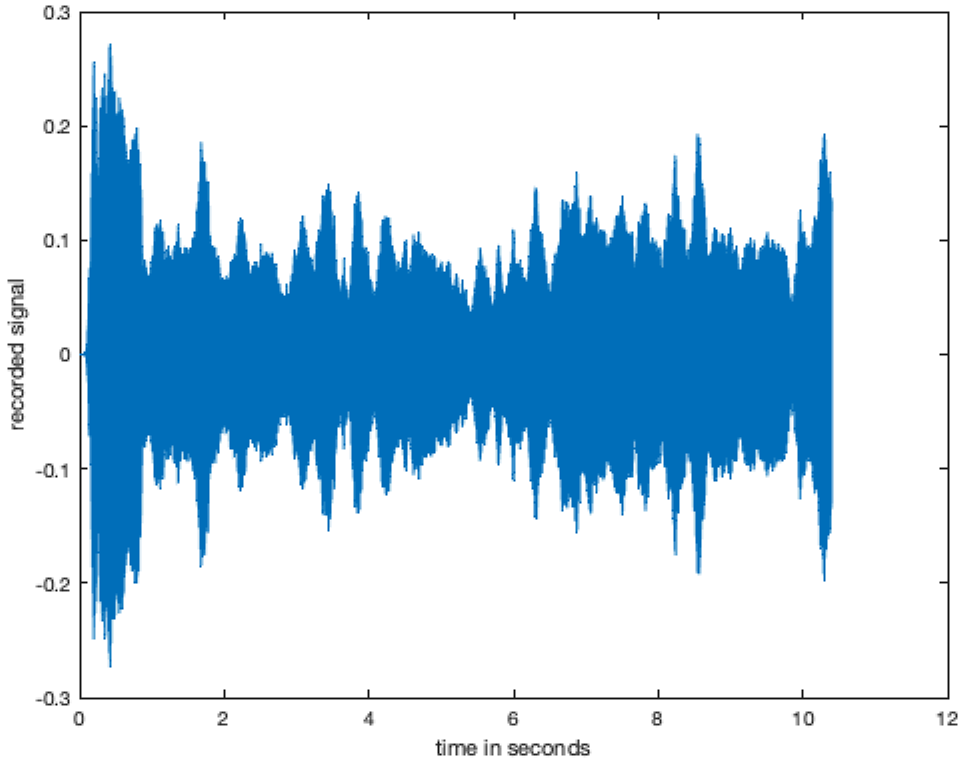
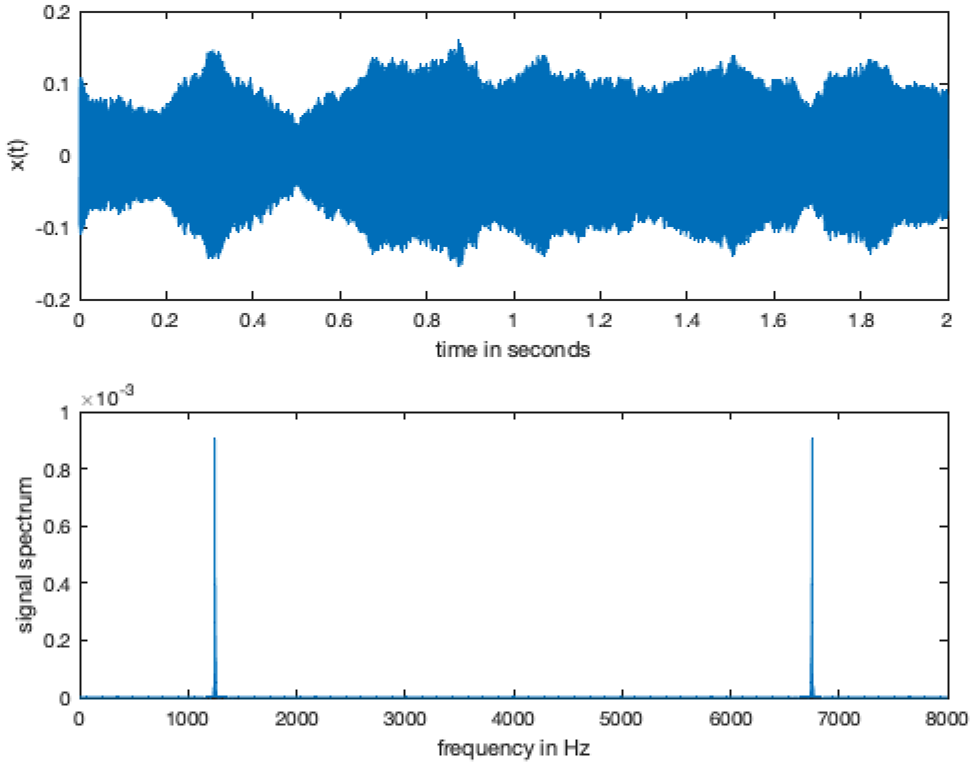
```
% TSRT78, Lab 1: Fundamental Signal Processing
% Matheus Bernat (matvi959) & Caspian Süsskind (cassu286)

% ===================== 4 Assignment: Whistle
 ============================


% Read wav file: extract data and sampling frequency
[y, fSamp] = audioread('whistle.wav');

% Check that 8000Hz
fSamp;
nSamp = size(y,1);


% Hear sound:
sound(y,fSamp);

% ------------------ QUESTION 1 ------------------
% A lot of stuff

% --------------- Plot signal in time axis
ts = 1/fSamp;
timeVector = ts*(0:nSamp-1); % time vector in seconds

figure;
plot(timeVector, y)
xlabel('time in seconds');
ylabel('recorded signal');

% --------------- Calculate energy of signal in time domain

% Get signal from 6 to 8 seconds
idx = (timeVector >= 6) & (timeVector <= 8);
y = y(idx);

nSamp = size(y, 1);
timeVector = ts*(0:nSamp-1);
figure; subplot(2,1,1);
plot(timeVector, y)
xlabel('time in seconds');
ylabel('x(t)');

totalEnergy_t = 0;
for i = 1:length(y)
    totalEnergy_t = totalEnergy_t + abs(y(i))^2;
end

% --------------- Plot spectrum
Yf = fft(y);
frequencyVector = ((0:nSamp-1)/nSamp)*fSamp;
subplot(2,1,2);
```

