Digital Signal Processing Laboratory

Experiment 4

By: Hardik Tibrewal (18EC10020)

Power Spectral Density Estimation

Objective:

(a) Estimation of the power spectral density of a random signal using the Welch non-parametric method
(b) Estimation of the PSD of a random signal using the Yule-Walker AR parametric method

Theoretical Background:

The power spectral density of a signal is the distribution of the power of the signal over the frequency domain. Generally, the square of the magnitude of the signal is used to estimate the power and the constant factors are neglected for ease of calculation. The DFT (or FFT) of a signal gives an estimate for signals without noise or with a good SNR, but has high variance of the signal's spectral estimate at a given frequency for noisy signals or noise-like signals. This can be mitigated by averaging over time (like in the Welch method), or using an Auto-Regressive model (Yule-Walker model) to describe a few parameters to help with the estimation.

The Welch method is a non-parametric method, meaning it does not require prior knowledge of the signal to be analysed. The signal is split into a number of blocks and windowed. Then the FFT of each block gives the PSD of that block (after division of FFT with normalized power of the window and block length). This is called a periodogram, and finally, the average of the periodograms is taken to calculate the final estimated PSD. An overlap of 50% is usually present between blocks since the window functions tend to attenuate the values present at the edge of the window, so the overlap gives a more accurate result.

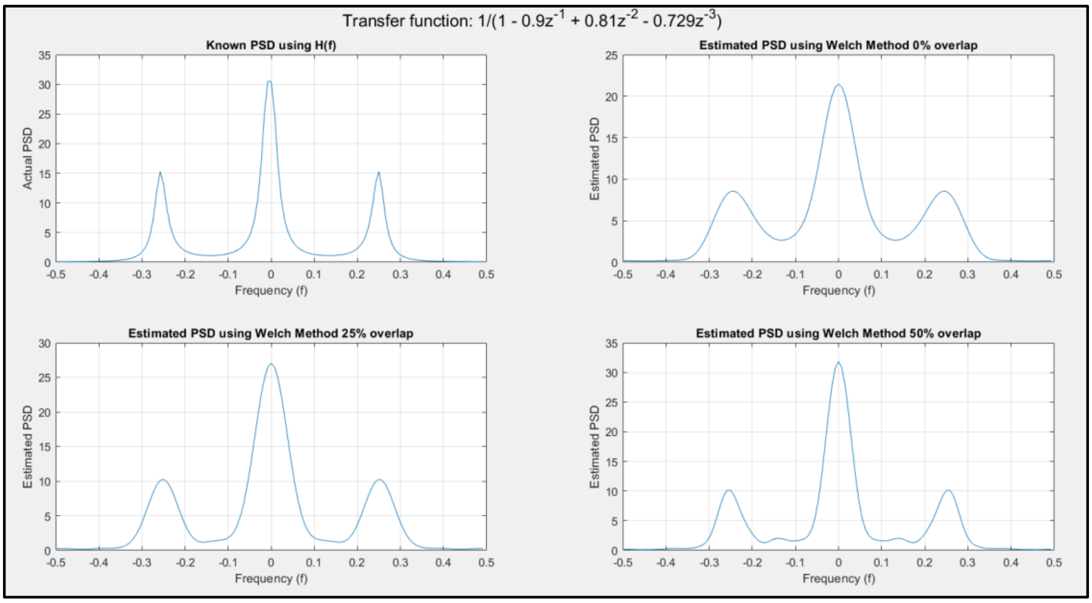
The Yule-Walker Auto-Regressive Method, which is a parametric method, is also implemented. The coefficients of the transfer function are estimated using the autocorrelation estimate of the signal. Using these coefficients, the estimated variance is also obtained. With the estimated variance and the transfer function, the PSD can be estimated.

