

カルマンフィルタ

はじめに

カルマンフィルタはリアルタイム時系列処理の手法である。特に観測系(センサ)のノイズが大きい場合に有効である。運動方程式といった物理法則を組み込むことで高精度の推定を可能とするフィルタとなる。カルマンフィルタは状態方程式と観測方程式の2つの方程式で構成される。

* 状態方程式

運動方程式といった方程式を当てはめてモデル化する。状態値は推定したいパラメータ(列ベクトル)とする

* 観測方程式

状態値を観測値に変換するための方程式

カルマンフィルタ

導出など、詳細は専門書[1]を参照されたい。本稿では天下りの的にカルマンフィルタのアルゴリズムを述べる。まず、状態方程式は(1)、観測方程式は(2)である。

$$X^{(k)} = AX^{(k-1)} + B_u u^{(k-1)} + Bv^{(k-1)} \quad (1)$$

$$y^{(k)} = C^T X^{(k-1)} + w^{(k-1)} \quad (2)$$

各項の説明を表1に示す。

X	n 次元の列ベクトル	状態推定値
A	$n \times n$ の行列	状態推定値を時間更新する関数
u	スカラー	制御あるいは外乱のシステムへの入力値
$B_u u$	n 次元の列ベクトル	制御あるいは外乱の項
v	スカラー	状態推定値に係るノイズ
Bv	n 次元の列ベクトル	状態推定値に係るノイズの項
y	スカラー(1変数)	観測値
C	n 次元の列ベクトル	状態推定値を観測値に変換する関数
w	スカラー	観測系のノイズ
k	序数	k 回目の時間更新を表す($k = 1, 2, \dots$) $k = 0$: 初期値

表1 各項の説明

いくつか補足する。

* A, B, C は時間に依らない. 既知かつ定常として与える.

* v, w は互いに独立で正規分布のノイズとする. すなわち, (3), (4)である.

$$v = N(0, \sigma_v) \quad (3)$$

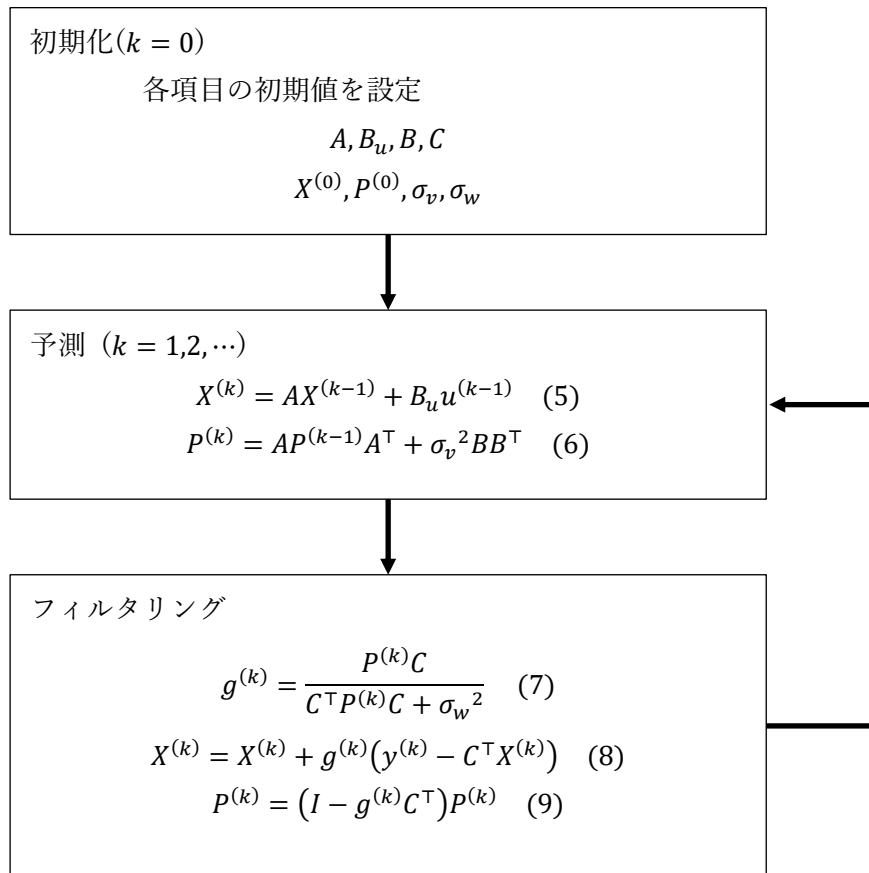
$$w = N(0, \sigma_w) \quad (4)$$

$$N(\text{平均値, 標準偏差})$$

これらはカルマンフィルタのシステムを調整するパラメータである.

