Streamlined real-factor FFTs

Mohammed Zafar Ali Khan
ICSL
IIT, Hyderabad-502205,India
Email: zafar@iith.ac.in

Shaik Qadeer
MJ College of Engg.
and Tech., EED,Hyd,India
Email: haqbei@gmail.com

***Abstract*—In this paper we show that Rader and Brenner's 'real-factor' FFT can be streamlined so that it requires lower computational complexity as compared to the Cooley- Tukey radix-2 FFT. We then show that the fixed point implementation of 'real-factor' FFT can be modified so that its noise-to-signal ratio (NSR) is lower than the NSR of Cooley-Tukey radix-2 FFT. Finally simulation results are presented which verify the suitability of 'real-factor' FFTs.**

***Index Terms*—DFT, FFT, real-factor FFT**

I. Introduction

Discrete fourier transform (DFT) is among the most fundamental operations in digital signal processing. However, the wide use of DFT makes its computational requirements an important issue. Direct computation of DFT requires on the order of N^2 operations where N is the transform size. The breakthrough of the Cooley-Tukey FFT [1] comes from the fact that it brings the complexity down to an order of $N \log_2 N$ operations.

Among the numerous further developments that followed Cooley and Tukey's original contribution are the Winograd fourier transform algorithm (WFTA) [2] and the 'real-factor' algorithms [3], [4] for reduction in the order of the multiplicative complexity. However both WFTA and 'real-factor' FFTs did not meet expectations once implemented as the number of additions (and data transfers) also matter in the implementation. In addition Rader and Brenner's 'real-factor' FFT is ill-conditioned i.e., "small computational errors lead to large output errors" [3] due to the large values that the twiddle factors can take. Note that the 'real-factor' FFT of Cho and Temes [4] is numerically well conditioned.

In this paper we present solutions for both the problems of Rader and Brenner's 'real-factor' FFT. We first show that the arithmetic complexity (multiplications and additions, also known as the flop count) of the Rader and Brenner 'real-factor' FFT can be reduced to about $4\frac{1}{2}N\log_2(N)$ which is less than the arithmetic complexity of radix-2 Cooley-Tukey FFT. We then show that fixed-point implementation of Rader and Brenner 'real-factor' FFT has lower NSR then radix-2 Cooley-Tukey FFT and as a result is more suitable as it requires only half the number of real multiplications. Additionally, the modified 'real factor' FFT is not ill-conditioned as the magnitude of all the twiddle factors is less than 1.

The essence of Radar and Brenner's DIT FFT is as follows: Let $\{A_k\}$ denotes N-point DFT of the sequence a_n of $N = 2^M$ i.e. $\{A_k\} = DFT_N\{a_n\}$. Then radix-2 DIT FFT is given by

$$A_k = B_k + W_N^k D_k, k = 0, \cdots, (N-1) \qquad (1)$$

where $W_N^k = \exp(-j2\pi/N)$ and the $N/2$-point sequences are defined as $\{b_n\} = \{a_{2n}\}_{n=0}^{N/2-1}$,$\{d_n\} = \{a_{2n+1}\}_{n=0}^{N/2-1}$, $\{B_k\} = DFT_{N/2}\{b_n\}$ and $\{D_k\} = DFT_{N/2}\{d_n\}$. Note that for a $N/2$-point sequence, if the index k, n is greater than $N/2-1$ we assume $k, n \mod \frac{N}{2}$. Since in general D_K and W_N^k are both complex in nature, the basic butterfly operation indicated in (1) requires '4' real multiplications and '4' real additions. Rader and Brenner introduced the following sequence [3]

$$c_n = d_n - d_{n-1} + q_n, n = 0, \cdots, \frac{N}{2} - 1 \qquad (2)$$

where $q_n = 2/N \sum_{m=0}^{\frac{N}{2}-1} d_m$ It follows that (1) can be rewritten as

$$A_0 = B_0 + C_0, A_{N/2} = B_0 - C_0,$$

$$A_k = B_k - \frac{j}{2} csc\left(\frac{2\pi k}{N}\right) C_k, k \neq 0, N/2-1 \qquad (3)$$

where $C_k = DFT_{N/2}\{c_n\}$ and where $\csc(x)$ is the cosecant of x. If we change the sign to plus in the definition of c_n, then the coefficient of C_k in (3) is $\frac{1}{2}sec(\frac{2\pi k}{N})$ where $sec(x)$ is the secant of x . In either

