




## シンプルなモデルとイラストでカルマンフィルタを直感的に理解してみる


[確率・統計\(/tags/%E7%A2%BA%E7%8E%87%E3%83%BB%E7%B5%B1%E8%A8%88/\)](/tags/%E7%A2%BA%E7%8E%87%E3%83%BB%E7%B5%B1%E8%A8%88/) 9
 [統計学\(/tags/%E7%B5%B1%E8%A8%88%E5%AD%A6/\)](/tags/%E7%B5%B1%E8%A8%88%E5%AD%A6/) 320
 [statistics\(/tags/statistics/\)](/tags/statistics/) 285
 [時系列解析\(/tags/%E6%99%82%E7%B3%BB%E5%88%97%E8%A7%A3%E6%9E%90/\)](/tags/%E6%99%82%E7%B3%BB%E5%88%97%E8%A7%A3%E6%9E%90/) 165
 [KalmanFilter\(/tags/KalmanFilter/\)](/tags/KalmanFilter/) 9


いいね 184

0  
コメント

 いいね


... (/MoriKen/items/0c80ef7574997776b43/likers)

MoriKen (/MoriKen) 2017年03月01日に更新 [🔄 \(/MoriKen/items/0c80ef75749977767b43/revisions\)](https://github.com/MoriKen/items/0c80ef75749977767b43/revisions) 16

## はじめに

カルマンフィルタは逐次ベイズフィルタの一種で、かつてのアポロ計画や、現代ではカーナビ等の身近な製品でも広く活用されています。モデルが線形である（あるいは線形に近似できる）ことや、ノイズがガウス分布に従うことを仮定する必要がありますが、その仮定が許容できるシステムでは実効性が高く、様々な場面で既に実用化されています。

そんなカルマンフィルタの式や、その導出を初めてみた時、こんな風に感じました。

「なんか、よく分からんが、複雑そうだなあ」

だって、こんなんですもの...

- 状態空間表現(動作モデル, 観測モデル)

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t \\ \mathbf{z}_t &= \mathbf{C}\mathbf{x}_t + \boldsymbol{\delta}_t \end{aligned}$$

- 予測ステップ (事前推定値, 事前誤差共分散)

$$\begin{aligned}\hat{\mathbf{x}}_t^- &= A\hat{\mathbf{x}}_{t-1} + B\mathbf{u}_t \\ \hat{\mathbf{P}}_t^- &= A\hat{\mathbf{P}}_{t-1}A^T + \Sigma_x\end{aligned}$$

- フィルタリングステップ (カルマンゲイン, 事後推定値, 事後誤差共分散)

$$\begin{aligned} \mathbf{K}_t &= \hat{\mathbf{P}}_t^- \mathbf{C}^T (\mathbf{C} \hat{\mathbf{P}}_t^- \mathbf{C}^T + \Sigma_z)^{-1} \\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C} \hat{\mathbf{x}}_t^-) \\ \hat{\mathbf{P}}_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{C}) \hat{\mathbf{P}}_t^- \end{aligned}$$

当初は、大群が押し寄せてきた感じがしました。

それでもなんとか、式を追っていけばこの形になるという事実は理解できたのですが、背後で何が起きているのかよくわからないと思っていました。

そこで、超シンプルなモデルを対象に、何が起きているのかイメージできることを目標に、私の考察を記述します。最もシンプルな例で解釈を進められるように、本記事で考察の対象とする信号は全て1次元とします。

備忘録ということで、変数等を全て厳密に定義しているわけではありません。ご容赦ください。

## 状態空間と観測空間が同一の場合

つまり、観測した信号がそのまま状態空間と同じ種類の信号を提供してくれる（観測方程式について $c=1$ ）、シンプルなモデルです。パラメータは全て1次元です。ノイズは全てガウス性を有するものとします。

## 定義

- 状態空間表現(動作モデル, 観測モデル)

$$x_t = ax_{t-1} + bu_t + e_t$$

$$z_t = x_t + \delta_t$$

- 予測ステップ（事前推定値，事前誤差共分散）

$$\hat{x}_t^- = a\hat{x}_{t-1} + bu_t$$

$$\sigma_{\hat{x}_t^-}^2 = a^2\sigma_{\hat{x}_{t-1}}^2 + b^2\sigma_x^2$$

- フィルタリングステップ（カルマンゲイン，事後推定値，事後誤差共分散）

$$k_t = \frac{\sigma_{\hat{x}_t^-}^2}{\sigma_{\hat{x}_t^-}^2 + \sigma_z^2}$$

$$\hat{x}_t = \hat{x}_t^- + k_t(z_t - \hat{x}_t^-)$$

$$\sigma_{\hat{x}_t}^2 = (1 - k_t)\sigma_{\hat{x}_t^-}^2$$

実は，誤差分散の文字の添字を上述のように表現することが一般的なのかまでは調査出来ていません。しかし，以後の説明で分散が重要になるので，全体として統一した表現になるようにするために，今回はこのような表現にしました。