項目が多く, 初学にて戸惑うことがある. これは後述のシミュレーションで確認いただきたい. なお, 観測値 y は多変数とすることもできるが[1], 本稿では1変数のみ扱う.

早速, カルマンフィルタのアルゴリズム(5)~(9)を図1に示す.



P : 状態推定値の誤差共分散行列
 g : カルマンゲイン

図1 カルマンフィルタのアルゴリズム

$P^{(0)}$ だが、事前情報がない場合、(10)とする[1].

$$P^{(0)} = \gamma I \quad (10)$$

$$\gamma = 0 \sim 1000$$

さらにカルマンフィルタのブロック図を示す(図2).

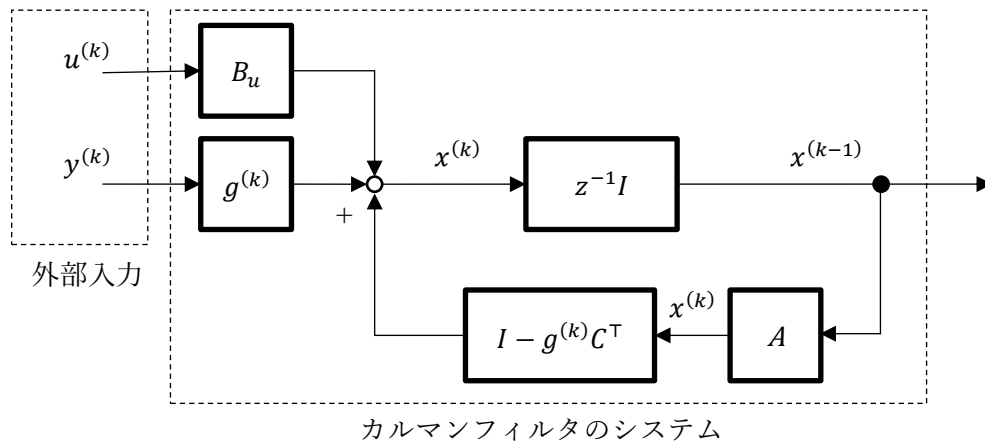


図2 カルマンフィルタのブロック図

アウトラインを述べる(もう少し詳しくは後述).

- ① 直前の状態値 $x^{(k-1)}$ から A を乗じて新しい状態値 $x^{(k)}$ を予測
- ② カルマンゲインによる予測値の修正
- ③ さらに、観測値との誤差, 外乱あるいは制御量を加えて修正

予測により事前推定値を算出, フィルタリングにより事後推定値を算出する手順と説明される[1]. なお, プログラミング上, 事前推定値と事後推定値を別ける必要性はないため, 本稿の数式に事前, 事後の表記をしていない. 他に時間更新の序数 k についてもプログラミング実装に沿う表記としている. 以上, ご承知おきいただきたい.

シミュレーション(自由落下+空気抵抗)

