

Fast Fourier Transform Approach for Signal Analysis

K. R. Chaudhari

Assistant Professor, Department of Electronics, Shree J. P. Arts & Science College, Bharuch, Gujarat, India
Affiliated to Veer Narmad Shouth Gujarat University, Surat, Gujarat, India

ABSTRACT

Fourier analysis converts time domain signal into frequency domain signal and vice versa and fast Fourier transform (FFT) is a mathematical tool which convert discrete time domain signal into frequency domain signal. FFT are widely used for many applications such as mathematics, engineering and science. FFT have been described as the most important numerical algorithms of our life. This paper focused on FFT and described about FFT and its MATLAB implementation. The FFT spectrums for the outputs are analyzed.

Keywords : FFT, DFT, Twiddle factor, MATLAB, Window.

I. INTRODUCTION

A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and IFFT is used for inverse discrete Fourier transform (IDFT) (Steven W. Smith, 1997). Classification of Fourier analysis and Fast Fourier Transform (FFT) is shown in figure 01 and 02. A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and IFFT is used for inverse discrete Fourier transform (IDFT). Classification of Fast Fourier Transform (FFT) is shown in figure.

Fast Fourier transform (FFT) is a algorithm, which is used for fast calculation and efficient computation of discrete Fourier transform (DFT). Two point DFT and 4-point DFT can be generalized in to 8-point, 16-point or 2r points. The direct computation of the DFT requires $2N^2$ evaluations of trigonometric function, $4N^2$ real multiplications and $4N(N - 1)$ real additions. This is primarily inefficient as it does not exploit the symmetry and periodicity properties of the phase factor W_N , which are given below:

$$\text{Periodicity Property } W_N^{k+N} = W_N^k$$

$$\text{Symmetry Property } W_N^{k+\frac{N}{2}} = -W_N^k$$

An efficient algorithm for DFT computation is the fast Fourier transform algorithm because FFT algorithms exploit the above two properties. FFT algorithm requires $(N/2) \log_2 N$ complex multiplications and $N \log_2 N$ complex additions.

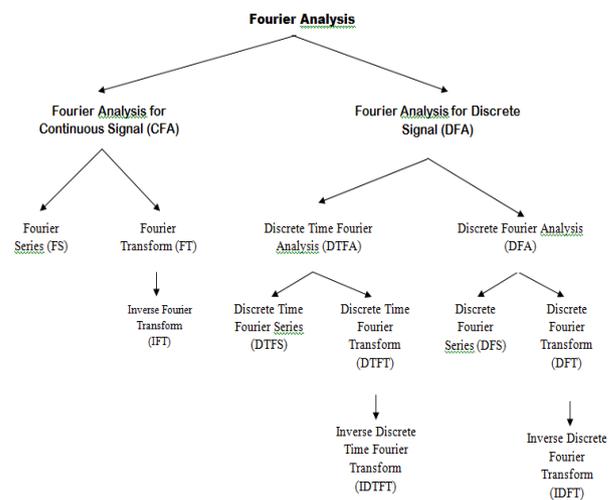


Figure 1 : Classification of Fourier analysis

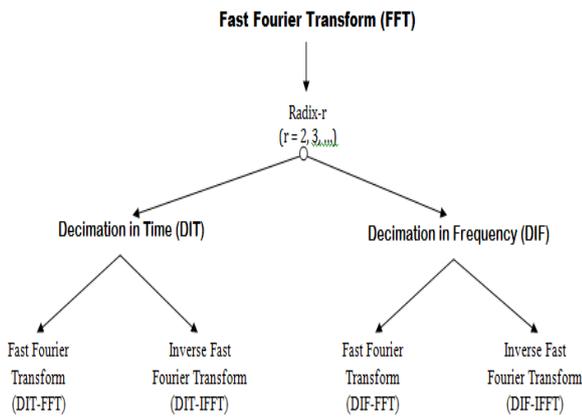


Figure 2 : Classification of Fast Fourier Transform (FFT)

There are two methods for computation of FFT algorithms. First is decimation in time (DIT) and second method is decimation in frequency (DIF). Let's discuss one of them, the "DIT" FFT algorithm for sequences whose length is a power of two ($N = 2^r$ for some integer r). Figure 03 shows 8-point DIT FFT,

Where

$$W = W_8 = e^{-\frac{j\pi}{4}} = \frac{(1 - j)}{\sqrt{2}}$$

The Fast Fourier Transform algorithm is used for fast convolution (linear, shift-invariant filtering).

