

# ステッピングモーターの制御に関する 実験的検証及び調査

名古屋大学情報学部コンピュータ科学科 3 年  
堀内颯太 ( 学籍番号 : 101730331 )  
horiuchi.sota@e.mbox.nagoya-u.ac.jp

実験日 2019 / 4 / 18

## 目次

1	初めに	2
2	実験 1：ステッピングモーターの運転	3
2.1	目的・概要 . . . . .	3
2.2	実験方法 . . . . .	3
2.3	実験結果 . . . . .	9
2.4	考察 . . . . .	10
3	調査課題 1：ステッピングモータの電流の立ち上がりの改善	12
4	調査課題 2：ステッピングモーターの特徴	13
5	調査課題 3：ステッピングモーターの台形的速度制御	18
5.1	実験方法 . . . . .	18
5.2	実験結果 . . . . .	21
5.3	考察 . . . . .	21
6	まとめ	21

## 1 初めに

本レポートでは、アクチュエータの一つであるステッピングモーターについて PC/AT 互換機を用いた制御を理解し、その特性を検証する。一般にアクチュエータとは入力されたエネルギーもしくはコンピュータが出力した電気信号を物理的運動に変換する機械・電気回路を構成する機械要素のことで、能動的に作動または駆動する機械である [1]。特に電気的なエネルギーを機械的なエネルギーに変換するアクチュエータを電動アクチュエータといい、ステッピングモーターは電動アクチュエータの一つである。

ステッピングモータとはフィードバックを必要としないで正確な位置決め制御を実現できるモータである。パルス信号により一定の角度だけ回転するため、回転角度を計測することなく正確に制御できる。これを利用して機械の精密な制御に用いられている。周波数に比例した回転速度が得られ、低速になるほど高いトルクを発生する。しかし、急激な周波数変化をもたらすとモータが外部のコイルの誘導についていくことができず、脱調を引き起こし、復帰が困難となる。

一般に PC/AT 互換機からステッピングモータを制御する際は、互換機本体についたパラレル・インターフェースのコネクタからパルス信号を送信し制御する。パラレルインターフェースとは、複数の信号線を用いて一度に複数ビット送受信するパラレル転送方式の接続インターフェースである。コンピュータ内部の伝送路やコンピュータと周辺機器の接続などの用途で使用されていた。近年では高速処理のため、シリアル式への移行が進んでいる。なぜなら、その処理の工程にパラレルは無駄が生じることがあるからである。シリアルは 1 本の信号線や回線を使って 1 ビットずつ順番にデータを送受信する。対してパラレルは複数の信号線や回線を用いてデータを複数ビットに分けてデータを送受信する。一見してパラレルの方が並行処理により早いと思われる。しかしパラレルでは次のデータを送るタイミングまでに、必要な読み込みを終えておくことやそれを一時的に蓄えるバッファが必要であり、工程自体が複雑になる。一方シリアルでは読み込んだ直後にそのままデータを送信することができるのでバッファが必要にならず、結果的にはシリアルのほうが高速にデータの送受信を行える。

本レポートでは、ステッピングモータの駆動回路に対して AT/PC 互換機から制御を行い回転動作を変更しながら回転を実行し、動作の原理及び回転制御とその状況を確認する (実験 1)。そしてステッピングモータの特徴についてまとめをしたのち、実験に対するまとめる。なお、実験内でステッピングモータの回転速度の制御及び停止制御を行ったが、調査課題で台形速度制御の調査を行い、先に回転速度制御するプログラムを台形制御で作成し、そのプログラムを実験内で使用した。そのため実験方法にプログラムを掲載することを控え調査課題内にプログラムを掲載し、結果及び考察を調査課題に記すことを先に断らせていただく。

## 2 実験 1：ステッピングモーターの運転

### 2.1 目的・概要

本実験では産業用ロボットやハードディスク、時計などの回転制御部分などでよく利用されるステッピングモータと、その駆動回路に対して AT/PC 互換機から制御を行い、回転動作を変更しながら回転を実行し、動作の原理及び回転制御とその状況を確認する。

### 2.2 実験方法

初めに、以下の手順で今回の実験を行うための実験道具の構成は以下のとおりである。

- 多出力直流安定化電源 (KENWOOD PW18-1. 8AQ)
- ステッピングモータ (Kiatronics 5V 28BYJ-48 5 ワイヤー)  
PM 型バイファイラ巻きの 4 相ステッピングモータ。クローボール数は 1 相につき 8 個。1 相励磁では 32 ステップで一回転でありその際のステップ角は  $11.25^{\circ}$ 。ギア比は 64
- PC/AT 互換機
- パラレルポート

今回は、モータに定格電圧を印加してユニポーラ 1 相、2 相、1-2 相励磁でモータをプログラムで運転 (正転・逆転・停止・速度制御) するために PC/AT 互換機のパラレルポートにステッピングモータ駆動回路を図 1 のように接続した。PC のパラレルポートの接続にはパラレルポート接続ケーブルを用いた。パラレルポート接続ケーブルの 1 はデータ取り込み同期信号のための STRBE である 1 を飛ばした、2, 3, 4, 5(それぞれビット 0, ビット 1, ビット 2, ビット 3) からブレッドボードに配線し、さらにブレッドボードの同出力部からステッピングモータドライバーボードの IN 部に接続した。多出力直流安定化電源は 12V に設定しドライバーボードの電源接続部に +12V, GND をそれぞれ電源の +12V, GND に接続した。

