有方程式: } ms^2 + cs + k = 0$$

という手順で、微分方程式である運動方程式を、固有値  $s$  の方程式に変換して手間を省いた。

同じように、強制振動系 ( $f(t) \neq 0$ ) も固有値だけにして手間を省きたい。ところが、右辺の  $f(t)$  に  $e^{st}$  は代入できないし、

$$\text{運動方程式: } m\ddot{x} + c\dot{x} + kx = f(t)$$

$$\downarrow x(t) = e^{st} \text{ の代入?}$$

$$\text{固有方程式: } ms^2 + cs + k = ???$$



表 10.1: ラプラス変換表

元の時間関数	そのラプラス変換	元の時間関数	そのラプラス変換
1	$\frac{1}{s}$	$\dot{x}(t)$	$sX(s)$
$\cos \Omega t$	$\frac{s}{s^2 + \Omega^2}$	$\ddot{x}(t)$	$s^2 X(s)$
$\sin \Omega t$	$\frac{\Omega}{s^2 + \Omega^2}$	$\frac{d^n x}{dt^n}$	$s^n X(s)$
$e^{\pm \lambda t}$	$\frac{1}{s \pm \lambda}$	$a x(t)$ (定数倍)	$a X(s)$ (定数倍)
$x(t)$	$X(s)$	$x(t) + y(t)$ (和)	$X(s) + Y(s)$ (和)

この先どうしたらよいか分からない。

### □ 10.2.2 ラプラス変換

そこで考案されたのがラプラス変換である。表 10.1 p.31 にしたがって、時間  $t$  の関数を、固有値  $s$  の関数に置き換える操作を、ラプラス変換という。逆の置き換え操作を、ラプラス逆変換という。

表にないルールは、次の公式で自作できる。

$$p(t) \mapsto P(s) = \int_0^{\infty} p(t)e^{-st} dt \quad (10.1)$$

このような手計算には、Octave のような数値計算ソフトではなくて、Maxima という数式処理ソフトを使うと便利である<sup>2)</sup>。端末のプロンプトで、

```
[...]$ maxima(Enter)
```

とすれば起動できる。Maxima のプロンプト (%i1) に続けてコマンドを打ち込む。

```
Maxima 5.9.1 http://maxima.sourceforge... 中略
:
(%i1) assume(s>0);(Enter)          変数 s は正と仮定
(%o1) [s > 0]
(%i2) integrate(1*exp(-s*t),t,0,inf);(Enter)
                                          $\int_0^{\infty} 1 e^{-st} dt$  の実行
(%o2)
      1
      -
      s
(%i3) integrate(sin(0m*t)*exp(-s*t),t,0,inf);
                                          $\int_0^{\infty} \sin(\Omega t) e^{-st} dt$  の実行
(%o3)
      Om
      -----
      2      2
      s  + Om
(%i4) quit();(Enter)                  終了
[...]$                                端末にもどった
```

### □ 10.2.3 運動方程式のラプラス変換

**実習 10.1** 強制振動系  $\ddot{x} + 2\zeta\dot{x} + x = f(t)$  をラプラス変換せよ。

<sup>2)</sup><http://ja.wikipedia.org/wiki/Maxima>

表 10.1 によって置き換える(ラプラス変換すると、

$$s^2 X(s) + 2\zeta s X(s) + X(s) = F(s) \quad (*)$$

となり、固有値  $s$  だけにできた。これが固有値の世界から見た運動方程式となる。 $X(s)$  は解となる未知関数、 $F(s)$  は  $f(t)$  から定まる関数であり、例えば、

$$f(t) = A \cos \omega t \xrightarrow{\text{表 10.1}} F(s) = A \frac{s}{s^2 + \omega^2}$$

である。

$X(s)$  や  $F(s)$  は、 $s$  の多項式かその分数なので普通に四則演算できる。ゆえに、式 (\*) は、

$$(s^2 + 2\zeta s + 1)X(s) = F(s) \quad (**)$$

と整理できる。この式を使って、固有方程式との関係を見てみよう。自由振動系に対しては  $f(t) = 0$  だから、式 (10.1) より  $F(s) = 0$  となり、固有値の世界の自由振動系は、

$$(s^2 + 2\zeta s + 1)X(s) = 0$$

と書ける。未知関数  $X(s)$  は一般に  $X(s) \neq 0$  だから、 $s^2 + 2\zeta s + 1 = 0$  となり、固有方程式が得られる。

以上、ラプラス変換した運動方程式の外力を 0 にしたものは、固有方程式と同値になる。このように、運動方程式のラプラス変換は、固有方程式の自然な拡張になっている。

### □ 10.2.4 伝達関数

ここで、式 (\*\*) をもう少し整理して、入力  $F(s)$  と出力  $X(s)$  の比を作ると、

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{s^2 + 2\zeta s + 1} \quad (***)$$

となる。この式を運動方程式  $\ddot{x} + 2\zeta\dot{x} + x = f$  の伝達関数と呼んだ。以上の計算を一般化して、次のような定義が考案された。

算法 11 (伝達関数) 運動方程式のラプラス変換における, 入力  $F(s)$  と出力  $X(s)$  の比:

$$G(s) = \frac{F(s)}{X(s)}$$

を, 伝達関数という. 伝達関数の分母は, 運動方程式の固有多項式に一致する.

### 10.3 伝達関数の使い方

運動方程式から固有値を計算すると自由振動が分かり, 周波数応答を計算すると強制振動が分った. これらは全て, 伝達関数からでも計算できる.

#### 10.3.1 自由振動の解析

算法 11 p.32 より, 伝達関数の分母は固有多項式に一致する. ゆえに, 固有方程式は,

$$\text{伝達関数 } G(s) \text{ の分母} = 0$$

とすれば復元できる.

実習 10.2 伝達関数  $G(s) = \frac{s+1}{s^2+s+1}$  が表わす振動系の固有値を求め, この系に生じる自由振動を予測せよ. (表 4.1 p.14)

固有方程式は,

$$G(s) \text{ の分母} = s^2 + s + 1 = 0$$

となる. Octave で解くと,

```
[...]$ octave
octave:1> roots([1,1,1])
ans =
  -0.50000 + 0.86603i
  -0.50000 - 0.86603i
```

より, 実部  $< 0$  より減衰, 虚部  $\neq 0$  より振動と分かる. ゆえに予測される自由振動は, 減衰振動である.

#### 10.3.2 強制振動の解析

伝達関数  $G(s)$  に  $i\omega$  を代入すると周波数伝達関数  $G(i\omega)$  が得られる. 算法 10 p.30 によれば,  $G(i\omega)$  の絶対値と偏角が, 定常応答の振幅比と位相差になる.

$$K(\omega) = |G(i\omega)|, \quad \phi(\omega) = \angle G(i\omega)$$

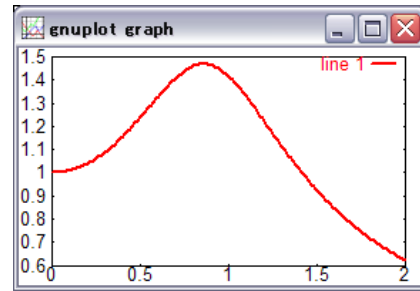
実習 10.3 伝達関数  $G(s) = \frac{s+1}{s^2+s+1}$  が表わす振動系の振幅比と位相差を求めよ.

Octave を使うと, 定義式から直接グラフを描ける.

