Problem 1

What is the solution to the difference equation $y[n] = Ky[n-1]$ with initial condition $y[0] = 1$ and $y[-1] = 0$? Next, find the ROC of z-transform of your solution for $y[n], K \leq 1$.

Solution.
From the differential equation,

$$
\begin{aligned}
y[n] &= Ky[n-1] + \delta[n] \\
Y(Z) &= K(z^{-1}Y(Z)) + z \\
Y(Z) &= \frac{z}{1 - Kz^{-1}} \\
\implies Y(Z) &= \frac{z^2}{z - K} \\
\implies Y(Z) &= z + \frac{Kz}{z - K}
\end{aligned}
$$

It's Z-inverse is $y[n] = K^n - \frac{1}{K}\delta[n+1]$.
Clearly, for this to converge, $|\frac{K}{Z}| < 1$, thus ROC being outside a circle of radius K.

■

Problem 2

Consider a discrete-time signal $x[n]$ passed through a bank of filters $h_k[n]$ where each filter is given by a modulated version of a baseband prototype filter $h[n]$

$$h_k[n] = h[n] \cdot exp\left[j(\frac{2n}{N})kn\right]$$

where $h[n]$ is a Hamming window, is assumed causal and lies over a duration $0 \leq n < N_w$, $\frac{2n}{N}$ is the frequency sampling factor. In this problem, you're asked to time-scale expand some simple input signals by time-scale expanding the filter bank outputs.

- State the constraint (with respect to the values N_w and N) such that the input $x[n]$ is recovered when the filter bank outputs are summed.
- If the input to the filter bank is the unit sample $\delta[n]$, then the output of each filter is a complex exponential with "envelope" $a_k[n] = h[n]$ and phase $\theta[n] = \frac{2n}{N}kn$. Suppose each complex exponential output is time-expanded by two by interpolation of its envelope and phase. Derive a new constraint (with respect to values N_w and N), so that the summed filter bank outputs equal $\delta[n]$.
- Suppose now that the filter bank input is

  $$x[n] = \delta[n] + \delta[n - n_0]$$

  and that the filter bank outputs are time-expanded as in part (b). Derive a sufficient condition on N_w, N and n_0 so that the summed filter bank output is given by

  $$y[n] = \delta[n] + \delta[n - n_0]$$

  that is, the unit samples are separated by $2n_0$ samples rather than n_0 samples.

Solution.
When the filter bank outputs are summed, for $x[n]$ to be recoverable,

$$\frac{2\pi}{N} < \frac{2\pi}{N_w}$$
$$N_w < N$$

Consider an impulse input, The output of each filter is:

$$\begin{aligned} X_k(n) &= \delta[n] * h[n]e^{j\frac{2n}{N}kn} \\ &= \sum_{m=0}^{n} \delta[m]h[n-m]e^{j\frac{2(n-m)}{N}k(n-m)} \\ &= h[n]e^{j\frac{2n}{N}kn} \end{aligned}$$

For interpolation by factor of 2, we add a zero-padding between each sample and then follow by a low-pass filter to attenuate all higher order aliases. Thus $N'_w = 2N_w \implies N'_w < N/2$. ■

Problem 3

Recall the "voiced" excitation to the digital model of speech production,

$$e[n] = \sum_{q=-\infty}^{\infty} \delta[n - qP]$$

Find the expression for the long term temporal auto-correlation, $r_e(\eta)$.

Solution.
Temporal auto-correlation refers to the successive values of the same signal. We have encountered short term auto-correlation function (STACF) during the lectures, so assuming no framing as the "long term" auto-correlation $r_e(\eta)$,

$$r_e(\eta) = \sum_{\forall m} e[m] \cdot e[m - \eta]$$

Let $e[m] \longleftrightarrow E(Z)$, therefore $R_e(Z) = E(Z) \cdot E(Z^{-1})$.