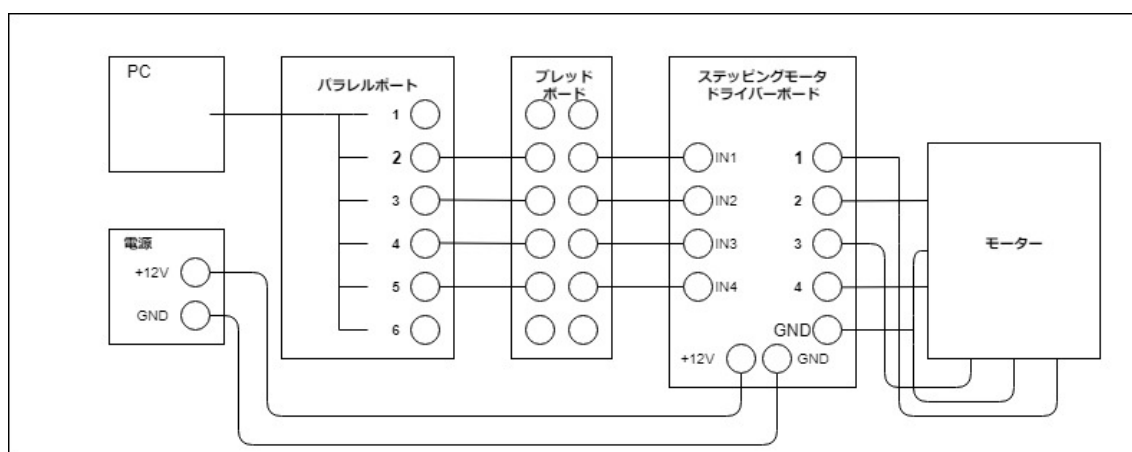


図 1: ステッピングモータ駆動回路に接続した回路図

3 つのプログラムを実行することでステッピングモータの 1 相、2 相、1-2 相励磁別々に正転及び逆転の運転

を行った。プログラムの実行コマンドは 1 相, 2 相, 1-2 相励磁, 順に `sudo ./step1 [f] [Rev]`, `sudo ./step2 [f] [Rev]`, `sudo ./step1-2 [f] [Rev]`, とし `f` はパルス信号の周波数 (Hz) とし, `Rev` には 0, 1 のいずれかを入力すると順に正転, 逆転が行われるようにした。使用したプログラムの詳細は下に示す。

すべてのプログラムにおいて最大自動起動波数の計測をした。1 相励磁においては前回のプログラム終了時の最終入力をしたピンの別による最大自動起動波数も測定した。モータの回転を計測するために図 2 のようにシャープペンシルの芯とモータの回転部をセロテープでつけた。振動によるモータ位置変動をなくすためにモータの裏と机をセロテープでつけた。携帯をモーターの上空に固定し, 回転を撮影, その動画で 1 相 2 相 1-2 相励磁においてプログラムを実行させモータを 5 回転させて半周にかかる時間を 10 分割して計測した。

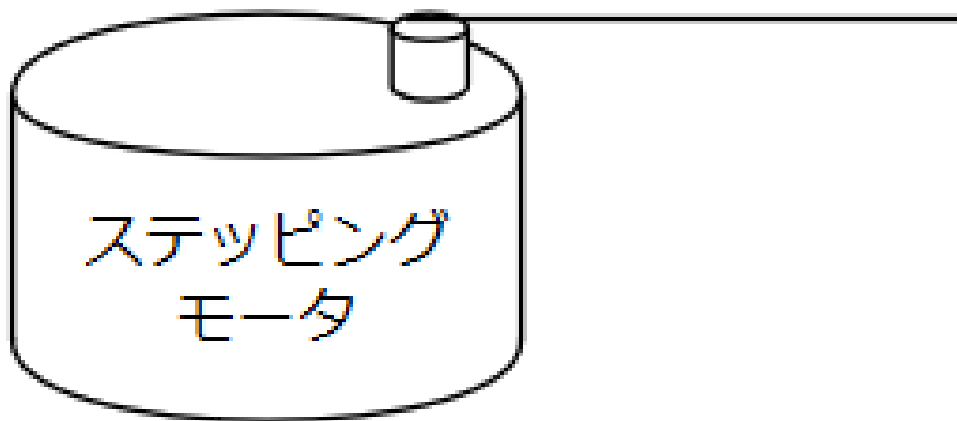


図 2: ステッピングモータの回転角計測方法

ソースコード 1: 1step.c

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/io.h>
4  #include <stdlib.h>
5
6  #define PPA_0 ((unsigned long)(0x378)) // パラレルポート出力アドレス
7  #define PBS 2 // パラレルポート byte size
8  #define TRUE 1
9
10 int main(int argc, char* argv[]) {
11     int f = atoi(argv[1]); //パルス信号の周波数 (Hz)
12     int reverse = atoi(argv[2]); //正転逆転の切り替え
```

```

13  int i;
14  double T = 1.0e6/f; //usleep の引数, 単位はマイクロ秒
15  printf("%lf\n", T);
16  int bi[8]; //パラレルポートのピンに対応するビットの配列
17  for (int j = 0; j < 8; j++){
18      bi[j] = 1<<j;
19  }
20
21  if ((i = ioperm(PPA_0, PBS, TRUE)) != 0) { //動作チェック
22      printf("1\n");
23      exit(-1);
24  }
25  if (reverse == 0){ //正転
26      while (1) { // 1 相励磁
27          outb(bi[0], PPA_0); //1 番ピン
28          usleep(T);
29          outb(bi[1], PPA_0); //2 番ピン
30          usleep(T);
31          outb(bi[2], PPA_0); //3 番ピン
32          usleep(T);
33          outb(bi[3], PPA_0); //4 番ピン
34          usleep(T);
35      }
36  }
37  else { //逆転
38      while (1) { //1 相励磁
39          outb(bi[3], PPA_0); //4 番ピン
40          usleep(T);
41          outb(bi[2], PPA_0); //3 番ピン
42          usleep(T);
43          outb(bi[1], PPA_0); //2 番ピン
44          usleep(T);
45          outb(bi[0], PPA_0); //1 番ピン
46          usleep(T);
47      }
48  }
49  }

```

ソースコード 2: 2step.c