本稿では自由落下の運動方程式に空気抵抗を加えたシミュレーションを実施する。まず自由落下の位置と速度は(10)と(11)である。

$$x = x_0 + v_0\Delta t + \frac{1}{2}g\Delta t^2 \quad (10)$$

$$v = v_0 + g\Delta t \quad (11)$$

Δt : 経過時間差分

x_0 : 初期位置

v_0 : 初速

g : 重力加速度

x : Δt 経過時点での位置

v : Δt 経過時点での速度

シミュレーションでは見かけの重力加速度を a とする。空気抵抗 k による力のつり合いは(12)となる。

$$ma = mg - kv\Delta t \quad (12)$$

k : 空気抵抗係数

簡単のため質量に応じた空気抵抗 k' を導入し、重力加速度の空気抵抗による減衰をモデル化する(13)。

$$a = g - \frac{k}{m}v\Delta t = g - k'v\Delta t \quad (13)$$

$$k' = \frac{k}{m}$$

$$a_0 = g$$

終端速度時: $a = 0$, すなわち $a = g \rightarrow 0$

状態変数として位置, 速度, 加速度とすれば, 漸化式(時間更新)は(14)~(16)となる。

$$x^{(k)} = x^{(k-1)} + v^{(k-1)}\Delta t + \frac{1}{2}a^{(k-1)}\Delta t^2 \quad (14)$$

$$v^{(k)} = v^{(k-1)} + a^{(k-1)}\Delta t \quad (15)$$

$$a^{(k)} = a^{(k-1)} - k'v^{(k-1)}\Delta t \quad (16)$$

空気抵抗の項は外乱として扱い, 行列で表せば(17)となる。

$$X^{(k)} = AX^{(k-1)} + B_u u^{(k-1)} \quad (17)$$

$$X^{(k)} = \begin{bmatrix} x^{(k)} \\ v^{(k)} \\ a^{(k)} \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_u = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$u^{(k)} = -k'v^{(k-1)}\Delta t \quad (a^{(k-1)} + u^{(k)} > 0)$$

さらに加速度 a に揺らぎを与える。これを偏差 σ_a とする(18)。

$$X^{(k)} = AX^{(k-1)} + B_u u^{(k-1)} + Bv \quad (18)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$v = N(0, \sigma_v) = N(0, \sigma_a \Delta t)$$

現実の物性では速度に比例した加速度の揺らぎだが、シミュレーションモデルでは単純に σ_a で表現する。終端速度に達した時点($a^{(k)} - k'v^{(k)}\Delta t \leq 0$)から、(19)により観測値データを作成する。

$$a = 0 + N(0, \sigma_a \Delta t) \quad (19)$$

一方、カルマンフィルタ運用時には、終端速度に達した時点で(20)、その後は(21)として外乱を与える。

$$u^{(k)} = -a^{(k)} \quad (20)$$

$$u^{(k)} = 0 \quad (21)$$

次に観測方程式である。シミュレーションでは位置のみを観測する。また計測誤差 σ_w があるものとする。以上により観測方程式は(22)となる。

$$y^{(k)} = C^T X^{(k-1)} + N(0, \sigma_w) \quad (22)$$

$$C = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

シミュレーションの条件

シミュレーションの条件を列挙する.

$$\Delta t = 0.05$$

$$X^{(0)} = \begin{bmatrix} x^{(0)} \\ v^{(0)} \\ a^{(0)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 9.8 \end{bmatrix}$$

$$u^{(0)} = 0$$

$$\gamma = 0$$

$$k' = 0.4$$

$$\sigma_a = 0.1$$

$$\sigma_w = 5$$

白色ノイズの作成方法: ボックスミュラー法

(一様乱数: Donald E.Knuth の減算アルゴリズム)

シミュレーションデータである観測値 $y^{(k)}$ とその真値 $x^{(k)}$ を図3, 速度の真値 $v^{(k)}$ を図4に示す. 真値には加速度の揺らぎも含まれる.