 ISSN 2076-1465

case, equation (3) requires 2 real multiplications and 2 real additions. But this is at the expense of more additions to find q_n and c_n from d_n. Therefore total real additions per stage will be $2N + N + 2N = 5N$. The real multiplications are given by N and the total flop count is given by $\sim 6N \log_2 N$. The rest of the paper is organized as follows: In section II we first present DIF version of new FFT that requires $\sim 5N \log_2 N$ flops in subsection II-A. In subsection II-B we derive the corresponding DIT FFT and in subsection II-C we show that the flop count can be further reduced to $\sim 4\frac{1}{2} N \log_2 N$. Notably this is the best known flop count that can be achieved by a radix-2 FFT. We then present a fixed-point implementation of Rader and Brenner's FFT based on the radix-2 FFTs developed in this paper which has a lower NSR than the radix-2 Cooley-Tukey FFT in Section III and conclusions are given in Section IV.

II. New FFT

In this section, we first derive a DIF FFT that requires $\sim 5N \log_2 N$ flops. The corresponding DIT version is also derived and the algorithm is further modified so that the flop count is further reduced to $\sim 4\frac{1}{2} N \log_2 N$.

A. Derivation of DIF version of New FFT

According to the principle of DIF-FFT, the even and odd DFT coefficients are given by

$$A_{2k} = DFT_{\frac{N}{2}}\{a_n + a_{n+N/2}\}, k = 0, \cdots, \frac{N}{2} - 1 \quad (4)$$

$$A_{2k+1} = DFT_{\frac{N}{2}}\{(a_n - a_{n+N/2})W_N^n\}, k = 0, \cdots, \frac{N}{2} - 1 \quad (5)$$

To eliminate the complex multiplication factor from (5), we define the sequence $\{c_n\}$ as

$$c_n[1 - W_N^{2n} + \delta(n)] = \{a_n - a_{n+N/2}\}W_N^n, n = 0, \cdots, \frac{N}{2} - 1. \quad (6)$$

Equation (6) can be simplified as

$$c_0 = \{a_0 - a_{N/2}\} \quad (7)$$

$$c_n = \frac{j}{2} \csc\left(\frac{2\pi n}{N}\right)\{a_{n+N/2} - a_n\}, n = 1, \cdots, \frac{N}{2} - 1. \quad (8)$$

Applying DFT on both sides of (6) we get

$$C_k - C_{k+1} + c_0 = A_{2k+1}, k = 0, \cdots, \frac{N}{2} - 1. \quad (9)$$

where

$$C_k = DFT_{N/2}\{c_n\}$$

and

$$DFT_{N/2}\{c_n\delta(n)\} = c_0.$$

Denote $P(n, N) \triangleq -\frac{j}{2}\csc(\frac{2\pi n}{N})$, then the DIF computation for the new algorithm is schematically represented for $N = 16$ in Figure.1.

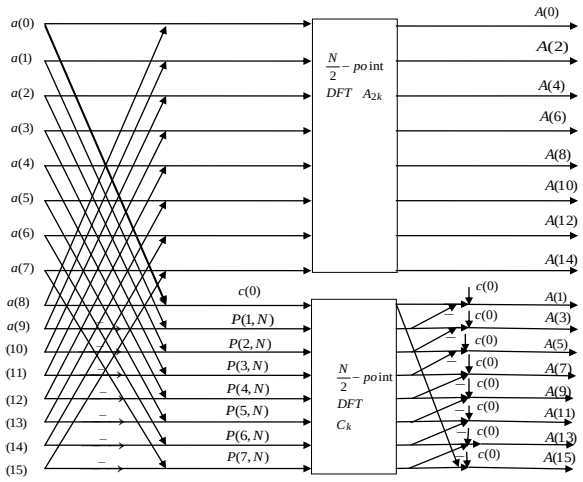

Fig. 1. The decomposition of the N-point DFT into two $N/2$-point DFTs for new DIF radix-2 FFT algorithm for $N = 16$

1) Computation complexity for Real Multiplications: The number of multiplications can be calculated from (8). For each value of 'n', '2' real multiplications are needed and the total number of real multiplications given two $N/2$ points DFTs is $N - 2$. Therefore the total number of real multiplications for a N-point DFT, $M(N)$ is given by

$$M(N) = N - 2 + 2M\left(\frac{N}{2}\right). \quad (10)$$

Solving by repeated substitutions we have,

$$M(N) = N \log_2(N) - 3N + 2. \quad (11)$$

2) Computation complexity for Real Additions: The number of real additions, given two $N/2$ point DFTs, is are calculated as follows:

- $2(N/2)$ real additions for evaluating (4) for $n = 0, \cdots, \frac{N}{2} - 1$,
- $2(N/2+1)$ real additions for evaluating (6) for $n = 0, \cdots, \frac{N}{2} - 1$,
- $4(N/2)$ real additions for evaluating (9) for $n = 0, \cdots, \frac{N}{2} - 1$.

