Documentation on

Project based Learning

Topic: Correlation Concepts

Prepared by

K. R. Deepthi, Assistant Professor, ECE

Department of Electronics and Communication Engineering

Correlation

Correlation is a statistic that establishes the relationship between two variables. In other words, it is the measure of association of variables.

Basis	Coííelation
Meaning	A statistical measuíe that defines co-íelationship oí association of two vaíiables.
Dependent and Independent vaíiables	No diffeíence
Usage	I˝o descíibe a lineaí íelationship between two vaíiables.
Objective	I˝o find a value expíessing the íelationship between vaíiables.

Similarities can be extracted between two different entities or between an entity and its delayed version. Based on this correlation is classifies into two types. They are Auto correlation and Cross Correlation.

Clear Goals of this work:

1. **To give depth in knowledge on Correaltion**
2. **To give hands on exposure on Matlab**
3. **To Extract similarities between two signals and calculate power spectral density.**

ABSTRACT:

Our project mainly focuses on comparing two audio signals. Here, we first lively record two audio signals with the users' knowledge. After that, we will start comparing the two signals based on few parameters like power level and the amount of similarity between the two.

Implementation Platform

Our project is purely based on software. All the buildings blocks of code, instructions used in it and the comparison console can be obtained through specific application-oriented software i.e. **MATLAB.** It is provided by **Mathworks** vendor. It is of version R2018b, works on 64-bit processor and compatible with all the three operating systems(windows, Linux, Mac-OS).

Toolbox Reference

1. xcorr:
   r = xcorr(x,y) returns the cross-correlation of two discrete-time sequences, x and y. Cross - correlation measures the similarity between x and shifted (lagged) copies of y as a function of the lag. If x and y have different lengths, the function appends zeros at the end of the shorter vector so it has the same length, *N*, as the other.

2. audiorecorder:
   recorder = audiorecorder(Fs,nBits,nChannels) sets the sample rate Fs (in Hz), the sample size nBits, and the number of channels nChannels.

3. getaudiodata:
   y = getaudiodata(*recorder*) returns recorded audio data associated with audiorecorder object *recorder* to double array *y*.

4. fft:
   y= fft(x) computes the discrete Fourier transform (DFT) of x using a fast Fourier transform (FFT) algorithm.

5. linspace:
   y= linspace(x1,x2,n) generates n points. The spacing between the points is (x2-x1)/(n-1).

6. nextpow2:
   y= nextpow2(A) returns the next higher power of 2 for each element in A.

7. corrcoef:
   y= corrcoef(A,B) returns the correlation coefficients between two random variables A and B.

HISTORY BEHIND PROJECT:

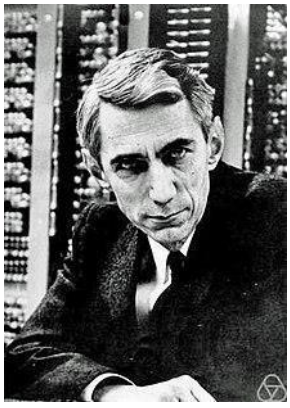
Audio processing systems have been a part of many people's lives since the invention of phonograph in the 1870s. The resulting string of innovations well-supported by the astounding technology have become a part of today's portable audio devices such as the ubiquitous MP3. The real motivation for audio signal processing began in the beginning in the 20th century with inventions like the telephone and radio that allowed for the transmission and storage of audio signals. Audio signal processing was necessary for early radio broadcasting as there were many problems with studio-to-transmitter links.

The theory of signal processing and its application to audio was largely developed at Bell Labs in the mid 20th century. Claude Shannon and Harry Nyquist's early work on communication theory, sampling theory, and Pulse-code modulation laid the foundations for the field.

The sampling theorem was implied by the work of Harry Nyquist in 1928 in which he showed that up to 2*B* independent pulse samples could be sent through a system of bandwidth *B*; but he did not explicitly consider the problem of sampling and reconstruction of continuous signals. The sampling theorem, essentially a dual of Nyquist's result, was proved by Claude E. Shannon. In 1948 and 1949, Claude E. Shannon published - 16 years after Vladimir Kotelnikov - the two revolutionary articles in which he founded the information theory. In Shannon 1948 the sampling theorem is formulated as "Theorem 13": Let $f(t)$ contain no frequencies over W. Then -

$$f(t) = \sum_{n=-\infty}^{\infty} Xn \frac{\sin \pi (2wt - n)}{\pi (2wt - n)} \quad \text{where } Xn = f(\frac{n}{2w})$$

In 1957, Max Mathews became the first person to synthesize audio from a computer, giving birth to computer music. On the other hand, the development of fast algorithms for DFT (Discrete Fourier Transform) can be traced to Carl Friedrich Gauss's unpublished work in 1805. James Cooley and John Tukey published a general version of FFT (Fast Fourier Transform) in 1965 that is applicable when *N* is composite and not necessarily a power of 2.