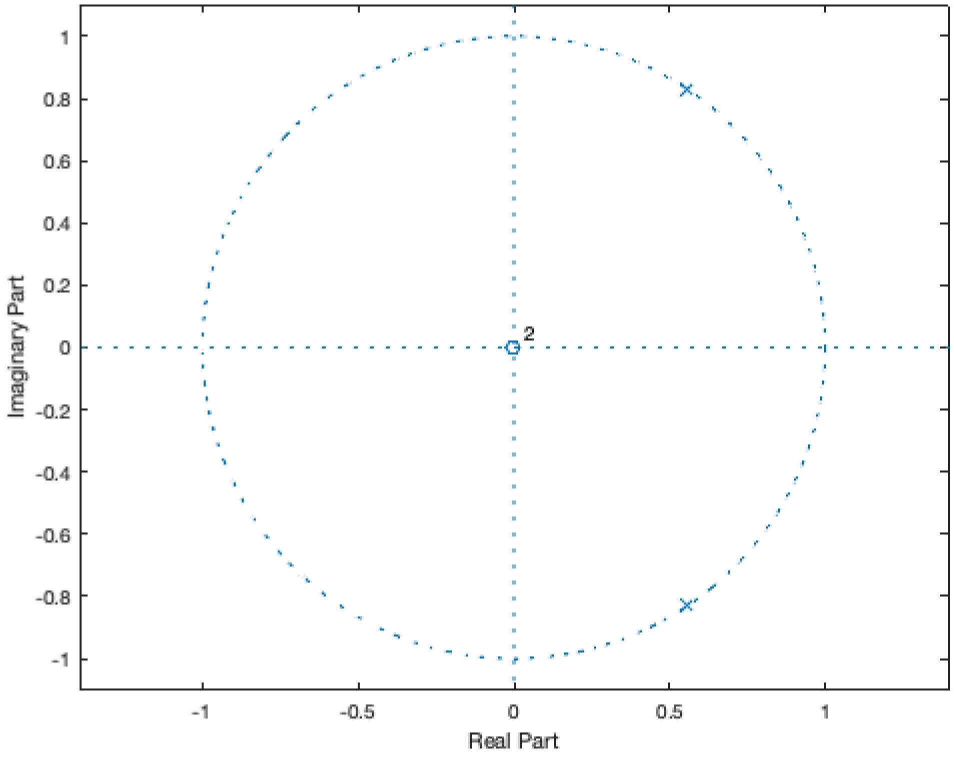
```
plot(frequencyVector , (abs(Yf).^2)*ts/nSamp)
xlabel('frequency in Hz');
ylabel('signal spectrum');

% By looking at spectrum, decide dominant frequency 1250
dominantFreq = 1250;
sth = 10;

yFiltered = bandpass(y, [dominantFreq - sth, dominantFreq + sth],
 fSamp);

% --------------- Calc energy of dominant frequency signal in time
 domain
dominantFreqEnergy_t = 0;
for i = 1:length(yFiltered)
    dominantFreqEnergy_t = dominantFreqEnergy_t+ abs(yFiltered(i))^2;
end

% ANSWERS:
totalEnergy_t;
dominantFreqEnergy_t;
% ------------------ QUESTION 2 ------------------
% Same calculations as in question 1, but in the frequency domain.

% Calculate energy in the frequency domain
totalEnergy_f = 0;
for i = 1:length(Yf)
    totalEnergy_f = totalEnergy_f + abs(Yf(i)^2)*ts/nSamp;
end

% Calculate energy of dominant frequency in frequency domain
idx = (frequencyVector >= dominantFreq - sth) & (frequencyVector <=
 dominantFreq +sth);
dominantFreqSignal = Yf(idx);

dominantFreqEnergy_f = 0;
for i = 1:length(dominantFreqSignal)
    dominantFreqEnergy_f = dominantFreqEnergy_f +
 2*abs(dominantFreqSignal(i))^2/nSamp;
end

% ANSWERS:
totalEnergy_f;
dominantFreqEnergy_f;

% ------------------ QUESTION 3 ------------------
% Calc harm. distortion using energy calculations from time and freq
 domain

% ANSWERS:
hdist_t = 1 - dominantFreqEnergy_t/totalEnergy_t; % 0.0030
hdist_f = 1 - dominantFreqEnergy_f/totalEnergy_f; % 0.0680
```

