Assessment (ECE2111)

Lab 03 Result Document

Name: Tan Jin Chun

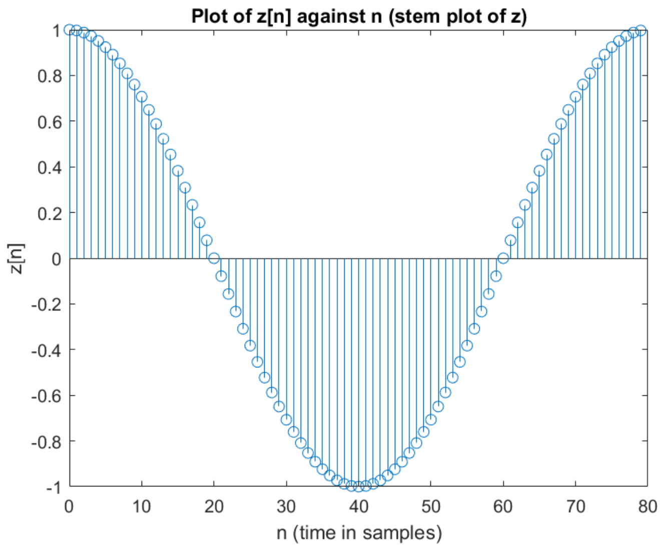
Student ID: 32194471

ECE2111 lab3 results document:

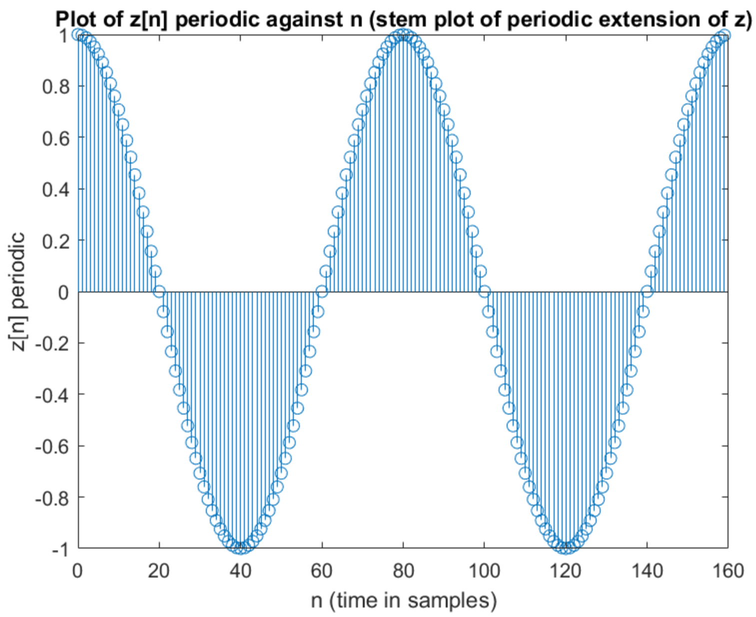
Name: Tan Jin Chun
Names of all students you discussed this lab with: Chong Yen Juin, Loh Jia Quan, Huan Meng Hui, Ku Yew Siang

Section 1:

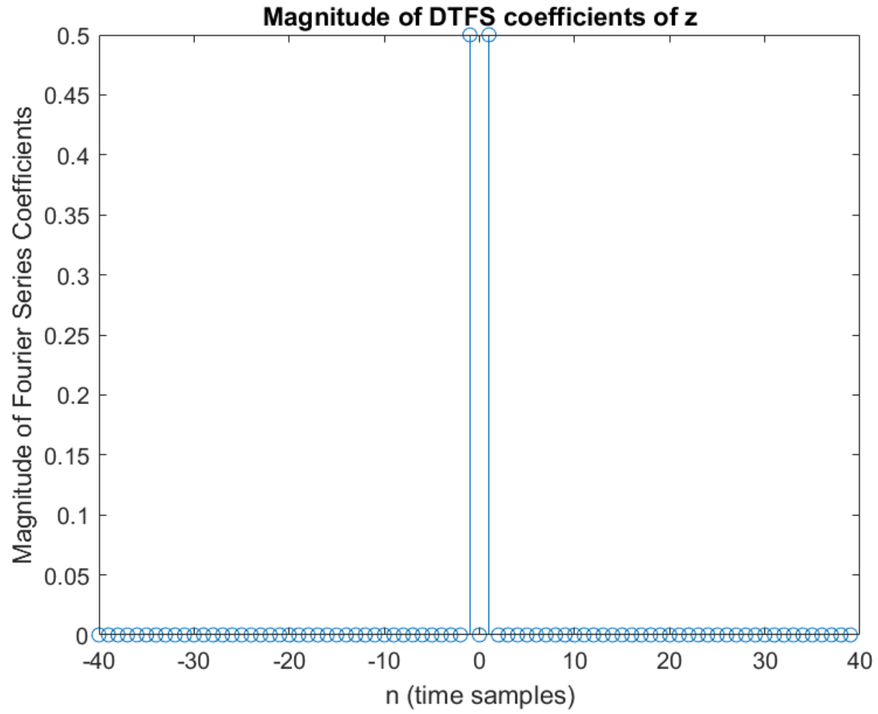
Question 1: Include your stem plot of z[n] vs n for 0 ≤ n ≤ 79, below:



