A Comparative Study on Discrete Fourier Transformation for Digital Signal Analysis

Saiful Islam*[1], Md. Sirajul Islam[2]

[1]*Department of Computer Science & Engineering, Stamford University Bangladesh, Dhaka 1217, Bangladesh*

[2]*Department of Mathematics, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj 8100, Bangladesh*

*Corresponding Author: Saiful Islam

Abstract- *In this article, the basic information on discrete signals, discrete Fourier series, discrete Fourier transformation and their computational implement of signal processing system are described. Now a day, digital signal processing (DSP) is an important research topic because it significantly increases the overall value of hearing protection. From millions of signals, DSP suppresses noise without blocking the speech signal easily. Again without compromising communication, DSP systems protect the users from unhealthy noise exposure. This study addressed some mathematical and graphical techniques for discrete signals reconstruction by using Discrete Fourier Transformation (DFT). DFT is one of the most popular analyzed techniques for DSP system. In this work, we will try to separate the input signal into the real and imaginary part by using DFT algorithm in MATLAB. On the other hand, we will try to reconstruct the given discrete signal with the help of MATLAB program with graphical representation.*

Key words- *Signal, Discrete Signals, Discrete Fourier series, Discrete Fourier transformation, Signal Reconstructions*

1. Introduction

Now a day, signal reconstruction from partial Fourier domain information has been interesting to a number of different authors both for particular applications and for its inherent theoretical value [1]. Previous work in this area has involved developing conditions under which signals are uniquely specified with Fourier transform magnitude or phase [2] or signed magnitude information and developing practical algorithms for recovering signals from this information. In this paper, we consider the problem of reconstructing signals from only discrete Fourier transform (or inverse discrete Fourier transform) sign information [3]. By interpolated DFT method with maximum sidelobe decay windows, a multi-frequency signal was analyzed [4]. Again, to improve the accuracy of periodic signal analysis another algorithm was performed [5]. This proposed approach required quite modest additional computational burden which make it suitable for real-time signal professing. Firstly, they showed that how the proposed method can be used in the case of DFT analysis of harmonic signals, and secondly, it was considered that the digital wattmeter application area in electrical power-system measurement. To analyze the exponential signal by the interpolated DFT algorithm another method was described [6]. In [7], DFT algorithm analyzed in low-cost power quality measurement systems based on a DSP processor.

The discrete Fourier transform (DFT) is the most important discrete transform, used to perform Fourier analysis in many practical applications. In DSP, the function is considered as any quantity or signal which varies over time, such as a radio signal, the pressure of a sound wave or daily temperature readings, sampled over a finite time interval (generally defined by a window function).

In case of image processing, the samples can be the values of pixels along a row or column of a raster image. As well as in DFT, exponential function or periodic signal converted into sine and cosine functions or $A+iB$. The discrete signal $x[n]$ (where n is time domain index for discrete signal) of length N is converted into discrete frequency domain signal of length N . The relation between input samples with sine and cosine the complex DFT output signal is given by [8]

$$X(k)=\frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{\frac{-i2\pi nk}{N}}=\frac{1}{N}\sum_{n=0}^{N-1}x[n][\cos\frac{2\pi nk}{N}-i\sin\frac{2\pi nk}{N}];\text{ where } 0\le k\le N-1$$

DFT tool convert time domain sampled data into frequency domain sampled data and vice versa. Application of DFT in the field of digital spectral analysis is Spectrum Analyzers, Speech Processing; Noise removing, Imaging and Pattern Recognition.

In second chapter some necessary preliminary definitions are described. In third chapter, an overview on discrete Fourier transform is given. After that by applying DFT algorithm, we have tried to separate the input signal into the real and imaginary part which is given in fourth chapter. Conversely, we have tried to reconstruct the given discrete signal with the help of MATLAB program with graphical representation in this part. Finally, a conclusion is added in chapter five.

2. Basic Definitions

2.1 Signals: Any real or complex function of the real time variable t is called signal. One the other hand, a signal is a single-valued function of one or more independent variables. An example of a signal $f(t)=\sin(2\pi t-t^2)$ is given.