```

1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/io.h>
4  #include <stdlib.h>
5
6  #define PPA_0 ((unsigned long)(0x378)) // パラレルポート出力アドレス
7  #define PBS 2 // パラレルポート byte size
8  #define TRUE 1
9
10 int main(int argc, char* argv[]) {
11     int f = atoi(argv[1]); //パルス信号の周波数 (Hz)
12     int reverse = atoi(argv[2]); //正転逆転の切り替え
13     int i;
14     double T = 1.0e6/f; //usleepの引数, 単位はマイクロ秒
15     printf("%lf\n", T);
16     int bi[8]; //パラレルポートのピンに対応するビットの配列
17     for (int j = 0; j < 8; j++){
18         bi[j] = 1<<j;
19     }
20
21     if ((i = ioperm(PPA_0, PBS, TRUE)) != 0) { //動作チェック
22         printf("1\n");
23         exit(-1);
24     }
25     if (reverse == 0){ //正転
26         while (1) { //2相励磁
27             outb(bi[0]+bi[1], PPA_0); //1, 2 番ピン
28             usleep(T);
29             outb(bi[1]+bi[2], PPA_0); //2, 3 番ピン
30             usleep(T);
31             outb(bi[2]+bi[3], PPA_0); //3, 4 番ピン
32             usleep(T);
33             outb(bi[3]+bi[0], PPA_0); //4, 1 番ピン
34             usleep(T);
35         }
36     }
37     else { //逆転
38         while (1) { //2相励磁
39             outb(bi[3]+bi[0], PPA_0); //4, 1 番ピン

```

```

40     usleep(T);
41     outb(bi[2]+bi[3], PPA_0); //3, 4 番ピン
42     usleep(T);
43     outb(bi[1]+bi[2], PPA_0); //2, 3 番ピン
44     usleep(T);
45     outb(bi[0]+bi[1], PPA_0); //1, 2 番ピン
46     usleep(T);
47 }
48 }
49 }

```

ソースコード 3: 1-2step.c

```

1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/io.h>
4  #include <stdlib.h>
5
6  #define PPA_0 ((unsigned long)(0x378)) // パラレルポート出力アドレス
7  #define PBS 2 // パラレルポート byte size
8  #define TRUE 1
9
10 int main(int argc, char* argv[]) {
11     int f = atoi(argv[1]); //パルス信号の周波数 (Hz)
12     int reverse = atoi(argv[2]); //正転逆転の切り替え
13     int i;
14     double T = 1.0e6/f; //usleep の引数, 単位はマイクロ秒
15     printf("%lf\n", T);
16     int bi[8]; //パラレルポートのピンに対応するビットの配列
17     for (int j = 0; j < 8; j++){
18         bi[j] = 1<<j;
19     }
20
21     if ((i = ioperm(PPA_0, PBS, TRUE)) != 0) { //動作チェック
22         printf("1\n");
23         exit(-1);
24     }
25     if (reverse == 0){ //正転
26         while (1) { //1-2 相励磁
27             outb(bi[0]+bi[1], PPA_0); //1, 2 番ピン

```