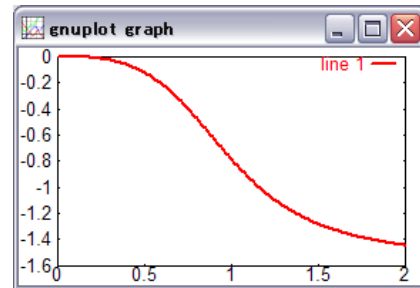
```
octave:1> function y=G(s)
> y = (s .+ 1) ./ (s.**2 .+ s .+ 1);
> endfunction
octave:2> Om=[0:0.01:2];
octave:3> K=abs( G(i*Om) );
octave:4> plot(Om, K);
```

<sup>3)</sup>この分子  $Ls+K$  は, フィードバック制御系でよく出てくる.

<sup>4)</sup>常微分方程式論が不要になるため.



```
octave:5> phi=arg( G(i*Om) );
octave:6> plot(Om, phi);
```



手計算もさほどではない. 分子と分母をそれぞれ指数関数にするのがコツである.  $i\omega$  を代入すると,

$$\begin{aligned} G(i\omega) &= \frac{(i\omega) + 1}{(i\omega)^2 + (i\omega) + 1} \\ &= \frac{1 + (\omega)i}{(1 - \omega^2) + (\omega)i} \\ &= \frac{\sqrt{1 + \omega^2} e^{i(\tan^{-1} \frac{\omega}{1})}}{\sqrt{(1 - \omega^2)^2 + (\omega)^2} e^{i(\tan^{-1} \frac{\omega}{1 - \omega^2})}} \\ &= \sqrt{\frac{1 + \omega^2}{(1 - \omega^2)^2 + (\omega)^2}} e^{i(\tan^{-1} \omega - \tan^{-1} \frac{\omega}{1 - \omega^2})} \end{aligned}$$

ゆえに,

$$\begin{aligned} K(\omega) &= |G(i\omega)| = \sqrt{\frac{1 + \omega^2}{(1 - \omega^2)^2 + (\omega)^2}} \\ \phi(\omega) &= \angle G(i\omega) = \tan^{-1} \omega - \tan^{-1} \frac{\omega}{1 - \omega^2} \end{aligned}$$

のように求まる.

#### 10.3.3 運動方程式の復元

自動制御工学の分野では, システムの表現として, 運動方程式の代りに伝達関数が用いられる. 例えば, いきなり伝達関数:

$$G(s) = \frac{Ls + K}{s^2 + 2s + 1} \quad (K, L \text{ は定数})$$

を与え<sup>3)</sup>, 運動方程式をなるべく解かずに済まそうとする<sup>4)</sup>. ところがどっこい, 振動波形を求めるときだけは, 運動方程式のほうを解くしかない. 運動方程式は, 表 10.1 p.31 を逆にたどると復元できる.

まず、伝達関数の定義 (算法 11 p.32) :

$$\frac{X(s)}{F(s)} = \frac{Ls + K}{s^2 + 2s + 1}$$

において、入出力  $F(s), X(s)$  の位置を左右に戻すと、

$$\iff (s^2 + 2s + 1)X(s) = (Ls + K)F(s)$$

$$\iff s^2X(s) + 2sX(s) + X(s) = LsF(s) + KF(s)$$

までいく。ここで、表 10.1 を用いると、 $s$  の関数を  $s$  倍する操作は、元の時間関数を微分する操作に等しいから、

$$\xrightarrow{\text{表 10.1}} \ddot{x}(t) + 2\dot{x}(t) + x(t) = L\dot{f}(t) + Kf(t)$$

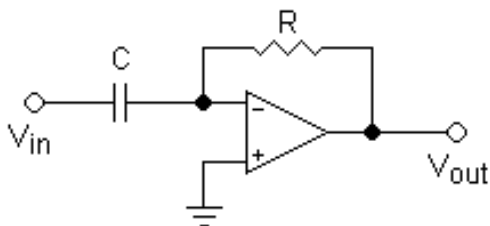
のような運動方程式が復元される。これを 1 階化して Octave で解けば、冒頭の伝達関数 (が表わすシステム) の振動波形が分かる。

ちなみに、この例題の運動方程式：

$$\ddot{x}(t) + 2\dot{x}(t) + x(t) = L\dot{f}(t) + Kf(t)$$

の右辺  $L\dot{f}(t) + Kf(t)$  は、PD 制御 (6.2.3 節 p.19) において、角変位センサしかセンサが無い状況を表わしている。 $f(t)$  を角変位の測定値とする。このとき、P 制御は測定値  $f(t)$  によって可能だが、D 制御に必要な角速度の測定値がない。そこで角変位の測定値  $f(t)$  を、電子回路がコンピュータで微分して、角速度の推定値  $\dot{f}(t)$  を作る。このような操作を伝達関数の分子  $Ls + K$  は表している。

微分回路の一例を示す。 $R$  は抵抗、 $C$  はコンデンサ、 $\triangleright$  はオペアンプといわれる電子部品である。入力電圧が  $V_{in} = v(t)$  のとき、近似的にはあるが、 $V_{out} \approx \frac{dv(t)}{dt}$  が出力される。



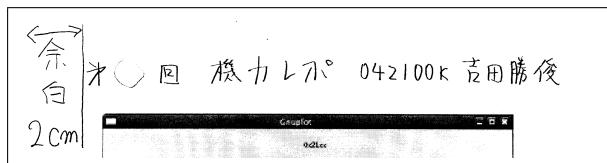
あるいはコンピュータで微分するには、 $\Delta t$  秒ごとに、角変位の測定データ  $\theta_1, \theta_2, \dots, \theta_n, \dots$  を取得する。時刻  $t = n \cdot \Delta t$  における角速度は、近似的に、

$$\frac{d\theta(t)}{dt} \approx \frac{\theta_n - \theta_{n-1}}{\Delta t} \quad \text{または} \quad \frac{\theta_{n+1} - \theta_n}{\Delta t}$$

と推定できる。これを極限操作  $\Delta t \rightarrow 0$  したのが数学的な微分  $\frac{d\theta(t)}{dt}$  だが、現実の測定システムでは  $\Delta t$  は例えばミリ秒程度に設定する。

## レポート提出方法

- A4 一枚以内 (両面使用可) . 左余白 2cm 以上 .
- 上に「第 X 回機力レポ 学籍番号 氏名」と明記 .
- ✂切 : 次回授業日の前日まで . 吉田准教授室 Box .  
次々回授業日からは受けとらない .
- 学籍番号なきグラフは盗作と見なす .