Claude Shannon

Harry Nyquist

John Tukey

INTRODUCTION:

In signal processing, **cross-correlation** is a measure of similarity of two series as a function of the displacement of one relative to the other. This is also known as a *sliding dot product* or *sliding inner-product*. It is commonly used for searching a long signal for a shorter, known feature. The cross-correlation is similar in nature to the convolution of two functions. In an auto-correlation, which is the cross-correlation of a signal with itself, there will always be a peak at a lag of zero, and its size will be the signal energy.

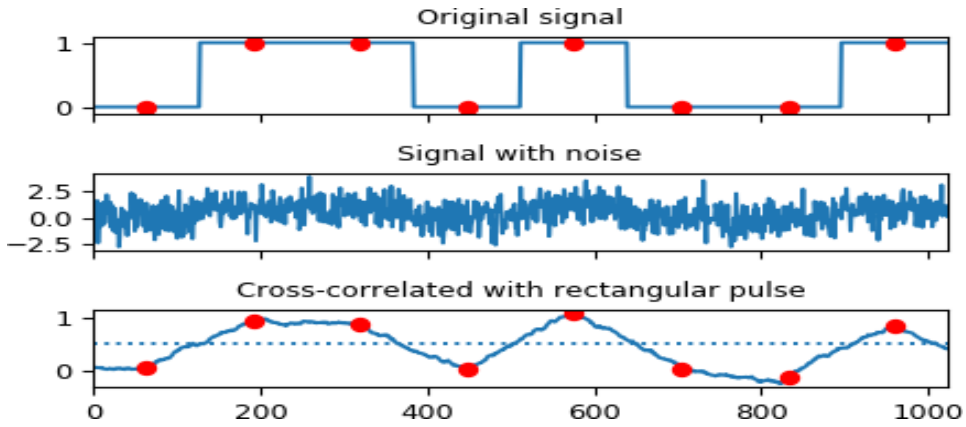


Fig. Cross-correlation performed between a rectangular signal and a noise signal

Two audio signals can be checked for the amount of similarity that exists between them by computing the coefficient of cross-correlation between the two. The audio signals can be declared as fairly similar if the magnitude of coefficient of correlation is near to 1. Other parameter which can be helpful in comparing two audio signals is the power spectral density. For continuous signals over all time, such as stationary processes, one must rather define the *power spectral density* (PSD); this describes how power of a signal or time series is distributed over frequency. Here, power can be the actual physical power, or more often, for convenience with abstract signals, is simply identified with the squared value of the signal. For example, statisticians study the variance of a function over time (or over another independent variable), and using an analogy with electrical signals (among other physical processes), it is customary to refer to it as the *power spectrum* even when there is no physical power involved.

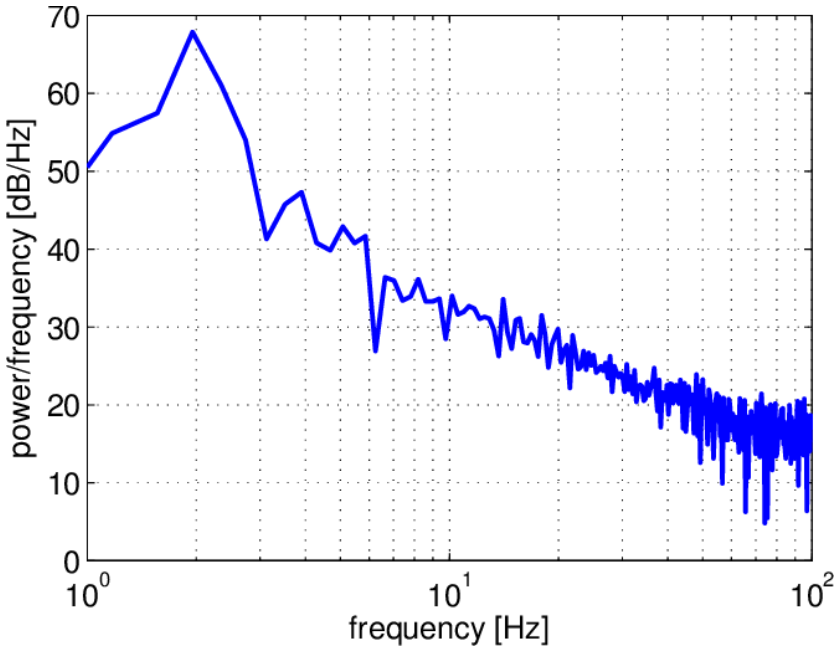


Fig. Power spectral density of a 2 Hz sinusoidal speed signal.

Hence, power spectral density gives an estimate of the power of the audio under consideration at each frequency. This observation provides a good basis for comparison of the audio signals.

METHODOLOGY:

In this project, cross-correlation is performed on two lively recorded audio signals. The description of the code algorithm is as follows:

Code algorithm description:

Firstly, we used the audiorecorder() function from signal processing toolbox to record audio from the user. The audio can be voice, music etc. The audio is sampled with a sampling rate 44.1KHz(the standard sampling rate for an audio signal) and the audio is received through one channel. Each sample is of 16 bit length. Secondly, the audio data is extracted from the audio recorder object created above. Now we have the entire time-domain sequence of the recorded audio stored in a column vector.