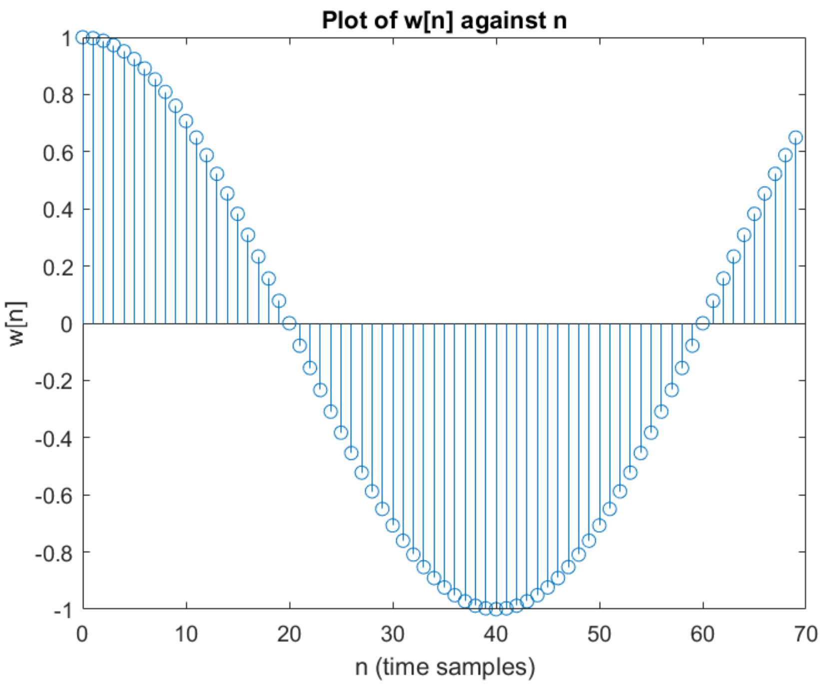
Question 2: Plot $z_{periodic}[n]$ vs n for 0 ≤ n ≤ 159. Find a simple formula for $z_{periodic}[n]$ as a function of n.



Question 3: Plot the magnitude of the discrete-time Fourier series coefficients of $z_{periodic}$. Include one period of the coefficients in your plot, and center the plot at k = 0.

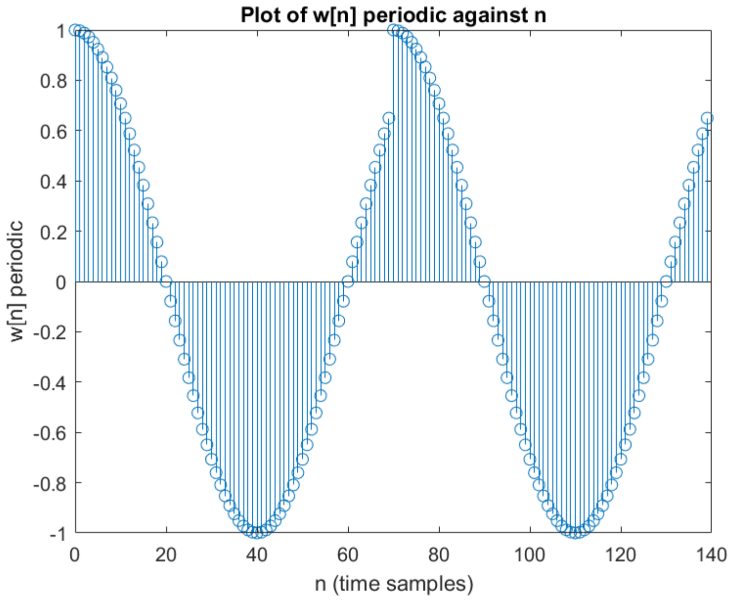


Question 4: Make a stem plot of w[n] vs n for 0 ≤ n ≤ 69.

