

ENGINEERS DO IT, SCIENTISTS DO IT, MATHEMATICIANS?

JOE SANTMYER

ABSTRACT. There is a magnificent mathematical gem missing from most numerical analysis curricula. A literature search of numerical analysis and mathematical modeling texts indicates its absence. What is missing from the numerical analysis toolbox and why isn't it there? The missing tool is the *Kalman filter*. The Kalman filter requires a modest knowledge of statistics. Is this why it is missing from the toolbox? Read and reach your own conclusion whether this piece of mathematics should be part of a numerical analysis or mathematical modeling curriculum.

1. MISSING FROM THE NUMERICAL ANALYSIS TOOLBOX

In a numerical analysis toolbox [3, 10] you find such things as Gauss-Jordon, Gauss-Seidel elimination, polynomial interpolation and extrapolation, the trapezoid rule for integration, Runge-Kutta method for ODE's, etc., but the Kalman filter is missing. In [2] it is claimed that "along with the simplex method" the Kalman filter is "probably the single most *useful* piece of mathematics developed this century." It played an essential role in getting the Viking I and II spacecraft from earth to Mars. It is used in navigation, computer vision, to track objects, etc., anywhere sensors obtain measurements subject to noise. Engineers and scientists have used the Kalman filter for several decades following the seminal paper [4] written by Rudolph Kalman. Yet mathematicians seem to have overlooked it.

Literally hundreds of journal articles have been published since Kalman's original work, extending the filter in numerous directions. If this is such a significant result in engineering and applied mathematics then why doesn't it appear in today's numerical analysis books and curriculum? One drawback is that it requires some probability and statistics, though the prerequisites are modest. Most students taking an applied mathematics or numerical analysis course would probably take a course in probability and statistics anyway. Still, there is no guarantee they do. The purpose of this article is to expose this useful tool to the mathematician and show that

the necessary probability and statistics required to understand it could easily be incorporated into a numerical analysis course. The little time spent on prerequisites is well worth the reward of providing knowledge of an extremely useful tool students might use later in graduate school or industry.

2. WHERE TO START

One way to introduce a Kalman filter is to show what it does. Consider a couple of examples which are slight variations of [11]. A Kalman filter is an algorithm, in fact, a recursive set of equations. As a comparison, consider a recursive equation to compute the mean of a set of data values obtained from measurements $x_1, x_2, \dots, x_{n-1}, x_n$ taken periodically over time where x_1 is the oldest value up to x_n , the most recent value. Convince yourself that if \bar{x}_n is the mean of n measurements then

$$\bar{x}_n = \frac{(n-1)\bar{x}_{n-1} + x_n}{n}$$

for $n \geq 1$ with initial condition $\bar{x}_0 = 0$. Normally we don't think of a mean or average as being calculated recursively. However, presented this way, it serves as a simple example which processes data recursively in a manner similar to a Kalman filter. And why recursive? Because that is what makes the Kalman filter so attractive. When it comes to processing data in *real-time* recursion translates into efficiency. And why the term filter? Because the average and Kalman algorithms "filter out" unwanted noise, revealing a, hopefully, good estimate of the truth.

3. WATER TANK

The first example takes measurements of the water level of a water tank. The contents of the tank cannot be seen and measurements are taken to determine water level over time. Assume the water level is constant over a short period when it is measured. Like all sensors or measuring devices they have inherent noise. The Kalman filter has proven to be an indispensable tool to deal with noise.

Before looking into the details of the Kalman filter equations, let's see how it compares to taking the average. Ten measurements are taken. Figure 1 shows the results of repeatedly taking the average of the measurements so far, a sort of *running* average. The true water level, measurements and running average are plotted in Figure 1. The data was purposely made noisy to see how the running average copes with it.

Figure 2 shows how the Kalman filter deals with the noisy measurements. If you compare Figures 1 and 2, you see little difference. In fact, when both the running average and Kalman filter plots were put in the same figure the

ENGINEERS DO IT, SCIENTISTS DO IT, MATHEMATICIANS?

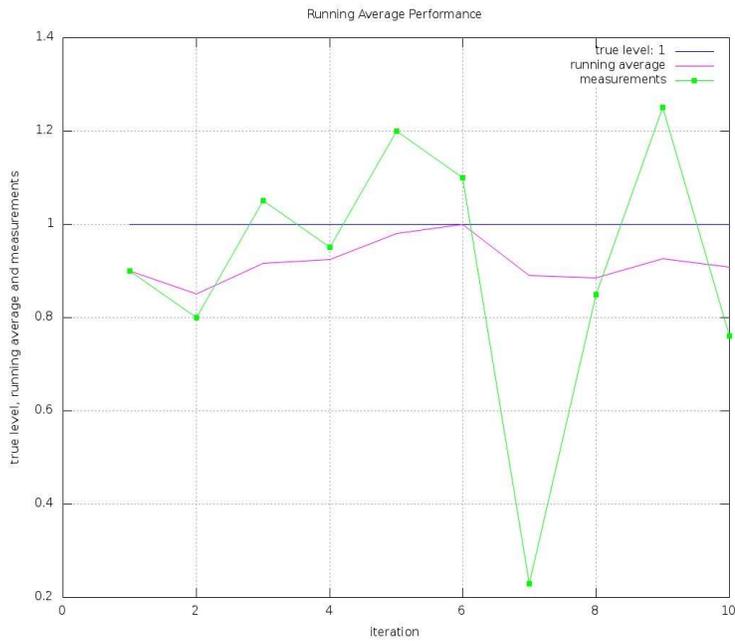


FIGURE 1. Tank filling using running average.

second wrote over and covered the first. That is why two separate figures were made.

