総合実験 1 倒立振子第 2 回：DC モーターのシステム同定

2025 年 4 月 17 日

第 2 回：モーターのシステム同定

この講義 (第 2 回) では、DC モーターの特性を実験データから理解し、制御設計の基礎となる**システム同定**の手法を学びます。

システム同定とは？

- 入出力データに基づき、システムのモデルを決定するプロセス。
- モータのモデリングに不可欠⇒制御器設計に不可欠。

本日の目標

- 実験によるモーター応答データ収集。
- データに基づくパラメータ推定。
- 推定モデルの妥当性検証。

本日の流れ

1. 理論 (モデル, 推定法)
2. 実験 (データ収集)
3. データ処理・同定
4. 評価・検証
5. まとめ

理論：DC モーターの運動方程式

DC モーターの基本的な運動方程式（回転運動）は、トルクのバランスとして表されます。

$$J\frac{d^2\theta(t)}{dt^2} = \tau_m(t) - \tau_f(t) - \tau_l(t)$$

各項の意味：

- $J\frac{d^2\theta(t)}{dt^2}$: 慣性モーメント ($J$) × 角加速度 ($\ddot{\theta}$)。慣性によるトルク。*(J は通常未知)*
- $\tau_m(t)$: モーター発生トルク。電流 $I(t)$ に比例すると仮定 ($= k_T I(t)$)。k_T はトルク定数。*(k_T は通常未知, $I(t)$ は既知)*
- $\tau_f(t)$: 摩擦トルク。主に角速度 $\omega(t) = \dot{\theta}$ に依存。単純化のため粘性摩擦 $B\omega(t)$ を仮定。B は粘性摩擦係数。*(B は通常未知)*
- $\tau_l(t)$: 外部からの負荷トルク。(今回は無視、または J に含まれると考える)

上記を角速度 $\omega(t)$、角加速度 $\alpha(t)$ で書き直すと、

$$J\alpha(t) = k_T I(t) - B\omega(t)$$

これはモーターの挙動を記述する基本的な物理モデルである。

理論：同定のための線形モデル

測定可能な量 $\alpha(t)$ について整理すると、

$$\alpha(t) = \left(\frac{k_T}{J}\right) I(t) - \left(\frac{B}{J}\right) \omega(t)$$

となります。ここで、比率 $\frac{k_T}{J}$ と $\frac{B}{J}$ がモーターの特性を表す重要なパラメータとなります。

システム同定のための線形モデル表現:
推定したい未知パラメータを

$$a = \frac{k_T}{J}, \quad b = -\frac{B}{J}$$

とおくと、モデルは以下のシンプルな線形関係式で表せます。

$$\boxed{\underbrace{\alpha(t)}_{\substack{\text{角加速度} \\ \text{(測定/計算により既知)}}} = \underbrace{a}_{\substack{\text{パラメータ 1} \\ \text{(未知・推定対象)}}} \cdot \underbrace{I(t)}_{\substack{\text{電流} \\ \text{(入力/測定により既知)}}} + \underbrace{b}_{\substack{\text{パラメータ 2} \\ \text{(未知・推定対象)}}} \cdot \underbrace{\omega(t)}_{\substack{\text{角速度} \\ \text{(測定/計算により既知)}}}}$$

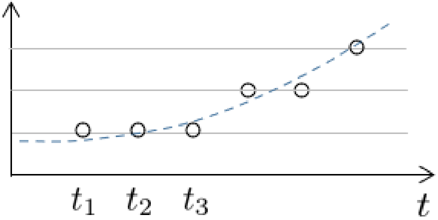
目標: 実験で得られる時系列データ (α, I, ω) を使って、この線形モデルにおける未知のパラ

理論：必要な信号とノイズ源、そして数値微分

我々のモデル $\alpha(t) = al(t) + b\omega(t)$ を使うには、角速度 $\omega(t)$ と角加速度 $\alpha(t)$ が必要です。

1. 信号の取得と課題:

- 角度 $\theta(t)$ はセンサー (エンコーダ) で測定される。
- センサー出力は通常、**離散的**な値であり、真の連続値との間に**量子化誤差**が生じる。
- $\omega(t), \alpha(t)$ は $\theta(t)$ から計算する必要がある。



2. 数値微分 (差分計算):

- 離散データから微分値を近似計算する中央差分:

