

DC サーボモータと制御系

1026-30-8137

11b 多田 拓生^{*1}

2020 年 11 月 4 日

^{*1} tada.takumi.34w@st.kyoto-u.ac.jp

電気電子工学実習報告書

1026-30-8137

11b 多田 拓生

実験日

2020/11/6

担当教員

伊藤陽介先生, 持山志宇先生

実験 1

1 目的

DC サーボモータの周波数応答の測定を元に伝達関数を同定する。ここで、制御対象としての DC サーボモータにはドライバやタコジェネレータ、ポテンシオメータを制御対象として含む。周波数応答測定に基づく同定を通して動的システムの周波数特性に親しむ。

2 原理

DC サーボモータは図 1 のような等価回路で表現できる。 $e(t)$ は電機子に誘起される逆起電力なので、印加される電圧 $v(t)$ 、電機子に流れる電流 $i(t)$ 、電機子の抵抗 R 、インダクタンス L に対して次式が成立する。

$$v(t) = Ri(t) + L \frac{d}{dt}i(t) + e(t) \quad (1)$$

また、電機子の回転速度を $\omega(t)$ とすると、比例定数を k_E として逆起電力に比例するので

$$e(t) = k_E \omega(t) \quad (2)$$

が成立する。また電機子に作用する回転トルク $\tau(t)$ をすると、比例定数を k_T として電機子電流に比例するので

$$\tau(t) = k_T i(t) \quad (3)$$

が成立する。

ここで、電機子の慣性モーメントを J 、負荷トルクを τ_L 、電機子の回転に関する粘性摩擦係数を D とすると運動方程式より

$$\tau(t) - \tau_L(t) = J \frac{d}{dt} \omega(t) + D\omega(t) \quad (4)$$

が成立する。ここで、式 1、2、3、4 をそれぞれラプラス変換すると

$$V(s) = RI(s) + sLI(s) + E(s) \quad (5)$$

$$E(s) = k_E \Omega(s) \quad (6)$$

$$T(s) = k_T I(s) \quad (7)$$

$$T(s) - T_L(s) = sJ\Omega(s) + D\Omega(s) \quad (8)$$

となる。よってこれらの式から $I(s)$ 、 $E(s)$ 、 $T(s)$ を消去し、また電氣的時定数 $T_E = L/R$ 、機械的時定数 $T_M = JR/k_E k_T$ を用いて近似すると

$$\Omega(s) = \frac{1/k_E}{(1 + sT_M)(1 + sT_E)} \quad (9)$$

が得られる。したがって印加電圧と回転速度の間の伝達関数は近似的に二次遅れ系で表現できる。

ここで回転角 $\Theta(t)$ に対して

$$\Omega(s) = s\Theta(s) \quad (10)$$

が成立するはずなので、式 9 より、

$$\Theta(s) = \frac{1/k_E}{s(1 + sT_M)(1 + sT_E)} V(s) \quad (11)$$

が成立するので、したがって電機子印加電圧と回転角の間の伝達関数 $P(s)$ は

$$P(s) = \frac{1/k_E}{s(1 + sT_M)(1 + sT_E)} \quad (12)$$

とわかる。

今回の制御では検出器をしてタコジェネレータとポテンシオメータを用いる。タコジェネレータの外部端子間に生じる起電力を検出信号 y_d として測定するので、回転速度と検出信号には

$$y_d(t) = k'_E \omega(t) \quad (13)$$

が成立する。 k'_E は逆起電力定数である。また、ポテンシオメータは回転角に応じて変化する可変抵抗なので、 k_P を定数としてポテンシオメータの検出信号は

$$y_d(t) = k_P \theta(t) \quad (14)$$

となる。

今回の制御では、速度制御ではタコジェネレータ、位置制御ではポテンシオメータからフィードバックすることでそれぞれ図 2、図 3 のようなブロック線図で表現できるフィードバック制御を行う。制御対象には DC サーボモータ、ドライバ、操作器が含まれるので、速度制御に対応する制御対象の伝達関数を $P_V(s)$ 、位置制御に対応する制御対象の伝達関数を $P(s)$ とするとそれぞれ

$$P(s) = \frac{k_A k'_E / k_E}{(1 + sT_M)(1 + sT_E)} \quad (15)$$

$$P(s) = \frac{k_A k_P / k_E}{s(1 + sT_M)(1 + sT_E)} \quad (16)$$

となる。

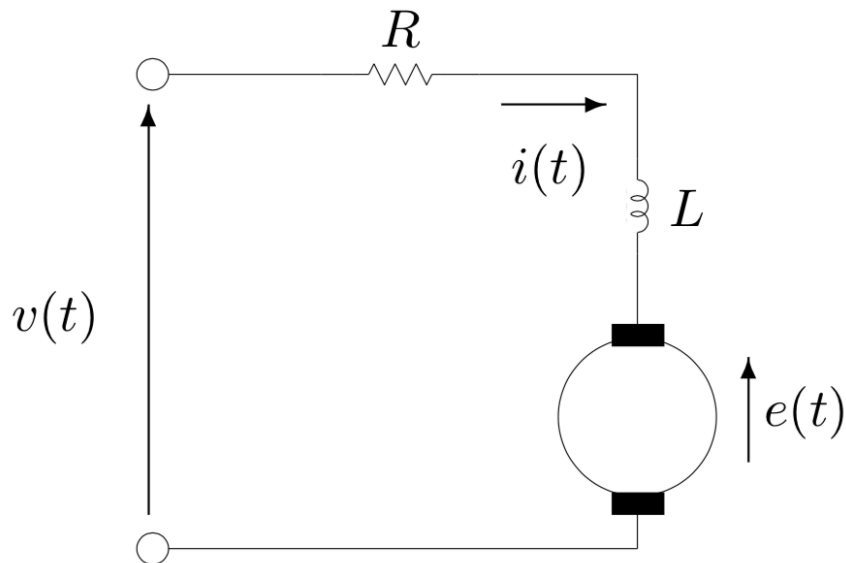


図 1 DC サーボモータの等価回路

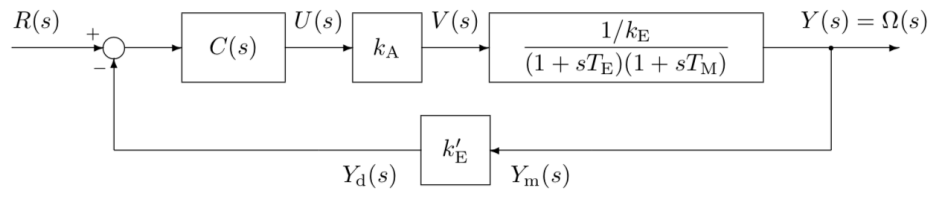


図 2 速度制御のブロック線図

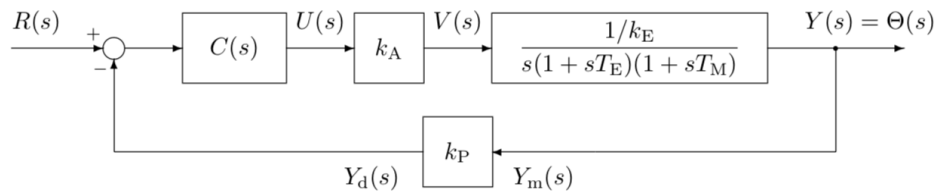


図 3 位置制御のブロック線図

3 方法

今回用いた器具を以下に示す。

- ファンクションジェネレータ：Z94575
- オシロスコープ：IWATSU DS-5110 B
- 電源 (小)：KENWOOD PR18-1.2A
- 電源 (大)：TEXIO PS40-10A (Z000323513)
- ブレッドボード：5 番
- DC サーボ：5 番
- サーボモータドライバ：MS100T05
- ポテンショメータ：J40S
- カップリング：アサ電子工業製
- コンバータ：SUW3 0515

今実験に用いた装置構成図を図 4 および図 5 に示す。図 4 は速度制御を行うとき、図 5 は位置制御を行うときに用いた。

まず、速度制御を行った時の伝達関数 $P_V(s)$ の同定について説明する。

図 4 のファンクションジェネレータの出力電圧を制御対象への入力、タコジェネレータの出力電圧を制御対象からの出力とみなして、入力周波数を 0.2Hz から 100Hz の間で 20 点ほど変化させながら電圧をオシロスコープで計測した。ここで、制御対象の入力、出力とみなしたところはそれぞれ図 2 における $C(s)$ への入力信号、 $Y_d(s)$ とみなせる。

