Suppose that we have a state, $x_t$, which is $m \times 1$, which evolves according to:

$$x_t = Cx_{t-1} + e_t$$

$e_t$ is a $r \times 1$ vector of “shocks”; $G$ is $m \times r$ and $C$ is $m \times m$. The state is, however, not necessarily observed. We observe variables that are linear combinations of the state. Assume that we observe a vector $y_t$, dimension $n \times 1$, potentially with noise. The “observer equation” is:

$$y_t = Dx_t + u_t$$

Here $D$ is $n \times m$ and $u_t$ is $n \times 1$, potentially allowing for measurement error.

We use a linear filter to forecast the state. In particular, our forecast of the current state is equal to our forecast from the previous period of today’s state plus an “updating” part that depends on innovations in the observer equation:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K(y_t - D\hat{x}_{t|t-1})$$

Our objective is to pick $K$ to minimize the variance of our errors. Let $\varepsilon_t = x_t - \hat{x}_{t|t}$. We choose $K$ to minimize the variance of this. The variance of this error can be written:

$$\text{var}(x_t - \hat{x}_{t|t}) = \text{var}(x_t - (\hat{x}_{t|t-1} + K(y_t - D\hat{x}_{t|t-1})))$$

Now plug in for $y_t$:

$$\text{var}(x_t - \hat{x}_{t|t}) = \text{var}(x_t - (\hat{x}_{t|t-1} + K(Dx_t + u_t - D\hat{x}_{t|t-1})))$$

Simplify:

$$\text{var}(x_t - \hat{x}_{t|t}) = \text{var}(x_t - \hat{x}_{t|t-1} - K(Dx_t + u_t - D\hat{x}_{t|t-1}))$$

$$= \text{var}(x_t - \hat{x}_{t|t-1} - KD(x_t - \hat{x}_{t|t-1}) - Ku_t)$$

$$= \text{var}((I - KD)(x_t - \hat{x}_{t|t-1}) - Ku_t)$$

Now, because $u_t$ is independent, we can move the variance operator through:

$$\text{var}(x_t - \hat{x}_{t|t}) = (I - KD)\text{var}(x_t - \hat{x}_{t|t-1})(I - KD)' + K\Sigma_u K'$$

For short hand, let $P_{t|t} = \text{var}(x_t - \hat{x}_{t|t})$ and $P_{t|t-1} = \text{var}(x_t - \hat{x}_{t|t-1})$. Then:

$$P_{t|t} = (I - KD)P_{t|t-1}(I - KD)' + K\Sigma_u K'$$
We want to choose $K$ so as to minimize this (i.e. minimize the variance of our error).  We can write this out in long hand a bit more:

$$P_{t|t} = P_{t|t-1} - KDP_{t|t-1} - P_{t|t-1}D'K' + KDP_{t|t-1}D'K' + K\Sigma_u K'$$

$$P_{t|t} = P_{t|t-1} - KDP_{t|t-1} - P_{t|t-1}D'K' + K(DP_{t|t-1}D' + \Sigma_u)K'$$

To minimize, we want to take the derivative with respect to 0 and set it equal to zero. We need to use a few matrix derivative rules. These are:

$$\frac{\partial A'x}{\partial x} = \frac{\partial x'A}{\partial x} = A'$$

$$\frac{\partial Ax}{\partial x} = \frac{\partial x'A}{\partial x} = A$$

$$\frac{\partial x'Ax}{\partial x} = Ax + A'x = x'(A + A')$$

$$\frac{\partial P_{t|t}}{\partial K} = -\frac{\partial (KDP_{t|t-1})}{\partial K} - \frac{\partial (P_{t|t-1}D'K')}{\partial K} + \frac{\partial (K(DP_{t|t-1}D' + \Sigma_u)K')}{\partial K}$$

To make the above rules operational, define an auxiliary variable $W = K'$. Then we can rewrite the above:

$$\frac{\partial P_{t|t}}{\partial K} = -\frac{\partial (W'DP_{t|t-1})}{\partial W} - \frac{\partial (P_{t|t-1}D'W)}{\partial W} + \frac{\partial (W'(DP_{t|t-1}D' + \Sigma_u)W)}{\partial W}$$

Now we can use those rules for derivatives. We have:

$$\frac{\partial (W'DP_{t|t-1})}{\partial W} = P_{t|t-1}D'$$

$$\frac{\partial (P_{t|t-1}D'W)}{\partial W} = P_{t|t-1}D'$$

$$\frac{\partial (W'(DP_{t|t-1}D' + \Sigma_u)W)}{\partial W} = W'((DP_{t|t-1}D' + \Sigma_u) + (DP_{t|t-1}D' + \Sigma_u)') + 2W'(DP_{t|t-1}D' + \Sigma_u)$$

