Table of Contents

Frequency Division Multiplexing for DSB-SC

This documents describes/implements the Frequency Division Multiplexing

```
%for DSB-SC modulation and
% demodulation of a song signal and a triangle signal.
```

Prepared for ELEC 301

by Beyzanur Çoban 64763

01.04.2020

Program Initialization

```
%Clear Variables and Close All Figure Windows

% Clear all previous variables
clear
% Close all previous figure windows
close all
```

Read Song File

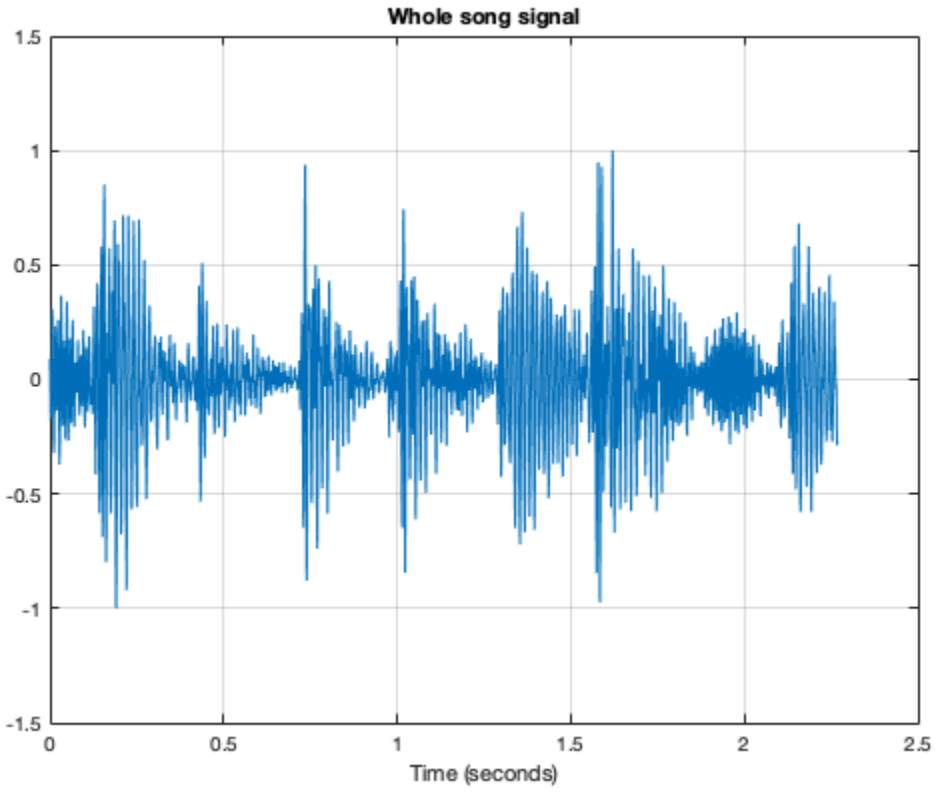
song.mat contains **song** variable containing Song samples and **Fs** which is the sampling frequency

```
% Load the song file
load song.mat
% song is the song samples
% Fs is the sampling frequency

% Transform the song to low rate sampling for listening (sound command
```

```
% requires sampling rate to be less than 44K
songlowrate=downsample(song,10);
% Listen to
sound(songlowrate,Fs/10);
% convert it to row array
song=reshape(song,1,length(song));
% Sampling Period
Ts=1/Fs;
% Sampling times
t=(0:1:(length(song)-1))*Ts;
```

Display the whole song

```
% Display the whole song
figure(1)
plot(t,song);
grid
title('Whole song signal');
xlabel('Time (seconds)');
```