次に、位置制御を行った時の伝達関数 $P(s)$ の同定について説明する。

図 3 のファンクションジェネレータの出力電圧を制御対象への入力、ポテンショメータの出力電圧を制御対象からの出力とみなして、入力周波数を 2Hz から 20Hz の間で 6 点ほど変化させながら電圧をオシロスコープで計測した。ここで、制御対象の入力、出力とみなしたところはそれぞれ図 3 における $C(s)$ への入力信号、 $Y_d(s)$ とみなせる。

それぞれの計測結果からボード線図を描き、伝達関数を同定する。

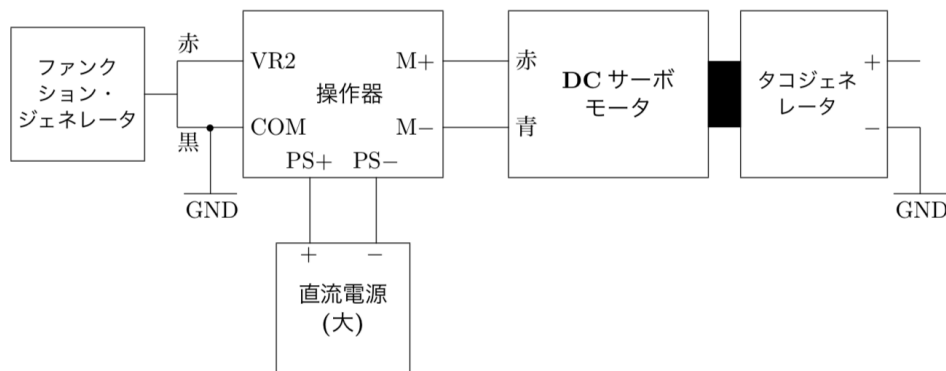


図 4 実験 1-1 の装置構成図

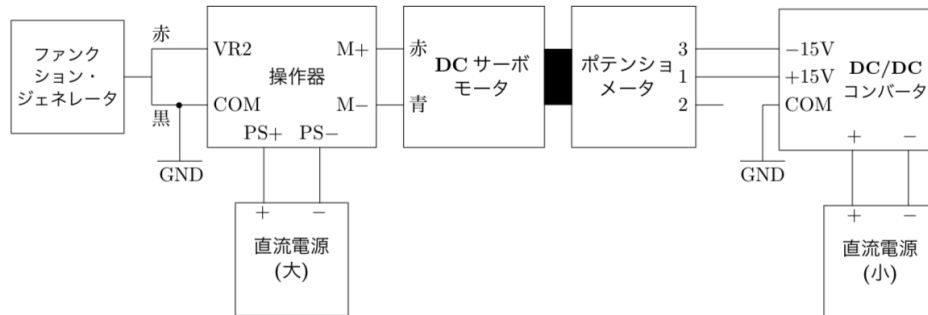


図 5 実験 1-2 の装置構成図

4 実験結果

位置制御によって得られたデータは、式 15、16 を元に得られる関係式

$$P(s) = \frac{k_P}{s k'_E} P_V(s) \quad (17)$$

から得られる変換式

$$|P_V| = \frac{\omega k'_E}{k_P} |P| \quad (18)$$

$$\arg P_V = \arg P + 90 \quad (19)$$

を用いて位相制御の計測値を速度制御のものと同様に扱えるように変換し、ボード線図を描いたものを図 6 に示す。

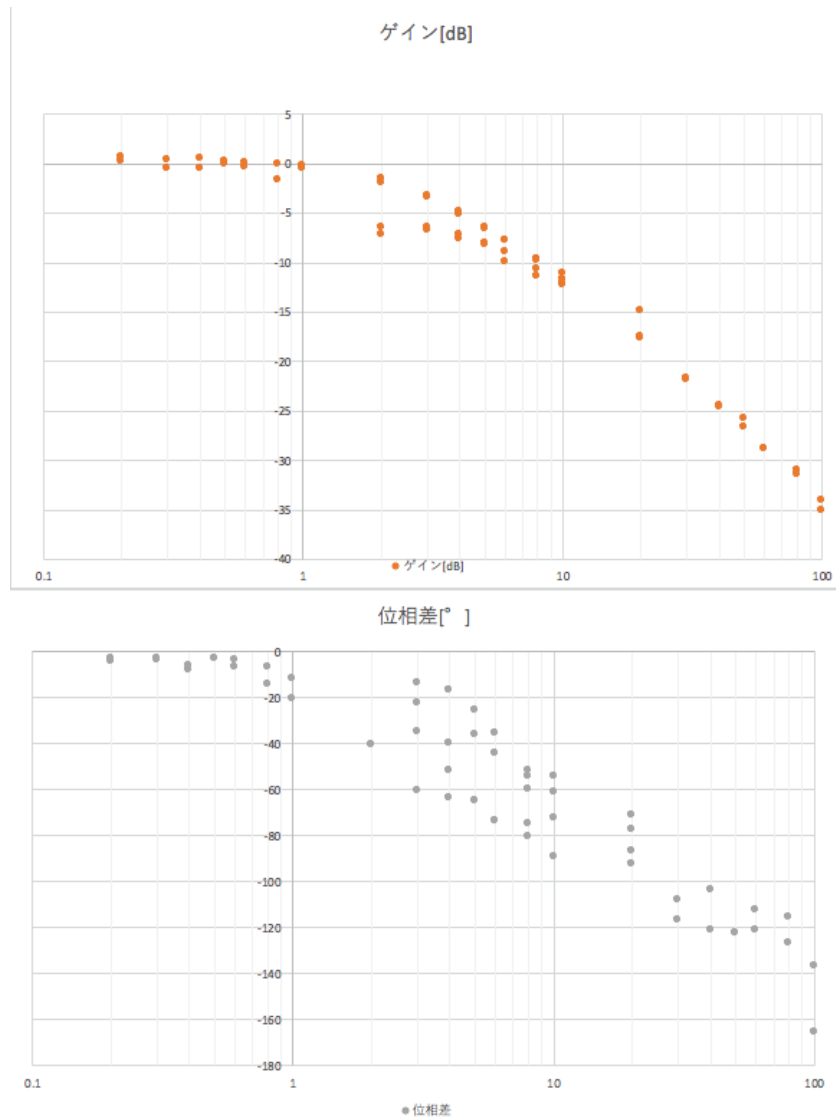


図 6 制御対象のボード線図

5 考察

まず、1 回転ポテンシオメータに対する k_p の値を考える。ポテンシオメータの両端子にはそれぞれ -15V と 15V の電源を接続した。1 回転ポテンシオメータは 1 回転する間に -15V から 15V へ変化するので、式 13 を参考に考えると k_p の値は $30/2\pi$ になると考えた。

次に逆起電力定数 k'_E の算出法について考える。

1 回転ポテンシオメータを DC サーボモータに取り付けた状態で、DC サーボモータ・ドライバに一定の直流電圧を与えて DC サーボモータを定速回転させ、この状態でのタコジェネレータ及びポテンシオメータの出力電圧波形を観測することによっても測定できる。その原理は、式 15、16 を参照して考えると、与えられた状況下では $P(s)$ と $P_V(s)$ の差異が生じるのはそれぞれの定数 k_A 、 k'_E によってだけである。したがってそれぞれの出力電圧波形の比をとると k_A/k'_E が求まるので、そこに k_A を代入すれば k'_E が求まる。

最後に、 K 、 T_E 、 T_M の同定を行った。まず、ボード線図の理論的な概形を考える。

式 15 を参考に理論的なボード線図の概形を描いたものが、図 7 に示す。これは、一次遅れのボード線図を元に考えた。これを元に、図 6 の特にゲインの方を使いそれぞれに値を同定した。実際の同定には、グラフを印刷して線を引いて求めた。求めた値はそれぞれ

$$K = 1.03 \quad (20)$$

$$T_M = 0.0909 \quad (21)$$

$$T_E = 0.00132 \quad (22)$$

となった。実際この値を用いて求めたグラフを実測値を元にしたグラフに重ねると図 8、9 のようになる。高周波部分は補助線が引きづらかったため、特に T_E の値の推定が難しく、その結果がゲイン図の高周波域での実測値とのズレが出たのだと考えた。

