Machine Learning

Asam Mahmood

Project 3

Problem 1

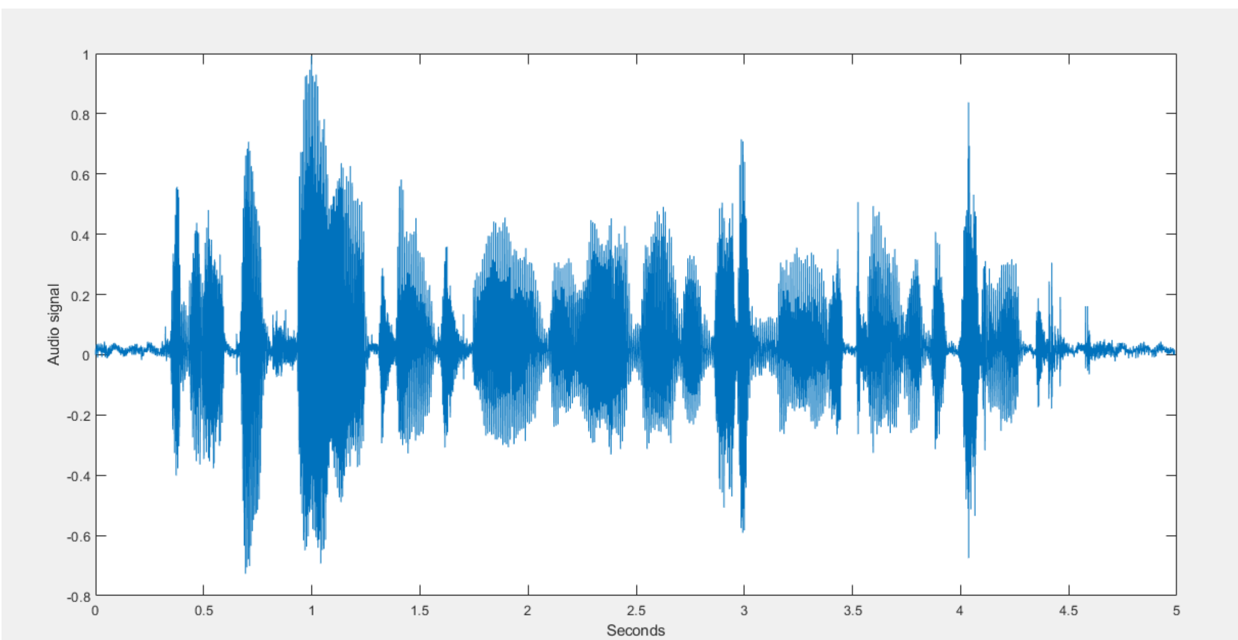
Read an audio file x and perform preliminary exploration of it.

- **Plot the signal versus time.**

Matlab Code:

```
[x,fs] = audioread('D:\New Folder\dft.wav');
x = x(:,1);
    dt = 1/fs;
     t = 0:dt:(length(x)*dt)-dt;
    plot(t,x); xlabel('Seconds'); ylabel('Audio signal');
```

Output:



• Divide the signal into frames that are approximately the duration of 5 to 10 times the speaker's pitch period. Frame overlap should be at least 50%. Arrange the frames as the columns of a matrix X.

Matlab code:

```
[x,fs] = audioread('D:\New Folder\dft.wav');
%x = x(:,1);
n=round(length(x)/20); %find how many samples will each frame contain
X=zeros(n,10); %preallocate the matrix for 10 colums of Nsamples/10 in each
for k=0:9
    X(:,k+1)=x(1+n*k:n*(k+1));
end
```