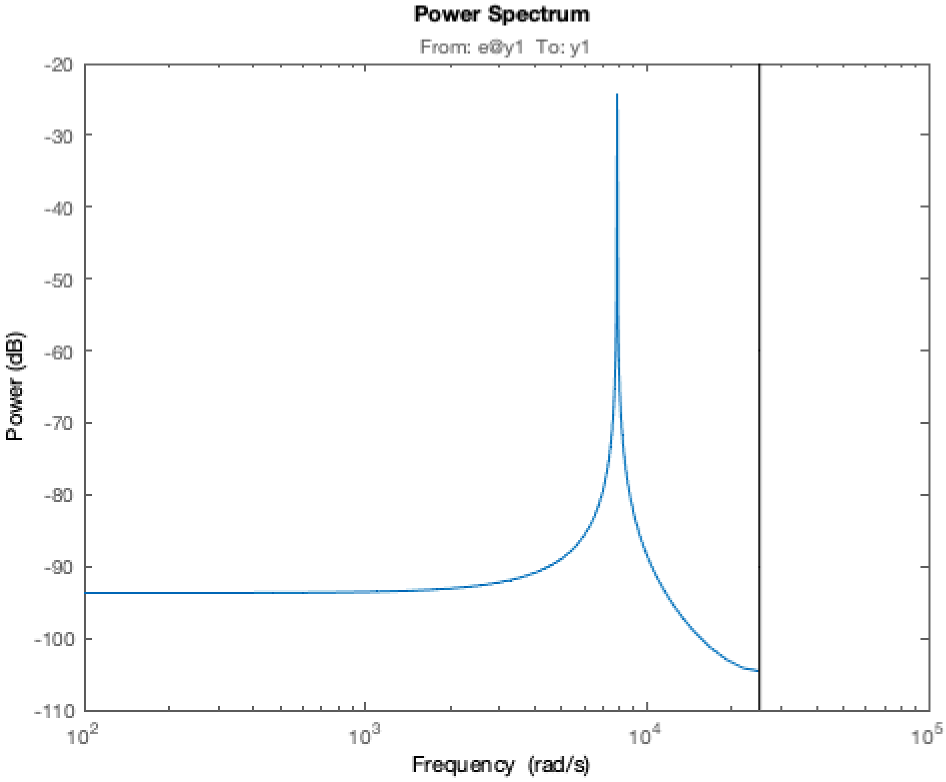
```
% ------------------ QUESTION 4 ------------------
% Estimate the purity measure based on an AR(2) and motivate why this
 model
% is suitable. How can this measure be compared to the harmonic
 distortion?

modelOrder = 2;
[th,P,lam,epsi] = sig2ar(y,modelOrder);
a1 = th(1,1); a2 = th(2,1);

figure;
zplane([1],[1 a1 a2]);

pole1 = -a1/2 + sqrt(((a1^2)/4)-a2);
pole2 = -a1/2 - sqrt(((a1^2)/4)-a2);

% ANSWERS:
distance = 1 - abs(pole1);

% ------------------ QUESTION 5 ------------------
arMod = ar(y, 2, 'Ts', ts);
figure, bode(arMod)

% Plot for non parametric method in QUESTION 1
```

```
Warning: A bode plot is not well defined for a time series model. The
 plot will
show the output spectrum of the model. Consider using the "idlti/
spectrum"
command instead.
```


0.3
0.2
0.1
0
-0.1
-0.2
-0.3
recorded signal
0
2
4
6
8
10
12
time in seconds


0.2
0.1
0
-0.1
-0.2
x(t)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
time in seconds
×10^-3
1
0.8
0.6
0.4
0.2
0
signal spectrum
0
1000
2000
3000
4000
5000
6000
7000
8000
frequency in Hz


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Imaginary Part
2
-1
-0.5
0
0.5
1
Real Part


Power Spectrum
From: e@y1 To: y1
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
Power (dB)
10^2
10^3
10^4
10^5
Frequency (rad/s)

Published with MATLAB® R2020a