Lecture 11 **Tracking an AR(2) Signal Corrupted by AR(1) Noise (11/1/19)**

It might seem strange that we would address the topic of Kalman Filtering (KF) prior to having an understanding of how it was derived. We will go through that derivation. It is my belief that most students would be far more motivated to go through it, were they first to appreciate what it is, and how valuable it can be. In this lecture we will attempt such motivation.

The following equations define what is meant as a *dynamical system*:

The *state* equation: $$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k \ . \tag{1a}$$

The *measurement* equation: $$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{D}_k \mathbf{v}_k \ . \tag{1b}$$

In (1a) the variable $\mathbf{x}_k = [x_k(2), x_k(2), \cdots x_k(n)]^{tr}$ is called the system *state* at time index *k*. [Note that the index can also be a spatial index.] The state evolves per (1a). In (1b) the variable $\mathbf{z}_k = [z_k(2), z_k(2), \cdots z_k(m)]^{tr}$ is called the *measurement* (or *observation*) at time index *k*. The variable $\mathbf{w}_k = [w_k(1), w_k(2), \cdots w_k(p)]^{tr}$ is called the *state driving white noise*. It can be viewed as the *input* to a system having an output $\mathbf{x}_k$. The variable $\mathbf{v}_k = [v_k(1), v_k(2), \cdots v_k(q)]^{tr}$ is called the *measurement white noise*. Notice that, unlike $\mathbf{w}_k$, which drives the state $\mathbf{x}_k$ to the next state $\mathbf{x}_{k+1}$, the measurement noise $\mathbf{v}_k$ does not drive the measurement $\mathbf{z}_k$ anywhere. It is simply additive measurement noise

Equations (1) do not define the KF. Rather, they simply model a dynamical system. Our motivation for KF will proceed by addressing what is called the *signal-plus-noise* problem.

Consider a **measurement process** $z_t = s_t + n_t$ that is the sum of a signal that is statistically independent of the noise, and where the signal and noise processes are:

Signal**:** The signal, s_t is an AR(2) process $s_t = -a_1 s_{t-1} - a_2 s_{t-2} + u_t$
Noise**:** The noise, n_t is an AR(1) process: $n_t = -b_1 n_{t-1} + v_t$

<u>Step 1</u>: Obtain the state and measurement equations.

The state and measurement equations needed to implement a Kalman Filter to estimate the signal are obtained as follows:

Let $\mathbf{s}_t = \begin{bmatrix} s_t & s_{t-1} \end{bmatrix}^{tr}$. We then have: $$\mathbf{s}_{t+1} = \begin{bmatrix} -a_1 & -a_2 \\ 1 & 0 \end{bmatrix} \mathbf{s}_t + \begin{bmatrix} u_{t+1} \\ 0 \end{bmatrix} \Rightarrow \mathbf{s}_{t+1} = \mathbf{F}_s \mathbf{s}_t + \mathbf{w}_t^{(s)} \quad ; \quad \mathbf{Q}_s = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & 0 \end{bmatrix}. \tag{2a}$$

We also have: $$n_{t+1} = -b_1 n_t + v_t = \mathrm{F}_n n_t + w_{t+1}^{(n)} \quad ; \quad \mathrm{Q}_n = \sigma_v^2 \ . \tag{2b}$$

Equations (2) can be expressed directly as:

$$\begin{bmatrix} s_{t+1} \\ s_t \\ n_{t+1} \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -b_1 \end{bmatrix} \begin{bmatrix} s_t \\ s_{t-1} \\ n_t \end{bmatrix} + \begin{bmatrix} u_{t+1} \\ 0 \\ v_{t+1} \end{bmatrix} \Rightarrow \mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{w}_t \quad ; \quad \mathbf{Q} = \begin{bmatrix} \sigma_u^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 \end{bmatrix} \tag{3a}$$

They can also be expressed as:

$$\begin{bmatrix} \mathbf{s}_{t+1} \\ n_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^{(s)} & 0 \\ \mathbf{0} & \mathrm{F}^{(n)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_t \\ n_t \end{bmatrix} + \begin{bmatrix} \mathbf{w}_t^{(s)} \\ \mathrm{w}_t^{(n)} \end{bmatrix} \Rightarrow \mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{w}_t \quad ; \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_s & 0 \\ \mathbf{0} & \mathrm{Q}_n \end{bmatrix}. \tag{3b}$$

The advantage of (3b) is that it highlights the composition of the signal and noise states.