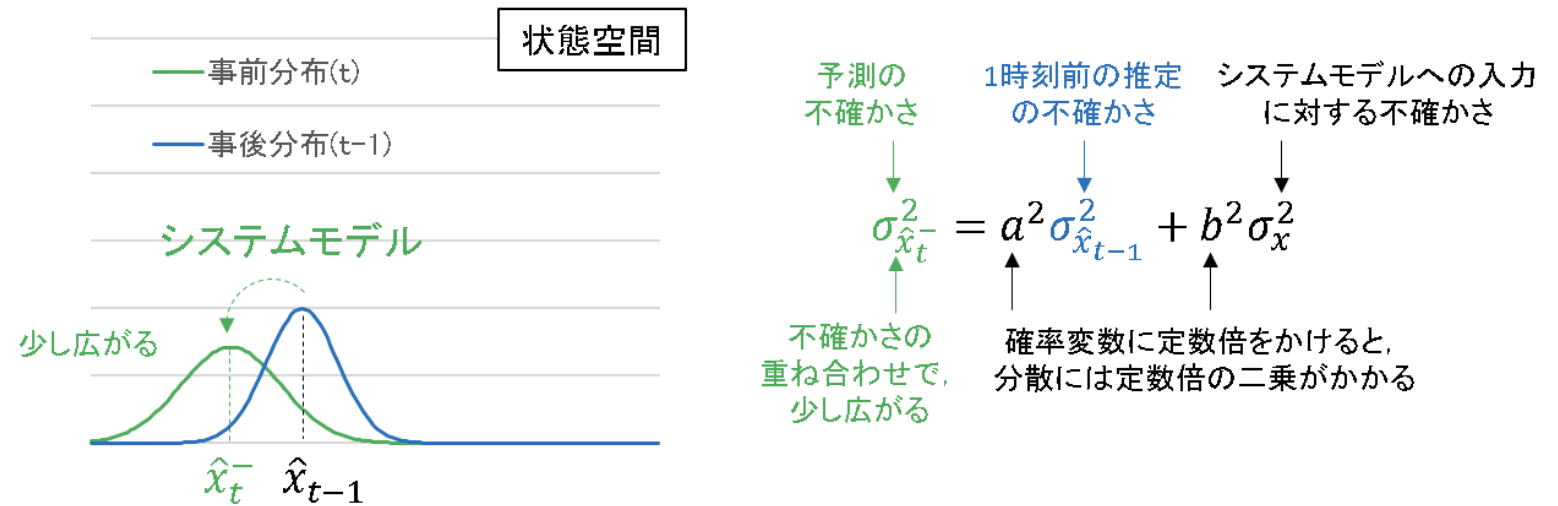
## 予測ステップについて

予測ステップでは，一時刻前の事後推定値からシステムモデルを元に事前推定値を算出します。これはただシステム方程式に代入すればいいだけです。

分散のところが少しごちゃついています，結論としては，

- 事前誤差分散 = 1時刻前の事後誤差分散 + システムモデルのノイズの分散

となります。



(<https://camo.qiitusercontent.com/5aca3dcac230877eb4c6b76e3a320bc005618c7c/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f36643037366164352d656466612d636361612d336264322d3837383238303031373561622e706e67>)

システムモデルで考えられうる不確かさを全て重ねあわせたものが，事前誤差分散となっていることが分かります。普通は，予測ステップで分散が広がります。システムへ入力が加えられると不確かさが増しそうな予感します。

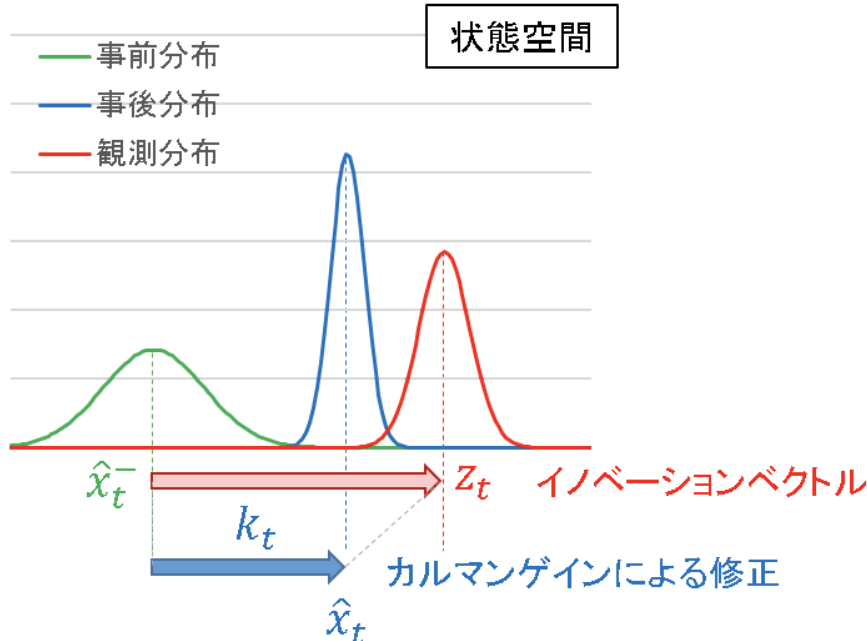
例えば，ロボットに速度指令を入力しても，ピッタリその通りに動くことはなく，必ずずれが生じます。

## フィルタリングステップについて

予測ステップで得られた事前推定値に対して，観測情報を元に修正を加えます。

### 推定値

事前推定値と観測値の差分，すなわち観測予測誤差をイノベーションベクトルと定義し，これにカルマンゲインをかけることで修正量を決定します。



$$\hat{x}_t = \hat{x}_t^- + k_t(z_t - \hat{x}_t^-)$$

修正ゲイン  $k_t$     イノベーション

$$= \hat{x}_t^- + \frac{\sigma_{\hat{x}_t^-}^2}{\sigma_{\hat{x}_t^-}^2 + \sigma_z^2} (z_t - \hat{x}_t^-)$$