The last line following from the fact that $\Sigma_u = \Sigma_u$ and $(DP_{t|t-1}D')' = DP_{t|t-1}D'$, which follows from the fact that $P_{t|t-1} = P_{t|t-1}$. Putting these together:

$$\frac{\partial P_{t|t}}{\partial K} = -2P_{t|t-1}D' + 2W'(DP_{t|t-1}D' + \Sigma_u) = 0$$

$$W' = P_{t|t-1}D'(DP_{t|t-1}D' + \Sigma_u)^{-1}$$

$$K = P_{t|t-1}D'(DP_{t|t-1}D' + \Sigma_u)^{-1}$$

We now have the optimal Kalman gain, we need to come up with a measure of $P_{t|t-1}$ in order to make this operational. Recall from above the definition of $P_{t|t-1}$:
\[ P_{t|t-1} = \text{var}(x_t - \hat{x}_{t|t-1}) \]

In words, this is the ex-ante variance of your forecast, whereas \( P_{t|t} \) is the ex-post variance (i.e. after you’ve gotten to see \( y_t \)). Our forecast of the state in period \( t \) conditional on information available in \( t-1 \) will simply result from plugging in our forecast of the \( t-1 \) state into the the equation governing the evolution of the actual state:

\[ \hat{x}_{t|t-1} = C\hat{x}_{t-1|t-1} \]

This is the forecast because the shock term is zero in expectation. We know that the actual state obeys:

\[ x_t = Cx_{t-1} + e_t \]

Plug this in:

\[
\begin{align*}
P_{t|t-1} &= \text{var}(Cx_{t-1} + e_t - C\hat{x}_{t-1|t-1}) \\
P_{t|t-1} &= \text{var}(C(x_{t-1} - \hat{x}_{t-1|t-1}) + e_t)
\end{align*}
\]

Using the independence of \( e_t \), we can move the expectations operator through to get:

\[ P_{t|t-1} = C \text{var}(x_{t-1} - \hat{x}_{t-1|t-1}) C' + \Sigma_e \]

But \( \text{var}(x_{t-1} - \hat{x}_{t-1|t-1}) = P_{t-1|t-1} \). Because this is unconditional variance (unconditional in the sense of not conditioning on particular realization), we have: \( P_{t-1|t-1} = P_{t|t} \). We have an expression for \( P_{t|t} \) from above:

\[ P_{t|t} = (I - KD)P_{t|t-1}(I - KD)' + K\Sigma_u K' \]

Plug this in:

\[ P_{t|t-1} = C ((I - KD)P_{t|t-1}(I - KD)' + K\Sigma_u K') C' + \Sigma_e \]

Because \( K \) is a function of \( P_{t|t-1} \), this is one equation in one unknown, and can in principle be solved for \( P_{t|t-1} \). We can in fact simplify this further:

\[
\begin{align*}
P_{t|t-1} &= C(P_{t|t-1} - KDp_{t|t-1} - P_{t|t-1}D'K' + K(DP_{t|t-1}D' + \Sigma_u)K')C' + \Sigma_e \\
&= C(P_{t|t-1} - KDp_{t|t-1} - P_{t|t-1}D'K' + P_{t|t-1}D'(DP_{t|t-1}D' + \Sigma_u)^{-1}(DP_{t|t-1}D' + \Sigma_u)K')C' + \Sigma_e \\
&= C(P_{t|t-1} - KDp_{t|t-1} - P_{t|t-1}D'K' + P_{t|t-1}D'K')C' + \Sigma_e
\end{align*}
\]

Now we’re left with:

\[ P_{t|t-1} = C(P_{t|t-1} - KDp_{t|t-1} - P_{t|t-1}D'K' + P_{t|t-1}D'K')C' + \Sigma_e \]
But now more stuff cancels out:

\[ P_{t|t-1} = C(P_{t|t-1} - KDP_{t|t-1})C' + \Sigma_e \]

Now plug in for the derived expression for \( K \) again:

\[ P_{t|t-1} = C(P_{t|t-1} - P_{t|t-1}D'(DP_{t|t-1}D' + \Sigma_u)^{-1}DP_{t|t-1})C' + \Sigma_e \]

This is now one expression in one unknown which can be solved for \( P_{t|t-1} \) (numerically in Matlab using fsolve). Given this, we then have \( K \).

