RECORD, PLAY AND SAVE THE AUDIO IN MATLAB

Table of Contents

Program Initialization

```
clc;
clear all;
close all;
```

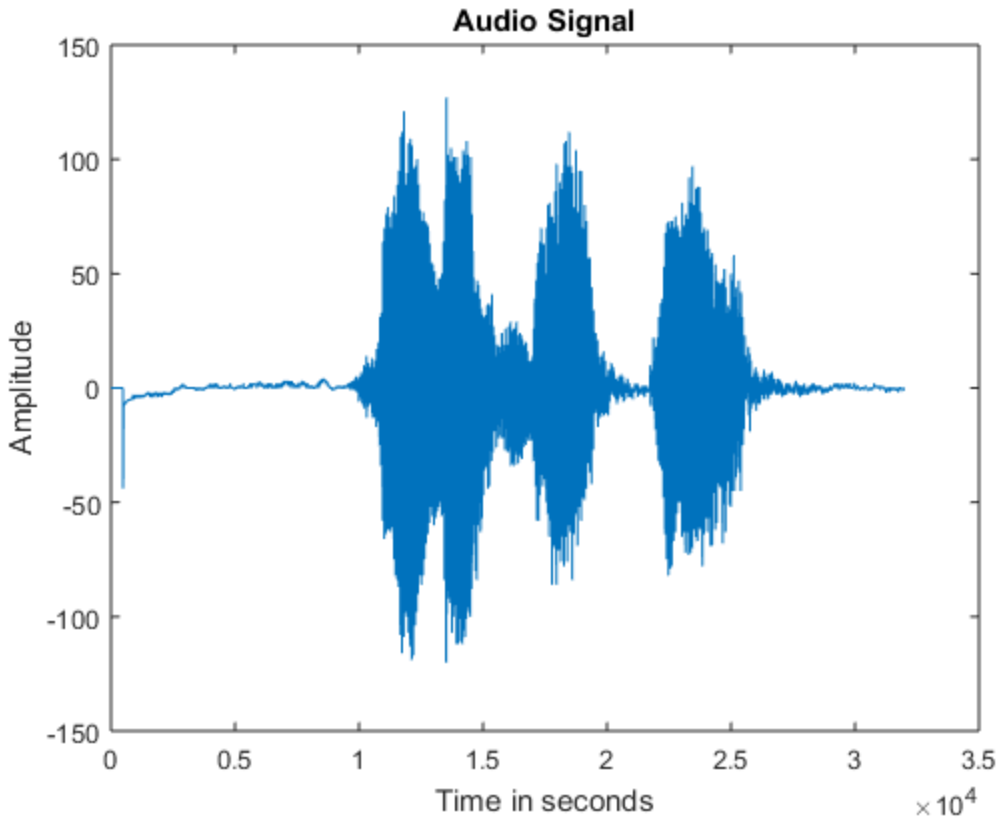
Record Audio

1. Create an **audiorecorder** object to record an audio of desired sampling rate and bits per sample
2. Record an audio of desired length (time in seconds) using **recordblocking** command.
3. Create a numeric array of the recorded audio using **getaudiodata** command
4. Convert the numeric array in terms of 2^(bits per sample-1)
5. Use **plot** to show the recorded sudio signal.

```
Fs = 16000; nBits = 8;
RecObject1 = audiorecorder(Fs, nBits, 1);
% Record an audio of 5 seconds
disp('Start Audio Record');
recordblocking(RecObject1, 2);
disp('End of Audio Record');
% Create the numeric array
AudioData = getaudiodata(RecObject1, 'double');
Data = AudioData.*(2^(nBits-1));

figure;
plot(Data); title('Audio Signal');
xlabel('Time in seconds');
ylabel('Amplitude');
```

```
Start Audio Record
End of Audio Record
```



Play the Audio

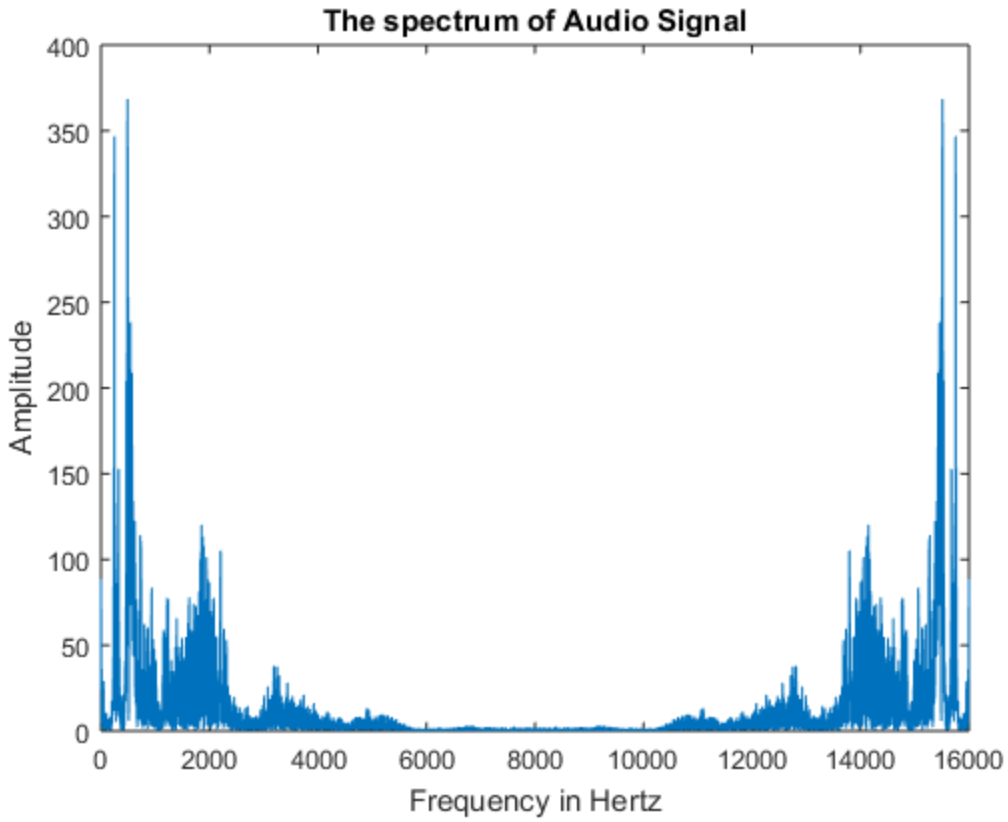
1. Play the recorded audio using **play** command.

```
play(RecObject1);
```

Plot the Spectrum

1. Generate the spectrum using **fft** and plot the same.

```
AudSpec1 = abs(fft(AudioData));
Fs1 = RecObject1.SampleRate;
N = length(AudioData);
df = Fs1/N;
K = 0 : (N-1);
freq1 = K.*df;
figure;
plot(freq1, AudSpec1); title('The spectrum of Audio Signal');
xlabel('Frequency in Hertz');
ylabel('Amplitude');
```



Save the Audio

1. Save the numeric array as audio file with extension .wav using **audiowrite** command
2. Save the numeric array as text format (ACII) with extension .dat using **dlmwrite** command
3. Save the numeric array as text format (ACII) with extension .mat using **save** command

```
audiowrite('Audio1.wav', AudioData, Fs1);
dlmwrite('Audio1.dat', Data);
save('Audio1.mat','Data','-ascii');
```

Published with MATLAB® R2015a