The measurement process $z_t = s_t + n_t$ is written as:

$$z_t = \mathbf{H}\mathbf{x}_t = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}\mathbf{x}_t \tag{4}$$

Hence, the measurement white noise covariance matrix is $\mathrm{R} = 0$.

<u>Step 2</u>: Obtain the initial prediction error covariance matrix.

If we choose the state initial condition to be $\hat{\mathbf{x}}_0^- = [0\ 0\ 0]^{tr}$, then the associated state prediction error is $\mathbf{e}_0^- = \mathbf{x}_0 - \hat{\mathbf{x}}_0^- = \mathbf{x}_0 - [0\ 0\ 0]^{tr} = \mathbf{x}_0$. The associated prediction error covariance matrix is then:

$$\mathbf{P}_0^- = Cov(\mathbf{e},\mathbf{e}^{tr}) = Cov(\mathbf{x}_0,\mathbf{x}_0^{tr}) = \begin{bmatrix} R_s(0) & R_s(1) & 0 \\ R_s(1) & R_s(0) & 0 \\ 0 & 0 & R_n(0) \end{bmatrix} \tag{5}$$

We are now in a position to implement a KF to estimate the *state* from the *measurement*.

***Numerical Example- Signal*:** $s_t = 0.5562 s_{t-1} - 0.81 s_{t-2} + u_t$; $R_s(0) = 1.0$. ***Noise*:** $n_t = 0.7 n_{t-1} + v_t$; $R_n(0) = 1.0$
Note that the SNR is 0 dB. The PSD plot in Figure 1 shows that even though SNR=0dB, the local SNR in the peak region of the signal PSD is ~13dB. The sample *mse* is ~0.3.

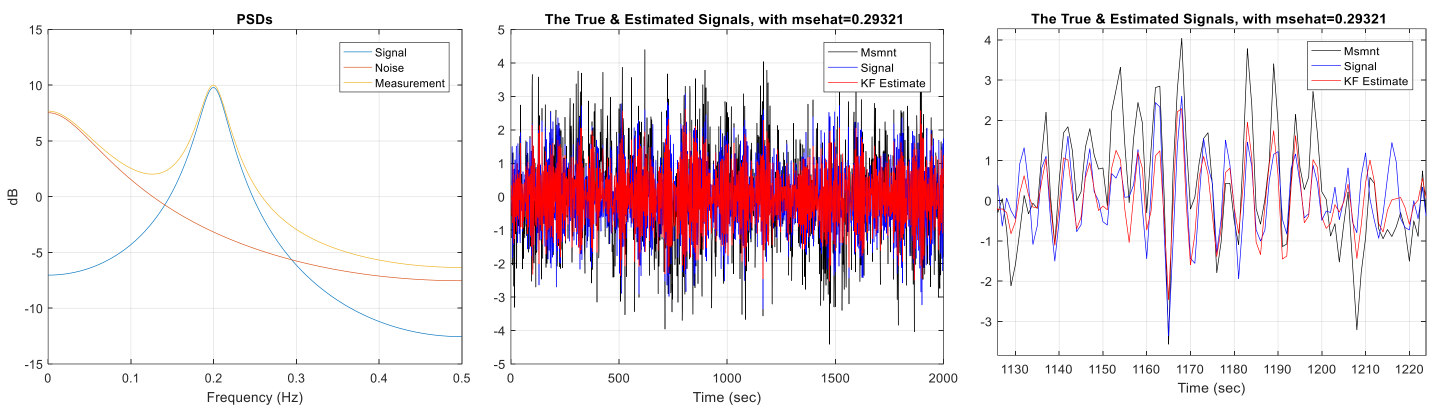

Figure 1 Plots of the PSDs and the prediction results.