```

28     usleep(T);
29     outb(bi[1], PPA_0); //2 番ピン
30     usleep(T);
31     outb(bi[1]+bi[2], PPA_0); //2, 3 番ピン
32     usleep(T);
33     outb(bi[2], PPA_0); //3 番ピン
34     usleep(T);
35     outb(bi[2]+bi[3], PPA_0); //3, 4 番ピン
36     usleep(T);
37     outb(bi[3], PPA_0); //4 番ピン
38     usleep(T);
39     outb(bi[3]+bi[0], PPA_0); //4, 1 番ピン
40     usleep(T);
41     outb(bi[0], PPA_0); //1 番ピン
42     usleep(T);
43 }
44 }
45 else { //逆転
46     while (1) { //1-2 相励磁
47         outb(bi[3]+bi[0], PPA_0); //4, 1 番ピン
48         usleep(T);
49         outb(bi[3], PPA_0); //4 番ピン
50         usleep(T);
51         outb(bi[2]+bi[3], PPA_0); //3, 4 番ピン
52         usleep(T);
53         outb(bi[2], PPA_0); //3 番ピン
54         usleep(T);
55         outb(bi[1]+bi[2], PPA_0); //2, 3 番ピン
56         usleep(T);
57         outb(bi[1], PPA_0); //2 番ピン
58         usleep(T);
59         outb(bi[0]+bi[1], PPA_0); //1, 2 番ピン
60         usleep(T);
61         outb(bi[0], PPA_0); //1 番ピン
62         usleep(T);
63     }
64 }
65 }

```

## 2.3 実験結果

本実験に対してまず、表 1 の最大自動起動波数を得た。1 相 2 相励磁に対し 1-2 相励磁はその 2 倍程度の最大自動起動波数であった。しかしこれらの周波数の値はおおよそその値であり、ムラが発生していた。具体的には 1 相では 955Hz で動くときもあれば 800Hz で動かない場合もあった。1 相励磁の前のプログラムの終了時の最終入力をしたピンの値による最大自動起動波数は各ピンに対して表 2 を得た。1 ピン 4 ピンは 900Hz 以上の最大自動起動波数を有していたのに対し、2 ピン 3 ピンでは 760Hz 付近の最大自動起動波数であった。また、周波数を 200Hz として 1 相 2 相 1-2 相励磁の半回転にかかる時間の計測により表 3～5 を得た。動画での撮影の計測であるとはいえ目視であるため正確な計測であるとは言えないが、1-2 相励磁ではおおよそ 1 相 2 相励磁の 2 倍程度の時間を要した。

表 1: 最大自動起動波数

相	周波数 [Hz]
1 相	955
2 相	1100
1-2 相	2000

表 2: 前回終了時ピン別最大自動起動波数

ピン	周波数 [Hz]
1	955
2	754
3	766
4	932

表 3: 1 相の半周に要した時間

項目	値 [s]
測定標本	10.219 10.258 10.358 10.531 10.252 10.216 10.162 10.287 10.013 10.112
平均	10.24080
不変分散	0.01960
標準偏差	0.13400
誤差	0.04426
測定値	$10.24080 \pm 0.04426$

表 4: 2 相の半周に要した時間

項目	値 [s]
測定標本	10.326 10.449 10.151 10.040 10.037 10.591 10.230 10.132 10.042 10.198
平均	10.21980
不変分散	0.03463
標準偏差	0.18708
誤差	0.05884
測定値	10.21980 ± 0.05884

表 5: 1-2 相の半周に要した時間

項目	値 [s]
測定標本	20.523 20.432 20.738 20.657 20.415 20.528 20.626 20.653 20.426 20.212
平均	20.52100
不変分散	0.02419
標準偏差	0.15555
誤差	0.04919
測定値	20.52100 ± 0.04919

## 2.4 考察

ステッピングモータの駆動において周波数によってモータが駆動しないことがある。これはステッピングモータの駆動方法が関係している。ステッピングモータは周りに配置したコイルの磁力で内部のモーターを回転させる。ある瞬間にモータの内部の軸を回転させるに際に速く回転をしていると、引き寄せているはずの軸を回転方向とは逆に引き寄せてしまう場合がある。また、ステッピングモータはコイルに電流が流れているときと流れていないときがあるため、交流電流が流れていると考えられる。コイルは交流を流すと周波数が高くなるにつれ電流が流れにくくなる。すると、トルクが減少するため負荷を回すことができなくなる。このためモータがある一定の周波数を超えると乱調、脱調と呼ばれる現象を起こす。乱調にはほかに振動が原因になる場合がある。表 1 に示すように 1-2 相励磁は 1 相 2 相励磁に対して 2 倍程度の最大自動起動波数を有している。これは先の理由により 1 相 2 相励磁に比べ 1-2 相励磁がモータの駆動を行うためのコイルの磁力を与える範囲が 1 相 2 相励磁より広いこと乱調が起きにくく、コイル同時に 2 つ使うことで 1 回転の速度は半分になり速度の上昇がほかの方式と比較して半分になったため 2 倍の周波数になったと考えられる。1 相励磁と 2 相励磁では引き付ける範囲が 2 相励磁のほうが大きいこと 2 相励磁のほうが 1 相励磁に比べ最大自動起動波数が高いと考えられる。モータの回転が 1 相よりも本実験のプログラムでは実行の際 1 相では常に 1 ピンからのコイルの引付から行い、正転では 2, 3, 4 という順番でコイルの引付が行われるがモータが 4 ピン 1 ピンに向いていてプログラムが実行させる際は 4 ピンであればそのまま 1 ピンに引き付けられ慣性が加わりながら回るこ

