

1. 位置制御プログラムの大まかな完成

以前よりカルマンフィルタにオプティカルフローセンサから得た値も組み込むことにより、位置推定値の精度が向上したといった結果を得ているが、その推定方法を用いたクアッドの位置制御用のプログラムを書いた。大まかには完成したが、以前に位置制御実験を行った際に比べ、機体のパラメータ（重量、スラスト、応答性）や位置推定方法が変わっているので位置制御用のPIDゲインは以前に比べ小さな値から始めようと考えている。

2. 分散値を調整してのカルマンフィルタのシミュレーション

今まででは推定した位置座標を基にクアッドの位置をリアルタイムで制御する必要があるため、カルマンフィルタはオンボードのRaspberry Pi上でリアルタイムで計算していた。しかし、同一の実験試行時における、センサの種類や分散の値を変更した際のフィルタリング結果をオフラインでシミュレーションする必要があるため、matlabを用いてカルマンフィルタを計算可能な環境を構築した。また、その際、全てのセンサ値がカルマンフィルタの計算に必要なため、Raspberry Pi側のプログラムも改め、加速度センサ、ジャイロセンサの値もロギングするようにした。次のfig. 1に分散を[加速度センサ, UWB, オプティカルフローセンサ] = [0.004, 0.01, 0.0006]とした際の推定結果

[加速度センサ, UWB, オプティカルフローセンサ] = [0.004, 0.01, 0.0006]とした際のUWB+IMU+Optにより推定した結果、[加速度センサ, UWB] = [0.4, 0.02]と設定した際のUWB+IMUにより推定した結果、オプティカルフローセンサを単純積分した結果を合わせて示す。なお、加速度センサ x, y, z の3軸、UWBアンカーの4つ、オプティカルフローセンサの x 軸方向推定速度、 y 軸方向推定速度とそれぞれセンサは複数のセンサ値を持つが、全て同様の分散値を設定している。

この結果は同一の試行時のデータを基に推定に組み込むセンサの種類を変えた結果である。オレンジ色のUWB+IMU+Optの計算における分散の決め方はシミュレーションにおいて、定常値がUWBより得た座標と近く、移動中の軌跡が滑らかになるように分散を調整した。それぞれの結果はまず、青色のUWB+IMUにて推定した結果は定常値（ターゲットポイントとなる4点において）それほど大きな分散は見られないが、移動中の軌跡はレーザーポインタが地面の線をトレースするように動かしたにも関わらず、波打っていることが分かる。この原因はUWBから得られる距離の計測値に分散が乗っており、それが4つのアンカ分相乗するためこういった推定値の分散が生じているためだと考えられる。次にオレンジ色のオプティカルフローセンサ値も参照した結果は定常値は青色の結果と同様に一定の値を

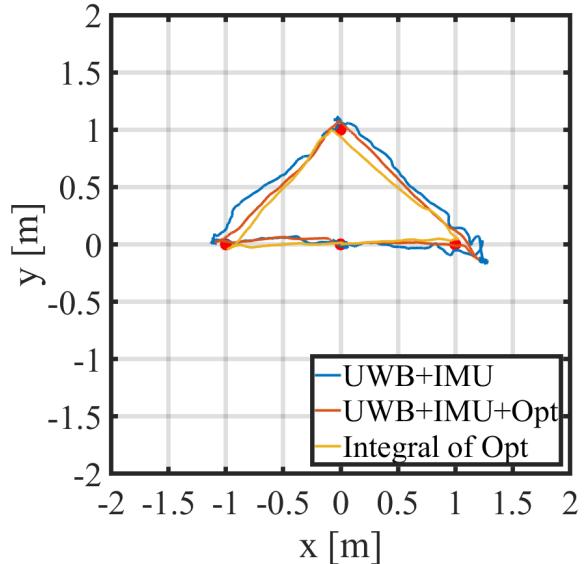


Fig.1: [加速度センサ, UWB, オプティカルフローセンサ] = [0.004, 0.01, 0.0006]とした際の推定結果

取れていることに加え、移動中の軌跡はオプティカルフローセンサの効果により分散値を低く抑えられていることが分かる。ここで $(x, y) = (0, 1)$ 付近における拡大図を次に示す。

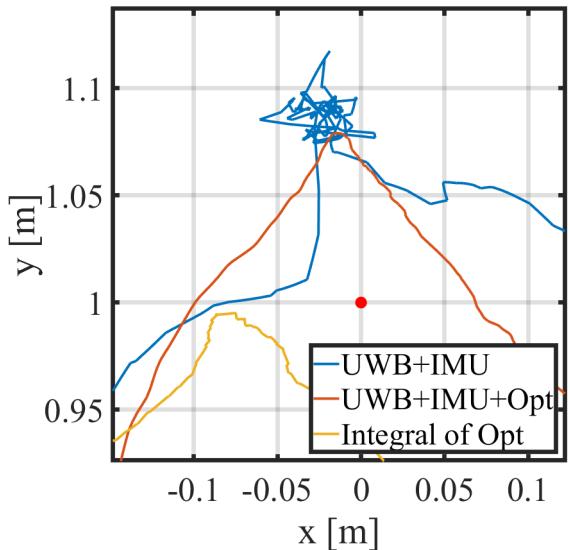


Fig.2: fig. 1 の $(x, y) = (0, 1)$ における拡大図

オプティカルフローセンサを参照した結果、していない結果共に定常値は目標値からおよそ 7.6cm ほど誤差

がある。この原因は UWB アンカの設置誤差によるものと考えられる。そして、オプティカルフローセンサを参照していない結果は、定常状態においても直径 5cm ほどの分散が生じているが、参照した結果は定常状態においても分散がなく、曲がり角が綺麗に出ているのが分かる。

