**Lecture 1 The Kalman Filtering Algorithm**

## AN OVERVIEW OF KALMAN FILTERING

### I. The Dynamical System Model

Let $z_k = [z_{k1}, \ldots, z_{k\,m}]^{tr}$ and let $x_k = [x_{k1}, \ldots, x_{k\,n}]^{tr}$ . Assume the relationship between these random processes is:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \tag{1a}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{1b}$$

where (1a) includes the known and non-random ($n \times n$ **matrix-valued**) parameter $F_k$, as well as the $n$-D non-measurable ***white noise*** **random process** $w_k$ with

$$E(\mathbf{w}_k \mathbf{w}_{k+j}^{tr}) = \mathbf{Q}_k \bullet \delta(j) \text{ and } E(\mathbf{v}_k \mathbf{v}_{k+j}^{tr}) = \mathbf{R}_k \bullet \delta(j)\,. \tag{2}$$

Equations (1) are discrete-time equations. Often they are arrived at by ***sampling*** the associated differential equations.

**Challenge #1:** To obtain a physically meaningful dynamical model for the state process $\boldsymbol{x(t)}$.

**Challenge #2:** To sample the state process $\boldsymbol{x(t)}$ in such a manner, so as to not distort (e.g. alias) the information in $\mathbf{x}_k$.

**Challenge #3:** To accurately characterize the state white noise process $\mathbf{w}_k$ covariance matrix $\mathbf{Q}_k$.

### II. The Kalman Filter Algorithm

From Figure 5.8 on p.219 of the book, we have the following KF algorithm:

***Step 1*:** For $k$=0 carefully choose a value for $\hat{\mathbf{x}}_0^-$ and compute the prediction error variance $\mathbf{P}_0^- = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^-)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^-)^{tr}]$.
***This is the state predictor prior to obtaining the current measurement $z_k$ in (1b).***

***Step 2*:** For $k$=0: Compute the Kalman gain: $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^{tr} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^{tr} + \mathbf{I})^{-1}$.

***Step 3*:** Compute the update estimate: $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$. ***This is the state predictor after obtaining the current measurement $z_k$ in (1b).***

***Step 4*:** Compute the update error variance: $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-$ .

***Step 5*:** Compute 1-step prediction and associated error variance: $\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}_k \hat{\mathbf{x}}_k$ ; $\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^{tr} + \mathbf{I}$

***Step 6*:** Go to *Step 2* and increment $k$ by one.

**Remark #1:** Recall that we only have measurements of $\mathbf{z}_k$ Hence, the KF is an algorithm that allows us to obtain an estimate, $\hat{\mathbf{x}}_k$, of the non-measurable state $\mathbf{x}_k$. In the jargon of electrical engineering, we are using $\mathbf{z}_k$ to '*filter* out' an estimate of $\mathbf{x}_k$.

**Remark #2:** Once we have $\hat{\mathbf{x}}_k$, *Step* 5 above provides us with a *prediction*, $\hat{\mathbf{x}}_{k+1}$, of unknown $\mathbf{x}_{k+1}$.

**Example** APPLICATION TO A TIME-VARYING AR(1) PROCESS

In this example we will mainly focus on the problem of detecting changes related to a time-varying first order autoregressive, AR(1), measurement process:

$$z_k = a_k z_{k-1} + v_k \quad ; \quad E(v_k^2) = \sigma_v^2(k) \tag{3a}$$

In the special case where the AR(1) parameter $a_k$ and the white noise variance $\sigma_v^2(k)$ do not depend on the time index, $k$, the process (3) is a *wss* random process as long as the condition $|a| < 1$ holds.

In the case where the AR(1) parameter $a_k$ changes slowly in relation to the sampling interval, the process (3a) is a locally *wss* process, and it is easy to show that the driving white noise is nonstationary with variance

$$\sigma_v^2(k) \cong (1 - a_k^2)\sigma_z^2(k). \tag{3b}$$

The model (3b) has the ability to capture the slow time variation in both the process power, and time-varying frequency content can be captured by the AR(1) parameter, $a_k$.