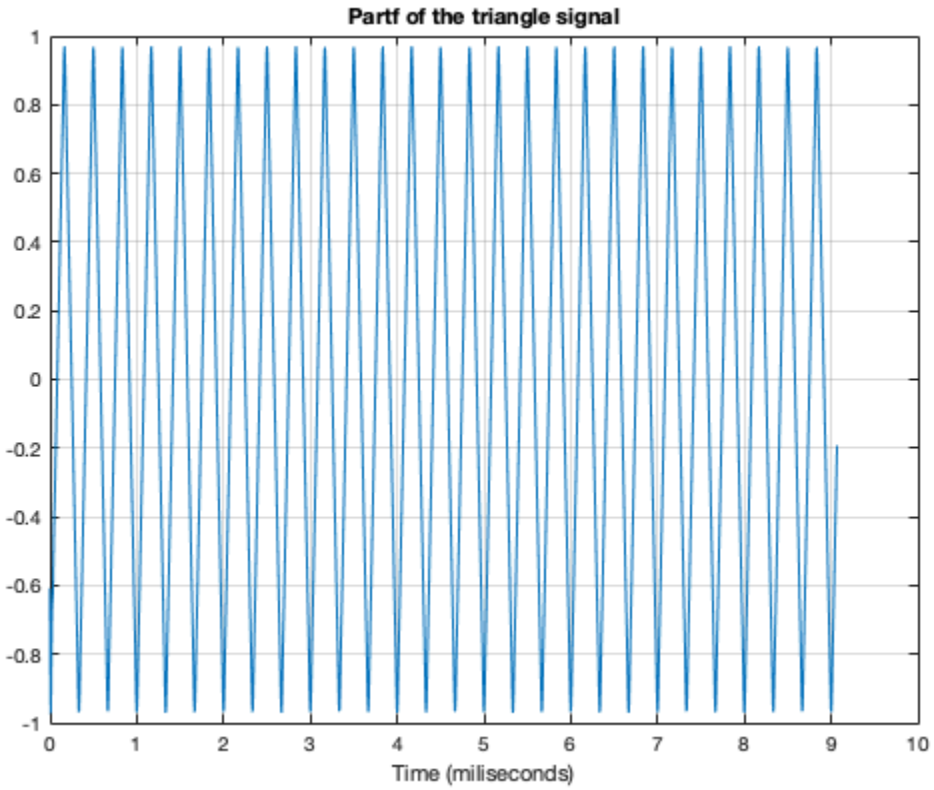
Create the triangle wave

```
% Fundamental freqeuncy
Fstr = 3000;
```

```
tr=sawtooth(2*pi*Fstr*t,1/2);
tr = lowpass(tr, 30e3, Fs);
```

Display the part of the triangle signal

```
% In order to get a clear vision, I plot only a part
% Display the part of the triangle signal
figure(2)
plot(t(1:3000)*1000, tr(1:3000));
grid
title('Partf of the triangle signal')
xlabel('Time (miliseconds)')
```



Generate Modulated Signal

Generate carrier signal and multiply with the song signal to obtain DSB-SC modulated waveform

Carrier frequency for song signal:

$f_c = 60kHz$

```
fc=60e3; % 60 kHz;
```

Generate carrier signal and multiply with the triangle signal to obtain DSB-SC modulated waveform

Carrier frequency for triangle signal;

$f_ctr = 120kHz$

```
fctr= 120e3;
```

Carrier signal for song signal:

$c(t) = cos(2\pi f_c t)$

```
c=cos(2*pi*fc*t);
```

Carrier signal for triangle

$ctr(t) = cos(2\pi f_c trt)$

```
ctr=cos(2*pi*fctr*t);
```

DSB-SC Modulated waveforms

$x(t) = s(t)c(t)$

```
x=song.*c;
```

$xtr(t) = tr(t)ctr(t)$

```
xtr=tr.*ctr;
```

Output of the transmitter

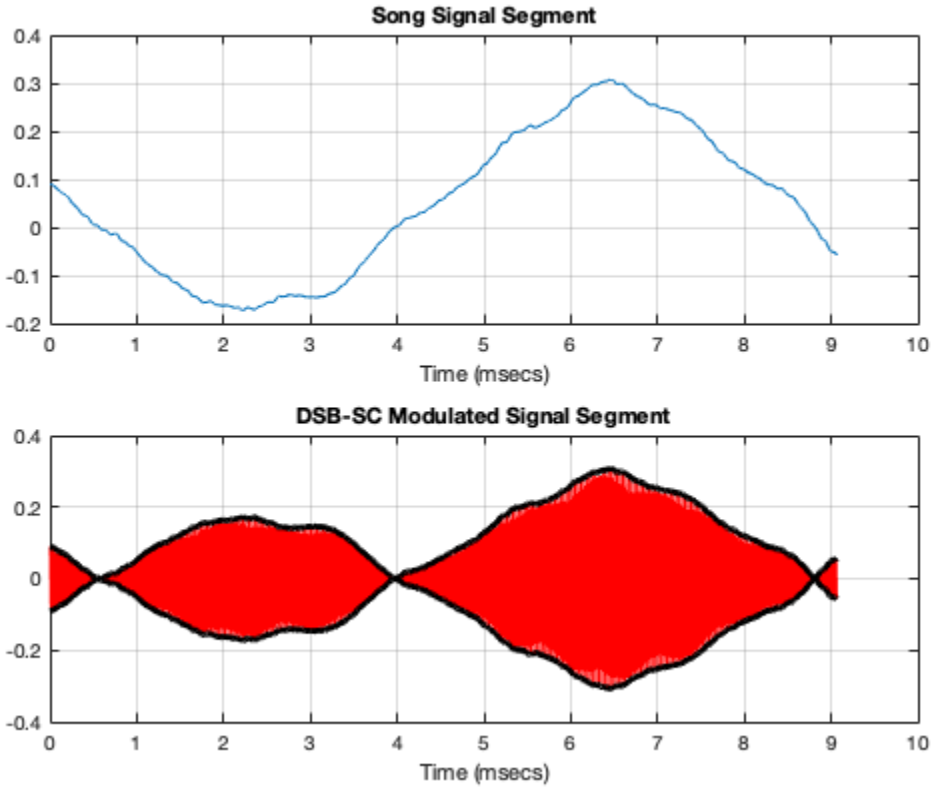
```
x_final = x + xtr;
```

Display the Segments of Signal and Modulated Signal (Song)

Display small section of the original signal and then the DSB-SC modulated version

```
figure(3)
% plot the song segment (for about 3000 samples)
subplot(2,1,1)
plot(t(1:3000)*1000, song(1:3000));
xlabel('Time (msecs)')
title('Song Signal Segment')
grid

subplot(2,1,2)
% plot the modulated signal
plot(t(1:3000)*1000,x(1:3000),'r');
hold on
% plot also positive and negative envelopes
p1=plot(t(1:3000)*1000,song(1:3000),'k');
p2=plot(t(1:3000)*1000,-song(1:3000),'k');
xlabel('Time (msecs)')
set(p1,'LineWidth',3)
set(p2,'LineWidth',3)
grid
```

```
title('DSB-SC Modulated Signal Segment')
```