こいつは何者か？

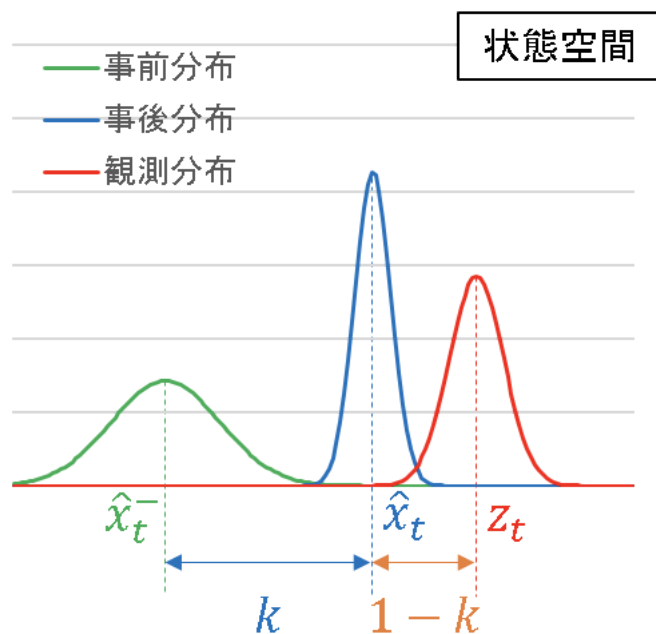
(<https://camo.qiitusercontent.com/bcecb2401c4606d13477e6d5415c7d92fa429d2f/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f30613130313937342d656237622d646665332d336563312d3730303364373737303230642e706e67>)

ここで気になるのは、カルマンゲイン

$$k_t = \frac{\sigma_{\hat{x}_t^-}^2}{\sigma_{\hat{x}_t^-}^2 + \sigma_z^2}$$

の意味です。分散の比を取っているようですが、ただ眺めているだけではイメージが湧きません。そこで、次のように考えてみます。

事後推定値は、事前推定値と観測値の内分点であり、内分比は  $k:1-k$  ( $0 < k < 1$ ) であるとし、内訳は、下図を参照ください。



高校数学の内分点の公式

$$\hat{x}_t = \frac{(1-k)\hat{x}_t^-}{k + (1-k)} + \frac{kz_t}{k + (1-k)}$$

$$= (1-k)\hat{x}_t^- + kz_t$$

$$= \hat{x}_t^- + k(z_t - \hat{x}_t^-)$$

観測更新の式と完全一致

$$= \hat{x}_t^- + \frac{\sigma_{\hat{x}_t^-}^2}{\sigma_{\hat{x}_t^-}^2 + \sigma_z^2} (z_t - \hat{x}_t^-)$$

予測の不確かさ    観測の不確かさ

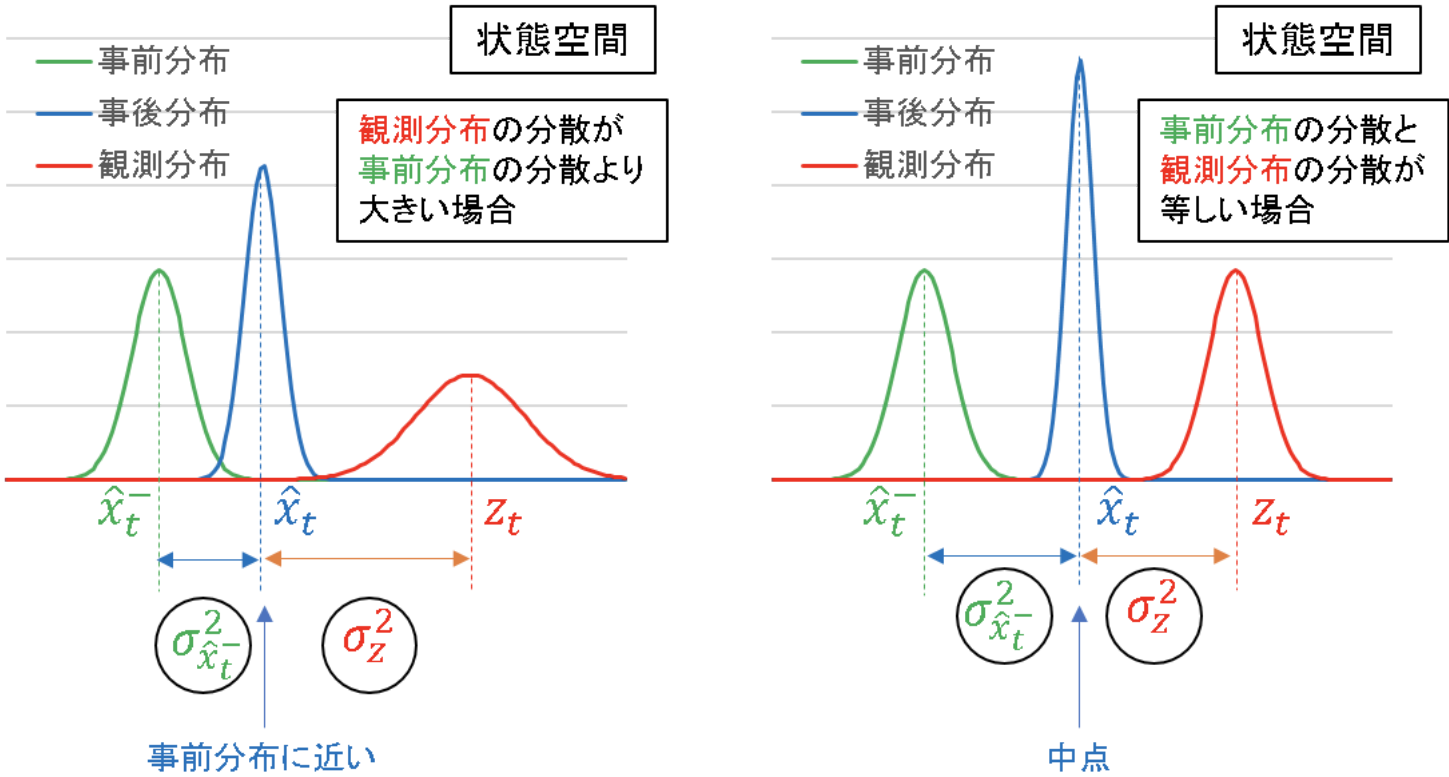
(<https://camo.qiitusercontent.com/7b88b950ba4a1d1abdaf4fd1ea3424e1ffd7c08a/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f65363061316530612d346365352d6465356332d356238612d6262393465396363633930302e706e67>)

