



Science

## **SIMPLIFIED BLUESTEIN NUMERICAL FAST FOURIER TRANSFORMS ALGORITHM FOR DSP AND ASP**

**Amannah, Constance Izuchukwu <sup>1</sup>, Bakpo, Francis Sunday <sup>2</sup>**

<sup>1,2</sup> Department of Computer Science, University of Nigeria, Nsukka, Enugu State



### **ABSTRACT**

*This research was designed to develop a simplified Bluestein numerical FFT algorithm necessary for the processing of digital signals. The simplified numerical algorithm developed in this study is abbreviated with SBNADSP. The methodology adopted in this work was iterative and incremental development design. The major technology used in this work is the Bluestein numerical FFT algorithm. The study set the pace for its goal by re-indexing, decomposing, and simplifying the default Fast Fourier Transform Algorithms (the Bluestein FFT Algorithm). The improved efficiency of the Bluestein FFT algorithm is accounted for by the obvious reduction in the number of operations and operators in the simplified Bluestein algorithms. The SBNADSP is designed to have four products, and three exponentiations against the default Bluestein FFT algorithm which has six exponentiations and eight products. Since the increase in the number of operators increases the length of operation, it is therefore reasonable to infer that the algorithm with the less number of operators will run shorter execution time than the one with greater operators. In line with this, we conclude that SBNADSP is of greater efficiency than the Bluestein numerical algorithm. The result of this study showed that a faster numerical algorithm other than the Bluestein fft algorithms is possible for the processing of digital signals.*

### **Keywords:**

*Bluestein FFT, FFT, Algorithm, Simplified, Efficiency, Fast.*

**Cite This Article:** Amannah, Constance Izuchukwu, and Bakpo, Francis Sunday, “SIMPLIFIED BLUESTEIN NUMERICAL FAST FOURIER TRANSFORMS ALGORITHM FOR DSP AND ASP” International Journal of Research – Granthaalayah, Vol. 3, No. 11(2015): 153-163.

## **1. INTRODUCTION**

### **BACKGROUND TO THE STUDY**

Signals play an important role in our daily life. A signal is the variable parameter that contains information and by which information is transmitted in an electronic system or circuit. The majority of the signals found in science are analog in nature. In analog signals, both dependent variable and independent variables are continuous. Such signals may be processed directly by analog systems (i.e., analog filters) for the purpose of changing their characteristics or extracting

some desired information. Digital filters are used for two general purposes: (1) separation of signals that have been combined, and (2) restoration of signals that have been distorted in some way. Analog filters can be used for these same tasks; however, digital filters can achieve far superior results.

Digital Signal Processing (DSP) is an area of Science and Engineering which has developed very rapidly over the past few decades. As a matter of fact, the techniques and applications of Digital Signal Processing (DSP) are as old as Newton and Gauss and also as new as today's digital computers and Integrated Circuits (ICs). This rapid development of Digital Signal Processing (DSP) has been a result of the significant advances in digital computer technology and IC fabrication techniques. Signal processing is a method of extracting information from the signal which in turn, depends upon the type of signal and the nature of information it carries. Thus, signal processing is concerned with representing signals in mathematical terms and extracting the information by carrying out algorithmic operations on the signal. Mathematically, a signal can be represented in terms of basic functions in the domain of the original independent variable or it can be represented in terms of basic functions in a transformed domain. Similarly, the information contained in the signal can also be extracted either in the original domain or in the transformed domain.

Digital signal processing techniques originated in the seventeenth century when finite difference methods, numerical integration methods, and numerical interpolation methods were developed to solve physical problems involving continuous variables and functions. There has been a tremendous growth since then. Today, Digital Signal Processing (DSP) techniques are applied in almost every field. The main reasons for such wide applications are due to the numerous advantages in Digital Signal Processing (DSP) techniques. As a matter of fact, digital circuits do not depend upon precise values of the digital signals for their operation. Also, digital circuits are less sensitive to changes in component values. They are also less sensitive to the variations in temperature, ageing and other external parameters. In a digital processor, the signals and systems coefficients are represented as binary words. This enables us to choose any accuracy by increasing or decreasing the number of bits in the binary words. The storage of digital data is very easy. Signals can be stored on various storage media such as magnetic tapes, disks and optical disks without any loss. On the other hand, the stored analog signals deteriorate rapidly as time progresses and hence cannot be recovered in their original form. Also, for processing very low frequency signals like seismic signals, analog circuits require inductors and capacitors of a very large size whereas digital processing is more suited for such type of applications.

