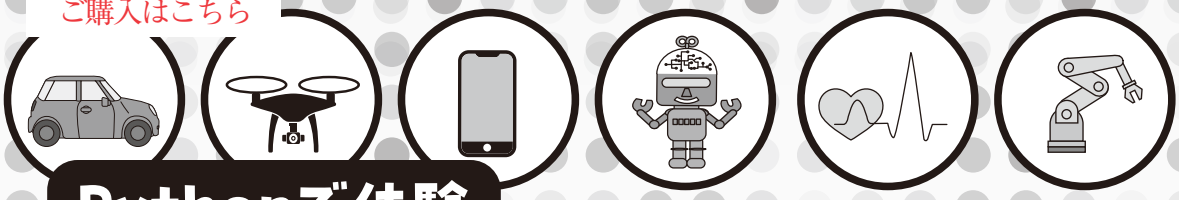

Pythonで体験 カルマン・フィルタ入門

ダウンロード・データあります

第2回 カルマン・フィルタで1次元運動を推定①

廣川 類

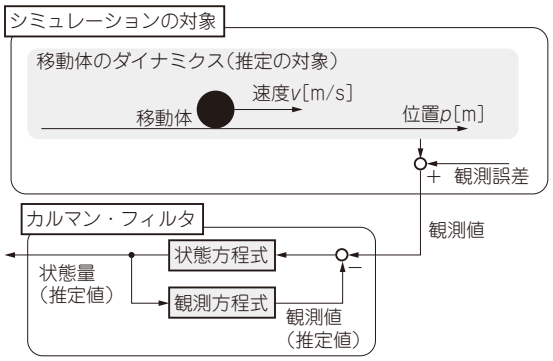

図1 移動体シミュレーション・モデルとカルマン・フィルタ

● カルマン・フィルタとは

カルマン・フィルタは，移動体のダイナミクスや動きの不確定性，センサ誤差の確率モデルを考慮して，最適な推定を行う強力なセンシング技術です．

● 今回のテーマ

今回から2回に分けて，点の1次元運動についてカルマン・フィルタを使って位置の推定を行います．なお，カルマン・フィルタの理論的詳細については触れずに概念を説明することで，考え方を理解していただこうと思います．

実行環境の準備

Python3のコードをJupyter Notebookで実行し，対話的に結果を確認しながら解説していきます．

本稿では，Python3の実行環境がインストール済みであることを想定して進めます．筆者は執筆時点で最新のバージョンであるPython3.11を使用しています．まず，PythonでJupyter Notebookを実行する環境を整えます．コマンド入力環境で次のように入力します．なお，ここではPython標準のパッケージマネージャであるpipを利用しますが，例えばcondaを利用することも可能です．

```
$ pip install notebook
```

同様に数値計算およびプロット，フィルタ計算に利用するパッケージをインストールします．

```
$ pip install numpy, matplotlib
```

その後，以下のようにJupyer Notebookを起動します．

```
$ jupyter notebook
```

ウェブ・ブラウザが自動的に起動し，Jupyter Notebookがオープンされます．

カルマン・フィルタを使ってみる

● カルマン・フィルタが必要とされる理由

移動体の運動を推定する最も簡単な例として，**図1**上に示す1次元の直線運動を取り上げます．**図1**上部が数値シミュレーションにより移動体の運動を求める部分で，センサにより観測した観測値を出力します．**図1**下部は，カルマン・フィルタにより観測値を入力として移動体の運動を推定する部分です．

実例として，直線道路を走行する自動車の位置をGPSで測定することを想定してみてください．自動車に搭載されているGPS受信機は，複数のGPS衛星との距離を測定して，自動車の位置を求めます．しかし，測定された観測値(距離)にはセンサ誤差が含まれており，位置を推定する精度が劣化します．また，移動体の運動は，地面との摩擦や風などの影響により，モデル化した運動方程式(ニュートンの法則：$F = ma$など)の値通りにはなりません．こうした運動モデルやセンサの誤差の影響を評価するため，運動モデルの誤差を考慮した移動体のダイナミクス(運動モデル)から観測誤差(センサ誤差)を含む観測値を模擬的に生成します．

● カルマン・フィルタの概要

カルマン・フィルタには，推定対象のダイナミクスを模擬した状態方程式，出力を模擬した観測方程式が含まれていて，移動体の位置，速度のようなシステムの状態を推定できます．このシステムの状態を表す量は状態量と呼ばれます．さらに，センサの誤差や移動