あとは高校数学で習った内分点の公式に代入すると、その計算結果がカルマンフィルタの事後推定値の更新式と一致することが分かります。

これにより、カルマンゲイン  $k$  の正体は事前推定値と観測値の何らかの指標に基づいた内分比であることが確認できました。そして、その内分比を決めるのが、それぞれの分散の大小関係であることも併せて理解できます。

例えば先に示した図では、事前分布の分散が大きい（観測分布の分散が小さい）ので、修正量が大きくなり、観測に近い事後推定値を算出します。この観測の分散をさらに小さくすると、事後推定値はより観測値側になることになります。

一方、観測の分散の方が大きいとしたら、事後推定値は事前推定値側に引っ張られることになります。また、事前分布と観測分布の分散が一致するとしたら、事後推定値はちょうどそれらの中点を指すことになります。



(<https://camo.qiitusercontent.com/127e5d11a61ccf133d3950a5bf54897628be14e0/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f32613839316439662d623934642d393533612d316231302d6362636364656432663730652e706e67>)

敢えて例えるならば、事後推定値を求める際に、事前分布と観測分布は互いに綱引きをしており、その引きの強さが分散の逆数というイメージです。より分散の小さい方（すなわちより情報が確からしい方）が強く引っ張る力を持つということは、直感的にも正しい特性と言えます。

分散

残るは、事後誤差分散の更新です。

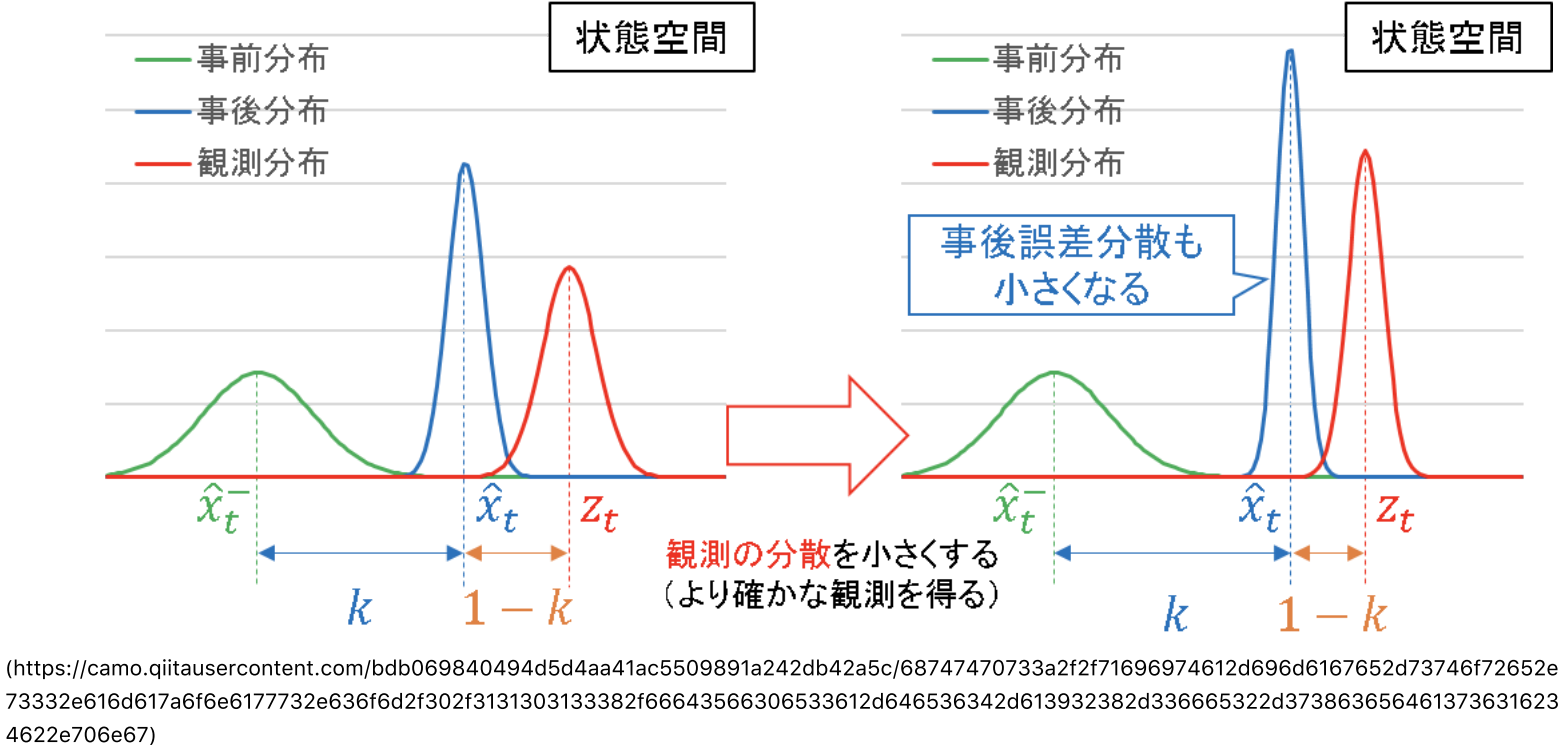
$$\sigma_{\hat{x}_t}^2 = (1 - k) \sigma_{\hat{x}_t^-}^2$$