Parallel signal processing tasks Digital Signal Processors to overtake general purpose processors from 2 to 3 orders in speed. This is because of architectural differences. Typical DSP application fields are audio signal processing, video signal processing and telecommunications devices. Digital signal processing requires a large amount of real-time calculations. The most common operation in digital signal processing is the sum of products calculation. Among such operations are well known convolution and Discrete Fourier Transform. To increase the speed, digital signal processors usually have many specialized arithmetic units, which can operate simultaneously. A signal is a function of independent variables such as time, distance, position, temperature, and pressure. This research is therefore designed to provide a compatible mechanism necessary for

achieving the transition to digital processing using the simplified Bluestein Numerical Algorithm for Digital Processing (SBNADSP).

### ***STATEMENT OF THE PROBLEM***

Our contemporary global society is driving towards making all actions and activities digitalized. In Nigeria for instance the Communication and the Broadcasting Institution is determine to go all digitalized in the year 2015. The speed and scope of transmitting from analog to digital remain issues that need scientific resolution. In view of the foregoing and for effective transition from analog to digital transmission, an efficient computing algorithmic platform is a requirement. This research is therefore designed to develop a simplified numerical algorithm of the Bluestein FFT algorithm necessary for achieving the speed of processing digital signals in digital computers. The fast numerical algorithm to be developed will be called Simplified Bluestein Numerical Algorithm for Digital Processing (SBNADSP).

### ***AIM OBJECTIVES OF THE STUDY***

The aim of the study is to develop a simplified Bluestein numerical algorithm for digital signal processing. In order to attain this aim, the following objectives were considered;

- To investigate the Bluestein FFT numerical algorithm for Digital Signal Processing
- To re-index the Bluestein FFT numerical algorithm for Digital Signal Processing
- To decompose the re-indexed Bluestein FFT numerical algorithm for Digital Signal processing
- To simplify the decomposed Bluestein FFT numerical algorithm

### ***SCOPE OF THE STUDY***

There are basically two types of digital signal processing (DSP) algorithms, namely Filtering Algorithms and Signal Analysis Algorithms. These algorithms can be implemented in any of the following forms; hardware, firmware, and software. This study is restricted to the software approach of DSP-algorithm implementation on a minicomputer or a personal computer. A detailed discussion of hardware, firmware, and DSP chip implementation is beyond the scope of this study. The acronymic description of this research is Simplified Bluestein Numerical Algorithm for Digital Processing (SBNADSP).

### ***SIGNIFICANCE OF THE STUDY***

When implemented, the algorithm designed in this work will be of significant assistance to the broadcasting industry in support of their bid to transit to an inclusive digital processing. Furthermore, the Digital Signal Processing algorithms of this research will.

## 2. RELATED LITERATURE

### 2.1. THE APPLICATION OF COMPLEX DIGITAL PROCESSING IN COMMUNICATIONS

[1] Examined the application of complex digital processing in communications. They observed that the concept of DSP is coloured with the principles of mathematics. They opined that complex DSP has no obvious connection with our everyday experience. Their reason is occasioned by the fact that many DSP problems are explained mainly by means of real numbers mathematics. Their study agreed that some DSP problems are based on mathematics, such as Fast Fourier Transform (FFT), z-transform, representation of periodical signals and linear systems, and so on. Their research shows that the imaginary part of complex transformations is usually ignored or regarded as zero due to the inability to provide a readily comprehensible physical explanation. The result of their investigation shows that two DSPs exist namely complex DSP and real DSP. Their position explains that the principle of complex DSP is Complex math and that of real DSP is real math. Their approach is not universally applicable and can only be used with problems and applications which conform to the requirements of complex math techniques. Making a complex number entirely mathematically equivalent to a substantial physical problem is the real essence of complex DSP.

It is clear to see that the world has gone digital and almost being completely controlled by numbers. The concept of theoretical DSP which presupposes that DSP has no obvious connection with our everyday experience has become a mere tale. DSP can now be applied to virtually every aspect of our human endeavor. What were considered as complex DSPs occasioned by complex math can now be processed using simulate able numerical algorithmic processes such as the difference equation and the Fast Fourier Transform algorithms. In line with this the need of investigating the feasibility of fast numerical algorithms for DSP applications is both necessary and timely. This study is therefore needful and timely considering the fact that some countries are yet to integrate into the digital communication even as Nigeria is preparing to do so in the year to come.

