

拡張カルマン・フィルタの効果を確認する

滝田 好宏

今回作成した100g未満ドローンは、制御アルゴリズムとして拡張カルマン・フィルタ(EKF)を使用しています。外乱の影響を受けやすい小型ドローンでも安定して飛行できるようになったのは拡張カルマン・フィルタを始めとする制御アルゴリズムの進歩のお陰です。

そこでこの章では、拡張カルマン・フィルタについて数式と、シミュレーションを行った結果を見ていきます。

拡張カルマン・フィルタのソースコードを見る

フライト・コードArduPilotは、拡張カルマン・フィルタを使った制御を行っています。

ArduPilotをダウンロードして開発環境を整えることで、拡張カルマン・フィルタをはじめとするさまざまなコードを見ることができます。そこでArduPilotの開発環境をインストールして実際に使っているソースコードを見てみましょう。

Windows PCの場合は仮想環境にUbuntuをインストールしてから行ってください。詳細は下記URLを参照してください。

<https://ardupilot.org/dev/docs/building-setup-linux.html>

● ArduPilot開発環境のインストール

Ubuntuのインストール環境を整えます。

```
$ sudo apt-get update
$ sudo apt-get install git
$ sudo apt-get install gitk git-gui
インストールする場所(一般的に$HOME)に移動し、下記のように入力します。
$ git clone https://github.com/
    Ardupilot/ardupilot.git
$ cd ardupilot
$ git submodule update --init
    --recursive
$Tools/environment_install/install-
```

```
/ardupilot
├── AntenaTracker アンテナ・トラッカ
├── ArduCopter マルチコプタ
├── ArduPlane 固定翼、垂直離着陸機
├── Rover 車両ロボット
├── build コンパイル結果
├── libraries 共通のプログラム
│   ├── AP-NavEKF 拡張カルマン・フィルタ関連
│   ├── Models
│   ├── AttErrVecMathExample
│   ├── Common 共通のmファイル
│   ├── GimbalEstimatorExample
│   └── testData テスト・データ
├── module
├── chibiOS リアルタイム OS
├── mavlink MAVLink関連
└── waf 開発環境
```

図1 ArduPilot開発環境のフォルダ構成

```
prereqs-ubuntu.sh -y
$ . ~/.profile
```

● コンパイル

ターゲットとなるボードの名前を確認します。例えばHolybro Kakute F7はKakuteF7となっているので、次のように入力します。

```
$ ./waf configure --board KakuteF7
マルチコプタでビルドしてフラッシュ・メモリに書き込むには、KakuteF7とUSBで接続してから次のコマンドを実行します。
```

```
$ ./waf copter --upload
コンパイルが完了したら、書き込みが開始されます。ただし、事前にブートローダが書き込まれている必要があります。
```

拡張カルマン・フィルタの中身

● ArduPilotのフォルダの構成

ダウンロードしたArduPilotのフォルダ内の構成は図1の通りです。