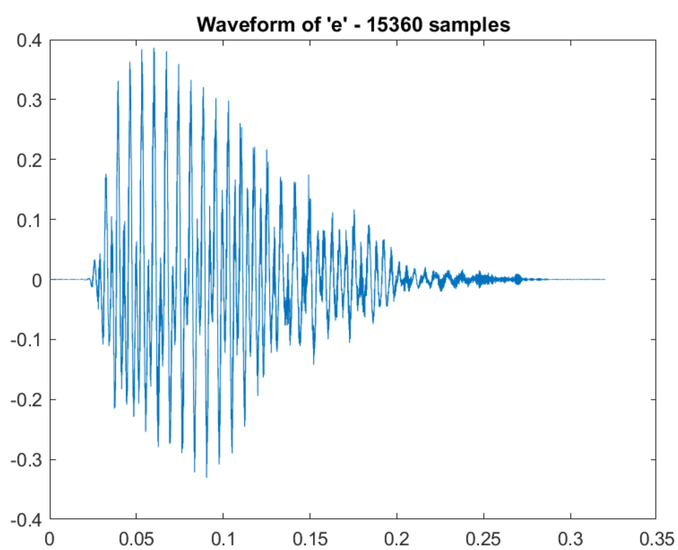
■

Problem 4

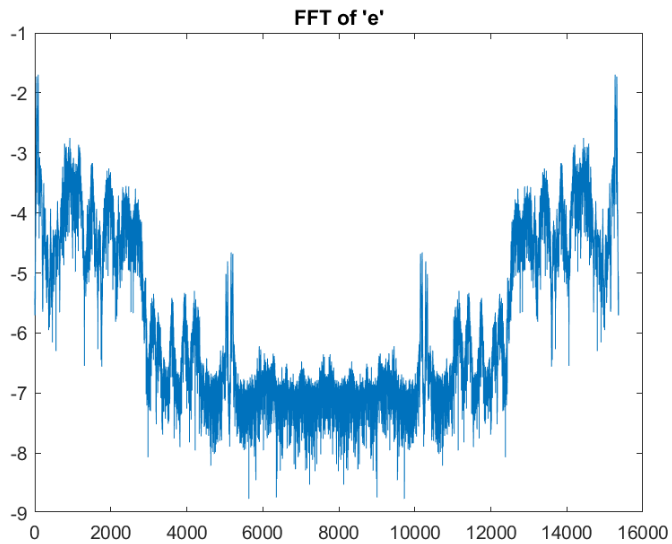
Programming Assignment: In this MATLAB exercise,

1. Perform the following operations with the vowel utterance of your choice:
   (a) Compute the short term auto-correlation function for a hamming window of $N = 512$ for $\eta = 0, 1, 2, \ldots, 256$.
   (b) Compute the $N = 512$ point magnitude spectrum of the waveform based on hamming window and stDFT. (the stDFT and conventional DFT are equivalent here because only the magnitude spectrum is required)
   (c) Repeat steps (a) and (b) after center clipping the waveform.
2. Comment the changes in both the auto-correlation and the spectrum. What do these changes indicate about the effects of the clipping operations on the waveform?
3. Estimate the pitch using the two auto-correlation results. Which results would you provide better performance in auto-correlation procedure?