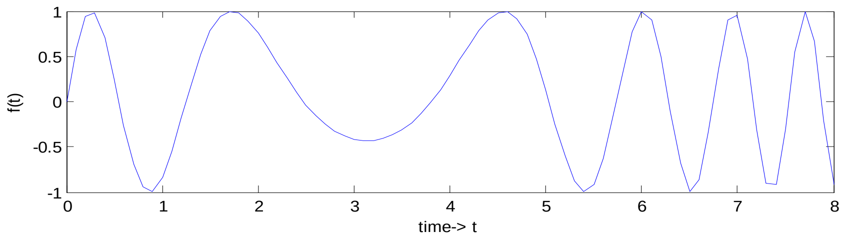

Fig. 1: Graph of the signal $\sin(2\pi t-t^2)$.

2.2 Continuous signals: A continuous-time signal is a function $f(t)$ of the real variable t defined for for $-\infty<t<\infty$. A continuous signal $f(t)=\sin\ 2\pi t$ is given in the following figure.

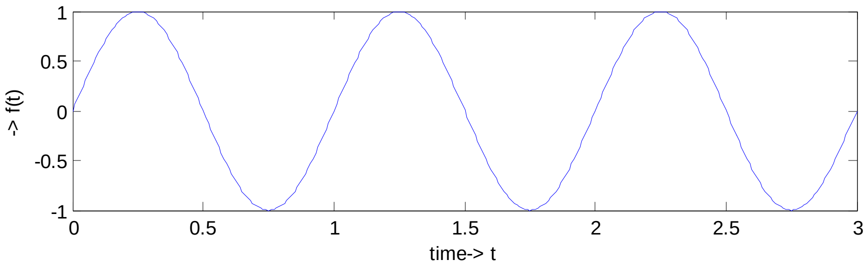

Fig. 2: A graph of the signal $\sin\ 2\pi t$.

2.3 Discrete time Signals: A discrete-time signal is a sequence of values of interest, where the integer index can be thought of as a time index, and the values in the sequence represent some physical quantity of interest. A discrete-time signal is a sequence $x[n]$ defined for all integers $-\infty<n<\infty$. We display $x[n]$ graphically.

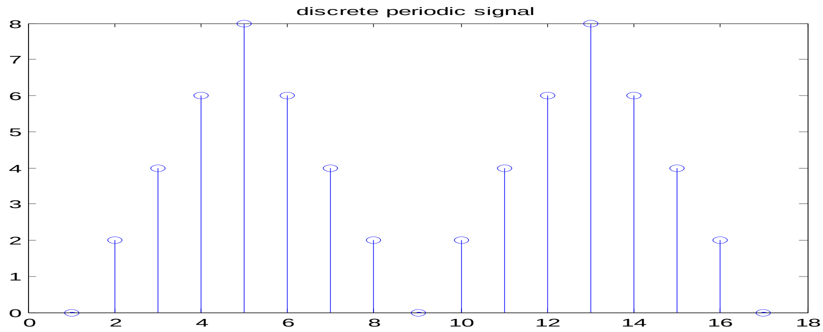

Fig. 3: A graph of a discrete signal.

2.4 Fourier series: Fourier analysis is named after **Jean Baptiste Joseph Fourier** (1768-1830), a French mathematician and physicist. If $f(t)$ is a real periodic function with period T, then $f(t)$ can be expanded in Fourier series of the form

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{n=\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t) \quad (1)$$

where fundamental frequency, $\omega_0 = \frac{2\pi}{T}$, harmonic frequency $n\omega_0$ and the coefficients are defined by

$$a_0 = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)dt; \quad a_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\cos n\omega_0 dt; \quad \text{and} \quad b_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\sin n\omega_0 dt, n \in N$$

2.5 Fourier transformation: The following equations give the general form of the Fourier transform [9] and inverse Fourier transforms, in terms of time *t* and frequency *ω*:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \quad (2)$$

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega \quad (3)$$

Here, equation (2) and equation (3) are known as Fourier transformation and inverse Fourier transformation respectively. The standard Fourier transform converts a function of time $f(t)$ to a function of frequency $F(\omega)$.