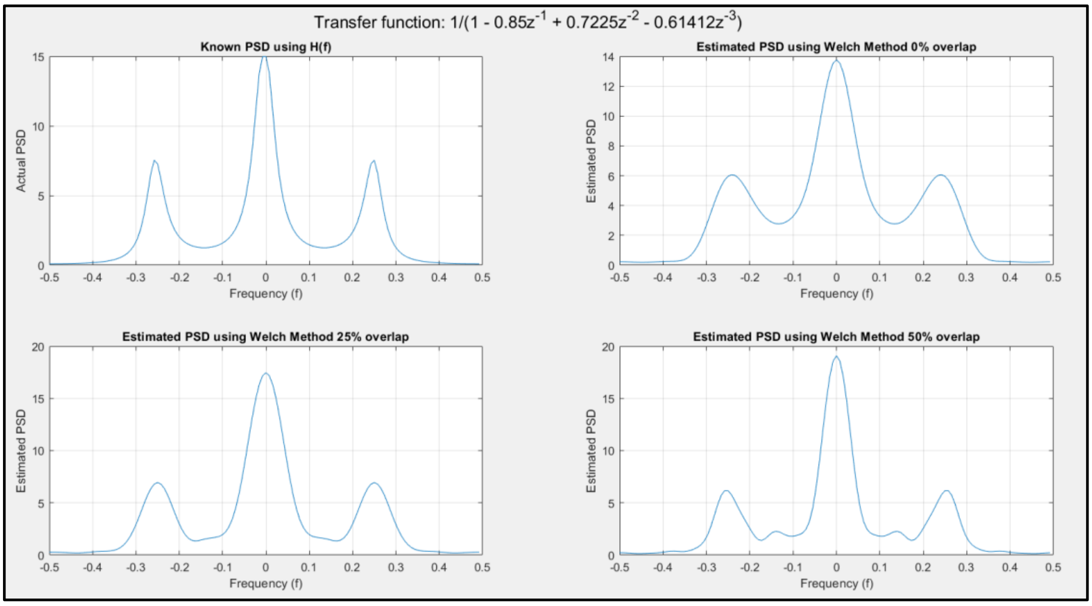
Pseudocode:

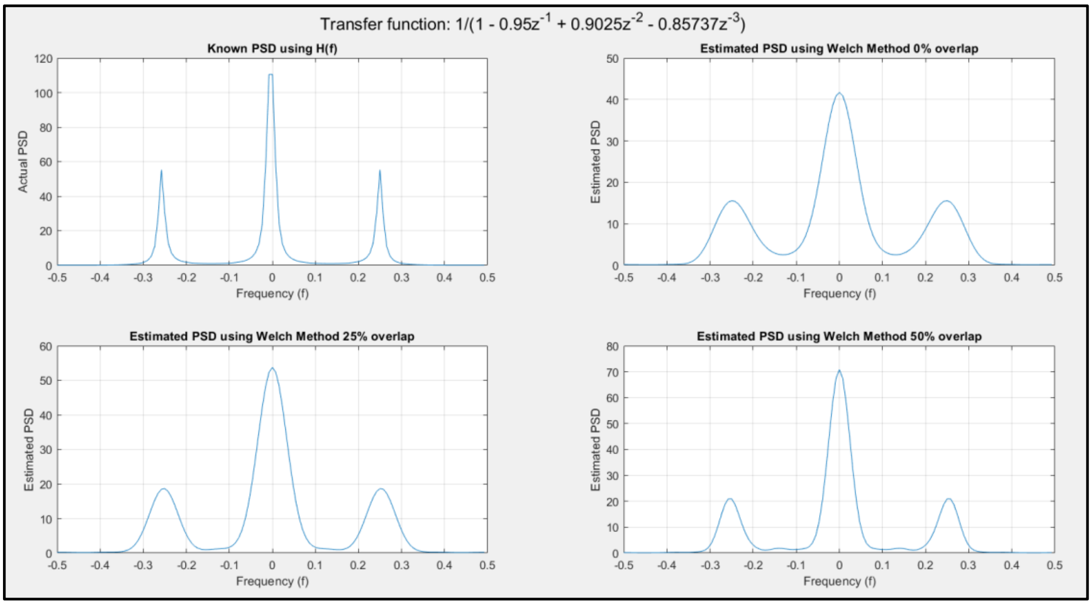
The random noise is generated using the *randn* function such that it has a desired mean and standard deviation. The transfer function and the length of the signal is defined. The noise is filtered using the transfer function to obtain the random sequence. Depending on the number of blocks and the fractional overlap between blocks, we decide the length of the blocks and the number of overlapping samples. A hamming window is then used to window each block of the random sequence, and the estimated PSD using the Welch method is obtained by taking the FFT of the windowed blocks and then normalising and averaging it. The ideal theoretical PSD is obtained by getting the frequency response using the *freqz* function, and then the variance is multiplied to the frequency domain transfer function since the sequence is a zero-mean random noise. This is compared with the PSD estimation from Welch method.

