



Discrete Fourier Transform: Approach To Signal Processing

Anant G. Kulkarni¹, Dr. M. F. Qureshi², Dr. Manoj Jha³

Research scholar, Department of Electronics Engineering, Dr. C. V. Raman University, Bilaspur, India¹

Principal, Professor, Department of Electrical Engineering, Government Polytechnic, Kanker-Narayanpur, Buser,
India²

Head, Department of Mathematics, Rungta Engineering College, Raipur, India³

ABSTRACT: There are three different way to calculate DFT. First method is using formula of DFT or simultaneous equation. Second is Correlation technique and third one is using Fast Fourier Transform (FFT). First method is useful for understanding of basic idea of DFT, but it is not fit for practical and application purpose. In second method is based on detecting known waveform in another signal. It is used for some specified application. This method is used, when DFT has less than about 32 points. Third method is genius method, where FFT algorithm decomposes a DFT with N points, into N DFTs each with single point It is faster than DFT. These all three methods produce an identical output. Here we focused on FFT. This paper described about of DFT using FFT and its MATLAB implementation. The FFT spectrums for the outputs are analysed.

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

INTRODUCTION

The Discrete Fourier Transform (DFT), exponential function or periodic signal converted into sin and cosine functions or A - jB form. 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, where k (where k is frequency domain index for discrete signal) varies from 0 to N - 1. The relation between input samples with sine and cosine the complex DFT output signal is given by:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \left[\cos \frac{2\pi nk}{N} - j \sin \frac{2\pi nk}{N} \right]; 0 \leq k \leq N - 1$$

Figure 01 is DFT tool which convert time domain sampled data into frequency domain sampled data and vice versa. Application of DFT is in field of digital spectral analysis is Spectrum Analysers, Speech Processing, Imaging and Pattern Recognition. Actually FFT is algorithm, which implement DFT. L. G. Johnson (1992) presented Conflict free memory addressing for dedicated FFT hard ware. For high speed single chip implementation, multibank memory address assignment for an arbitrary fixed radix fast Fourier transform (FFT) algorithm is developed. The memory assignment and bank is used to minimize memory size and allow simultaneous access to all the data needed for calculation of each of the radix. Address generation for table lookup of twiddle factors is also included to have small size and high speed. He and Torkelson (1998) proposed a Fast Fourier Transform - Algorithms and Applications presents an introduction to the principles of the fast Fourier transform (FFT). It covers FFTs, frequency domain filtering, and applications to video and audio signal processing. It also has adopted modern approaches like MATLAB examples and projects for better understanding of diverse FFTs. Fast Fourier Transform - Algorithms and Applications is designed for senior undergraduate and graduate students, faculty, engineers, and scientists in the field, and self-learners to understand FFTs and directly apply them to their fields, efficiently. It provides a good reference for any engineer planning to work in this field, either in basic implementation or in research and development. Application of



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

DFT is in field of **Filter Design** calculating Impulse Response from Frequency Response and calculating Frequency Response from Impulse Response

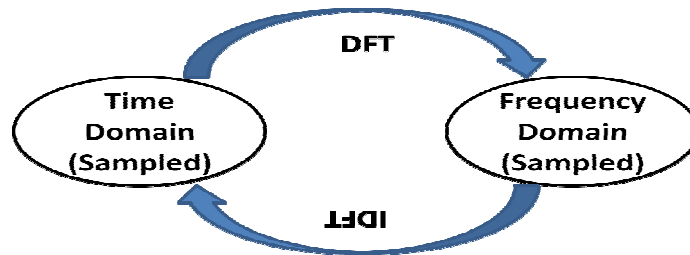


Figure 01: DFT application

II. THE FAST FOURIER TRANSFORM (FFT) IS SIMPLY AN ALGORITHM FOR EFFICIENTLY CALCULATING THE DFT

In MATLAB DFT is easily determine using `dft` command, so that DFT of sequence $x(n) = \{0, 1, 2, 3\}$ (where $N = 4$) is,

```
>> x = [0 1 2 3];
>> dft = fft(x); % The DFT
>> figure(1),stem([0 1 2 3],real(dft))
>> figure(2),stem([0 1 2 3],imag(dft))
>> ifft(dft)
>> x = [0 1 2 3]
```

x = 0 1 2 3

```
>> dft = fft(x)
```

dft = 6.0000 -2.0000 + 2.0000i -2.0000 -2.0000 - 2.0000i

Figure 02 shows the result of output of DFT in magnitude of real and imaginary signal.

