Homework #5
Due M 3/25

You must turn in your code as well as output files. Please generate a report that contains the code and output in a single readable format.

Visit the book website to download companion software, including all the example problems.

`http://www.wiley.com/WileyCDA/WileyTitle/productCd-1118414322.html`

1. (KLT 6.4)

   Solution

   The error function can be found using the correlation properties of white noise ($r_{xx}(k) = \sigma^2\delta(k) = \delta(k)$) input as

$$\begin{aligned}\xi &= E[d^2(n)] - 2p^T w + w^T R w \\ &= (b_0^2 + b_1^2 + b_2^2) - 2(b_0 w_0 + b_1 w_1 + b_2 w_2) + (w_0^2 + w_1^2 + w_2^2)\end{aligned}$$

   using the the following definitions

$$R = E\left[x(n)x^T(n)\right] \qquad p = E\left[d(n)x(n)\right]$$

$$R = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) \end{bmatrix} \qquad p = \begin{bmatrix} r_{dx}(0) \\ r_{dx}(1) \\ r_{dx}(2) \end{bmatrix}^T$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}^T = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix}^T.$$

   The optimal solution can be found by solving the least squares problem

$$w^o = R^{-1}p = p = \left[\begin{array}{ccc} 0.2 & 0.5 & 0.3 \end{array}\right]^T.$$

   The minimum MSE can be found as

$$\xi_{min} = E[d^2(n)] - p^T w^o = (b_0^2 + b_1^2 + b_2^2) - p^T w^0 = 0.$$

   The results can be verified using Matlab as shown below (Fig. 1).

2. (KLT 6.6)

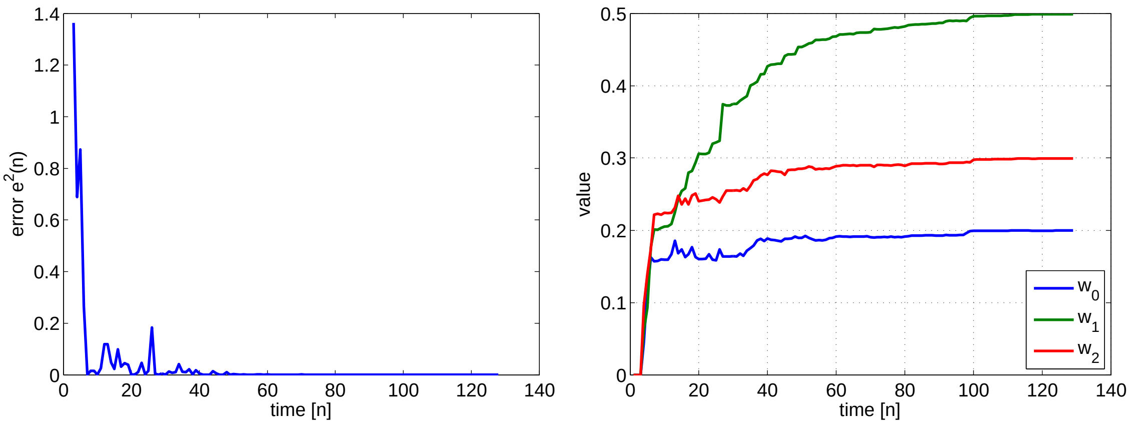
   Solution

   While this problem asks you to compute the correlation using Matlab `xcorr.m` you should know how to compute this by hand. Note that sample correlation is the same as convolution without and flipping.

   (a) $r_{xy}(k) = \begin{bmatrix}0 & 0 & 0 & 3 & 7 & -11 & 14 & 13 & -15 & 28 & 6 & -2 & 21 & 12 & 12 & 6 & 4\end{bmatrix}^T.$

   (b) $r_{xx}(k) = \begin{bmatrix}2 & 10 & 12 & 5 & -1 & 19 & 8 & 15 & 26 & 15 & 8 & 19 & -1 & 5 & 12 & 10 & 2\end{bmatrix}^T.$

   (c) $r_{yy}(k) = \begin{bmatrix}6 & -7 & 16 & 2 & -10 & 35 & -10 & 2 & 16 & -7 & 6\end{bmatrix}^T.$



(a) $e^2(n)$ vs. n

(b) $w(n)$ vs. n

Figure 1: KLT 6.4

3. (KLT 9.10)

   Solution

   The energy estimation is ploted in Fig. 2. Notice these values have been normalized for visualization. The frame length is 256 and the window lengths are $\alpha_s = 1/16$ and $\alpha_l = 1/128$. Please see the end for code.