The next step is to convert the extracted audio data into a frequency-domain variant. A discrete-time sequence is transformed into frequency domain using the discrete Fourier transform. This is where the FFT (Fast Fourier Transform) algorithm pitches in. Naive approach to calculate DFT aka Discrete Fourier Transform has a time complexity of O(N^2)

which is very large if the number of samples in the data are considerably high just as in an audio signal. Fortunately, the FFT (usually Cooley-Tukey radix-2 decimation in time FFT) does the job with a time complexity of O(NlogN) which is a way smaller than that of naive approach at the scale of the sampling rate of 44.1KHz. Luckily, MATLAB has a built-in function for radix-2 FFT as a part of its huge, dynamic signal processing toolbox.

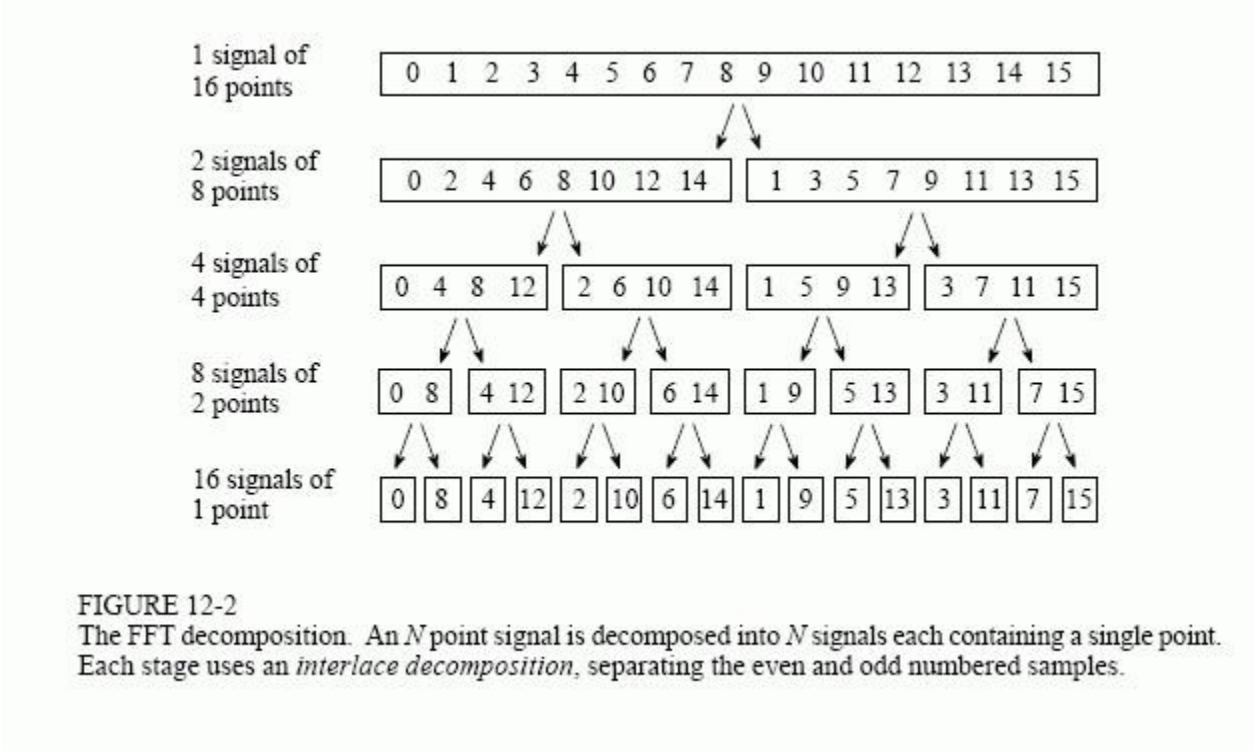


FIGURE 12-2
The FFT decomposition. An *N* point signal is decomposed into *N* signals each containing a single point. Each stage uses an *interlace decomposition*, separating the even and odd numbered samples.

Image Credits: DSP guide book by Steven W. Smith-MIT

It is so that the radix-2 FFT works best if the number of samples in the data are a power of two. Hence the length of the time-domain sequence is converted into a power of 2 just greater than it. Hence to plot the frequency domain version of the sequence, the x-axis i.e. the frequency must be divided into 'n' evenly spaced points where n is the converted length of sequence divided by two (since the FFT gives a symmetrical spectrum). The formula for calculating FFT theoretically is as shown below:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

where k=*0,1,2,...N-1*

Hence the next step is to compute the FFT of the audio signal and get the frequency variant of our audio signal. After this, The power spectral density of the audio signal is computed and is plotted. For this, we make use of Welch object to calculate the welch power spectral density. Welch's method is based on the concept of using periodogram spectrum estimates, which are the result of converting a signal from the time domain to the frequency domain. This method is an improvement on the standard periodogram spectrum estimating method and on Barlett's method, in that it reduces noise in the estimated power spectra, in exchange for reducing the frequency resolution. Due to the noise caused by imperfect and finite data, the noise reduction from Welch's method is often desired.