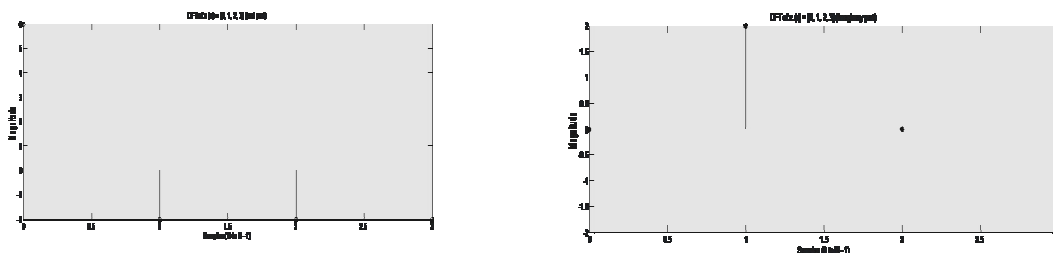


Figure 02: Magnitude of real and imaginary components.

For development of the FFT twiddle factor W_N play important role, this is given by,

$$W_N = e^{-\frac{j2\pi}{N}}$$

This leads to the definition of the *twiddle factors* as:

$$W_N^{nk} = e^{-\frac{j2\pi nk}{N}}$$

The twiddle factors are simply the sine and cosine basis functions written in polar form. Note that the 8-point DFT sixty four complex multiplications. In general, an N-point DFT requires N^2 complex multiplications. The number of multiplications required is significant because the multiplication function requires a relatively large amount of DSP



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

processing time. In fact, the total time required to compute the DFT is directly proportional to the number of multiplications plus the required amount of overhead.

There are three methods for determining DFT (John G. Proakis et al, 2006). First is analytical method (using formula).

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}, \text{ where } k = 0 \text{ to } N-1, N \text{ is length of sequence or signal.}$$

Second is, by using twiddle factor in formula. Then formula is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \text{ where } W_N^{nk} = e^{-j2\pi nk/N}.$$

and third method is matrix method, which formula is shown below,

$$[X(k)]_{N \times 1} = [W_N^{nk}]_{N \times N} [x(n)]_{N \times 1},$$

$$\text{where } [W_N^{nk}]_{N \times N} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \dots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ W_N^0 & W_N^{N-1} & W_N^{2N} & \dots & W_N^{N-1} \end{bmatrix}_{N \times N}$$

Similarly for IDFT, this is given by (John G. Proakis et al, 2006)

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi nk/N}, \quad n = 0, \dots, N-1.$$

Negative and zero indices not used in MATLAB DFT, because the starting point index in MATLAB is 1.

In the area of spectral analysis, medical imaging, filter bank, telecommunications, data compression are application of DFT.

and matrix formula for IDFT

$$[X(n)]_{N \times 1} = \frac{1}{N} [W_N^{nk}]_{N \times N}^* [X(k)]_{N \times 1},$$

DFT is operates on sampled periodic time domain signal. The signal must be periodic in order to be decomposed into the summation of sinusoids. Finite numbers of samples (N) are available for inputting into the DFT. This dilemma is overcome by placing an infinite number of groups of the same N samples “end-to-end,” thereby forcing mathematical (but not real-world) periodicity. Equation for N-point DFT is given by:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \left[\cos\left(\frac{2\pi nk}{N}\right) - j \sin\left(\frac{2\pi nk}{N}\right) \right] \dots (4.30)$$

Here X(k) is frequency domain signal, where k is frequency domain index for discrete signal. N is total length of discrete sequence. X(n) is discrete time domain signal, where n is time domain index for discrete signal. Value of k and n are from 0 to N – 1. The DFT output spectrum, X(k), is the correlation between the input time samples and N cosine and N sine waves. Figure 03 shows the concept of correlation. In this figure, the real part of the first four output frequency points is calculated, therefore, only the cosine waves are shown. A similar procedure is used with sine waves in order to calculate the imaginary part of the output spectrum. The first point, X(0), is simply the sum of the input time samples, because cos(0) = 1. The scaling factor, 1/N, is not shown, but must be present in the final result. X(0) is the average value of the time samples, or simply the DC offset. The second point, Re X(1), is obtained by multiplying