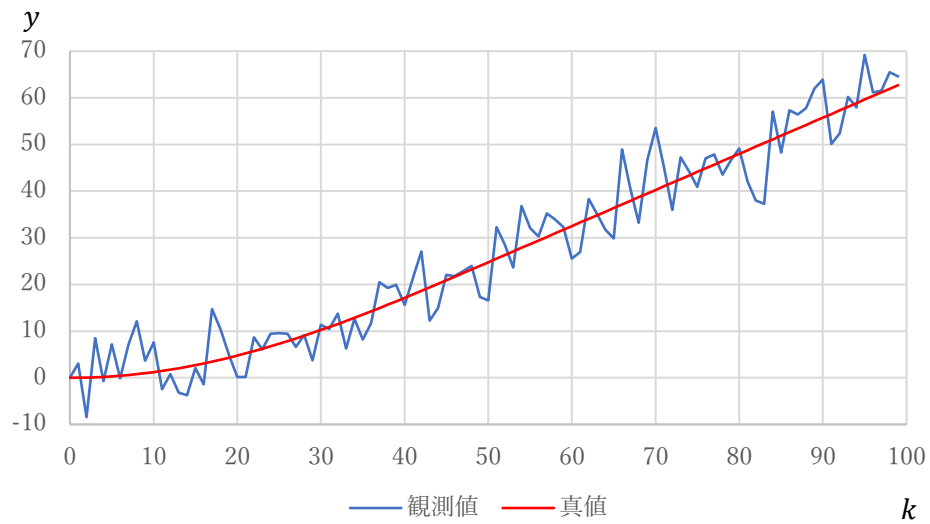


図3 観測値 $y^{(k)}$ と真値 $x^{(k)}$

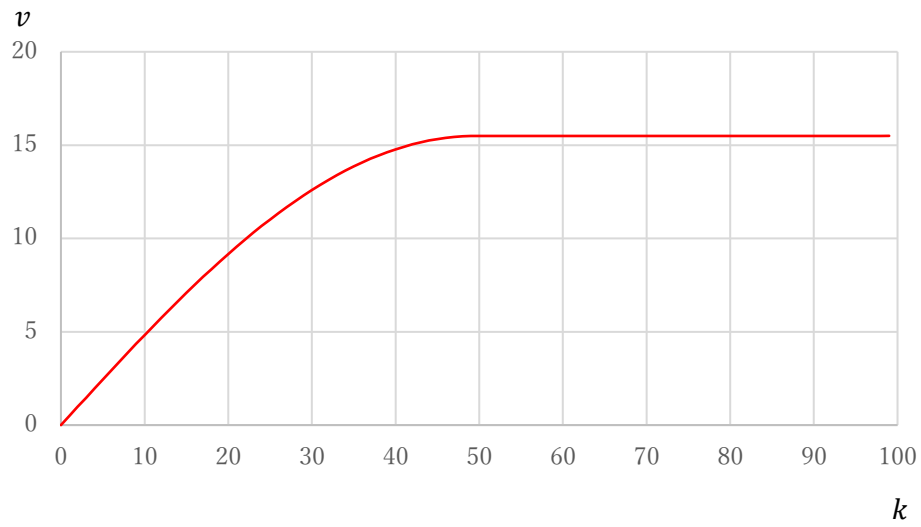


図4 速度 $v^{(k)}$ の真値

シミュレーション結果

位置の推定値を図5, 速度の推定値を図6で示す.