以後は値が必要になった時はこの値を用いることにする。

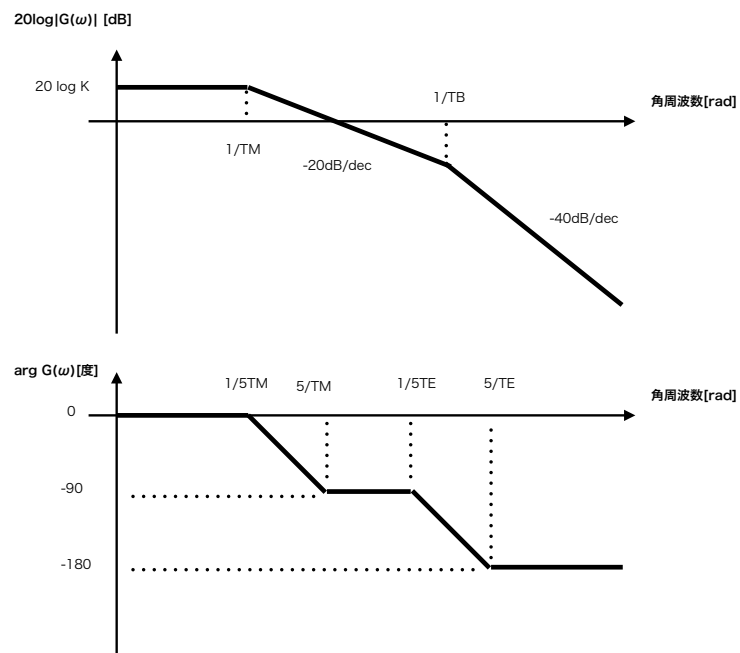


図 7 理論的なボード線図

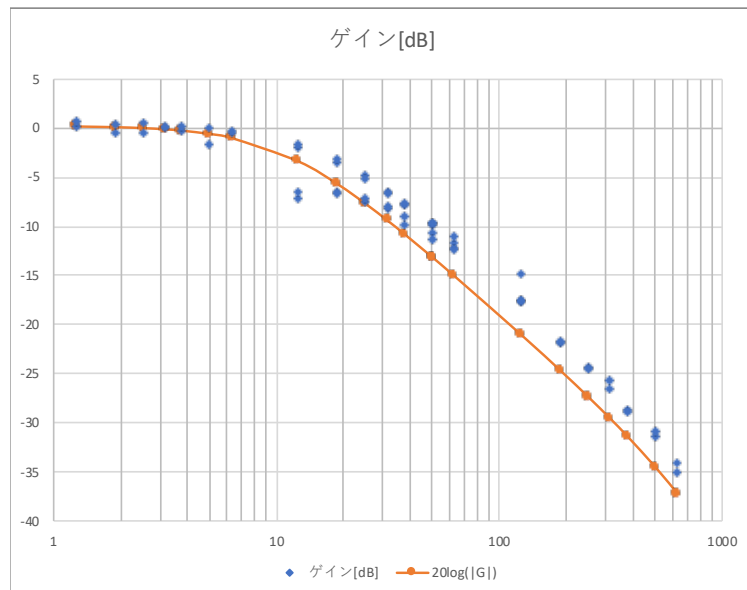


図 8 推測値と実測値

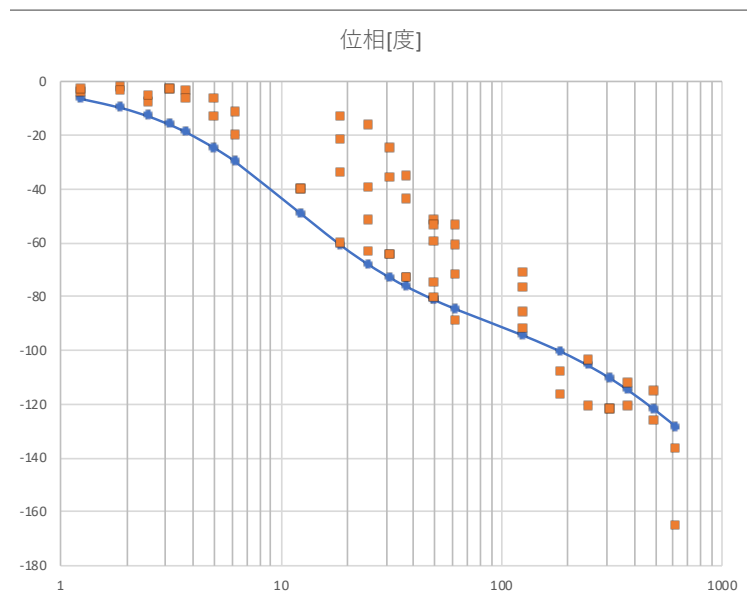


図 9 推測値と実測値 2

実験 2

6 目的

制御用アナログ回路を作成しそれを各制御に用いて測定する。その結果を測定し、安定性や定常偏差、応答について議論する。

7 原理

今実験では、制御用アナログ回路は図 10 および 11 に示す構成のものを使用する。
まず、図 10 において、

$$v_{\text{out}}(t) = K_C(v_{\text{ref}}(t) - v_{\text{in}}(t)) \quad (23)$$

の関係が成り立つ。この式を導出する。

オペアンプの + での電圧を V_+ 、-での電圧を V_- とすると、

$$\frac{v_{\text{in}} - V_-}{R_1} = \frac{V_- - v_{\text{out}}}{R_2} \quad (24)$$

$$V_+ = \frac{R_2}{R_1 + R_2} v_{\text{ref}} \quad (25)$$

が成立し、仮想短絡のことを考えると $V_+ = V_-$ が成立するので、

$$v_{\text{out}}(t) = \frac{R_2}{R_1}(v_{\text{ref}}(t) - v_{\text{in}}(t)) \quad (26)$$

が得られる。したがって $K_C = \frac{R_2}{R_1}$ が得られる。

次に、図 11 において

$$C(s) = K_C \frac{\alpha(1 + sT)}{1 + s\alpha T} \quad (27)$$

が成立する。

この式を導出する。まず回路の左側は 10 と同じなので、図中の境目では

$$V_X = -\frac{R_2}{R_1}(V_{\text{in}} - V_{\text{ref}}) \quad (28)$$

となる。オペアンプの仮想短絡を考えると、

$$\frac{V_X}{\frac{R_3}{1+sC_1}} = \frac{V_{\text{out}}}{\frac{R_4}{1+sC_2}} \quad (29)$$

となるので、これを変形して

$$C(s) = \frac{R_2 R_4}{R_1 R_3} \frac{1 + s C_1 R_3}{1 + s C_2 R_4} = \frac{C_2 R_2}{C_1 R_1} \frac{\frac{C_2 R_4}{C_1 R_3} (1 + s C_1 R_3)}{1 + s \frac{C_2 R_4}{C_1 R_3} C_1 R_3} \quad (30)$$

したがって

$$K_C = \frac{C_2 R_2}{C_1 R_1} \quad (31)$$

$$\alpha = \frac{C_2 R_4}{C_1 R_3} \quad (32)$$

$$T = C_1 R_3 \quad (33)$$

が算出できる。

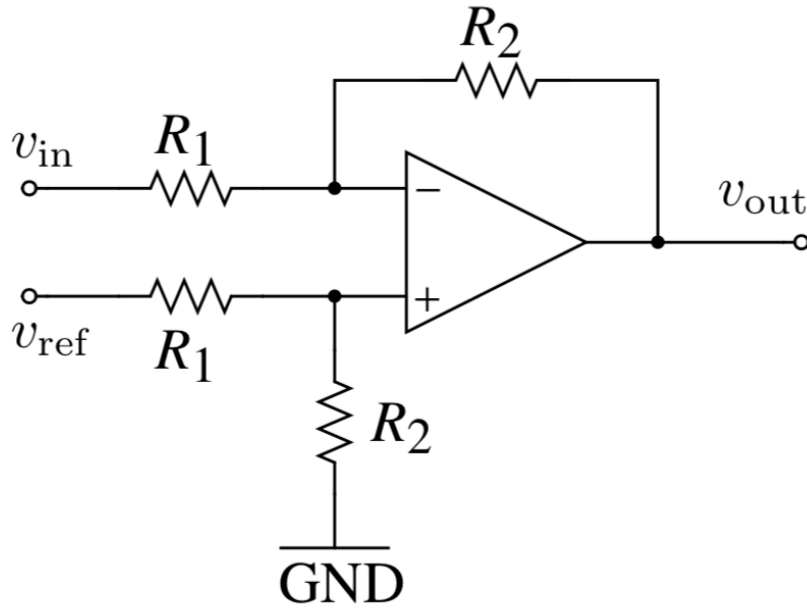


図 10 制御用アナログ回路 (a)

