

研究目的:GPS信号を受信できない環境で、災害時などに、無人航空機を用いて屋内の探索を行い、被害を最小限にする。

昨年度までの研究:

屋内を飛行する無人航空機の製作とその制御に関する研究を行った。Navio2をフライトコントローラーとして使用し、自作した機体を屋内で自律飛行制御させた。特に簡単な演算処理での自己位置を維持する制御に関して研究を行った。

制御システムの概要:

一般的に無人航空機の操作は人間が行う。しかし、今回制作した制御システムでは人間が行う操作を、コンピュータがさまざまなセンサ値を処理し、飛行コースを計算しフライトコントローラーに操作命令することで飛行する。屋内を探索するために、探索を行いたい屋内に赤線を引き、機体に取り付けられているカメラで、赤線を認識し、飛行制御を行う。

環境:

センサ処理: Tello
操作命令:
OS: Ubuntu 18.04.4 LTS
CPU: Intel Core i3-4000M
実装RAM: 8GB
使用言語: Python 2.7.17
フライトコントローラー: Tello

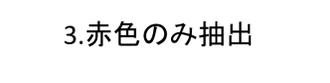
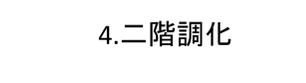
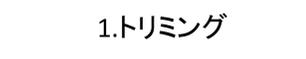
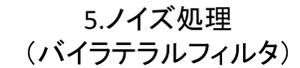
使用機体: Tello
機体寸法:
98×92.5×41 (mm)
重さ: 80 (g)
カメラ解像度:
5MP(2592×1936)
視野角: 82.6°



画像から赤線を認識する方法

赤線の認識:

以下の1~6のように入力画像から赤色のみを認識する処理を行う。6の画像から四角を認識する処理を行う。認識された四角は最も外側の輪郭の角、四点の座標を出力し、そこから対角線の交点と長辺の傾きを計算し制御に用いた。



結果画像

6.二階調化

5.ノイズ処理
(バイラテラルフィルタ)

1.トリミング

2.彩度を上げる

4.二階調化

3.赤色のみ抽出

赤線をたどるために最適なプログラムの作成

実験1: 直線を安定して飛行する

目的:

作成した3種類の赤線をたどるプログラムのうちどれが安定して飛行できるか調べる。

作成したプログラムの概要

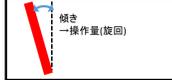
プログラム1: 機体の位置を、線の中心X座標が画面の中央に合わせる比例制御を行う

プログラム2: 機首を、線の傾きに合わせる比例制御を行う

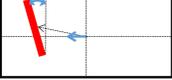
プログラム3: 線の中心座標へのベクトルと線の傾きを組み合わせた比例制御を行う



プログラム1



プログラム2



プログラム3

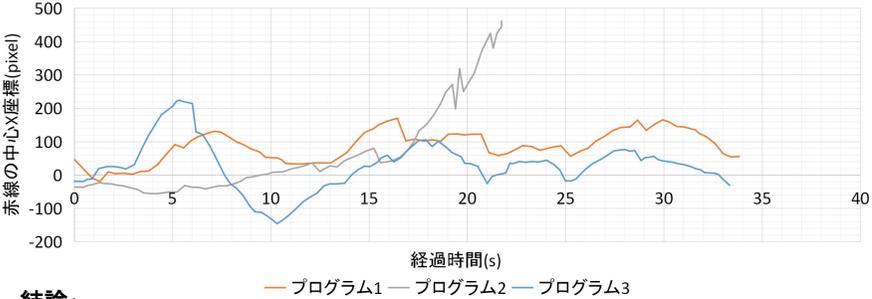
実験方法:

赤線を直線に10mのコースを飛行させる。

- 赤線10mの端で離陸命令を実行する。
- 高度を0.3mまで降下する。
- 赤線の端に到達する又は赤線を捉えられなくなるまで制御プログラムを実行する。経過時間、線の中心座標、傾き、機体の加速度・高度を随時記録する。
- 3つのプログラムで1~3の手順を行う。

結果:

実験1 結果



結論:

10mの赤線をたどる制御はプログラム3が最も安定して飛行できる。

実験2: 直線に安定して復帰する

目的:

作成した3種類の赤線をたどるプログラムのうち、機体が線から外れた時、どの制御が最も早く、安定して線上に復帰するかを調べる。

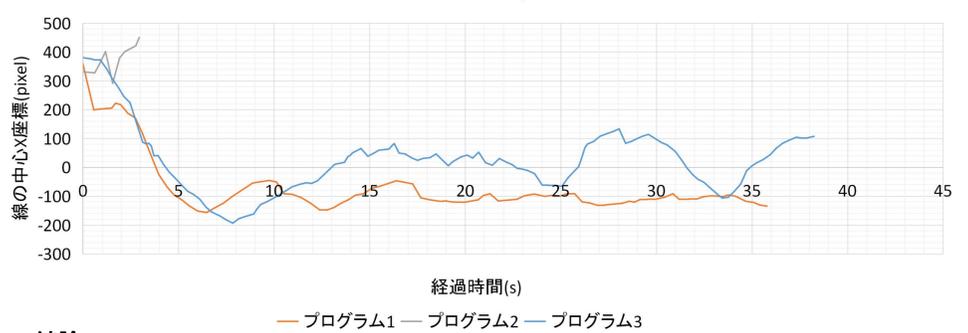
実験方法:

赤線を直線に10mのコースを飛行させる。スタート地点は赤線の端から、右に0.3m移動した点とする。

- スタート地点で離陸命令を実行する。
- 高度を0.3mまで降下する。
- 赤線の端に到達する又は赤線を捉えられなくなるまで制御プログラムを実行する。経過時間、線の中心座標、傾き、機体の加速度・高度を随時記録する。
- 3つのプログラムで1~3の手順を行う。

結果:

実験2 結果



結論:

赤線にずれた位置から赤線上に復帰し、10mの赤線をたどる制御はプログラム3が最も赤線に近い位置を飛行できる。ただし、振動現象がみられる。

屋内を探索する

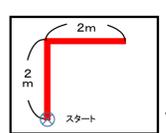
実験3: カーブを安定して曲がる制御

目的:

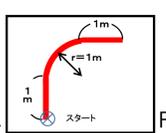
プログラム3がカーブなどにも対応できるかを調べる。

実験方法:

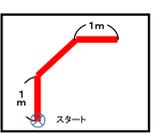
以下の図のように直角、円形、45°の角2つで赤線のカーブを作り、それらが角を曲がるか検証した。スタート地点はそれぞれのカーブの手前1m地点とする。またスタート地点からカーブ、及びカーブからさらに1m進んだ地点までは赤線を延長した。



直角コース



円形コース



45°の角2つのコース

- スタート地点で離陸命令を実行する。
- 高度を0.3mまで降下する。
- 赤線の端に到達する又は赤線を捉えられなくなるまで制御プログラムを実行する。経過時間、線の中心座標、傾き、機体の加速度・高度を随時記録する。
- 3種類のカーブでそれぞれ1~3の手順を行う。

結果:

直角	円形	45°の角2つ
成功	成功	成功

ただし、直角ではカーブを曲がったのち線上を安定するのに時間を有した。

結論:

プログラム3を用いてすべてのカーブを曲がるのが可能である。

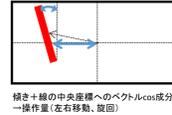
実験4: 実験3からの改良

目的:

実験3の結果から、画像中心から線の中心座標までの距離を、機首の方向制御に追加した。この実験では、これの効果を検証する。

実験方法:

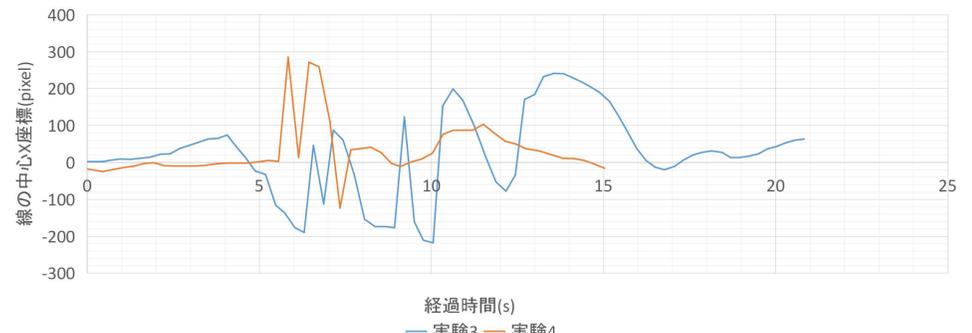
- スタート地点で離陸命令を実行する。
- 高度を0.3mまで降下する。
- 赤線の端に到達する又は赤線を捉えられなくなるまで制御プログラムを実行する。経過時間、線の中心座標、傾き、機体の加速度・高度を随時記録する。



実験4で使用するプログラム

結果:

実験4 結果



結論:

線の中心座標と画像中心との距離を含めた、機首の方向制御は十分に効果がある。

結論と今後の展望

ここまでの実験結果から、赤線をたどる方法を用いた屋内の探索は可能といえる。よって、今後は実際に建物内に赤線を引き、様々な状況が複合的にある環境での実験を行いたい。また、被災者捜索やその支援を行うことが目的であるため、被災者を見つけることなども実験していきたい。

参考文献

- GitHub, "hanyazou/TelloPy", <https://github.com/hanyazou/TelloPy>
- OpenCV.jp, "OpenCV 2.2 Python Reference --opencv v2.2 documentation", http://opencv.jp/opencv-2.2_org/py/#
- 井戸伸彦, "Telloプログラミング覚書", http://staff.gku.ac.jp/~ido/doc/sem/tello_programming.pdf
- 渡辺光貴, "視覚センサを用いたライトレースのための自律移動ロボット", 第51回自動制御連合講演会