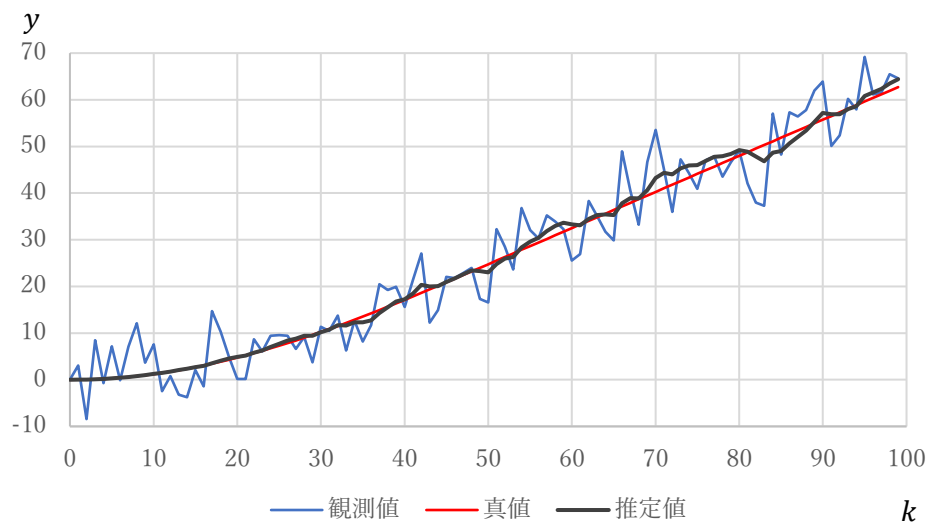


図5 位置 $x^{(k)}$ の推定値

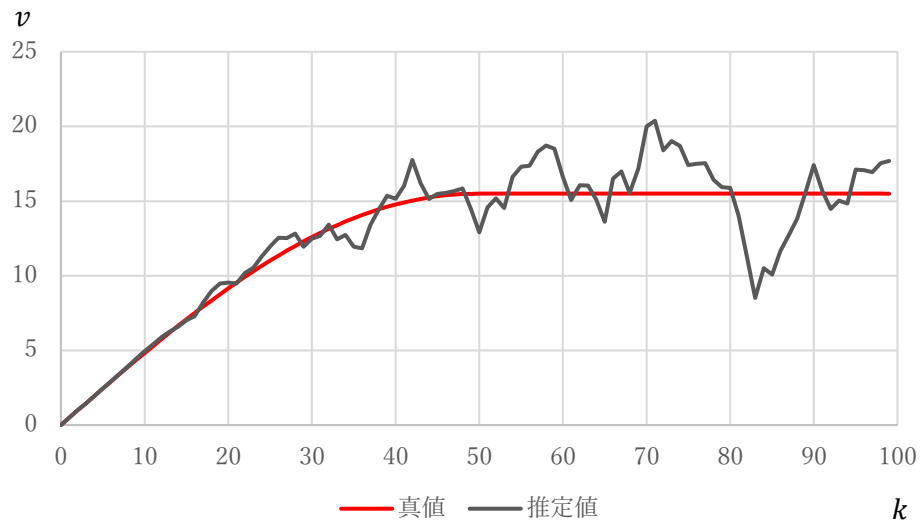


図6 速度 $v^{(k)}$ の推定値

観測誤差が大きいにも関わらず，真値に近く追従している．速度に関しては終端速度に達してからの乱高下が目立つ．

移動平均との比較

観測値の移動平均(図7)，さらに指数平滑(図8),(図9)と比較してみる

条件

移動平均の窓幅: 7

指数平滑: $x^{(k)} = ry^{(k)} + (1-r)x^{(k-1)}$ (23)

