Ec4 制御工学 IB　第 2 回

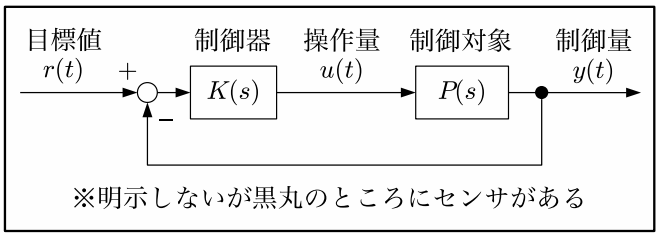
1　制御とは

制御とは**対象となる物（システム）を思うように操る**こと。人間が操作する制御を**手動制御**，コンピュータで操作を行う制御を**自動制御**という。制御方式には主に 2 つの方式がある。

シーケンス制御	あらかじめ定められた順序に従って動作させる制御
フィードバック制御	センサから得られる現在の状況に基づいて動作させる制御

2　ブロック線図

制御工学では制御対象や制御を行うシステム（**制御器**）をブロックで表し，入力と出力の伝わる経路を矢印で表すことで制御システムの構造を図で表現する。この図を**ブロック線図**という。制御対象への入力を**操作量**，制御対象からの出力（つまり制御したい量）を**制御量**と呼ぶ。



フィードバック制御の構成を表すブロック線図

3　制御の目的

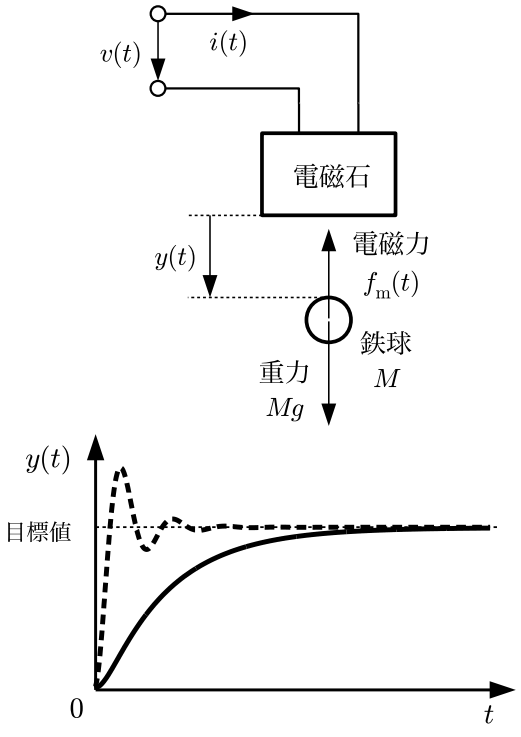
これらの制御の基本的な目的は**制御量を目標値に一致させること**である。もう少し詳しく言うと，

(1) 制御対象の安定化（放っておいても暴走しない）
(2) 目標値追従（最終的にちゃんと目的の状態になる）
(3) 外乱の影響の抑制（多少操作を間違えても OK）

の 3 点が制御の主な目的である。これらを満たす制御器を作るために必要なことを学ぶのが制御工学である。

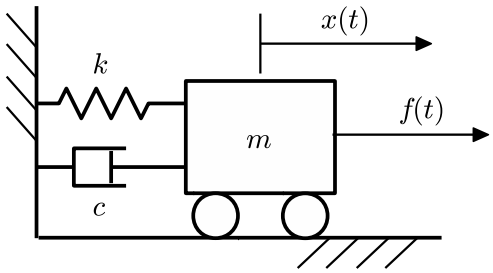
例えば右図のような電磁石で鉄球を浮上させるシステムを考える。入力（操作量）は電磁石に加える電圧 $v(t)$ であり，出力（制御量）は鉄球の位置 $y(t)$ である。この制御では，鉄球を浮かせることと目標の位置に留めることが目標である。

また，ただ目標値にするまでの過程も重要である。右図の実線と破線の 2 つの応答はどちらも目標値に追従している。求められるスピード感や目標値を超えてよいかどうかなどからどちらが良いか判断することになる。