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

each time sample by each corresponding point on a cosine wave which makes one complete cycle in the interval N and summing the results. The third point, $\text{Re } X(2)$, is obtained by multiplying each time sample by each corresponding point of a cosine wave which has two complete cycles in the interval N and then summing the results. Similarly, the fourth point, $\text{Re } X(3)$, is obtained by multiplying each time sample by the corresponding point of a cosine wave which has three complete cycles in the interval N and summing the results. This process continues until all N outputs have been computed. A similar procedure is followed using sine waves in order to calculate the imaginary part of the frequency spectrum. The cosine and sine waves are referred to as *basic functions*. Correlation of time samples with basic functions using the DFT for $N = 8$ are shown below:

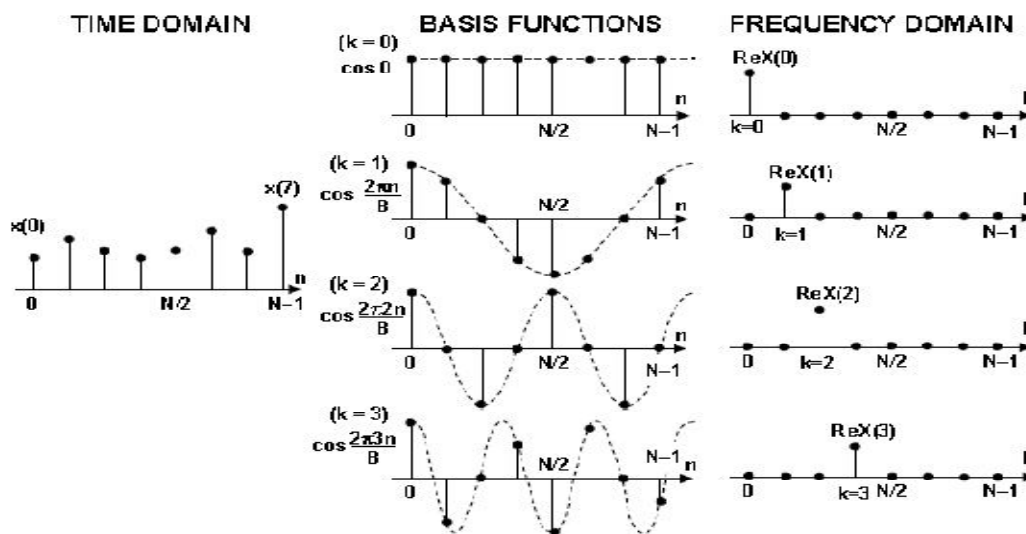


Figure 03: Correlation of time samples with basic functions using the DFT

III. THE FAST FOURIER TRANSFORM (FFT) VS. THE DISCRETE FOURIER TRANSFORM (DFT)

FFT are faster than DFT, for this reason it is called Fast Fourier Transform (FFT). FFT is algorithm which is used for calculating DFT. DFT has N^2 multiplication, where FFT has $\frac{N}{2} \log_2 N$ Multiplication.

For $N = 1024$ points DFT computations DFT takes 1,048,576 computation and for FFT it takes 10, 240 computations. The FFT is over 100 times faster. However, the number of computations given is for calculating 1024 harmonics from 1024 samples. The computational efficiency of the FFT versus the DFT becomes highly significant when the FFT point size increases to several thousand as shown in Table I.

Table I: Comparison between speed of DFT and FFT

SN	N	DFT Multiplications, N^2	FFT Multiplications, $\frac{N}{2} \log_2 N$	FFT Efficiency
1	256	65536	1024	64 : 1
2	512	262144	2304	114 : 1
3	1024	1048576	5120	205 : 1
4	2048	4194304	11264	372 : 1
5	4096	16777216	24576	683 : 1



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

IV. FREQUENCY RESPONSE OF RECTANGULAR, HAMMING, AND BLACKMAN WINDOWS FOR N = 256

In signal processing, a window function is a mathematical function that is zero-valued outside of some chosen interval. Applications of window functions are spectral analysis, filter design and non-negative smooth "bell-shaped" curves. For spectral analysis, functions $\cos \omega t$ is zero, except at frequency $\pm\omega$ is the output of Fourier transform. However, many other functions and waveforms do not have convenient closed form transforms.