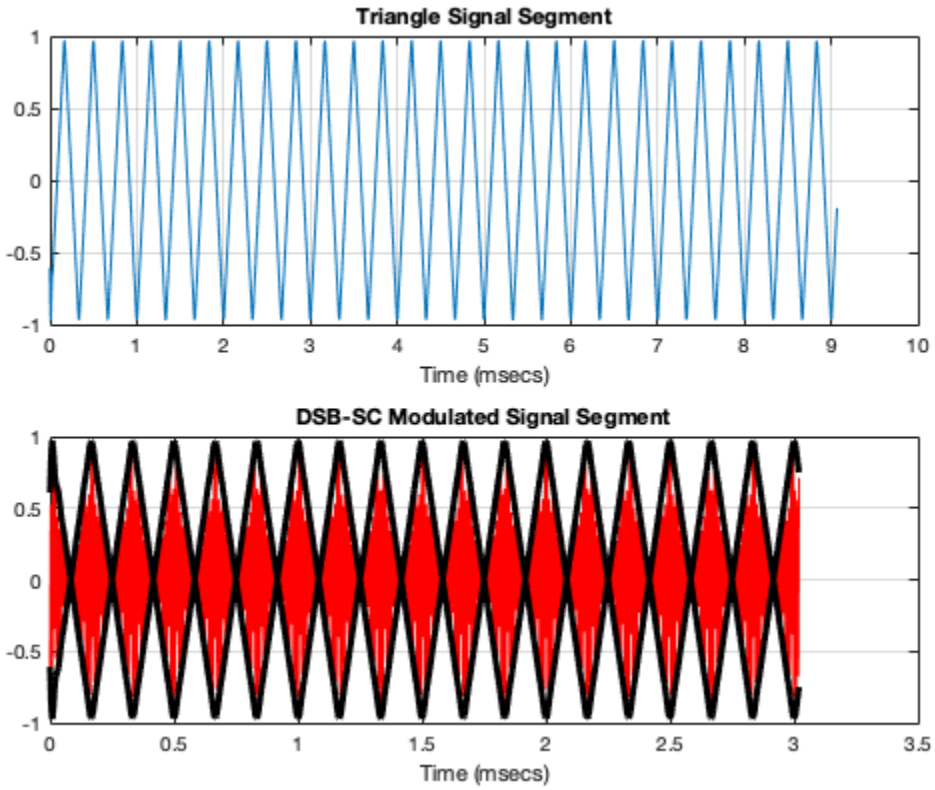


Display the Segments of Signal and Modulated Signal (Triangle)

Display small section of the original triangle signal and then the DSB-SC modulated version

```
figure(4)
% plot the triangle segment (for about 3000 samples)
subplot(2,1,1)
plot(t(1:3000)*1000, tr(1:3000));
xlabel('Time (msecs)')
title('Triangle Signal Segment')
grid

subplot(2,1,2)
% plot the modulated signal
plot(t(1:1000)*1000,xtr(1:1000),'r');
hold on
% plot also positive and negative envelopes
p1=plot(t(1:1000)*1000,tr(1:1000),'k');
p2=plot(t(1:1000)*1000,-tr(1:1000),'k');
xlabel('Time (msecs)')
set(p1,'LineWidth',3)
set(p2,'LineWidth',3)
```

```
grid
title('DSB-SC Modulated Signal Segment')
```



The DSB-SC Receiver Processing

Coherent DSB-SC Receiver operation

First multiply with the receiver carrier (which is assumed to be in phase)

$$y(t) = 2x_final(t)c(t)$$

```
y=2*x_final.*c;
```

$$ytr(t) = 2x_final(t)ctr(t)$$

```
ytr=2*x_final.*ctr;
```

Then low pass filter this signals

$$z(t) = y(t) * h_{LP}(t)$$

```
z = lowpass(y,30e3,Fs);
```

$$ztr(t) = ytr(t) * h_{LP}(t)$$

```
ztr = lowpass(ytr, 300, Fstr);
```

Fourier Transforms of Song, Modulated and Demodulated Signals

Calculate and Display the Fourier Transforms of the song,modulated and demodulated signals

Calculate the Fourier Transform of the song signal

```
[ftsong,freqs]=fouriertransform(song, Fs);
```

Calculate the FT of the triangle signal

```
[fttr, freqs] = fouriertransform(tr, Fs);
```

Calculate the Fourier Transform of the DSB-SC signal of song signal

```
[ftx,freqs]=fouriertransform(x,Fs);
```

