Adaptive filter

This computer lab will investigate the adaptive filter and how it works in DSP (e.g. How can we remove noise). To better understand this, in the first step, we generate the signal and noise, respectively. We can then analysis their autocorrelation and power spectrum. Given the digital input signal $x(n)$ and $w(n)$:

$$x(n) = A_1 \sin(2\pi f_1 n T_s)$$

where:

- $A_1 = 1$ are the amplitudes, Number of samples $n = 0, \ldots, N-1$, $N = 2000$
- $f_1 = 20$ Hz,
- $T_s = 1/f_s$ is the sampling time in s, $f_s = 2000$
- $w(n)$ is 0 mean with $\sigma = 1$ random noise (Gaussian distribution), $N = 2000$

Complete the following tasks:

1. (a) Generate $x(n)$ in MATLAB.

```
clear all; s_u = 1; s_n = 1; rand('seed',s_u); randn('seed',s_n);
N = 2000;
fs = 2e3; Ts = 1/fs;
n = 0:Ts:(N-1)*Ts;
f = 20;
s = sin(2*pi*f*n);
v1 = randn(1,length(s));
x = s + v1;
figure
subplot(311), plot(n,s), title('signal')
subplot(312), plot(n,v1), title('noise')
subplot(313), plot(n,x), title('signal + noise')
```

2. Task 1: using lowpass filter

```
L = 11;
f1 = f+10;  %cut-off frequency 30
f_sampling = fs;
F1 = f1/f_sampling;
l = -(L-1)/2:(L-1)/2;
h = 2*F1*sinc(2*l*F1);
s_estimate = conv(h,x);
figure, plot(n,s_estimate(1:N)*N/4)
title('Estimated signal after filtering using a lowpass filter')
```

3. Task 2: using your code to perform highpass filter

```
f1 = f-10;  %cut-off frequency 10
```

4. Task 3: using your code to perform bandpass filter

```
f1 = f-10;  %cut-off frequency 10
f2 = f+10;  %cut-off frequency 30
```

5. Task 4: understand the adaptive filter

```
L = 11;
w = [-4 -1 zeros(1,L-2)]';
mu = 0.005;
for i = L+1:length(x)
   y(i) = w(:,i-L)'*v2(i:-1:i-L+1)';
   e(i) = x(i) - y(i);
   w(:,i+1-L) = w(:,i-L) + mu*e(i)*v1(i:-1:i-L+1)';
end
figure,
plot(n,e), title('estimated signal using adaptive noise canceller')
figure
subplot(211), psd(x,length(x),fs), ax=axis; title('Power Spectrum:
signal + noise ')
subplot(212), psd(e,length(e),fs), axis(ax); title('Power Spectrum:
Estimated signal ')

z=w(1:2,:);

%Error Performance Surface/Contour
clear J
npts = 40;
scale = 10;
variance = 1;
H = convmtx([1 0.5],L);
r = H*[1;zeros(L,1)]; r_dx = r(1:2);
R = H*H'; R_xx = R(1:2,1:2);
for w11 = -npts:npts
   for w22 = -npts:npts
      w1 = w11/scale;
      w2 = w22/scale;
      J(w11+npts+1,w22+npts+1) = variance - 2*r_dx.'*[w1;w2] + [w1
w2]*R_xx*[w1;w2] ;
   end
end
figure
mesh(-npts/scale:1/scale:npts/scale,-npts/scale:1/scale:npts/scale,J)
xlabel('w_1'), ylabel('w_0'), zlabel('J = E[|e^2|]')
title('Quadratic Error Performance Surface')

figure
contour(-npts/scale:1/scale:npts/scale,-
npts/scale:1/scale:npts/scale,J)
xlabel('w_1'), ylabel('w_0'), zlabel('J = E[|e^2|]')
title('Quadratic Error Performance Contour')

figure
contour(-npts/scale:1/scale:npts/scale,-
npts/scale:1/scale:npts/scale,J)
xlabel('w_1'), ylabel('w_0'), zlabel('J = E[|e^2|]')
title('Trajectory of the adaptive filter coefficients')
```

```
for n=1:5:length(z)
   hold on, plot(z(2,n),z(1,n),'-+'), hold off
   pause(0.01)
end

figure
contour(-npts/scale:1/scale:npts/scale,-
npts/scale:1/scale:npts/scale,J)
xlabel('w_1'), ylabel('w_0'), zlabel('J = E[|e^2|]')
title('Trajectory of the adaptive filter coefficients')
hold on, plot(z(2,:),z(1,:),'-+'), hold off
```