In either case, the Fourier transform can be applied on one or more finite intervals of the waveform. In general, the transform is applied to the product of the waveform and a window function. Any window affects the spectral estimate computed by this method. FFT windows reduce the effects of leakage and only change the shape of the leakage. The signal is measured during a finite measurement time or 'window'. The infinite duration impulse response can be converted to a finite duration impulse response by truncating the infinite series at $n = \pm N$. But, this result in undesirable oscillations in pass band and stop band of the digital filter. This is due to the slow convergence of the Fourier series near the point of discontinuity. These undesirable oscillations can be reduced by using a set of time-limited weighting functions, $w(n)$, called as window functions, to modify the Fourier coefficients.

There are various window functions like Rectangular, Hamming, Hanning, Blackman and Kaiser Window function, which converted discontinuities in $H(e^{j\omega})$ into transition bands between values on either side of the discontinuity. Here we take simple example of window i. e. Hamming window.

In figure 4 (a) shows Hamming window. These represent the frequency response of Hamming window for high pass filter and figure 4 (b) represents the same for low pass filter. It has moderate transmission width of main lobe as compare to rectangular and Blackman window. Minimum stop band attenuation is approximately -53 dB and the peak of first lobe is approximately -43 dB.

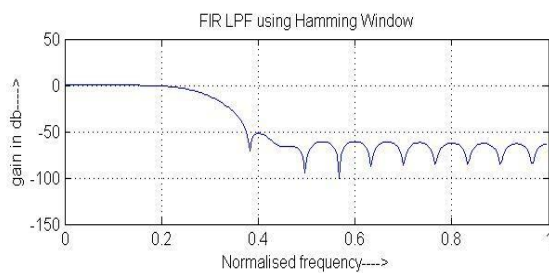


Figure 4 (a): High pass filter using Hamming window method

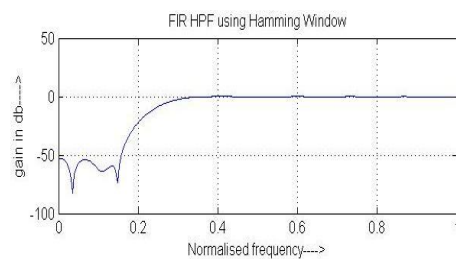


Figure 4 (b): Low pass filter using Hamming window method

Leakage presents in frequency domain due to discontinuities at the endpoints of the data window and because of the harmonics are generated. In addition to the side lobes, the main lobe of the sine wave is coated over several frequency bins. This process is similar to multiplying the input sine wave by a rectangular window pulse which has the familiar $[\sin(x) / x]$ frequency response and associated smearing and side lobes. Window function data points are usually pre-calculated. Window function data points stored in the DSP memory to minimize their impact on FFT processing time. The frequency response of the rectangular, Hamming, and Blackman windows is shown in Figure 5.