Let $A(N)$ denote the total number of real additions for computing a N point DFT. It follows that

$$A(N) = 4N + 2 + 2A\left(\frac{N}{2}\right). \quad (12)$$

Solving by repeated substitutions we have,

$$A(N) = 4N \log_2(N) - 2N - 2. \quad (13)$$

The total flop count is then given by $T(N) = 5N \log_2(N) - 5N$. Importantly, the DIF version of the

FFT presented in Subsection II-A require the same number of computations (real adds + real multiplies) as normal radix-2 DIT or DIF FFT while requiring lesser number of multiplies. Also as compared to DIT version of Rader and Brenner's FFT the DIF version requires N fewer real additions.

B. DIT version of New FFT

In this subsection we derive dual DIT version of the DIF FFT presented in subsection II-A. The dual DIT FFT for the DIF FFT of Section II is given by:

$$c_n = d_n - d_{n-1}, n = 0, \cdots, \frac{N}{2} - 1. \tag{14}$$

It follows that (3) is still valid with the exception that C_0, is defined as

$$C_0 = \sum_{n=0}^{\frac{N}{2}-1} c_n \tag{15}$$

The arithmetic complexity of this DIT FFT is again $T(N) = 5N\log_2(N) - 5N$. Note that the difference between Rader and Brenner's FFT and our FFT is in the computation of C_0. In Rader and Brenner's FFT $\frac{1}{N}C_0 = q$ is added to all c_i which requires $2N-2$ real additions while in our version C_0 is directly computed as in (15) which requires $N-2$ real additions.

C. Modified DIF version of New FFT

In this subsection we show that the computational complexity can be reduced further. If we add c_0 to (8) for $n = \frac{N}{4}$ then

$$c_{\frac{N}{4}} = -\frac{j}{2}csc(\frac{\pi}{2})\{a_{\frac{N}{4}} - a_{3N/4}\} + c_0, \tag{16}$$

and rest of the sequence $\{c_n\}, n = 0, \cdots, N/4 - 1, N/4+1, \cdots, N/2-1$ is given by (7)-(8). Equation(9), for this case, modifies to

$$A_{2k+1} = \begin{cases} C_k - C_{k+1} + 2c_0 & k \text{ is even} \\ C_k - C_{k+1} & \text{otherwise} \end{cases} \tag{17}$$

The DIF computation for the new algorithm is schematically represented for $N = 16$ in Figure 2. With this substitution it is obvious that the number of real multiplications are same. However, $\frac{N}{2}-2$ fewer real additions are required compared to the previous subsections i.e. II-A, II-B. Accordingly, the number of real additions are given by

$$A(N) = \frac{7}{2}N + 4 + 2A\left(\frac{N}{2}\right). \tag{18}$$

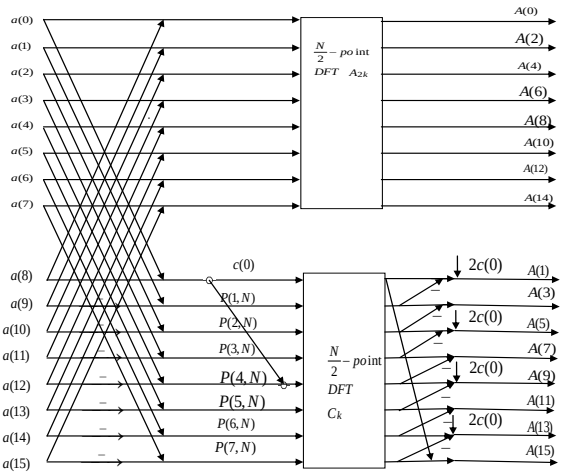

Fig. 2. The decomposition of the N-point DFT into two $N/2$-point DFTs for new DIF radix-2 FFT for $N = 16$.