Concept for convolution is $h = f \star g$,

Formula for convolution of continuous time signal is

$$h(x) = \int_{-\infty}^{+\infty} f(t)g(x-t)dt,$$

but convolution of discrete signals involves a sum:

$$h[x] = \sum_{t=-\infty}^{\infty} f[t]g[x-t].$$

When signal f and g in formula have same length N that mean convolution working with finite sequence. When signals are periodic so f and g "wrap around". Then circular convolution introduced. Formula for circular convolution is given by

$$h[x] = \sum_{t=0}^{N-1} f[t]g[x-t \text{ mod } N] \quad \text{for } x = 0 \dots N-1.$$

Convolution can be determining using Fourier transform. For $h = f \star g$ take Fourier transform of h , f and g then simple multiplication of F and G is H ($H \leftrightarrow FG$). Corresponding procedure is same for linear as well as circular convolution. Similarly DFT of circular convolution of two signals is the multiplication of their DFT's. But in DFT there are N^2 real multiplications and additions. If we compute the DFT of sequence f and the DFT of sequence g , multiply them point-by-point, and then compute the inverse DFT, we'll get the same answer. This is referred Fourier convolution. Figure 05 shows Fourier convolution.

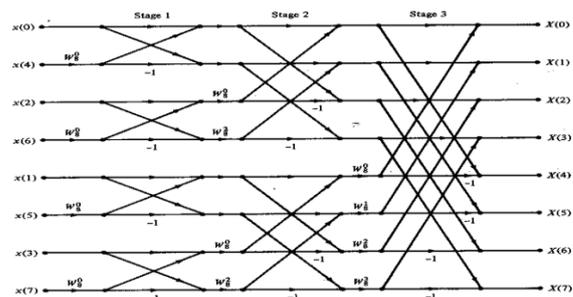


Figure 3. Flow diagram of Radix-2, 8-point DIT FFT

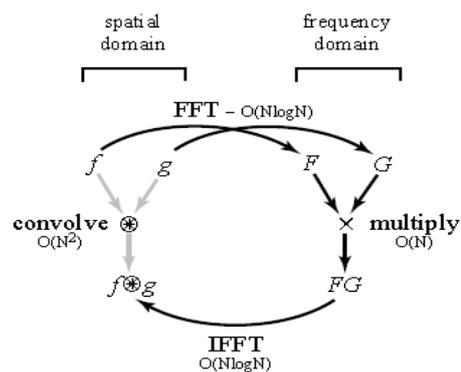


Figure 4. Fourier Convolution

Using FFT algorithm, the two DFT's and the one inverse DFT have a total cost of $6 N \log_2 N$ real multiplies, and the multiplication of transforms in the frequency domain has a negligible cost of $4N$ real multiplies. Here we require N^2 multipliers. For large N ,

it is cumbersome steps but this algorithm can be modified to do standard “linear” convolution by padding the sequences with zeros appropriately. FFT overcome all difficulties.

II. METHODS AND MATERIAL

Interpretation of Result of FFT

In interpretation of result, it is important factor that how FFT shows the results of transform and correctly interpret. Here MS excel® and Matlab® is used for implementation of Radix-2 DIT FFT. Assume in frequency domain number of points is k . For Signal is sampled with frequency $f_s = 1 / \Delta t$, then frequency resolution is $\Delta f = f_s / k$. The maximum frequency of the spectrum is $= f_s / 2$ (Nyquist frequency). N is length of sequence then k vary from 0 to $N - 1$. FFT has summation of terms then FFT must be scaled by dividing them by the number of points, k . Spectrum components of FFT is as follows:

- (i) When $f = 0$, it has to be real and imaginary components is zero.
- (ii) At $k/2$ points are for positive frequencies in increasing order $\Delta f, 2\Delta f, 3\Delta f, \dots, (k/2-1)\Delta f, (k/2)\Delta f$ or $f_s/2$. The element for the Nyquist frequency is real and is shared with the one for $f = -f_s/2$, since they are equal.
- (iii) The next $k/2-1$ elements contain the negative frequency components. They are also placed in mathematically increasing order from the most negative frequency $f = -(k/2-1)\Delta f$ until $f = -\Delta f$. In other words, they are sorted from maximum negative to minimum negative frequency. In this sense they appear to be in reverse order of frequency.

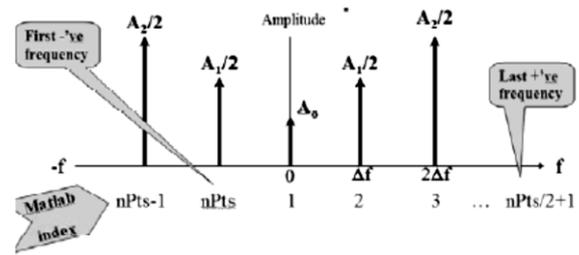


Figure 5. Interpretation of Result of FFT

III. RESULTS AND DISCUSSION

Investigation of the Behavior of the FFT Algorithm In MATLAB

Matlab’s function `fft ()` performs the discrete Fourier transform (DFT). For better understanding of fast Fourier transform (FFT) algorithm (With the help of functions in Matlab – Structured Programming and simple example) generate sinusoidal signal with the known amplitude, frequency and phase, and then subject it to FFT and study its behavior to recover the characteristics of the signal. A function called ‘sin generator’.

First decide the sampling frequency FFT size signal. Figure 06 shows sinusoidal sin signal with amplitude is 5, frequency is 100, phase zero, sampling frequency is 1000 and number of points is 2^8 . Then calculate Nyquist rate and Δf . For Δf which is sampling frequency / size of FFT. Nyquist rate is sampling frequency / 2. Then apply FFT. Keep the first $(\frac{FFT\ size}{2} + 1)$ elements and truncate the rest. For scaling divide all elements by FFT size. For obtaining folding FFT Spectrum into a One-Sided spectrum, leave the first element and the last (Nyquist) unchanged and multiply all elements by 2. For obtaining magnitude and phase plot apply the formula. Calculate magnitude for each component, which is

$$\sqrt{real\ component^2 + imaginary\ component^2} .$$

Similarly calculate other spectral parameters as required i. e. phase is $\arctan\left(\frac{imag}{real}\right)$.

Calculate $\left(\frac{FFT\ size}{2} + 1\right)$ values of frequency from 0 to Nyquist with a step of Δf . Finally plot the spectrum as required. Result of frequency spectrum shown in figure 07.

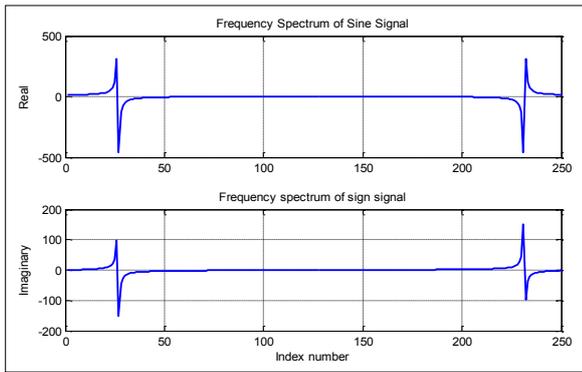


Figure 6. Real and imaginary components of FFT spectrum as returned by fft()

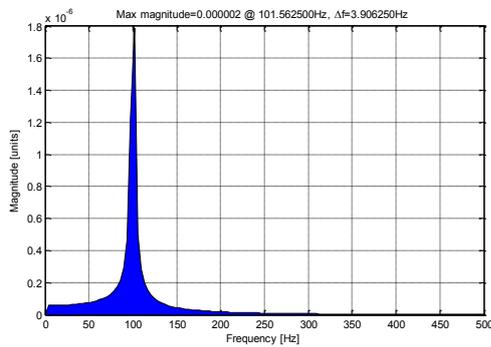


Figure 7. Magnitude spectrum: stem

Comments on above result is that ideal magnitude spectrum of the analyzed signal should be a single line at 100Hz. Simple **Simulink model of FFT** is shown in **figure 08**. The box referred by display block, which shows the value of its input on its icon. Amplitude 100, frequency $50 \times 2\pi$ rad/sec and phase $\pi/2$ sinusoidal signal is given to FFT block. Result of simulation is shown in figure 09.