$$x^{(0)} = y^{(0)}, r = 0.5, 0.15$$

シミュレーション結果

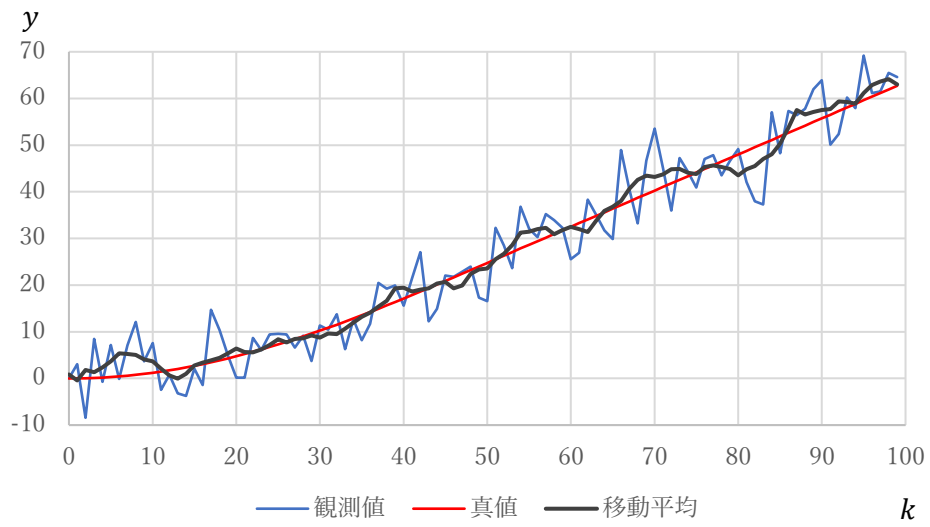


図7 移動平均

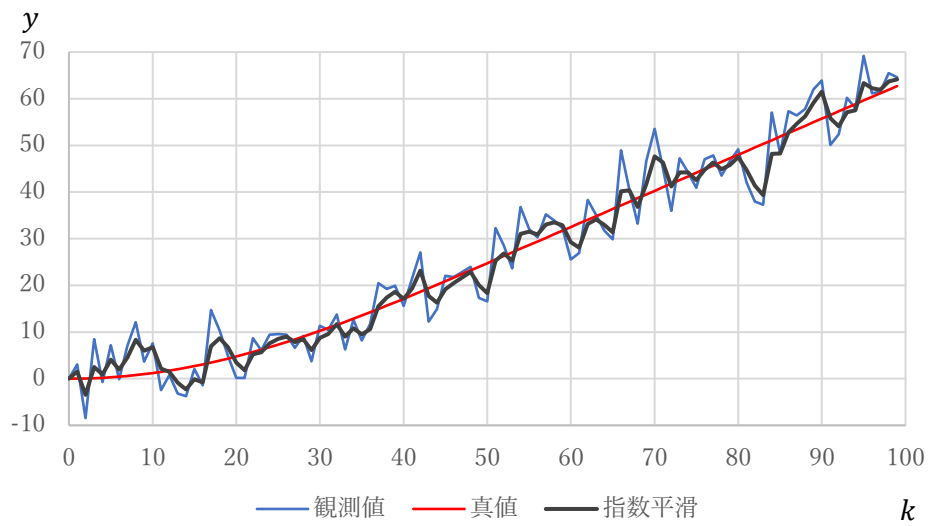


図8 指数平滑($r = 0.5$)

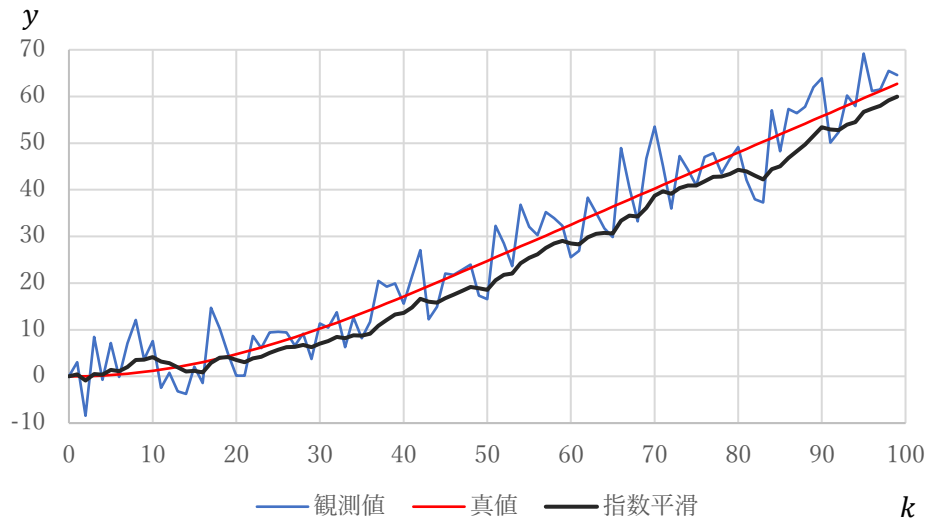


図9 指数平滑($r = 0.15$)

カルマンフィルタの精度の高さを示す結果である。

本稿の移動平均(図7)は参照点から左右に算出窓を確保している。リアルタイムシステムでは未来の情報を利用することはできず、現実には不可能な手法である。過去の情報だけに依存するとすれば、右肩あがりの曲線は数値が下方傾向となる。指数平滑では、この傾向が顕著である。指数平滑では直近のデータの重みが大きく、過去のデータほど指数関数的(過渡特性→ラプラス変換, 等比数列, z変換)に重みは減衰する。実は、カルマンゲインが一定値となる系は指数平滑そのものである[1]。これは図5と図9で、終端速度以降の曲線の形状が似ていることに注目いただきたい。状態が不変といった系では指数平滑で十分である。筆者の経験上、低感度だが高速フレームレートといった工業用カメラの撮像では、単純な指数平滑が有効である。現実にはFPGAによる実装もなされている。