### 2.2. THE INTERPOLATION IN DIGITAL SIGNAL PROCESSING AND NUMERICAL ANALYSIS

[2] Investigated the Interpolation in Digital Signal Processing and Numerical Analysis. He reviewed some of the methods being used in interpolation such as Lagrange, Hermite, Shannon, and Piecewise polynomials interpolators, but there are lots of other methods that he did not mention. [2] Concluded that it is impossible to give a general comparison of them and say which one is the best. It depends on the case we are dealing with he added to choose a method he further added; "I don't think there will be any guarantee that the chosen method be the best possible one. Some people have compared some of these methods in their papers". For instance, [3] says "unlike small-kernel convolution methods, which have poor accuracy anywhere near the Nyquist limit, the accuracy of the FFT method is maintained to high frequencies, very close to Nyquist". From [2], it is obvious that [2] did not effectively determine nor recommend an efficient algorithm for DSP and numerical analysis. Furthermore, the work did not also delve into detailed numerical principles compatible enough with DSP operations. [2] Dealt passively on FFT. [2] Did not also reconcile the place of DSP in relation to numerical algorithms. In view of the foregoing, we can therefore see the vacuum created by [2]; the non-determination of the place of efficient numerical algorithms

for processing digital signals. Our study is poised to fill this gap. Hence our study is consequential, timely, and pivotal.

### **2.3. EFFICIENT DIGITAL FILTERS**

In [4] investigation of Efficient Digital Filters, filters can be sharpened to meet the need of predefined DSP operations. His work shows that filters (FIR) can be sharpened either by reducing passband ripple or increasing stopband attenuation. His work proved that an existing system can still be used to perform the function of an improved new filter by simply sharpening the existing filter. This way, an old filter becomes new not by creation but by improvement. Simply stated, filter sharpening is a technique for creating a new filter from an old one. From this line knowledge it is obvious that DSP activities can be analyzed with filters and the filters can be sharpened in different ways. [5]'s work centered on software filter. He did not however introduce math-based into his work. This cannot be exhaustively concluded considering the fact that DSP is mainly the analysis and manipulation of mathematical principles in relation to signals. In such case the math platform becomes the filter which could be interlink with software filters. To enhance the effectiveness of DSP operation numerical algorithms could best be considered especially if efficiency is the focus. This is an area that is compactable with contemporary communications systems. A research work addressing this gap is vital and timely. My research is aimed at investigating efficient numerical algorithms for convenient DSP operations. Its DSP filters will be math-based and could be sharpened to address the contemporary issues of DSP activity in signal driven society.

### **2.4. DSP ON NUMERICAL RELAY FOR OVERCURRENT PROTECTION**

[6] Identified numerical relays as an area of application of DSP. The important variables in their research work are Numerical Relay and over current protection. [7] Described numerical relay as being embedded with specialized digital signal processor (DSP) as the computational hardware. As indicated in their work, over current simply means increase in current, positive or upward adjustment in current output. Their result showed that numerical relays are able to improve the performance of the protection operation considerably. [8, 9] concluded that by using DSP as the relay processor, the relay is capable of meeting the fundamental protective requirements such as reliability, sensitivity, selectivity and speed. Furthermore, the accuracy of the numerical relays is not limited to protection only but high performance relays is also enabled with advance communication, self-supervision, ability to control, metering and also event recording. In their recommendations, they explained that numerical relays provide a wide range of protection functions such as overcurrent, directional-overcurrent, under voltage, overvoltage and also other types of protection. The faults that occupy distribution systems will affect the power system reliability, security and quality.

The research above focused more on numerical relays which is a customized platform for implementing DSP. The platform is connected to a current mechanism. The numerical relays interfaced with the current mechanism such that the numerical relays act as monitor to the current distribution across the current mechanism. This research relates in some way with Faster Bluestein Numerical Algorithm for Digital Processing (SBNADSP) in that it provided an areas of application of DSP. However it differs in scope, contents, and methodology. Simplified Bluestein Numerical

Algorithm for Digital Processing (SBNADSP) seeks to apply numerical algorithms as a method in DSP. Simplified Bluestein Numerical Algorithm for Digital Processing (SBNADSP) is not platform sensitive as the method could be applied to all computing platforms without recourse to hardware-software compatibility.