It is common to write the Kalman filter not in terms of the innovation in the observer equation but rather in terms of the observed variables themselves. Start with the generic form of the filter:

\[
\begin{align*}
\tilde{x}_{t|t} &= \tilde{x}_{t|t-1} + K(y_t - D\tilde{x}_{t|t-1}) \\
\tilde{x}_{t|t} &= (I - KD)\tilde{x}_{t|t-1} + Ky_t \\
\tilde{x}_{t|t} &= (I - KD)C\tilde{x}_{t-1|t-1} + Ky_t
\end{align*}
\]

The last line follows from the fact that \( \tilde{x}_{t|t-1} = C\tilde{x}_{t-1|t-1} \).

**A Simple Example**

Consider a univariate state that is white noise:

\[ x_t = \epsilon_t \]

We observe the state with noise:

\[ y_t = x_t + u_t \]

Using the notation above, \( C = 0 \) and \( D = 1 \). In that case, \( P_{t|t-1} \) trivially reduces to \( \sigma_e^2 \). Then our optimal Kalman gain is:

\[
\begin{align*}
K &= P_{t|t-1}D'(P_{t|t-1}D' + \sigma_u^2)^{-1} \\
K &= \frac{\sigma_e^2}{\sigma_e^2 + \sigma_u^2}
\end{align*}
\]

Put differently, the optimal Kalman gain is the signal to noise ratio. If there is no noise, you perfectly observe the state each period. If there is a lot of noise, in this simple example, you basically always think that the state is equal to zero.

**A More Interesting Example**

This example actually simulates some data. The Matlab code is titled “univariate_kalman_example.m” and uses the function files “kalman.m” and ”vecy.m”. Consider a univariate state that is not white noise:

\[ x_t = \rho x_{t-1} + \epsilon_t \]
We observe the state with noise:

\[ y_t = \alpha x_t + u_t \]

Using our matrix notation from above, \( \rho = C \) and \( \alpha = D \). The forecasting rule will take the form:

\[ x_{t|t} = Ax_{t-1|t-1} + Ky_t \]

The Kalman gain takes the form from above, while \( A = (I - KD)C \). Choosing the following parameter values: \( \rho = 1, \alpha = 1, \sigma^2_x = 1, \) and \( \sigma^2_u = 1 \), I get the following numerical values: \( A = 0.3820 \) and \( K = 0.6180 \). I simulated data from these processes (initial conditions of 0 for both \( x_t \) and \( x_{t|t} \)) and came up with the forecast of the state at each date. The actual series and forecasted series are plotted below:

![Time series plot](image)

It’s pretty clear from inspection that the filter does a good job. In fact, the root mean squared error (i.e. the square root of the average squared discrepancy) is 0.7623.

To make things clear, let’s suppose that the noise variance quadruples to 4. My new coefficients for the forecasting rule are \( A = 0.6096 \) and \( B = 0.3904 \). This makes sense – the signal is now nosier, so I place less weight on it and more weight on my past forecasts. The time series plot with these parameters is:
It’s pretty clear that the forecasts are worse. In fact, the RMSE here is 2.1112, or about triple what it was above.

**A Multivariate Example**

Now consider a multivariate example. Suppose we have a two variable state evolving according to:

\[
\begin{bmatrix}
  x_{1,t} \\
  x_{2,t}
\end{bmatrix} = \begin{bmatrix}
  1 & 0 \\
  0.3 & 0.5
\end{bmatrix} \begin{bmatrix}
  x_{1,t-1} \\
  x_{2,t-1}
\end{bmatrix} + \begin{bmatrix}
  e_{1,t} \\
  e_{2,t}
\end{bmatrix}
\]

Here \( x_{1,t} \) evolves according to a univariate random walk. \( x_{2,t} \) depends positively on lags of \( x_{1,t} \) and positively on its own lag. There are shocks to both. I assume that the variance-covariance matrix of shocks is:

\[
\Sigma_e = \begin{bmatrix}
  \sigma_1^2 & 0 \\
  0 & \sigma_2^2
\end{bmatrix}
\]

I assume that there is one variable that can be observed. It obeys:

\[
y_t = \begin{bmatrix}
  1 & 1
\end{bmatrix} \begin{bmatrix}
  x_{1,t} \\
  x_{2,t}
\end{bmatrix} + u_t
\]

In essence, you can observe the sum of the states plus a noise term. The variance of the noise is \( \sigma_u^2 \). I assume that all variance terms are unity. I generate data (using the same seed) as above and come up with forecasts of the state. Here are the time series plots of \( x_{1,t} \) (left) and \( x_{2,t} \) (right):
We again see that the forecasts are pretty good. The second variable inherits the basic shape/trend from the random walk variable, but jumps around a bit more. The RMSEs for each, respectively, are: 0.9445 and 0.9265. Again, naturally, if I increase the noise variance, I increase the RMSEs.