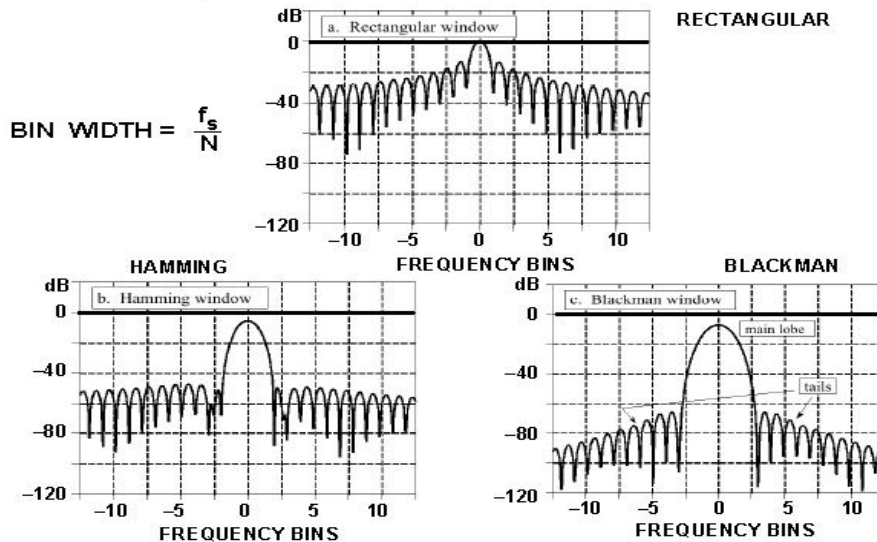


Figure 5: Frequency responses of the rectangular, Hamming, and Blackman windows.

V. FAST FOURIER TRANSFORM AND MATLAB IMPLEMENTATION

Signal is one dimensional, two dimensional or multidimensional physical variables, which vary according to time, space or any independent variables. There are two main categories of signal. First is continuous time signal and second is discrete time signal. Main source of the signal is continuous time signal. With the help of sampling we can convert continuous time signal into discrete time signal. In discrete time signal, time is in discrete form and magnitude is in continuous format. Figure 6 shows continuous and discrete time signal. Next extension digital signal is completely discrete signal; time and magnitude both are in discrete form.

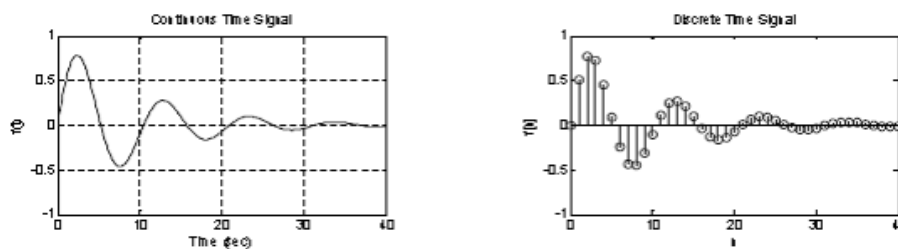


Figure 6: Continues-time signal and Discrete-time signal

Similarly periodic and non-periodic are two types of signals. Periodic signal consist fundamental period T, Which is minimum interval on which a signal repeats. Period T is reciprocal of f ($T = 1/f$). Harmonic frequency is kf or kf_0 .

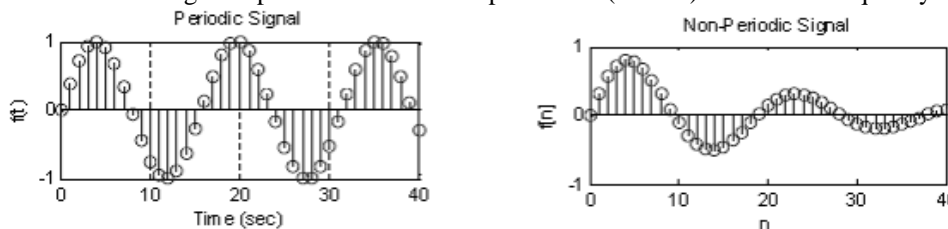


Figure 7: Discrete Periodic Signal and Non-Periodic Signal



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

Any periodic signal can be approximated by a sum of many sinusoids at harmonic frequencies of the signal (kf_0) with appropriate amplitude and phase. Sinusoidal signal consist complex exponential function which having positive and negative harmonic frequencies. Euler formula shows

$$e^{i\omega t} = \sin \omega t + j \cos \omega t$$

Using Fourier analysis, table shows the time domain signal and frequency domain signal relation:

SN	Types of signal	Frequency domain characteristics
1	Continues-time periodic signal	Discrete non-periodic spectrum
2	Continues-time non-periodic signal	Continues non-periodic spectrum
3	Discrete non-periodic signal	Continues periodic spectrum
4	Discrete periodic signal	Discrete periodic spectrum

Last two transformations are very useful because discrete is associated with both time-domain and frequency domain is because computers can only take finite discrete time signals. Using the Fourier series representation we have Discrete Fourier Transform (DFT) for finite length signal. DFT can convert time-domain discrete signal into frequency domain discrete spectrum. Assume for signal $x[n]$, where n vary from $n = 0$ to $N - 1$. Then DFT of the signal is a sequence for

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

Output signal $X[k]$ is frequency domain signal, which is completely discrete in nature due to k . k varies from 0 to $N - 1$. Motor current, voltage, speed, developed electromagnetic torque along with supply main currents are sensed and analyzed (Namburi N. R. et. al, 1985). Applng FFT analysis obtained motor current in frequency domain. Simulation model is shown in figure 8 and it result is shown in figure 9. Figure 10 shows result for faulty condition.