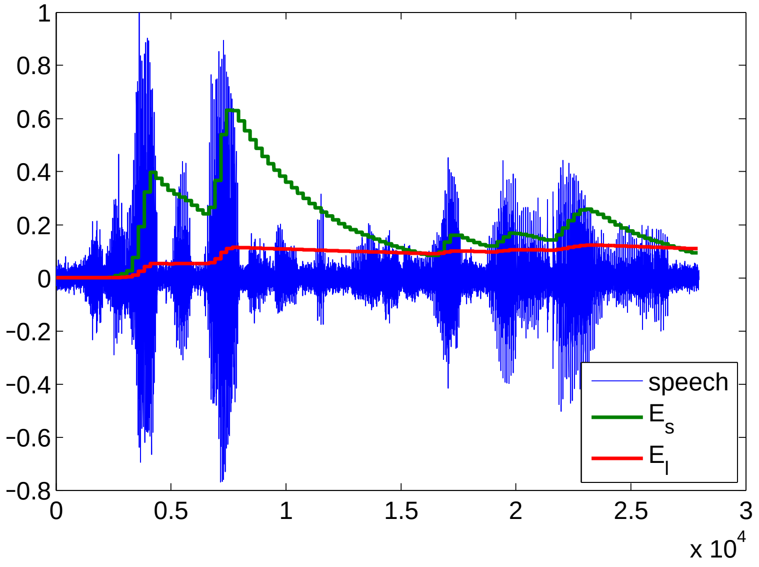


Figure 2: KLT 9.10

4. (KLT 9.11)

   Solution

   The noise floor estimation is ploted in Fig. 3.

5. (KLT 9.12)

   Solution

   The voiced areas is ploted in Fig. 4.

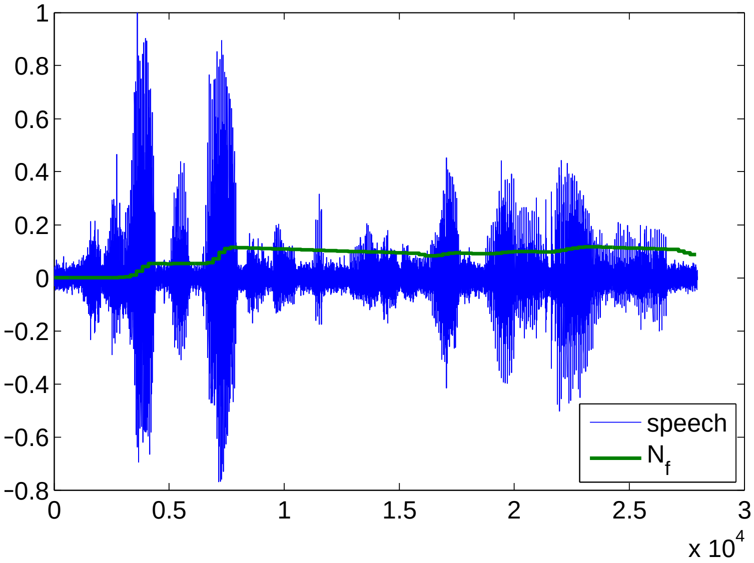


Figure 3: KLT 9.11

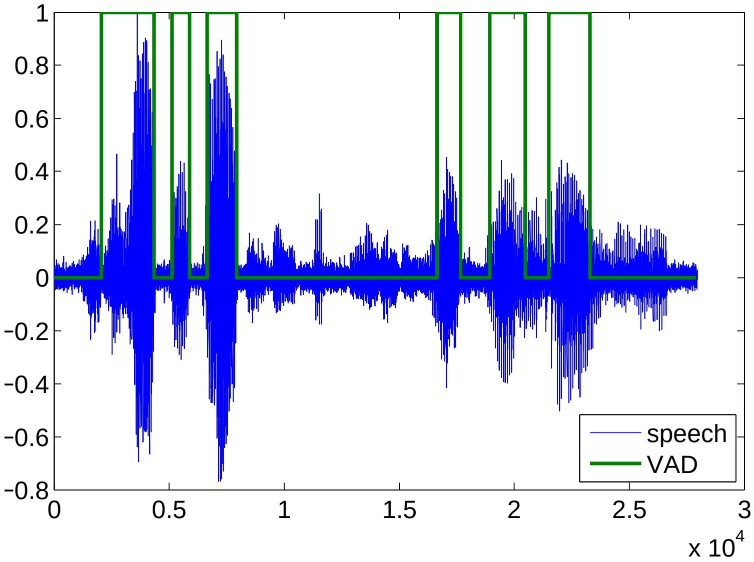


Figure 4: KLT 9.12

6. (KLT 9.13)

Solution

(a) The different window sizes are required by the VAD algorithm to respond quickly to the start of speech (E_s) while providing loudness consistency during a word utterance (E_l).

(b) During the onset of speech, the output of E_s is larger.

(c) During the offset of speech, the output of E_l is larger and takes longer to decay.

