

Autocorrelation of a Sound signal

Hema Kale¹, Dr. S.S. Limaye²

¹ ETC Department, Jhulelal Institute of Technology Nagpur, India

² Principal, Jhulelal Institute of Technology, Nagpur, India

ABSTRACT: Autocorrelation is a measure of how similar a signal is to itself. It is often used in [signal processing](#) for analyzing functions or series of values, such as [time domain](#) signals. In this paper a scheme for finding out the autocorrelation of a sound signal is proposed. The main aim of this scheme is to find out the frequency of a complex signal like a speech signal using relatively easily available tools like Computer, MATLAB, Mike etc. The scheme presents a design of an interesting experiment for the Digital Signal Processing Laboratory.

Keywords: DSP, Autocorrelation, cross correlation, convolution

I. INTRODUCTION

Our study of signal processing systems has been dominated by the concept of convolution and we have somewhat neglected its close relative 'correlation'. Convolution is generally between a signal and a filter, we think of it as a system with a single input and a stored coefficients. Correlation is usually between two signals, we think of it as a system with two inputs and no stored coefficients.

We have already learned that crosscorrelation is a measure of similarity between two signals, while autocorrelation is a measure of how similar a signal is to itself.

Autocorrelation is the [cross-correlation](#) of a [signal](#) with itself. It is the similarity between observations as a function of the time lag between them. It is a mathematical tool for finding repeating patterns, such as the presence of a periodic signal obscured by noise, or identifying the [missing fundamental](#) frequency in a signal implied by its [harmonic](#) frequencies. It is often used in [signal processing](#) for analyzing functions or series of values, such as [time domain](#) signals.

The idea of autocorrelation is to provide a measure of similarity between a signal and itself at a given lag. There are several ways to approach it, but for the purposes of pitch/tempo detection, one can think of it as a search procedure. In other words, we step through the signal sample-by-sample and perform a correlation between once reference window and the lagged window. The correlation at "lag 0" will be the global maximum because we are comparing the reference to a verbatim copy of itself. As we step forward, the correlation will necessarily decrease, but in the case of a periodic signal, at some point it will begin to increase again, then reach a local maximum. The distance between "lag 0" and that first peak gives us an estimate of our pitch/tempo.

Computing sample-by-sample correlations can be very computationally expensive at high sample rates, so typically an FFT-based approach is used. Taking the FFT of the segment of interest, multiplying it by its [complex conjugate](#), then taking the inverse FFT will give the [cyclic autocorrelation](#).

In the special case where $y(n) = x(n)$, we have the autocorrelation of $x(n)$, which is defined as the sequence

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) \quad (1)$$

or equivalently, as

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n+1)x(n) \quad (2)$$

In dealing with finite duration sequences, it is customary to express the auto-correlation and cross correlation in terms of the finite limits on the summation. In particular, if $x(n)$ and $y(n)$ are causal sequences of length N [i. e. $x(n) = y(n) = 0$ for $n < 0$ and $n \geq N$], then cross correlation and autocorrelation sequences may be expressed as

$$r_{xy}(l) = \sum_{n=1}^{N-|k|-1} x(n)y(n-l) \quad (3)$$

$$r_{xx}(l) = \sum_{n=1}^{N-|k|-1} x(n)x(n-l) \quad (4)$$

where $i = l, k = 0$ for $l \geq 0$, and $i = 0, k = l$ for $l < 0$

The autocorrelation sequence of a signal attains its maximum value at zero lag. This result is consistent with the notion that a signal matches perfectly with itself at zero shift.

In this paper a program is developed for finding out the frequency of a complex signal like a speech signal using relatively easily available tools like Computer, MATLAB, Mike etc. The objective of this paper is to design an interesting experiment for the Digital Signal Processing Laboratory.

The rest of the paper commences with section II giving simplified introduction of the scheme for finding out the autocorrelation and subsequently finding the frequency of a sound signal. Then simulation results are discussed in section III by considering two different signals one is a speech signal and other is a sound signal of an instrument. At the end conclusions are presented in section IV.

II. PROPOSED SCHEME

In this section the actual program for finding out the frequency of a complex signal such as a speech signal is presented. Autocorrelation is a measure of how similar a signal is to itself. It is the similarity between observations as a function of the time lag between them.

Every signal is 100% correlated with itself. So we should leave out peak of autocorrelation occurring at origin and find the next peak. Position of this peak corresponds to the period of the wave measured in number of samples. For finding out the frequency, we have to identify the first peak at which autocorrelation is maximum. This peak corresponds to the number of samples after which signal is going to repeat itself.

Here sampling frequency is taken as equal to 44.1 KHz.

Therefore,

$$\text{Period of the wave } T = \frac{m_i}{44100}$$

Where

m_i – Number of samples after which signal is going to repeat itself.

Consequently

$$\text{frequency of the wave (Hz)} = \frac{1}{T} = \frac{44100}{m_i}$$

For finding out the frequency of a sound signal following steps are given,

Connect Mike to Personal Computer (PC) and also install a sound recorder software in the PC.

Using Mike and sound recorder record audio signal.