- **Evaluate the DFT of each frame by using MATLAB's fX=fft(x,nfft).**

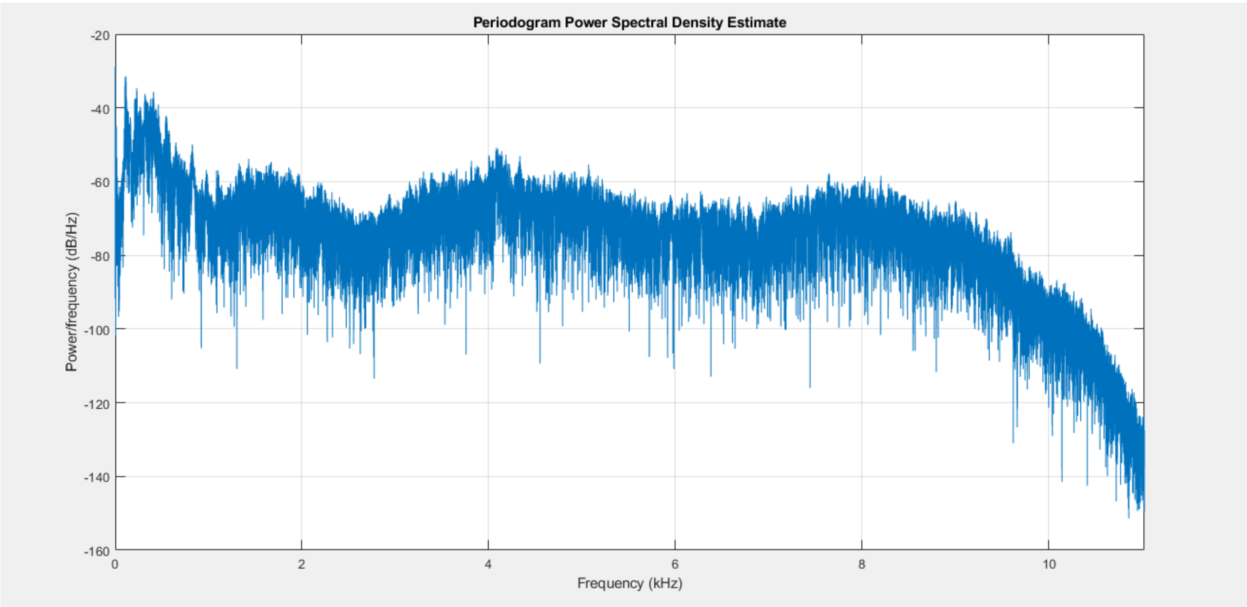
Plot the spectral magnitude and the log spectral magnitude for 0 ≤ f ≤ Fs 2 of each frame with a pause(0.3) between displays to make a movie. You may adjust the amount of pause.

- **Plot the spectrogram of the signal.**

Matlab code:

```
clc
clear all
[x,fs] = audioread('D:\New Folder\dft.wav');
%x = x(:,1);
n=round(length(x)/20); %find how many samples will each frame contain
X=zeros(n,10); %preallocate the matrix for 10 colums of Nsamples/10 in each
for k=0:9
    X(:,k+1)=x(1+n*k:n*(k+1));
end
    dt = 1/fs;
     t = 0:dt:(length(x)*dt)-dt;
    plot(t,x); xlabel('Seconds'); ylabel('Audio signal');
    figure
    plot(psd(spectrum.periodogram,x,'Fs',fs,'NFFT',length(x)));
```

Output:



Problem 2

2. Use time-domain methods described in Chapter 4 of reference 1.

(a) Use Eqn. 4.6 to find the Short-time energy (STE) of each frame. Use a Hamming window for each frame.

Matlab code:

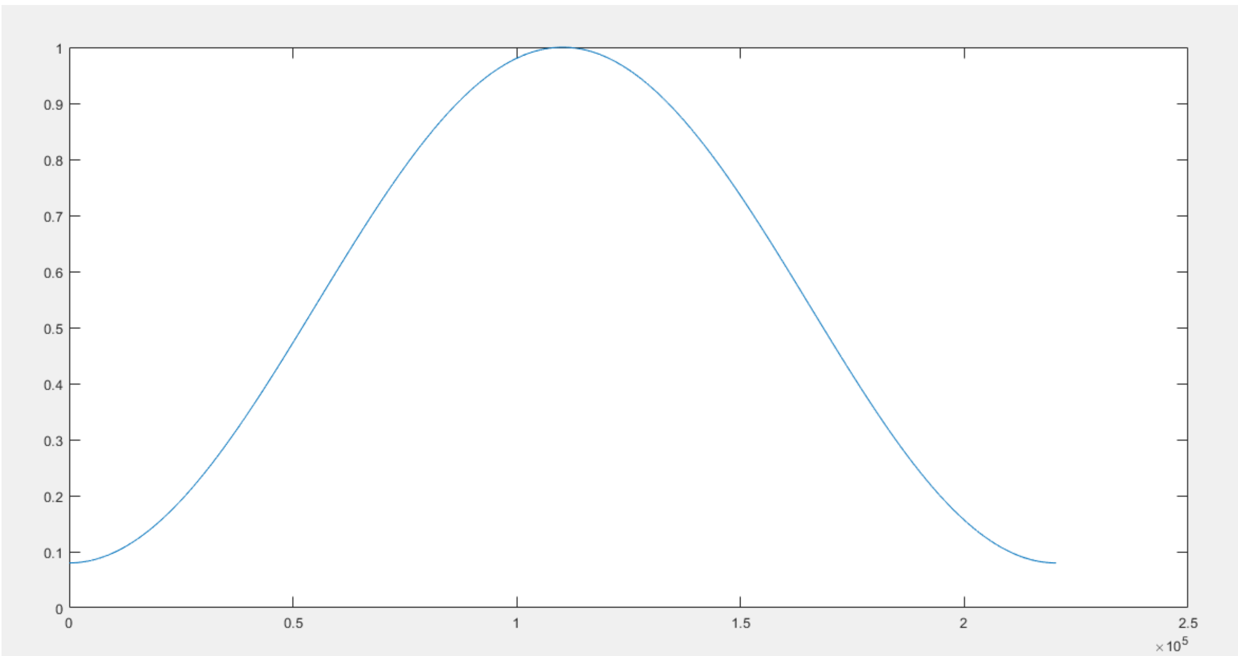
```
close all
clear
clc
[X,Fs] = audioread('dft.wav');
dt=1/Fs;
L=length(X);
signal_duration=L/Fs; % duration of the whole speech signal in seconds
prompt = 'Enter the speech frame duration ';
frame_duration=input(prompt); % duration of short time frames in msec
frame_length=ceil(frame_duration*Fs); % Number of samples of the window
nfft = 2^nextpow2(frame_length); % Number of DFT points
w = hamming(frame_length); % type of window
prompt = 'Enter the number of overlapping points ';
```

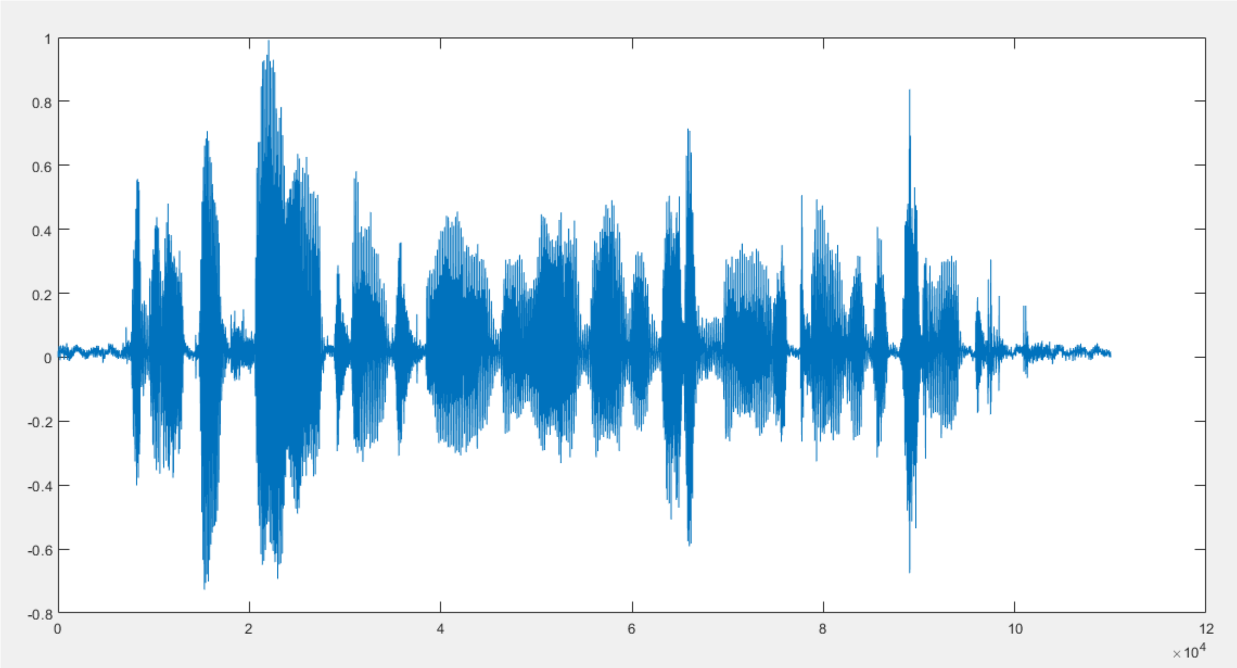
(b) Use Eqn. 4.7 to find the Short-time zero crossing rate (STZCR)of each frame.

```
n_overlap=input(prompt); % duration of short time frames in msec
```

```
prompt = ('Enter the sample index to begin the windowing ');
index=input(prompt); % duration of short time frames in msec

Nbr_frames=floor((L-n_overlap)/(frame_length-n_overlap)); % Total number of
over-lapped frames which will divide the whole signal
signal_framed=zeros(L,Nbr_frames);
Y=zeros(nfft,Nbr_frames);
t=zeros(1,Nbr_frames);
zcd_signal = dsp.ZeroCrossingDetector;
x=frame_length-n_overlap;
```

