

【ドローンの製作について】

Drone Production

電子系コース 丸茂 賢斗 (MARUMO Kento)

1 はじめに

「いつかはこの手でドローンを作りたい！」今まで電子系の学生として学んできた私はそう思っていた。そういった経緯から電子回路やプログラム学習の集大成として念願であるドローンの自作に取り組むことにした。

同じくドローンを作りたいという仲間がいたため、共同製作をすることになった。目標として、ドローンが空中で静止する飛行状態である「ホバリング」を目指し、研究開発を行った。

2 自作した部分

- ・フレーム
- ・制御回路
- ・プログラム

上記の3つを自作し、市販モジュール等と組み合わせたりをしながら機体を製作した。右のように完成した。



3 部品の選定

制御用マイコンとして、Arduino nano 33 BLE を使用した。理由として Arduino nano にジャイロ、加速度、磁気センサそして BLE 機能を有するもので、かつ小型軽量であることからドローン製作に適していると考えたことが挙げられる。モータは安定して高速回転が可能なブラシレスモータ (ARRIS S2205) を使用した。

4 製作の過程

3D プリンタで出力したフレームにモータ、プロペラの取付け



4つのモータとプロペラは回転の向きに応じてつけなければならない。確認しても間違った向きで取り付けてしまったことがあった。チェックリスト等で手順の間違いをなくしていく方法が考えられる。

Arduino やセンサ、給電回路等の配線

配線の量が多すぎて、スペースがなかった。しかしフレームを大きくするとモータの出力を大きくしないといけないため、重量と配線のバランス調整が難しい。結束バンドを利用し線をまとめることで対策した。



コーディング (Arduino IDE)

```
last_pitch = pitch;
last_roll = roll;
last_yaw = yaw;

pwm_pitch = pitch_pid(target_pitch, pitch);
pwm_roll = roll_pid(target_roll, roll);
pwm_yaw = yaw_pid(target_yaw, yaw);

dis = US_pid(target_d, distance);

//Serial.println(ax);

//Serial.println(pitch);
//Serial.println(roll);
//Serial.println(yaw);

//PWM信号の計算
int16_t fr_pwm = (throttle + dis * Kr) - pwm_roll + pwm_pitch + pwm_yaw;
int16_t fl_pwm = (throttle + dis * Kr) + pwm_roll + pwm_pitch - pwm_yaw;
int16_t br_pwm = (throttle + dis * Kr) - pwm_roll - pwm_pitch - pwm_yaw;
int16_t bl_pwm = (throttle + dis * Kr) + pwm_roll - pwm_pitch + pwm_yaw;

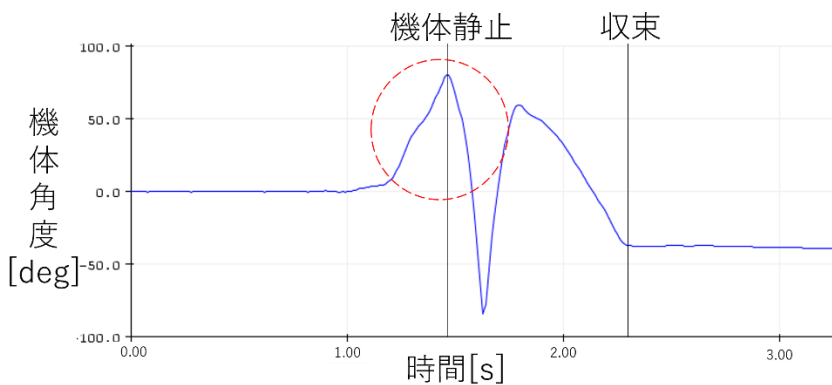
set_pwm(fr_pwm, fl_pwm, br_pwm, bl_pwm);

Current_time = micros() - last_time;
if(Current_time > 0 && Current_time < 9000){
```

機体の改良をしながら制御プログラムを作るというのは、確定要素が少なく困難であった。特に制御構造が混乱しないよう見やすいコーディングを心掛けた。プログラミングスキルを磨く、絶好の機会となった。

5 姿勢制御

ドローンをホバリングさせるために、各センサから得られた加速度、角速度、地磁気の値を Madgwick フィルターと呼ばれる計算ライブラリに入力し、機体の角度、すなわち姿勢角を取得した。Madgwick フィルターを使用することで姿勢角は算出できたが、初動においては、傾きとは逆の方向で大幅にずれた角度が算出される現象が発生してしまった。(下画像点線内)



問題解決のため、Madgwick フィルターのゲインや、サンプリング周波数の調整を行ったが、解決できなかった。

姿勢角から機体がどのように傾いているかを算出し、その傾きに対して PID 制御でモータの出力を変化させる。そうすることで目標の角度を保つことができるはずである。しかし上のような問題があるため、初動では角度を保つことができなかった。

6 結果

結果としては、安定したホバリングを実現できなかった。傾きながらも空中に浮き上がることはできたが、浮いたままその場に留まることができなかった。

機体の角度は算出できており、モニタ上ではそれに対してのモータ出力もうまくいっているが、実動作として飛行が何故か不安定であった。このことから、機体、プログラム共に更なる改良が必要だとわかった。

7 反省・考察

ホバリングが安定しない理由として、以下の点が考えられる。

- ・機体の重心が不安定で、離陸時すぐに傾いている
- ・姿勢角の算出に時間がかかり、応答が追いついていない
- ・PID 制御のゲイン調整が甘い

特に機体の重心という点においては、重要度が高いと考えている。なぜなら重心が機体の真ん中にあれば、離陸時に大きなモータ出力の補正が無くても、安定して浮くことができるため、飛行の安定に繋がると考えられるからだ。

他にも、私の知識やプログラミングスキルが足りなかったことで、コーディングに時間がかかってしまったということも考えられ、制御工学の奥深さを体感した。

8 おわりに

私の創造研究には、本研究以外にもう一つの研究があり、限られた時間の中二つの研究を進める事は非常に難しかったと感じています。そんな中ですが、共同開発をしてくれた仲間と先生方に、大変感謝しています。満足いかない結果になってしまいましたが、電子回路設計やプログラミングのスキルが飛躍的に向上しました。