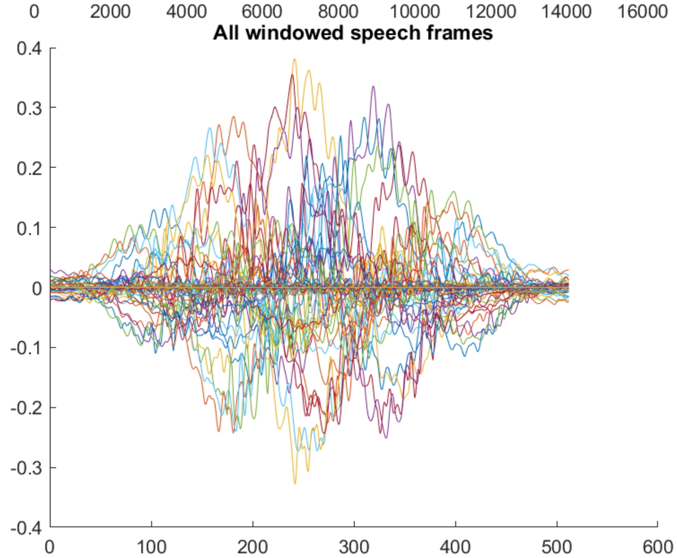
Solution.
The spectrum loses its earlier apparent formants when clipped. The STACF looks similar in both cases for the 20th window (out of 59 total windows), but upon checking with other windows, clipped waveform gave increasing peaks.
From the figure (6) obtained, $\Delta\eta = 150$, thus pitch is 150 where window length is 512 and sampling rate is 48000. Thus pitch is 320 Hz.