## **2.5. THE MATHEMATICS OF DIGITAL SIGNAL PROCESSING**

Digital signal processing (DSP) is a very mathematical subject, so to make past articles clearer and future articles possible, the basic mathematical ideas needed are presented in [10]. Part I of the DSP mathematics provided in [10] presents a general review of algebra and trigonometry, including such areas as equations, graphs, polynomials, logarithms, complex numbers, infinite series, radian measures, and the basic trigonometric functions.

An important mathematical tool which [10] described is Fourier's theorem, which states that any periodic waveform can be described as the sum of a number, possibly an infinite number, of sinusoidal variations, each with a particular frequency, amplitude, and phase. Furthermore, there is a method for determining exactly what these frequencies, amplitudes, and phases must be in order to re-construct the waveform by adding together sine waves, which are seen to be the basic "building blocks" of periodic waveforms. Actually there are a few other requirements as well as periodicity; suffice it to be said that any waveform which could exist in the physical world will obey these other conditions (called the Dirichlet conditions) [10]. In the 19th century, the German mathematician Euler proved the following remarkable entity:  $e^{ix} = \cos x + i \sin x$  thereby relating algebraic exponentials to the trigonometric functions. This key formula is the basis for much of the mathematics used in signal processing, for it allows some very powerful manipulations to be made using sinusoidal functions that would otherwise prove very tedious.

## **3. METHODOLOGY**

The Fast Fourier Transform (FFT) is surely the most widely used signal processing algorithm. It is the basic building block for a large percentage of algorithms in current usage. An FFT is a way to compute the same result more quickly. This work investigated the Bluestein FFT algorithm. The methodology adopted in this work was iterative and incremental development design. This is also known as iterative and incremental build model for software development. The basic idea behind this method is to develop a system ( algorithm) through repeated cycles (iterates) and in smaller portions at a time ( incremental), allowing the software developer (researcher) to take advantage of what was learned during development of earlier parts or versions of the system. Incremental development means that different parts of a software project are continuously integrated into the whole, instead of a monolithic approach where all the different parts are assembled in one or a few milestones of the project. So as a first result, iterative development doesn't need to be incremental and vice versa, but these methods are a good fit.

Considering the fact that the Fast Fourier Transform algorithm already exist, and also that our work is aimed at developing a faster one, we can see that our work has a relationship with available system. For this reason our methodology is step down to iterative reconstruction methodology. This method has four basic components beginning with requirements and climaxing with

implementation and test. Fig.1 below illustrates these components of reconstructive iterative methodology.