4　数学的モデルとラプラス変換

制御対象の数学的モデルは運動方程式などによって**微分方程式**の形で定式化できる。例として質量–バネ–ダンパ系（**MSD システム**）の位置 $x(t)$ を外力 $f(t)$ で制御することを考える。MSD システムの数学的モデル（運動方程式）は



$$m\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2} = f(t) - kx(t) - c\frac{\mathrm{d}x(t)}{\mathrm{d}t}$$

となる。MSD システムの微分方程式であれば手計算で解析的に解くこともできるが，より複雑な制御対象を考えていくと手計算で解くのは難しい。コンピュータを使ってルンゲクッタ法などで数値的に解くことはできるが，**ある入力を与えたときの出力（制御量）**を調べることはできても，**制御量を仕様通りに制御するにはどのような入力を与えればよいか**を調べることは難しい。

制御工学では**ラプラス変換**を応用することで，適切な入力の解析を行う。ラプラス変換を使うことで，微分方程式は代数方程式に書き換えることができるからである。$x(0) = 0, \dot{x}(0) = 0$ とし，$X(s) = \mathcal{L}[x(t)]$, $F(s) = \mathcal{L}[f(t)]$ として MSD システムの微分方程式の両辺をラプラス変換すると次の式になる。

$$ms^2X(s) = F(s) - kX(s) - csX(s)$$

5　システムの伝達関数と時間応答

ラプラス変換の導入によって**伝達関数**が定義できる。伝達関数を用いて**時間応答**（過渡応答・定常応答）や**周波数応答**を調べることで，制御の目的である安定性や目標値追従性などの評価を可能にする理論体系が整っている。制御対象の伝達関数をもとに適切な制御器の設計を行うための学問が**制御工学**である。

伝達関数はシステムの入出力関係を表す関数で，

$$\text{伝達関数} = \frac{\text{制御量のラプラス変換}}{\text{操作量のラプラス変換}}$$

で定義される。MSD システムにおける伝達関数 $P_{\mathrm{MSD}}(s) = X(s)/F(s)$ は次の式になる。

$$(ms^2 + cs + k)X(s) = F(s)$$

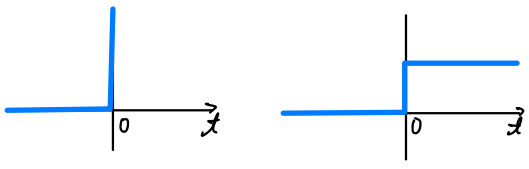
$$\Leftrightarrow \quad P_{MSD}(s) = \frac{X(s)}{F(s)} = \frac{1}{ms^2 + cs + k}$$

伝達関数が $P(s)$ で与えられるシステムに入力 $u(t)$ を与えたときの出力の時間変化 $y(t)$ のことを**時間応答**という。ラプラス変換を使わないと微分方程式を解くことになるが，

$$y(t) = \mathcal{L}^{-1}[Y(s)] = \mathcal{L}^{-1}[P(s)U(s)]$$

であるから，入力のラプラス変換を固定と考えれば，伝達関数が時間応答を決めることになる。

特に入力が単位インパルス関数 $\delta(t)$ の場合を**（単位）インパルス応答**，入力が単位ステップ関数 $u_{\mathrm{s}}(t)$ の場合を**（単位）ステップ応答**という。



6　一次システムの時間応答

制御工学では基本的に伝達関数が係数が実数の有理式

$$P(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}$$

で表されるシステム（**線形時不変システム**）について考え，分母の次数 n に対応して **n 次システム**と呼ぶ。また，基本的に $m < n$ の伝達関数（**（厳密に）プロパーな伝達関数**）について考える。

