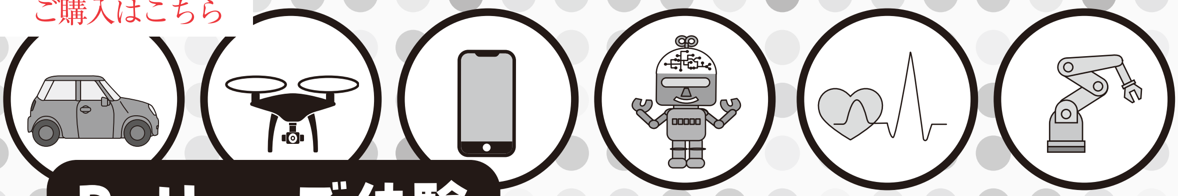
ご購入はこちら



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3回 カルマン・フィルタで1次元運動を推定②

廣川 類

● **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**

図1に示す，点の1次元運動について，カルマン・フィルタで状態量（点の位置や速度）を推定します．この工程を次に示します．

①ダイナミクスのモデル化
②センサのモデル化
③数値シミュレーションの準備（ステップ時間，センサ誤差モデル化など）
④数値シミュレーションの実行（誤差を含まない移動体の真の動きを状態方程式より求め，移動体に搭載されたセンサで観測されるデータをセンサ誤差を考慮して模擬的に生成）
⑤カルマン・フィルタによる推定
⑥カルマン・フィルタによる推定値を数値シミュレーションの真値と比較することで推定精度を評価する

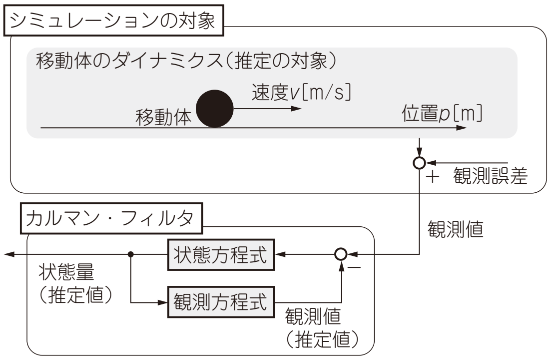


図1　移動体シミュレーション・モデルとカルマン・フィルタ

前回，①から④の運動の数値シミュレーションにより移動体の運動を求めるところ（**図1**上部）まで解説しました．今回は，⑤のカルマン・フィルタを使った位置と速度の推定と⑥の推定値の評価を行います（**図1**下部）．

カルマン・フィルタを使った位置と速度の推定

使用する計算式を定義

● **状態量ベクトルを定義**

ダイナミクス・モデルを定義するために，システムの内部的な状態を表す変数として状態量ベクトル$\boldsymbol{x}$を定義します．ここで，状態量$\boldsymbol{x}$は，確率事象として定義されるものとし，平均値$\bar{\boldsymbol{x}}$および共分散行列$\boldsymbol{P}$を期待値（E[.]）により，次のように定義します．

$$\bar{\boldsymbol{x}} = E[\boldsymbol{x}] \cdots\cdots (1)$$

$$\boldsymbol{P} = E[(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{x}-\bar{\boldsymbol{x}})^{\mathrm{T}}] \cdots\cdots (2)$$

今回扱う，**図1**の1次元の移動では，速度v_kで運動する物体の位置p_kは前ステップの位置p_{k-1}に，ステップ間の位置変化量$T_s v_{k-1}$を足すことにより得られるとします（T_s：サンプリング時間）．このとき，状態量ベクトル$\boldsymbol{x}_k = [p_k \;\; v_k]^{\mathrm{T}}$とします（上付きのTは転置行列）．状態量ベクトルを期待値で表す場合と位置などの具体的なパラメータを使って表す場合の関係についてはコラム1を参照ください．

● **推定の対象となる状態方程式（ダイナミクス・モデル）を定義**

今回扱う推定対象のダイナミクスは線形モデルで記述し，ダイナミクス・モデルをステップごとに離散的に記述するものとしました（線形離散系と呼ぶ）．このとき，kステップ目の状態量ベクトルを$\boldsymbol{x}_k$として，