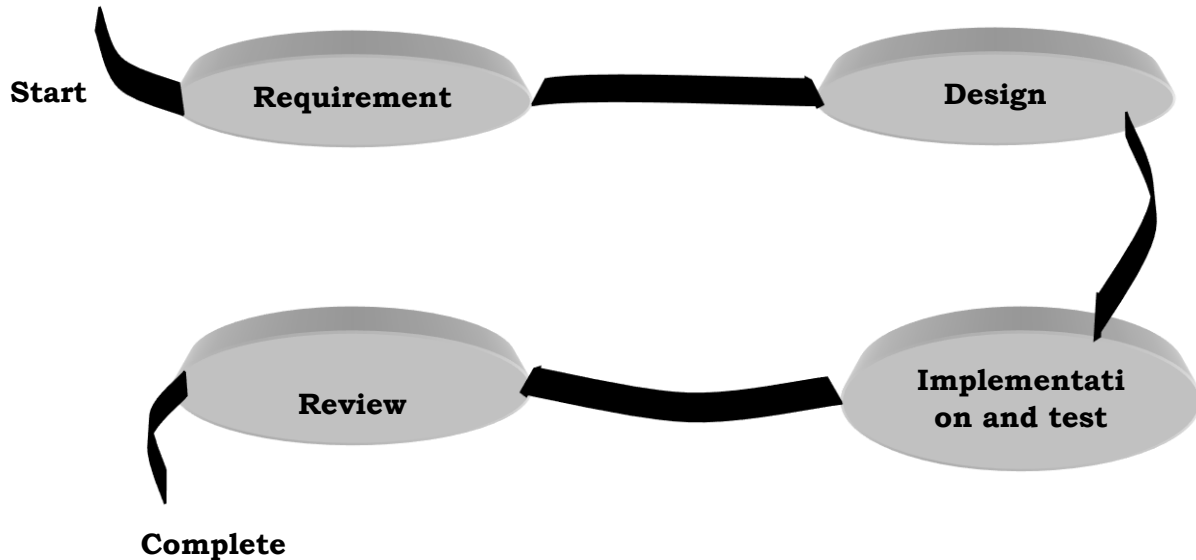


Figure1: Reconstructive Iterative Development Model

### 3.1. AN OVERVIEW OF THE EXISTING SYSTEM

The existing system is generally referred to as Bluestein Fast Fourier Transforms (FFTs) algorithm. An FFT is any method devised to compute the same results in  $O(N \log N)$  operations. All known FFT algorithms require  $O(N \log N)$  operations. There are species of FFTs. In this work, we investigated the Bluestein FFTs.

#### The Bluestein [11] FFT algorithm

The Bluestein FFTs algorithm is traced and transformed from the Discrete Fourier Transform (DFT). The DFT is of two phases; the forward and inverse phases. Each of the four-FFTs is described below beginning with the DFT.

#### Discrete Fourier Transform (DFT)

##### DFT Formula

Let  $x_0, \dots, x_{N-1}$  be complex numbers. The DFT is defined by the formula:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j 2\pi k \frac{n}{N}}, \quad k=0 \dots N-1 \quad \dots (1)$$

Evaluating this definition directly requires  $O(N^2)$  operations.

#### Inverse Discrete Fourier Transform (IDFT)

The inverse transform of DFT:

$$\text{DFT: } F^{-1}[n] = \sum_{k=0}^{N-1} f[k] e^{j \frac{2\pi}{N} nk}$$

The required inverse DFT becomes

$$F[k] = \frac{1}{N} \sum_{n=0}^{N-1} F[n] e^{+j \frac{2\pi}{N} nk} \dots (2)$$

From the inverse transform formula, the contribution to  $f[k]$  of  $F[n]$  and  $F[N - n]$  is

$$f_n[k] = \frac{1}{N} \left[ F[n] e^{j \frac{2\pi}{N} nk} + F[N - n] e^{j \frac{2\pi}{N} (N - n)k} \right]$$

For all  $f[k]$  real,  $F[N - n] = \sum_{k=0}^{N-1} f[k] e^{-j \frac{2\pi}{N} (N - n)k}$

But  $e^{-j \frac{2\pi}{N} (N - n)k} = \underbrace{e^{-j 2\pi k}}_{1 \text{ for all } k} e + j \frac{2\pi n}{N} k = e + j \frac{2\pi}{N} nk$

i.e.  $F[N - n] = F^*(n)$  (i.e. the complex conjugate)

Comparing equations (1) and (2) we infer that the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a  $\frac{1}{N}$  factor. Any fast Fourier Transform (FFT) algorithm can easily be adapted from it.

Bluestein’s FFT algorithm also known as chirp Z-transform algorithm can be used to compute prime length DFTs in  $O(N \log N)$  operations. However, it is not restricted to prime lengths, and it can compute other kinds of transform. The Bluestein’s FFT algorithm can be expressed from DFT as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W^{-kn}, \quad k = 0, 1, 2, \dots, N - 1$$

Multiplying and dividing by  $W^{(n^2 + k^2)/2}$  yields

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W^{-kn} W^{1/2 (n^2 + k^2)} W^{-1/2 (n^2 + k^2)} \\ &= W^{-1/2 k^2} \sum_{n=0}^{N-1} [x(n) W^{-1/2 n^2}] W^{1/2 (k - n)^2} \\ &= W^{-1/2 k^2} (x_q * W_q) \dots 3 \end{aligned}$$

**Eqn (3) is called the Bluestein’s FFT algorithm**

Where ‘\*’ denotes convolution and  $x_q$  and  $w_q$  are defined by

$$\begin{aligned} x_q(n) &\triangleq x(n) W^{1/2 n^2}, \quad n = 0, 1, 2, \dots, N - 1 \\ w_q(n) &\triangleq W^{1/2 n^2}, \quad n = -N + 1, -N + 2, \dots, -1, 0, 1, 2, \dots, N - 1 \\ W &\triangleq \exp(j 2\pi / N) \end{aligned}$$

The Bluestein’s FFT algorithm provides  $N \log N$  complexity for any positive integer DFT – length  $N$  whatsoever, even when  $N$  is prime. The sequence  $x_q$  above consists of the original data sequence  $x(n)$  multiplied by a signal  $\exp(j\pi^2/N)$  what can interpreted as a sampled complex sinusoid with instantaneous radian  $2\pi n/N$  frequency. Such signals are called chirp signals. For this reason, Bluestein algorithm is also the chirp-z transform algorithm.