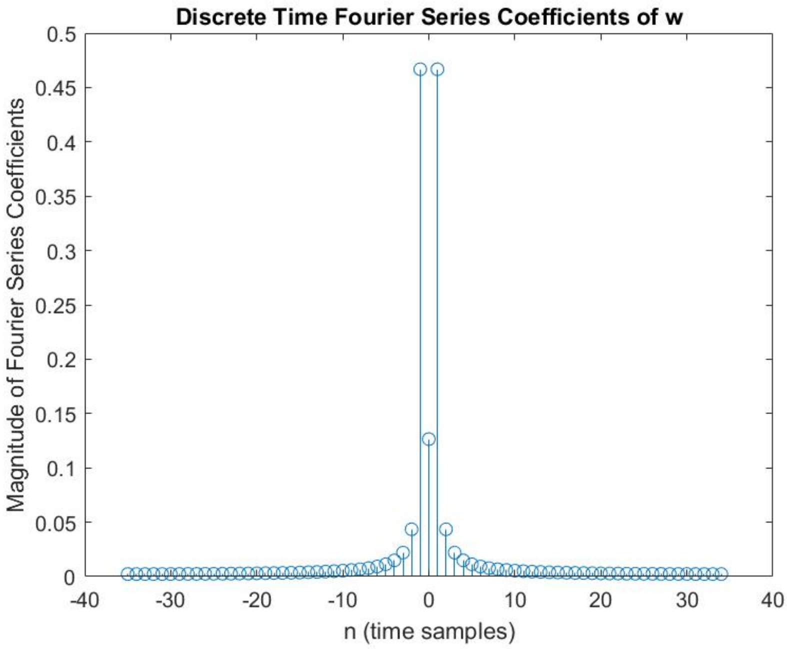


Question 5: Plot $w_{periodic}[n]$ vs n for $0 \leq n \leq 139$. Comment on the difference between the plots of $w_{periodic}$ and $z_{periodic}$

The plot for wperiodic is smooth and zperiodic is not a smooth plot.



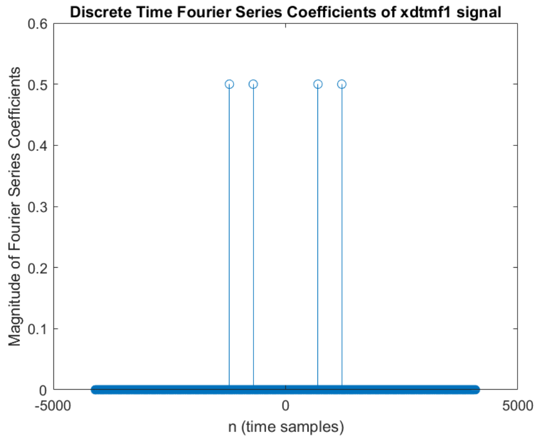
Question 6: Plot the magnitude of the discrete-time Fourier series coefficients of $w_{periodic}$. Include one period of the coefficients in your plot, and center the plot at k = 0. How does the plot differ from the plot of the magnitude of the discrete-time Fourier series coefficients of $z_{periodic}$? Why do you think this is the case?

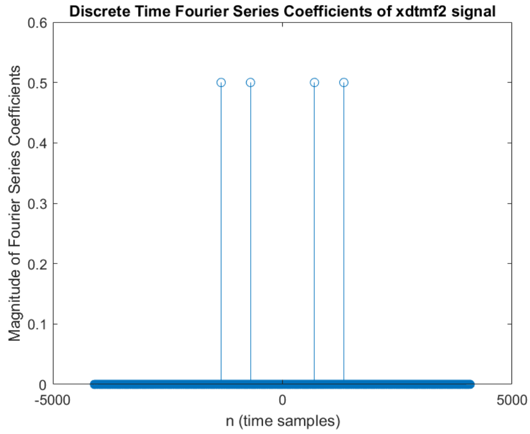


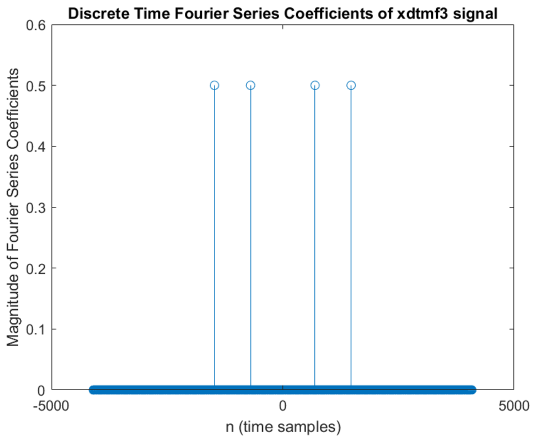
The plot differs in such a way that the discrete-time Fourier series coefficients of w periodic looks more like a bell-shape curve than z periodic.