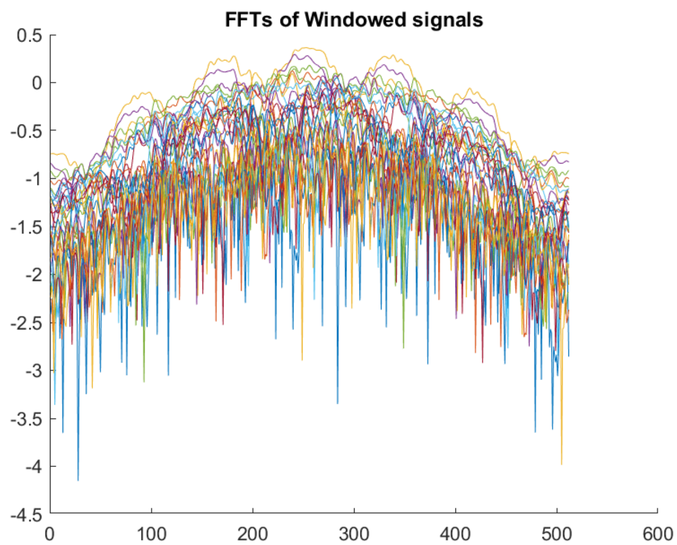
Waveform recorded



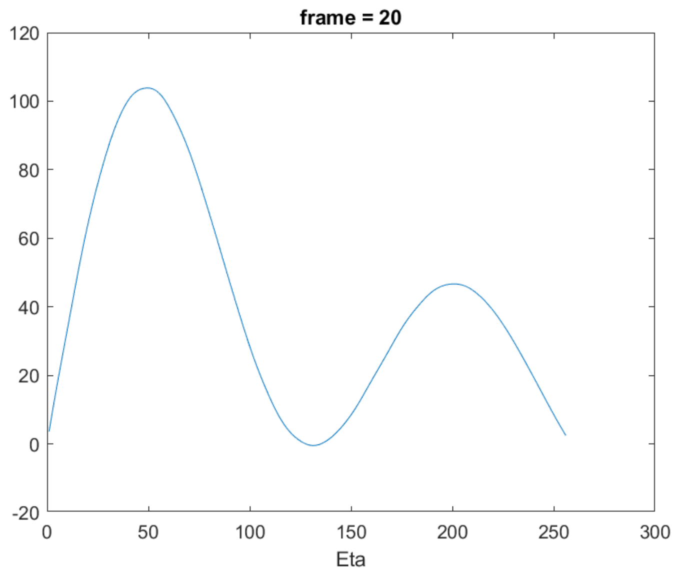
FFT of above recorded waveform



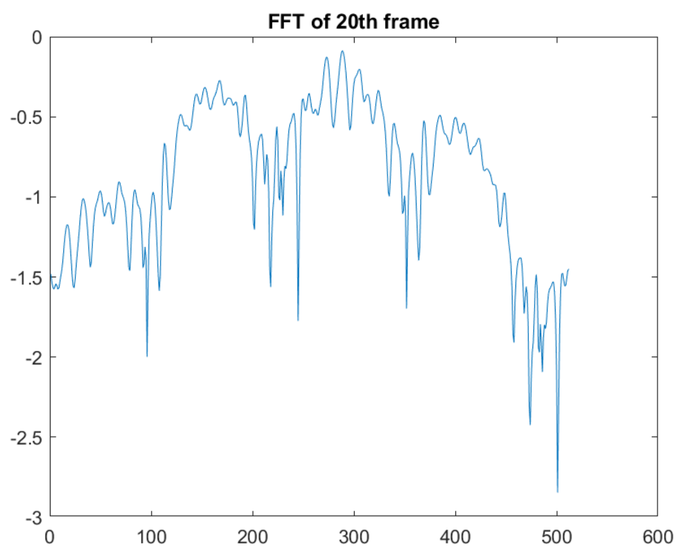
All windowed Signals



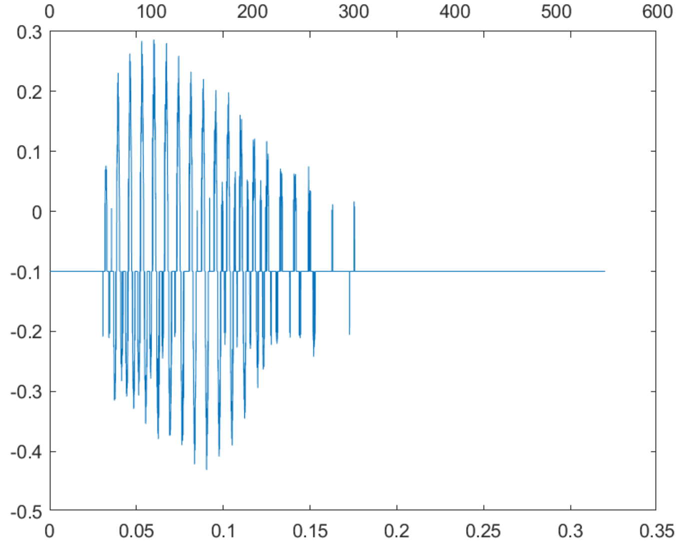