とができ, 1 ピンではそのままコイルの引付けに合わせて回転を行える. しかし, 2, 3 ピンでは初めの 1 ピンの引付の際今後まわる回転と逆回転に力が加わり引付の回転速度についていけず 1, 4 ピンよりも低い最大自動起動波数になるために表 2 のようになったと考えられる.

本実験のステッピングモータのステップ角は 1 相励磁 2 相励磁では  $11.25^{\circ}$ , 1-2 相励磁では半分の  $5.625^{\circ}$  である. このため 1 相 2 相励磁ではパルス信号 32 回ごとに内部のモータは 1 回転をする. ギア比 64 のため外部のモータは  $32 \times 64 = 2048\text{step}$  ごとに 1 回転することになると考えられる. 表 3 と表 4 のように 1 相 2 相励磁でおよそ 10.24 秒で半周したのはこのためであると考えられる. また, 表 5 で 1-2 相励磁が 1 相 2 相励磁より 2 倍の時間がかかったのはステップ角が 1-2 相励磁で 1 相 2 相励磁の半分であるために外部のモータが 1 回転するために  $64 \times 64 = 4096\text{step}$  必要であるためだと考えられる.

### 3 調査課題 1：ステッピングモータの電流の立ち上がりの改善

駆動電圧が 3.5V, 相電流が一相当たり 1.75A, 巻線抵抗  $2\Omega$  のユニポーラタイプのステッピングモータの立ち上がりをよくするために外付け抵抗を付けたときのモータに供給する電圧, 外部抵抗で消費される電力はどの程度かを考える. 立ち上がりが良くなるというのは時定数を小さくすることを指す. 時定数とは, 回路に流れる電流が定常電流になるまでの変化速度を表す定数のことである. 一般には最大電流に対し 63.2% になるまでの時間を指し, この五倍でほぼ定常電流になるといわれている. ステッピングモータのコイル抵抗を  $R_w$ , 外部抵抗を  $R$ , ステッピングモータのコイルインダクタンスを  $L$  とするコイルと抵抗を直列につなぐ場合を考える. ここで時定数  $\tau$  は

$$\tau = \frac{L}{R_w + R}$$

と表せる [2]. すなわち, 時定数  $\tau$  は外部抵抗  $R$  を入れることで小さくなり, 電流の立ち上がりが改善されると考えられる. ここではインダクタンスを考慮せず考える. 電圧を  $E=3.5V$ , 全体の抵抗は  $R_w + R$ , モーターに供給する電圧を  $V_m$  とするなら  $V_m$  は複素表示で

$$V_m = \frac{R_w E}{R + iR_w}$$

となるので

$$|V_m| = \frac{R_w E}{\sqrt{R^2 + R_w^2}}$$

である. なので外部抵抗により消費される電力  $|P_R|$  は

$$|P_R| = \left| \frac{RE^2}{(R + iR_w)^2} \right| = \frac{RE^2}{\sqrt{(R^2 - R_w^2)^2 + 4R_w^2 R^2}} = \frac{RE^2}{R^2 + R_w^2}$$

である. さて  $R = R_w$  とすると

$$|V_m| = \frac{7}{\sqrt{8}}$$

$$|P_R| = \frac{24.5}{8}$$

となる. この場合消費電力は外付け抵抗を付けない場合に比べて多くの電力を消費することになる. 大きな抵抗を使うことによるモータへの供給電圧の低下も問題である. そのため外付け抵抗をつけることは望ましいとはいえない. そこで電圧の値を一定の値に保つのではなく電流の値を一定の値に保つように設定する方法を考える. 駆動初期の立ち上がりの際に流したい電流  $I$  を流すことのできる電圧よりも高い電圧を印加させ  $I$  よりも高い電流になると電圧をオフにして電流を減らす, また  $I$  よりも低い値になると高い電圧をかける. これを繰り返し, 設定した  $I$  に近い電流を流し続けることができ, 消費電力も抑えられると考えられる.

## 4 調査課題 2：ステッピングモーターの特徴

ステッピングモータにはさまざまな特徴がある。まず第一に、起動と停止、正逆転駆動、可変速 駆動、間欠駆動、マイクロステップ駆動といった動作的特徴を有してる。ステッピングモータの回転角度は入力パルス信号数に比例して決まり、回転速度は入力するパルス周波数に比例する。したがって、これらの性質により回転子の位置や速度の情報をフィードバックすることなく電力パルスの入力によってのみで位置決め制御、速度制御ができる。これによりオープンループ駆動が可能となる。ステッピングモータ以外では回転子の位置や速度はセンサを付けてフィードバックをすることでしか得られなく、これをクローズド制御という。高価な変位センサー、エンコーダなどの部品により出来上がった位置検出機構、フィードバック処理制御回路は不要であり、駆動回路を含めたコストは一般にステッピングモータを使用したほうが安く簡易的である。また、直流モータのブラシのように機械的摩耗の心配はなく、保守を要さない。したがって、ステッピングモータは位置決め制御系の駆動源として幅広く使われている。

ステッピングモータは固定子の相巻線の励磁電流を切り替えることで、あるモータに固有の回転角度、すなわちステップ角で歩進する。そのため、相の切り替えを必要とするため複数相が存在する。ステッピングモータの相数は様々な特性と関係を持つ。

### 1. 高分解能

PM 型についてまた後述するが、永久磁石式 PM 型回転子は N 極と S 極が交互に回転子の回転方向周上に等ピッチで配置された構造で、回転し局数は N 極数と S 極数の和であり、極対数は N 極と S 極を 1 対と考えると、その対数をいう。ここで極対数を  $N_p$ 、固定子相数を P とするとステッピングモータの分解能を決めるステップ角  $\Theta_s$  は、回転子が永久磁石式の場合は

$$\Theta_s = \frac{180^\circ}{PN_p}$$

で表される。つまり、回転子 1 回転の機械角度  $360^\circ$  を極数  $2N_p$  で割ることで 1 極あたりの機械角度を得ることができ、この 1 極あたりの機械角度は固定子の相数で分割されステップ角度になる。ステップ角を小さくして分解能を向上させるには、 $N_p$  を大きくするか P なる固定子相数を増やすかどちらかの方法があり、どちらも併用する。[2]