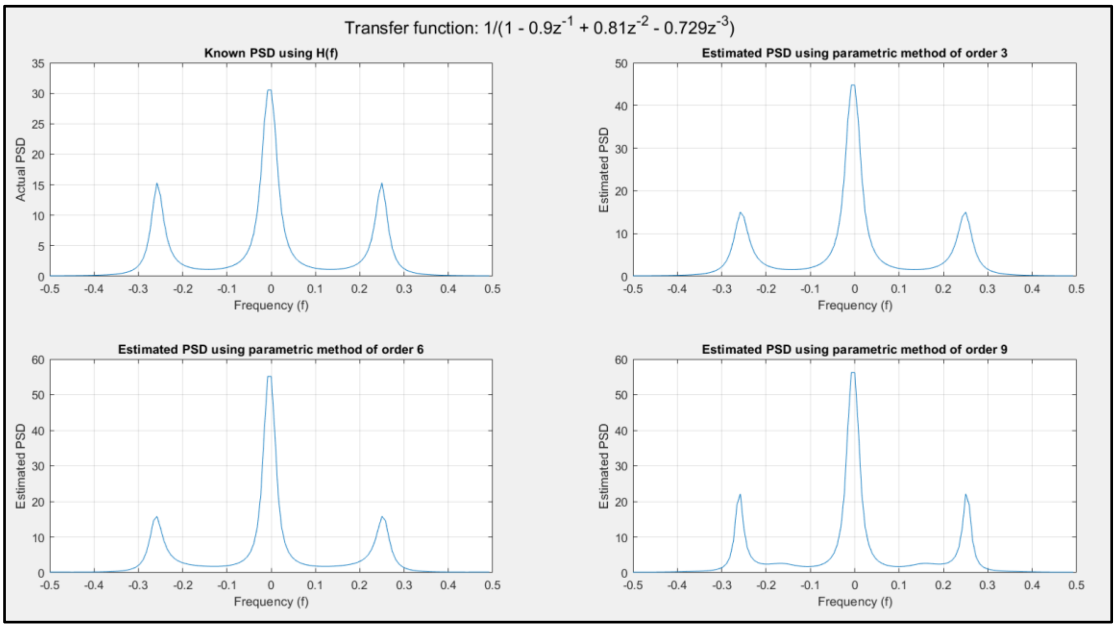
For the parametric method, prior knowledge of the source generating the random sequence is used to determine the order of the transfer function. Based on the autocorrelation estimates, the Yule-Walker coefficients are obtained through matrix operations as justified by the theory. The estimated variance is also calculated using these coefficients and the autocorrelation estimate. The *freqz* function is used to obtain the frequency response which is multiplied by the estimated variance to obtain the estimated PSD.