(c) Plot the STE and STZCR versus time in the same figure with the signal as shown in Fig. 4.4.





(d) Comment on the results.

By the idea of creation, the discourse signal comprise of voiced, unvoiced and quiet locales. Further the energy related with voiced area is huge contrasted with unvoiced district and quietness locale won't have least or irrelevant energy.
(v) The ZCR can be characterized as U(n) in eq., with T[S(n)] = 0.5|sgn[S(m)] - sgn[S(m-1)]|, where the mathematical indication of S(n) is given in eq. what's more, W(n) is a rectangular window scaled by 1N as given in eq., would yeild zero-intersections/test, or by FSN to yield zero-intersections.

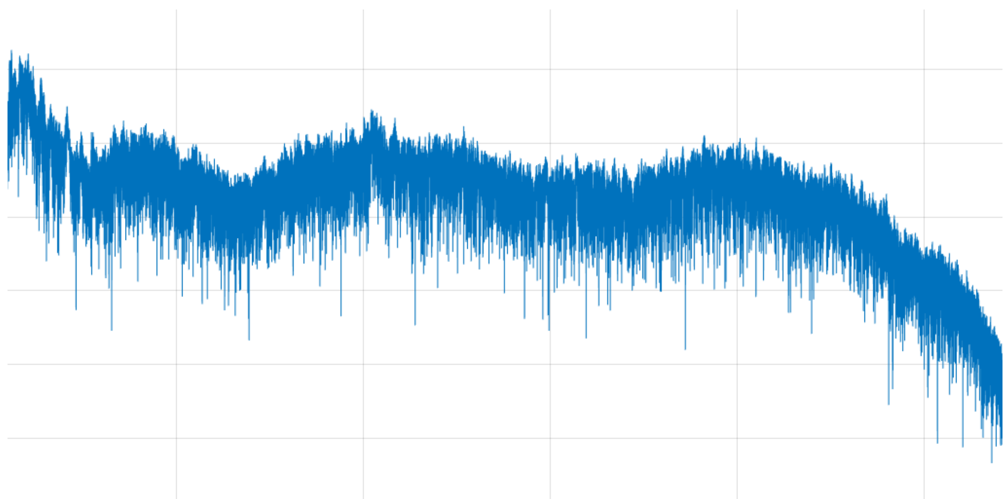
(e) Find the Short-time autocorrelation (STACF of each frame as described in Section 4.2.

```
for k=1:Nbr_frames
        signal_framed(:,k)=[zeros(1,(k-1)*x) w' zeros(1,L-frame_length-(k-
1)*x)]'.*X; % Frameing
        E(:,k)=sum(signal_framed(:,k).*signal_framed(:,k)); % energy per each
over-lapped frame
        numZeroCross(:,k) = zcd_signal(signal_framed(:,k)); % zero crossing
per each over-lapped frame
        t(:,k)=(k)'; % over-lapped frame index
end
```

(f) Plot the STACF for each segment. Study your result and comment on your observations. Include only one demonstrative STACF plot in your

```
% find the frame with the maximum energy (voiced frame )
find(E==max(E));
% find the frame with the maximum zero crossing ( unvoiced frame )
find(numZeroCross==max(numZeroCross));
```

```
for k=index:Nbr_frames
            signal_framed(:,k)=[zeros(1,(k-1)*x) w' zeros(1,L-frame_length-(k-1)*x) ]'.*signal;
            Y(:,k)=abs(fft(nonzeros(signal_framed(:,k)),nfft));
            P1(:,k) = Y(1:nfft/2+1,k);
            P1(2:end-1,k) = 2*P1(2:end-1,k);
End
```