From the above plots, we make the following observations:

(O1): The *psd* plot shows that, even though the *signal-to-noise ratio* (*SNR*), which is typically defined as $SNR \overset{\Delta}{=} \sigma_s^2 / \sigma_n^2 = R_s(0)/R_n(0)$, is 1.0 or 0dB, the *effective SNR* that is the *SNR* in the frequency region in which notable signal is present, is much greater than 0dB. Hence, the problem is not a challenging as it might seem. In fact, we could simply use a *band-pass filter* (*BPF*) and get a pretty good estimate of the signal.

(O2): From the KF results, we see that, even though we desire to get only an estimate of the signal, we also get an estimate of the noise- for free ☺.

We will now review some of the elements of the KF code.

Matlab Code:

```
% PROGRAM NAME: AR2plus1.m    10/25/17
% SIGNAL:
```

```
% The following is for Rs0 = 1.0
Rs0 = 1;
a1=-.5562; a2=.81;
%Recover Rs(1) and Var_u
A=[1 -a1 -a2; 0 1+a2 0;0 a1 1];
v=[1 -a1 -a2]';
b=A^-1*v;
varu=b(1);
Rs1 = b(2);
nfft=4096;
Ss=varu*(abs(fft([1,a1,a2],nfft))).^-2;
SsdB=10*log10(Ss(1:nfft/2));
%NOISE
b1 = -0.7;
SNRdB=0; %Specified SNR in dB $$$$$
SNR=10^(SNRdB/10);
varn = 1/SNR; %
varv = varn*(1-b1^2);
Rn0 = varn;
Sn=varv*(abs(fft([1,b1],nfft))).^-2;
SndB=10*log10(Sn(1:nfft/2));
figure(1)
Sx=Ss+Sn;
SxdB=10*log10(Sx(1:nfft/2));
df=1/nfft;
f=0:df:.5-df;
plot(f,[SsdB;SndB;SxdB])
title('PSDs')
legend('Signal','Noise','Measurement')
xlabel('Frequency (Hz)')
ylabel('dB')
grid
%=================================
% DATA GENERATION
u=varu^0.5 * randn(1,2500);
v=varv^0.5 * randn(1,2500);
n = zeros(1,2500);
s = zeros(1,2500);
z = zeros(1,2500);
for k = 3:2500
    s(k) = -a1*s(k-1) - a2*s(k-2) + u(k);
    n(k) = -b1*n(k-1) + v(k);
end
s = s(501:2500);
n = n(501:2500);
z = s + n;
%=====================================
% DEFINE KF MODEL MATRICES
% TO BE USED BY kfwss.m below
F = [-a1 -a2 0;1 0 0 ;0 0 -b1];
Q = [varu 0 0;0 0 0;0 0 varv];
H = [1 0 1];
R = 0;
xhat_old = [0;0;0];
P_old = [Rs0 Rs1 0;Rs1 Rs0 0;0 0 Rn0];
% This portion computes the Kalman filter state estimate
% for wss state and measurement processes.
% MODEL:
% x(k) = F x(k-1)  +  w(k-1)
% z(k) = H x(k) + v(k)
I=eye(3);
K=[];
xhat=[];
for k=1:2000
Kk= P_old*H'*(H*P_old*H' + R)^(-1);
K=[K,Kk];
xhatk=xhat_old + Kk*(z(k)-H*xhat_old);
xhat=[xhat,xhatk];
Pk=(I-Kk*H)*P_old;
xhat_old=F*xhatk;
P_old=F*Pk*F' + Q;
end
% ======================================
figure(2)
tvec = 1:2000;
plot(tvec,z,'k',tvec,s,'b')
title('Sample of the Measurement & Signal')
legend('Measurement','Signal')
xlabel('Time (sec)')
```

```
grid
hold on
shat = xhat(1,:);
err = s - shat;
msehat = mean(err.^2);
plot(tvec,shat,'r')
title(['The True & Estimated Signals, with msehat=',num2str(msehat),' '])
legend('Msmnt','Signal','KF Estimate')
mse=mean((s-shat).^2)
```