In relation to the process (3), suppose that the AR(1) parameter is changing in the manner described in Figure 1.

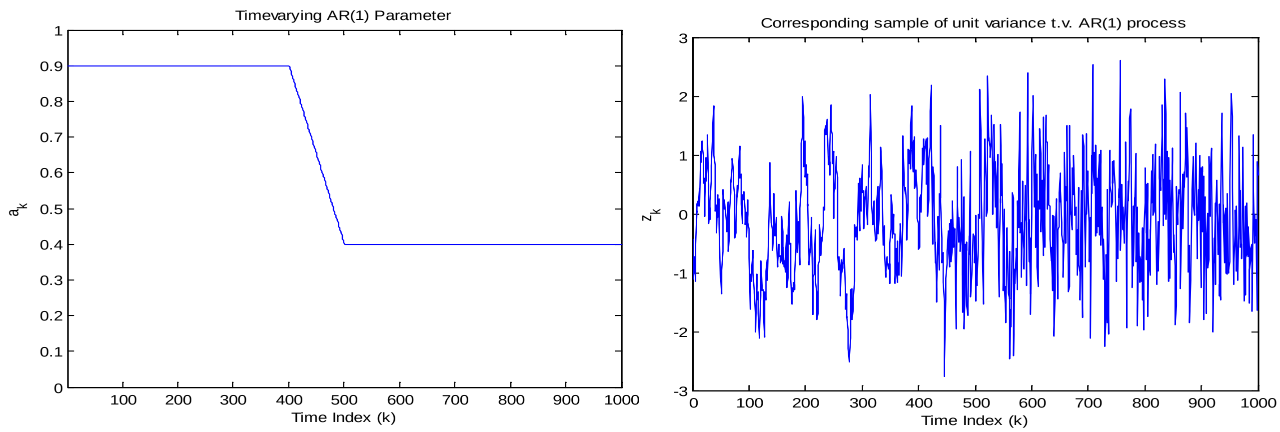


**Figure 1.** Time-varying AR(1) parameter, $a_k$ (left), and partial realization of $z_k$ (right).

The process power is held constant, while the dynamics go from slow to fast over time. As shown in Figure 1, we see that the change from a *wss* process with *a=0.9* to one with *a=0.4* does not occur instantaneously. Rather, there is a transition region (400-500) where the parameter value decreases linearly.

**QUESTION:** What can you say about the general structure of the partial realization in Figure 1?

**ANSWER:** 

____________________________________________________

To implement a KF we will use the following state model:

$$a_k = a_{k-1} + w_k \quad \text{with} \quad Q_k = \sigma_w^2 \qquad \text{(random walk model, also called the total ignorance model)} \qquad (4a)$$

We will assume that the measurement model is:

$$z_k = a_k z_{k-1} + v_k \quad \text{with} \quad \sigma_v^2(k) = 1 - a_k^2 \qquad (4b)$$

Hence, the quantities in the model (2) become: $x_k = a_k$; $F_k = 1$; $Q_k = \sigma_w^2$; $H_k = z_{k-1}$; $R_k = 1 - \hat{a}_k^2$.

There are two items here that make this KF suboptimal in the sense of *minimizing the mean squared error between the state process $x_k$ and any estimator of it*:

**1.** The parameter $H_k = z_{k-1}$ is not a non-random quantity. However, because at time $k$ we have knowledge of $z_{k-1}$, it is *conditionally* non-random. In this sense, the KF is known as an *extended* KF.

