Digital Signal Processing Lab

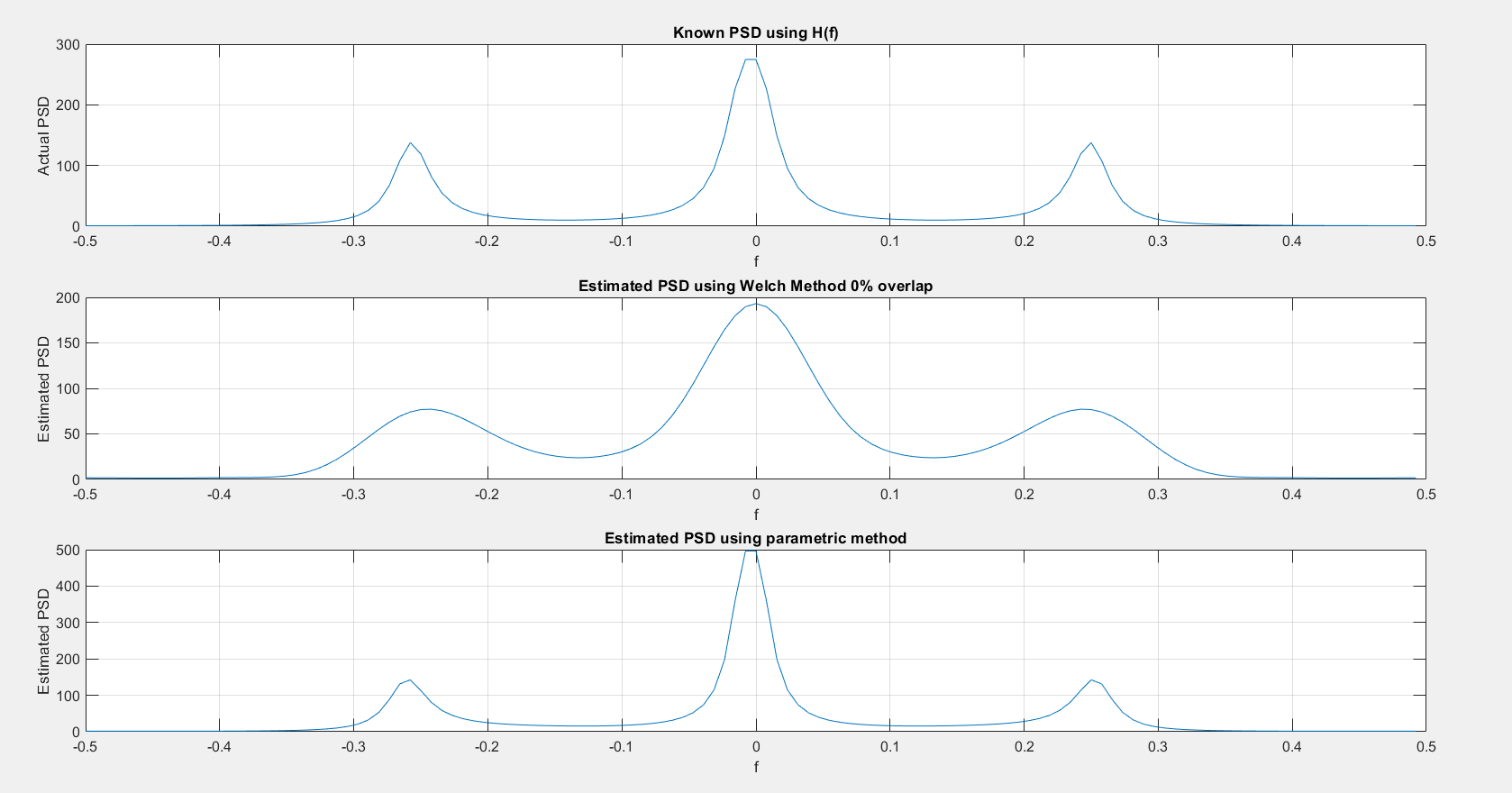