Windowed Signals FFT



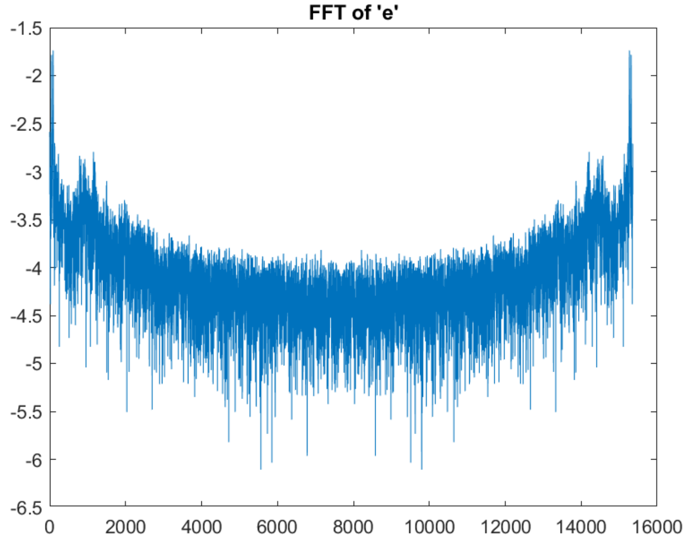
$R_{20}[\eta]$



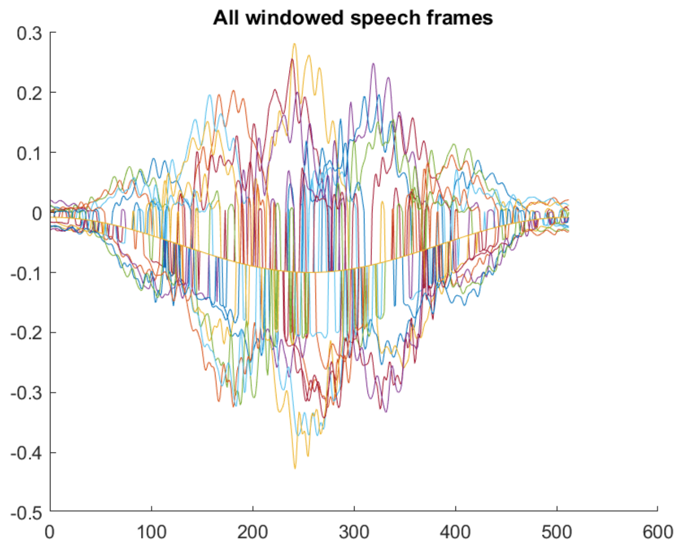
FFT of 20th frame



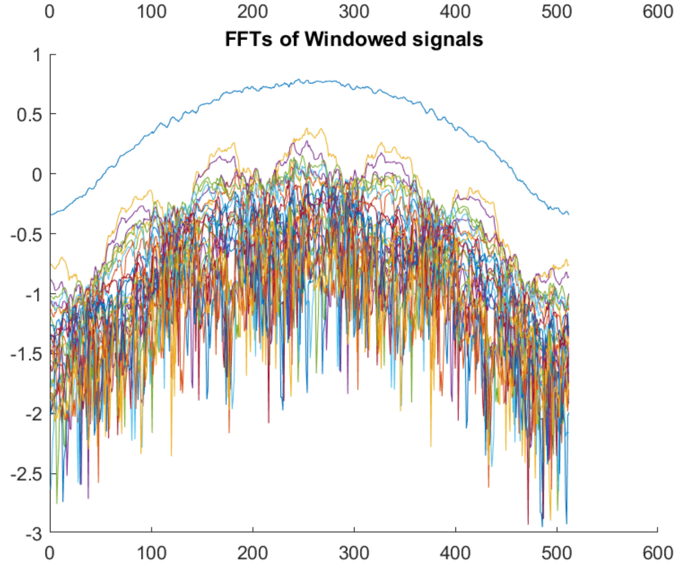
Center-clipped (threshold = 0.1)



