Kalman Filter Notes – March 25, 2004

1. Introduction to the Kalman Filter:

- "best linear unbiased estimate" Taligrand (J. Japan Meteor. Soc., 1997)
- "optimal recursive data processing algorithm", Maybeck (1979)
- Least squares estimator that takes into effect:
  - Error is measurement (accts for confidence)
  - Noise in modeled system

  This is why it is called a "filter".

Original system studied using several assumptions. Errors in measurements and system noise are:

- white noise
- gaussian
- System is linear

2. Taligrand example

Consider measuring a scalar, x^t, such as temperature with two different instruments (such as a mercury thermometer and a thermograph). Each of these instruments has its own error properties.

$$z_1 = x^t + \varepsilon_1$$
$$z_2 = x^t + \varepsilon_2$$

The errors properties are that they are unbiased and that their variance is known:

$$Mean[\varepsilon] = 0$$
$$Var[\varepsilon^2] = \sigma^2$$

and to keep things simple, the correlation between the errors is zero,

$$Cor[\varepsilon_1 \varepsilon_2] = 0$$

How do we get the best, x^a, the best analysis of temperature? Lets assume that a linear combination of the two measurements can give one the best answer, i.e.,

$$x^a = a_1 z_1 + a_2 z_2$$

So lets determine the a's. We need two constraints. One obvious one, which actually means we want an unbiased estimate, so that,

$$a_1 + a_2 = 1$$

else the temperature would always be either too high or too low. Second, we want to minimize the analysis variance, i.e.,

$$Var\left[\left(x^a - x^t\right)^2\right] = \sigma^2$$

This could be done in several ways, but it might be obvious that,

$$a_1 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

$$a_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

The physical interpretation of this result is that the less you believe in the accuracy of one measurement, more weight should be given to the other measurement in the analysis.

3. Variational Approach

The same problem can be formulated as a variational problem: Minimize the distance between the analysis and the observations and take into account the accuracy of the measures. Create a cost function J(x),

$$J(x) = \frac{(x - z_1)^2}{\sigma_1^2} + \frac{(x - z_2)^2}{\sigma_2^2}$$

By determining the minimum of J(x), we find where the analysis value x^a will meet our requirement. Interestingly, we can convert either of these problems, given the assumptions, into a sequential estimator (a Kalman filter).

4. Kalman Filter

The first estimate of our temperature is given by,

$$x^a(t_1) = z_1$$
$$\sigma^2(t_1) = \sigma_1^2$$

i.e., our first guess is simply the measurement from our first instrument. Note the use of "t" here could mean time, or just a sequence of observations at the same time. The second estimate of the temperature is given by,

$$x^a(t_2) = x^a(t_1) + K(t_2)\left[z_2 - x^a(t_1)\right]$$
$$\sigma^2(t_2) = \sigma^2(t_1)\left[1 - K(t_2)\right]$$

where "K" is called the "Kalman Gain". Note that as in the first example, we have an estimate of the analysis variable AND its variance. Also notice that the analysis variance MUST decrease as we add information. So what does the K look like? Well, it's the same as before, the Kalman gain is given as

$$K(t_2) = \frac{\sigma^2(t_1)}{\sigma^2(t_1) + \sigma_2^2}$$

One can show trivial that this set of equations is the identical solution shown in Section (2).

So have we gained anything? Maybe so, assuming our problem can live within the working assumptions of Gaussian distributions and white noise power spectra. If the observations can be processed sequentially (i.e., they are uncorrelated), then one can take a problem that often is written as a variational method (and therefore needs to be solved as a matrix problem which can be messy) and convert it into a recursive algorithm that processes each observation in a very simple manner. This methodology is very minimalist in memory and cost. This is the power associated with the Kalman Filter.