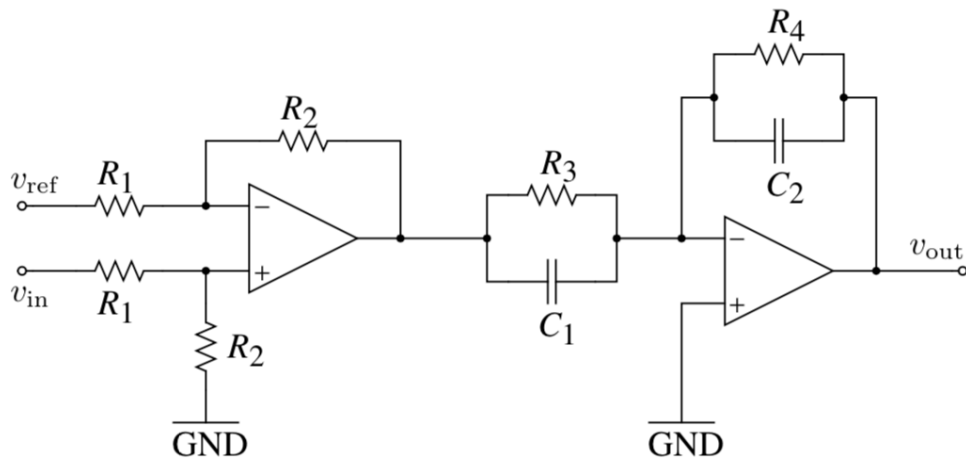


図 11 制御用アナログ回路 (b)

8 実験

上で示した制御回路 (a)、(b) を用いて周波数特性を測定し、そのボード線図を書く。制御回路は上の回路図を参考に回路を組み、全体の測定装置は図 12 のように組んだ。そして周波数を 0.2 500Hz まで変化させながら周波数特性を測定した。

なお、回路に用いた素子は (a) の測定装置では $R_1 = 20\text{k}\Omega$ 、 $R_2 = 300\text{k}\Omega$ を用いた。また、(b) においては $R_1 = 470\text{k}\Omega$ 、 $R_2 = 100\text{k}\Omega$ 、 $R_3 = 2\text{k}\Omega$ 、 $R_4 = 7.5\text{k}\Omega$ 、 $C_1 = 10\mu\text{F}$ 、 $C_2 = 1\mu\text{F}$ を用いた。