Calculate the Fourier Transform of the DSB-SC signal of triangle signal

```
[ftxtr,freqs]=fouriertransform(xtr,Fs);
```

Calculate the FT of the transmitter output

```
[ftx_final,freqs]=fouriertransform(x_final,Fs);
```

Calculate Fourier Transform after receiver carrier multiplication of song signal

```
[fty,freqs]=fouriertransform(y,Fs);
```

Calculate Fourier Transform after receiver carrier multiplication of triangle signal

```
[ftytr,freqs]=fouriertransform(ytr,Fs);
```

Calculate Fourier Transform of the receiver output of song signal

```
[FTz,freqs]=fouriertransform(z,Fs);
```

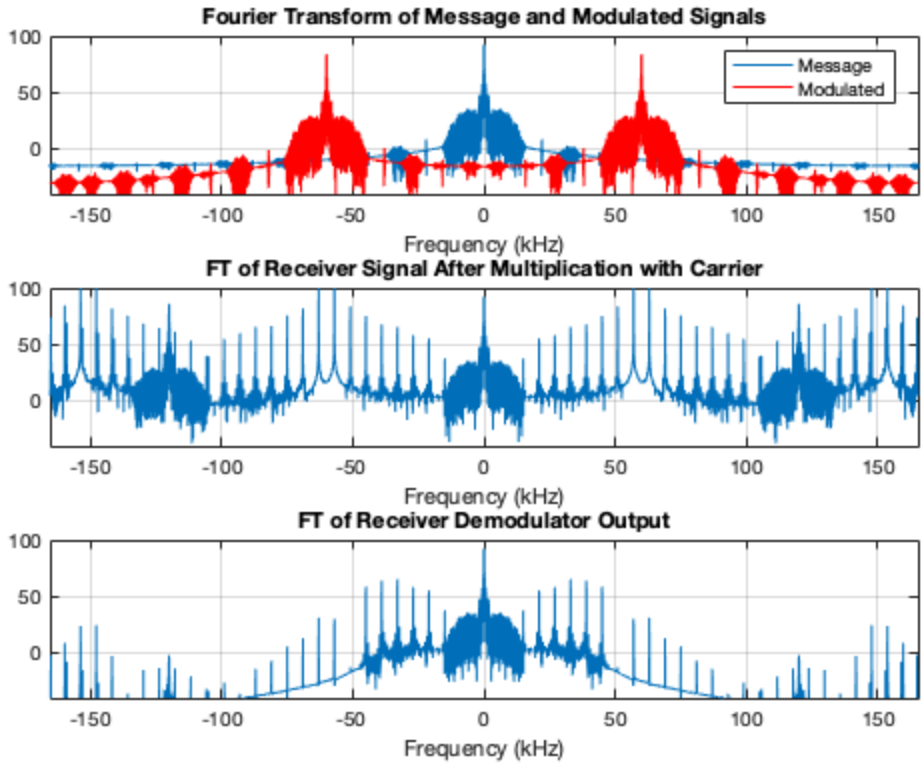
Calculate Fourier Transform of the receiver output of triangle signal

```
[FTztr,freqs]=fouriertransform(ztr,Fs);
```

Display these Fourier Transforms

```
figure(5)
subplot(3,1,1);
plot(freqs/1000, 20*log10(abs(ftsong)));
hold on
plot(freqs/1000, 20*log10(abs(ftx)),'r');
grid
legend('Message','Modulated','Location','Best')
xlabel('Frequency (kHz)');
title('Fourier Transform of Message and Modulated Signals')
axis([-Fs/2000 Fs/2000 -40 100])
subplot(3,1,2);
plot(freqs/1000, 20*log10(abs(fty)));
axis([-Fs/2000 Fs/2000 -40 100])
grid
```