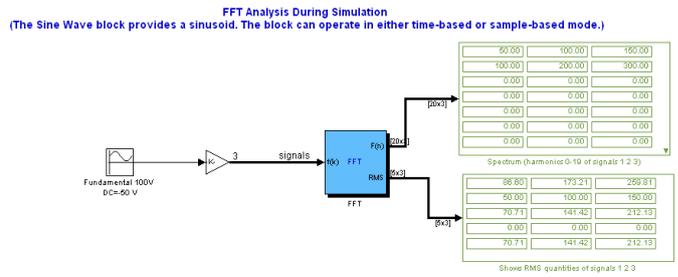


Figure 8. Simulink model for FFT analysis

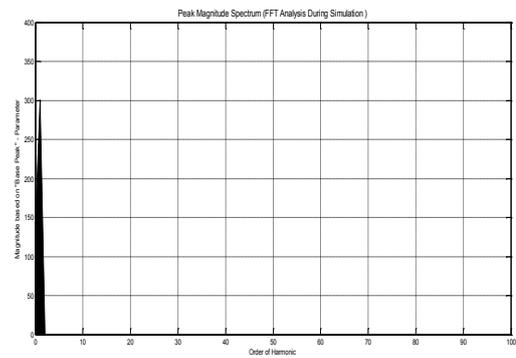


Figure 9. Frequency spectrum of sinusoidal signal

IV. CONCLUSION

It has already been found that the use of the FFT methods has greatly increased the effectiveness of digital methods for a very wide range of problems such as spectral analysis, signal processing, Fourier spectroscopy, image processing, and the solution of differential equations. Most efforts of the past several years have caused the reprogramming of previous procedures of economy in computer usage. Recent developments have been in the application of Fourier methods to problems which, due to computational effort, would not be tractable were it not for the use of the FFT method. Presently, some computers with particular purpose are being manufactured for the real-time digital Fourier methods. There are numerous areas in which a greater degree of development in future can be expected. The numerical solution of differential equations, multiple time series analysis, filtering and image processing are some the problems among these.

V. REFERENCES

- [1]. Grewal B. S., Grewal J. S., (2007), Higher Engineering Mathematics, Khanna Publishers, 40th edition, ISBN No. 81-7409-195-5.
- [2]. John G. Proakis, Dimitris K Manolakis, (2006), Digital Signal Processing (4th Edition), ISBN-13: 978-0131873742.
- [3]. Steven W. Smith, 1997, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, ISBN 0-9660176-3-3, 1997.
- [4]. Emmert, J.M., Badhri Jagannathan, Sandeep Umarani, (2003), An FFT approximation technique suitable for on-chip generation and analysis of sinusoidal signals, Defect and Fault Tolerance in VLSI Systems, 2003. Proceedings. 18th IEEE International Symposium, pp 361-368, 3-5 Nov. 2003.
- [5]. Namburi N. R., and Barton T. H., "Time Domain Response of Induction Motors with PWM Supplies", IEEE Trans. Industry Applications, vol. IA – 21, no. 2, pp. 448 – 455, Mar. 1985.
- [6]. Sample page from Numerical Recipes in FORTRAN 77: The Art of Scientific Computing (ISBN 0-521-43064-X) Copyright (C) 1986-1992 by Cambridge University Press.

Cite this article as :

K. R. Chaudhari, "Fast Fourier Transform Approach for Signal Analysis", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 2 Issue 1, pp. 179-183, January-February 2016.
Journal URL : <http://ijsrst.com/IJSRST1621110>