観測の分散が小さい(より正確な情報を得た)ときほど、  
修正ゲインが大きくなる  
↓  
修正ゲインが大きいほど、  
事後分布の不確かさが小さくなる

(<https://camo.qiitusercontent.com/6a950277ba29584522b3c1f53876f07ee50f8e5f/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f32343962316539662d663063362d353731612d383835312d3561656436373634633635372e706e67>)

事前誤差分散に 1-k をかけることで、事後誤差分散を算出します。つまり、事後誤差分散は必ず事前誤差分散よりも小さくなります。これまで示した図中でも、それは確認できます。直感的には、推定に必要な情報が複数得られたので、より自信をもって答えることができる、といったイメージです。

更にいうと、追加で得られる観測情報が、事前の予測情報よりも確かであるほど分散が小さくなります。その度合を示すのが 1-k であり、これは事前分布と観測分布の分散同士の競合関係を表すので、直感的な解釈とも一致します。



## 状態空間と観測空間が異なる場合

次に、状態空間と観測空間が異なる場合を扱います。つまり、状態空間と観測空間の間に写像が存在する場合です。一応、簡単のためこれらの空間の写像が全単射であることを仮定します。（ここでは1次元なので、 $c$ が0でも $\infty$ でもなければOK。）

## 定義

- 状態空間表現（事前推定値，事前誤差共分散）

$$\begin{aligned}x_t &= ax_{t-1} + u_t + \epsilon_t \\z_t &= cx_t + \delta_t\end{aligned}$$

- 動作予測（事前推定値，事前誤差共分散）

$$\begin{aligned}\hat{x}_t^- &= a\hat{x}_{t-1} + bu_t \\ \sigma_{\hat{x}_t^-}^2 &= a^2\sigma_{\hat{x}_{t-1}}^2 + b^2\sigma_x^2\end{aligned}$$

- 観測更新（カルマンゲイン，事後推定値，事後誤差共分散）

$$\begin{aligned}k_t &= \frac{c\sigma_{\hat{x}_t^-}^2}{c^2\sigma_{\hat{x}_t^-}^2 + \sigma_z^2} \\ \hat{x}_t &= \hat{x}_t^- + k_t(z_t - c\hat{x}_t^-) \\ \sigma_{\hat{x}_t}^2 &= (1 - k_t c)\sigma_{\hat{x}_t^-}^2\end{aligned}$$