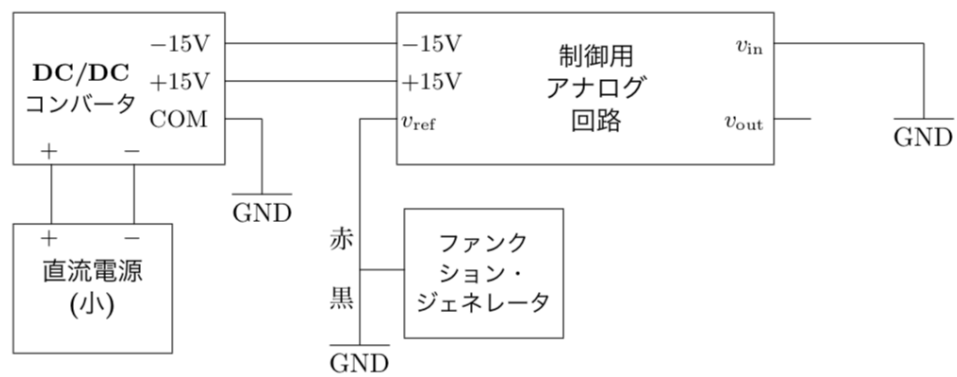


図 12 実験 2 の測定装置

9 結果

上で求めた理論値を用いて描いた理論グラフと測定によって得られた値をプロットしたものをそれぞれ図 13、14、15、16 に示す。

ここで用いた測定値は、与えられたものである。

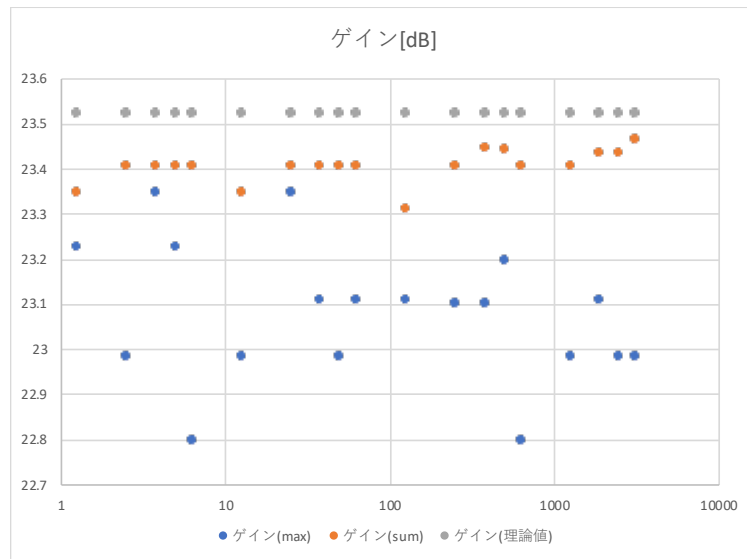


図 13 ゲイン (a) の測定値と理論値

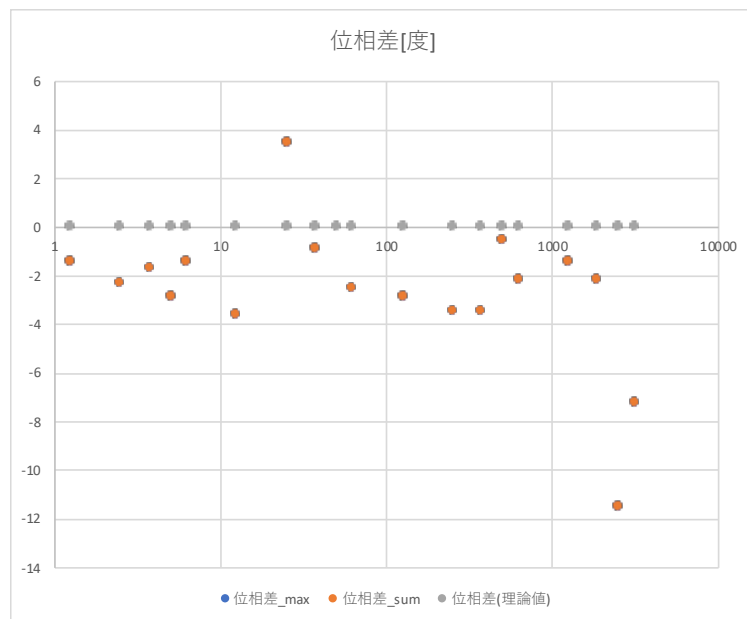


図 14 位相差 (a) の測定値と理論値

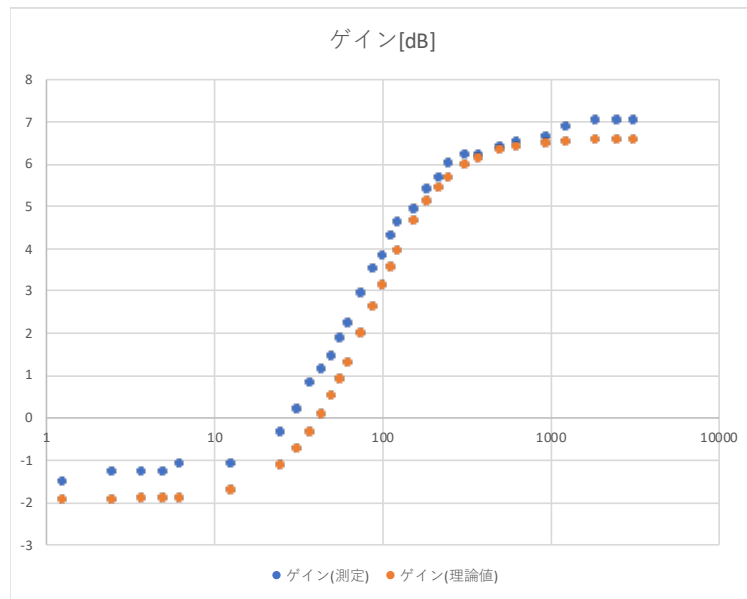


図 15 ゲイン (b) の測定値と理論値

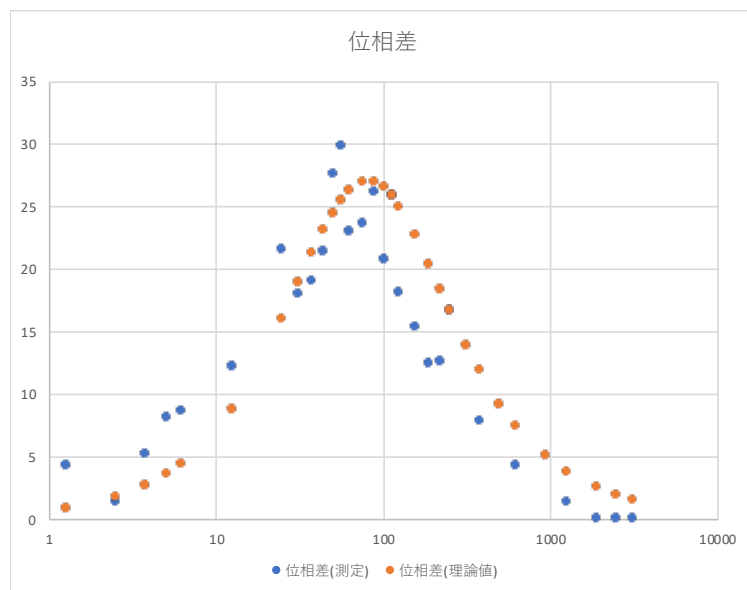


図 16 位相差 (b) の測定値と理論値

10 考察

今回の測定値はオシロスコープから出力された csv ファイルとして与えられた。それを処理するために Rust で書いたプログラムを用いた。そのコードをソースコード 1 に示す。

このコードでは、ふた通りの方法でゲインを求めたのち、正弦波が 0 を示した点および 0 付近を示した点を列挙して、実際のグラフと対比させながら位相差を求めた。ゲインについて、まず全データの最大値、最小値を入力、出力それぞれについて求め、比を求めた。しかし、この方法は最大値を示す一点のみから求めるのでノイズに弱いと考えた。二つ目の方法として、全データを絶対値をとって足し込んだのち、測定間隔をかけ積分値とみなし、全データの観測時間をファイルの最初の行と最後の行から求め振幅を計算してゲインを求めた。

この方法では低周波での観測において、位相差が 0 付近のデータを示すデータの数が少ないと正しくデータが取れない可能性があるため目視で求めた測定データを図 17、18、19、20 に示すが、今回の計測ではむしろ位相差の高周波領域において上手く観測できていなかったことがわかった。これは、プログラムによる処理を高周波領域から始めたこと、および観測するにつれて実際のデータに合わせてプログラムのパラメータを調整したことによって生じた観測誤差だと考えた。

これらのグラフより、算出した各パラメータの値は正しいと結論づけた。

ソースコード 1 csv_manager.rs

```
use competitive::prelude::*;

#[argio()]
// csv ファイルを読み取る
fn main(input: [(f32, f32, f32); 600]) {
    // 微小時間の幅を求める
    let dx = input[23].0.clone() - input[22].0.clone();
    // 配列の用意
    let mut v = Vec::new();
    let mut w = Vec::new();
    for i in 0..600 {
        // ch1, ch2 のデータを一旦こちらにコピー
        v.push(input[i].1.clone());
        w.push(input[i].2.clone());
    }
    // 振幅の最大値を求める
    let max_ch1 = v.clone().into_iter().fold(0.0, f32::max);
```

```

let max_ch2 = w.clone().into_iter().fold(0.0, f32::max);
println!("ch1's max = {}", max_ch1);
println!("ch2's max = {}", max_ch2);

// 全ての値を足しこむ変数を用意
let mut sum1 = 0.0;
let mut sum2 = 0.0;
for i in 0..600 {
    // 全て絶対値をとって足しこむ
    sum1 += v[i].clone().abs();
    sum2 += w[i].clone().abs();
}
// 振幅を計算
let amp1 = dx * sum1 * 3.14 / 2.0
    / (input[599].0.clone() - input[0].0.clone());
let amp2 = dx * sum2 * 3.14 / 2.0
    / (input[599].0.clone() - input[0].0.clone());
println!("ch1::sum1: {}", amp1);
println!("ch2::sum2: {}", amp2);

// ゲインを出力
println!("rate = {}", max_ch2 / max_ch1);
println!("rate2 = {}", sum2 / sum1);

println!("");
println!("");

// こっから下で $-3e-2 \sim 3e-2$ の間に入る点をピックアップ
// 特に丁度0になった点はメッセージを短くして目立つようにした
for i in 0..600 {
    if input[i].1 == 0.0 {
        println!("ch1 = 0: {}, {}",
            input[i].0.clone(), input[i].1.clone());
    } else if input[i].1 >= -3.00e-2

```

```

        && input[i].1 <= 3.00e-2
    {
        println!(
            "maybe ch1 = 0: {}, {}",
            input[i].0.clone(),
            input[i].1.clone()
        );
    }
    if input[i].2 == 0.0 {
        println!("ch2 = 0: {}, {}",
            input[i].0.clone(), input[i].2.clone());
    } else if input[i].2 >= -3.00e-2
        && input[i].2 <= 3.00e-2
    {
        println!(
            "maybe ch2 = 0: {}, {}",
            input[i].0.clone(),
            input[i].2.clone()
        );
    }
}
}

```

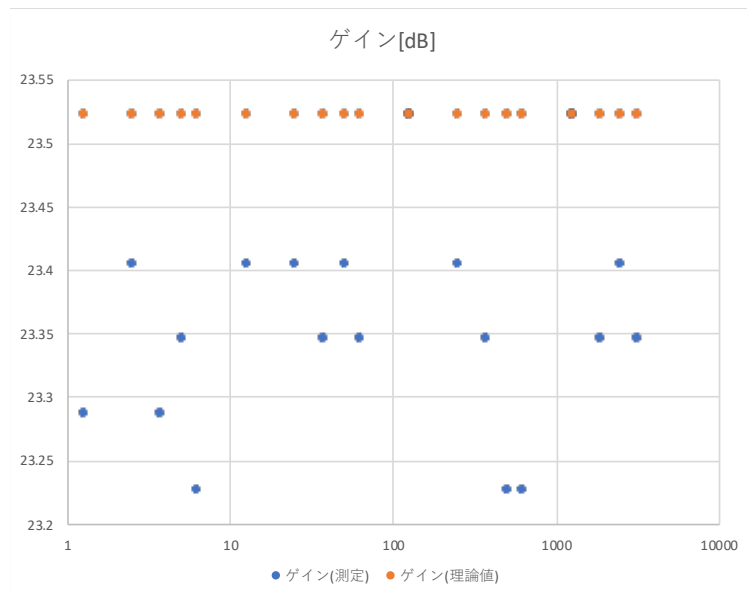


図 17 ゲイン (a) の測定値と理論値 (目視)

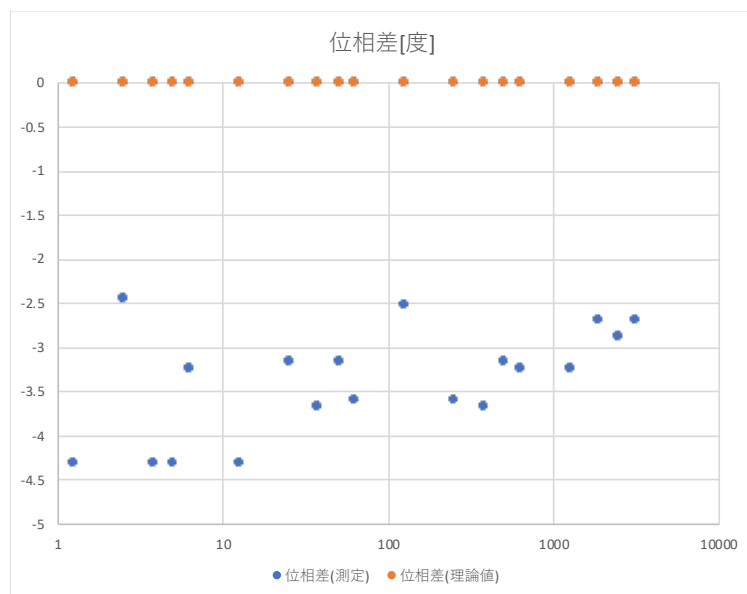


図 18 位相差 (a) の測定値と理論値 (目視)