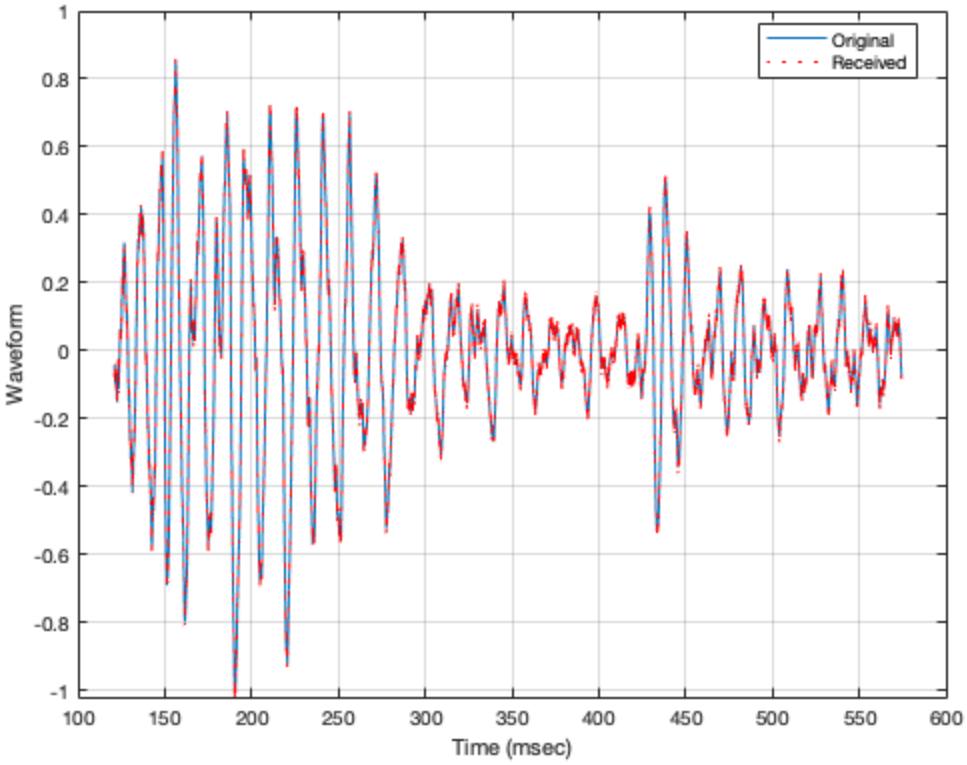
```
xlabel('Frequency (kHz)');
title('FT of Receiver Signal After Multiplication with Carrier')
subplot(3,1,3)


plot(freqs/1000, 20*log10(abs(FTz)));
axis([-Fs/2000 Fs/2000 -40 100])
grid
xlabel('Frequency (kHz)')
title('FT of Receiver Demodulator Output')
```



