

1. UWB モジュールの測距キャリブレーション

UWB の Tag のプログラムは測距周期向上などのマイナーチェンジ等は各所、修正してきたが、要である測距のキャリブレーションに至っては赤堀さんが作製した当時より手をつけずにいた。以前のバージョンだと、Anchor に電源を入れる順番により、各 UWB 間で計測される距離が微妙に違うという経験を赤堀さんがしていたため、電源を入れる順番が違っても同じ様に測距できる様にプログラムを改良してあった。しかし、全ての Anchor を Teensy に換装後、自身でテストしたところ、つなぐ順番によって計測される距離が変わったり、各 Anchor 間で大きく数値に違いがあったりするということはなかった。そこで、以前までの修正プログラムを廃止し、実際の距離と UWB により測距される距離とで誤差を引くことでキャリブレーションを行い直した。Fig. 1 にキャリブレーションした後の測距結果を示す。リファレンスは 1m～5m までの各 1m 刻みである。なお、50cm 以下付近では大きく測距精度が落ち、測距される値が負の値になるなどして信頼性が欠けるため（実用上問題ないが）、1m 以下では測距していない。図より分かるようにスペック通りの十分な精度を持っていることが分かる。キャリブレーションでは各 UWB にて同じ値を引いたが UWB の個体差を考慮して違う値を用いれば UWB 間の誤差をもう少し詰められるかもしれない。

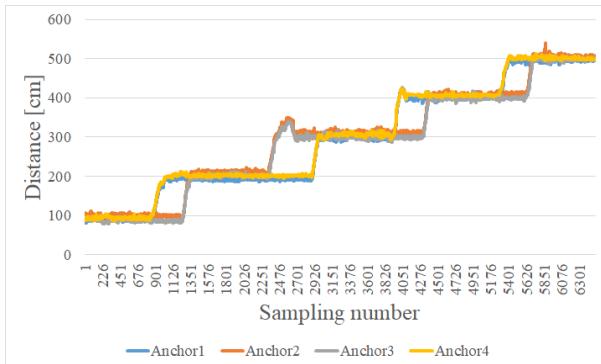


Fig.1: キャリブレーション後の距離測定結果

2. ヨー角の推定値の見直し

位置制御がうまくいっていないことを鑑みて、ヨー角の推定が上手くいっていないことを疑った。そこで推定方法を見直した。ヨー角の値はカルマンフィルタを用いた位置の推定の際や位置を制御する際に大きく関与しているため、UWB を用いた位置の推定がいくら上手くいっていてもヨー角がおかしいと、それらに大きな影響を及ぼす。NAVIO2 に搭載される地磁気センサのキャリブレーションが面倒なため、今まで角速度の積分値のみを用いてヨー角を算出していた。したがって、誤差が生じてヨー角がドリフトしていた。そこで、カルマンフィルタにおける角速度を積分してヨー角を推定する状態方程式はそのままにし、観測方程式のヨー角の観測値に Pixracer からのヨー角の推定値を用いるようにカルマンフィルタを組み直した。Fig. 2 に模式図を示す。Raspberry Pi で推定するヨー角は高い周波数において推定できるが、ドリフトを含む。それを Pixracer からの地磁気センサも用いた推定値で補正することで精度の高いヨー角が得られると考えた。Pixracer と Raspberry Pi 間の通信は 10Hz 程度と低いが、Raspberry Pi で高い周波数で推定できていることやヨー角はピッチ角やロール角ほど高い周波数で推定する必要がないため、問題ないと考えている。これにより、以前まではヨー角の推定値が 1 フライト後（特に制御なし）に 30 度ほどドリフトしていたのが、10 度までに収まった。また、Ardupilot における ESC のキャリブレーションも行い、ブラシレスモータごとの出力差が小さくなつたため、ヨー角がドリフトして回るということも軽減した。

ているため、UWB を用いた位置の推定がいくら上手くいっていてもヨー角がおかしいと、それらに大きな影響を及ぼす。NAVIO2 に搭載される地磁気センサのキャリブレーションが面倒なため、今まで角速度の積分値のみを用いてヨー角を算出していた。したがって、誤差が生じてヨー角がドリフトしていた。そこで、カルマンフィルタにおける角速度を積分してヨー角を推定する状態方程式はそのままにし、観測方程式のヨー角の観測値に Pixracer からのヨー角の推定値を用いるようにカルマンフィルタを組み直した。Fig. 2 に模式図を示す。Raspberry Pi で推定するヨー角は高い周波数において推定できるが、ドリフトを含む。それを Pixracer からの地磁気センサも用いた推定値で補正することで精度の高いヨー角が得られると考えた。Pixracer と Raspberry Pi 間の通信は 10Hz 程度と低いが、Raspberry Pi で高い周波数で推定できていることやヨー角はピッチ角やロール角ほど高い周波数で推定する必要がないため、問題ないと考えている。これにより、以前まではヨー角の推定値が 1 フライト後（特に制御なし）に 30 度ほどドリフトしていたのが、10 度までに収まった。また、Ardupilot における ESC のキャリブレーションも行い、ブラシレスモータごとの出力差が小さくなつたため、ヨー角がドリフトして回るということも軽減した。