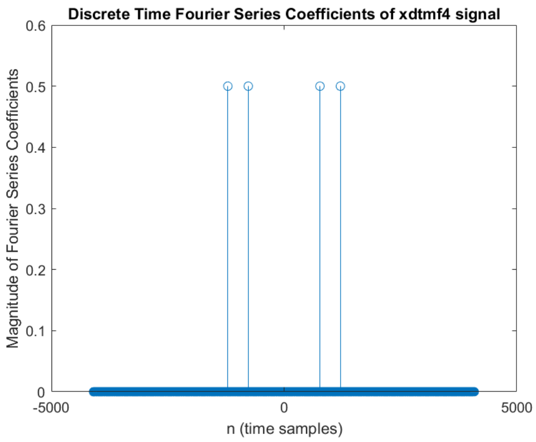
Section 2:

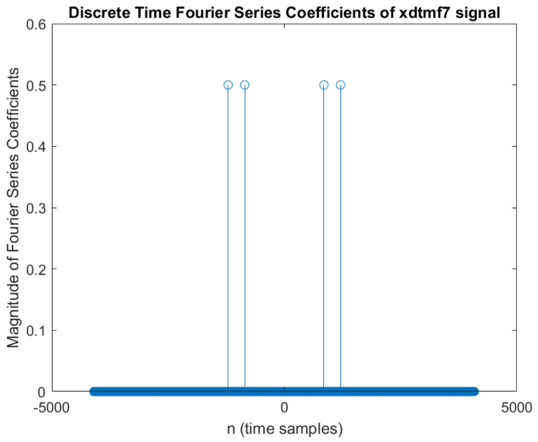
Question 2(b): Plot the magnitude of the discrete-time Fourier series coefficients of the signal. Use a stem plot. Arrange the horizontal axis so that the zero frequency component is shown in the center of the plot.

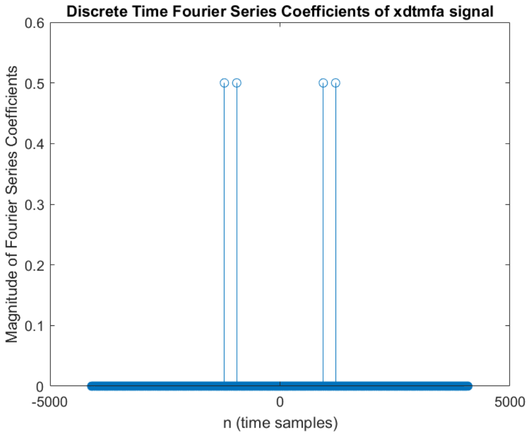


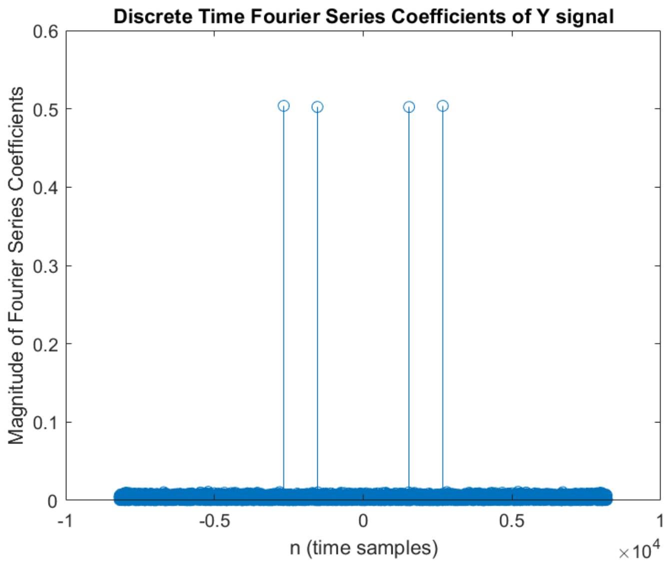












Question 2(c)
Find the indices of the discrete-time Fourier series coefficients of the signal that have large magnitude.

The indices that have large magnitude for xdtmf1 signal are -1209, -697, 697, 1209
The indices that have large magnitude for xdtmf2 signal are -1336, -697, 697, 1336
The indices that have large magnitude for xdtmf3 signal are -1477, -697, 697, 1477
The indices that have large magnitude for xdtmf4 signal are -1209, -770, 770, 1209
The indices that have large magnitude for xdtmf7 signal are -1209, -852, 852, 1209
The indices that have large magnitude for xdtmfa signal are -1209, -941, 941, 1209