### 2. 振動性

多相になるほど隣相同士の静トルクカーブの交点のトルク値  $T_g$  と最大静止トルク  $T_h$  との比、あるいは差を小さくすることができる。 $T_g$  はモータが負える負荷トルクの限界を意味し、 $\frac{T_h - T_g}{T_h}$  がトルク変動であるトルクリップルであり、多相ほどこの値が小さくなり低振動化に寄与する。[2]

### 3. 高速応答

ステッピングモータがその性質上、高速になると電流周波数大による書く速度  $\omega$  (電気角) が増大する。すると、コイルのリアクタンス  $\omega L$  が増大し、電流が流れにくくなる。これにより、コイルインピーダンス  $Z$  は直流抵抗  $R$  に比べ、リアクタンス  $\omega L$  のほうが大幅に大きくなる、したがって、定電圧駆動時の高速時の一層分電流は印加電圧  $V$  より概略  $\frac{V}{\omega L}$  となる、機械角速度  $\omega$  は極対数に対して比例する関係にあるから、多相になるほど極対数が減るので流れる電流が増加して高速化に対応することができる。[2]

ステッピングモータのモータの種類は先述した PM 型のほかに VR 型、HB 型があり、それぞれ用途により型を使い分ける。

## 1. PM 型

PM(Permanent Magnet) 型回転子とは、インナーロータ型で言えば、図 3 のようにロータが永久磁石で構成され、これの周りに複数のコイルを配設されている。PM とは永久磁石のことであり、永久磁石を用いていることに由来している。このロータ極が磁化されていることにより磁束密度が増大されているため、強いトルクが発生する。そのため、微細なステップ角の調整にはロータの極数と駆動力を発生するコイルの数を増やす必要がある。例えば、固定子の歯数を 8 個にして回転子の磁極を倍の 4 極にすることで図 3 の場合 1 ステップあたりのステップは  $45^\circ$  になる。どの電流コイルにも電流が流れていない無励磁の場合でも永久磁石の働きによりトルクが発生する。この性質を効率的に利用することで、省エネ型の駆動システムを構成することが可能である。[4]

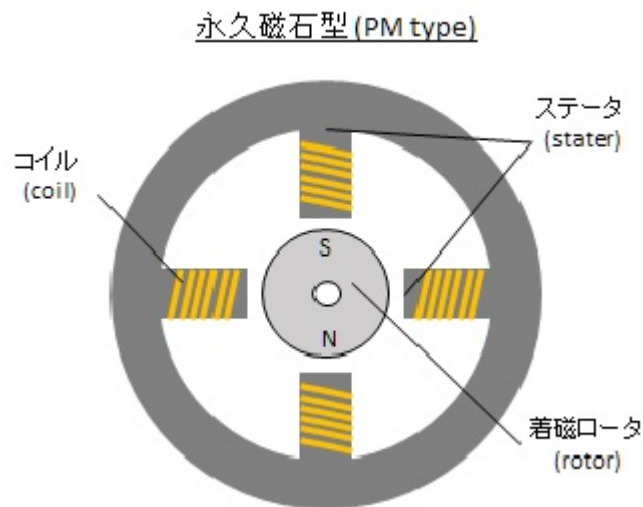


図 3: PM 型の構造 [3]

## 2. VR 型

VR(Variable Reluctance) 型回転子は、図 4 のような回転子が珪素銅板やソリッド状電磁軟鉄などの磁性体によりできており、その周りのステータにまかれたコイルから構成されている。電流を流すことによりコイルが固定子磁極から電磁石になることで、回転子歯を吸引することでトルクを発生させて歩進する。磁気抵抗が回転子の歯と溝で変化する可変レクタンス型であることから VR 型と呼ばれる。PM 型と異なり、永久磁石を用いずロータとステータのコイルに電流を流し磁束を発生させるため、比較的大きな電流を流す必要がある。そのため温度上昇が高くなる。このモータはステータ部のスロットにまかれたコイルに次々と電流を流し、ステータ磁極が回転磁界を構成し、ロータ歯を次々と吸引することで回転力を発生する。しかし、ステップ角を小さくして分解能を上げるには数多く細かな歯形の精密加工とステータとロータ間のギャップ長を短くすることが必要で低コスト化、小型化が難しい。[4]

### 可変リラクタンس型 (VR type)

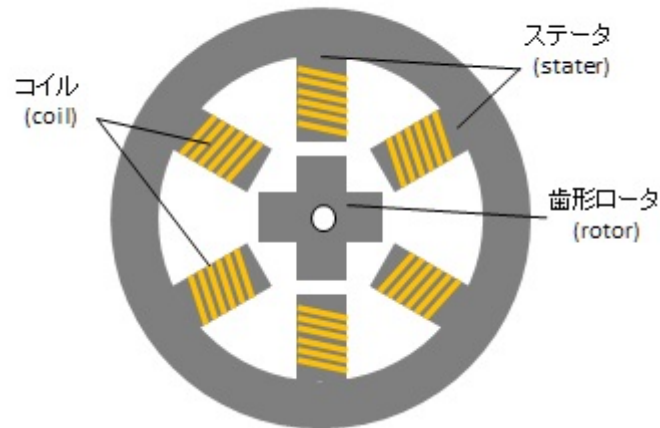


図 4: VR 型の構造 [3]

### 3. HB 型

HB(Hybrid) 型は回転子の構造が PM 型と VR 型の複合型であり、ハイブリッドを意味している。回転子は磁性体からなり、その外周に多数の歯を等ピッチで有した歯車に似た形状である (図 5)。図 6 VR 型の回転子と同一構造の回転子を 2 個もち、1 つは外周の歯ピッチの  $1/2$  ずらして回転軸方向に 2 極に磁化した円盤状または円筒状永久磁石を挟んで餅回転軸とともに固定された構造をしている。ステータ歯極及びこれと対向したロータ磁極で構成され、さらにそのロータは多数の誘導子である歯車上突極を有し、これに回転軸と同方向に磁化された永久磁石が組み込まれている。したがって永久磁石を挟んで片方が N 極他方が S 極二次化される。構造的には先の二つよりも複雑になるが基本的には VR 型と PM 型を一体化構造としたものと考えることができる。一般にハイブリッド型は回転子極数が PM 型よりも増やすことができ、漏洩磁束も小さくなるため分解能が高い。そのため高精度、高トルク、微小ステップ角と様々な優れた特徴をもつ。



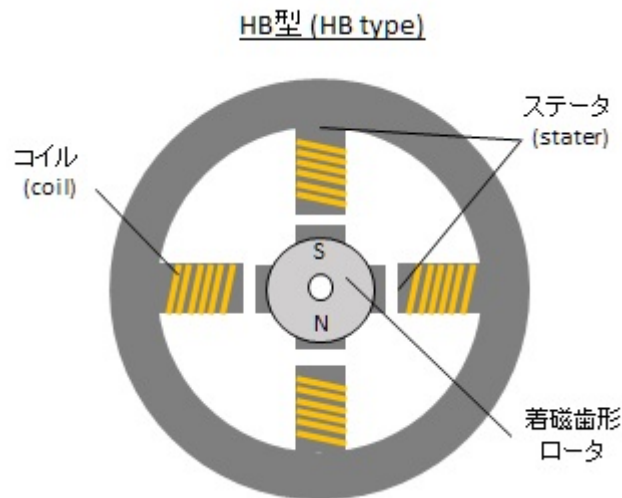


図 5: HB 型の構造 (断面)[3]

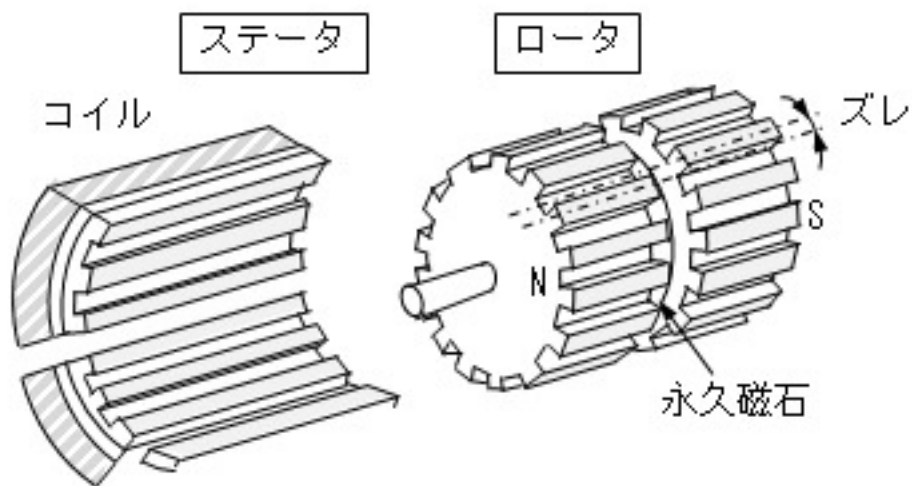


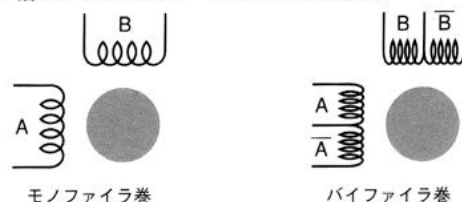
図 6: HB 型の構造 [5]

VR 型ステッピングモータの場合は固定子磁極が回転子を吸収する際、固定子極性は N, S どちら問うことはないが、他の型の際回転子がすでに N 極 S 極に磁化されているために固定子磁極は極性の反転が必要となる。固定子磁極を励磁する際にはバイポーラ方式とユニポーラ方式が存在する、バイポーラ方式は励磁するコイルをユニファイラ巻き、つまり線を同一方向にのみ巻き、交流を流すことで時局性を反転させる。一方、コイルに 2 本の線を巻き倍ファイラ巻きとし、すなわちコイルを二つ用意して、別々に直流通電をした場合、互いに極性が逆になるようにしたもののがユニポーラ方式である。図 7 はユニポーラ方式とバイポーラ方式の 2 相式の略図である。ユニポーラ方式の場合、図 8 の C 端子を正、A,  $\bar{A}$  端子を負に接続した場合、別々に接続することで磁化極性が反転するが、同時に接続した場合主極の磁束は打ち消されるため注意が必要である。バイポーラ方式の場合 8 の A と  $\bar{A}$  を同時にわずかな時間でもオンにすると電源が短絡化され大電流が流れる。そのた

め短絡防止回路の付加も必要となりバイポーラ方式の場合ユニポーラ方式と比べ駆動回路が複雑になる。

