EHB328E – Machine Learning for Signal Processing Matlab HW1

Project Members

Veli Bulur – 040150051

Mehmet Şerbetçioğlu – 040160056

MLhw1_1.m | mlhw1_2.m | ehb328Hw2.m

```
clear, clc, close all;

% Firstly read the audio.
materialfolder = 'hw1materials';
soundname = dir([materialfolder filesep 'polyushka.wav']);
[poly, Fs] = audioread([materialfolder filesep soundname.name]);

% Figure the original spectrum of audio.
figure
spectrogram(poly,'yaxis')
title('Original Sound Spectrum')

notesfolder = 'notes15';
portion = 1:Fs*5;

poly_spectrum = stft(poly', 2048, 256, 0, hann(2048));
poly_stft = abs(poly_spectrum);
poly_phase = poly_spectrum./(poly_stft+eps);

% Adding the 15 notes in to 'notes' array.
listname = dir([materialfolder filesep notesfolder filesep '*.wav']);
notes = [];
for i=1:length(listname)
  [s, Fs] = audioread([materialfolder filesep notesfolder filesep listname(i).name]);
  s = s(:,1);
  s = resample(s, 16000, Fs);
  spectrum = stft(s', 2048, 256, 0, hann(2048));
  middle = ceil(size(spectrum, 2) /2);
  note = abs(spectrum(:, middle));
  note(find(note<max(note(:))/100)) = 0 ;
  note = note/norm(note);
  notes = [notes, note];
end
```

```
34
35      % To find the W, pseudoinverse of notes * stft of the audio poly.
36 -    W = pinv(notes)*poly_stft;
37
38      % W's negative values go to zero.
39 -    for i=1:numel(W)
40 -        if W(i)<0
41 -            W(i) = 0;
42 -        end
43 -    end
44
45 -    polysynth_stft = notes*W;
46
47      % Figure of synthesised power spectrum of audio's stft.
48 -    figure
49 -    imagesc(polysynth_stft)
50 -    title('Synthesised Power Spectrum2')
51
52      % For figure of synthesised sound spectrum, invert the stft of audio.
53 -    polysynth_spectrum = polysynth_stft.*poly_phase;
54 -    polysynth = stft(polysynth_spectrum, 2048, 256, 0, hann(2048));
55 -    polysynth = polysynth';
56
57      % Figure the synthesised sound spectrum.
58 -    figure
59 -    spectrogram(polysynth,'yaxis')
60 -    title('Synthesised Sound Spectrum')
61
62      % Obtain the synthesised audio in format of wav.
63 -    audiowrite('poly_synth.wav', polysynth, Fs);
64
65      function [f,fp] = stft( x, sz, hp, pd, w) ...
```

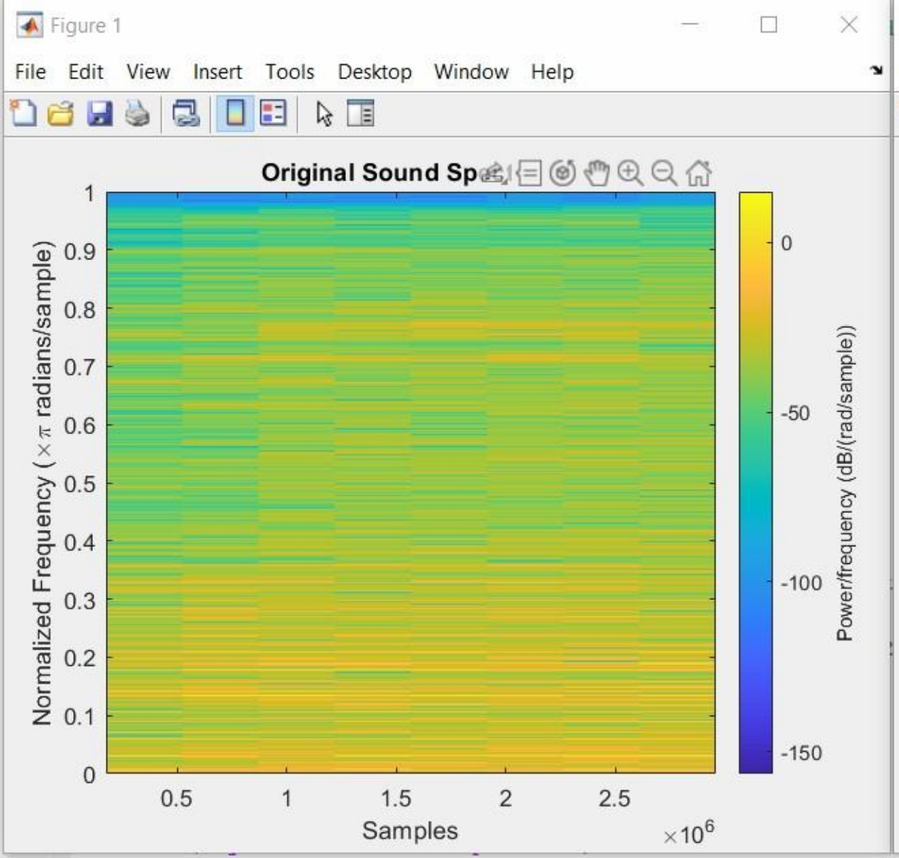
Figure 1
File Edit View Insert Tools Desktop Window Help
Original Sound Sp
Normalized Frequency (×π radians/sample)
Samples
×10^6
Power/frequency (dB/(rad/sample))