**2.** We do not have knowledge of $R_k = \sigma_v^2(k)$, which is the driving white noise for the process (3). However, we do know that the process power (3b) is constant and known. Hence, with our estimate $\hat{a}_k$ we can estimate this white noise variance as $\hat{\sigma}_v^2(k) = \max\{0\,,\, \sigma_z^2(1-\hat{a}_k^2)\}$. [You will eventually be able to show this, yourself ☺ ]

**The most difficult quantity to specify is the state noise variance $Q_k = \sigma_w^2$:**

If we set this value to zero, then we are forcing the AR parameter $a_k$ to remain constant and equal to the initial condition specified (in this case, 0.9). If we make this variance to large, then the KF will track more rapid changes in the AR parameter, but with more variability. A similar trade-off occurs in **sliding window types of analyses**. If the window is short, more rapid changes can be tracked, but with higher variability. A longer window leads to slower response to changes, but with less variability.

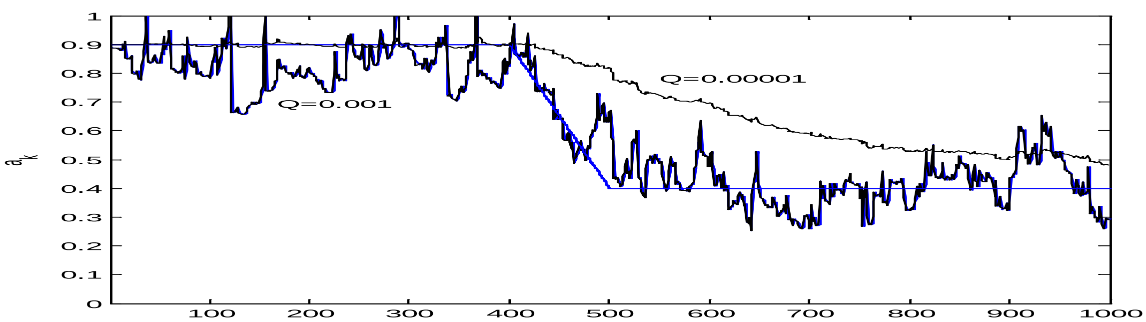


**Figure 3.** KF estimates of $a_k$ for $Q_k = \sigma_w^2$: 0.00001 and 0.001.

The trade-off between bias and variability is clear. What is not clear is how this translates into change detection. If we define change as change in relation to the value 0.9, then Figure 3 would include numerous false detections for a threshold value of, say, 0.8. If we desire to detect whether $a_k$ is 0.9 or 0.4, with say a threshold value of 0.65, then the only incorrect detections occur in the transition region.

**QUESTION**: What is the implication of the random walk (or, ignorance) model (4a)?

**ANSWER**: Since the sampling interval is $\Delta = 1$, we have $\sigma_{a_k} = \sigma_w \sqrt{k}$. [You will prove this in due course. ☺ ] In words, we are assuming that as time, *k*, progresses, we know less and less about $a_k$. We know the initial condition, $a_0 = 0.9$. However, after, say, $k = 400\,\text{sec.}$, we are assuming $\sigma_{a_k} = 20\sigma_w$. For $\sigma_w = \sqrt{0.0002} = 0.014$, the $\pm 4\sigma_w$ uncertainty for $a_{400}$ is $\pm 0.056$. Even so, by using the measurements $\{z_k\}_{k=1}^{400}$, we continue to have a pretty good estimate $\hat{a}_{400}$. We are constantly updating this estimate, even as a sliding window would do.

**QUESTION**: How would I find the optimal value for $\sigma_w$?

**ANSWER**: In previous times this was an extremely difficult thing to do. Nowadays, we can run simulations to find it. These will also give us an idea of how sensitive the KF performance is in relation to departures away from $\sigma_w$. If the performance is robust in this respect, then we can be comforted by the fact that we need not have a precise knowledge of $\sigma_w$.

**QUESTION**: What if I have a better model for $a_k$? For example, what if I know that in the good condition $a_k$ follows the more realistic model:

$a_k = a_{good} + da_k$ where $da_k = \alpha\, da_{k-1} + w_k$. Then how would I incorporate this model? And would it perform any better?