Experiment 4b

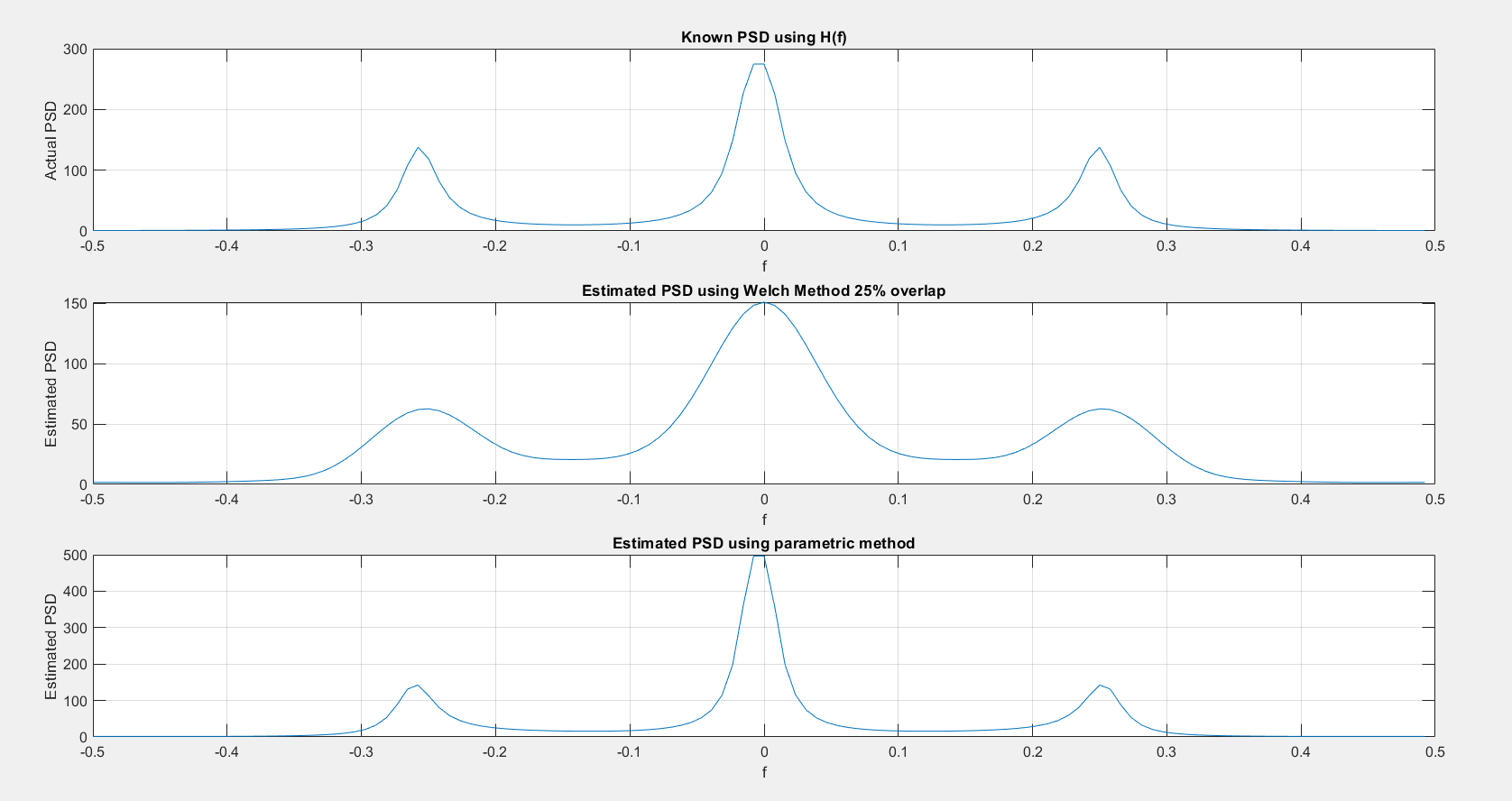