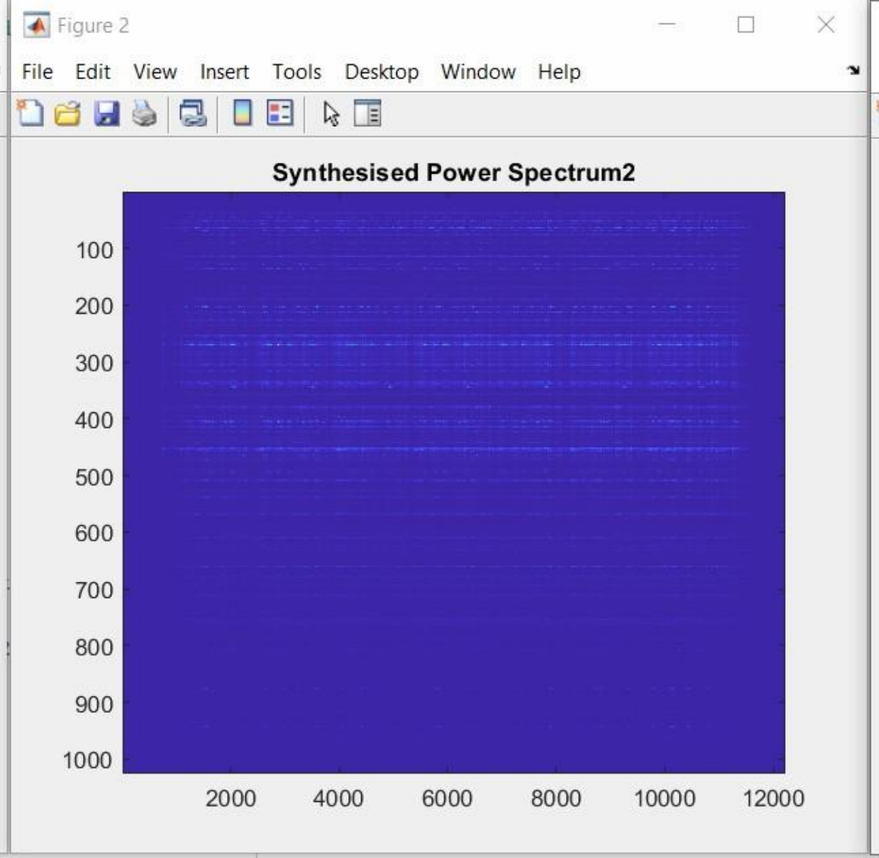
Figure 2
File Edit View Insert Tools Desktop Window Help
Synthesised Power Spectrum2