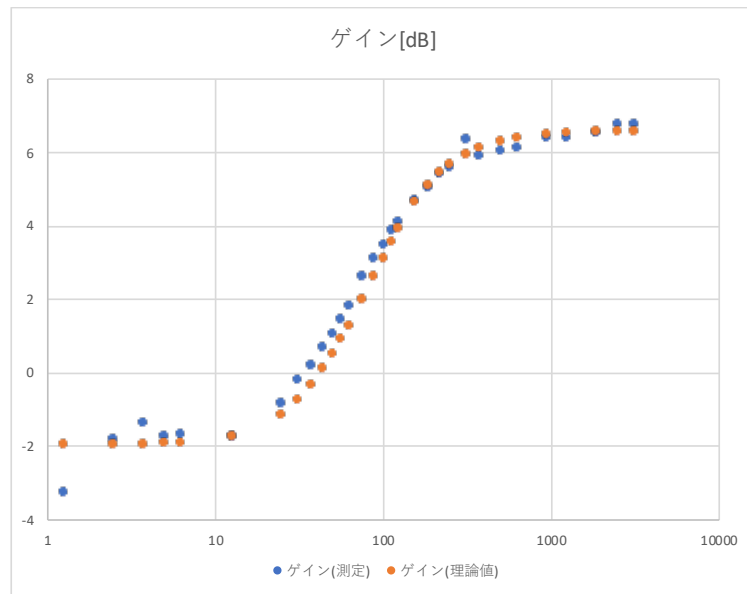


図 19 ゲイン (b) の測定値と理論値 (目視)

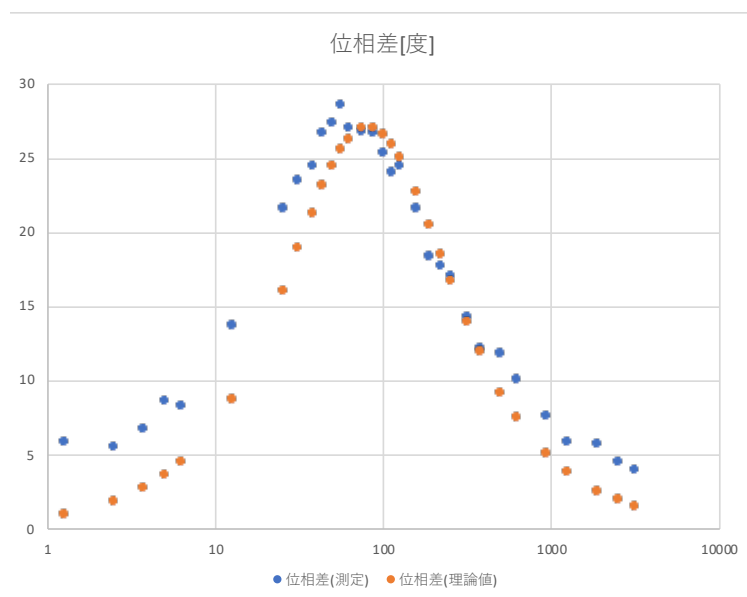


図 20 位相差 (b) の測定値と理論値 (目視)

実験 3

11 目的

実験 2 で作成したアナログ制御回路を用いて DC サーボモータの速度制御、位置制御をそれぞれ行い、安定性と定常偏差について調べる。また実験結果と Octave を用いたシミュレーションを比

較する。

12 原理

今実験で用いた回路のボード線図は、図 21 のようなブロック線図で表現できる。

したがって、制御系の安定性を調べるには

$$\frac{C(s)P(s)}{1 + C(s)P(s)} \quad (34)$$

についてラウスやナイキストの安定判別を行えば良い。また、定常偏差の伝達関数は

$$\frac{1}{1 + C(s)P(s)} \quad (35)$$

として得られる。

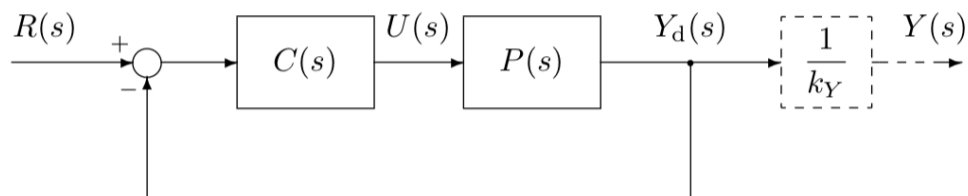


図 21 ブロック線図 (概略図)

13 実験

アナログ制御回路には、実験 2 で作成したものをを用いた。このアナログ制御回路を用いて DC サーボモータの速度制御および位置制御を行う。安定性を先に判別し、安定な組み合わせに関してのみステップ応答を測定した。

また、Octave を用いて理論的なグラフを描き、実験結果を重ねてステップ応答を比較した。

14 結果

まず (a)、(b) の制御回路を用いて速度制御、位置制御をした時の安定性をラウスの安定判別法を用いて判別した。そして安定な制御系は最終値公式を用いて定常偏差を計算した。

ラウスの安定判別法は、式 34 の分母 $A(s)$ について行う。

まず (a) の制御回路を用いて速度制御した場合の分母は、

$$A_1(s) = 1.117 \times 10^{-4}s^2 + 9.13 \times 10^{-2}s + 16.5 \quad (36)$$

となる。この式は 2 次式であり、係数が全て正なのでこの制御は安定であると考えた。この制御系のステップ応答の定常偏差を最終値公式を用いて求めると

$$\lim_{s \rightarrow 0} \frac{1}{1 + C(s)P(s)} = \lim_{s \rightarrow 0} \frac{1}{1.17 \times 10^{-4}s^2 + 9.13 \times 10^{-2}s + 16.45} = 0.061 \quad (37)$$

次に (a) の制御回路を用いて位置制御した場合の分母は

$$A_2(s) = 3.35 \times 10^{-6}s^3 + 2.61 \times 10^{-3}s^2 + 2.86 \times 10^{-2}s + 73.7 \quad (38)$$

となる。これに対しラウスの安定判別を行うと

行	1	2
1	3.35×10^{-6}	2.86×10^{-2}
2	2.61×10^{-3}	73.65
3	-6.59×10^{-2}	
4	73.65	

となる。ここで 3 行 1 列目で負の数が見れるので、この制御系は不安定であると考えた。

次に、(b) の制御回路を用いて速度制御する場合を考えると

$$A_3(s) = 8.78 \times 10^{-7}s^3 + 8.02 \times 10^{-4}s^2 + 1.15 \times 10^{-1}s + 1.82 \quad (39)$$

となる。これに対しラウスの安定判別法を用いると

したがってこの制御系は安定であると考えた。この制御系の定常偏差を求めると

$$\lim_{s \rightarrow 0} \frac{1}{1 + C(s)P(s)} = \frac{1 \times 1.03}{1.82} = 0.57 \quad (40)$$

最後に (b) の制御回路を用いて位置制御する場合を考えると

行	1	2
1	8.78×10^{-7}	1.15×10^{-1}
2	8.02×10^{-4}	1.82
3	1.13×10^{-1}	
4	1.82	

$$A_4(s) = 2.51 \times 10^{-8} s^4 + 2.29 \times 10^{-5} s^3 + 2.82 \times 10^{-3} s^2 + 1.07 \times 10^{-1} s + 3.92 \quad (41)$$

となる。これに対しラウスの安定判別法を用いると

行	1	2	3
1	2.51×10^{-8}	2.82×10^{-3}	3.92
2	2.29×10^{-5}	0.107	
3	2.70×10^{-3}	3.92	
4	1.82		
5	3.92		

となる。したがって、この制御系も安定であると考えた。この制御系の定常偏差は

$$\lim_{s \rightarrow 0} \frac{1}{1 + C(s)P(s)} = \frac{0}{3.91} = 0 \quad (42)$$

と求まる。

以上の結果は、実験時に実際に制御させ安定かどうかを現象として判定した結果と合致する。

以上の結果を元に、制御回路 (a) での速度制御および制御回路 (b) での位置制御を行い、実測値と計算結果を比較したグラフを図 22、23、24 に示す。制御回路 (b) による制御は理論値に近い結果が得られたが、制御回路 (a) による制御は期待通りの結果にならなかった。