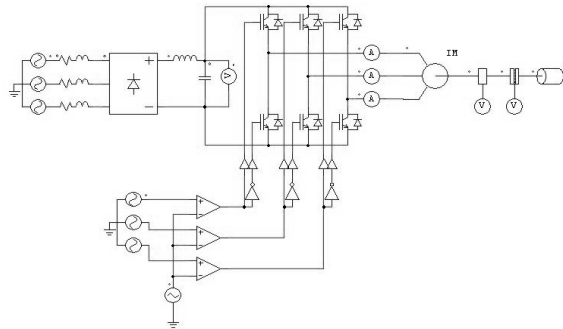


Figure 8: Simulation Model

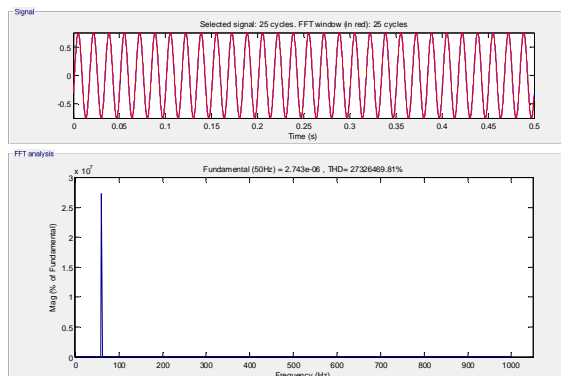


Figure 9: Magnitude vs Frequency plot of healthy induction motor for phase A.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

VI. CONCLUSION

From simulation result and their FFT analysis in figure 9 and 10, it is cleared that for line to ground fault at one of the motor phase terminals phase A having single spike with small another harmonics as compared to healthy motor. 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 DFT-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.

REFERENCES

- [1] Grewal B. S., Grewal J. S., Higher Engineering Mathematics, Khanna Publishers, 40th edition, ISBN No. 81-7409-195-5, 2007.
- [2] Johnson L.G., "Conflict free memory addressing for dedicated FFT hardware", Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions, Volume: 39, Issue: 5 pp. 312 - 316, May 1992.
- [3] John G. Proakis, Dimitris K Manolakis, "Digital Signal Processing" (4th Edition), ISBN-13: 978-0131873742, 2006.
- [4] 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.
- [5] <http://www.aip.de/groups/soe/local/numres/bookpdf/f12-2.pdf>
- [6] <http://www.dspguide.com/ch8/6.htm>
- [7] Steven W. Smith, 1997, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, ISBN 0-9660176-3-3, 1997.
- [8] <http://www.ele.uri.edu/~hansenj/projects/ele436/fft.pdf>
- [9] 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.
- [10] <http://www.cs.cmu.edu/afs/andrew/scs/cs/15-63/2001/pub/www/notes/fourier/fourier.pdf>
- [11] http://www.analog.com/static/imported-files/seminars_webcasts/MixedSignal_Sect5.pdf
- [12] <http://www.staff.vu.edu.au/msek/frequency%20analysis%20-%20fft.pdf>
- [13] 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.
- [14] Shousheng. He and Mats Torkelson, "Design and Implementation of a 1024-point Pipeline FFT Processor", IEEE Custom Integrated Circuits Conference, May. 1998, pp. 131-134.
- [15] Shousheng He and Mats Torkelson, "Designing Pipeline FFT Processor for OFDM (de)Modulation", IEEE Signals, Systems, and Electronics, Sep. 1998, pp. 257-262.
- [16] Shousheng He and Mats Torkelson, "A New Approach to Pipeline FFT Processor", IEEE Parallel Processing Symposium, April. 1996, pp.776-780.