注意 レポートは、新入生を読者に想定して書くこと。グラフの軸にラベルが無かったり、解説や考察が無いと、何をどう理解すればよいか分からない。レポートとは、ある出来事の体験者が、未体験者に向けて書く文書。

第 3 回 レポート課題 次の伝達関数で表される振動系を考える。

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

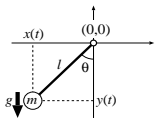
- 1) この系に生じる自由振動を固有値で予測せよ。
- 2) この系の振幅比 (共振曲線)  $K(\omega)$  を Octave で描け。
- 3) この系の位相差  $\phi(\omega)$  を Octave で描け。

# 第 11 講

## 解析力学 (ラグランジュ形式の力学)

**学習目標**

- ベクトル的な方法で、簡単な運動方程式が導ける。(復習)
  - ニュートンの運動法則が使える。
- 解析力学の方法で、単振り子の運動方程式を算出できる。

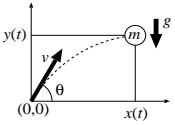


- 座標変換, 運動エネルギー, 位置エネルギー
- ラグランジュの運動方程式

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 1 - 2

**復習 2**

- 地平面から角度  $\theta$  の初速度  $v$  で発射された質点  $m$  の運動  $x(t), y(t)$  を求めよ。重力加速度  $g$  は下向きとする。



1.  $x$  方向の外力  $F_x$  と,  $y$  方向の外力  $F_y$  を調べる.
2.  $x$  方向と  $y$  方向の運動方程式をたてる.
 
$$m\ddot{x}(t) = F_x, \quad m\ddot{y}(t) = F_y$$
3. 2回積分し, 速度  $\dot{x}(t), \dot{y}(t)$  と運動  $x(t), y(t)$  の公式を 自作!
4. 初期値を調べ, 積分定数を定める.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 5 - 6

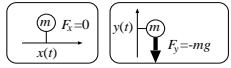
**ベクトル的な方法(復習)**

- ニュートンの運動法則
  - 第1法則: 外力を受けない物体は, 静止か等速直線運動.
  - 第2法則:  $m\ddot{x} = F$  (運動方程式)
    - ※ 線形振動系では  $F = -c\dot{x} - kx + f(t)$
  - 第3法則:  $F_{12} = -F_{21}$  (作用・反作用)
- 手順
  1. 第3法則から, 質点に加わる力を決める.
  2. 第2法則の運動方程式を, 積分で解く.
    - ※ 第1法則は第2法則の帰結だから現代的には使わない

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 2 - 3

**復習 2 の解答例 (1/3)**

1.  $x$  方向の外力  $F_x$  と,  $y$  方向の外力  $F_y$  を調べる.



2.  $x$  方向と  $y$  方向の運動方程式をたてる.
 
$$m\ddot{x}(t) = 0, \quad m\ddot{y}(t) = -mg$$
3. 2回積分.
 
$$\int \ddot{x}(t) dt = \int 0 dt \quad \int \ddot{y}(t) dt = \int -g dt$$

$$\dot{x}(t) = u_0 \quad \dot{y}(t) = -gt + v_0$$

$$\int \dot{x}(t) dt = \int u_0 dt \quad \int \dot{y}(t) dt = \int -gt + v_0 dt$$

$$x(t) = u_0 t + x_0 \quad y(t) = -gt^2/2 + v_0 t + y_0$$

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 6 - 7

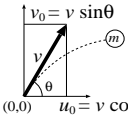
**復習 1**

- 現代の微積分を用いて, 第2法則から第1法則を導け.
  - 第1法則の仮定は, 外力  $F = 0$ .
  - これを前提に第2法則:  $m\ddot{x} = F = 0$  を解け.
  - 解は時間関数  $x(t)$  (これを運動という)
  - $x(t)$  が 0 か等速運動になれば, 第1法則は成立.
- 微積分学の基本定理:  $\int \dot{x} dt = x + x_0$  ( $x_0$  は積分定数)
  - 積分すると, 微分の階数を1つ減らせる.
  - 積分定数の具体値は, 物理的な初期条件から定める.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 3 - 4

**復習 2 の解答例 (2/3)**

4. 初期値を調べる.



積分定数は,  $x(0) = x_0 = 0 \quad y(0) = y_0 = 0$

$$\dot{x}(0) = u_0 = v \cos \theta \quad \dot{y}(0) = v_0 = v \sin \theta$$

したがって, 運動は,

$$x(t) = (v \cos \theta)t, \quad y(t) = -\frac{1}{2}gt^2 + (v \sin \theta)t$$

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 7 - 8

**復習 1 の解答例**

- 速度  $\dot{x}(t)$  の解析:  $m\ddot{x}(t) = 0$  (第1法則の仮定  $F = 0$ )
 
$$\Leftrightarrow \ddot{x}(t) = 0 \quad \because m \neq 0$$

$$\Rightarrow \int \ddot{x}(t) dt = \int 0 dt$$

$$\Rightarrow \dot{x}(t) + C'_1 = 0, \quad v_0 := -C'_1$$

$$\Rightarrow \dot{x}(t) = v_0 \quad v_0 \text{ は積分定数}$$

$\therefore$  初速度が  $\dot{x}(0) = v_0 = 0$  なら速度は 0.  $\dot{x}(0) = v_0 \neq 0$  なら等速運動となるが, これは第1法則を意味する.
- 運動  $x(t)$  の解析:  $\dot{x}(t) = v_0$ 

$$\Rightarrow \int \dot{x}(t) dt = \int v_0 dt$$

$$\Rightarrow x(t) + D'_1 = v_0 t + D'_2, \quad x_0 := D'_2 - D'_1$$

$$\Rightarrow x(t) = v_0 t + x_0 \quad x_0 \text{ は積分定数}$$

※ 以上で, 外力  $F = 0$  ときの運動法則が解明された.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 4 - 5

**復習 2 の解答例 (3/3)**

- 別解
  - 即, 1階化!
 
$$x_1 := x, \quad x_2 := \dot{x}, \quad x_3 := y, \quad x_4 := \dot{y} \text{ と定義して,}$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 0 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -g \end{cases}$$
  - Octave で解けば, 運動  $x(t) = x_1(t), y(t) = x_3(t)$  が求まる.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 8 - 9

### 単振り子

棒は変形しないと仮定

■ **運動方程式 2 本** :  $m\ddot{x} = P \sin\theta (= F_x)$   $P$  は張力  
 $m\ddot{y} = -mg + P \cos\theta (= F_y)$

■ **未知数 3 個** :  $x(t)$ ,  $y(t)$ ,  $P$ . ※このままでは解けない.

■ **拘束条件 1 本** :  $x^2(t) + y^2(t) = l^2$ . ※この式で  $P$  を消去?

☆ ベクトル流だと未知の張力  $P$  が面倒くさい.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 9 - 10

### 2. 運動エネルギー

■ 直交座標で書くと,  $T = \frac{1}{2}m(\dot{x}^2(t) + \dot{y}^2(t))$ .

■ 同じ  $T$  を一般化座標で書く.

■ 運動と速度を  $\theta(t)$  で書くと,  
 $x(t) = -l \sin \theta(t) \Rightarrow \dot{x}(t) = -l\dot{\theta}(t) \cos \theta(t)$   
 $y(t) = -l \cos \theta(t) \Rightarrow \dot{y}(t) = l\dot{\theta}(t) \sin \theta(t)$   
 $\therefore \dot{x}^2(t) + \dot{y}^2(t) = l^2\dot{\theta}^2(t)(\cos^2 \theta(t) + \sin^2 \theta(t)) = l^2\dot{\theta}^2(t)$

■  $T = \frac{1}{2}ml^2\dot{\theta}^2(t) //$

■ ついでに位置エネルギー :  $U = mgy(t) = -mgl \cos \theta(t)$ .

☆ おしまい, 未知の張力  $P$  を使わずに済んだ.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 13 - 14

### 機構学的な自由度

1. 単振り子を機構学的に見ると, 姿勢は  $\theta$  だけで決まる.

■ ならば, 1 自由度だから, 運動方程式は本来 1 本でよい.