カルマンフィルタのアウトライン

シミュレーションでは状態方程式は3次元(3変数)であった。カルマンフィルタのアウトラインを掴むため1次元とし、さらに外乱はないものとするれば、「予測」(5), (6)は(24), (25)となる。

$$x^{(k)} = ax^{(k-1)} \quad (24)$$

$$p^{(k)} = ap^{(k-1)} + \sigma_v^2 \quad b^2 = a^2 p^{(k-1)} + b^2 \sigma_v^2 \quad (25)$$

ここで、共分散行列 P は推定値の分散 p となっている(26)。

$$p^{(k)} = (\sigma_x^{(k)})^2 \quad (26)$$

さらにシミュレーションと同じく $b = 1$ とすれば, (25)は(27)となる.

$$\left(\sigma_x^{(k)}\right)^2 = a^2 \left(\sigma_x^{(k-1)}\right)^2 + \sigma_v^2 \quad (27)$$

当然だが, 時間更新の a (24)が分散の係数 a^2 (27)となっている.

次に「フィルタリング」である. カルマンゲイン(7)は(28)となる.

$$g^{(k)} = \frac{p^{(k)}c}{cp^{(k)}c + \sigma_w^2} = \frac{\left(\sigma_x^{(k)}\right)^2 c}{c^2 \left(\sigma_x^{(k)}\right)^2 + \sigma_w^2} \quad (28)$$

シミュレーションと同じく $c = 1$ とすれば, (28)は(29)となる.

$$g^{(k)} = \frac{\left(\sigma_x^{(k)}\right)^2}{\left(\sigma_x^{(k)}\right)^2 + \sigma_w^2} \quad (29)$$

状態推定値の更新(8)は(30), 分散の更新(9)は(31)となる.

$$x^{(k)} = x^{(k)} + g^{(k)}(y^{(k)} - x^{(k)}) = g^{(k)}y^{(k)} + (1 - g^{(k)})x^{(k)} \quad (30)$$

$y^{(k)} - x^{(k)}$: 観測値と状態推定値の誤差

$$\left(\sigma_x^{(k)}\right)^2 = (1 - g^{(k)}) \left(\sigma_x^{(k)}\right)^2 \quad (31)$$

カルマンゲインは(29)から 0~1 の範囲の数値と判る. (29)の分母は分散(誤差)の総計となっている. そして, (30)が指数平滑(23)に一致することも確認できた. また(30)よりカルマンゲインは現在の観測値の重みであり, 状態推定値の更新は推定値と観測値の間となることが判る. 他に(29)からは, 観測値のノイズが大きいほどカルマンゲインを小さく(現在の重みを小さく, 過去の重みを大きく)すべきことが判る. 一方, 推定値の分散だが, これは推定値の誤差を表しており, 観測値に誤差がなければ($\sigma_w = 0$), カルマンゲイン=1 となる. この場合, 現在の推定値をそのまま適用してよいことになる.

非線形カルマンフィルタ

非線形カルマンフィルタとは A, C を非線形関数 F, H と置き換えて定義される(32),(33).

$$X^{(k)} = F(X^{(k-1)}) + B_u u^{(k-1)} + B_v v^{(k-1)} \quad (32)$$

$$y^{(k)} = H(X^{(k-1)}) + w^{(k-1)} \quad (33)$$

カルマンフィルタに限らず、非線形関数を級数展開して高次の項を切り捨てることで線形化する、というやり方がある。つまり近似である。もう一つのやり方としては統計的サンプリング法がある[1]。この中には画像処理で用いられるパーティクルフィルタもある。本稿ではこれ以上の説明はせず、紹介にとどめる(筆者には統計的サンプリング法の経験がない)。

まとめ

カルマンフィルタの感触が伝われば幸いである。筆者が常々述べていることだが、手法を利用できるようなモデル化が肝要である。カルマンフィルタの長所として、やはりノイズ耐性と推定精度である。一方、カルマンフィルタの最大の難点は、そのモデル化にあるように思う。

参考文献

- [1] 足立修一, 丸田一郎 カルマンフィルタの基礎
東京電機大学出版局. 2012.