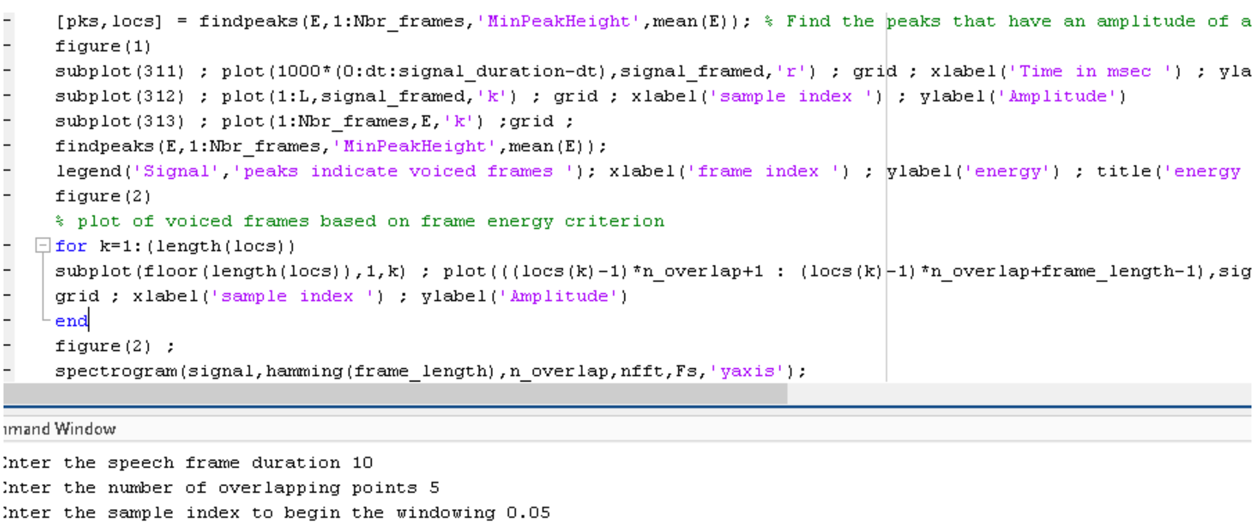
It has many plots like this

(g) Write a program to determine if a segment is voiced or unvoiced using the STE, STZCR and STACF. The program should consider STE and STZCR thresholds to determine a frame's label. It should also consider the distinguishing properties of the STACF (periodicity) for the label. You may use a vote or a weighted vote of the results of the three measures to make a final decision for a segment's label as Voiced or Unvoiced.

```
[pks,locs] = findpeaks(E,1:Nbr_frames,'MinPeakHeight',mean(E)); % Find the peaks that have an amplitude of at least mean of frame energy
figure(1)
subplot(311) ; plot(1000*(0:dt:signal_duration-dt),signal_framed,'r') ; grid ; xlabel('Time in msec ') ; ylabel('Amplitude')
subplot(312) ; plot(1:L,signal_framed,'k') ; grid ; xlabel('sample index ') ; ylabel('Amplitude')
subplot(313) ; plot(1:Nbr_frames,E,'k') ;grid ;
findpeaks(E,1:Nbr_frames,'MinPeakHeight',mean(E));
legend('Signal','peaks indicate voiced frames '); xlabel('frame index ') ; ylabel('energy') ; title('energy per frame ')
figure(2)
```

```
% plot of voiced frames based on frame energy criterion
for k=1:(length(locs))
subplot(floor(length(locs)),1,k) ; plot(((locs(k)-1)*n_overlap+1 : (locs(k)-
1)*n_overlap+frame_length-1),signal((locs(k)-1)*n_overlap+1 : (locs(k)-
1)*n_overlap+frame_length-1)); ...
grid ; xlabel('sample index ') ; ylabel('Amplitude')
end
figure(2) ;
spectrogram(signal,hamming(frame_length),n_overlap,nfft,Fs,'yaxis');
```

Output:



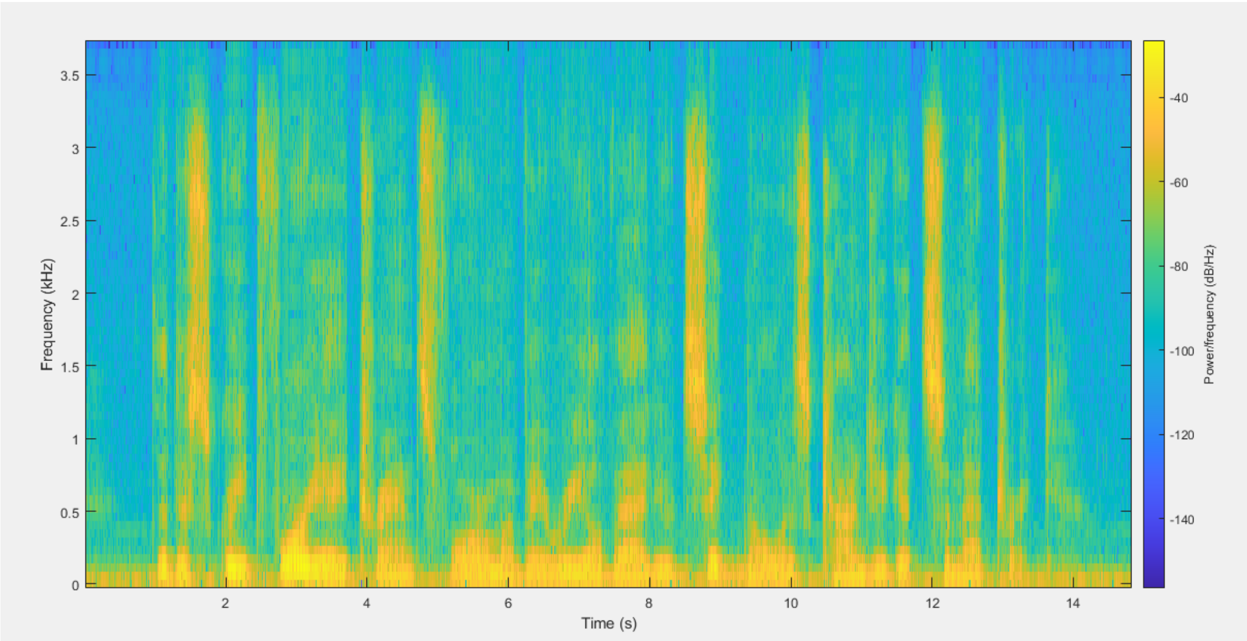
Problem 3:

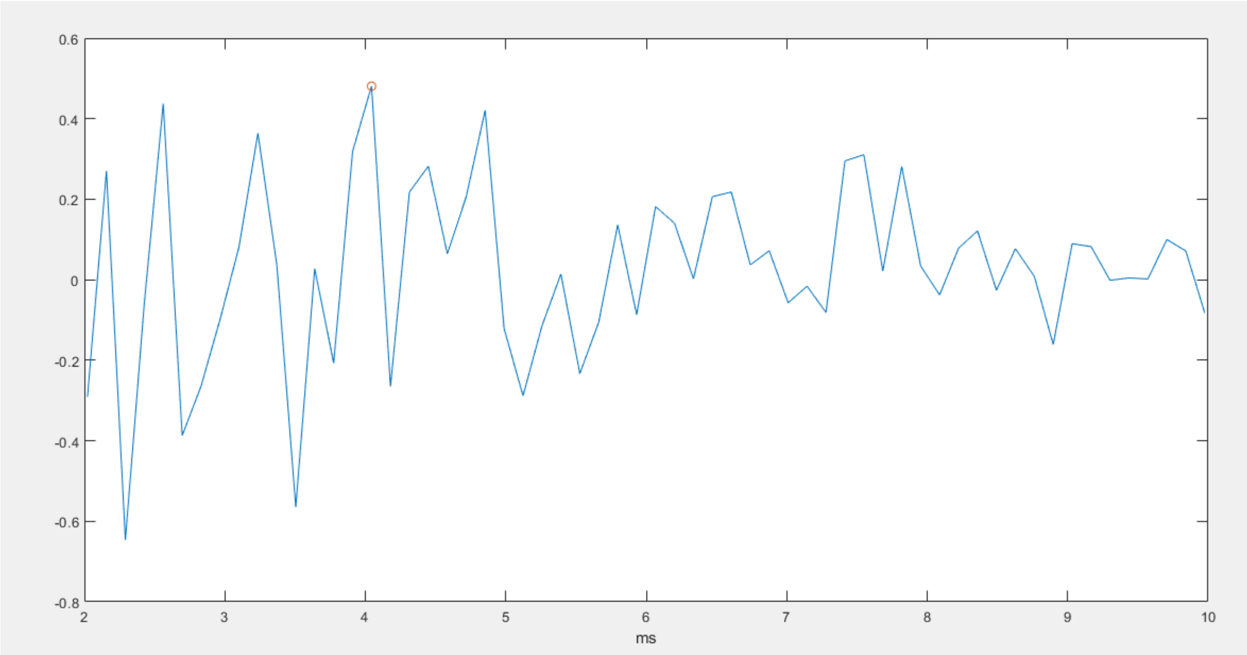
Matlab code:

```
X = audioread('dft.wav');
segmentlen=100;
noverlap=90;
NFFT=128;
Fs=7418;
spectrogram(X,segmentlen,noverlap,NFFT,Fs,'yaxis');

dt=1/Fs;
I0=round(0.1/dt);
Iend=round(0.25/dt);
x=X(I0:Iend);

c=cceps(x);

t=0:dt:length(x)*dt-dt;
```

```
trng=t(t>=2e-3 & t<=10e-3);
crng=c(t>=2e-3 & t<=10e-3);
[~,I]=max(crng);
fprintf('complex ceptrum F0 estimate is %3.2f Hz.\n',1/trng(I))

figure

plot(trng*1e3,crng);
xlabel('ms');
hold on
plot(trng(I)*1e3,crng(I),'o');
hold off

[b0,a0]=butter(2,335/(Fs/2));
xin=abs(x);
xin=filter(b0,a0,xin);
xin=xin-mean(xin);
zc=zerocrossrate(xin);
F0=0.5*Fs*zc;
fprintf('zero crossing F0 estimate is %3.2f Hz.\n',F0)
```