Having computed the FFT and the power spectral density of the first audio signal, we plot the corresponding spectra and repeat the above process i.e. right from recording for the second audio signal. Now, we start comparing the two audio signals by checking for the differences in the power spectral density and calculating the coefficient of cross-correlation between the frequency domain variants of the audio signals. In this project, we take the second signal as a function of lag and perform cross-correlation. Subsequently, we obtain the cross correlation coefficient of the two audio signals. The correlation coefficient gives an estimate of the amount of similarity that exists between the audio signals. Finally, the auto-correlation and cross-correlation coefficients are displayed as a 2x2 matrix. Mathematically, the formula for cross-correlation is given by the formula:

$$R_{12}(\tau) = \int_{-\infty}^{\infty} x_1(t)x_2(t-\tau)dt[\text{Positive shift}]$$

The MATLAB code corresponding to the above algorithmic description is given in the next page.

MATLAB Code:

```
clc; clear; close all;

disp('Recording...');

rec=audiorecorder(44100,16,1);
%Sampling rate = 44100, Bits per sample=16, Single channel

recordblocking(rec,5);
  %Recording audio for five seconds.

disp('Stopped recording. Now playing...');

play(rec);

  pause(5);
  %Program execution is paused until the recorded audio is played.

y=getaudiodata(rec);
  %Storing the recorded audio in an array.

l=length(y);

nfft=2^nextpow2(l);

f=22050*linspace(0,1,nfft/2+1);

xf=abs(fft(y,nfft));

figure(1);

plot(f,xf(1:nfft/2+1));

title('Frequency domain of first audio signal');

xlabel('Frequency'); ylabel('Amplitude');

h=spectrum.welch;

d=psd(h,y,'FS',44100);

figure(2);
```

Continued in the next page

```
plot(d); title('Power spectral density of first voice signal');
xlabel('Frequency'); ylabel('Power');
disp('Recording second time...');
rec2=audiorecorder(44100,16,1);
recordblocking(rec2,5);
disp('Stopped recording. Now playing...');
play(rec2);
pause(5);
y2=getaudiodata(rec2);
l2=length(y2);
nfft2=2^nextpow2(l2);
f2=22050*linspace(0,1,nfft2/2+1);
xf2=abs(fft(y2,nfft2));
figure(3);
plot(f2,xf2(1:nfft2/2+1));
title('Frequency domain of second audio signal');
xlabel('Frequency'); ylabel('Amplitude');
h2=spectrum.welch;
d2=psd(h2,y2,'FS',44100);
figure(4);
plot(d2);
title('Power spectral density of second voice signal');
```

Continued in the next page

```
xlabel('Frequency'); ylabel('Power');

c=xcorr(y,y2);

figure(5);

plot(c); title('Cross-correlation of audio signals');

e=xcorr(xf,xf2);
 %Performing cross-correlation taking xf2 as a function of lag.

figure(6);

plot(e);

   title('Cross-correlation of frequency domain representations of recorded audio signals');

R=corrcoef(xf,xf2);

   disp('Correlation coefficient matrix of frequency domain representations of recorded audio
 signals = ');

disp(R);
…….end of code.
```

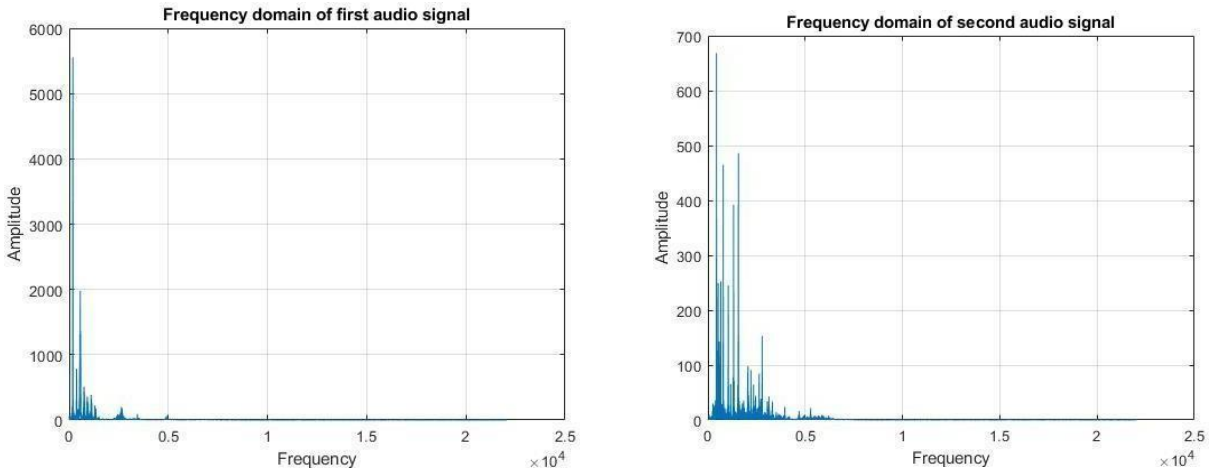
RESULTS

The above code was executed using MATLAB R2018b software where in the first audio signal is a voice signal and the other is of a mobile ringtone. The below results is displayed in the command window:

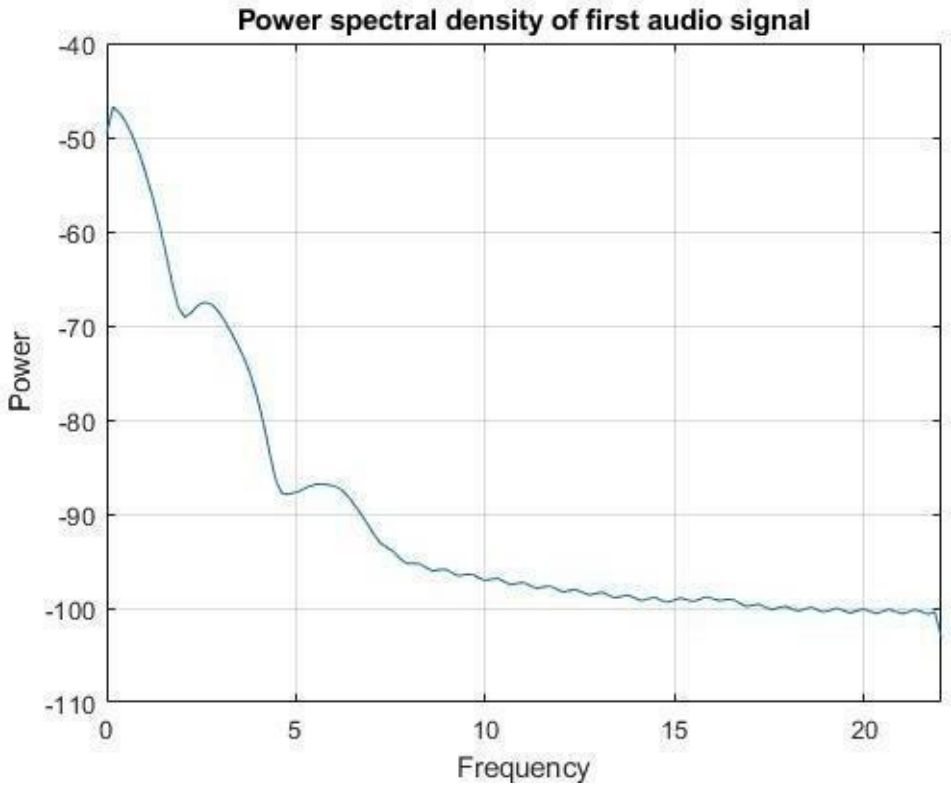
Recording...
Stopped recording. Now playing...
Recording second time...
Stopped recording. Now playing…
Correlation coefficient matrix:

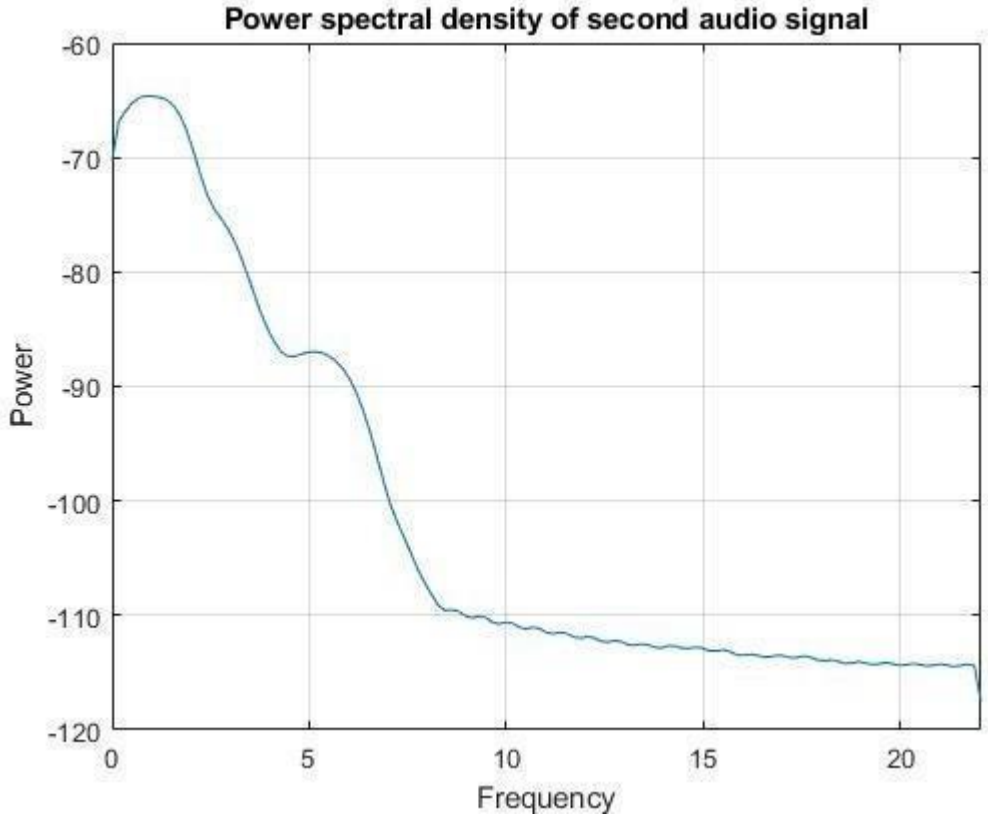
1.0000 (Auto-correlation of first signal)	*0.1179 (Cross-correlation. Second signal as function of lag)*
0.1179 (Cross-correlation. First signal as function of lag)	*1.0000 (Auto-correlation of second signal)*