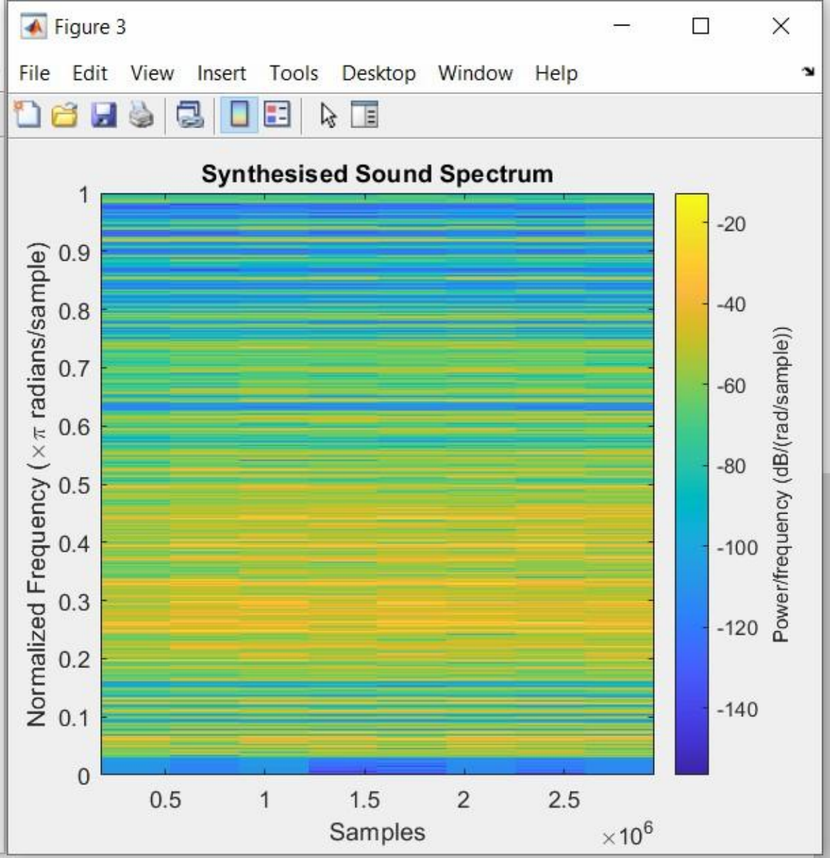
Figure 3
File Edit View Insert Tools Desktop Window Help
Synthesised Sound Spectrum
Normalized Frequency (×π radians/sample)
Samples
×10^6
Power/frequency (dB/(rad/sample))

MLhw1_1.m | mlhw1_2.m | ehb328Hw2.m | +

```
clear all; clc; close all;

% Firstly read the 3 audios.
[audioA, fsA] = audioread('silentnight_piano.aif');
[audioB, fsB] = audioread('silentnight_guitar.aif');
[audioC, fsC] = audioread('littlestar_piano.aif');

% Given audios have two channels, we need first channel.
% Then find audios' stft.
audioAL = audioA(:,1);
spectrumA = stft(audioAL', 1024, 256, 0, hann(1024));
music_stftA = abs(spectrumA);

audioBL = audioB(:,1);
spectrumB = stft(audioBL', 1024, 256, 0, hann(1024));
music_stftB = abs(spectrumB);

audioCL = audioC(:,1);
spectrumC = stft(audioCL', 1024, 256, 0, hann(1024));
music_stftC = abs(spectrumC);
sphaseC = spectrumC ./(abs(spectrumC)+eps);

% Before transformation the audio's negative values go to zero.
for i=1:numel(music_stftC)
    if music_stftC(i) < 0
       music_stftC = 0;
    end
end

% To find the audio D's stft, get these following steps.
X =  music_stftB * pinv(music_stftA);
stftD = X * music_stftC;
```

```
33      % Finally invert the audio D's stft.
34 -    audioDL = stft(stftD.*sphaseC, 1024, 256, 0, hann(1024));
35 -    audioDL = audioDL';
36 -    soundsc(audioDL(portion), fsC);
37
38      % Obtain audio D which means 'littlestar_guitar' from the others in format wav.
39 -    audiowrite('audioD_syns.wav', audioDL, fsC);
40
41      % Matlab file in the attached zip combines the two codes. Please inspect that one as well.
42      % These codes can be found at https://github.com/velibulur/ehb328hw2 and https://github.com/meserbetcioglu/ehb328hw2
43      % Written by Veli Bulur and Mehmet Şerbetçioğlu
```