**ANSWER**: Recall the general state model: $\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k$. We would need to extend this model to accommodate the constant $a_{good}$. For example:

$$\begin{bmatrix} da_{k+1} \\ a_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} \alpha_a & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & a_k \end{bmatrix} \begin{bmatrix} da_k \\ da_k \\ z_k \end{bmatrix} + \begin{bmatrix} w_{k+1} \\ 0 \\ v_{k+1} \end{bmatrix} + \begin{bmatrix} 0 \\ a_{good} \\ 0 \end{bmatrix}. \quad (5)$$

The rightmost term in (5) is a deterministic ‘input’ to the system. We will address how to modify the KF equations to accommodate this input in due course.

**QUESTION**: But then what happens if/when $a_{good}$ changes to $a_{bad}$?

**ANSWER**: Good question! First we would need to reliably detect a change from $a_{good}$. If we know that it changes to a known $a_{bad}$, then we could change the middle state input (5) to $a_{bad}$. Alternatively, we could run two KFs simultaneously, and choose the one that is more likely to be the correct one. This method is known as *multiple model adaptive estimation* (MMAE). We will also cover this topic in due course.

**CONCLUSION**: It is reasonable to speculate that the more we know, the better the KF will do. However, this knowledge has a price. We need to know more concepts to incorporate this knowledge.

**Matlab Code** This code is related to the code used to obtain Figure 1. It is not the exact code.

```
% PROGRAM NAME: tvaAR1KF.m     10/24/17
% Track a t.v. AR(1) Parameter
clear all
close all
%================================
% Construct t.v. AR(1) parameter
r1=400; %Length of a1
A1=0.9; %Values for a1
a1=A1*ones(1,r1); %Values for a1
r2=200; %Length of ramp reduction in a
A3=0.6; %Values for a3
dA=A1-A3; %Total drop in A
dr=dA/r2;
ramp=0:dr:dr*r2;
 a2= A1- ramp;
 a3=A3*ones(1,r1+r2-length(a2));
 a=[a1 a2 a3];
 % Compute driving noise variance for unity variance AR(1)
 sige2=(1 - a.^2);
 % Generate t.v. AR(1) Measurement Process
 z(1)=randn(1,1);
 for t=2:1000
    z(t)=a(t)*z(t-1) + sige2(t)^(0.5)*randn(1,1);
 end
%=========================================================
% Compute the KF estimate of a t.v. AR(1) parameter
kmax=length(z);
F=1;% Random walk model: a(k)=a(k-1) + u(k)
Q=0.0002; %Assumed Var(u) $$$ PLAY WITH THIS $$$
% INITIAL CONDITIONS:
xhat_old = 0.9;
P_old = Q;
%--------------------
I=1;
K=zeros(1,kmax); xhat=0.9*ones(1,kmax);
for k=2:kmax
   H=z(k-1);
   R=max([0,(1 - a(k)^2)]); %Estimate of the white noise variance for var(z)=1.0

   Kk=P_old*H'*(H*P_old*H' + R)^(-1);
   K(k)=Kk;
   xhat(k)=xhat_old + Kk*(z(k) - H*xhat_old);
   Pk=(I - Kk*H)*P_old;
   xhat_old=F*xhat(k);
   P_old=F*Pk*F' + Q;
end
%=================================
%Various Plots:
ahat=xhat;
figure(1)
 plot(a)
 %axis([1,1000,0,1])
hold on
plot(ahat)
legend('a','ahat')
title(['Time-varying AR(1) Parameter & KF Estimate for Q= ',num2str(Q),' '])
grid
%------------
BW=-log(a)/(2*pi); BWhat=-log(ahat)/(2*pi);
figure(2)
 plot(BW)
 %axis([1,1000,0,1])
hold on
```

```
plot(BWhat)
legend('BW','BWhat')
title(['Time-varying BW & KF Estimate for Q= ',num2str(Q),' '])
xlabel('Time (sec)')
ylabel('Hz')
grid
%------------
figure(3)
 plot(z)
 title('Realization of Unit Variance t.v. AR(1) Process')
 grid
```