2. 棒が変形しないなら, 張力  $P$  は運動に関係ないはず.

■ 実際, 張力  $P$  は消去される.

■ 結局は消去される  $P$  を, わざわざ考えるのは何故か?  
 $\therefore$  第2法則  $m\ddot{x} = P$  でできないから.

☆ もっと単刀直入に, 導けないか?!  $\Rightarrow$  解析力学

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 10 - 11

### 運動方程式の自動算出 (1/2)

i)  $L = T - U = \frac{1}{2}ml^2\dot{\theta}^2(t) + mgl \cos \theta(t)$

■ **算法**  $\frac{\partial f}{\partial x} \stackrel{\text{定義}}{\iff} x$  以外を定数とみなし,  $f$  を  $x$  で微分.

※ 偏微分は, 字面だけで機械的に微分する!

■ **実習**  $\frac{\partial L}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \frac{1}{2}ml^2\dot{\theta}^2 + mgl \cos \theta \right)$  を計算せよ.

■ **実習**  $\frac{\partial L}{\partial \dot{\theta}} = \frac{\partial}{\partial \dot{\theta}} \left( \frac{1}{2}ml^2\dot{\theta}^2 + mgl \cos \theta \right)$  を計算せよ.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 14 - 15

### 解析力学の方法

■ 問題に合わせて用意するもの

- 座標変換. (機構学的自由度ぎりぎりの座標  $\leftrightarrow$  直交座標)
- 質点の運動エネルギー  $T$ .
- 質点の位置エネルギー  $U$ .

■ **運動方程式の半自動算出**

- $L = T - U$  (ラグランジアン) を作る.
- 公式 (ラグランジュの運動方程式) に代入する.  

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = 0$$

☆ なんと! 運動方程式を「算出」できてしまう

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 11 - 12

### 運動方程式の自動算出 (2/2)

ii) ラグランジュの運動方程式に代入する.

■ **実習**  $\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0$  に代入して, 単振り子の運動方程式を導け.

☆ 1階化して Octave で解けば, 単振り子の運動が求まる

■ 1 自由度ぎりぎりの  $\theta$  の運動方程式が, 単独で求まった.

■ 未知の張力  $P$  を一度も使わずに済んだ.

※ 後半からは機械的な計算なので, 数式処理ソフトでできる.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 15 - 16

### 1. 座標変換

■ 機構学的自由度を表すのに, ぎりぎり必要な座標を, 好きなように取る. このような座標を一般化座標という

■ 単振り子の姿勢表現に必要な十分な座標 :  $\theta$  それだけ

■ 座標変換 :  $(x(t), y(t)) = (-l \sin \theta, -l \cos \theta)$

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 12 - 13

### 確認演習

■ **実習** 単振り子の運動方程式を導け. (講義内容そのまま)

※ 角度  $\theta$  のとり方が違うだけ.

Mechanical Dynamics: 第11講 ラグランジュ形式の力学 - 16 - 17

# 第 12 講

## 倒立ロボットの力学

解析力学(ラグランジュ形式の力学)では,エネルギーまで求めれば,あとは機械的な微分計算で運動方程式が自動算出された.このような単純作業は数式処理ソフトにやらせるのが現代の主流である.フリーソフトで入門しよう.

### 12.1 数式処理ソフト入門

$(x+1)^2 = x^2 + 2x + 1$  のように文字式を計算可能なソフトウェアを,数式処理ソフトという.ここではフリーソフトの Maxima<sup>1)</sup>を用いる.Octave とは全く文法が異なるので,以下練習する.

#### 12.1.1 対話的な使用法

**実習 12.1** 端末上で Maxima を起動し終了せよ.

```
[t0x2lxx@a10xx ~]$ maxima      Maxima に入る
Maxima 5.9.2 http://maxima.sourceforge...
--- 中略 ---
(%i1) 1+1;                        ; を付けよ!
(%o1) 2                          答えの行
(%i2) quit();                    Maxima を出る

[t0x2lxx@a10xx ~]$              端末に戻った
```

**実習 12.2** (1)~(3) の実行例を,そのまま実行せよ.

#### (1) 四則演算と微分積分

```
(%i1) 1+2-3*4/5;                四則演算
(%o1) 3
-
5
(%i2) 2^3;                      べき乗
(%o2) 8
(%i3) diff(x^3+x^2,x);          微分してみる
(%o3) 3 x + 2 x
(%i4) diff(x^2+x^3,x,2);        2回微分してみる
(%o4) 6 x + 2
(%i5) diff(x^2*sin(y),y);       yで偏微分
(%o5) x cos(y)
(%i6) integrate(x^3+x^2,x);      積分してみる
(%o6) 4/3 x + 2/3 x^2
(%i7) integrate(x^3+x^2,x,0,1);  0から1まで
(%o7) 7/12                      定積分
```

<sup>1)</sup><http://maxima.sourceforge.net/>

#### (2) 代入と行列演算

```
(%i1) a:5;                      数の代入
(%o1) 5
(%i2) a;                          代入されてる
(%o2) 5                            ベクトルの代入
(%i3) v:[1,2,3];                  [1, 2, 3]
(%o3) [1, 2, 3]
(%i4) v[2];                       ベクトル第2成分
(%o4) 2
(%i5) A:matrix([-1,2,3],[4,5,6],[7,8,9]); 行列の代入
(%o5) [ -1 2 3 ]
      [ 4 5 6 ]
      [ 7 8 9 ]
(%i6) A[2,1];                     行列の(2,1)成分
(%o6) 4
(%i7) transpose(A);              行列の転置
(%o7) [ -1 4 7 ]
      [ 2 5 8 ]
      [ 3 6 9 ]
(%i8) determinant(A);            行列式
(%o8) 6
(%i9) rank(A);                   行列の階数
(%o9) 3
(%i10) invert(A);                逆行列
(%o10) [ 1 1 ]
       [ - - 1 - - ]
       [ 2 2 ]
       [  ]
(%i11) A . v;                    行列とベクトルの積
(%o11) [ 12 ]
       [ 32 ]
       [ 50 ]
(%i12) v . v;                    ベクトルの内積
(%o12) 14
(3) ユーザー関数の定義と活用
(%i1) f(x):=x^2+sin(x)*cos(x);    1変数関数の定義
(%o1) f(x) := x + sin(x) cos(x)
(%i2) diff(f(x),x);              試しに微分
(%o2) 2 x - sin(x) + cos(x) + 2 x
(%i3) g(x,y):=x^2*sin(y);        2変数関数の定義
(%o3) g(x,y) := x sin(y)
(%i4) diff(g(x,y),y);            試しに偏微分
(%o4) x cos(y)
(%i5) matrix([cos(t),-sin(t)],  行列値関数の定義
              [sin(t),cos(t)])$ ; の代りに $
(%o5) [ cos(t) - sin(t) ]      だと出力を抑制
(%i6) rot(t);                    確認
(%o6) [ cos(t) - sin(t) ]
```

```
(%08)      [          ]
           [ sin(t)  cos(t) ]
(%i9) diff(rot(t),t);      試しに t で微分
           [ - sin(t)  - cos(t) ]
(%o9)      [          ]
           [  cos(t)  - sin(t) ]
```