### 3.2. THE PROPOSED SYSTEM

The proposed system is abbreviated as SBNADSP, which means Simplified Algorithm for Digital Signal process. The SBNADSP is a numerical-based algorithm for computing discrete sequence digital signals. The proposed system works with digital signal inputs.

The efficiency of the FFT algorithms is already determined. Evaluating the DFT's sums directly involves  $N^2$  complex multiplications and  $N(N-1)$  complex additions of which  $O(N)$  operations which can be saved by eliminating trivial operations such as multiplications by one.

#### 3.2.1. DESIGN OF THE PROPOSED SYSTEM

The proposed algorithm results from further decomposition, re-indexing and simplification of the Bluestein Fast Fourier Transform algorithm. The procedure is outlined below:

##### The DFT Algorithm

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi k n / N} \dots (4)$$

Where  $i2\pi / N$  = twiddle factor of unity. The DFT has some relationship with the Bluestein FFT algorithm.

##### The Bluestein FFT Algorithm

$$X_{(K)} = W^{-1/2} K^2 (xq * Wq)_K \dots (5)$$

$$Wq(n) \triangleq W^{-1/2} n^2, \quad n = -N + 1, -N + 2, \dots -1, 0, 1, 2, \dots N - 1 \dots (6a)$$

$$W \triangleq \exp(j2\pi / N) \dots (6b)$$

$$Xq(n) \triangleq x(n) N^{-1/2} n^2, \quad n = 0, 1, 2, \dots N - 1 \dots (6c)$$

Eqn (5) can be re-indexed by substituting for  $X_q$  and  $W_q$

$$X(K) = W^{-1/2} K^2 [(x(n) N^{-1/2} n^2) * W^{1/2} n^2]_k \dots (7)$$

Collecting like term we have;

$$\begin{aligned} X(K) &= W^{-1/2} * W^{1/2} K^2 [(x(n) N^{-1/2} n^2) * W^{1/2} n^2]_k \\ &= W^{-1/2} * W^{1/2} (K^2 * K)(n^2 * n^2) [x(n) N^{-1/2}] \\ &= W^{-1/2 + 1/2} (K^{2+1})(n^{2+2}) [x(n) N^{-1/2}] \\ &= W^0 (K^3)(n^4) [x(n) N^{-1/2}] \end{aligned} \dots (7a)$$

$$= K^3 n^4 [x(n) N^{-1/2}] \dots (8)$$

$$K = -N + 1, -N + 2 \dots -1, 0, 1, 2, \dots N - 1$$

$$n = 0, 1, 2 \dots N - 1$$

$x(n)$  is defined as;

$$x(n) = kn = K^2 / 2 + n^2 / 2 - [(k - n)^2 / 2]$$

Eqn (8) is the product of re-indexing and simplification. We can further decompose eqn. (8) by referencing the relationship between the DFT and the Bluestein FFT algorithm. The twindle factor of unity  $i2\pi/N$  exists in DFT in eqn (4) but as  $W$  in the Bluestein FFT algorithm in eqn (7).

Since  $W$  is defined as:

$$W \triangleq \exp(j2\pi/N), \text{ it therefore holds that the twindle factor also exist in } W. \text{ Thus, we}$$

can express the relationship as:

$$W = i2\pi/N = j2\pi/N$$

Substituting  $W = \exp(j2\pi/N)$  into eqn. (7a) We have;

$$X(k) = \exp(j2\pi/N) K^3 n^4 [x(n) N^{-1/2}] \dots (9)$$

Since the twindle factor is unity (equal to 1), a substitution of the unity factor into eqn (9) yields;

$$X(k) = \frac{1}{N} k^3 n^4 [x(n) N^{-1/2}] \dots (10)$$

$$= N^{-1} k^3 n^4 [x(n) N^{-1/2}]$$

$$= N^{-1} * n^{-1/2} k^3 n^4 [x(n)] \dots (11)$$

$$= N^{-1-1/2} k^3 n^4 [x(n)]$$

$$= N^{-3/2} k^3 n^4 [x(n)] \dots (12)$$

Eqn (12) is the resulting algorithm with three exponentiations and four products. This is a simplified or minimized Bluestein FFT algorithm which has six exponentiations and eight products. Such reduction in the number of operators can account for speed of the generated algorithm.