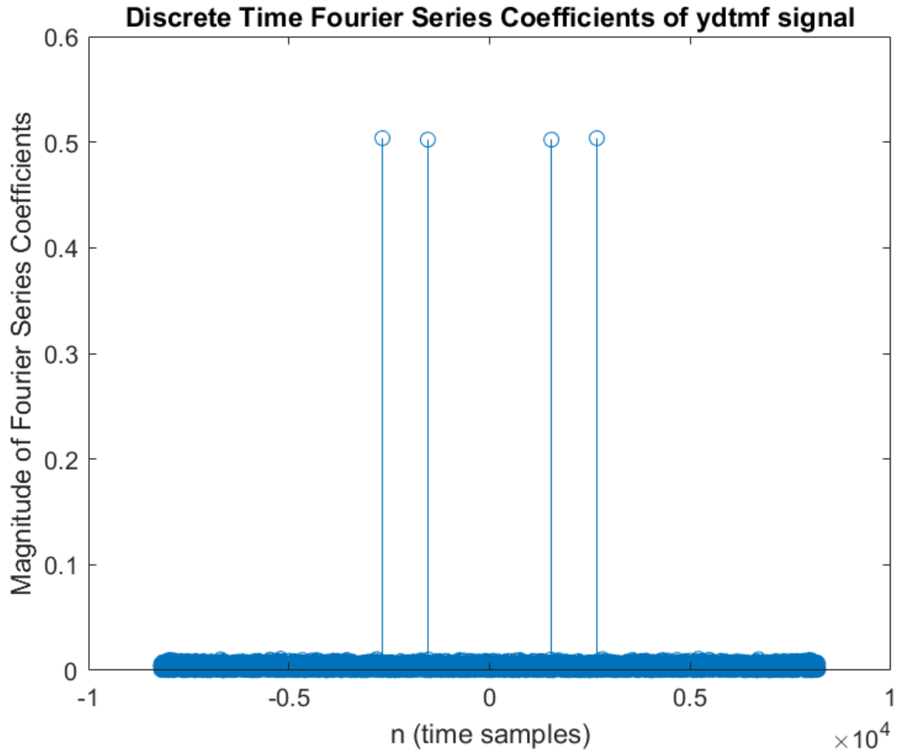
Question 3: Determine the frequencies ω_{ri} for i = 1,2,3,4 and ω_{cj} for j = 1,2,3.

Frequency when i = 1 is 697 Hz, so ωri = 697*2π
Frequency when i = 2 is 770 Hz, so ωri = 770*2π
Frequency when i = 3 is 852 Hz, so ωri = 852*2π
Frequency when i = 4 is 941 Hz, so ωri = 941*2π
Frequency when j = 1 is 1209 Hz, so ωri = 1209*2π
Frequency when j = 2 is 1336 Hz, so ωri = 1336*2π
Frequency when j = 3 is 1477 Hz, so ωri = 1477*2π

Question 4: What is the duration of the signal `ydtmf`*?*

2 seconds. 16384/8192 = 2 s

Question 6: Plot the magnitude of the discrete-time Fourier series coefficients of `ydtmf`*. Use a stem plot. Arrange the horizontal axis so that the zero frequency component is shown in the center of the plot.*



Question 7: Find the indices of the discrete-time Fourier series coefficient of `ydtmf` *that have large magnitude.*

The indices that have large magnitude for ydtmf signal are -2672, -1540, 1540, 2672.

Question 8: Which button press does `ydtmf` *correspond to?*

The button press ydtmf correspond to the number 5. Divide the indices by 2 and the indices obtained would be -1336, -770, 770, 1336 . This would be the 2 row and the 2 column which corresponds to the number 5. The indices need to be divided by 2 since the sampling rate is doubled that of the current sampling rate.

Section 3:

Question 2: At the sampling rate `Fst`*, how many samples correspond to 30ms of audio?*

2880 samples. 96000*30e-3 = 2880 samples.

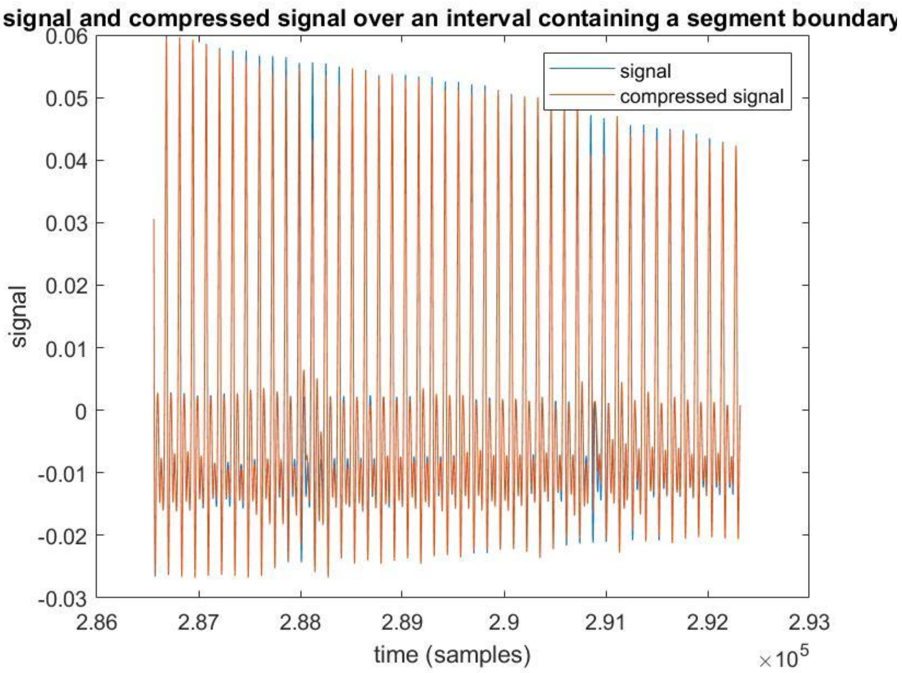
Question 6: Explain how the command
`Ysegtrunc = Yseg.*(abs(Yseg) > threshold*maxval);`
works.