2.6 Discrete Fourier series: For the given sequence $f[n]$ with periodicity N, we have $f[n+mN] = f[n], m \in ¢$, then the Fourier expansion of $f[n]$ is

$$f[n] = \sum_{k=0}^{N-1} \alpha_k e_k[n] \quad (4)$$

where the Fourier basis functions are

$$e_k[n] = e^{\left(\frac{2\pi ikn}{N}\right)} \tag{5}$$

And the Fourier coefficients $\alpha_k = \left\langle f[n], e^{\left(\frac{2\pi ikn}{N}\right)} \right\rangle = \frac{1}{N}\sum_{k=0}^{N-1} f[n]e^{\left(\frac{2\pi ikn}{N}\right)}$ (6)

3. Overview on Discrete Fourier Transform

Before we introduce the DFT we consider the sampling of the Fourier transform of a periodic discrete-time sequence. For this reason, a relation between the sampled Fourier transform and the DFT has already established. When a signal is periodic in time domain then it is possible to use discrete time Fourier series (DTFS) representation. Then the frequency domain spectrum will be discrete and periodic. If the signal is non-periodic or of finite duration the frequency domain representation is periodic and continuous this is not convenient to implement on the computer. By using this periodicity property of DTFS representations exactly, the finite duration sequence will be possible to represent in frequency domain. Then this is mentioned as a discrete Fourier transformation.

This is known to all that DFT is an essential mathematical tool. This can be used for software execution for a definite digital processing system (DSP). DFT gives a method to transform a given 19 sequence to frequency domain and to represent the spectrum of the sequence using only k frequency values, where k is an integer that takes N values, $K = 0, 1, 2, \dots\dots N-1$. The advantages of DFT are:

1. It is computationally convenient.
2. The finite length sequence of DFT has made the frequency domain analysis much easier than the existing technique of continuous Fourier transform.

3.1 Discrete Fourier transformation

Let $x[n]$ be a discrete-time signal with period N. Then the discrete Fourier transformation $X[k]$ of $x[n]$ is the discrete-time signal defined by

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi n}{N}k} \tag{7}$$

where N is called the size of the DFT. The DFT $X[k]$ of $x[n]$ is a sequence of complex numbers. Here $x[n]$ is assumed to be zero outside the input data window $n = 0,1,\dots\dots, N-1$. Signal samples can be computed from the DFT coefficients using a similar equation as follows:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi n}{N}k}, \tag{8}$$

where $n = 0,1,\dots\dots, N-1$. Also, N is called the size of the DFT. Here equation (8) is known as inverse discrete Fourier transformation. In the domain $n \in [0, N-1]$ this is the **inverse transform** of equation (7). In this interpretation, each $X[k]$ is a complex number that encodes both amplitude and phase of a sinusoidal component $e^{\frac{j2\pi kn}{N}}$ of function $x[n]$. The sinusoid's frequency is k cycles per N samples.

3.2 Frequency Response of Signals

Generally, systems are analyzed in the time domain by using convolution. A parallel analysis can be done in the frequency domain. *By* using Fourier transform, every input signal can be represented as a group of cosine waves, each with a specified amplitude and phase shift. As well, the DFT can be used to represent every output signal in a similar form. This implies that any linear system can be completely described by how it changes the amplitude and phase of cosine waves passing through it. This information is known as the system's frequency response. Since both the frequency response and the impulse response contain complete information about the system of signals, there must be a one-to-one correspondence between the two. If one is given, you will calculate the other. The connection between the frequency response and the impulse response is one of the foundations of signal processing. A system's frequency response is the Fourier Transform of its impulse response which is illustrated by these relationships in Fig 4.

Keeping with standard DSP notation, lower case variables are used to represent impulse responses, while upper case variables are used to represent the corresponding frequency responses. Since $H[\,]$ is the common used for the frequency response and $h[\,]$ is the common symbol for the impulse response, systems are described in the time domain by convolution, which is: $x[n] * h[n] = y[n]$. In the frequency domain, the input spectrum is multiplied by the frequency response as an equation: $X[f] \times H[f] = Y[f]$ resulting in the output spectrum.

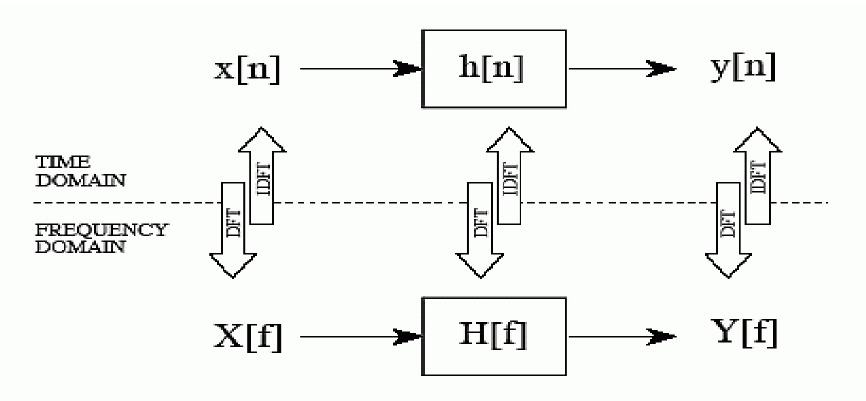

Fig. 4: Comparing system operation in the time domains.

