Bias in a Kalman Filter

Rob Kutschke

November 24, 2007

Abstract

This note uses a small toy problem, which can be computed exactly, to illustrate the bias of the final result of kalman filter towards its starting value. It also illustrates the numerical precision problem that occurs.

1 The Toy Problem and Its Exact Solution

Given, N measurements and their errors, (x_i, σ_i) for $i = 1 \ldots N$, compute the mean and the error on the mean, $(\bar{x}, \sigma)$. The well known exact solution is,

$$\frac{1}{\sigma^2} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \tag{1}$$

$$\frac{\bar{x}}{\sigma^2} = \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} \tag{2}$$

2 The Kalman Filter Equations

The Kalman filter is an iterative process that accumulates a state vector η and its covariance matrix V, by adding one measurement at a time. The kalman filter equations to add a one dimensional measurment are:

$$\begin{aligned} V_n &= V_{n-1} - \frac{V_{n-1} D D^T V_{n-1}}{\sigma_n^2 + D^T V_{n-1} D} \\ \eta_n &= \eta_{n-1} + V_n D \frac{d_n - d_n(\eta_{n-1})}{\sigma_n^2}, \end{aligned} \tag{3}$$

where η_n is the track parameter vector using information from points $1 \ldots n$; V_n is the covariance matrix of η, using information from points $1 \ldots n$; d_n is the measured quantity at the n^{th} point; $d_n(\eta_{n-1})$ is the predicted value of the measured quantity at point n, using only information from points $1 \ldots (n-1)$; and where the vector D is defined by,

$$D_j = \left. \frac{\partial d_n}{\partial \eta_j} \right|_{\eta = \eta_{n-1}}. \tag{4}$$

Finally, the quantities with a subscript of $n-1$ have the same information as their counterparts with a subscript n but only use information from points $1 \ldots (n-1)$.

In a standard track fitting problem η is a column 5-vector, V is a 5x5 symmetric matrix and D is a column 5-vector. By the statement of the problem d_n and $d_n(\eta_{n-1})$ are scalars.

3 Doing The Toy Problem with a Kalman Filter

For the toy problem the state vector has length of 1 and the above equations simplify considerably. Here I will introduce a slightly different notation:

- $\bar{x}_n$ the kalman filter estimate of x, using information from points 1...n. This corresponds to η_n in the above equations.
- s_n^2 the kalman filter estimate of the error-squared on x, using information from points 1...n. This corresponds to V_n in the above equations.
- D is trivially 1.

With this notation, the kalman filter for the toy problem become,

$$s_n^2 = s_{n-1}^2 - \frac{s_{n-1}^4}{\sigma_n^2 + s_{n-1}^2} \tag{5}$$

$$\bar{x}_n = \bar{x}_{n-1} + s_n^2 \frac{x_n - \bar{x}_{n-1}}{\sigma_n^2} \tag{6}$$

The equation for s_n can be simplfied,

$$\frac{1}{s_n^2} = \frac{1}{s_{n-1}^2 - \frac{s_{n-1}^4}{\sigma_n^2 + s_{n-1}^2}} \tag{7}$$

$$= \frac{\sigma_n^2 + s_{n-1}^2}{s_{n-1}^2\left(\sigma_n^2 + s_{n-1}^2\right) - s_{n-1}^4} \tag{8}$$

$$= \frac{\sigma_n^2 + s_{n-1}^2}{s_{n-1}^2\, \sigma_n^2} \tag{9}$$

$$= \frac{1}{s_{n-1}^2} + \frac{1}{\sigma_n^2}. \tag{10}$$

One can set $s_1^2 = \sigma_1^2$ and start the recursion at $n = 2$ to obtain:

$$\frac{1}{s_n^2} = \sum_{i=0}^{n} \frac{1}{\sigma_i^2} \tag{11}$$

which reproduces Equation 1; and we identify $s_N = \sigma$.

The equation for $\bar{x}_n$ also simplfies,

$$\frac{\bar{x}_n}{s_n^2} = \frac{\bar{x}_{n-1}}{s_n^2} + \frac{x_n - \bar{x}_{n-1}}{\sigma_n^2} \tag{12}$$

$$= \bar{x}_{n-1}\left(\frac{1}{s_n^2} - \frac{1}{\sigma_n^2}\right) + \frac{x_n}{\sigma_n^2} \tag{13}$$

$$= \frac{\bar{x}_{n-1}}{s_{n-1}^2} + \frac{x_n}{\sigma_n^2} \tag{14}$$

One can set $(\bar{x}_1, s_1) = (x_1, \sigma_1)$ and start the recursion at $n = 2$:

$$\frac{\bar{x}_n}{s_n^2} = \sum_{i=0}^{n} \frac{x_i}{\sigma_i^2} \tag{15}$$

which reproduces Equation 2; and we identify $\bar{x}_N = \bar{x}$ and $s_N = \sigma$.

4 Starting the Kalman Filter

So far this toy problem is missing one critical feature of a real tracking problem. A track fit begins with a set of starting track parameters η_0 and a starting covariance matrix $V_0 = \mathrm{diag}(\infty)$; as a practical matter the starting value of the diagonal elements is some large number. How large will be discussed below. The Kalman filter then adds the measurements, one at a time, starting from the first measurement.

The reason that starting values are required is that the first measurement does not fully determine the track parameter vector. Indeed the first several measurements may over-determine a subset of the track parameters yet leave other track parameters undetermined. In the toy problem, however, the first measurement fully determines the state vector so the artifice of the starting parameters is not required.