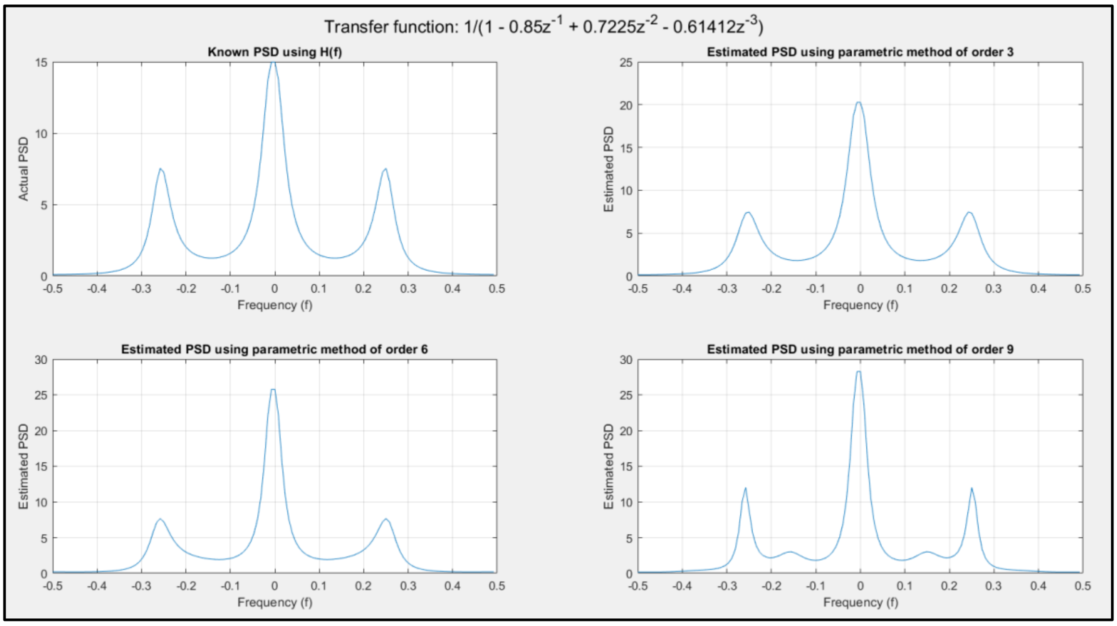
<u>Simulation Results:</u>

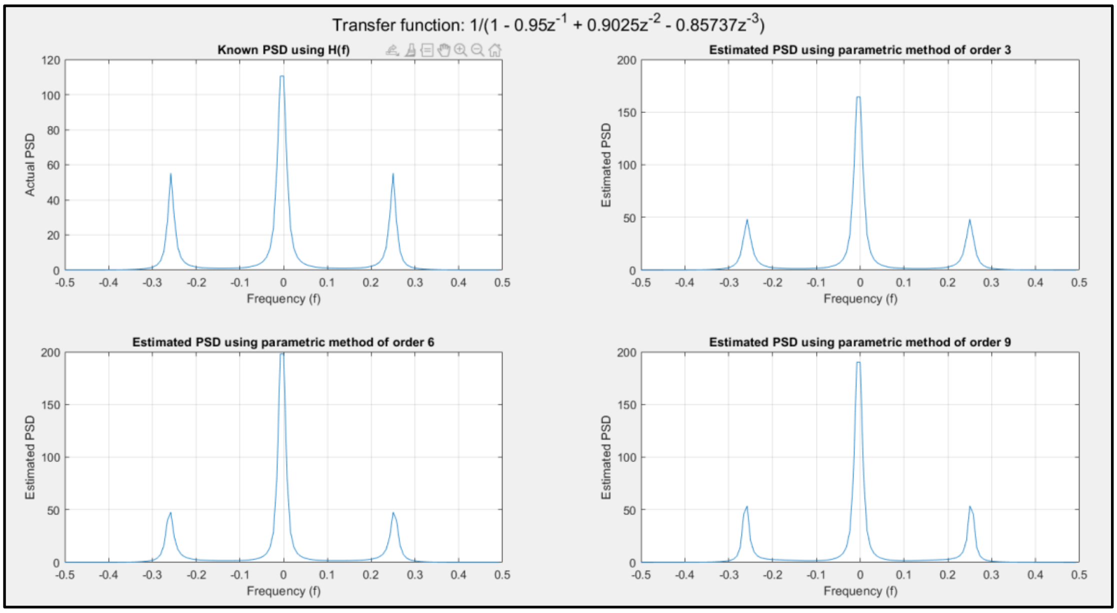







Transfer function: 1/(1 - 0.9z^-1 + 0.81z^-2 - 0.729z^-3)
Known PSD using H(f)
Actual PSD
Frequency (f)
Estimated PSD using parametric method of order 3
Estimated PSD using parametric method of order 6
Estimated PSD using parametric method of order 9
Estimated PSD


Transfer function: 1/(1 - 0.85z^-1 + 0.7225z^-2 - 0.61412z^-3)
Known PSD using H(f)
Actual PSD
Frequency (f)
Estimated PSD using parametric method of order 3
Estimated PSD using parametric method of order 6
Estimated PSD using parametric method of order 9
Estimated PSD


Transfer function: 1/(1 - 0.95z^-1 + 0.9025z^-2 - 0.85737z^-3)
Known PSD using H(f)
Actual PSD
Frequency (f)
Estimated PSD using parametric method of order 3
Estimated PSD using parametric method of order 6
Estimated PSD using parametric method of order 9
Estimated PSD

Results:

The PSD estimated using both the methods has a plot similar to the theoretical plot. In the Welch method, we see the plots getting sharper and closer in magnitude to the theoretical plots as we increase overlap. In the Yule-Walker AR method, we see the plots getting sharper and the magnitude increasing with an increase in the order of the model.

Discussion:

Estimation of the PSD of a signal gives the spread of the power with respect to the frequency, and the Welch method and Yule-Walker methods are used to estimate the PSD of noise-like signals, since the FFT gives results with a high variance for such type of signals.