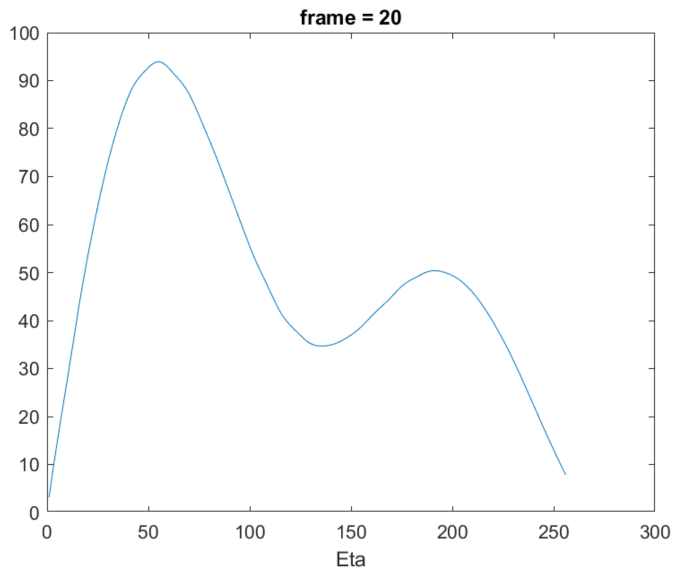
Clipped waveform FFT



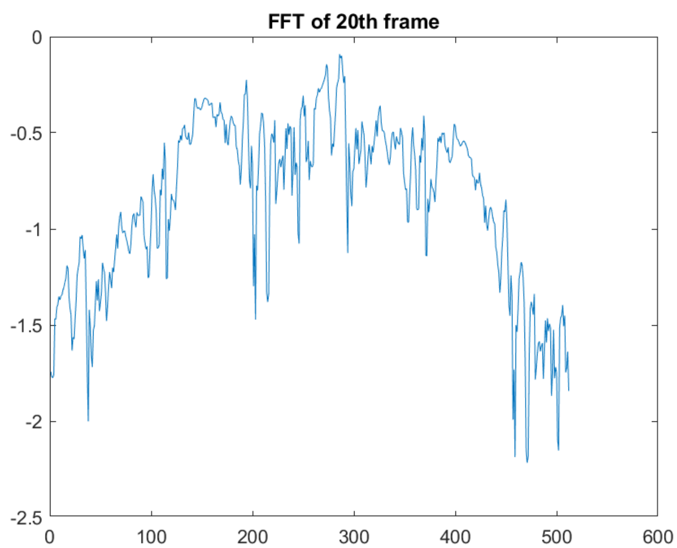
Clipped windowed signals



Clipped windowed signals FFT



Clipped $R_{20}[\eta]$



FFT of 20th clipped frame

∎

MATLAB Code

```
clc; clear; close all;
[y, fs] = audioread('e-vowel-utterance.wav');
y1 = y(:,1);
y1 = y1(27000:end);
y1 = y1(1:15360);
n = 1/fs:1/fs:length(y1)/fs;
l = length(y1);
figure;
plot(n, y1);
title("Waveform of 'e' - 15360 samples");

```

```
% Complete spectrum
fy1 = fft(y1);
figure;
plot(log10(abs(fy1)/l));
title("FFT of 'e'");

ham = hamming(512, "symmetric");
lh = length(ham);
% Assumed 50% overlap, with 512-bit window
y_split = zeros(length(1:lh/2:l-lh/2), lh);
yw = zeros(length(1:lh/2:l-lh/2), lh);
size(yw);
index = 1;
for i=1:lh/2:l-lh/2
    y_split(index, :) = (y1(i:i+lh-1));
    yw(index, :) = (y1(i:i+lh-1).*ham);
    index=index+1;
end
figure;
hold on;
for i=1:59
    plot(1:512, yw(i,:))
end
title("All windowed speech frames");
hold off;

eta = 256;
R = zeros(eta, 59);
% Extending window to 1536 length
ham_ = zeros(lh*3, 1);
ham_(lh:2*lh-1) = ham;
for i=1:59
    R_ = 0;
    y_ = ham_;
    if (i-1 < 1)
        y_(1:lh) = zeros(lh, 1);
    else
        y_(1:lh) = y_split(i-1, :);
    end

    y_(lh+1:2*lh) = y_split(i, :);

    if (i+1 > 59)
        y_(2*lh+1:3*lh) = zeros(lh, 1);
    else
```

```
            y_(2*lh+1:3*lh) = y_split(i+1, :);
        end
        for eta=1:256
            for j=1:lh
                R_ = R_ + ham_(2*lh-j)*y_(j)*ham_(2*lh-j-eta)*y_(j+eta);
            end
            R(eta, i) = R_;
        end
end
nthframe = 20;
figure;
plot(R(:,nthframe));
title("frame = "+ nthframe);
xlabel("Eta");

% Now computing the fft of each frame
fR = fft(yw);
figure;
plot(log10(abs(fR(nthframe, :))))
title("FFT of "+nthframe+"th frame")
figure;
hold on;
for i=1:59
    plot(log10(abs(fR(i, :))));
end
title("FFTs of Windowed signals")
hold off;

% Now start by clipping off y1
clip_amp = 0.1;
for i=1:length(y1)
    if(y1(i)<clip_amp && y1(i)>0)
        y1(i)=0;
    elseif(y1(i)>-clip_amp && y1(i)<0)
        y1(i)=0;
    else
        y1(i)=y1(i)-clip_amp;
    end
end
plot(n, y1);
% Now re-run from line 10
```