In the time domains, an input signal is convolved with an impulse response, resulting in the output signal, that is, $x[n] * h[n] = y[n]$. Again, in the case of frequency domain the input spectrums are multiplied by a frequency response function $H[f]$ which gives us the resulting output spectrum that is, $X[f] \times H[f] = Y[f]$. The DFT and inverse DFT relate the signal into two domains [10].

4. The Discrete Fourier Transforms Algorithm

DFT is a simple algorithm. It consists of stepping through the digitized data points of the input function, multiplying each point by sine and cosine functions as we go along and summing the resulting products into accumulators (where one for the sine component and the another one for the cosine). When we have processed every data point in this manner, generally we divide the accumulators (i.e. the sum-totals of the preceding process) by the number of data points. Then the resulting quantities are the average values for the sine and cosine components at the frequency being investigated. We must repeat this process for all integer multiple frequencies up to the frequency that is equal to the sampling rate minus 1 (i.e. twice the Nyquest frequency minus 1), and the job is done.

4.1 The DFT computer program

The MATLAB program for the signal $\cos(t)+\frac{1}{9}\cos(at)+\frac{1}{25}\cos(bt)+\frac{1}{49}\cos(ct)$ is given below:

```
function DFT
clc
clear all
close all
a=input('');
b=input('');
c=input('');
pi=3.1416;
k1=pi/8;
k2=1/pi;
for i=1:16
x(i)=(i-1)*k1;
y(i)=f(x(i));
end

for i=1:16
    if(rem(i,2)~=0 && i<=7)
        kc(i)=1/(i*i);
    else
   kc(i)=0;
end
    ks(i)=0;
end

for j=1:16
   fc(j)=0;
   fs(j)=0;
for i=1:16
      fc(j)=fc(j)+y(i)*cos((j-1)*(i-1)*k1);
      fs(j)=fs(j)+y(i)*sin((j-1)*(i-1)*k1);
     end
 fc(j)=fc(j)/16;
 fs(j)=fs(j)/16;
end

for i=1:16
  zr(i)=0;
  zi(i)=0;
  for j=1:16
      zr(i)=zr(i)+fc(j)*cos((i-1)*(j-1)*k1)+fs(j)*sin((i-1)*(j-1)*k1);
     zi(i)=zi(i)-fc(j)*sin((i-1)*(j-1)*k1)+fs(j)*cos((i-1)*(j-1)*k1);
    end
end
fprintf('frequency\t\tF(cos)\t\t\t\tF(sin)\t\tkc\t\t\tks\t\t\ty\t\t\tzr\t\t\tzi\n')

for i=1:16
fprintf('%d\t\t\t\t%.3f\t\t\t\t%.3f\t\t%.3f\t\t%.3f\t\t%.3f\t\t%.3f\t\t%.3f\n',i-
1,fc(i),fs(i),kc(i),ks(i),y(i),zr(i),zi(i));
end

function z=f(x)
    z=cos(x)+(1/9)*cos(a*x)+(1/25)*cos(b*x)+(1/49)*cos(c*x);
end
end
```

After executing the signal $\cos(t)+\frac{1}{9}\cos(at)+\frac{1}{25}\cos(bt)+\frac{1}{49}\cos(ct)$ in the MATLAB, we get the following Fig. 5.