Output:





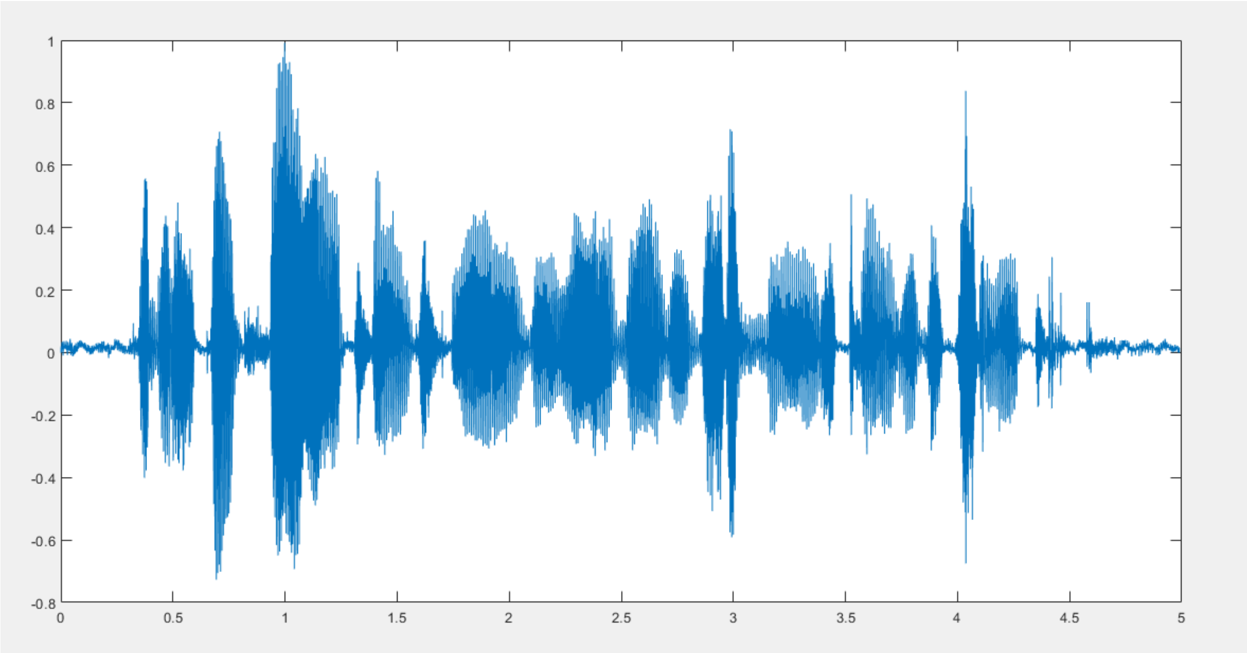
complex ceptrum F0 estimate is 247.27 Hz.
zero crossing F0 estimate is 58.32 Hz.