次にあえてオプティカルフローセンサの分散より UWB の分散値を極端に下げてシミュレートした結果を fig. 3 に示す。分散は [加速度センサ, UWB, オプティカルフローセンサ] = [0.004, 0.000001, 0.06] である。

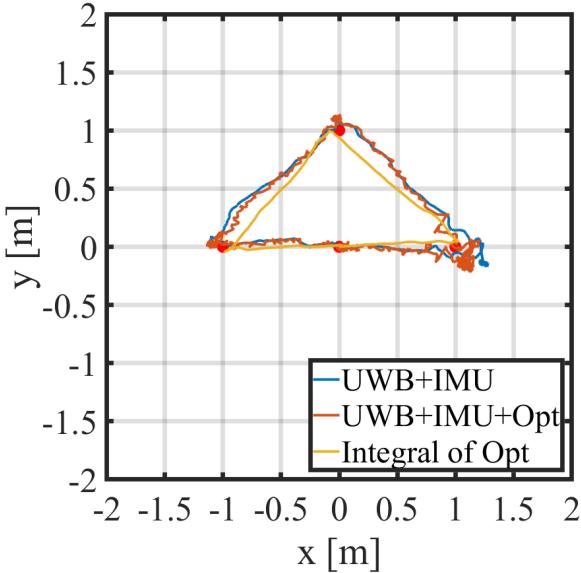


Fig.3: [加速度センサ, UWB, オプティカルフローセンサ] = [0.004, 0.01, 0.00006] とした際の推定結果

もう既に他の結果から予想できるが、分散の大きい UWB の値に大きく引っ張られるため、定常値の分散は大きく、軌跡も荒く、ギザギザしたものになる。オプティカルフローセンサの値を参照していないフィルタにおいても IMU の分散より UWB の分散を大きく下げる同様の結果を得られた。

最後に分散を [加速度センサ, UWB, オプティカルフローセンサ] = [0.004, 0.1, 0.0006] とオプティカルフローセンサの分散を小さくした際の結果を示す。

軌跡や角は滑らかであるが、定常状態における座標が UWB の参照具合を大きくした時よりも大きく右下の方へ外れていることが分かる。この理由はオプティカルフローセンサ成分が強めになったことで推定値の分散は小さくなつたが、UWB による補正が小さくなつたため、積分した際のドリフトの影響が大きくなつたためであると言える。

これらのシミュレーションでは、今まで Raspberry Pi を用いてリアルタイムで計算していた値をオンラインで色々と値を変えながら推定結果を得られるため、この Matlab のカルマンフィルタを作製した意義は大きかったと言える。また、これら 3 つの結果より、オプティカルフローセンサを今までの推定フィルタに組み込み、適切に分散を調整すれば、定点ホバリング時の推定値の分散が原因のふらつきを抑えることが可能になるとを考えている。修論の結は今述べたようにオプティカルフローを組み込んだ位置推定、位置制御を用い、定点でのホバリ

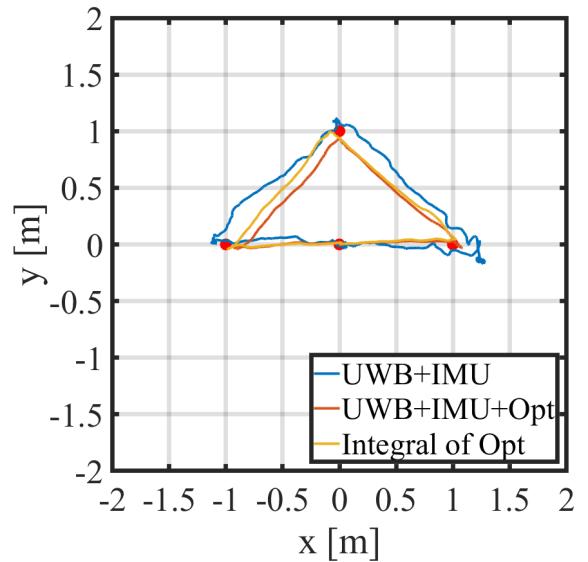


Fig.4: [加速度センサ, UWB, オプティカルフローセンサ] = [0.004, 0.1, 0.00006] とした際の推定結果

ング飛行、3 点間を順に移動する飛行制御を行い、ステレオカメラで撮影した真値と比較し、評価することでオプティカルフローセンサを組み込んだことで精度が向上したということで締めたいと現在考えている。

3. ステレオカメラを用いた画像（動画）の撮影に関して

以前、瀬戸が脚の角度を解析するのに動画ではなく、画像(jpg)を用いる方法を提言していたが、これからは自身もこの方法を試そうと考えている。というのも以前までは、mp4 にて撮影した動画をわざわざ結合し、その後 PNG に変換するという作業を行ってから画像解析という手法をとっていた。画像解析自体はそれほど時間はかかるないが、結合と画像分割に 1 分ほどの動画で 30 分ほど時間を要していた。したがって、最初から画像で撮影し、解析が可能ならかなり時間を削減可能である。プログラムを修正し、試してみて可能ならこの方法を正式に採用しようと考えている。

4. 今後の予定

- ・飛行制御の準備
- ・クアッドを飛行させ、ステレオカメラを用いた位置の真値を計測可能かの確認