Display the Original Song and the Receiver Output Segments

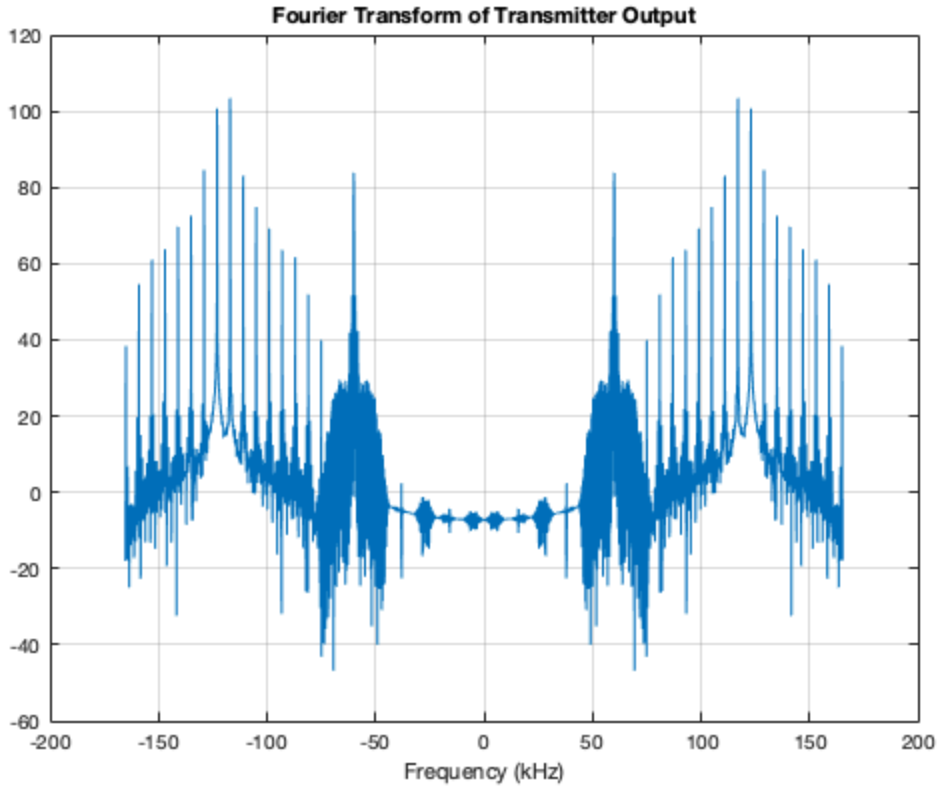
They are hardly distinguishable!

```
figure(6)
plot(t(40000:190000)*1000,song(40000:190000))
hold on
plot(t(40000:190000)*1000,z(40000:190000),'r:')
grid
xlabel('Time (msec)');
ylabel('Waveform');
legend('Original','Received','Location','Best');
```



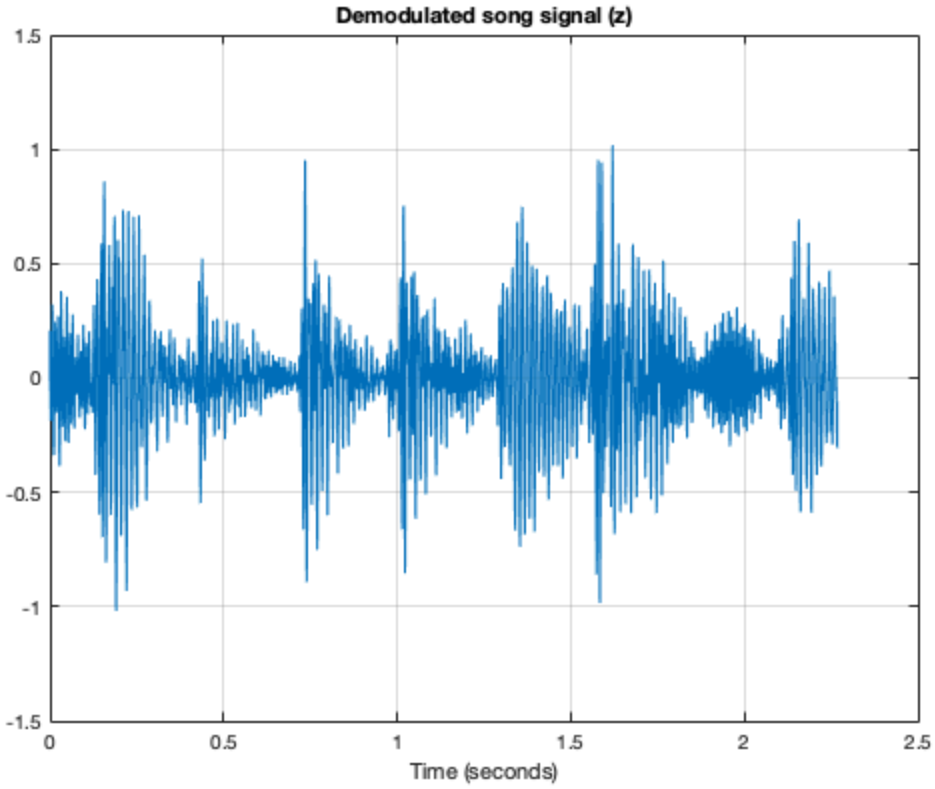
Display the Fourier Transform of the Transmitter Output (Question 2a)

```
figure(7)
plot(freqs/1000, 20*log10(abs(ftx_final)));
grid
xlabel('Frequency (kHz)');
title('Fourier Transform of Transmitter Output')
```



