



**YILDIZ TEKNİK ÜNİVERSİTESİ  
ELEKTRİK-ELEKTRONİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BLM 3620 SAYISAL İŞARET İŞLEME**

## **ÖDEV 3**

Muhammet Ali Şen - 20011701

## Spectrogram Nedir?

Spektrogram, zamana göre değiştiği için bir sinyalin frekans spektrumunun görsel bir gösterimidir. Bir ses sinyaline uygulandığında, spektrogram'lara bazen sonograflar, ses baskıları veya ses birimleri denir. Short Time Fourier Transform (STFT) olarak da adlandırılabilir.

### CEVAP-1 KODLAR

```
%% data
[data,Fs] = audioread('Audio1.wav');

channel = 1;
signal = data(:,channel);
samples = length(signal);

spectrogram_dB = 100;
%% decimate
% decim = 1 ise bisey olmaz
decim = 2;
if decim>1
    signal = decimate(signal,decim);
    Fs = Fs/decim;
end
nfft=1024;
window = hamming(nfft);
window = window(:);
win = window;

[sg,fsg,tsg] = myspecgram(signal,win,Fs);
%[sg,tsg,fsg] = myspecgram(signal,win,Fs);

% Lineer den FFT cevrimi
sg_dBpeak = 20*log10(abs(sg))+20*log10(2/length(fsg));
%sg_dBpeak=fft(sg.*hamming(nfft))*4/nfft; % hanning only

% saturation of the dB range :
% saturation_dB = 60; % dB range scale (means , the lowest displayed level
is XX dB below the max level)
min_disp_dB = round(max(max(sg_dBpeak))) - spectrogram_dB;