### □ 12.1.2 プログラムファイルの実行

長い処理でタイプミスすると、それまでのキーボード入力が無駄になって、かなりへこむ。そういうときは、プログラムファイルを書いて実行すればよい。出力が見にくいときは、行末の ; を \$ にすると、余計な出力を抑制できる。

```

----- プログラム「sample.mac」 -----
f(x):=x^2+sin(x)*cos(x)$
df:=diff(f(x),x)$          微分
trigsimp(trigreduce(df));  整理

```

実習 12.3 プログラム「sample.mac」のファイルを作り、右の (1), (2) の要領でそれぞれ実行せよ。

#### (1) 起動後に実行する方法

```
[t0x2lxx@a10xx ~]$ gedit sample.mac
[t0x2lxx@a10xx ~]$ maxima
--- 中略 ---
(%i1) batch("sample.mac");
2
(%i2)      f(x) := sin(x) cos(x) + x
(%i3)      df : diff(f(x), x)
           2      2
(%o3)      - sin (x) + cos (x) + 2 x
(%i4)      trigsimp(trigreduce(df))
(%o4)      cos(2 x) + 2 x
(%i5) quit();          端末に戻る .
[t0x2lxx@a10xx ~]$
```

#### (2) 起動と同時に実行する方法

```
[t0x2lxx@a10xx ~]$ maxima -b sample.mac
--- 中略 ---
(%o4)      cos(2 x) + 2 x
[t0x2lxx@a10xx ~]$          自動で端末に戻る
```

豆知識 `trigsimp(trigreduce(%cdots$))` は三角関数を簡略 ( $\sin^2 + \cos^2 = 1$  など) するための処理。多項式の簡略は `ratsimp(%cdots$)`。

## □ 12.2 解析力学 (ラグランジュ形式の力学)

11 講で学んだ内容は、 $n$  変数の場合に一般化できる。

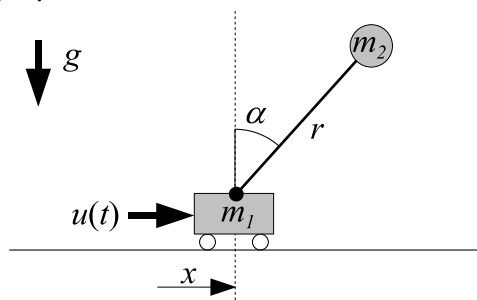
算法 12 (ラグランジュ形式) 機構の姿勢が一意に定まる一般化座標  $q_1, q_2, \dots$  をとる。機構の全運動エネルギーが  $T$ 、全位置エネルギーが  $U$  であるとき、この機構の運動方程式は、ラグランジュ関数  $L = T - U$  を介して、

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0 \quad (i = 1, 2, \dots)$$

で求まる。これをラグランジュの運動方程式という。

注意 一般化座標の個数だけ、ラグランジュの運動方程式を求める。例えば、一般化座標が  $q_1, q_2, q_3$  で 3 個なら、ラグランジュの運動方程式も  $i = 1, 2, 3$  で 3 本求める。

この算法を使って、次の倒立ロボットの運動方程式を求めてみよう。ただし計算は Maxima に手伝ってもらおう。現代の研究者は大概そうする<sup>2)</sup>。



滑らかに水平移動できる質量  $m_1$  [kg] の台車に、質量  $m_2$  [kg] の振子が滑らかに回転支持されている。棒の自重は無視できるものとする。台車には水平方向に外力  $u(t)$  [N] を加えることができる。

### □ 12.2.1 運動学 (座標変換)

この機構の姿勢を台車の水平変位  $x$  [m]、振子の角変位  $\alpha$  [rad] で表わす。すなわち一般化座標は  $(q_1, q_2) = (x, \alpha)$ 。このとき、質点  $m_1, m_2$  の直交座標  $x_1, x_2$  は、台車の重心高さを  $h$  (定数) として、次のよう書ける。

$$\mathbf{x}_1 = (x, h), \quad \mathbf{x}_2 = (x + r \sin \alpha, h + r \cos \alpha)$$

<sup>2)</sup> 企業の技術者は 3D-CAD の解析機能を使うと思うが、これは「解析力学」をマウス操作で装飾したものである。

□ 12.2.2 エネルギー

(1) 運動エネルギー  $T$  台車と振子の運動エネルギーはそれぞれ、(ドットは時間微分)

$$T_1 = \frac{m_1}{2}(\dot{x}_1 \cdot \dot{x}_1), \quad T_2 = \frac{m_2}{2}(\dot{x}_2 \cdot \dot{x}_2)$$

である．例えば  $T_1$  を Maxima で計算すると、

```
(%i1) x1:[x(t),h]$
(%i2) x2:[x(t)+r*sin(a(t)),h+r*cos(a(t))]+$
(%i3) T1:( (diff(x1,t).diff(x1,t))*m1/2 );
                                d      2
                                m1 ( -- (x(t)))
                                dt
(%o3) -----
                                2
```

\$ は ; の代り . 出力の抑制  
運動エネルギー 1

のような結果が得られる．同様に  $T_2$  も計算できる．このまま対話的に進めてもよいが、プログラムファイルに残したほうが、計算の経過を再利用しやすい．

プログラム「robo1.mac」

```
x1:[x(t),h]$
x2:[x(t)+r*sin(a(t)),h+r*cos(a(t))]+$
T1:( (diff(x1,t).diff(x1,t))*m1/2 )$
T2:( (diff(x2,t).diff(x2,t))*m2/2 )$
T:T1+T2$
trigsimp(T);
```

\$ は ; の代り . 出力の抑制  
運動エネルギー 1(台車)  
運動エネルギー 2(振子)  
全運動エネルギー  
三角関数の整理!

実習 12.4 12.1.2 節の要領で、プログラムを実行せよ。

プログラム「robo1.mac」の実行結果は、

```
[t0x21xx@a10xx ~]$ maxima
(%i1) batch("robo1.mac");
--- 中略 ---
(%i7) trigsimp(T)
                                d      2      d      d
(%o7) ((m2 + m1) ( -- (x(t))) + 2 m2 r cos(a(t)) ( -- (a(t))) ( -- (x(t)))
                                dt      dt      dt
                                2 d      2
                                + m2 r ( -- (a(t))) )/2
                                dt
```

Maxima 起動  
途中まで自動処理

となり、全運動エネルギー  $T$  が、次のように求まる．

$$T = T_1 + T_2 = \left( (m_1 + m_2)\dot{x}^2 + 2m_2 r \dot{\alpha} \dot{x} \cos \alpha + m_2 r^2 \dot{\alpha}^2 \right) / 2 = \frac{(m_1 + m_2)}{2} \dot{x}^2 + m_2 r \dot{\alpha} \dot{x} \cos \alpha + \frac{m_2 r^2}{2} \dot{\alpha}^2$$

(2) 位置エネルギー  $U$   $m_1, m_2$  の鉛直変位より、 $U = m_1 gh + m_2 g(h + r \cos \alpha)$  である．

□ 12.2.3 ラグランジュの運動方程式

したがって、ラグランジュ関数  $L = T - U$  は、

$$L = T - U = \frac{m_1 + m_2}{2} \dot{x}^2 + m_2 r \dot{\alpha} \dot{x} \cos \alpha - \frac{m_2 r^2}{2} \dot{\alpha}^2 - m_1 gh - m_2 g(h + r \cos \alpha)$$

となる．算法 12 p.37 にしたがって、この  $L$  をラグランジュの運動方程式：

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = Q_x, \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = Q_\alpha \tag{12.1}$$

に代入すれば、一般化座標  $(x, \alpha)$  それぞれに関する運動方程式が求まる． $Q_x, Q_\alpha$  はそれぞれ  $x, \alpha$  方向の力で、この問題では  $Q_x = u(t)$ 、 $Q_\alpha = 0$  である<sup>3)</sup>．

<sup>3)</sup>角度  $\alpha$  方向の力とは、トルク [Nm] のこと．



運動方程式 (12.1) を書き下すために,  $L$  の偏微分やその時間微分を求める. まず, 先のプログラムにラグランジュ関数までを書き足しておく.