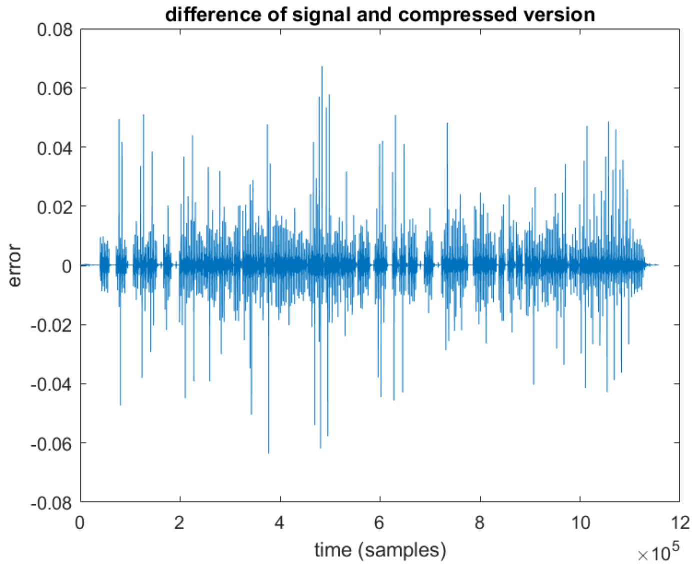
The code above take the magnitude of Yseg within 0.01 of the max magnitude.
Yseg is a M by P Matrix that elementwise multiplication with the right terms in the bracket
abs(Yseg) takes the absolute value of each entry in the matrix Yseg.
When Yseg undergoes elementwise multiplication with the M-by-P logical matrix consisting 0s and 1s, the element multiply with 0 will not be kept only those multiplying with 1 will be kept

Question 7: By typing `whos Ysegtrunc` *find out how much memory this variable requires. Compare this with the memory required to store* `y`*.*

Ysegtrunc requires 833816 memory.
Y requires 9262080 memory.

Question 8: Choose a time- interval on which `ytrunc` *and* `y` *are both non-zero, that crosses a 30ms segment boundary. Over this time-interval, plot both signals. What do you notice? Plot the difference between* `ytrunc` *and* `y`*. What do you notice? What do you think is the basic problem with our compression method?*





The plot shows there is a significant variation between original y signal and the truncated y signal especially within the 30ms interval.

This compression causes the original signal lose its resolution through an approximation over every 30ms. This is because the threshold is too large. The threshold is 0.01 of the largest magnitude of the Fourier Series Coefficient. A large threshold would cause many coefficients to be smaller, causing it to be invalid. The removal of the coefficients approximates the output, resulting in a static signal.

Code for section 1:

Code from section 1.2:

Paste your script in here.

```
% Written by Tan Jin Chun
% Last Modified : 2/9/2021
% Lab03T01

clear all;close all;clc

%% Question 1
% Initialising the variables
n = 0:79;

% Sampling frequency
Fs = 160;

% Time (Converting continuous time to discrete time)
t = n./Fs;

% Equation
x = cos(4*pi.*t);

% Initialising variable y
y = size(n);

% Need to generate the signal from 1 to 80, use a for loop for that
% (u[n]-u[n-80])
for i = 1:80

    % All of the rows from 1 to 80 will be filled with ones
    y(1,i) = 1;

end

% Getting the formula of z
z = x.*y;

% Plotting the stem plot
stem(n,z);
xlabel("n (time in samples)");
ylabel("z[n]");
title("Plot of z[n] against n (stem plot of z)");

%% Question 2
% Initialising the variables
n_extend = 0:159;

% Converting from discrete time to continous time
t = n_extend./Fs;

% Initialising variable z_periodic
z_periodic = size(n_extend);

% Using back the old formula above
```

```
% First half is the same
z_periodic(1,1:80) = z;

% The second half is also the same
z_periodic(1,81:160) = z;

% Plotting the stem plot
figure
stem(n_extend,z_periodic);
xlabel("n (time in samples)");
ylabel("z[n] periodic");
title("Plot of z[n] periodic against n (stem plot of periodic
extension of z)");

%% Question 3
% Period of 80
% Getting the length of z_periodic
N = length(z);

% Computing the discrete time Fourier Series Coefficient
% Use the fft() function
X = (1/N)*fft(z);

% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
stem(-floor(N/2):(N-1-floor(N/2)), fftshift(abs(X)));

% Labelling the graph
title("Magnitude of DTFS coefficients of z");
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");

% % Plotting the phase of the Fourier Series
% hold on
% stem(-floor(N/2):(N-1-floor(N/2)), fftshift(angle(X)));
% hold off
%%  Question 4
% Stem plot for w[n] = xs[n](u[n]-u[n-70])
n_new = 0:69;

% Time (Converting continuous time to discrete time)
t_new = n_new./Fs;

% Equation
x_new = cos(4*pi.*t_new);

% Initialising variable y
y_new = size(n_new);

% Need to generate the signal from 1 to 70, use a for loop for that
% (u[n]-u[n-70])
for i = 1:70

    % All of the rows from 1 to 80 will be filled with ones
```

```
    y_new(1,i) = 1;

end

% Getting the formula of z
w = x_new.*y_new;

% Plotting the stem plot
figure
stem(n_new,w);
xlabel("n (time samples)");
ylabel("w[n]");
title("Plot of w[n] against n");

%% Question 5
% Initialising the variables
n_extend_new = 0:139;

% Converting from discrete time to continous time
t_extend_new = n_extend_new./Fs;

% Equation
x = cos(4*pi.*t_extend_new);

% Initialising variable z_periodic
w_periodic = size(n_extend_new);

% Using back the old formula above
% First half is the same
w_periodic(1,1:70) = w;

% The second half is also the same
w_periodic(1,71:140) = w;

% Plotting the stem plot
figure
stem(n_extend_new,w_periodic);
xlabel("n (time samples)");
ylabel("w[n] periodic");
title("Plot of w[n] periodic against n");

%% Question 6
% Period of 70
N = length(w);

% Computing the discrete time Fourier Series Coefficient
% Use the fft() function
X_new = (1/N)*fft(w);

% Plotting the magnitude of the fourier coefficient
figure;

% Plotting the magnitude of the fourier coefficient
stem(-floor((N)/2):(N-1-floor((N)/2)), fftshift(abs(X_new)));
```

```
% Labelling the graph
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");
title("Discrete Time Fourier Series Coefficients of z_periodic");

% Plotting the phase of the Fourier Series
% hold on
% stem(-floor((N-1)/2):(N-1-floor((N-1)/2)),
fftshift(angle(X_new)));
% hold off
```

Code for section 2

Code from section 2.1:

Paste your script in here.

```
% Written by Tan Jin Chun
% Last Modified : 2/9/2021
% Lab03T02

clear all;close all;clc

%% Question 1

% Initialising the variable
Fs = 8192;
Ts = 1/Fs;

% Time (Converting continuous time to discrete time)
t = 0:Ts:1;

% Loading the files
load dtmfclean.mat

%% Question 2
% Question 2(a)
% Listening to the sound (total of 6 sounds)
% soundsc(xdtmf1);
% soundsc(xdtmf2);
% soundsc(xdtmf3);
% soundsc(xdtmf4);
% soundsc(xdtmf7);
% soundsc(xdtmfa);

% Question 2(b)
% Getting the lengths of the signals
N1 = length(xdtmf1);
N2 = length(xdtmf2);
N3 = length(xdtmf3);
N4 = length(xdtmf4);
N7 = length(xdtmf7);
Na = length(xdtmfa);

% Computing the discrete time Fourier Series Coefficient
% Use the fft() function
X1 = (1/N1)*fft(xdtmf1);
X2 = (1/N2)*fft(xdtmf2);
X3 = (1/N3)*fft(xdtmf3);
X4 = (1/N4)*fft(xdtmf4);
X7 = (1/N7)*fft(xdtmf7);
Xa = (1/Na)*fft(xdtmfa);

% Figure X1
% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
x1 = -floor(N1/2):(N1-1-floor(N1/2));
```

```
y1 = fftshift(abs(X1));
stem(x1, y1);

% Finding the indices of the discrete time Fourier Series that have
a large
% magnitude
maxval = max(y1);
idx = find(y1 == maxval)

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of xdtmf1 signal");
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");

% Figure X2
% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
stem(-floor(N2/2):(N2-1-floor(N2/2)), fftshift(abs(X2)));

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of xdtmf2 signal");
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");

% Figure X3
% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
stem(-floor(N3/2):(N3-1-floor(N3/2)), fftshift(abs(X3)));

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of xdtmf3 signal");
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");

% Figure X4
% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
stem(-floor(N4/2):(N4-1-floor(N4/2)), fftshift(abs(X4)));

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of xdtmf4 signal");
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");

% Figure X7
% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
stem(-floor(N7/2):(N7-1-floor(N7/2)), fftshift(abs(X7)));

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of xdtmf7 signal");
```

```
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");

% Figure Xa
% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
stem(-floor(Na/2):(Na-1-floor(Na/2)), fftshift(abs(Xa)));

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of xdtmfa signal");
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");

%% Question 3

%% Question 4
% Loading the files for dtmfnoisy.mat
load dtmfnoisy.mat

% The duration
% number of samples divided by sampling frequency
duration = length(ydtmf)/Fs;

% Ans is 2

%% Question 5
% Listening to the sounds
% soundsc(ydtmf);

%% Question 6
% Getting the lengths of the signals
Y = length(ydtmf);

% Computing the discrete time Fourier Series Coefficient
% Use the fft() function
Y1 = (1/Y)*fft(ydtmf);

% Figure y
% Plotting the magnitude of the fourier coefficient
figure
% Plotting the magnitude
stem(-floor(Y/2):(Y-1-floor(Y/2)), fftshift(abs(Y1)));

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of ydtmf signal");
xlabel("n (time samples)");
ylabel("Magnitude of Fourier Series Coefficients");
```