How were these plots produced and what do the Kalman equations look like? The implementation of the Kalman and running average equations, together with the plots were produced with the software package *Octave* [9]. *Octave* is very similar to *Matlab* [6] but *Octave* is free and open source. The Kalman equations during the n th iteration for the water tank example

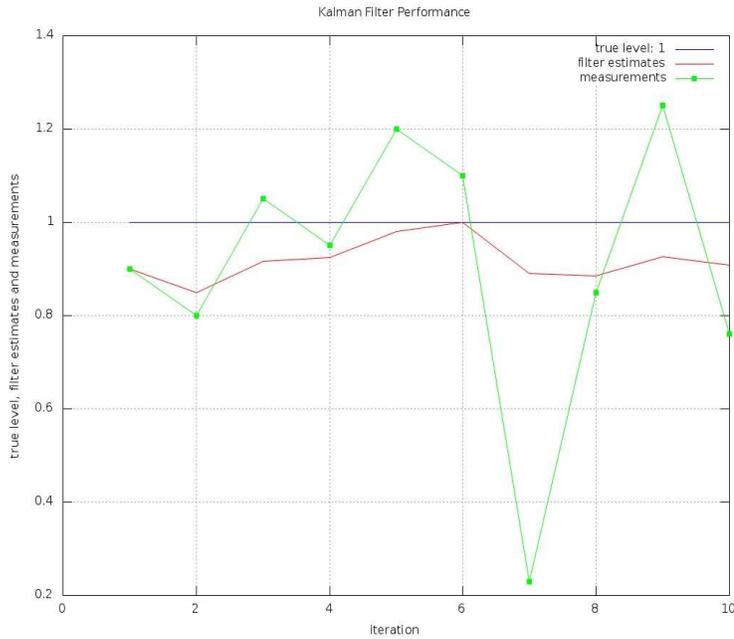


FIGURE 2. Tank filling using Kalman filter

are

$$\begin{aligned}
 x_n &= x_n^-, \\
 P_n &= P_n^- + Q, \\
 K_n &= P_n^- (P_n^- + R)^{-1}, \\
 x_n &= x_n^- + K_n (y_n - x_n^-), \\
 P_n &= (I - K_n) P_n^-, \\
 x_{n+1}^- &= x_n, \\
 P_{n+1}^- &= P_n;
 \end{aligned}$$

where x_n^- is the previous state, P_n^- is the previous state variance, Q is the process variance, R is measurement noise, K_n is referred to as the *Kalman gain*, y_n is a measurement, I is the identity matrix and P_n is used to update P_n^- . I know what you are thinking. What do the previous state variance, process variance, measurement noise, Kalman gain, etc., mean? Please be patient. More is said about them below. Whatever is going on, you must

ENGINEERS DO IT, SCIENTISTS DO IT, MATHEMATICIANS?

admit, somehow the Kalman estimates, like the running average, begin to approach the true water level in spite of the noisy data.

The first two equations are called the *prediction* equations, the next three the *Kalman* equations and the last two the *update* equations. Only the middle three equations are often referred to as the Kalman equations proper. Initially, $x_0^- = 0$, $P_0^- = 1000$ (high state variance), $R = 0.1$, $Q = 0.0001$ (small process variance), measurements are taken every $\Delta t = 1$ sec, and true water level is 1.

Upper case letters normally represent matrices and lower case vectors. In the water tank example all quantities are one dimensional (scalars). The exponent -1 on the right side of the equation for K_n usually represents the inverse of a matrix but since all quantities here are scalars it denotes the reciprocal of a number. The algorithm repeatedly uses the equations to recalculate as more measurements are processed. The Kalman filter estimates plotted in Figure 2 are the values x_n^- .

4. VEHICLE POSITION AND VELOCITY

The next example estimates the position and velocity of a vehicle from simulated noisy sensor data. Figure 3 shows plots of the true position of the vehicle, estimated position obtained from a Kalman filter and running average, and the measurements. As Figure 3 indicates, the Kalman filter clearly out performs the running average. In fact, in spite of how noisy the data is, the Kalman filter is obviously giving a very good estimate of the true position. The running average drifts away from the true position over time. Figure 4 shows the difference between true position minus measurement and true position minus Kalman estimated position, and Figure 5 shows true and estimated velocity. The figures clearly show that whatever is going on here the results look good. There must be something to this Kalman filter.