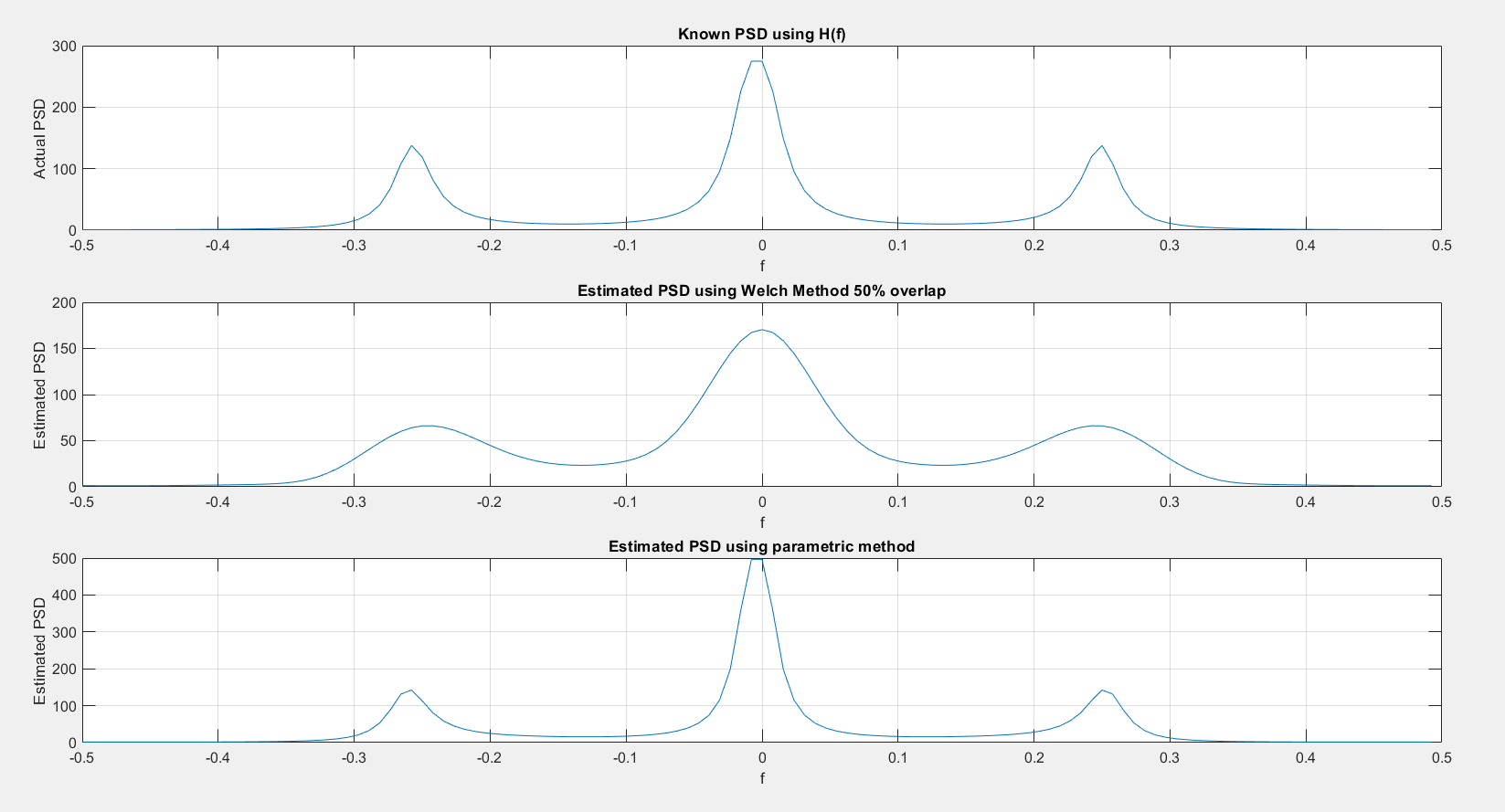
By Hardik Tibrewal (18EC10020)