Note that, in a multivariate context, I need not actually have any noise in the observation, and some of the observed variables could in fact be states themselves. In particular, modify the above to be:

\[ y_t = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} 0 \]

In other words, \( y_t = x_{2,t} \) without noise. The process for the state remains the same. In essence then, the filtering problem becomes this: how can I best forecast \( x_{1,t} \) given that I observe \( x_{2,t} \) and know the joint stochastic process for them? Well, it’s easy – just plug the new numbers in. Below are the simulated and forecasted time paths of \( x_{1,t} \) and \( x_{2,t} \):
Note, as we would expect, the forecast of $x_{2,t}$ is the actual value – it isn’t observed with noise, so the RMSE is 0. Perhaps interestingly, the forecast of $x_{1,t}$ is now worse than before (RMSE of 1.8267) – that’s because the observed variable now contains less information about it than it did in the process before.

**Kalman Smoothing**

The normal Kalman filter as described above is a forward filter in the sense that it makes forecast of the current state using only current and past observed $y$s. However, if one is interested in doing a retrospective, there is more information available on the state at any particular date $t$ using observed values not only from $t$ and below but also making use of variables dated later than $t$. The basic idea of Kalman smoothing is that we can do better than normal Kalman filtering if we want to look at states retrospectively as opposed to in real time.

I’m going to use what is called a fixed interval smoother. It consists of two steps – a forward pass that is the normal Kalman filter, and a backward pass. In the forward pass, we form estimates of the Kalman filter in the usual way described above:

$$x_{t|t} = Ax_{t-1|t-1} + Ky_t$$

$$x_{t|t-1} = Cx_{t|t}$$

The backward pass of the filter proceeds as follows. First, define the following two coefficients:

$$\tilde{C} = C^{-1}(I - \Sigma_e P_{t+1|t}^{-1})$$

$$\tilde{K} = C^{-1}\Sigma_e P_{t+1|t}^{-1}$$

Suppose our data set has $T$ observations running from $t = 1, ..., T$. The backward pass algorithm is as follows, for $k = 1, ..., T - 1$ (i.e. $k < T$):
In other words, your estimate of a past state is a linear combination of your estimate of the past state one period forward using all information and your estimate of the past state one period forward using only information prior to that period.

To operationalize this, begin in period $T$ (i.e. the last observed period of time). Pretty clearly, $x_{T|T}$ will be the “normal” Kalman filter estimate from above (there is no future information to take advantage of). Then $x_{T-1|T}$ will be:

$$x_{T-1|T} = \tilde{C} x_{T|T} + \tilde{K} x_{T|T-1}$$

The $x_{T|T-1} = C x_{T-1|T-1}$ from the normal Kalman filter. Now go back one more period:

$$x_{T-2|T} = \tilde{C} x_{T-1|T} + \tilde{K} x_{T-1|T-2}$$
$$x_{T-2|T} = \tilde{C} x_{T-1|T} + \tilde{K} C x_{T-2|T-2}$$

So, to implement this, we first do the normal Kalman filter, which will give us the $x_{k+1|k}$ values for $k = 1, ..., T - 1$. Then we start by working backwards, recursively finding the $x_{k+1|T}$. At the end of the day, we will have a retrospective estimate of the state. Given that we’re employing more information, this retrospective view ought to be superior to the normal “real time” Kalman filter estimates.

Below is a plot of the actual series vs. the Kalman smoothed series of $x_{1,t}$ from the second version of the multivariate process described above (i.e. $x_{2,t}$ is observed perfectly every period).
This plot is “smoother” than the one using the “real time” Kalman filter from above. The RMSE here is 1.5272, which, as predicted, is less than under the normal real time filter. The reason why the resulting series is “smoother” is because, in real time, when we observe $x_{2,t}$ move, we really aren’t sure whether it was because of a shock to $x_{1,t}$ or $x_{2,t}$. Hence, our forecast of $x_{1,t}$ reacts to shocks to $x_{2,t}$, even though the actual $x_{1,t}$ process does not. With the benefit of hindsight (i.e. observing future observations), we can eliminate some of that, so the resulting estimate is smoother than the normal Kalman filter (and better too).