The moving vehicle can be described by the following linear system

$$\begin{aligned} \mathbf{x}_{n+1} &= A\mathbf{x}_n + B\mathbf{u}_n + \mathbf{w}_n && \text{state equation} \\ \mathbf{y}_n &= C\mathbf{x}_n + \mathbf{z}_n && \text{output equation} \end{aligned}$$

where A , B and C are matrices, n is a time index, vector \mathbf{x}_n is the *state* of the system, vector \mathbf{u}_n is a known input to the system, vector \mathbf{y}_n is measured output, vector \mathbf{w}_n is process noise and vector \mathbf{z}_n is measurement noise. The state of the system is the vehicle's position and velocity. In the absence of noise, velocity and position can be updated from time index n to $n + 1$,

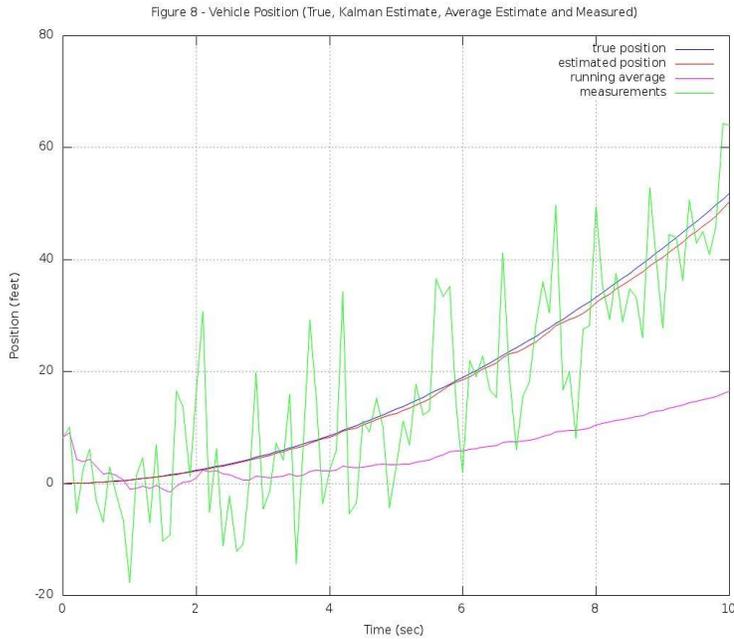


FIGURE 3. Vehicle position, Kalman filter and average estimates.

constituting time interval Δt , as follows

$$v_{n+1} = v_n + (\Delta t)u_n$$

$$p_{n+1} = p_n + (\Delta t)v_n + \frac{(\Delta t)^2}{2}u_n$$

where u_n is input acceleration and the second equation was obtained from the first by integrating. When noise is taken into consideration the equations for velocity and position become

$$v_{n+1} = v_n + (\Delta t)u_n + v_n^{\sim}$$

$$p_{n+1} = p_n + (\Delta t)v_n + \frac{(\Delta t)^2}{2}u_n + p_n^{\sim}$$

where v_n^{\sim} and p_n^{\sim} are velocity and position noise. The later two equations for velocity and position are more realistic because they take instrument noise into account.

ENGINEERS DO IT, SCIENTISTS DO IT, MATHEMATICIANS?

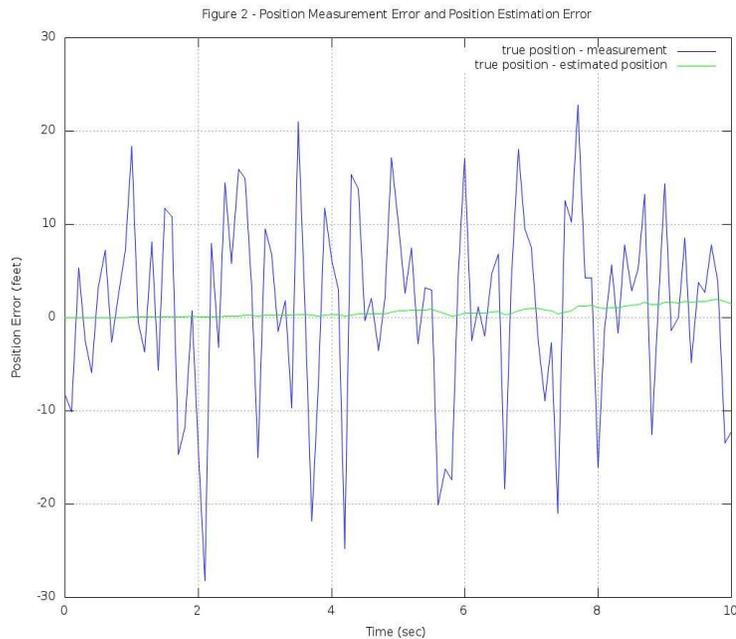


FIGURE 4. Vehicle position measurement and estimate error

5. SYSTEM STATE

The state vector is $\mathbf{x}_n = \begin{pmatrix} p_n \\ v_n \end{pmatrix}$ and the linear system can be written as

$$\mathbf{x}_{n+1} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \mathbf{x}_n + \begin{pmatrix} (\Delta t)^2/2 \\ T \end{pmatrix} \mathbf{u}_n + \mathbf{w}_n$$

$$\mathbf{y}_{n+1} = \begin{pmatrix} 0 & 1 \end{pmatrix} \mathbf{x}_n + \mathbf{z}_n.$$