sg_dBpeak(sg_dBpeak<min_disp_dB) = min_disp_dB;

% plots spectrogram
figure(2);
imagesc(tsg,fsg,sg_dBpeak);colormap('gray');
%imagesc(fsg,tsg,sg_dBpeak);colormap('hot');
axis('xy');colorbar('vert');grid
title([' Spectrogram / Fs = ' num2str(Fs/2) ' Hz / Delta f = '
num2str(fsg(2)-fsg(1)) ' Hz ']);
xlabel('Time (s)');ylabel('Frequency (Hz)');
%xlabel('Frequency (Hz)');ylabel('Time (s)');
```

```

function [sg,freq_vector,time] = myspecgram(signal,win, Fs)
nfft = length(win); %
Overlap = 0.5;
dt = 1/Fs;
signal = signal(:);
samples = length(signal);

% uzunluk nfft den kisa ise zero ile doldur
if samples<nfft
    s_tmp = zeros(nfft,1);
    s_tmp((1:samples)) = signal;
    signal = s_tmp;
    samples = nfft;
end

offset = fix(Overlap*nfft);
noverlap = Overlap*nfft;
spectrum = fix((samples-noverlap)/(nfft-noverlap));

sg = [];
for i=1:spectrum
    start = (i-1)*offset;
    sw = signal((1+start):(start+nfft)).*win;
    sg = [sg (abs(fft(sw))*4/nfft)];
    %sg=fft(sw.*hamming(nfft))*4/nfft; % hamming only
    time(i) = (start+nfft/2)*dt;
end

if rem(nfft,2) % nfft tek
    select = (1:(nfft+1)/2)';
else
    select = (1:nfft/2+1)';
end

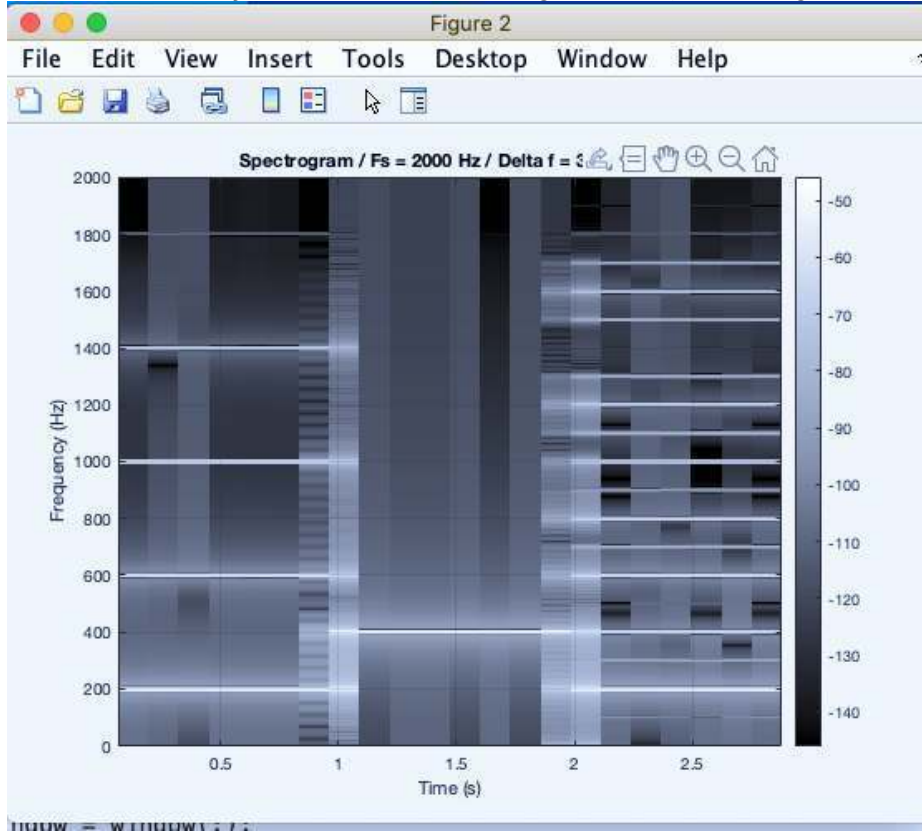
sg = sg(select,:);
freq_vector = (select-1)*Fs/nfft;

end

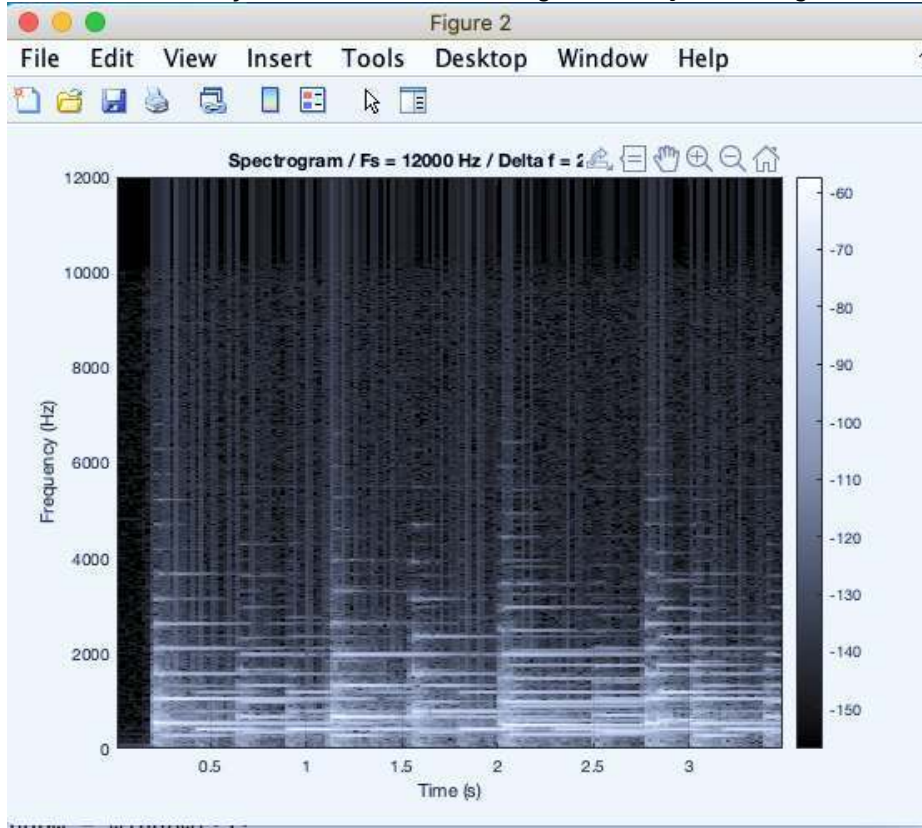
```