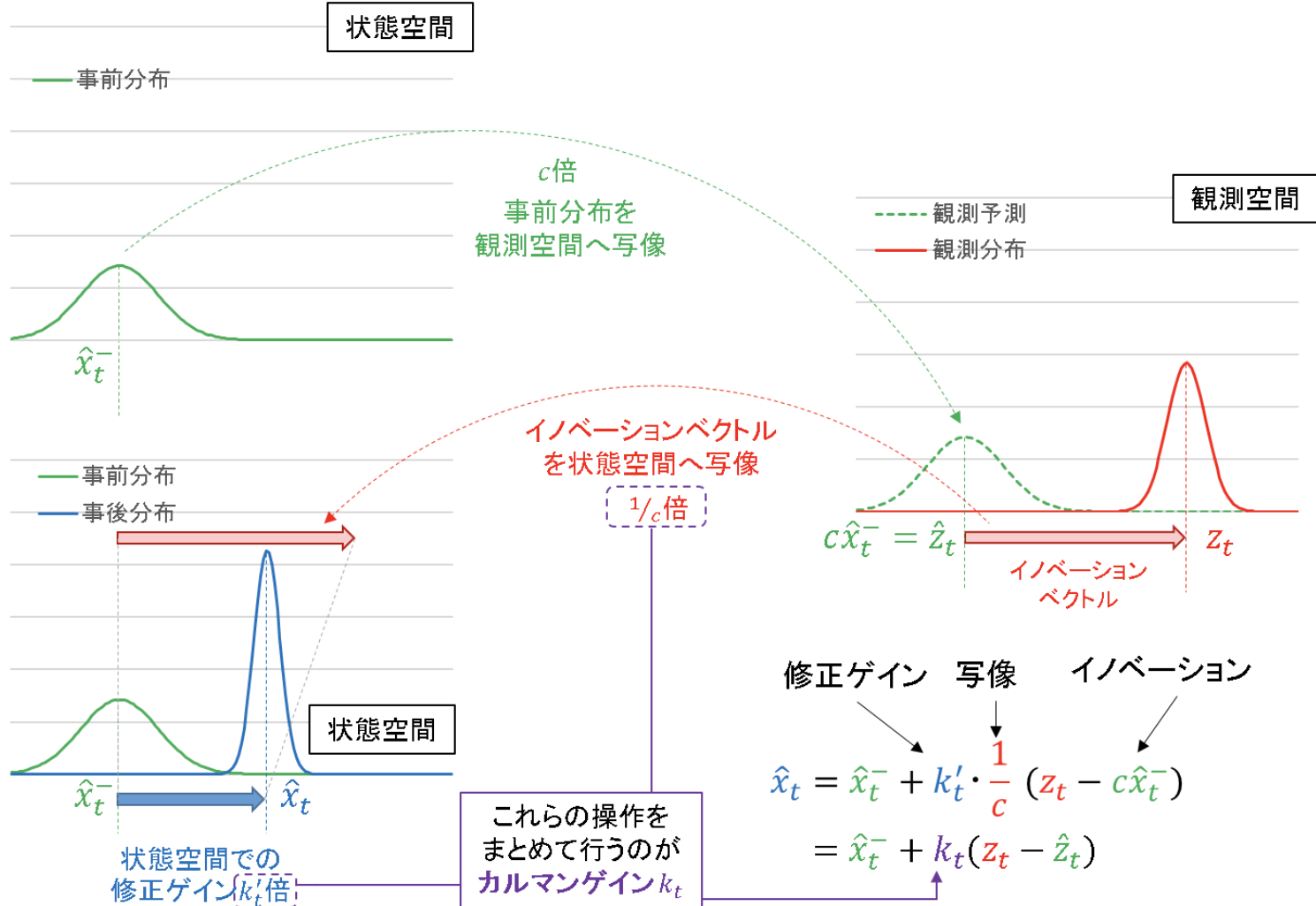
## 予測ステップ

「状態空間と観測空間が同一の場合」と変わりませんので、省略します。

## フィルタリングステップ

### 推定値

「状態空間と観測空間が同一の場合」では状態空間に関する一つのグラフで完結しましたが、ここでは観測空間に関するもう一つのグラフを用意してあげる必要があります。下図がそのイメージです。



(<https://camo.qiitusercontent.com/4a5940ad4d435dad3d682e837bb1e5f70af6c02/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f37663038396666662d313637392d663934632d303835642d3463326538343839343965352e706e67>)

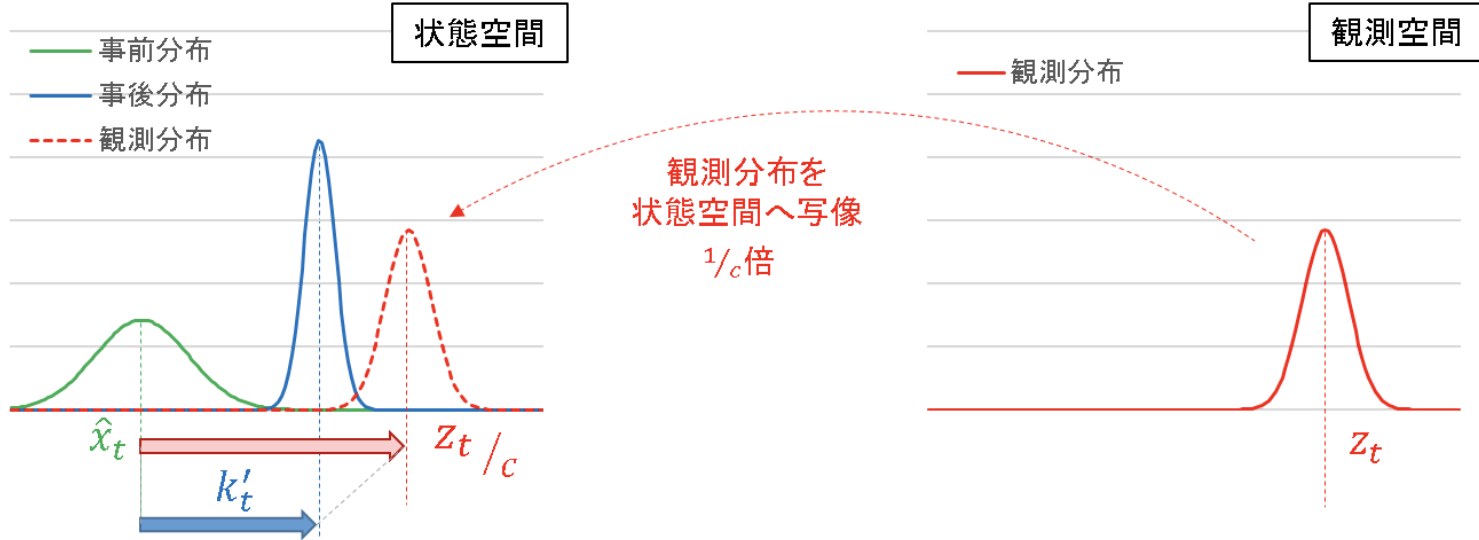
先の場合と比べると、状態空間と観測空間を行き来している様子を確認できます。

1. まず、事前分布を観測空間に写像します。（このとき写像によっては中心も分散も変わりますが、説明のため同じ形のグラフを使っています。）
2. 1の写像を元に、観測空間内でイノベーションベクトル（観測予測誤差）を求めます。
3. 2で求めたイノベーションベクトルを状態空間へ写像します。
4. 3で写像されたイノベーションベクトルに、状態空間における修正ゲインをかけたものを、フィルタリングの修正量とする。

※今扱っている信号は一次元なので、イノベーション「ベクトル」と言うのは本当はまずいかもしれません...

短く言うと、フィルタリングステップとは、「観測空間のイノベーション」を「状態空間へ写像」したものに「修正ゲイン」をかけたもの、となります。このうち、「観測空間へ写像」と「修正ゲイン」をかける役割を同時に担うのが、カルマンゲインの役割と解釈できます。

ここまでの記述でも性質は理解できるのですが、状態空間と観測空間を行き来するので、少しごちゃついたイメージは否めません。そこで、別の見方をしてみます。先に観測空間の信号を状態空間に写像してしまうという発想です。

