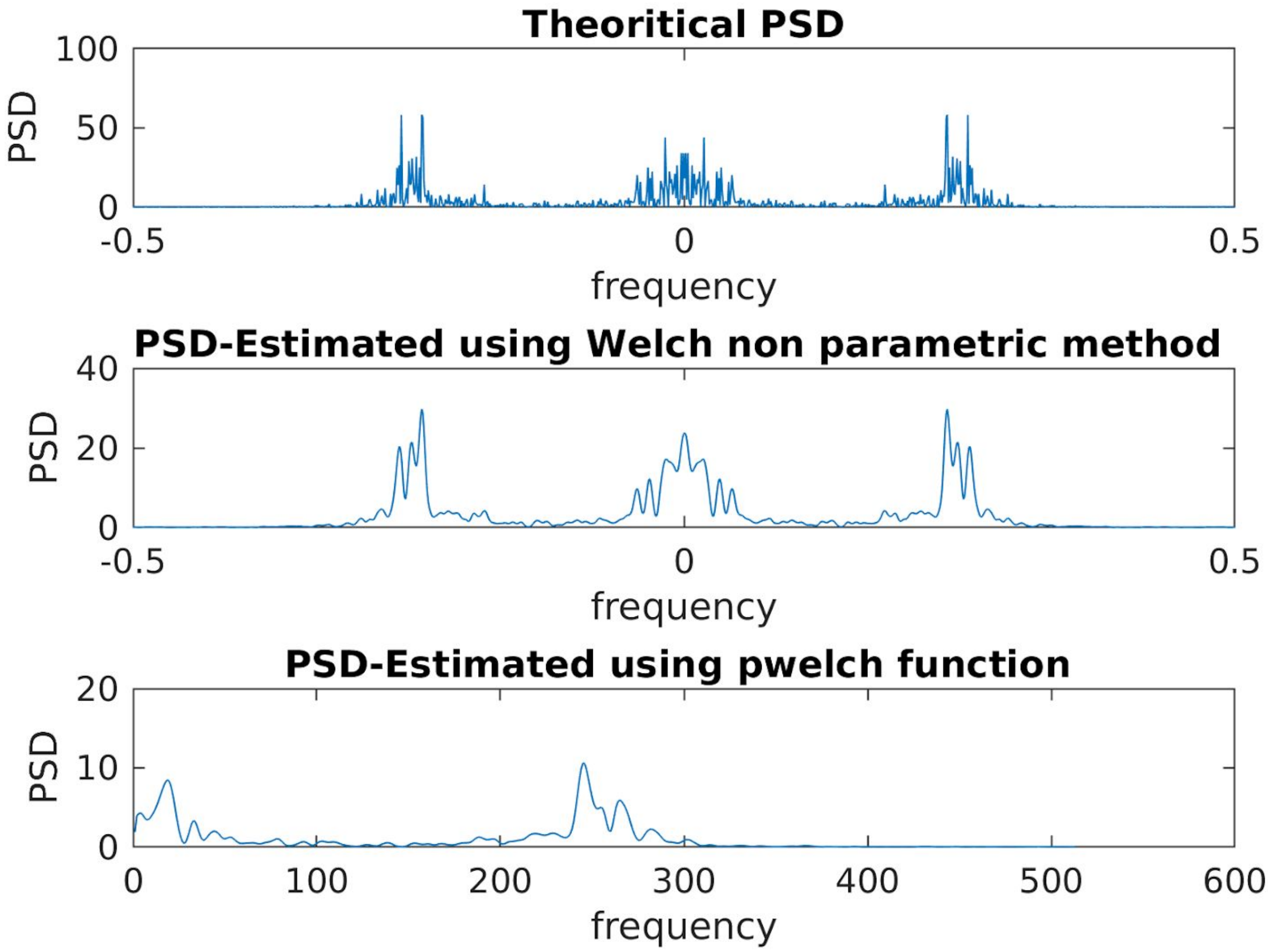
Experiment 05

Part 1: Welch Method

Code:

```
sigma = 1;
N = 1000;
noise = sigma.*normrnd(0,1,1,N);

b = 1;
a = [1, -0.9, 0.81, -0.729];
x = filter(b, a,noise);
f = fftshift(fft(x, 1024));
f_hat=linspace(-1,1,1024)*1/2;
subplot(3,1,1);
plot(f_hat,(abs(f).^2/N)*(sigma^2));
xlabel( "frequency");
ylabel("PSD");
title(" Theoritical PSD ")

L = 7;                          %number of blocks
D = 125;                                %overlap between blocks
M = (N + (L-1)*D)/L;            %size of block
y = [];

for block = 1:L
   y(:,block) = x((block-1)*D+1:(block -1)*D + M );
end

w = hamming(M);
sq_w = w.*w;
U = sum(sq_w)/M;

win_sig = [];
fft_sig = [];

for block = 1:L
  win_sig(:,block) = y(:,block).*w;
  fft_sig(:,block) = fft(win_sig(:,block),1024);
  power(:,block) = (abs(fft_sig (:,block)).^2)./(M*U);
End
p_welch = [];
for j = 1:1024
  p_welch (j)= 0;
  for block = 1:L
    p_welch (j) =  p_welch (j)+ power(j,block);
```

```
  end
    p_welch (j)=  p_welch (j)/L;
end

subplot(3,1,2);
plot(f_hat,fftshift(p_welch));
xlabel( "frequency");
ylabel("PSD");
title("PSD-Estimated using Welch non parametric method ");

subplot(3,1,3);
pxx = pwelch(x,w,0,1024);
plot(pxx);
xlabel( "frequency");
ylabel("PSD");
title("PSD-Estimated using pwelch function ");
print(gcf,'06a.png','-dpng','-r300');
```