一次システムの伝達関数は以下の式で表される。

$$P(s) = \frac{K}{Ts+1}, \qquad （K：定常ゲイン，T：時定数）$$

$\mathcal{L}^{-1}\left[\frac{1}{s+a}\right] = e^{-at}$

一次システムの単位ステップ応答を考えよう。

(1) 入力 $u(t) = 1$ のラプラス変換 $U(s)$ を求める

$U(s) = \mathcal{L}[u(t)] = \frac{1}{s}$

(2) $Y(s) = P(s)U(s)$ を部分分数分解する

$$Y(s) = P(s)U(s) = \frac{K}{(Ts+1)s} = \frac{A}{s} + \frac{B}{Ts+1} = \frac{k}{s} - \frac{KT}{Ts+1}$$

$$A = \lim_{s\to 0} sY(s) = \lim_{s\to 0} \frac{K}{Ts+1} = K, \quad B = \lim_{s\to -\frac{1}{T}} (Ts+1)Y(s) = \lim_{s\to -\frac{1}{T}} \frac{k}{s} = -KT$$

(3) $Y(s)$ の逆ラプラス変換 $y(t)$ を求める

$\mathcal{L}^{-1}\left[\frac{k}{s}\right] = k, \quad \mathcal{L}^{-1}\left[\frac{k}{s+\frac{1}{T}}\right] = ke^{-\frac{t}{T}}$

$\therefore y(t) = k - ke^{-\frac{t}{T}} = k(1 - e^{-\frac{t}{T}})$

一次システムの**定常応答**，つまり $y_\infty = \lim_{t\to\infty} y(t)$ を調べると，

$$y_\infty = \lim_{t\to\infty} k(1 - e^{-\frac{t}{T}}) = k(1-0) = k$$

となる。なお，定常応答を調べるのに便利なラプラス変換の**最終値の定理**があり，それは

$$y_\infty = \lim_{s\to 0} sY(s)$$

というものである。この最終値の定理を使って一次システムの定常応答を調べると

$$y_\infty = \lim_{s\to 0} s\frac{K}{(Ts+1)s} = \lim_{s\to 0} \frac{K}{Ts+1} = \frac{K}{0+1} = K$$

となり，一致することがわかる。

7 Python-control を使ったステップ応答の描画

伝達関数の定義には `tf` 関数（tf は transfer function の略）を使う。例えば

$$P(s) = \frac{3}{2s+1} \tag{1}$$

を定義するなら以下のようにすればよい。

```
In[1]: import numpy as np
In[2]: import matplotlib.pyplot as plt
In[3]: from control.matlab import *
In[4]: n = [3]
In[5]: d = [2, 1]
In[6]: P = tf(n, d)
```

分子多項式の係数と分母多項式の係数をそれぞれリストで用意して伝達関数を定義している。定義した伝達関数のステップ応答を調べるには `step` 関数を使う。

```
In[7]: t = np.linspace(0, 10, 100)
In[8]: y, t = step(P, t)
In[9]: plt.plot(t, y)
       plt.show()
```

課題

(1) $K = 1$ で, $T = 1, 3, 5$ [s] の 3 通りの一次システムの時間応答を Python で重ねてプロットせよ。
- 横軸の範囲は 0 s から 20 s で 100 点
- グリッド線あり，凡例あり (`"T=1"`, `"T=3"`, `"T=5"`)，軸ラベルあり (`"t [s]"`, `"y"`)
- $T = 1$：青実線，$T = 3$：緑破線，$T = 5$：赤一点鎖線

(2) プロットしたグラフをもとに一次システムにおける時定数 T の意味を簡単に述べよ（もちろん厳密に述べてもよい)。

(3) 電気回路の RC 直列回路は時定数 RC の一次システムであることが知られている。収束を早めるためには抵抗 R や静電容量 C をどうすればよいだろうか。

提出方法　Jupyter Notebook で作成し，HTML にしてダウンロードして，Teams の課題タブから提出
提出期限　次回授業まで
ファイル名　“出席番号 2 桁_授業回_氏名.html”　（例）`00_02_KazukiSakai.html`

Jupyter Notebook の便利機能

- コマンドモード（枠が青色）で `[X]` を押せばそのセルが削除される
- コマンドモードで `[A]` を押せば上にセルが追加される (Above)
- コマンドモードで `[B]` を押せば下にセルが追加される (Below)
- ツールバーの早送りみたいな記号 (▷▷) を押すと，最初から実行し直してくれる。