Speech Code

```
s = load('TIMIT.ASC');
x = s/max(s) + 0.02*randn(size(s));

scale = 8;
fs = 8000;
win = 2^scale;
alphas = 1/2^(scale-4);
alphal = 1/2^(scale-1);
betal = 5;

%fft params
N = round(length(x)/win);
delf = fs/win;
K1 = find(delf*(0:N-1)>=300,1);
K2 = find(delf*(0:N-1)>1000,1)-1;

VAD = zeros(1,N+1); Es = zeros(1,N+1); El = zeros(1,N+1);
Nf = zeros(1,N+1); En = zeros(1, N+1);
for f=1:N

    %get frame
    it = f*win;
    ib = (f-1)*win+1;
    xc = x(ib:it);

    %compute frame energy
    Xc = fft(xc);
    En(f+1) = sum(abs(Xc(K1:K2)).^2);

    %signal energy
    Es(f+1) = (1-alphas)*Es(f) + alphas*En(f+1);
    El(f+1) = (1-alphal)*El(f) + alphal*En(f+1);

    %noise floor
    if(Nf(f)<Es(f+1))
        Nf(f+1) = (1-alphal)*Nf(f)+alphal*En(f+1);
    else
        Nf(f+1) = (1-alphas)*Nf(f)+alphas*En(f+1);
    end

    %threshold
    Tr(f+1) = Nf(f+1)/(1-alphal) + betal;

    %VAD
    if(En(f+1) > Tr(f+1)), VAD(f+1) = 1; end
end
```

The results of the VAD algorithm can be plotted together (with normalization for visualization) in Fig. 5.

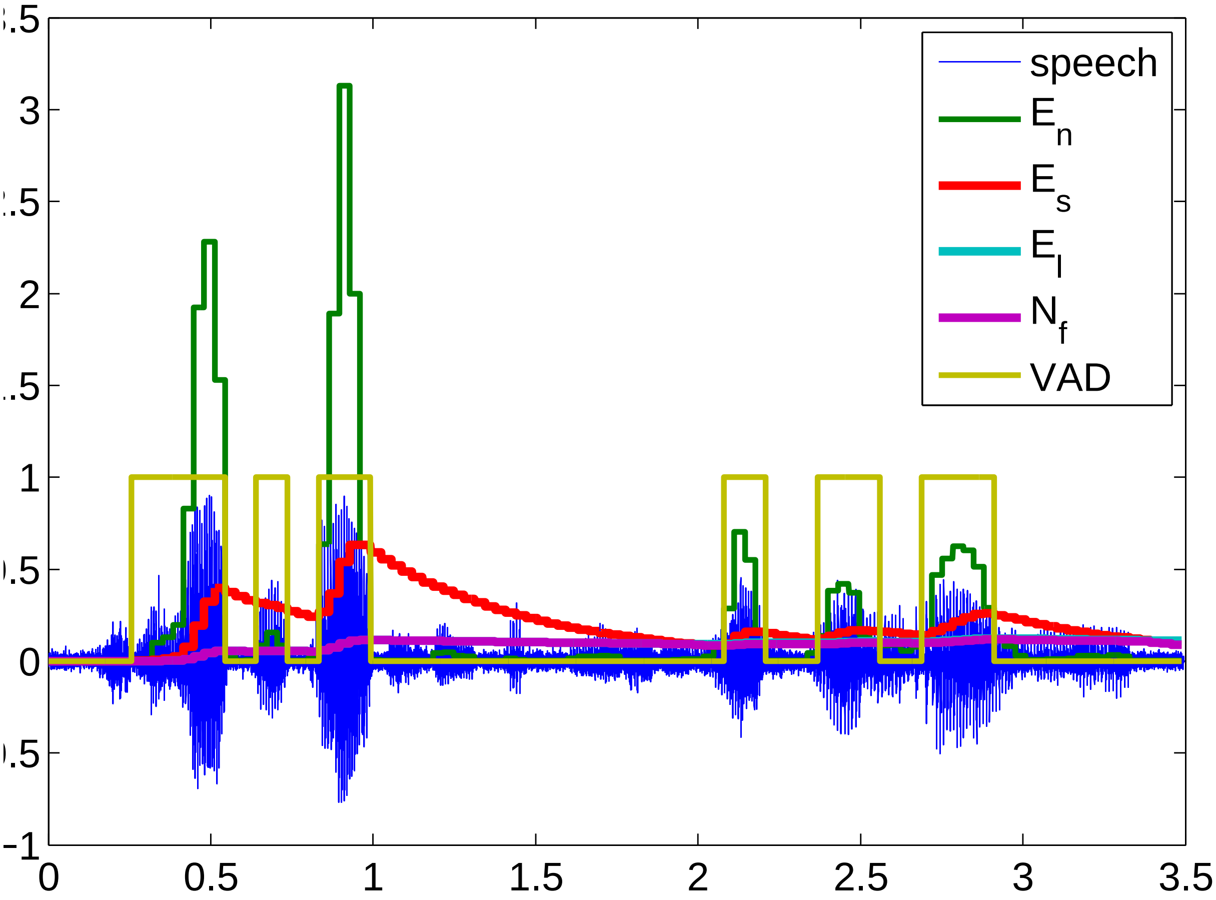


Figure 5: Voice Activity Detection (VAD) Overlay