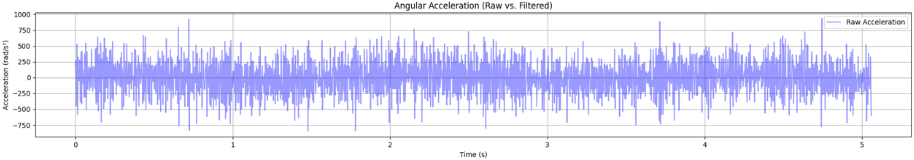
$$\omega[k] \approx \frac{\theta[k+1] - \theta[k-1]}{2\Delta t}, \alpha[k] \approx \frac{\omega[k+1] - \omega[k-1]}{2\Delta t}$$

Δt はサンプリング周期

理論：微分によるノイズ増幅とフィルタリング

問題点：数値微分はノイズを増幅する！

- センサーノイズや量子化誤差など、元の信号に含まれる高周波成分は、差分計算（微分）によってその影響が顕著になる。
- 特に二階微分である角加速度 $\alpha[k]$ は、ノイズの影響を非常に受けやすくなる。
- 下図（例）は、ノイズを含む信号とその微分（さらにノイズが増幅）を示す。



結果：ノイズの多い $\omega[k], \alpha[k]$ をそのまま使うと、パラメータ推定 (a, b) の精度が悪化。

解決策：ローパスフィルタリング

→フィルタ処理された $\omega[k], \alpha[k]$ を用いて、次のステップであるパラメータ推定を行う。

理論：最小二乗法への準備 (連立方程式)

Step 1: 各時刻での関係式 (再掲)
フィルタリング後のデータを用い、各時刻 t_k での関係式を考えます:

$$\alpha[k] \approx aI[k] + b\omega[k]$$

($\alpha[k], I[k], \omega[k]$ は既知、a, b は未知)

Step 2: 全データ点を連立方程式に
全データ点 ($k = 1, \ldots, N$) について、この関係式を書き並べると:

$$\begin{aligned} aI[1] + b\omega[1] &\approx \alpha[1] \\ aI[2] + b\omega[2] &\approx \alpha[2] \\ &\vdots \\ aI[N] + b\omega[N] &\approx \alpha[N] \end{aligned}$$

(N はデータ点の総数で、通常 $N \gg 2$ である)

理論：最小二乗法の行列表現

以下の行列 $\boldsymbol{X}$、ベクトル $\boldsymbol{\beta}$、ベクトル $\boldsymbol{y}$ を用いて

$$\boldsymbol{X}\boldsymbol{\beta} \approx \boldsymbol{y}$$

と簡潔に書ける。

$$\underbrace{\begin{bmatrix} I[1] & \omega[1] \\ I[2] & \omega[2] \\ \vdots & \vdots \\ I[N] & \omega[N] \end{bmatrix}}_{\substack{\boldsymbol{X}\ (N\times 2) \\ \text{計画行列} \\ \text{(既知)}}} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\substack{\boldsymbol{\beta}\ (2\times 1) \\ \text{パラメータ} \\ \text{(未知)}}} \approx \underbrace{\begin{bmatrix} \alpha[1] \\ \alpha[2] \\ \vdots \\ \alpha[N] \end{bmatrix}}_{\substack{\boldsymbol{y}\ (N\times 1) \\ \text{観測ベクトル} \\ \text{(既知)}}}$$

課題 (再掲):
実際のデータにはノイズが含まれるため、方程式 $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{y}$ を**厳密に**満たす $\boldsymbol{\beta}$ は通常存在しない。

理論：最小二乗解

Step 3: 誤差最小化によるパラメータ推定
方程式 $\boldsymbol{X}\boldsymbol{\beta} \approx \boldsymbol{y}$ において、「最も当てはまりの良い」未知パラメータ $\boldsymbol{\beta}$ を見つけるために、誤差の二乗和 $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$ を最小化する。

この最小化問題の解 $\hat{\boldsymbol{\beta}}$ は、以下の式（正規方程式の解）で与えられる。

$$\boxed{\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y}}$$

ここで、

- X^T は行列 X の**転置** (Transpose) 。
- $(X^T X)^{-1}$ は行列 $X^T X$ の**逆行列** (Inverse Matrix) 。
- $\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}$ が推定されたパラメータ。

実装上の注意: プログラムでは、数値安定性のため $(X^T X)^{-1}$ を直接計算せず、*QR* 分解や *SVD* (例: *'numpy.linalg.lstsq'*) を用いるのが一般的である。