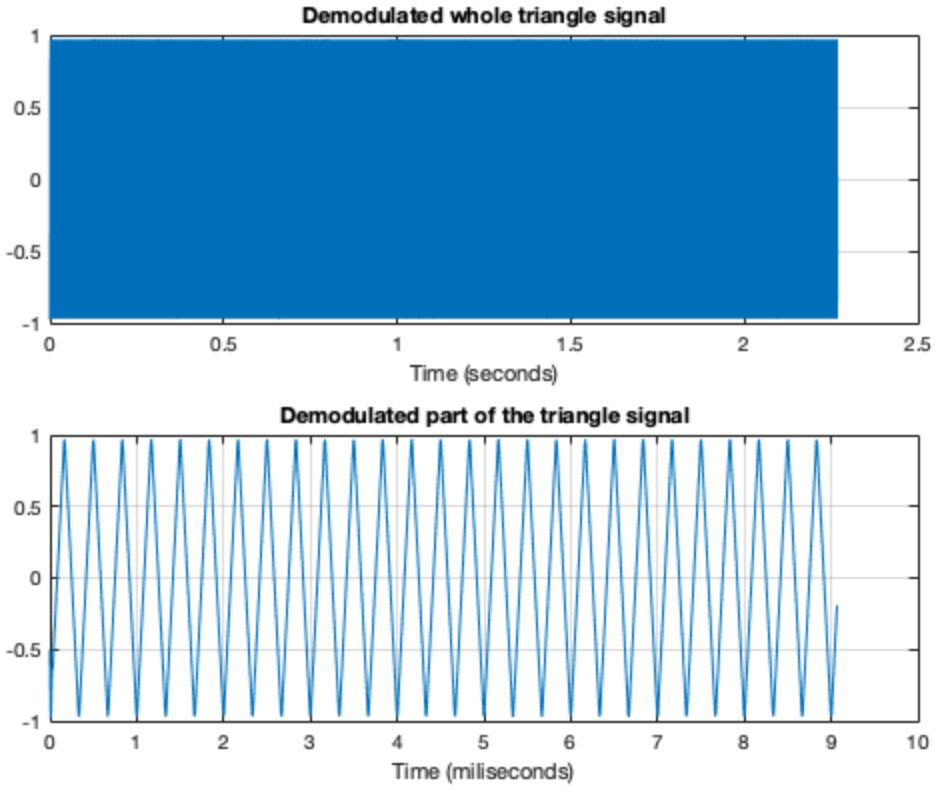
Display the time waveform of song signal (Question 2b)

```
figure(8)
plot(t, z)
grid
title('Demodulated song signal (z)')
xlabel('Time (seconds)')
```



Display the time waveform of triangle signal (Question 2b)

```
figure(9)
% Whole triangle signal
subplot(2,1,1);
plot(t, ztr)
grid
title('Demodulated whole triangle signal')
xlabel('Time (seconds)')

% A part of the triangle signal
subplot(2,1,2);
plot(t(1:3000)*1000, ztr(1:3000));
grid
title('Demodulated part of the triangle signal')
xlabel('Time (miliseconds)')
```



Play the demodulated sound

Downsampling

```
zlowrate=downsample(z,10);
% Listen to
sound(zlowrate,Fs/10);

% Although I get the signal z and it is equal to the original song
 signal,
% due to interference, I hear a beep sound
```

Published with MATLAB® R2018b