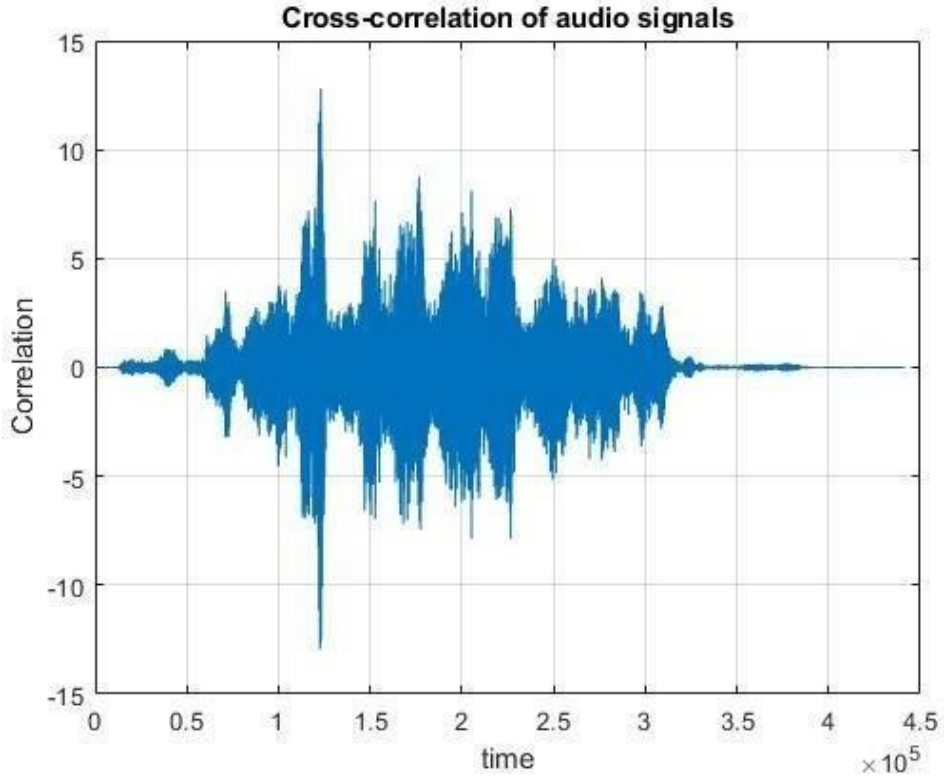
Here are the figures obtained after the execution of the MATLAB code:



Frequency domain variants of voice (left) and mobile ringtone (right)




Power spectral density of second audio signal
-60
-70
-80
-90
-100
-110
-120
Power
0
5
10
15
20
Frequency


Cross-correlation of audio signals
15
10
5
0
-5
-10
-15
Correlation
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
time
×10^5