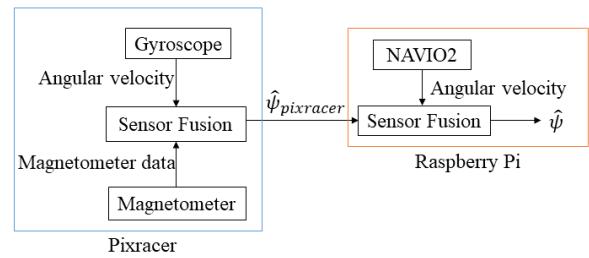


Fig.2: ヨー角の推定模式図

3 地点でのドローンの移動

上記のような UWB モジュールのハード、ソフトにおける改良やヨー角の推定方法の見直しをした後、1 点での位置保持実験を行ったところ、体感として以前より保持性能が上がったと感じたため、3 点での移動をする実験を行った。プロポのトリガーを切り替えることで原点 $(x, y, z) = (0.0, 0.0, 0.0)$ 、Point1 : $(0.0, -0.8, 1.0)$ 、Point2 : $(0.0, +0.8, 1.0)$ の 3 点を移動させた。なお、スロットル制御に関しては以前の実験と同様に緊急時に備え、自分で行った。位置の推定値を Fig. 3 に示す。また、その実験時の指令信号も合わせて示す。-1 の時が Point1 の指令を与えていた時であり、+1 の時が

Point2 の指令を与えていた。そして Fig. 4 に $x - y$ 平面内での移動軌跡も示す。

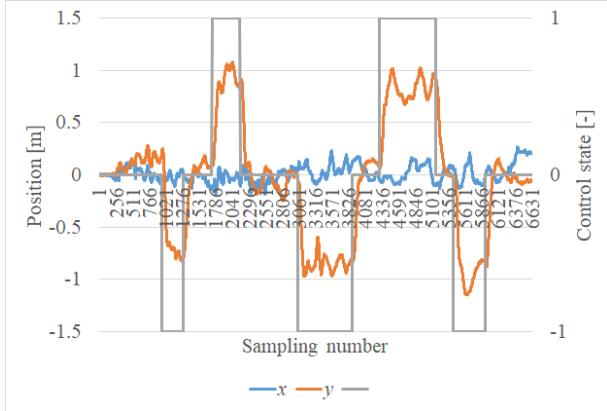


Fig.3: 3 点での位置制御結果

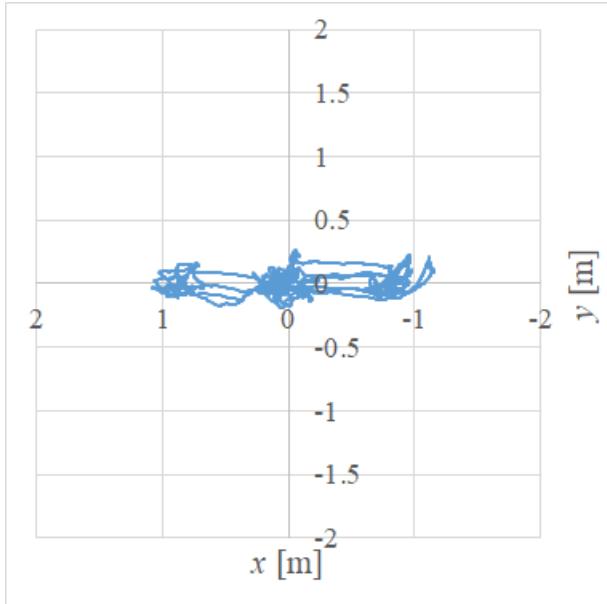


Fig.4: $x - y$ 面内での移動軌跡

図より、位置の指令を出すと約 20cm 以内の精度で位置を保持できていることが分かる。また、常に目標値が 0 の x 方向も多少の振動はあるものの、0 に近い値を取れている。また、この実験とは別に 3 点での位置保持に加え、前回、単体だと上手くいった高さ制御も加えてみたが、やはり同時だとまだまだうまくいかなかった。したがって、高さ制御も位置保持制御と同時にできるよう詰めていく予定である。

4. 今後の予定

- ・高さ制御及び、位置保持制御を同時に出来るよう調整
- ・オプティカルフローセンサの制御への組み込み方の検討