#### 4. DISCUSSION OF RESULTS

The result of this study shows that we can have more efficient algorithm than the Bluestein numerical algorithm for the processing of digital signals. The faster algorithm resulted from the re-indexing and modification of the Bluestein FFT algorithm. The authors by this result therefore succeeded in developing an algorithm that is faster than the FFT algorithms. The improved efficiency of the Bluestein FFT algorithm is accounted for by the obvious reduction in the number of operations and operators in the simplified Bluestein algorithms. The SBTNADSP is designed to have four products, and three exponentiations against the default Bluestein FFT algorithm which has six exponentiations and eight products. Since the increase in the number of operators increases the length of operation, it is therefore reasonable to infer that the algorithm with the less number of operators will run shorter execution time than the one with greater operators. In line with this, we conclude that SBNADSP is of greater efficiency than the Bluestein numerical algorithm.

#### 5. SUMMARY AND CONCLUSION

##### 5.1. SUMMARY

This study was aimed at designing as amplified Bluestein numerical algorithm for digital signal processing. This aim was actually achieved. The study had as it basis the Bluestein numerical algorithm (BTNA). This Algorithm was noted for a computing speed of  $O(N\log N)$ . This obvious

constant speed provided a basis for the simplification of the BNA in search of a more efficient algorithm.

## 5.2. CONCLUSION

In-depth analysis that proceeded the development of this study revealed that the DSP algorithms are predicated on the discrete Fourier transform (DFT) of  $O(N^2)$  computing speed. The DFT gave rise to the fast Fourier transform (FFT) algorithms of  $O(n \log n)$  computing speed. This study exhaustively investigated the BNA. This study therefore infers that the SBNADSP is a more efficient numerical algorithm with less number of operators thereby leading to shorter execution time.

## 6. REFERENCES

- [1] Vladimir, P. and Zlatka, N., Georgi, I., Miglen, O., (2011). *Complex Digital Signal Processing in Telecommunications: Applications of Digital Signal Processing*, Dr. Christian Cuadrado-Laborde (Ed.), 307-406.
- [2] Saeed, B. (2003). *Interpolation in Digital Signal Processing and Numerical Analysis*. New York: Springer-Verlag.
- [3] Fraser, D. (1989). *Interpolation by the FFT Revisited An Experimental Investigation*, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, (37)5, pp. 665-675.
- [4] Matthew, P.D. (2000). *Efficient Digital Filters*, *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-
- [5] Hamming, R. and Kaiser, J. (1977). *Sharpening the Response of a Symmetric Non recursive Filter by Multiple Use of the Same Filter*, " *IEEE Trans. Acoustics, Speech, Signal Proc*, ASSP, (25) 5, pp. 415 – 422 .
- [6] Yin, L. G., Agiieswari, K. R., (2005). *Evaluation of DSP Based Numerical Relay for Over Current Protection Centre for Communication Service Convergence Technologies (CCSCT)*. Department of Electronics and Communication Engineering, Beijing China.
- [7] Areva, T.D. (1995). *Network Protection and Automation Guide*. Prentice Hall. New Jersey.
- [8] Khederzadch, M. (2006). *Back-up protection of distance relay second zone by directional overcurrent relays with combined curves*, in *IEEE Power Engineering Society General Meeting*.
- [9] Price, E. (2010). *The next step in the evolution of protection and control implementation*, in *Annual conference for Protective relay Engineers, Moscow, Russia*, pp 1-16.
- [10] *An Introduction to the Mathematics of Digital Signal Processing: Part I: Algebra, Trigonometry, and the Most Beautiful Formula in Mathematics* Author(s): F. R. Moore Source: *Computer Music Journal*, Vol. 2, No. 1, (Jul., 1978), pp. 38-47 Published by: The MIT Press Stable
- [11] Pavan Kumar K.M., Priya Jain, Ravi Kiran S, Rohith N., Ramamani K. *FFT Algorithm: A Survey*. *The International Journal of Engineering and Science (IJES)* Volume 2 Issue 4 pages 22-26, 2013 ISSN(e): 2319-1813 ISSN(o):