6. PROCESS AND MEASUREMENT NOISE COVARIANCE

Assume measurements are taken every $\frac{1}{10}$ sec, *i.e.*, $\Delta t = 0.1$ sec. Also assume $p_0 = 0$, $v_0 = 0$ and acceleration is constant at 1 ft/sec^2 , that is, $a_0 = a = 1 \text{ ft/sec}^2$. Then $p = p_0 + (\Delta t)v_0 + \frac{(\Delta t)^2}{2}a_0 = 0 + (.1)(0) + \frac{(0.1 \text{ sec})^2}{2} \times \frac{1 \text{ ft}}{\text{sec}^2} = 0.005 \text{ ft}$. Assume acceleration noise is $\sigma_a = 0.2 \text{ ft/sec}^2$. The standard deviation of position noise is therefore $\sigma_P = (0.005)(0.2) = 10^{-3}$. Also $v = v_0 + (\Delta t)a_0 = 0 + (0.1 \text{ sec})\left(\frac{1 \text{ ft}}{\text{sec}^2}\right) = 0.1 \text{ ft/sec}$. The

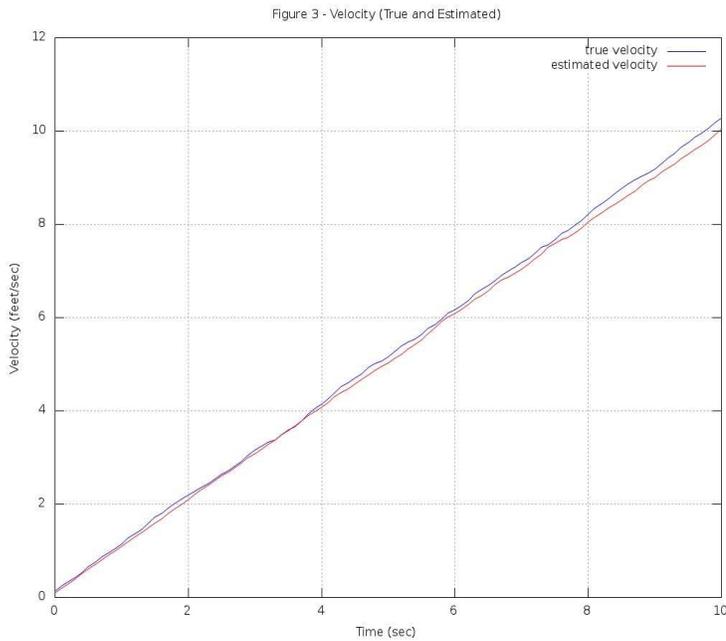


FIGURE 5. Vehicle velocity true value and Kalman filter

standard deviation of velocity noise is therefore $\sigma_V = (0.1)(0.2) = 0.02 = 2 \times 10^{-2}$. This accounts for process noise that originates from acceleration noise. Consequently, the process noise vector $\mathbf{w} = \begin{pmatrix} 10^{-3} \\ 2 \times 10^{-2} \end{pmatrix}$.

Assume process position noise P is a random variable (RV) having a normal distribution with mean $\mu_P = 0$ and variance σ_P^2 . The statistician's notation for this is $P \sim N(0, \sigma_P^2)$, which we adopt here. Assume process velocity noise V is an RV and $V \sim N(0, \sigma_V^2)$. So the standard deviation for process position noise is $\sigma_P = \frac{(\Delta t)^2}{2} a \sigma_a$ and the standard deviation for process velocity noise is $\sigma_V = (\Delta t) a \sigma_a$. The process noise vector is $\mathbf{w} = \begin{pmatrix} \sigma_P \\ \sigma_V \end{pmatrix}$. Consequently, the process noise covariance matrix is

$$\begin{aligned}
S_w &= \text{Cov}(P, V) \\
&= E[\mathbf{w}\mathbf{w}^T] \\
&= E \left[\begin{pmatrix} \sigma_P \\ \sigma_V \end{pmatrix} \begin{pmatrix} \sigma_P & \sigma_V \end{pmatrix} \right] \\
&= \begin{pmatrix} \sigma_P^2 & \sigma_P \sigma_V \\ \sigma_V \sigma_P & \sigma_V^2 \end{pmatrix} \\
&= \begin{pmatrix} \left(\frac{(\Delta t)^2}{2} a \sigma_a \right)^2 & \frac{(\Delta t)^2}{2} a \sigma_a (\Delta t) a \sigma_a \\ (\Delta t) a \sigma_a \frac{(\Delta t)^2}{2} a \sigma_a & ((\Delta t) a \sigma_a)^2 \end{pmatrix} \\
&= a^2 \sigma_a^2 \begin{pmatrix} \frac{(\Delta t)^4}{4} & \frac{(\Delta t)^3}{2} \\ \frac{(\Delta t)^3}{2} & (\Delta t)^2 \end{pmatrix} \\
&= (1)^2 (0.2)^2 \begin{pmatrix} \frac{(0.1)^4}{4} & \frac{(0.1)^3}{2} \\ \frac{(0.1)^3}{2} & (0.1)^2 \end{pmatrix} \\
&= \begin{pmatrix} 10^{-6} & 2 \times 10^{-5} \\ 2 \times 10^{-5} & 4 \times 10^{-4} \end{pmatrix}.
\end{aligned}$$

Assume measurement position noise P_m is an RV and $P_m \sim N(0, \sigma_{P_m}^2)$ where the measurement position noise scalar is $z = \sigma_{P_m} = 10$ ft. Consequently, the measurement position covariance matrix $\text{Cov}(P_m, P_m) = S_z = (\sigma_{P_m})(\sigma_{P_m})^T = \sigma_{P_m}^2 = 10^2$.