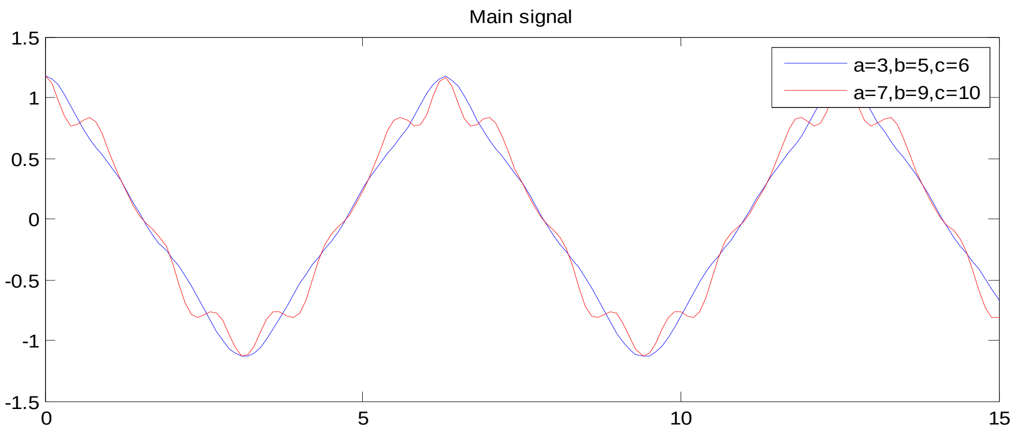

Fig. 5: Graph of the signal $\cos(t)+\frac{1}{9}\cos(at)+\frac{1}{25}\cos(bt)+\frac{1}{49}\cos(ct)$.

After execute the above program in MATLAB we get the following figure:

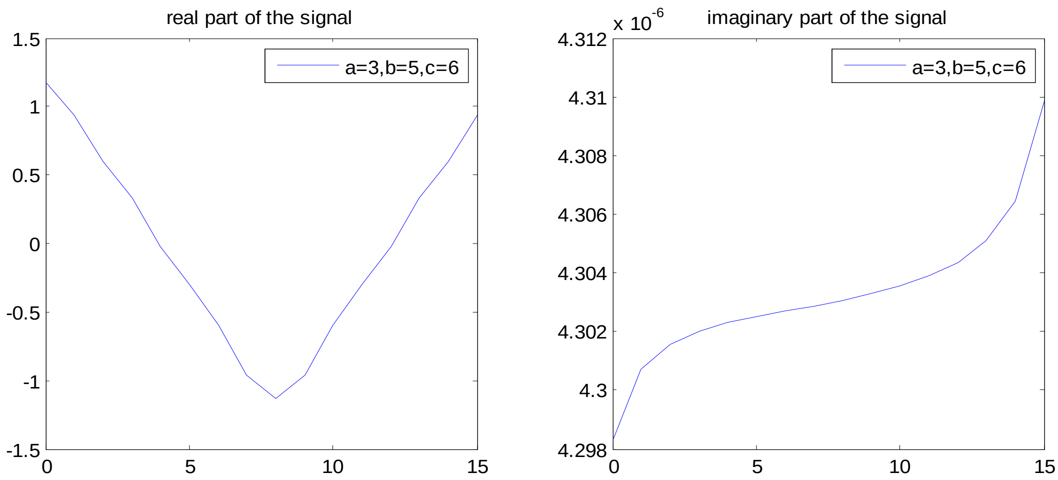

Fig. 6: Real part and imaginary part of the signal $\cos(t)+\frac{1}{9}\cos(at)+\frac{1}{25}\cos(bt)+\frac{1}{49}\cos(ct)$.

4.2. Reconstructions of Signal

The inverse transform is, intuitively, a very simple operation. We know what the amplitudes of the sinusoids are (from the forward transform), so we simply reconstruct all of these sinusoids and sum them together. Nothing could be simpler. It is noted that the process of extracting the individual frequency components yielded only half amplitude values for all although the constant term and the Nyquist frequency (where Harry Nyquist was an well known electronic engineer and according to his name Nyquist frequency was addressed), is half of the sampling rate of a discrete signal processing system term. However, we will extract components for both the negative and positive frequencies (i.e. both above and below the Nyquist).

The inverse discrete Fourier transform for the signal $\cos(t)+\frac{1}{9}\cos(at)+\frac{1}{25}\cos(bt)+\frac{1}{49}\cos(ct)$ is given below:

```
function IDFT
clc
clear all
pi=3.1416;
k1=pi/8;
k2=1/pi;
a=3;
b=5;
c=6;
for i=1:16
   x(i)=(i-1)*k1;
   y(i)=f(x(i));
end

for i=1:16
   if(rem(i,2)~=0 && i<=7)
      kc(i)=1/(i*i);
   else
   kc(i)=0;
  end
ks(i)=0;
end
for j=1:16
   fc(j)=0;
 fs(j)=0;
for i=1:16
           fc(j)=fc(j)+y(i)*cos((j-1)*(i-1)*k1);
           fs(j)=fs(j)+y(i)*sin((j-1)*(i-1)*k1);
       end
   fc(j)=fc(j)/16;
   fs(j)=fs(j)/16;
end

for i=1:16
   z(i)=0;
   for j=1:16
z(i)=z(i)+fc(j)*cos((i-1)*(j-1)*k1)+fs(j)*sin((i-1)*(j-1)*k1);
end
end
fprintf ('frequency\ty(I)\t\tz(I)\n')

for i=1:16
   fprintf ('%.1f\t\t\t%.3f\t\t%.3f\n',i-1,y(i),z(i));
end
function z=f(x)
z=cos(x)+(1/9)*cos(a*x)+(1/25)*cos(b*x)+(1/49)*cos(c*x);
end
end
```