5. Extended Kalman Filter

So far we have talked about measurements for analysis, but what about modeling systems? The following example shows the extended Kalman filter, which enables the use of the Kalman filter for nonlinear systems (removing one of the assumptions). Suppose we have a forecast model and a real world system that can be described as,

$$x^t(t) = G\left[x^t(t-\Delta t)\right]$$
$$x^f(t) = F\left[x^a(t-\Delta t)\right]$$

where "t" means the true state, "f" means the forecast state, and "a" means the analysis. The "G" and "F" are the nonlinear operators that advance the "solution" from one state to the next. The forecast model has error, and we represent this error as,

$$\eta(t) = G\left[x^t(t-\Delta t)\right] - F\left[x^a(t-\Delta t)\right]$$

and again we assume that $\eta(t)$ has a Gaussian distribution and white noise spectrum such that the mean is zero and the variance is Q. Note, that this type of error is not realistic for actual forecast models, as they have biases, etc. In our system, the observations are given as,

$$y_k(t) = H\left[x^t(t)\right] + \varepsilon_o$$
$$Mean\left[\varepsilon_o\right] = 0$$
$$Var\left[\varepsilon_o \varepsilon_o^T\right] = R$$

where ε is the observational noise and y_k is a column vector of observations. The operator "H" converts the knowledge of the true state of the atmosphere into variables that the forecast model understands (and needs for the analysis). Just like in the Kalman filter, if the observations are uncorrelated we can write down a sequential estimator for the analysis as,

$$x^a(t) = x^f(t) + K(t)\left[y_k(t) - H\left[x^f(t)\right]\right]$$

and recursive improve the analysis over each observation $y_{k.}$ What does K(t) look like here? It's a bit more involved because everything is a matrix, but if the covariance between the model noise η(t) and the observation error ε(t) is zero, then

$$K(t) = P^f(t) + H^T\left[HP^f(t)H^T + R(t)\right]^{-1}$$

which is pretty much a mess and incomprehensible to most people. $P^f(t)$ is the forecast error covariance, and that is actually were the work for all of this comes in. The $P^f(t)$ has to be *forecast* from another equation, which also involves the $P^a(t)$, the analysis error covariance. These equations look like,

$$P^f(t) = \tilde{F}\left[P^a(t-\Delta t)\right]F^T + Q$$
$$P^a(t) = P^f(t) - K(t)H\left[P^f(t)\right]$$

which is even more messy. Now we have to evolve the model and two error analysis covariances matrices! For realistic models, this is where the whole concept falls on itself because these matrices are the size of N^2, where N = # of grid points x number of variables. For a modern NWP model, N ~ 10^8, which means that NxN is WAY TO BIG to store or deal with! There are ways to cut down the size (and in fact, the Ensemble KF uses these implicitly), but this is why until recently, the use of Kalman filtering in geophysical modeling was almost never used……until someone came up with a crafty idea.

6. Ensemble Kalman Filter

Evensen (1994) proposed that instead of evolving the two covariance matrices directly that the equation for $P^f(t)$ could estimated from an *ensemble of model forecasts*. Since we have no absolute knowledge of the true state of the atmosphere, we use the mean values from the ensemble to approximate x^t. The equations that are used for the ensemble Kalman filter (EnKF) look like (relative to our previous examples):

$$
\begin{aligned}
&x_i^f(t) = F\left[x_i^a(t-\Delta t)\right], \quad i = 1, n \\
&P^f(t) = \frac{1}{n-1}\sum_i\left(\left[x_i^f(t) - \bar{x}(t)\right]\left[x_i^f(t) - \bar{x}(t)\right]^T\right) \\
&K(t) = P^f(t) + H^T\left[HP^f(t)H^T + R(t)\right]^{-1} \\
&x_i^a(t) = x_i^f(t) + K(t)\left[y_k(t) - H\left[x_i^f(t)\right]\right] \\
&P^a(t) = \frac{1}{n-1}\sum_i\left(\left[x_i^a(t) - \bar{x}(t)\right]\left[x_i^a(t) - \bar{x}(t)\right]^T\right)
\end{aligned}
$$

Notice that we have the R(t) term, the observational error accounted for, but notice the Q(t) cannot be accounted for here explicitly. This is one of the problems with this methodology. How many ensemble members do we need to have in order for $P^f(t)$ to be accurately estimated? Well, it turns out that probably having 1000 ensemble members is a very safe number, and probably 100 is workable. However, it is going to depend on the situation and dynamics involved, as well as how well the ensemble "spans" the solution space for the problem. Therefore, there is no hard and fast rule. Most NWP applications of

EnKF use between 50-200 members. Experiments with idealized chaotic systems (e.g., Lorenz models) often use 1000.