7. LINEAR SYSTEM AND KALMAN EQUATIONS

Now consider the state, output and Kalman equations for the n th iteration. First, a few preliminaries. Matrices A , B , and C are used in the equations below where $A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} \frac{(\Delta t)^2}{2} \\ \Delta t \end{pmatrix}$ and $C = \begin{pmatrix} 1 & 0 \end{pmatrix}$. Matrix A is called the *transition* matrix and B the *input* matrix. The initial state vector is $\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. The initial input acceleration, and all subsequent acceleration values are $\mathbf{u}_0 = \mathbf{u}_n = 1$, which are scalars in this case. The random process noise vector is $\mathbf{w}_n = \begin{pmatrix} \frac{(\Delta t)^2 r_1}{2} \\ (\Delta t) r_2 \end{pmatrix}$ and the n th noise measurement vector (which is actually a scalar) is $\mathbf{z}_n = \sigma_{P_m} r_3$ where r_1 , r_2 and r_3 are random numbers. New random numbers are generated during each iteration of the algorithm. All random numbers are produced by

the `randn` library function available in Octave and Matlab which generates random numbers based on RV R where $R \sim N(0, 1)$. The initial process covariance matrix is $P_0 = S_w$. The state, output and Kalman equations are

$$\begin{aligned}\mathbf{x}_n &= A\mathbf{x}_n^- + B\mathbf{u}_n + \mathbf{w}_n \\ \mathbf{y}_n &= C\mathbf{x}_n + \mathbf{z}_n\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{x}}_n &= A\hat{\mathbf{x}}_n^- + B\mathbf{u}_n \\ \mathbf{v} &= \mathbf{y}_n - C\hat{\mathbf{x}}_n \\ S &= CP_nC^T + S_z \\ K_n &= AP_nA^T S^{-1} \\ \hat{\mathbf{x}}_{n+1} &= \hat{\mathbf{x}}_n + K_n\mathbf{v} \\ P_{n+1} &= AP_nA^T - AP_nC^T S^{-1}CP_nA^T + S_w\end{aligned}$$

where \mathbf{x}_n^- is the previous state, $\hat{\mathbf{x}}_n^-$ and $\hat{\mathbf{x}}_n$ are the previous and next *state estimate*, \mathbf{v} is the *innovation* vector, S is the *covariance of innovation*, and K_n is the *Kalman gain*. This is the jargon used by engineers that work in the linear system world. The first two equations, state and output equations, form the linear system. The remaining equations are considered the Kalman equations. The last two equations prepare the state estimate and process covariance matrix for the next iteration. Exponent T denotes the transpose of a matrix and exponent -1 is the inverse of a matrix.

Here is a little explanation of the above equations. In the calculation of $\hat{\mathbf{x}}_{n+1}$, the first term $\hat{\mathbf{x}}_n$ is the propagated state estimate from $A\hat{\mathbf{x}}_n^- + B\mathbf{u}_n$ in the absence of a measurement. The second term $K_n\mathbf{v}$ is a correction term. It represents the amount by which to correct the propagated state estimate. Consider the equation for the Kalman gain K_n . If measurement noise is large, S_z will be large, so K_n is small and we don't give much credibility to measurement \mathbf{y}_n . On the other hand, if measurement noise is small, S_z is small, so K_n is large and we give more credibility to the measurement.

8. REAL-TIME EMBEDDED SOFTWARE

Nowhere else does a Kalman filter have greater impact than in *real-time* processing. The world of embedded micro-controllers containing a micro-processor, together with analog to digital (ADC) sensors, timers, wireless communication, USB connections, etc., is growing exponentially, and prolific in our everyday lives. Cell phones, sensors in automobiles, robots,

ENGINEERS DO IT, SCIENTISTS DO IT, MATHEMATICIANS?

appliances, medical monitoring devices, routers, etc., just to name a few, contain micro-controllers embedded inside them that process measurements in real-time. A Kalman filter can be implemented efficiently in the real-time embedded software to deal with noise inherent in the measuring devices to produce reliable results. The impact on our lives is obvious. This is why knowledge of it is a tremendous asset for a student entering the working world or graduate school.

9. LOST AT SEA

Graphs illustrating the Kalman filter and seeing how well it performs in the presence of noisy data is all well and good but the mathematician asks “where’s the proof?” A complete proof of the Kalman filter is outside the scope of this article. There are many references such as [1] which provide a proof in a general multidimensional context required by the Kalman filter. However, some steps towards quenching the mathematician’s thirst for proof can be presented through the following variant of a one dimensional example found in [7].