しかし、バイポーラ方式とユニポーラ方式では同じ線径である場合巻き数がバイポーラ方式のほうが2倍あり、コイル抵抗が2倍になる。ユニポーラ方式の巻き数を  $N$  抵抗を  $R$  とすると低速時のリアクタンスが小さいためにこれを無視して  $\frac{V}{R}$  が電流となる。この時バイポーラ方式では電流が  $\frac{V}{2R}$ 、巻き数が  $2N$  となる。電流×コイルの巻き数がトルクに比例する。そのため、電流×コイルの巻き数を考えるとユニポーラ方式とバイポーラ方式では同じ値になる。しかし消費電力を考えると消費電力 = 電圧<sup>2</sup>/抵抗 なのでバイポーラ方式のほうが消費電力が  $\frac{1}{2}$  小さい。よって低速時の効率はバイポーラ方式のほうが良いことがわかる。そのため小型化する場合や低速時、大きなトルクが必要な場合は駆動回路が複雑になるが、バイポーラ方式のほうが良い。たいして、高速で駆動する場合バイポーラは巻数が多いためインダクタンスが増え電流の減少が大きくなり、高速時のトルクの減少はユニポーラより大きくなるため向かない。[2]

●図3.53 2相ステッピングモータのコイル表示例



注) 英文字の頭にある一印は、逆極性の磁界を発生するように接続されることを意味します。

図 7: ユニポーラ方式 (左) とバイポーラ方式 (右)[6]

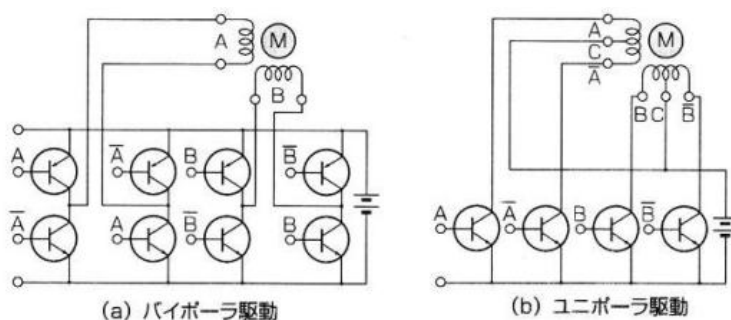


図 8: ステッピング・モータの駆動方式 [7]

## 5 調査課題 3：ステッピングモーターの台形的速度制御

初めに記載した通り、ここでは実験で行った台形速度制御についてまとめる。

### 5.1 実験方法

加速と減速を直線的に行う速度制御を台形速度制御という。そのため今回は、ある時間までにある速度までモータを加速させ、その後一定時間定速で運転し、その後モータを減速させていくことを行った。そこで以下のプログラムを作成し、`sudo ./trapezoid [f max] [t]`(`f max`:最大周波数 [Hz], `t`:加減速にかかる時間を示す [20ms]) を実行コマンドとし、実行を行い、1 相励磁における最大応答周波数の計測を行った。

ソースコード 4: trapezoid.c

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/io.h>
4  #include <stdlib.h>
5  #include <time.h>
6
7  #define PPA_0 ((unsigned long)(0x378)) // パラレルポート出力アドレス
8  #define PBS 2 // パラレルポート byte size
9  #define TRUE 1
10
11 int main(int argc, char* argv[]) {
12     int f_max = atoi(argv[1]); //パルス信号の最大周波数 (Hz)
13     int t = atoi(argv[2]); //加減速にかかる時間 (/20ms)
14     int i;
15     double f; //パルス信号の周波数 (Hz)
16     double T = 1; //usleep の引数, 単位はマイクロ秒
17     clock_t x;
18     int bi[8]; //パラレルポートのピンに対応するビットの配列
19     for (int j = 0; j < 8; j++){
20         bi[j] = 1<<j;
21     }
22
23     if ((i = ioperm(PPA_0, PBS, TRUE)) != 0) { //動作チェック
24         printf("1\n");
25         exit(-1);
26     }
```

```

27 while (1) { //1 相励磁, 台形速度制御, 約  $t*5*10^{-5}$  秒かけて  $f_{max}$  まで加速, その後約 5
    ↳ 秒間最大周波数で駆動したのち, 約  $t*5*10^{-5}$  秒かけて減速, 停止
28     if (T >= 0){
29         x = clock();
30         if (x <= t){
31             f = (double)f_max/(double)t*x;
32             T = 1.0e6/f;
33         }
34         else if (x < 100000+t){
35             T = T;
36         }
37         else if (x < 100000+2*t){
38             f = -(double)f_max/(double)t*(x-2*t-100000);
39             T = 1.0e6/f;
40         }
41         else{
42             T = -1;
43         }
44         outb(bi[0], PPA_0); //1 番ピン
45         usleep(T);
46         x = clock();
47         if (x <= t){
48             f = (double)f_max/(double)t*x;
49             T = 1.0e6/f;
50         }
51         else if (x < 100000+t){
52             T = T;
53         }
54         else if (x < 100000+2*t){
55             f = -(double)f_max/(double)t*(x-2*t-100000);
56             T = 1.0e6/f;
57         }
58         else{
59             T = -1;
60         }
61         outb(bi[1], PPA_0); //2 番ピン
62         usleep(T);
63         x = clock();
64         if (x <= t){