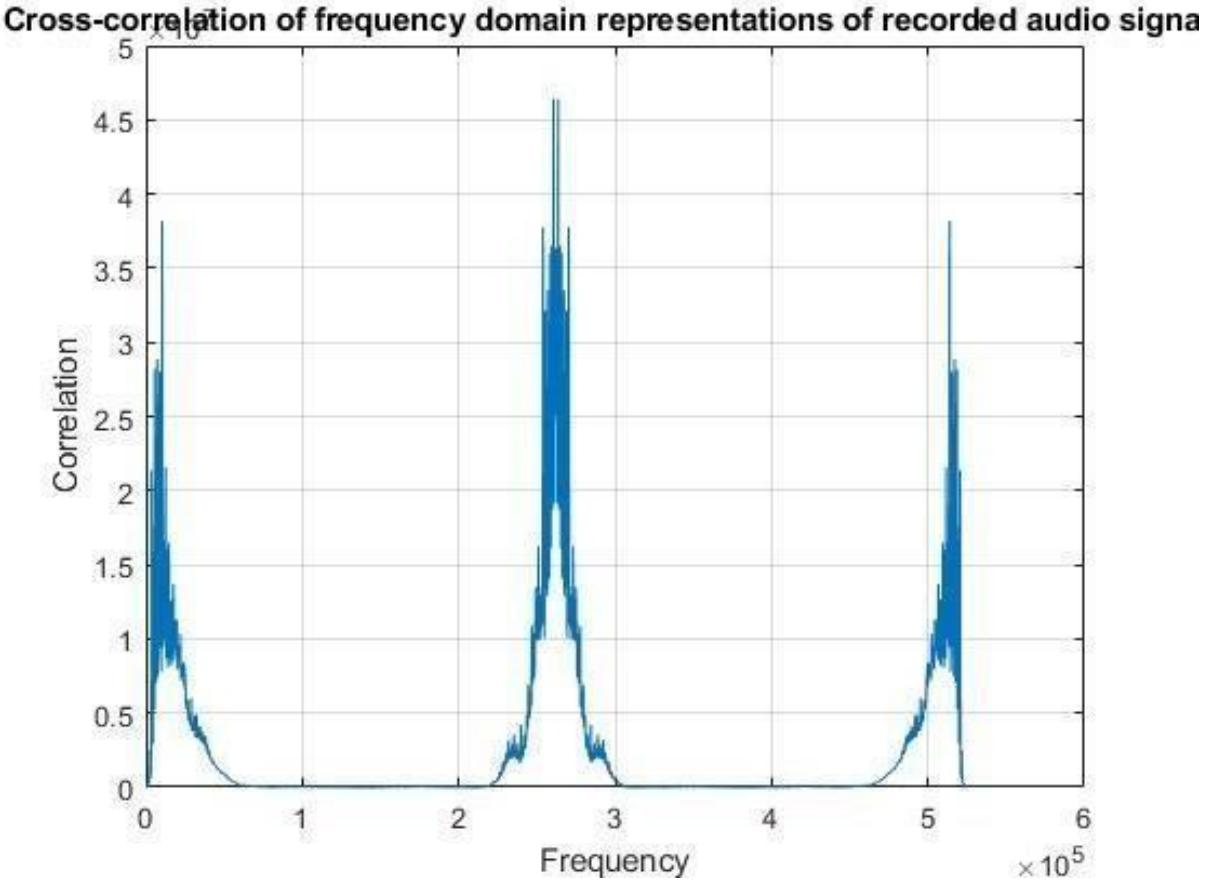


Fig. Cross-correlation of frequency domain variants

Further, it was observed that coefficient of cross-correlation is always greater than 0.7 if the two audio signals are speech signals of same person.

APPLICATIONS

Comparison of audio signals has several applications in audio signal processing which in turn can be applied in real world situations. Below are some of the examples of those applications:

- Automatic singing evaluation using cross-correlation. For example, pitch can be compared in frames if there is a single note. Extra the pitch for each frame, take the median, and compare to the target note.
- Audio signal comparison can be a part of the process of speaker recognition as it compares the audio signals based on mathematical parameters in frequency domain. Speaker recognition is the identification of a person from characteristics of voices. One of the main applications of speaker recognition is Voice authentication.
- The approach to this project i.e. correlation is applied in radar systems.