Problem 4:

Matlab code:

Part a

```
clear; clc; close all;
[x,sr]=audioread('D:\New Folder\dft.wav');
%[x,sr]=audioread('sa1.wav');
td=1/sr;
%soundsc(x,sr)
lx=length(x);
t=[0:lx-1]*td;
 figure
 plot(t,x)
```



Part b

```
%[x, sr]=frame2im(x, 125, 0.9, sr);
nof=size(x,2);
lof=size(x,1);
nfft_fr=2^11;
fxm=fft(x,nfft_fr);

afxm=abs(fxm(1:nfft_fr/2,:));
t_fr=[0:nof-1]*sr;
f=[0:nfft_fr/2-1]/nfft_fr*sr;
figure
imagesc(t_fr,f,20*log10(afxm)), axis xy, colormap(jet), colorbar
M=20;  %no of lags acf
```

Part c

```
r=zeros(M,M);
dy=M/sr*1e3;  %duration (ms) of a frame with M samples
V=zeros(M,M,nof);
D=zeros(M,nof);
VV=V;
Dv=D;
```

Part d

```
for k=1:1:nof
    y=x(:,k);
```

Part e

```
    [ym, hty]=frames(y,dy , 0.9, sr);%Divide each frame into 10 sample frames with 9
sample overlap
    nofy=size(y,2);
    r=x.*x;    %acf estimate  /nof
```

Part f

```
    [v d]=eig(V);

    V(:,:,k)=v;
```

Part g

```
    lambda=diag(d);
    D(:,k)=lambda;
```

Part h

```
    vym=y;
    rv=vym; %acf estimate /nofy
```

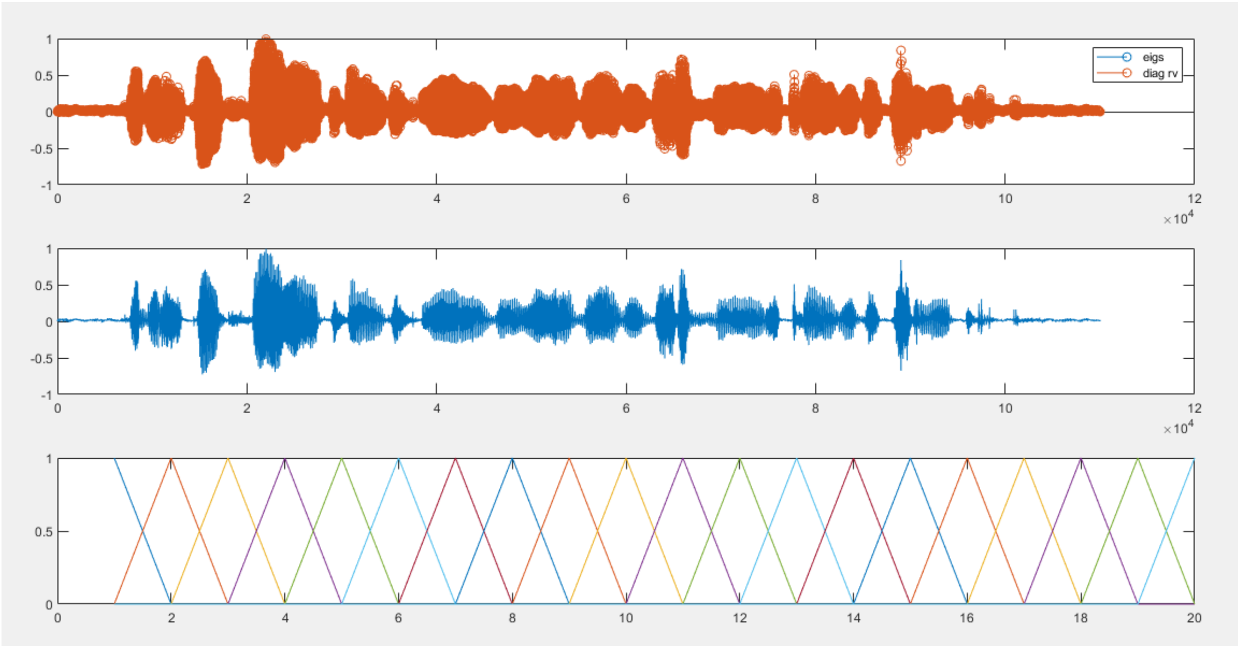
Part i

```
    [vv dv]=eig(VV);
    Vv(:,:,k)=vv;
```

Part j

```
    lambdav=v;
    subplot(311),stem(lambda), hold, pause(0.3), stem((rv)), legend('eigs','diag
rv'), hold off
    subplot(312), plot(y)
    subplot(313), plot(v)

end
figure
plot(D)
```

Part k


eigs
diag rv