**Aim:**

Power Spectrum Estimation using Parametric Method (Yule-Walker AR Model)

**Plots:**

(Denominator of transfer function is 1z0 - 0.9z-1 + 0.81z-2 - 0.729z-3)



**Code:**

clc

clear all

close all

mean = 0;

std\_dev = 3;

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);

%L = 8;

for ov\_lap = [0, 0.25, 0.5]

M = 16;

D = ov\_lap;

L = floor((N-1)/(M\*(1-D)));

num\_blocks = ceil((N-1)/(M\*(1-D)));

X\_divs = zeros(num\_blocks, M);

end\_case = L ~= num\_blocks;

for ii = 1:L

X\_divs(ii,:) = X((1+(ii-1)\*floor(M\*(1-D))):(M+(ii-1)\*floor(M\*(1-D))));

end

if end\_case

idx = 1+L\*floor(M\*(1-D));

X\_divs(L+1,:) = [X(idx:end)' zeros(1, M-(N-idx+1))];

end

n = 0:1:(M-1);

hamming = 0.54 - 0.46\*cos(2\*pi\*n/(M-1));

U = (1/M)\*sum(hamming.\*hamming);

P\_n = zeros(num\_blocks, M);

for ii = 1:num\_blocks

P\_n(ii,:) = X\_divs(ii,:).\*hamming;

end

f = -0.5:1/N:(0.5-(1/N));

cosine = 0; sine = 0;

P\_f = zeros(num\_blocks, N);

for ii = 1:num\_blocks

for F = 1:length(f)

cosine = 0; sine = 0;

for jj = 1:M

cosine = cosine + cos(2\*pi\*f(F)\*jj)\*P\_n(ii,jj);

sine = sine + sin(2\*pi\*f(F)\*jj)\*P\_n(ii,jj);

end

idx = floor((N - length(f))/2)+F;

P\_f(ii, idx) = (cosine^2 + sine^2)/(M\*U);

end

end

Pw\_f = zeros(1, N);

for ii = 1:num\_blocks

Pw\_f = Pw\_f + P\_f(ii,:);

end

Pw\_f = Pw\_f/num\_blocks;

[H, W] = freqz(1,A,N/2);

figure();

subplot(311);

l1 = (abs(H).^2).\*std\_dev^2;

l2 = flip(l1);

l = [l2' l1'];

plot(f, l);

grid on;

xlabel('f');ylabel("Actual PSD");title("Known PSD using H(f)");

subplot(312);

plot(f, Pw\_f);

grid on;

xlabel('f');ylabel("Estimated PSD");title("Estimated PSD using Welch Method "+num2str(D\*100)+"% overlap");

p = 6;

r = zeros(p+1);

for ii = 0:p

for jj = 1:(N-ii)

r(ii+1) = r(ii+1) + X(jj)\*X(jj+ii);

end

r(ii+1) = r(ii+1)/N;

end

mat = zeros(p,p);

mat2 = zeros(1,p);

for ii = 1:p

mat2(1,ii) = -r(ii+1);

end

for ii = 1:p

for jj = 1:p

mat(ii,jj) = r(abs(ii-jj)+1);

end

end

mat\_inv = inv(mat);

coeff\_a = mat2\*mat\_inv;

coeff\_a = coeff\_a';

new\_std\_dev = 0;

for ii = 1:p

new\_std\_dev = new\_std\_dev + coeff\_a(ii,1)\*r(ii+1);

end

new\_std\_dev = new\_std\_dev + r(1);

A\_new = ones(p+1);

for ii = 1:p

A\_new(ii+1) = coeff\_a(ii);

end

[h\_new, w\_new] = freqz(1, A\_new(:,1), N/2);

subplot(313);

l2\_new = (abs(h\_new).^2)\*(new\_std\_dev);

l1\_new = flip(l2\_new);

l\_new = [l1\_new' l2\_new'];

plot(f,l\_new);

grid on;

xlabel('f');ylabel("Estimated PSD");title("Estimated PSD using parametric method");

end