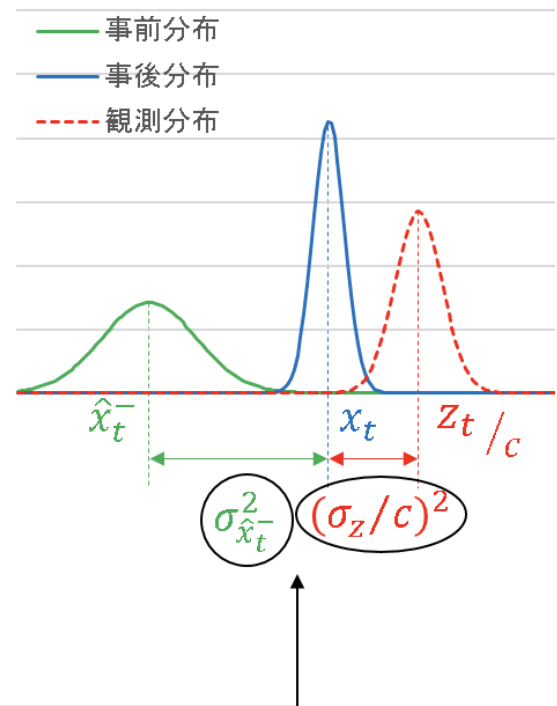


(<https://camo.qiitusercontent.com/93a7269d3cfc2f2b4286bb33cc424fa320e4398b/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f39383066363236642d303630362d323336312d386138632d6632393335633539303865382e706e67>)

こう考えると、以降は「状態空間と観測空間が同一の場合」と同様の流れで、シンプルに考えることができます。あとは、こんな考え方に変えても式が成り立つのか、というところです。

下図のように分解していくと、修正量が「状態空間での修正ゲイン」×「状態空間での観測誤差」の形式に変形できることが確認できました。

$$\begin{aligned}
 \hat{x}_t &= \hat{x}_t^- + k_t(z_t - \hat{z}_t) \\
 &\quad \text{ただの代入} \\
 &= \hat{x}_t^- + \frac{c\sigma_{\hat{x}_t^-}^2}{c^2\sigma_{\hat{x}_t^-}^2 + \sigma_z^2}(z_t - c\hat{x}_t^-) \\
 &\quad \text{役割を意識して分解} \\
 &= \hat{x}_t^- + \frac{\sigma_{\hat{x}_t^-}^2}{\sigma_{\hat{x}_t^-}^2 + (\sigma_z/c)^2} \cdot \frac{1}{c} \cdot (z_t - c\hat{x}_t^-) \\
 &\quad \text{写像} \quad \text{イノベーション} \\
 &= \hat{x}_t^- + \frac{\sigma_{\hat{x}_t^-}^2}{\sigma_{\hat{x}_t^-}^2 + (\sigma_z/c)^2} \left( \frac{z_t}{c} - \hat{x}_t^- \right) \\
 &\quad \text{状態空間での修正ゲイン } k' \quad \text{事前分布と観測分布の分散の比}
 \end{aligned}$$



(<https://camo.qiitusercontent.com/ee16882c61a00e03c09513afb8d1c6c668ff88fd/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f39323431646361372d626334652d373864362d616333372d3065353932646236636461362e706e67>)

そして、修正ゲイン&ltmath>k'</math>はやはり「分散の比」となります。観測の分散が&ltmath>c^2</math>で割られていますが、これは写像の結果を反映しているので、直感に反しない結果と言えます。

## 分散

残るは分散です。これも先の例と全く同様に議論できます。



状態空間での  
修正ゲイン  $k'$



$$\sigma_{\hat{x}_t}^2 = (1 - kc) \sigma_{\hat{x}_t^-}^2$$

観測の分散が小さい(より正確な情報を得た)ときほど、  
修正ゲインが大きくなる



修正ゲインが大きいほど、  
事後分布の不確かさが小さくなる

(<https://camo.qiitusercontent.com/e431506e63cae316a410ca1e1fa9ede620c2361a/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f33326332336530622d656535362d663437372d636330312d6230643563306535306531342e706e67>)

ただし、cがかかっている点だけが異なりますが、これは状態空間という土俵に立たせるために必要な操作です。先ほど、

- 「カルマンゲイン」 = 「状態空間での修正ゲイン」 × 「状態空間への写像(1/c)」

という関係を考えましたが、これを

- 「状態空間での修正ゲイン」 = 「カルマンゲイン」 ÷ 「状態空間への写像(1/c)」
- 「状態空間での修正ゲイン」 = 「カルマンゲイン」 × 「観測空間への写像(c)」

に変形したものと考えられます。カルマンゲインには観測空間の値がかけられることになっているので、その整合性が合うような操作となっています。

## 多次元空間との類推

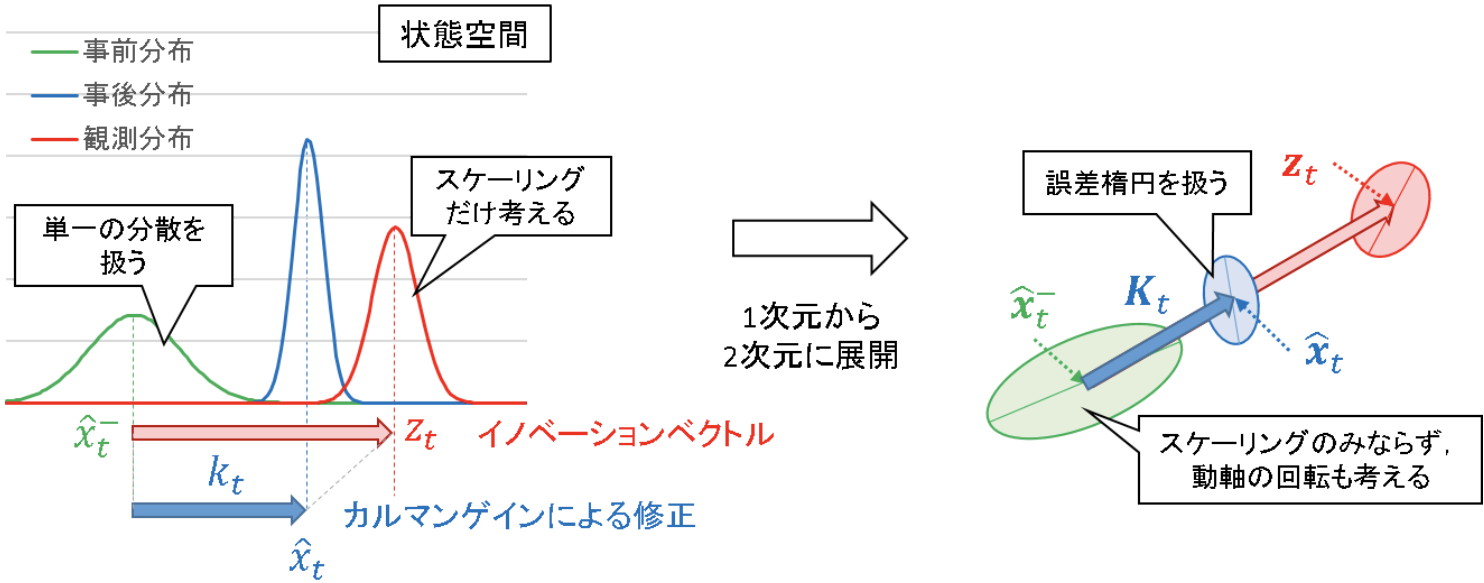
以上の解釈を元に、多次元のカルマンフィルタも同様に類推させることができます。下の表を眺めて左右の対応関係を比較すると、スカラの計算がベクトルの計算になっているだけ、ということが分かります。