```

```

65     f = (double)f_max/(double)t*x;
66     T = 1.0e6/f;
67 }
68 else if (x < 100000+t){
69     T = T;
70 }
71 else if (x < 100000+2*t){
72     f = -(double)f_max/(double)t*(x-2*t-100000);
73     T = 1.0e6/f;
74 }
75 else{
76     T = -1;
77 }
78 outb(bi[2], PPA_0); //3 番ピン
79 usleep(T);
80 x = clock();
81 if (x <= t){
82     f = (double)f_max/(double)t*x;
83     T = 1.0e6/f;
84 }
85 else if (x < 100000+t){
86     T = T;
87 }
88 else if (x < 100000+2*t){
89     f = -(double)f_max/(double)t*(x-2*t-100000);
90     T = 1.0e6/f;
91 }
92 else{
93     T = -1;
94 }
95 printf("%lf\n", f);
96 outb(bi[3], PPA_0); //4 番ピン
97 usleep(T);
98 }
99 }

```

このプログラムでは、加速にかかる時間と最大周波数を受け付けることでどの時間ではどのパルス周期にする必要があるかをパルス信号を送るごとに計測し、usleep 関数でパルス信号を送るタイミングをずらすことで台形速度制御を可能にしている。実行時の  $t$  が  $1/20\text{ms}$  という単位になっているのは C 言語の Clock 関数を使用

するためであり、clock の単位で  $t$  を受け付けているためである。

## 5.2 実験結果

加速時間を 5s で行うとき 1 相の最大応答周波数は 2400Hz であった。それ以上の周波数にすると乱調を起こした。

## 5.3 考察

最大応答周波数と最大自動起動周波数が異なるのは、自動起動周波数は止まっている状態から回転をさせた際の周波数を示し、応答周波数は回転している状態からの周波数を示しているからである。モータが回転している際慣性力により回転子はコイル引付にある程度ついていくことができるため、応答周波数のほうが自動起動周波数よりも最大値が大きいと考えられる。しかし、一定の周波数以上になると乱調を起こしモータが逆回転や停止を起こすのは回転していても回転子がコイルの引付に反応できなくなるためと考えられる。

この台形制御に関していえば、一定速度にする際に加速度が急激に変化する、そのため一定速度に変わる際にトルクの値が急激に変化するため、回転がコイルの引付の周波数に同期できず乱調を起こすことがあると考えられる。そのため加速度の減少を緩やかにおこなうことが一定速度にする際に乱調を発生させないために必要になることがあると考えられる。

## 6 まとめ

本実験では信号処理を用いてステッピングモータの制御を行った、実験の調査の結果ステッピングモータは正確な位置決め運転を簡単に実現でき、パルス信号によって回転角度・回転速度を正確に制御できるために機械の精密な制御に用いられていることが分かった。また、速度制御には注意が必要であり急な加減速を行う際は注意をしなければならないという欠点も確認した。更に、ステッピングモータは使用用途に応じて駆動方式や回転方法を選ぶ必要があることもわかった。

## 参考文献

- [1] アクチュエータとは（基礎知識） | 松定プレゼンション (2019/4/22)  
<https://www.matsusuda.co.jp/news/2018/05/words/actuator.html>
- [2] ステッピングモータの使い方 — 坂本正文
- [3] 技術解説ステッピングモータの基本—インデックスプロ (2019/4/22)  
<https://www.indexpro.co.jp/article/detail/2/6>
- [4] 図解メカトロニクス入門シリーズ — アクチュエータ入門改訂 2 版 雨宮好文
- [5] HB 型ステッピングモータ—日本パルスモータ-株式会社 (2019/4/24)  
[http://www.pulsemotor.com/technology/terms/eng\\_hbtypesutepinnngumota.html](http://www.pulsemotor.com/technology/terms/eng_hbtypesutepinnngumota.html)
- [6] 小型モータのすべて—見城—技術評論社
- [7] モータの動作原理と特性—荻野—SOLITON 34 SPECIAL REPORT