Code for section 3

Code from section 3.1:

Paste your script in here.

```
% Written by Tan Jin Chun
% Last Modified : 2/9/2021
% Lab03T03

clear all;close all;clc

%% Question 1

% Loading the signal
[y, Fst] = audioread('trumpet.wav');

%% Question 2
% Getting the total number of samples
% Sampling Rate * Time
M = Fst * 30e-3;

%% Question 3
% Getting the value of P
P = length(y)/M;

% Using the reshape built-in function to reshape my signal
yseg = reshape(y, [M, P]);

%% Question 4
% Yseg corresponds to the discrete time Fourier Series coefficients
% Getting the lengths of the signals
N = length(y);

% Preallocating Yseg for effieciency
Yseg = zeros(size(yseg));

% Computing the discrete time Fourier Series Coefficient for yseg
% Use the fft() function
% Using a for loop
for i = 1:P
    Yseg(:,i) = (1/M)*fft(yseg(:,i));
end

% Using the reshape built-in function to reshape my signal
% Yseg = reshape(X,[M,P]);
% figure;
% stem(-floor(N/2):(N-1-floor(N/2)), fftshift(abs(Yseg)));
% plot(Yseg)
%% Question 5
% Getting the maximum value of Yseg
maxval = max(max(abs(Yseg)));

%% Question 6
% Initialising the variable
threshold = 0.01;
```

```
% Making a new array of Fourier Series coefficients
Ysegtrunc = Yseg.*(abs(Yseg) > threshold*maxval);
% plot(Ysegtrunc);

%% Question 7
% Converting a sparse storage format in Matlab
Ysegtrunc = sparse(Ysegtrunc);

% Finding out how much memory the Ysegtrunc variable requires
% It requires 833816 bytes
whos Ysegtrunc

% Comparing with the memory that is required to store y
% It requires 27789464 bytes
whos y

%% Question 8
% Question 8(a)
% Converting Ysegtrunc back from the sparse data format to normal
data
% format
Ysegtrunc = full(Ysegtrunc);

% Converting the columns of Ysegtrunc back into time domain signal
using
% ifft
% Using a for loop
% Preallocation
X = zeros(size(Ysegtrunc));

% Using a for loop
for i = 1:P
    X(:,i) = ifft(M*Ysegtrunc(:,i));
end

% % Question 8(b)
% % Reshaping back the signal into a single vector
ytrunc = reshape(X, [1, M*P]);

% Question 8(c)
% Playing signal y
% soundsc(y);
% pause
%
% % Playing signal ytrunc
% soundsc(ytrunc);
% pause

% Question 8(d)
% First Plot
time_interval = 99.5*M:101.5*M;
figure;
plot(time_interval, y(time_interval)');
xlabel("time (samples)");
ylabel("signal");
```

```
title("signal and compressed signal over an interval containing a
segment boundary");
hold on;
plot(time_interval, ytrunc(time_interval));
legend("signal","compressed signal");
hold off;

% Second Plot
% Initialising the variables
time = 0:length(y)-1;
error = ytrunc - y';
figure;
plot(time, error);
xlabel("time (samples)");
ylabel("error");
title("difference of signal and compressed version");
```