CEVAP-1

AUDI02.WAW dosyası  $nfft=1024$  ile görselleştirildiğinde



AUDI01.WAW dosyası  $nfft=1024$  ile görselleştirildiğinde

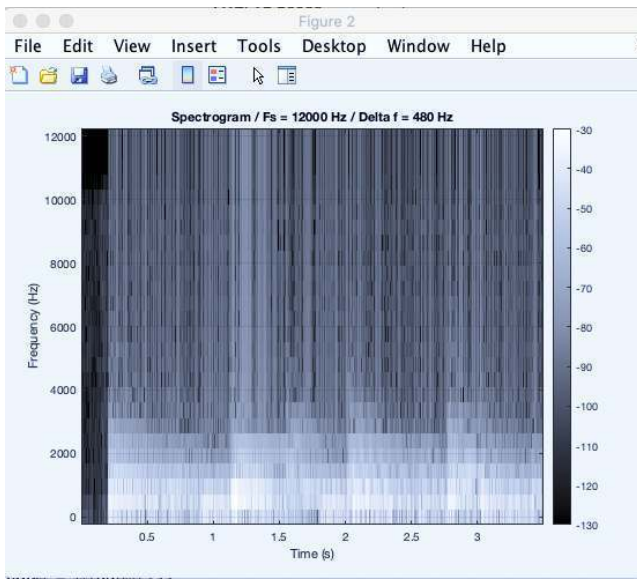


### CEVAP-3

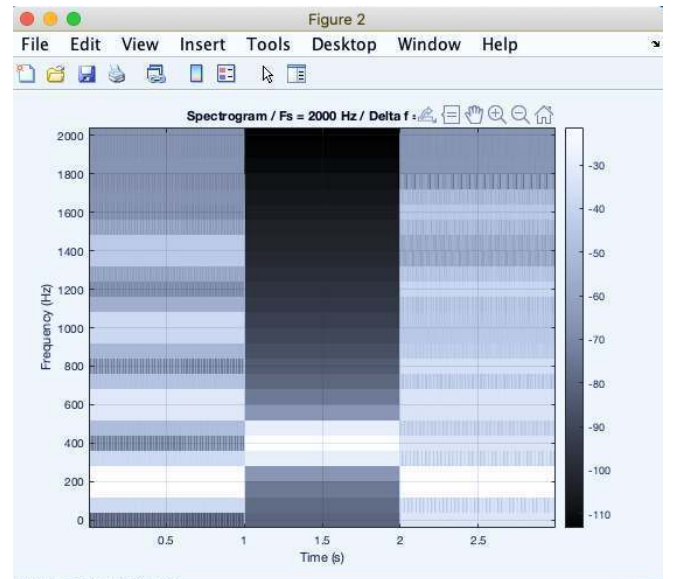
Audio1 dosyası da belirli zaman dilimlerinde daha az frekans da sesler olduğu için daha net görüntüler görülürken, audio2 dosyası birçok farklı frekanstaki sesin aynı zaman diliminde bulunması nedeniyle daha karışık görülmektedir.

Pencereleme (windowing) işlemi periyodik olmayan sinyallerin fourier dönüşümlerinin daha doğru çıkması için yapılmaktadır. Genellikle ses sinyallerinde kullanılır. Örnek olarak 512 lik bir pencere için 256 ve katları olacak şekilde bir pencereleme yapılırsa periyodik olmayan sinyaller için kesişim pencerelerinin sayısı daha fazla olacağı için periyodik olmayan sinyallerde daha doğru sonuç doğuracaktır.

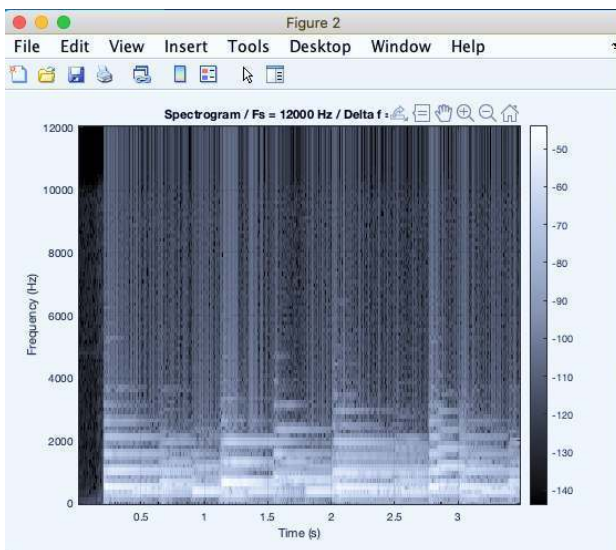
### CEVAP-4



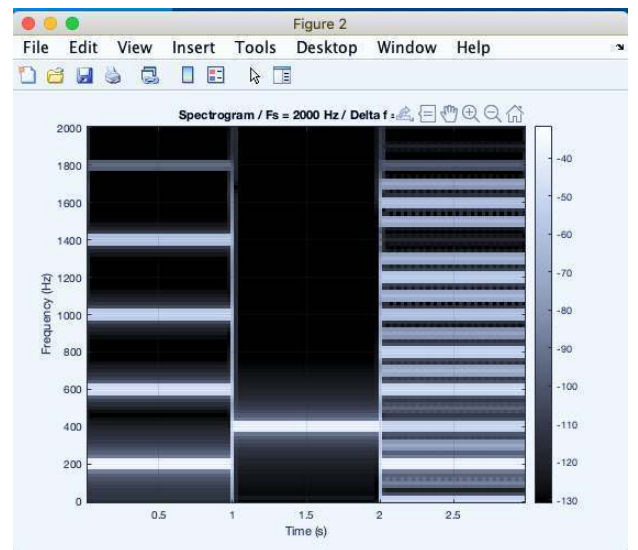
Audio1 nfft=50



Audio2 nfft=50



Audio1 nfft=200



Audio1 nfft=200