Kalman Filters

In 1960 Rudy Kalman published a paper that showed how to optimally track linear systems where errors are present in the model and in the measurements. It is a process that can easily be done in real time. NASA used his work in the Ranger, Mariner, and Apollo programs in the 1960's. Kalman filters answer the question "How do you update a 'best' estimate for the state of a linear system, as new, but still inaccurate, data pours in?"

By following this outline you will derive the Kalman filter in one dimension and for normally distributed errors, and hopefully in the process, understand what the Kalman filter is trying to do. You start with a linear system having a random error vector, $\boldsymbol{w_K}$ with zero mean and covariance matrix $\boldsymbol{Q}$. The state equation $\boldsymbol{x_{k+1}}=\boldsymbol{A}\boldsymbol{x_k}+\boldsymbol{B}\boldsymbol{u_k}+\boldsymbol{w_k}$ works well for this.

Covariance between two random variables x_i and x_j is an expected value given by:

$cov(x_i,x_j)=E[(x_i-\mu_i)(x_j-\mu_j)]$ where $\mu_i=E[x_i]$.

The covariance matrix, **Q**, is:

$E[\boldsymbol{w_k}\boldsymbol{w_k^T}]=\int_{-\infty}^{\infty}(\boldsymbol{w_k}-\boldsymbol{\mu_k})(\boldsymbol{w_k^T}-\boldsymbol{\mu_k^T})f_w(\boldsymbol{w_k})d\boldsymbol{w_k}=\int_{-\infty}^{\infty}\boldsymbol{w}\boldsymbol{w^T}f_w(\boldsymbol{w})d\boldsymbol{w}$ where the last equality is because we assumed that the error distribution is stationary (it is the same for all k) and because it has zero mean. The joint probability density function, $f_w(\boldsymbol{w})$, is a function of n random variables, w_i, the components of $\boldsymbol{w}$, if the state vector, $\boldsymbol{x_k}$ has n states. Note that for a one state system, $\boldsymbol{Q}$ is just the variance. $\boldsymbol{Q}=Q=E[(x-\mu)^2]=\sigma^2$

The measurements, $\boldsymbol{y_k}$, are likewise corrupted by errors, $\boldsymbol{v_k}$, with zero means and covariance matrix, $\boldsymbol{R}$. The output from our linear system is then given by $\boldsymbol{y_k}=\boldsymbol{C}\boldsymbol{x_k}+\boldsymbol{v_k}$.

The idea is to come up with a recursive method of improving the estimate of $\boldsymbol{x_k}$ that gets better with each measurement we get. In order to do that, Kalman did a prediction/correction process. Let $\boldsymbol{\tilde{x}_k}$ be the predicted state estimate, and $\boldsymbol{\hat{x}_k}$ be the corrected state estimate (calculated after the output $\boldsymbol{y_k}$ is measured). The way we will correct the state is by letting the corrected estimate of the state be a combination of the prediction, and a weighted difference between the predicted measurement, $\boldsymbol{\tilde{y}_k}=\boldsymbol{C}\boldsymbol{\tilde{x}_k}$ and the actual one, $\boldsymbol{y_k}$. This difference, $(\boldsymbol{y_k}-\boldsymbol{\tilde{y}_k})$, is known as the residual or inference. The corrected estimate of the state is then: $\boldsymbol{\hat{x}_k}=\boldsymbol{\tilde{x}_k}+\boldsymbol{K}(\boldsymbol{y_k}-\boldsymbol{\tilde{y}_k})$. Kalman then needed to find the best $\boldsymbol{K}$ in some sense. What Kalman did was to minimize the variance of the error of his estimate, $E[(\boldsymbol{x_k}-\boldsymbol{\hat{x}_k})^2]$. Below, you will follow in Kalman's steps for a single state system, like an RC circuit.

1. For a single state system show that $\boldsymbol{x_k}-\boldsymbol{\hat{x}_k}=(1-KC)(x_k-\tilde{x}_k)-Kv_k$ where you can see that all the vectors and matrices are 1x1, and are indicated that way by non-bold type.
2. Show (justifying all steps) for this case, that $E[(\boldsymbol{x_k}-\boldsymbol{\hat{x}_k})^2]=(1-KC)^2E[(x_k-\tilde{x}_k)^2]+K^2R$.[1]
3. Minimize the result of 2, above, to find the optimum K for each time step, k in terms of $E[(x_k-\tilde{x}_k)^2]$, C, and R.
4. In order to finish finding K, you need to show $E[(x_k-\tilde{x}_k)^2]=A^2E[(x_{k-1}-\hat{x}_{k-1})^2]+Q$.[2]
5. Now draw a flow chart for the algorithm.
6. Using Google, or any other method you like, find the generalizations of the equations you developed above for n=1, that work for a general $n\geq 1$.
7. Make a flow chart for a Kalman Filter. The objective of this is to understand the process of Kalman filtering better, so that you can write some software that implements a Kalman Filter.
8. Make a MATLAB/octave script to demonstrate Kalman filtering on an RC circuit.
9. Bonus: If you can come up with a good practical way to estimate **R** and **Q** as your filter runs, so that each step improves your estimate of **Q** and **R**, I will be grateful, and promise to remember your contribution when I'm working on the evaluations necessary to come up with meaningful grades for this class.

---

1 Hint: v_k and x_k are uncorrelated. $E[x_kv_k]=E[x_k]E[v_k]$

2 Hint: w_k and x_k are uncorrelated. $E[x_kw_k]=E[x_k]E[w_k]$