Never-the-less most of the critical features of the starting parameters can be added to the toy problem, albeit in a rather artificial way. Consider a starting value of x_0 with an error of σ_0, where $\sigma_0 >> \sigma$. With this addition one can set $\bar{x}_0 = x_0$ and $s_0^2 = \sigma_0^2$ and then start the kalman filter recursion at n=1. This has the additional feature that the equal treatment of all N measurements is manifest.

With the addition of the starting parameters, the final results of the Kalman filter are,

$$\frac{1}{s_N^2} = \sum_{i=0}^{N} \frac{1}{\sigma_i^2} \tag{16}$$

$$\frac{\bar{x}_N}{s_N^2} = \sum_{i=0}^{N} \frac{x_i}{\sigma_i^2} \tag{17}$$

These expressions differ from Equations 1 and 2 in that the sum starts at 0, not at 1. Therefore,

$$\frac{1}{s_N^2} = \frac{1}{\sigma^2} + \frac{1}{\sigma_0^2} \tag{18}$$

$$\frac{\bar{x}_N}{s_N^2} = \frac{\bar{x}}{\sigma^2} + \frac{x_0}{\sigma_0^2} \tag{19}$$

5 Computing the Bias

In the limit that $\sigma_0 >> \sigma$, the expression for s simplifies to,

$$s_N^2 = \frac{\sigma^2}{1 + \frac{\sigma^2}{\sigma_0^2}} \tag{20}$$

$$\approx \sigma^2 \left(1 - \frac{\sigma^2}{\sigma_0^2}\right), \tag{21}$$

plus terms of order $(\sigma/\sigma_0)^4$ and higher. And the expression for x_K simplifies to,

$$\bar{x}_N = \sigma^2 \left(1 - \frac{\sigma^2}{\sigma_0^2}\right) \left(\frac{\bar{x}}{\sigma^2} + \frac{x_0}{\sigma_0^2}\right) \tag{22}$$

$$= \left(1 - \frac{\sigma^2}{\sigma_0^2}\right) \left(\bar{x} + x_0 \frac{\sigma^2}{\sigma_0^2}\right) \tag{23}$$

$$= \bar{x} - (\bar{x} - x_0) \frac{\sigma^2}{\sigma_0^2}, \tag{24}$$

plus higher order terms.

In the above, the only approximation is that $\sigma_0 >> \sigma$. Provided this is true, the result holds independent of the starting value, x_0! Therefore, provided σ_0 is sufficiently large, the final result has a negligibly small bias towards the starting value.

In a realistic tracking problem, the initial value of any track parameter is likely to be within no worse than 10 σ of the final answer. So, for $\sigma_0 \approx 30\ \sigma$ or larger, the bias of the final answer away from the true answer will be no more than about 0.01 σ and will often be much smaller than this.

Another consideration arises in realistic tracking problems. The propagation of the track parameters and their error matrix from one measurement point to the next is non-linear in the track parameters. This non-linearity can provide additional constraints that the starting values be close enough to the true values that the derivatives D are "good enough".

6 Note on Numerical Precision

The numerical precision problem arises when computing s_1 from the starting value of the error, σ_0, and the measurement error on the first data point,

$$s_1^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_1^2 + \sigma_0^2} \tag{25}$$

It was shown earlier that this simplfies to $s_1 = \sigma_1$ plus terms of order σ_1^2/σ_0^2. The problem is that kalman filter code will actually evaluate this using a general purpose subroutine that works for all steps in the iteration; an example of what such code might look like shown in Table 1.

```
double updateError( double Ssq, double Sigsq ){
     double denom = Sigsq + Ssq;
     double ratio = Ssq*Ssq/denom;
     double Ssqnew = Ssq - ratio;
     return Ssqnew ;
}
```

Table 1: Code fragment used to evaluate Equation 25.

The problem can be even more accute in a real tracking problem when the state vector has a length of 5. In such a case, the numerical precision problem remains until enough hits have been added to add information to all track parameters. A particularly bad problem occurs when a subset of the track parameters are very well determined but one or two remain undetermined; in this case V contains some matrix elements that are very large, of order σ_0^2, and others that are very small.

There are several possible ways to attack this problem. One option is to identify the problem cases and branch to special code to handle the case. This can be done but it has the potential downside of an ever increasing list of special cases.

A second option might be to keep V^{-1} instead of V as part of the state information. As is seen in the toy problem, updating V^{-1} is particularly simple to update and does not suffer from numerical precision problems: if $1/\sigma_0^2$ underflows to zero that is just fine. The full matrix form of the equations are also numerically more robust when expressed in V^{-1}. I understand that some kalman filter code does do this; in this method, V^{-1} is refered to as the information matrix.

However TRF is not well suited to this solution. In TRF, each hit is described in a basis that is natural for that hit; the side effect is that one must perform a basis transformation on (η, V) between hits. If the state information is changed to (η, V^{-1}) then new basis transformation code needs to be written. If there are N types of bases in TRF, there must be N^2 transformation functions, each of which would require of order a week to write. This makes such a change daunting. There is another possible complication that I am not sure of: if V can be propagated with standard gaussian error propagation, I am concerned that propagation of V^{-1} can grow tails. My only basis for this belief is that, in standard tracking problems, if $1/p_T$ has gaussian errors then p_T often has non-gaussian errors.

7 Summary

This note has shown that a Kalman filter has a bias away from the true answer and towards the starting value of $(\bar{x} - x_0)\sigma^2/\sigma_0^2$, where $\bar{x}$ is the final value of the parameter, x_0 is its starting value, σ is the final error on the parameter and σ_0 is the starting value for the error on the parameter. This argues for a large value of σ_0.

This note has also shown that if σ_0 is too large, then there is a numerical precision problem at the first iteration.

Code for a realistic problem may have a small window of σ_0 that works well and it can happen that no value of σ_0 works well. In that case alternative solutions need to be investigated. There are solutions but none are simple.