You are lost at sea at night and have no idea of your location. You use the stars to locate your position. Your location is specified as the one dimensional distance from a fixed reference point. At time t_1 you determine that your location is z_1 based on your measurements. However, your measurements have inherent inaccuracies and you assume that the variance of your measurements is $\sigma_{z_1}^2$. Your friend, a trained navigator, takes an independent fix of your location. Immediately after you, at time t_2 , obtain a more accurate measurement with mean z_2 and (smaller) variance $\sigma_{z_2}^2$. You have two measurements of your location. How do you combine them?

10. BAYESIAN ESTIMATION

Assume a *prior* distribution of RV M to have a normal distribution with mean z_1 and variance $\sigma_{z_1}^2$, that is, $M \sim N(z_1, \sigma_{z_1}^2)$. Let Z_2 be a normal RV for the sample measurement with mean z_2 and variance $\sigma_{z_2}^2$, i.e., $Z_2 \sim N(z_2, \sigma_{z_2}^2)$. A method [5, 8] known as *Bayesian estimation* is used to determine the *posterior* distribution of M , given $Z_2 = z_2$, namely

$$p(\mu|z_2) = \frac{h(\mu)f(z_2|\mu)}{g(z_2)}$$

where the probability density function (PDF) of the prior distribution is

$$h(\mu) = \frac{1}{\sigma_{z_1}\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\mu-z_1}{\sigma_{z_1}}\right)^2} \quad \text{since} \quad M \sim N(z_1, \sigma_{z_1}^2)$$

and, for $M = \mu$, the PDF of the sampling distribution is

$$f(z_2|\mu) = \frac{1}{\sigma_{z_2}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{z_2-\mu}{\sigma_{z_2}}\right)^2} \quad \text{since} \quad Z_2 \sim N(z_2, \sigma_{z_2}^2).$$

The marginal PDF $g(z_2)$ will be determined later.

Plug in $h(\mu)$, $f(z_2|\mu)$ and $g(z_2)$ to get

$$p(\mu|z_2) = \frac{1}{2\pi\sigma_{z_1}\sigma_{z_2}g(z_2)} e^{-\frac{1}{2}\left(\frac{\mu-z_1}{\sigma_{z_1}}\right)^2 - \frac{1}{2}\left(\frac{z_2-\mu}{\sigma_{z_2}}\right)^2}.$$

which looks like a PDF for a normal distribution. It is, indeed, normal. To show this, the exponent (power) of e , denoted here as $P(\mu)$, is rewritten so that completing the square can be done.

11. REARRANGE $P(\mu)$

A little algebra yields

$$P(\mu) = -\frac{1}{2} \left(\frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2} \right) \mu^2 + \left(\frac{z_1}{\sigma_{z_1}^2} + \frac{z_2}{\sigma_{z_2}^2} \right) \mu - \frac{1}{2} \left(\frac{z_1^2}{\sigma_{z_1}^2} + \frac{z_2^2}{\sigma_{z_2}^2} \right).$$

Let $\frac{1}{\sigma^2} = \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2}$ so that the coefficient of μ^2 in $P(\mu)$ becomes $-\frac{1}{2\sigma^2}$. If $-\frac{1}{2\sigma^2}$ is a common factor of the μ^2 and μ terms of $P(\mu)$, it can be factored out of these terms. Completing the square can then be performed to help identify $p(\mu|z_2)$ as the PDF of a normal distribution.

To get $-\frac{1}{2\sigma^2}$ as a factor of μ solve

$$\frac{\mu_1}{\sigma^2} = \frac{z_1}{\sigma_{z_1}^2} + \frac{z_2}{\sigma_{z_2}^2}$$

for μ_1 to get

$$\mu_1 = \frac{\sigma_{z_2}^2 z_1}{\sigma_{z_1}^2} + \frac{\sigma_{z_1}^2 z_2}{\sigma_{z_2}^2}.$$

Consequently

$$\begin{aligned} P(\mu) &= -\frac{1}{2\sigma^2} \mu^2 + \frac{\mu_1}{\sigma^2} \mu - \frac{1}{2} \left(\frac{z_1^2}{\sigma_{z_1}^2} + \frac{z_2^2}{\sigma_{z_2}^2} \right) \\ &= -\frac{1}{2\sigma^2} (\mu^2 - 2\mu_1\mu) - \frac{1}{2} \left(\frac{z_1^2}{\sigma_{z_1}^2} + \frac{z_2^2}{\sigma_{z_2}^2} \right). \end{aligned}$$

12. COMPLETE THE SQUARE

Complete the square to get

$$\begin{aligned} P(\mu) &= -\frac{1}{2\sigma^2}(\mu - \mu_1)^2 + \frac{\mu_1^2}{2\sigma^2} - \frac{1}{2}\left(\frac{z_1^2}{\sigma_{z_1}^2} + \frac{z_2^2}{\sigma_{z_2}^2}\right) \\ &= -\frac{1}{2}\left(\frac{\mu - \mu_1}{\sigma}\right)^2 + R \end{aligned}$$

where $R = \frac{\mu_1^2}{2\sigma^2} - \frac{1}{2}\left(\frac{z_1^2}{\sigma_{z_1}^2} + \frac{z_2^2}{\sigma_{z_2}^2}\right)$. Function $p(\mu|z_2)$ can now be written as

$$p(\mu|z_2) = \frac{e^R}{2\pi\sigma_{z_1}\sigma_{z_2}g(z_2)} e^{-\frac{1}{2}\left(\frac{\mu - \mu_1}{\sigma}\right)^2}.$$