プログラム「robo2.mac」

```
x1:[x(t),h]$
x2:[x(t)+r*sin(a(t)),h+r*cos(a(t))]+$
T1:(diff(x1,t).diff(x1,t))*m1/2)$
T2:(diff(x2,t).diff(x2,t))*m2/2)$
T:T1+T2$
U:(m1*g*h + m2*g*(h + r*cos(a(t))))$
L:T-U;
```

\$ は ; の代り. 出力の抑制

運動エネルギー 1(台車)  
運動エネルギー 2(振り)  
全運動エネルギー  
ポテンシャル  
ラグランジュ関数

これを利用して, (12.1) の第 1 式にある  $\frac{\partial L}{\partial \dot{x}}$ ,  $\frac{d}{dt} \frac{\partial L}{\partial \dot{x}}$ ,  $\frac{\partial L}{\partial x}$  を順に求めてみよう.

```
[t0x2lxx@a10xx ~]$ maxima
(%i1) batch("robo2.mac");
--- 中略 ---
(%i9) dx:=diff( x(t), t );
(%o9)
d
-- (x(t))
dt
(%i10) Ldx:=diff( L, dx );
(%o10)
d d
m2 (--- (x(t)) + r cos(a(t)) (--- (a(t)))) + m1 (--- (x(t)))
dt dt dt
(%i11) Ldxdt:=diff( Ldx, t );
(%o11)
d
m2 (--- (x(t))) + m2 r cos(a(t)) (--- (a(t)))
dt dt
- m2 r sin(a(t)) (--- (a(t)))
dt
さらに
(%i13) Lx:=diff( L, x(t) );
(%o13)
0
```

となる. 必要な式を拾うと, (%o12) と (%o13) より,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = (m_1 + m_2)\ddot{x} + (m_2 r \cos \alpha)\ddot{\alpha} - (m_2 r \sin \alpha)\dot{\alpha}^2, \quad \frac{\partial L}{\partial x} = 0$$

である. これから  $x$  に関する運動方程式が得られる.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = Q_x \iff (m_1 + m_2)\ddot{x} + (m_2 r \cos \alpha)\ddot{\alpha} - (m_2 r \sin \alpha)\dot{\alpha}^2 = u(t) \quad (12.2)$$

以上の数式処理を, プログラムファイルとして整理しておく.

プログラム「robox.mac」

```
x1:[x(t),h]$
x2:[x(t)+r*sin(a(t)),h+r*cos(a(t))]+$
T1:(diff(x1,t).diff(x1,t))*m1/2)$
T2:(diff(x2,t).diff(x2,t))*m2/2)$
T:T1+T2$
U:(m1*g*h + m2*g*(h + r*cos(a(t))))$
L:T-U$
dx:=diff( x(t), t );
Ldx:=diff( L, dx );
Ldxdt:=diff( Ldx, t );
Lx:=diff( L, x(t) );
eqnx:( Ldxdt - Lx );
ratsimp(trigsimp(eqnx));
```

実習 12.5 プログラム「robox.mac」を改造して「roboa.mac」とし、振子の傾斜角  $\alpha$  に関する運動方程式、

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = Q_\alpha \quad (12.3)$$

を導け。

### 12.3 倒立ロボットの運動方程式

以上、水平変位  $x$  と傾斜角  $\alpha$  に関する運動方程式は、(12.2)、(12.3) より、

$$\begin{cases} (m_1 + m_2)\ddot{x} + (m_2 r \cos \alpha)\ddot{\alpha} - (m_2 r \sin \alpha)\dot{\alpha}^2 = u(t) & (x \text{ 方向}) \\ (m_2 r \cos \alpha)\ddot{x} + (m_2 r^2)\ddot{\alpha} - g m_2 r \sin \alpha = 0 & (\alpha \text{ 方向}) \end{cases} \quad (12.4)$$

$\alpha$  方向を整理すると、 $\cos \alpha \ddot{x} + r \ddot{\alpha} - g \sin \alpha = 0$ .

であることが判明した。式 (12.4) の解  $(x(t), \alpha(t))$  が、倒立ロボットの運動を表わす。

(1) 非線形な運動方程式 思い返せば、線形振動系の運動方程式

$$m\ddot{x} + c\dot{x} + kx = f(t)$$

は、 $\ddot{x}$ ,  $\dot{x}$ ,  $x$  の 1 次式になっていた。「1 次式」を「線形式」ともいうので、「線形」振動系なわけだ。

これに対して、倒立ロボットの運動方程式 (12.4) には、傾斜角  $\alpha$  について、

$$\cos \alpha, \quad \sin \alpha, \quad (\dot{\alpha})^2$$

のような、1 次式でない項が含まれている。このような 1 次式でない項を非線形であるという (0 次式はカウントしない)。非線形項を含む運動方程式を、非線形な運動方程式という。

(2) 数値解法の強み べつに非線形でいいじゃねーか、とできればよいのだが、困ったことに、

非線形な運動方程式の解を式で書き下すのは、一般には不可能！

という現実が立ちただかる。例えば、倒立ロボット (12.4) の解  $(x(t), \alpha(t))$  は、現在の数学では書き出せない。

しかし非線形であっても数値解なら求まる。数値解が表す姿勢を時々刻々と描いていけば、倒立ロボットのアニメーションが完成する。この用途にはふたたび Octave が役にたつ。

さて、どんな動きが出てくるか。次回、乞御期待！

# 第 13 講

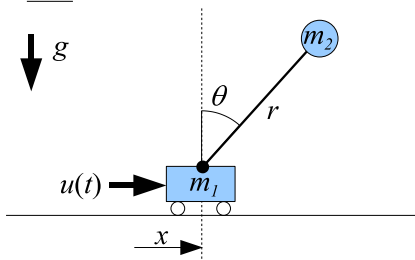
## ロボット制御への応用 II

これまでに学んだ機械力学の素養をベースに、ロボット制御の技術を開発してみよう。これをお勉強として見れば、諸君にはまだ予備知識が足りない。しかし仕事の世界では、その程度の素養でも十分に担当可能と見なされる。

### 13.1 倒立ロボット

前回、次の倒立ロボットの運動方程式を求めた。

注意：傾斜角を  $\theta$  におき直した！



復習すると、ロボットの姿勢を台車の水平変位  $x$  と、振り子の傾斜角  $\theta$  で表すとき、一般化座標<sup>1)</sup>を  $(x, \theta)$  とする運動方程式 p.40 は、算法 12 p.37 より、

$$\begin{cases} (m_1 + m_2)\ddot{x} + (m_2 r \cos \theta)\ddot{\theta} \\ \quad - (m_2 r \sin \theta)\dot{\theta}^2 = u(t) & x \text{ 方向} \quad (*) \\ \cos \theta \ddot{x} + r\ddot{\theta} - g \sin \theta = 0 & \theta \text{ 方向} \end{cases}$$