		一次元(スカラ)	多次元(ベクトル)
予測	推定値	$\hat{x}_t^- = a\hat{x}_{t-1} + bu_t$	$\hat{\mathbf{x}}_t^- = A\hat{\mathbf{x}}_{t-1} + B\mathbf{u}_t$
	分散	$\sigma_{\hat{x}_t^-}^2 = a^2 \sigma_{\hat{x}_{t-1}}^2 + b^2 \sigma_x^2$ <p>予測の不確かさ ↓ 1時刻前の推定 の不確かさ ↓ システムモデルへの入力 に対する不確かさ ↓</p> <p>不確かさの 重ね合わせで、 少し広がる ↑ 確率変数に定数倍をかけると、 分散には定数倍の二乗がかかる ↑</p>	$\hat{\mathbf{P}}_t^- = A\hat{\mathbf{P}}_{t-1}A^T + B\boldsymbol{\Sigma}_x B^T$
フィルタリング	カルマン ゲイン	$k_t = \frac{c\sigma_{\hat{x}_t^-}^2}{c^2\sigma_{\hat{x}_t^-}^2 + \sigma_z^2}$ <p>予測の不確かさ ↑ 観測の不確かさ ↓</p>	$\mathbf{K}_t = \hat{\mathbf{P}}_t^- \mathbf{C}^T (\mathbf{C}\hat{\mathbf{P}}_t^- \mathbf{C}^T + \boldsymbol{\Sigma}_z)^{-1}$
	推定値	$\hat{x}_t = \hat{x}_t^- + k_t(z_t - c\hat{x}_t^-)$ <p>修正ゲイン ↑ 写像 ↑ イノベーション ↓</p>	$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{C}\hat{\mathbf{x}}_t^-)$
	分散	$\sigma_{\hat{x}_t}^2 = (1 - k_t c) \sigma_{\hat{x}_t^-}^2$ <p>状態空間での 修正ゲイン <math>k'_t</math> ↓</p>	$\hat{\mathbf{P}}_{t-1} = (\mathbf{I} - \mathbf{K}_t \mathbf{C}) \hat{\mathbf{P}}_t^-$

(<https://camo.qiitusercontent.com/ae6982e88bbc909427c9d072c73d382c567d6584/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f32353866323762612d336433312d396365662d363632612d646566613066373934376>)



2次元に展開した場合のイメージを例にすると、下記ようになります。



(<https://camo.qiitousercontent.com/6a6a1cf029652ebf9d5fdf4045d7bc8fd74efae9/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3131303133382f36643631626639312d373134642d313339372d646261312d6134373631626633623666392e706e67>)

1次元では、分散というスカラー値を扱えばよかったので、それがどれくらい大きくなるか、小さくなるか、というスケーリングだけ考えればことが足りていました。

2次元の場合、誤差の生じる方向が2つになるので、誤差楕円を考える必要があります。かつ、その楕円の動軸がどの方向を向くのかも、推定特性に大きく影響を与えます。この性質を評価するために、1次元でいうところの「分散」が「分散共分散行列」(<https://ja.wikipedia.org/wiki/%E5%88%86%E6%95%A3%E5%85%B1%E5%88%86%E6%95%A3%E8%A1%8C%E5%88%97>)」に展開されていきます。

3次元の場合も同様の発想で、扱う図形が楕円体になるというイメージです。

それ以上の次元は幾何学的なイメージは困難ですが、本質的には同様のフローで展開されていきます。

## おわりに

以上が、今回メモとして残しておきたかったことです。いくつかの参照資料を読みながら自分なりに考えることで、「式展開したらこうなった」から、何となくでも「背景にはこういう意味がある」という段階まで落としこむことができました。

数学を専門にしているわけでもない、しがなき素人学生の備忘録です。誤りもあると思いますので、ご容赦願います。その際は、厚かましくはございますが、ご指摘を頂けましたら幸いです。

## 参考文献

1. S Thrun, et al., "Probabilistic Robotics", the MIT Press, 2005. (<http://www.probabilistic-robotics.org/>)
2. 足立ら, "カルマンフィルタの基礎", 東京電機大学出版局, 2012. (<http://www.tdupress.jp/books/isbn978-4-501-32890-0.html>)