The marginal PDF $g(z_2)$ is defined as

$$\begin{aligned} g(z_2) &= \int_{\text{range of } \mu} h(\mu)f(z_2|\mu)d\mu \\ &= \int_{-\infty}^{\infty} h(\mu)f(z_2|\mu)d\mu \\ &= \int_{-\infty}^{\infty} \frac{e^R}{2\pi\sigma_{z_1}\sigma_{z_2}} e^{-\frac{1}{2}\left(\frac{\mu - \mu_1}{\sigma}\right)^2} d\mu \\ &= \frac{e^R\sigma\sqrt{2\pi}}{2\pi\sigma_{z_1}\sigma_{z_2}} \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\mu - \mu_1}{\sigma}\right)^2} d\mu \\ &= \frac{\sigma e^R}{\sigma_{z_1}\sigma_{z_2}\sqrt{2\pi}}. \end{aligned}$$

Plug $g(z_2) = \frac{\sigma e^R}{\sigma_{z_1}\sigma_{z_2}\sqrt{2\pi}}$ into the equation for $p(\mu|z_2)$ to get

$$p(\mu|z_2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\mu - \mu_1}{\sigma}\right)^2}.$$

This is the PDF of a normal distribution with mean μ_1 and variance σ^2 where

$$\mu_1 = \frac{\sigma_{z_2}^2 z_1}{\sigma_{z_1}^2} + \frac{\sigma_{z_1}^2 z_2}{\sigma_{z_2}^2} \quad \text{and} \quad \frac{1}{\sigma^2} = \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2}.$$

13. YOUR TURN

The above derivation of μ_1 and σ was based on Bayesian estimation. Another approach [12] is to use χ^2 estimation. An outline of this approach is given and you are asked to fill in the details. Perhaps this can be the

start of a capstone or undergraduate research project that you, the student, would like to pursue. Start by considering the following expression

$$\chi^2 = \frac{(z_1 - \hat{x})^2}{\sigma_{z_1}^2} + \frac{(z_2 - \hat{x})^2}{\sigma_{z_2}^2}$$

where z_1, z_2, σ_{z_1} and σ_{z_2} are all the same as before. Treat χ^2 as a function of \hat{x} and take the derivative with respect to \hat{x} . Set the derivative equal to zero and solve for \hat{x} . Does it look familiar?

If $\sigma_{z_1}^2 = E[(z_1 - \mu_x)^2]$ and $\sigma_{z_2}^2 = E[(z_2 - \mu_x)^2]$ where E is expected value, find $\sigma_x^2 = E[(\hat{x} - \mu_x)^2]$ by plugging in the expression you obtained for \hat{x} . Keep in mind that z_1 and z_2 are *independent*. What do you get for σ_x^2 ?

14. OBSERVATIONS OF μ_1 AND σ^2

Note that $\sigma^2 < \min(\sigma_{z_1}^2, \sigma_{z_2}^2)$. Therefore, two measurements yield a PDF with smaller variance, and hence, a better estimate of the true position than either single measurement provides. Suppose $\sigma_{z_2}^2 = \sigma_{z_1}^2$. Use the equation for μ_1 to show that $\mu_1 = \frac{z_1 + z_2}{2}$ which is the average of the two measurements. Suppose $\sigma_{z_2}^2 > \sigma_{z_1}^2$. Then the coefficient of z_1 in the expression for μ_1 is larger than the coefficient of z_2 giving more weight to the estimate z_1 when calculating μ_1 . This is desirable because z_1 would be the better estimate since it has a smaller variance. Based on both measurements z_1 and z_2 an estimate of μ_1 was obtained. As it turns out, it is the maximum likelihood estimate, the weighted least squares estimate and the linear estimate whose variance is less than that of any other linear unbiased estimate. Consequently, it is the “best” estimate according to any reasonable criterion.

15. A LITTLE KALMAN FILTER

The equation for μ_1 can be rearranged to derive a formula for the estimate $\hat{x}(t_2) = \mu_1$ in terms of the estimate $\hat{x}(t_1)$. Using the fact that $\hat{x}(t_1) = z_1$ show that

$$\hat{x}(t_2) = \hat{x}(t_1) + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} [z_2 - \hat{x}(t_1)].$$

This yields the following *Kalman Filter*

$$\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)[z_2 - \hat{x}(t_1)]$$

where

$$K(t_2) = \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}.$$

$K(t_2)$ is the Kalman gain that was mentioned earlier. This equation provides one form of the Kalman filter formula. This gives a way of deriving the

ENGINEERS DO IT, SCIENTISTS DO IT, MATHEMATICIANS?

best estimate of your position, $\hat{x}(t_2)$ at time t_2 based on the best estimate $\hat{x}(t_1)$ at time t_1 and the measurement z_2 at time t_2 .