またステップ応答の結果と理論値を Octave を用いて重ねたものをそれぞれ図 25、26、27、28、29、30 に示す。なお、制御回路 (a) による速度制御の v_{out} はオシロスコープからデータを取り出す段階でデータが壊れてしまい、描画できなかった。そのため計算結果に基づくシミュレーションのみを描画した。

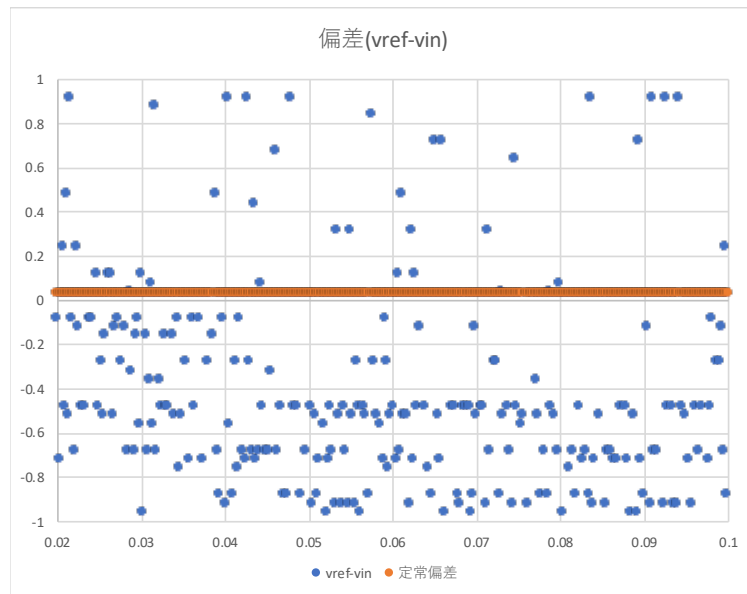


図 22 制御回路 (a) の速度制御

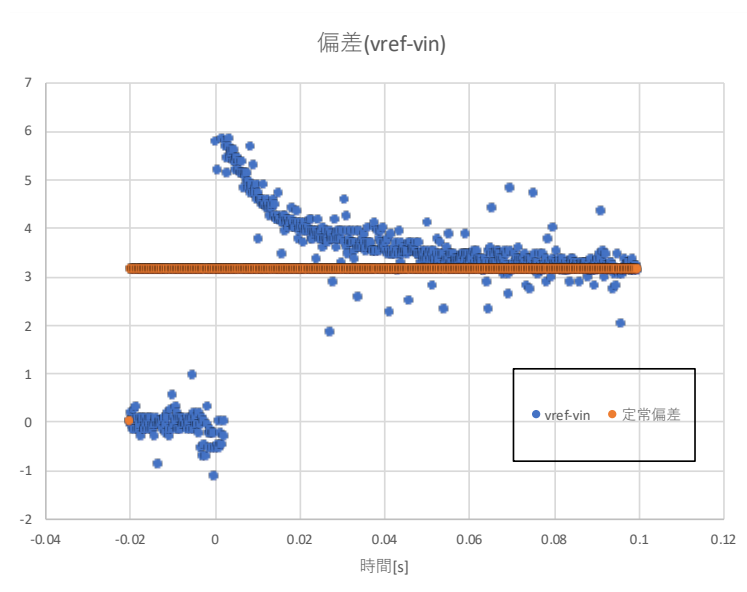


図 23 制御回路 (b) の速度制御

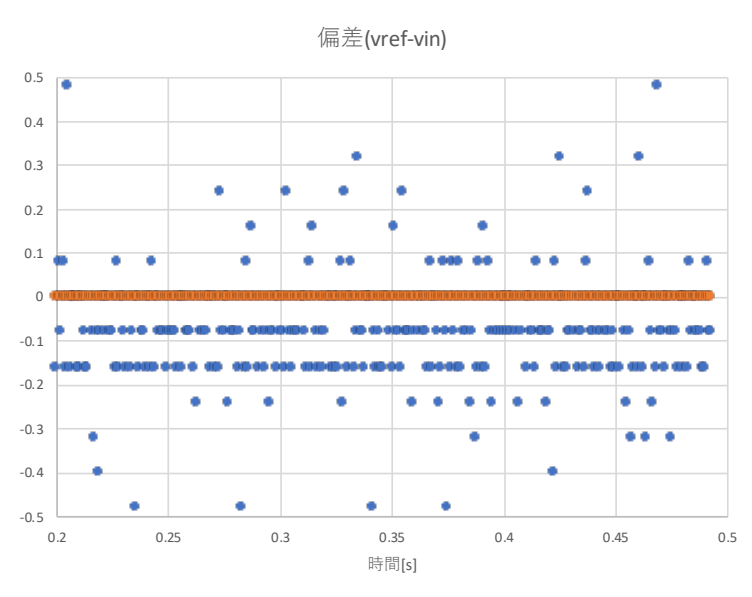


図 24 制御回路 (b) の位置制御

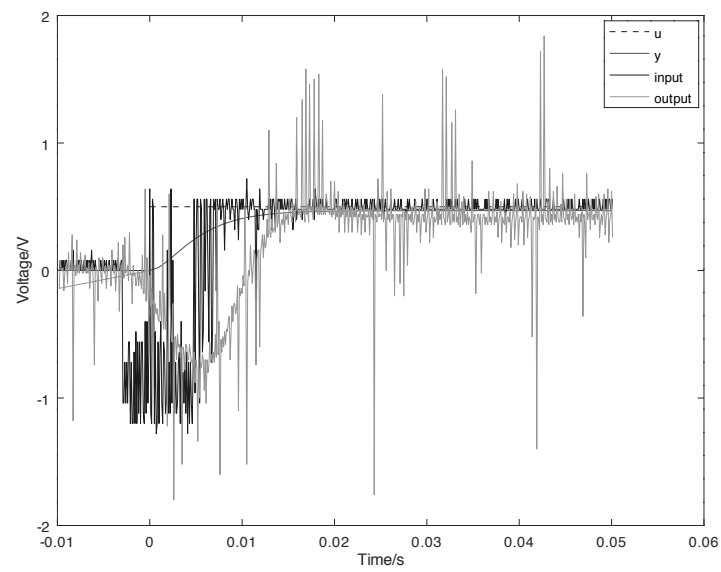


図 25 (a) の速度制御のステップ応答 v_{in}

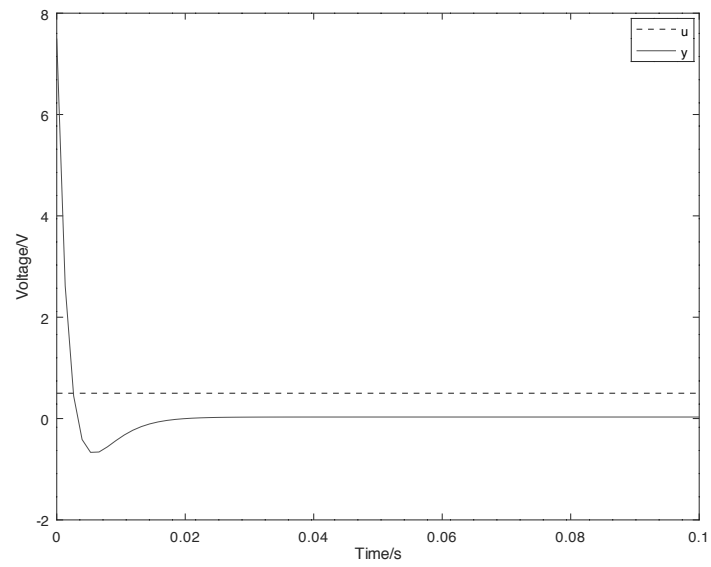


図 26 (a) の速度制御のステップ応答 v_{out}

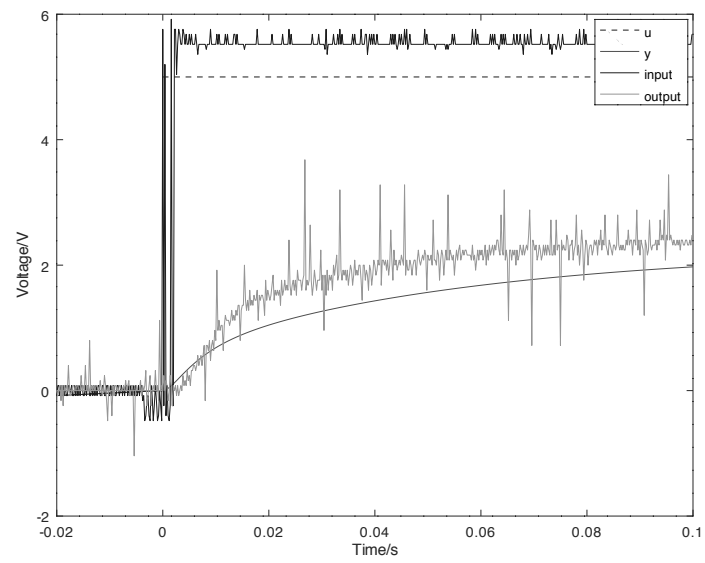


図 27 (b) の速度制御のステップ応答 v_{in}

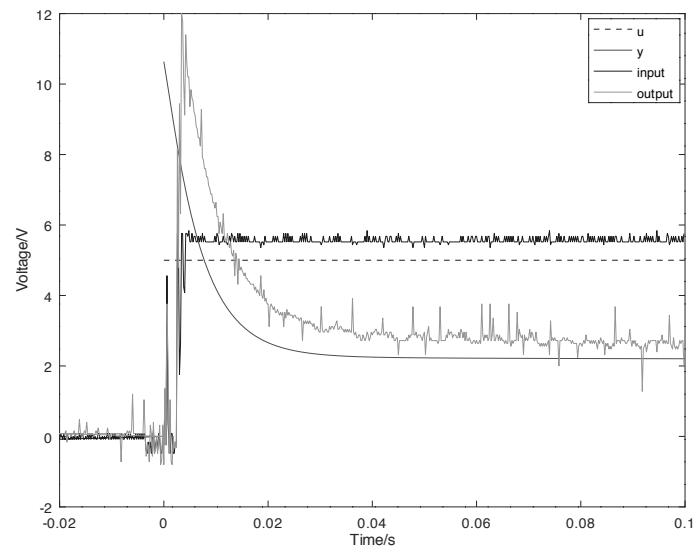


図 28 (b) の速度制御のステップ応答 v_{out}

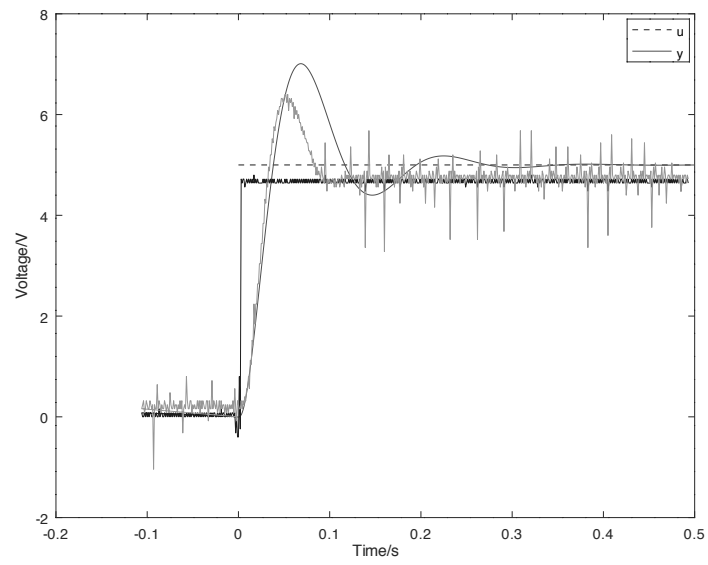


図 29 (b) の位置制御のステップ応答 v_{in}

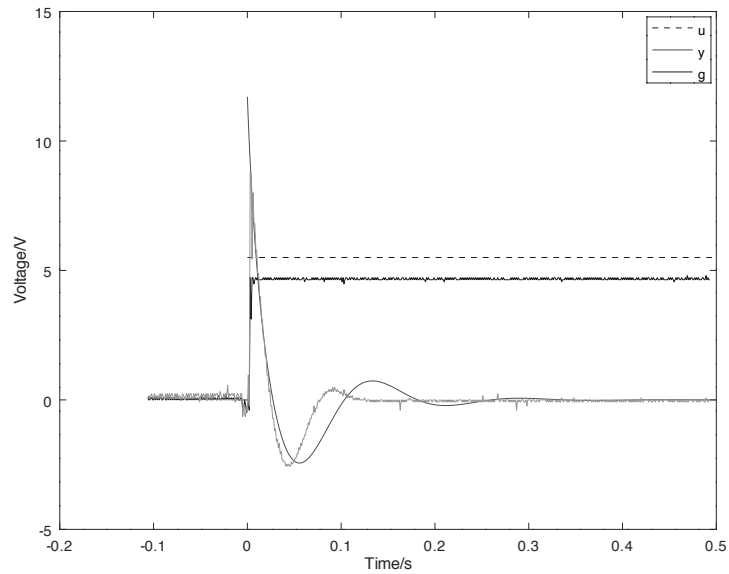


図 30 (b) の位置制御のステップ応答 v_{out}

15 考察

まず定常偏差の計算結果と実測値が制御回路 (a) において一致が見られなかったのは、目標値の設定電圧が低かったことが原因として考えられる。比例ゲイン (a) では今回用いた素子により 15 倍のゲインが得られるものであったが、オシロスコプの測定範囲を考慮して目標電圧が 0.5V を設定した。したがって、電圧が低い分期待通りの制御ができなかったのだと考えた。