となった。

#### 13.1.1 1 階化の工夫

運動方程式 (\*) を Octave で解くには 1 階化 (3.4.1 節) しなければならない。式 (\*) をよく見ると、加速度  $\ddot{x}, \ddot{\theta}$  の 1 次連立方程式になっている。行列で書くと、

$$\underbrace{\begin{bmatrix} M + m & ml \cos \theta \\ \cos \theta & l \end{bmatrix}}_{A_\theta} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} ml\dot{\theta}^2 \sin \theta + f(t) \\ g \sin \theta \end{bmatrix}$$

$$\therefore \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = A_\theta^{-1} \begin{bmatrix} ml\dot{\theta}^2 \sin \theta + f(t) \\ g \sin \theta \end{bmatrix}$$

である。以上を解いた答えが  $\ddot{x} = \quad, \ddot{\theta} = \quad$  であるとき、 $y_1 = x, y_2 = \dot{x}, y_3 = \theta, y_4 = \dot{\theta}$  とおくことで、

$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = \quad \\ \dot{y}_3 = y_4 \\ \dot{y}_4 = \quad \end{cases}$$

<sup>1)</sup>問題を解く人に都合のよい座標を一般化座標という。

のように 1 階化できる。この 1 階化の計算は、Maxima の段階でもできるし、Octave の段階でもできる。

比較すると、Maxima の段階で 1 階化しておけば、Octave によるアニメーション作成時に逆行列計算がいらぬので、アニメーションが速くなる。反面、1 階化後の文字式は複雑なので、これを Octave に転記するときに、スペルミスを犯しやすい。最終的には好みの問題だが、今回の選択としては、逆行列計算を Octave にやらせる作戦でいこう。

#### 13.1.2 ちなみに線形化

倒れ角  $\theta$  と、角速度  $\dot{\theta}$  が十分に小さいとき、

$$\sin \theta \approx \theta, \quad \cos \theta \approx 1, \quad \dot{\theta}^2 \approx 0$$

と近似できる。このような近似法を線形化という。式 (\*) に代入すると、

$$\begin{cases} (m_1 + m_2)\ddot{x} + m_2 r \ddot{\theta} = u(t) \\ \ddot{x} + r \ddot{\theta} - g \theta = 0 \end{cases}$$

となる。第 2 式の  $\ddot{x}$  を第 1 式に代入すると、

$$(m_1 + m_2)(-r \ddot{\theta} + g \theta) + m_2 r \ddot{\theta} = u(t)$$

より、

$$(m_1 r) \ddot{\theta} - g(m_1 + m_2) \theta = -u(t) \quad (**)$$

が得られる。第 6 講では、この式 (\*\*) を近似的な運動方程式として用いた。

### 13.2 制御を受けるロボットの挙動

外力  $u(t)$  を、具体的な制御力に置き換えると、倒立ロボットの動作が実現する。 $u(t)$  として何を代入すればよいのか探していこう。

#### 13.2.1 制御なしの自由運動

まず、制御を加えない  $u(t) = 0$  のときの自由運動を観察しよう。次頁のプログラム「roboanim.m」を実行せよ。

#### 13.2.2 振り子の倒立制御

プログラムの 5 行目が制御力  $u(t)$  だが、これを、  
`ut = 20*y(3) + 2*y(4);`  
 へ変更、15 行目の初期値  $(x(0), \dot{x}(0), \theta(0), \dot{\theta}(0))$  を、  
`x0 = [0; -3; 0; 1.2];`

```

global m1 m2 r;
m2=1/3; m1=1-m2; r=1;
function dy = eqn(y, t)
    global m2 m1 r;
    ut = 0.0; # (1)ここに制御力!
    A = [m1+m2, m2*r*cos(y(3)); cos(y(3)),r];
    b = [m2*r*(y(4)**2)*sin(y(3))+ut; 9.8*sin(y(3))];
    h = A\b; #(Aの逆行列)*bと同じ.inv(A)*bと書くより速い
    dy(1) = y(2);
    dy(2) = h(1);
    dy(3) = y(4);
    dy(4) = h(2);
endfunction
n = 500;
x0 = [0; 0; 0.5; 0]; # (2)ここに初期値!
t = linspace(0, 0.02*n, n);
x = lsode("eqn", x0, t);
__gnuplot_set__ size 0.7,1;
__gnuplot_set__ xrange[-5.5:5.5];
__gnuplot_set__ yrange[-5.5:5.5];
__gnuplot_set__ xzeroaxis;
for i=1:n
    title(sprintf("x=%f, angle=%f",x(i,1),x(i,3)));
    x1x = x(i,1); x1y = 0; #台車の座標
    x2x = x1x + r*sin(x(i,3)); x2y = r*cos(x(i,3)); #振り子の座標
    plot([x1x,x2x],[x1y,x2y]); #棒の描画
    if ( i==1 ) pause(2);
    else pause(0.02); #この行の数値を大きくすると遅くなる
    endif
endfor

```

に変更し、実行せよ。y(3)は倒れ角  $\theta$ , y(4)は角速度  $\dot{\theta}$  であるから、この制御力を数式で書くと、

$$u(t) = a_1\theta + a_2\dot{\theta}, \quad a_1 = 20, a_2 = 2 \quad (13.1)$$

であり、倒れ角  $\theta$  に関する PD 制御となっている。 $\theta$  そのものの係数  $a_1$  を比例ゲイン、その微分の係数  $a_2$  を微分ゲインという。

式(13.1)の制御力  $u(t)$  には、台車の位置  $x$  が加味されていない。別の言い方をすると、この制御方式のロボットには、自分の位置  $x$  を制御する知能がない。その結果、倒れ角が目標の  $\theta = 0$  へ到達した後も、台車は  $x$  方向の自由運動(等速直線運動)を続けている。

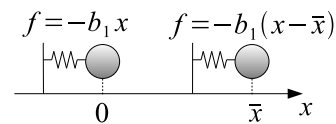
### □ 13.2.3 台車の位置制御

そこで逆に、台車の位置  $x$  を制御する知能を与えてみよう。簡単のため、倒れ角  $\theta$  は自由状態とする。

まず、ロボットを到達させたい位置を  $x = \bar{x}$  としよう。これを目標値という。目標値  $\bar{x}$  に対して、

$$u(t) = -b_1(x(t) - \bar{x})$$

のような制御力を考える。 $b_1 > 0$  のとき、これは  $\bar{x}$  を原点とする線形復元力(ばね)として効く。



結果的に次のようなカラクリが実現できるはずだ。

- 目標値  $\bar{x}$  より台車が左にある【 $x(t) < \bar{x}$ 】  
 $\Rightarrow u(t) > 0$  より、台車は右に押される。
- 目標値  $\bar{x}$  より台車が右にある【 $x(t) > \bar{x}$ 】  
 $\Rightarrow u(t) < 0$  より、台車は左に押される。

こうして、目標値  $\bar{x}$  を原点とする復元力が発生するので、ロボットは人為的な原点  $\bar{x}$  に向うだろう。

以上の仮説を Octave で検証しよう。到達させたい目標位置を  $\bar{x} = 3$  としよう。5行目の制御力の部分を、

```
bx=3; ut = -4*(y(1)-bx);
```