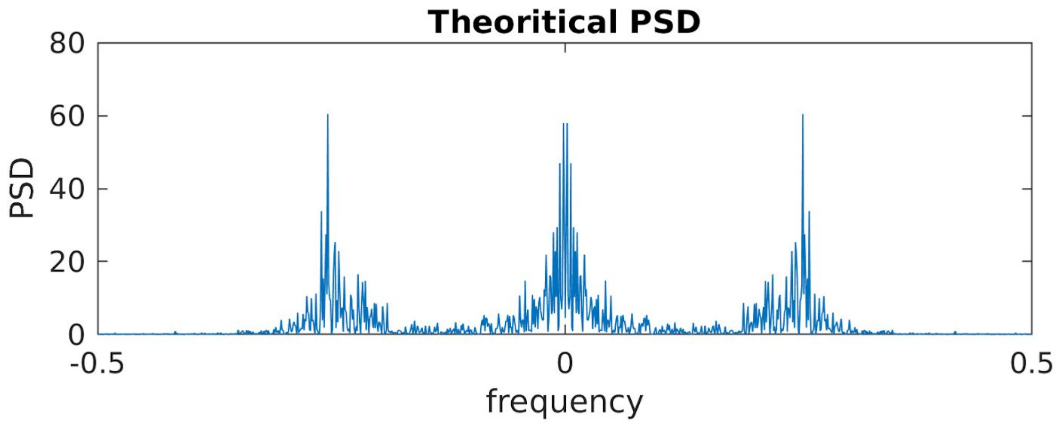
- The data is divided into segments with 50% overlap between them and Hamming window is used to compute the modified periodogram , which reduces the importance or weight given to the end samples.
- The final result is the average of the windowed periodograms that tends to decrease the variance of the estimate that would have occurred if we took a single periodogram estimate of the entire signal.

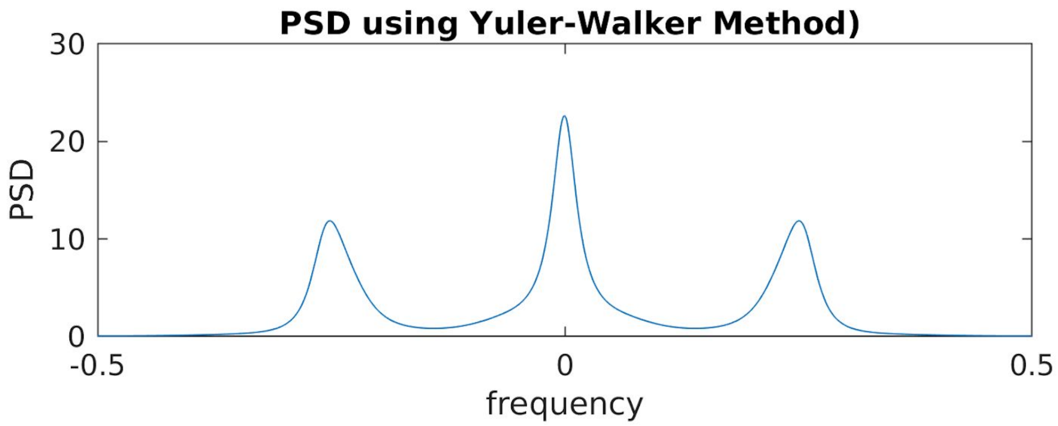
Part 2: Yule-Walker AR Parametric Method

Code:

```
N=1000;
p=10;

sigma = 1;
noise = sigma.*normrnd(0,1,1,N);
x = filter(1, [1, -0.9, 0.81, -0.729], noise).';
f = fftshift(fft(x, 1024));
f_hat=linspace(-1,1,1024)*1/2;

subplot(2,1,1);
plot(f_hat,abs(f).^2/N);
xlabel( "frequency");
ylabel("PSD");
title(" Theoritical PSD ")
rxx = zeros(p+1,1);

for m = 1:(p+1)
   for n = 1: N-m
      rxx(m,1)=rxx(m,1)+x(n)*x(n+m-1);
   end
   rxx(m,1)= (rxx(m,1)./N);
end

R = zeros(p+1,p+1);
for i=1:p+1
   for j=1:p+1
      R(i,j)= (rxx(abs(i-j)+1,1));
   end
end


AR = R(1:p,1:p);
C=R(2:p+1,1);
A=-inv(AR)*C;
var=AR(1,1)+ sum(A(:,1).*C(:,1));       %%%%%%%extimated var
```

```
Y = zeros(1024,1);
for f = 1:1024
   for k = 1:p
      e = exp(-1j*2*pi*f/1024*k);
      Y(f,1) = Y(f,1) + A(k,1)*e;
   end
   Y(f,1) = Y(f,1) +1;
end

P = var./(Y.*conj(Y));
subplot(2,1,2);
plot(f_hat, fftshift(P));
xlabel( "frequency");
ylabel("PSD");
title(" PSD using Yuler-Walker Method)");
```





- We do not use any windowing in this method because AR methods normally use linear prediction that extrapolates the signal and they have reduced side lobes.
- A lower model order results in a smooth spectrum and less resolution. But a higher order model may have false peaks due to extra poles that will appear in the filter function.