Save the recorded audio file with .wav extension.

Then use MATLAB program for finding out autocorrelation of an audio signal with itself as below,

Suppose the name of the recorded sound file is violin.wav.

The proposed MATLAB program for finding out the frequency of a sound signal is as follows,

% Autocorrelation

clear all; close all; clc;

[m d] = wavfinfo('violin.wav') % Get the number of samples in the sound file

x1=wavread('violin.wav');

start=30000; %Leave first near about 30,000 samples in the sound file.

x=x1(start:start+1023); %Take 1024 samples of the sound file.

i=1:1024;

t=i/44100;

y=xcorr(x); % Find autocorrelation of the sound signal

y1 = y(1024:1024+600); % Take near about 600 samples of the autocorrelation function.

subplot(2,1,1);plot(t,x); % Plot the sound signal

xlabel('time');

ylabel('x');

subplot(2,1,2);plot(y1); % plot the autocorrelation function

xlabel('no. of samples');

ylabel('y1');

m=0;

% Finding out the maxima

% Leave near about first 50 samples of autocorrelation

% function and take next 550 samples. Compare each sample % with previous maximum and find out the maxima.

for(n=50:600)

if(y1(n)>m)

m=y1(n);

mi= n;

end

end

mi % no. of samples after which signal repeats itself.

f=44100/mi % Frequency of a sound signal

In this way frequency of a sound signal i.e. f can be obtained with the help of proposed program.

III. SIMULATION RESULTS

Simulations are carried out using the set up consisting of mike, speakers, PC, software 'MATLAB' and sound recorder 'Audacity'. Here sampling frequency is taken as equal to 44.1 KHz.

For the first set of simulations a speech signal is recorded. Then frequency of this recorded signal is found out using the proposed program in MATLAB. As shown below Fig.1 is plotted for a speech signal. Upper part of the figure i.e. figure a shows audio signal in time domain and lower part i.e. figure b shows autocorrelation of the recorded speech signal versus number of samples. In this figure b leave the first 50 samples, then after that observe the position of the first peak. This position of the first peak gives the value of mi and then frequency of the recorded signal is calculated using the equation,

$$f \text{ (Hz)} = \frac{44100}{m_i}$$

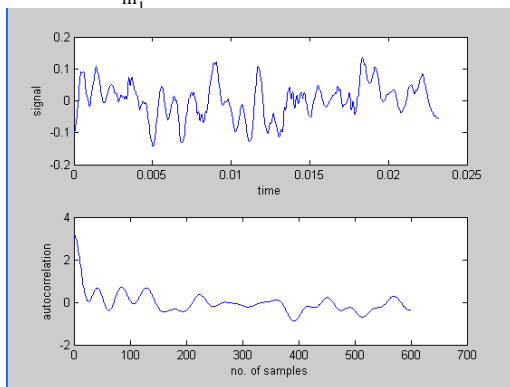


Figure 1. a) Input audio signal in time domain, b) Autocorrelation of the signal versus no.of samples.

For the recorded speech signal simulation results shows that the recorded file contains 159490 samples. Position of the first peak is obtained at mi= 85. Therefore resultant frequency of speech signal comes out to be equal to 518.8235 Hz.

For the second set of simulations the sound of a musical instrument 'violin' is recorded. Then frequency of this recorded signal is found out using the same procedure explained above. As shown below Fig. 2 is plotted for the sound of instrument 'violin'.

For the recorded sound signal simulation results shows that the recorded file contains 73728 samples. Position of the first peak is obtained at mi= 152. Therefore resultant frequency of a sound signal comes out to be equal to 290.1316 Hz.

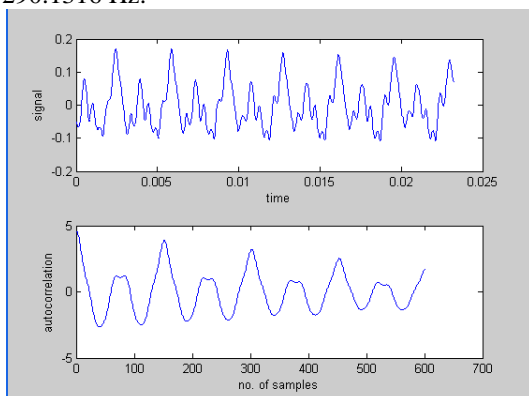


Figure 2 (a) Input sound signal in time domain (b) Autocorrelation of the signal versus no. of samples.

IV. CONCLUSION

In this paper a scheme is proposed for finding out the frequency of a recorded signal such as speech signal. Using data acquisition toolbox in MATLAB, the scheme can be further modified for finding the frequency of a Real Time Signal.

REFERENCES

Lab Manuals:

[1] Digital Signal Processing Lab Manual.

Books:

[2] Alan V. Oppenheim and Ronald W. Schaffer "Discrete-Time Signal Processing", 3rd Edition, Pearson, 2009.

[3] John G. Proakis and Dimitris G. Manolakis. "Digital Signal Processing. Principles, Algorithms, and Applications", 4th Edition, Pearson, 2007.