**EECS 360**

**2016/10/4**

**Chen Long**

**Objective**

The object of this lab is to learn how to ask MATLAB to play a specific frequency in a matrix.

**Description**

The first step that we need to do is setup sampling rate, time duration and frequency. After we have these setup, we use a sine function to implement, which we use sin(2\*pi\*f\*t). After we have the function, we use the commend “soundsc” to play the sound. When there are several frequencies that we need to play, we should use a matrix and an if loop to play the sound.

To do this, we setup the time duration first, the time duration needs to be 0.5s. We then setup the matrix, since we have 8 frequencies to play, we make an 8\*1 matrix, the commend line is “zeros(8, length(t))”. What this commend does is to create 8 rows, and the length of each row is 0.5s. On each row, we input a sine function with the frequency in, the commend line is “mat(row num, :) = sin(2\*pi\*f\*t)”. After we finish building the matrix, we use a for loop to play the sound row by row.

The next few problems uses similar method besides change the order of the rows that we wish to play and adding more 0 frequency to each row. By doing it, we change the length of each row by adding 500 unit, the commend line is “mat1=zeros(8,length(t)+500)”, what this line does is adding 500 more 0 to each row, the rest is the same as the previous.

To read a sound track file, we use the commend line “[p,fs]=audioread('Five\_Columns\_Long.wav')”, p is the graph that we want to plot, and “fs” is the sampling rate.

**Result**

The result is correct since that we can check the result by hearing the sound. On the second problem, the sound track is the song that we are really familiar with, even though I don’t know the name of the sound. The next problem we can check it by hearing the bass after each frequency since we add 500 0 frequency after each sine function. The frequency graph of the WAV file will be paste at the end of this report.

**Conclusion**

This is a really interesting lab!!! I actually play music instruments like MP3 player, earbuds a lot, as we all know, in order to have a good music quality, the sound tracking is a basic requirement. We want a sound file that is lossless. To check if the sound is being compressed, we check if the high frequency is being cut. This lab shows us how to make a song and how to check the frequency of a song, which is really awesome to my own interest.

% Task 1

clear all;

fs=6283.2;

T=1/fs;

t=0:T:0.01;

f=100;

x=sin(2\*pi\*f\*t);

soundsc(x,fs);

% subplot 2

subplot(222);

f1=1000;

x1=sin(2\*pi\*f1\*t);

soundsc(x1,fs);

% subplot 3

subplot(223);

f2=2000;

x2=sin(2\*pi\*f2\*t);

soundsc(x2,fs);

subplot(221);

plot(t,x);

subplot(222);

plot(t,x1);

subplot(223);

plot(t,x2);

% Task 2

% part a

fs=44100;

t=0:1/fs:0.5;

mat=zeros(8,length(t));

f1=261.626;

f2=293.665;

f3=329.628;

f4=349.228;

f5=391.995;

f6=440;

f7=493.883;

f8=0;

mat(1,:)=sin(2\*pi\*f1\*t);

mat(2,:)=sin(2\*pi\*f2\*t);

mat(3,:)=sin(2\*pi\*f3\*t);

mat(4,:)=sin(2\*pi\*f4\*t);

mat(5,:)=sin(2\*pi\*f5\*t);

mat(6,:)=sin(2\*pi\*f6\*t);

mat(7,:)=sin(2\*pi\*f7\*t);

mat(8,:)=sin(2\*pi\*f8\*t);

for n=1:7;

soundsc(mat(n,:),fs);

pause(0.5);

end;

list=[1,1,5,5,6,6,5,8,4,4,3,3,2,2,1];

for n=1:length(list);

soundsc(mat(list(n),:),fs);

pause(0.5);

end;

% part b

mat1=zeros(8,length(t)+500);

mat1(1,:)=[sin(2\*pi\*f1\*t), zeros(1, 500)];

mat1(2,:)=[sin(2\*pi\*f2\*t), zeros(1, 500)];

mat1(3,:)=[sin(2\*pi\*f3\*t), zeros(1, 500)];

mat1(4,:)=[sin(2\*pi\*f4\*t), zeros(1, 500)];

mat1(5,:)=[sin(2\*pi\*f5\*t), zeros(1, 500)];

mat1(6,:)=[sin(2\*pi\*f6\*t), zeros(1, 500)];

mat1(7,:)=[sin(2\*pi\*f7\*t), zeros(1, 500)];

mat1(8,:)=[sin(2\*pi\*f8\*t), zeros(1, 500)];

for n=1:length(list);

soundsc(mat1(list(n),:),fs);

pause(0.5);

end;

% Task 3

figure (2);

[p,fs]=audioread('Five\_Columns\_Long.wav');

subplot(211);

plot(p);

title('Five Columns\_Long');

[p,fs]=audioread('Going\_Home.wav');

subplot(212);

plot(p);

title('Going Home');