The above Kalman filter equation can be used to derive an analogous formula to calculate the variance $\sigma_x^2(t_2)$ in terms of $\sigma_x^2(t_1)$ and $K(t_2)$. Show that

$$\sigma^2 = [1 - K(t_2)]\sigma_{z_1}^2.$$

Since $\sigma_{z_1}^2 = \sigma_x^2(t_1)$ and $\sigma^2 = \sigma_x^2(t_2)$ the last equation can be rewritten as

$$\sigma_x^2(t_2) = [1 - K(t_2)]\sigma_x^2(t_1).$$

This equation provides a formula for calculating $\sigma_x^2(t_2)$ from $\sigma_x^2(t_1)$ and $K(t_2)$ analogous to the equation used to calculate $\hat{x}(t_2)$ from $\hat{x}(t_1)$ and $K(t_2)$. The pair of equations for $\hat{x}(t_2)$ and $\sigma_x^2(t_2)$ solve what is referred to as the *static estimation problem*.

16. DYNAMIC ESTIMATION PROBLEM

Returning to the “lost at sea” example, suppose you travel for some time before taking a third measure z_3 of your position at time t_3 . Assume that the best model of your motion is given by the differential equation

$$\frac{dx}{dt} = u + w$$

where u is a nominal velocity and w noise that is assumed to have a normal PDF with zero mean and variance σ_w^2 . Let t_3^- represent the time just before t_3 when the third measurement z_3 is taken.

Given a third measurement z_3 of your position $x(t_3)$ at t_3 , a fourth PDF having a normal distribution can be defined. The fourth PDF at time t_3^- has a mean $\hat{x}(t_3^-)$ and variance $\sigma_x^2(t_3^-)$ which satisfy the following two equations

$$\begin{aligned}\hat{x}(t_3^-) &= \hat{x}(t_2) + u[t_3 - t_2] \\ \sigma_x^2(t_3^-) &= \sigma_x^2(t_2) + \sigma_w^2[t_3 - t_2].\end{aligned}$$

The value of $\hat{x}(t_3^-)$ is the best estimate of your position at time t_3^- just before measurement z_3 at time t_3 is taken and $\sigma_x^2(t_3^-)$ is the variance in that estimate. If the Kalman filter equation for $\hat{x}(t_2)$ above is applied with $\hat{x}(t_3^-)$ being the previous best estimate at time t_3^- , z_3 the measurement at time t_3 we get

$$\hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)[z_3 - \hat{x}(t_3^-)]$$

for the best estimate $\hat{x}(t_3)$ in position at time t_3 where

$$K(t_3) = \frac{\sigma_x^2(t_3^-)}{\sigma_x^2(t_3^-) + \sigma_{z_3}^2}.$$

Similarly, if the equation above for $\sigma_x^2(t_2)$ is applied with $\sigma_x^2(t_3^-)$ being the best estimate of the variance at time t_3^- , we can calculate the best estimate of the variance $\sigma_x^2(t_3)$ at time t_3 as follows

$$\sigma_x^2(t_3) = [1 - K(t_3)]\sigma_x^2(t_3^-).$$

Extending equations $\hat{x}(t_3^-)$ and $\sigma_x^2(t_3^-)$ to vectors yields the general Kalman filter algorithm for updating from one measurement to the next measurement. Extending equations $K(t_3)$ and $\sigma_x^2(t_3)$ to vectors yields the Kalman filter update at a measurement time.

REFERENCES

- [1] R. Brown, P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, 1997.
- [2] J. Casti, *Five More Golden Rules*, John Wiley & Sons, New York, 2000.
- [3] G. Golub and J. Ortega, *Scientific Computing an Introduction to Parallel Computing*, Academic Press, San Diego, 1993.
- [4] R. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, Transactions of the ASME - Journal of Basic Engineering, **82** (Series D), (1960), 35–45.
- [5] H. Larson, *Introduction to Probability Theory and Statistical Inference*, John Wiley & Sons, New York, 1969.
- [6] Matlab, (2013); <http://www.mathworks.com/products/matlab/>.
- [7] P. Maybeck, *Stochastic Models, Estimation and Control*, Vol. 1, Academic Press, Inc., New York, 1979.
- [8] I. Miller and M. Miller, *Mathematical Statistics with Applications*, Prentice Hall, New Jersey, 2004.
- [9] GNU Octave, (2013); <https://www.gnu.org/software/octave/>.
- [10] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, Cambridge University Press, New York, 1992.
- [11] D. Simon, *Kalman Filtering* (2013), <http://www.masys.ur1.tw/AU/2013SP/OpenCV/0513/kalman-dan-simon.pdf>.
- [12] M. Spetsakis, *Computer Vision*, (2013), <http://www.cse.yorku.ca/cspeople/faculty/minas/>.

MSC2010: 97N40, 97N60, 97R20, 97R30

Keywords and phrases: numerical analysis, mathematical modeling, real-time processing, embedded software

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, EAST CENTRAL UNIVERSITY, ADA, OKLAHOMA 74820-6999
E-mail address: santmyerjoe@yahoo.com