After executing the MATLAB program, we get the following figure:

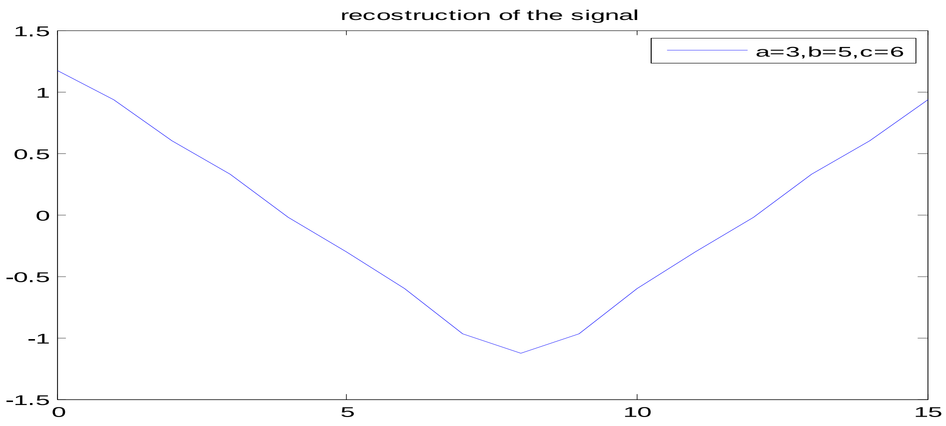

Fig. 7: Reconstruction of the signal.

5. Conclusion

This study proposes a method to detect the component of continuous signals and discrete signals. A discrete periodic signal can be used when only one period of the signal is analyzed. If we want to reconstruct some component from a signal then we can apply the discrete Fourier transformation algorithm. We have described different types of signals, Fourier series, Fourier transformation and discrete Fourier transformation (DFT). Finally, an algorithm for reconstructing a signal is described and explained by graphically with the help of MATLAB program. Our main object was to reconstruct a signal by DFT algorithm and finally we completed this work.

6. References

[1] Oppenheim, A. V., Lim, J. S., & Curtis, S. R. (1983). Signal synthesis and reconstruction from partial Fourier-domain information. *JOSA*, *73*(11), 1413-1420.

[2] Hayes, M., Lim, J., & Oppenheim, A. (1980). Signal reconstruction from phase or magnitude. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(6), 672-680.

[3] Hayes, M. H. M. H. (1982). The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transforms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *30*(2), 140-154.

[4] Belega, D., & Dallet, D. (2009). Multifrequency signal analysis by interpolated DFT method with maximum sidelobe decay windows. *Measurement*, *42*(3), 420-426.

[5] Xi, J., & Chicharo, J. F. (1996). A new algorithm for improving the accuracy of periodic signal analysis. *IEEE Transactions on Instrumentation and Measurement*, *45*(4), 827-831.

[6] Wu, R. C., & Chiang, C. T. (2010). Analysis of the exponential signal by the interpolated DFT algorithm. *IEEE Transactions on Instrumentation and Measurement*, *59*(12), 3306-3317.

[7] Szmajda, M., Górecki, K., & Mroczka, J. (2007). DFT algorithm analysis in low-cost power quality measurement systems based on a DSP processor. In *2007 9th International Conference on Electrical Power Quality and Utilisation* , IEEE. pp. 1-6.

[8] M. H. Hayes (1988). Signal Reconstruction from Phase or Magnitude ScD. Thesis, MIT, Dept. Elec. Eng. & Comp. Sci, June.

[9] John G. Proakis., & Dimitris K Manolakis. (2006). “Digital Signal Processing” (4th Edition), ISBN-13: 978-0131873742.

[10] Smith, W. (1999). The Scientist and Engineer’s Guide to Digital Signal Processing.