In the Welch method, we used the average of 8 periodograms to estimate the PSD of the random sequence. The blocks are made to overlap since the windowing attenuates the values at the ends of the block, so overlapping reduces that effect. It is observed that increasing the amount of overlap generally improves the estimation, however too much overlap causes some inaccuracies. Generally, an overlap of 50% is taken, but as we see here, the 25% overlap appears to be a better approximation. Depending on various parameters, the percentage overlap should be varied to suit the requirements. Changing the transfer function does not vary the shape of the PSD by much, but the magnitude changes quite a bit. However, this does not seem to affect the estimation.

In the Yule-Walker method, we assume the sequence to be generated by passing zero-mean white noise through a p^{th} order transfer function. The autocorrelation information is used to estimate the PSD by finding the coefficients of the model and estimating the variance. The order of the model is varied, and increasing the order seems to lead to an improvement in the estimation. The centre peak sees a sharp increase in magnitude, due to the difference between the estimated and actual parameters, as well as the different order. Increasing the order indefinitely will cause a deviation from the theoretical PSD, and also be computationally heavier.

Appendix

Code:

```
clc
clear all
close all

mean = 0;
std_dev = 1;
N = 128;
rng('default');
noise = std_dev.*randn(N,1) + mean;
denom = 0.9;
A = [1, -denom, denom^2, -denom^3];

X = filter(1, A, noise); %Random Sequence
L = 8;

[H, W] = freqz(1,A,N/2);
H = (abs(H).^2).*std_dev^2;
H1 = [flip(H); H];
figure();
sgtitle("Transfer function: 1/(1 - "+abs(A(2))+"z^{-1} + "+abs(A(3))+"z^{-2} - "+abs(A(4))+"z^{-3})")
subplot(221);
f = -0.5:1/N:(0.5-(1/N));
plot(f, H1);
grid on;
xlabel('Frequency (f)');ylabel("Actual PSD");title("Known PSD using H(f)");

%%% Welch Method %%%
for ov_lap = [0, 0.25, 0.5] %fractional overlap
M = round(N/(L*(1-ov_lap)+ov_lap));
D = round(ov_lap*M);

window = hamming(M);
U = (1/M)*sum(window.^2);

X_divs = zeros(M, L);
P_xx = zeros(N, 1);
l=1; r=M;

for ii = 1:L
   X_divs(:,ii) = X(l:r);
   l = r-D+1;
   r = l+M-1;
end

for ii = 1:L
   X_divs(:,ii) = X_divs(:,ii).*window;
   P_xx(:) = P_xx(:) + (abs(fft(X_divs(:,ii),N)).^2)/(L*M*U);
end

subplot(2,2,ov_lap*4+2);
plot(f, fftshift(P_xx));
grid on;
xlabel('Frequency (f)');ylabel("Estimated PSD");title("Estimated PSD using Welch Method
"+num2str(100*ov_lap)+"% overlap");
end
```

```
figure();
sgtitle("Transfer function: 1/(1 - "+abs(A(2))+"z^{-1} + "+abs(A(3))+"z^{-2} - "+abs(A(4))+"z^{-3})")
subplot(221);
f = -0.5:1/N:(0.5-(1/N));
plot(f, H1);
grid on;
xlabel('Frequency (f)');ylabel("Actual PSD");title("Known PSD using H(f)");

%%% Parametric Method %%%
for p = [3, 6, 9]
    R = xcorr(X)/N; %Autocorrelation calculation
    r = R(N:N+p);

    %Matrix initialisation
    mat = zeros(p,p);
    mat2 = -1*(r(2:p+1));

    for ii = 1:p
        for jj = 1:p
            mat(ii,jj) = r(abs(ii-jj)+1);
        end
    end

    coeff_a = [1; mat\mat2]; %Coefficients. mat\mat2 gives inv(mat)*mat2
    new_var = sum(coeff_a.*r); %Variance

    [h_new, w_new] = freqz(1, coeff_a(:,1), N/2);
    h_new =  (abs(h_new).^2)*(new_var);
    l_new = [flip(h_new); h_new];
    subplot(2,2,p/3+1);
    plot(f,l_new);
    grid on;
    xlabel('Frequency (f)');ylabel("Estimated PSD");title("Estimated PSD using parametric method of order "+p);
end
```