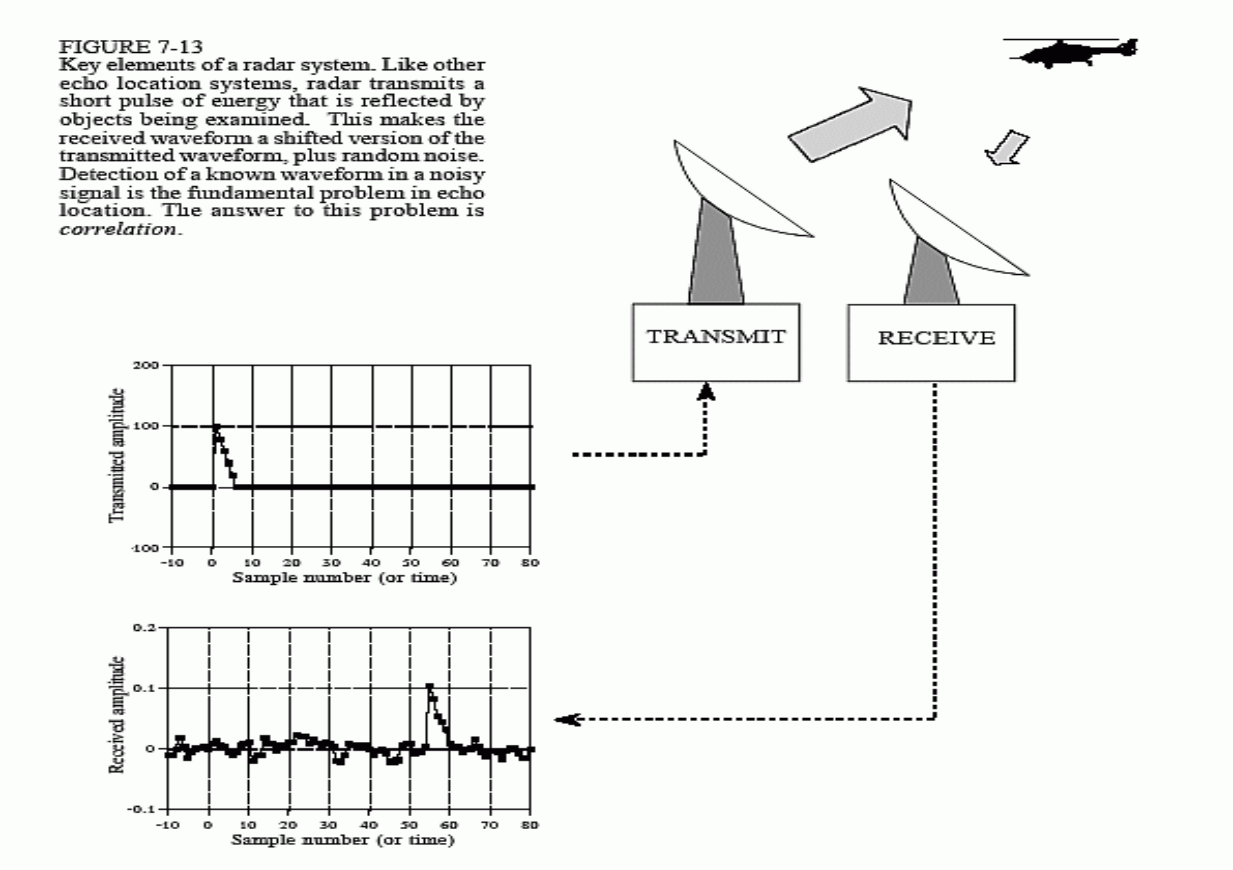


Image credits: DSP guide book by Steven W. Smith, MIT

CONCLUSION AND FUTURE SCOPE

Thus the two audio signals were compared based on their power spectral densities and the amount of similarity was estimated by utilizing the cross-correlation.

The project can be further extended to be applied in real world situations such as those mentioned under the applications part. The algorithm used takes into account the cross-correlation of two signals. This can be applied in Analog and Digital communications for comparing two radio waves wherever required. Also, the algorithm used in the project can be implemented on an FPGA to build a working model such as a cross-correlation system with comparators.

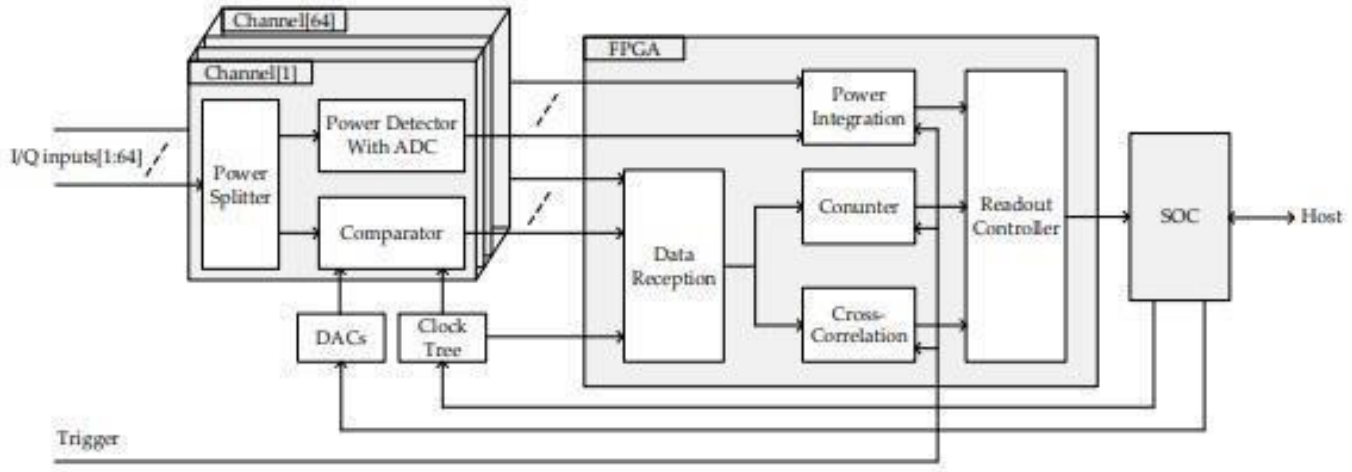


Figure 5. Block diagram of the correlation system.

BIBLIOGRAPHY

- Steven W. Smith, Ph.D. MIT, *The Scientist and Engineer's guide to Digital Signal Processing.*
- Ian Vince Mcloughlin, *Speech and Audio processing - A MATLAB based approach, Cambridge University Press.*
- Signal Processing Toolbox, *the MathWorks user's guide.*
- Wikipedia contributors, "Cross-correlation," *Wikipedia,The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Cross-correlation&oldid=936791961.
- MathWorks community, MATLAB Answers, *"How to tell if two signals are similar? ".*
- Wikipedia contributors, "Fast Fourier transform," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Fast_Fourier_transform&oldid=949417616.
- Wikipedia contributors, "Welch's method," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Welch%27s_method&oldid=943604784.

For any queries/feedback kindly contact

Mrs K.R. Deepthi,

Assistant Professor,

Department of ECE.

Mail ID: r.deepthi@staff.vce.ac.in

Phone No: 9493550926