次にステップ応答の理論値と実測値の比較だが、制御回路 (a) の速度制御以外では、概ね理論値と実測値が一致している。しかし、差動増幅器による速度制御では初めに電圧低下が起こったため、理論値と実測値が一致していない。これは、初めに高い電圧をかけた時に、モーターの慣性によって逆向きに誘導された電流によって生じた逆向きの起電力なのではないかと考えた。これは (b) においては間に挿入されたコンデンサによって緩和されているのではないかと考えた。

これを補填するために、Octave 上で逆向きのステップ信号を入力して理論値を計算し直したグラフを 31、そのためのコードをソースコード 2 に示す。これにより理論値と実測値が近い結果になった。

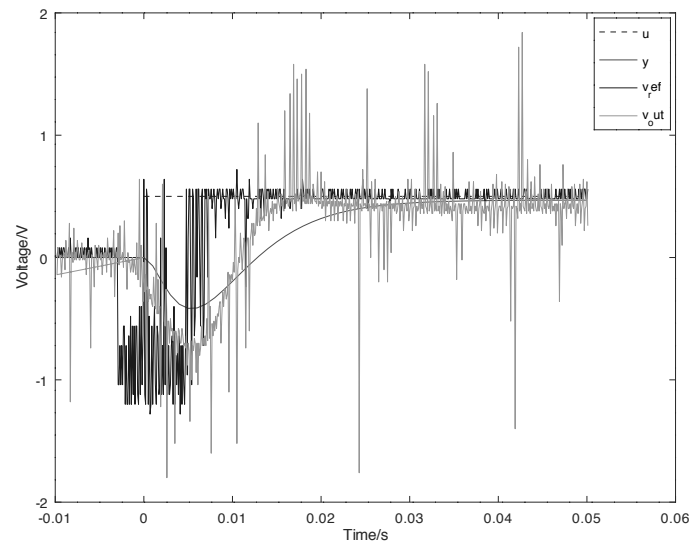


図 31 入力を補填したステップ応答

ソースコード 2 vain_2.m

```
pkg load control;

clear;

R1=20e3;
R2=300e3;
K=1.03;
TE=1.3267e-3;
TM=0.0909;

A=csvread("1023_01in.csv");

L=0.010;
stepsize=0.5;

G=tf([K*R2], [R1*TE*TM R1*(TE+TM) R1+K*R2]);
G_neo=G*tf([-L/2,1], [L/2,1]);
```



```

[y_unit1 t]=step(G,0.05);
y1=y_unit1*stepsize*(-1);

[y_unit2 t]=step(G_neo,0.05);
y2=y_unit2*stepsize*2;

y=y1+y2;

u=ones(length(t),1)*stepsize;

figure
plot(t,u,'b--',t,y,'r-',
      A(:,1), A(:,2), 'b-', A(:,1), A(:,3), 'g-')
legend('u', 'y', 'v_{ref}', 'v_{out}');
xlabel('Time/s');
ylabel('Voltage/V');

```

参考文献

- [1] 電気電子工学実習テキスト. 2020.
- [2] 北野正雄. 『電子回路の基礎』.