15行目の初期値を、

```
x0 = [0; 0; pi; 0];
```

に変更せよ。

実行結果より、この制御方式では、台車が振動してしまっただけで減衰しない。そこで、プレーキ力  $c\dot{x}$  に相当する項も作るために、台車速度  $\dot{x}(t)$  もフィードバック

クしてやる．

$$u(t) = -b_1(x(t) - \bar{x}) - b_2\dot{x}(t) \quad (13.2)$$

プログラムの 5 行目を，例えば，

```
bx=3; ut = -4*(y(1)-bx) - 1.5*y(2);
```

とせよ．脱力状態の振り子をぶら下げながら，台車が目標値  $\bar{x}$  を中心に減衰振動するはずだ．

### 13.3 倒立ロボットの位置制御

以上，振り子の倒立制御と，台車の位置制御にそれぞれ成功した．これらを組み合わせれば，倒立しながら目標に向かう倒立ロボットが実現しそうである．

#### 13.3.1 倒立制御と位置制御の重ね合わせ

もっとも素朴なアイデアとして，振り子の倒立制御の制御力と，台車の位置制御の制御力を，単純に足し合せてみよう．

$$u(t) = \underbrace{a_1\theta(t) + a_2\dot{\theta}(t)}_{\text{倒立制御 (13.1)}} - \underbrace{b_1(x(t) - \bar{x}) - b_2\dot{x}(t)}_{\text{位置制御 (13.2)}}$$

条件  $a_1 = 20, a_2 = 2, b_1 = 4, b_2 = 1.5$  において，それぞれが制御に成功しているわけだから，これで，目標に向かう倒立ロボットが実現できそうである．

同じことをプログラムに書くには，5 行目を，  

```
ut = 20*y(3)+2*y(4)-4*(y(1)-3)-1.5*y(2);
```

とすればよからう．また，こんどは倒立状態から始めるため，15 行目の初期値は，

```
x0 = [0; 0; 0; 0];
```

としておく．

実行結果では，目標値  $\bar{x} = 3$  に向おうとする台車に，足元をすくわれる形で振り子がバランスを崩す．もはや立ち上がることはできず ( $\theta \neq 0$ )， $\theta$  が小さくなれないので，無意味な制御力  $u(t) = a_1\theta + \dots$  が効き続けて，ロボットは暴走を始める．

このように，目的の異なる 2 つの制御力を単純に足し合せてただけでは，互いに邪魔をしあって，望む制御結果が得られない場合がある．

#### 13.3.2 制御工学の問題

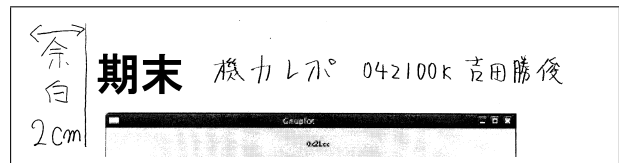
それでは，どのような制御力  $u(t)$  を用意してやれば，倒立しながら目標位置  $\bar{x}$  に向かうロボットが実現できるのだろうか？

このようなロボットの実現可能性については，人間の立位歩行が 1 つの証拠を与える．我々人間は，本来なら不安定な立位を維持したまま，自在に目標位置へと移動していくわけで，そのための制御力  $u(t)$  は，何らかの方法で実現可能と見るほかない．

制御力  $u(t)$  の実現方法を扱う学問を制御工学というが，その詳細を述べることは本講義の範囲を超える．ここでは，制御工学の知識を仮定しないで，倒立ロボットの制御力  $u(t)$  の実現問題を，そっくり期末レポートの課題としよう．健闘を祈る．

### レポート提出方法

- A4 一枚以内 (両面使用可)．左余白 2cm 以上．
- 上に「期末 機力レポ 学籍番号 氏名」と明記．
- ✂切：試験期間の前日まで．吉田准教授室 Box．  
夏季休業開始後は受けとらない．
- 学籍番号なきグラフは盗作と見なす．



期末レポート課題 初期状態  $x(0) = \dot{x}(0) = \theta(0) = \dot{\theta}(0) = 0$  (原点で倒立静止) から，倒立したまま目標位置  $\bar{x} = 3$  へ移動して停止するロボットの制御方法を，試行錯誤的に実現せよ．

- (1) そのための制御力  $u(t)$  を考案し，プログラム「roboanim.m」の 5 行目の改造方法を示せ．
- (2) 改造したプログラムを実行し，倒立状態で目標位置  $x = \bar{x} = 3$  に停止する実行例を示せ．(レポートには実行時の画像を 2~3 コマ示せばよい)

配点 この期末レポートは，この授業の成績評価の 40% を占めます．

#### 採点基準と配点

項目	無	有	+工夫
考案した制御力の式	0	5	8
制御力の原理の解説	0	5	8
プログラム改造方法	0	5	8
実行結果を示す画像	0	5	8
実行結果の解説と考察	0	5	8
満点			40

# 第 14 講

## まとめ

この授業の冒頭で、3つの達成目標<sup>1)</sup>を掲げた。

- 1) 機械力学の基礎理論の理解を深める。(学科 D:専門知識)
- 2) コンピュータ演習によって機械力学に密接した情報技術を習得する。(学科 C:工学基礎)
- 3) 問題解決型の課題演習によって自主的な解決能力を養成する。(学科 H:解決能力)

目標 2) については、数値計算プログラムを改造してグラフを出力するレポート課題を 4 回反復したので、機械力学は何をどう計算する学問なのか、一定の見通しが得られたはずである。

目標 3) については、これは期末レポートとして現在進行中だが、あえて不十分な予備知識のまま、倒立ロボットの実現問題に取り組んでもらっている。

目標 1) が本題だが、この授業を十分理解した諸君には、次の 20 項目が実行可能である(順不同)。

### A. 自由振動の理論 — 固有値解析

- 1) 自由振動を表わす運動方程式を書き下す。
- 2) Octave で、振動波形(運動方程式の解)を求める。
- 3) 固有値を使って、振動波形を予測する。
- 4) 減衰比と固有振動数を定義し、実用する。
- 5) PD 制御の効果を、自由振動系の言葉で表わす。
- 6) 固有値を使って、倒立ロボットを設計する。

### B. 強制振動の理論 — 周波数応答解析

- 7) 強制振動を表わす運動方程式を書き下す。
- 8) 過渡応答と定常応答を区別し、色分けする。
- 9) 高校数学で、定常応答を手計算する。
- 10) オイラーの公式で、定常応答を手計算する。
- 11) 定常応答から、振幅比と位相差を取り分ける。
- 12) 周波数応答(振幅比と位相差)を定義する。
- 13) 周波数応答を、ボード線図に描く。

### C. 制御理論の基礎 — 伝達関数

- 14) 定常応答の計算過程から、周波数伝達関数を取り分け、伝達関数を定義する。
- 15) 伝達関数から固有値を求める。(自由振動)
- 16) 伝達関数から周波数応答を求める。(強制振動)
- 17) 運動方程式をラプラス変換する。

### D. 運動方程式の導出 — 解析力学

- 18) ラグランジュの運動方程式で、倒立ロボットの運動方程式を導く。
- 19) ロボットの運動を、アニメーション表示する。
- 20) ロボットの制御方法を、試行錯誤的に検討する。

以上は機械力学のエッセンスであり、物体の運動や振動を扱う技術において<sup>2)</sup>、どれも重要なものばかりである。

<sup>1)</sup>第 1 講スライドを参照せよ。

<sup>2)</sup>動く機械は全て該当する。