Solving by repeated substitution we have,

$$A(N) = \frac{7}{2}N\log_2(N) - \frac{1}{2}N - 4. \tag{19}$$

The total flop count is then given by $T(N) = 4\frac{1}{2}N\log_2(N) - \frac{7}{2}N - 2$. Importantly, the flop count of this radix-2 FFT is less than other radix-2 FFTs including standard DIT, DIF radix-2 FFTs. Note that the DIT version for this modification can be similarly obtained as in subsection II-A.

III. Fixed-point analysis of New FFT

In this section we consider the fixed-point implementation of 'real factor' FFTs presented in section II. An important parameter in the analysis of fixed-point implementation is the noise-to-signal ratio (NSR) [16], [17]. Note that the NSR of Cooley-Tukey FFT is $4N2^{-2b}$, where b is the bit precision[16], [17]. The NSR of Rader and Brenner's algorithm is known to be very poor because of the large values of $\csc\left(\frac{2\pi n}{N}\right)$ [3]. Accordingly we first modifiy the real-factor FFT for fixed-point implementation in Subsection III-A and then derive their NSR in Subsection III-B. Verification of the desired NSR is then presented in Subsection III-C

A. A Real-factor FFT for fixed point implementation

To start with we modify (7) as

$$\hat{c}_0 = \frac{c_0}{2^l} = \frac{\{a_0 - a_{N/2}\}}{2^l} \tag{20}$$

$$\hat{c}_n = \frac{c_n}{2^l} = \left[\frac{j\csc\left(\frac{2\pi n}{N}\right)}{2^{l+1}}\right]\left\{(a_{n+N/2} - a_n)\right\}, \quad n = 1, \cdots, \frac{N}{2} - 1. \tag{21}$$

where l is chosen such that $\csc\left(\frac{2\pi}{N}\right) \leq 2^{l+1}$. It follows that $\frac{\csc\left(\frac{2\pi n}{N}\right)}{2^{l+1}} \leq 1, \forall n$. This modification ensures that the real multiplication factors are less than 1. *The resulting algorithm is not ill-conditioned as the magnitude of all the twiddle factors is less than 1*. Applying DFT on both sides of (6) (with this modification) we get

$$\left\{\hat{C}_k - \hat{C}_{k+1} + \hat{c}_0\right\} 2^l = A_{2k+1}, k = 0, \cdots, \frac{N}{2} - 1. \tag{22}$$

where

$$\hat{C}_k = DFT_{N/2}\{\hat{c}_n\}$$

and

$$DFT_{N/2}\{c_n\delta(n)\} = c_0.$$

The purely imaginary twidle factor is now given by $P(n,N) \triangleq -\frac{j}{2^{l+1}}\csc(\frac{2\pi n}{N})$. A similar algorithm with purely real twidle factors can be similarly developed. Note that in both the cases $0 \leq P(n,N) \leq 1$. The advantage of the presented fixed-point algorithm can be observed from (22) where in the effect of large values of $\csc(\theta)$ has been captured in l and the effective precision at the output is $l+b$ bits. In the next section we show by deriving the NSR that this algorithm has effectively converted the disadvantage of large twidle factors into an advantage.

B. NSR of modified real-factor FFT

Following the procedure of calculating the NSR of Cooley-Tukey [16], [17], the main source of noise in a DFT or FFT calculation is the quantization noise which is mainly required due to finite register lengths and is associated with each real multiplication in the DFT/FFT computation. Note that better and more accurate noise models have been considered in literature [5]-[15], however the essence of these models is that other factors are negligible.

Following [16], [17], the input signal is assumed to be such that successive sequence values are uncorrelated i.e. its a white signal with uniform amplitude distribution of real and imaginary parts in the interval $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$. For convenience, it is further assumed that a_n is of infinite precision, thus yielding the average signal power of a_n as

$$E\{a_n\} = \frac{1}{3}. \tag{23}$$

In fixed point arithmetic overflow has to be prevented by scaling. If an overall scaling factor of $1/s$ is used, then using Parseval's theorem and the DFT equation, the output signal power is given by [16], [17]

$$E\{|As_k|^2\} = \frac{N}{3s^2}. \tag{24}$$

Now to avoid overflow in (20) and (21), the input a_n is scaled by $1/2$ at each stage. As the number of stages is $v = \log_2 N$, the total scaling is $1/N$ and the output signal power is given by

$$E\{|As_k|^2\} = \frac{1}{3N}. \tag{25}$$

Note that there is no need of scaling in (22) as will be discussed later. Similarly, the output noise power can be estimated as follows; major part of the overall noise in FFT computation will arise due to finite world length effects which requires rounding off the results obtained after multiplications. Now one butterfly of 'real-factor' FFT algorithm requires one complex-real multiplication which in turn contains two real multiplications. Moreover, each real multiplication contributes a roundoff error $\varepsilon(n,k)$. To compute the variance of error in A'_k, where $A'_k = A_k + \varepsilon(n,k)$ we assume that:

1) The errors are uniformly distributed random variables over the range $(-1/2)2^{-b} to (1/2)2^{-b}$. Therefore, each error source has variance $\sigma_b^2 = \frac{2^{-2b}}{12}$.
2) The errors are uncorrelated with each other.
3) All the errors are uncorrelated with input and consequently also with the output.

Overall noise variance at each (real-factor) butterfly can then be written as,

$$\sigma_B^2 = 2\sigma_b^2 = \frac{2^{-2b}}{6}. \tag{26}$$

A more generalized model is given in [13] as $\sigma_b^2 = \beta\frac{2^{-2b}}{12}$ where $3 \leq \beta \leq 4$. However, for simplicity, we assume the model of (26). Now the mean-square magnitude of the noise at each output node is the sum of the contributions of each noise source (one per required butterfly) while computing the output of that node. As the scaling is assumed to be stage by stage, the noise introduced at previous stages will be also be scaled along with signal. Thus, noise sources introduced at different stages in FFT will contribute different amount of noise at the output.In a N-point FFT with $v = \log_2(N)$ stages, noise source originating at m^{th} array will propagate to the output with multiplication by $\left(\frac{1}{2}\right)^{v-m-1}$. Also, each output node connects to 2^{v-m-1} butterflies and therefore to 2^{v-m} noise sources that originates at m^{th} array. Thus at each output node, the mean-square magnitude of the noise is,

$$\begin{aligned} E\{|An_k|^2\} &= \sigma_B^2 \sum_{m=0}^{v-1} 2^{v-m}\left(\frac{1}{2}\right)^{2(v-m-1)} \\ &= 4\sigma_B^2\left(1 - \frac{1}{2^v}\right) \end{aligned} \tag{27}$$

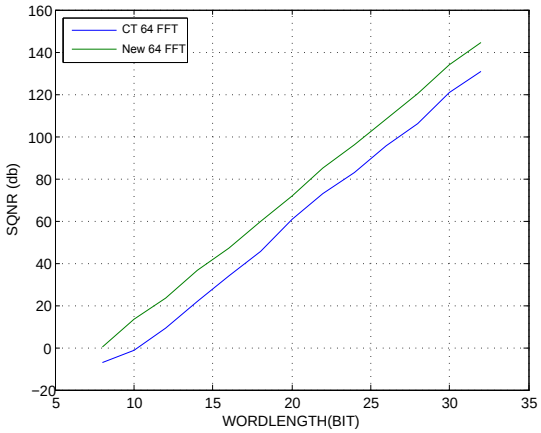

Fig. 3. SQNR for Cooley-Tukey (CT) FFT and the modified 'real-factor' (New) FFT for $N = 64$ and different wordlengths.

For large N, $E\{|An_k|^2\} \simeq 4\sigma_B^2 = \frac{2}{3}2^{-2b}$. The output noise to signal ratio for the case of step-by-step scaling (corresponding to $k = N$) and white input can be obtained as

$$\frac{E\{|An_k|^2\}}{E\{|A_k|^2\}} = 2N2^{-2b}. \tag{28}$$

Note that no additional scaling is required for computing (22). This can be seen from the fact that there is no overflow in $\hat{C}_k, \forall k$ because of the stage by stage FFT scaling. Also we know from the DFT equation that a scaling of $1/N$ is sufficient for overflow in computation of A_k [16], [17] which is obtained again because of the stage by stage scaling. So it follows that as A_k is computed in (22) there is no need of additional scaling. It follows that real factor FFT's have lower NSR as compared to radix-2 Cooley-Tukey FFT.

C. Simulation results for Signal to Quantization Noise Ratio (SQNR)

Figure 3 gives the SQNR simulation results for 64 point DFT for the Cooley Tukey (market as 'CT') FFT algorithm and the 'real-factor' FFT (marked as 'New') presented in this paper when the twiddle factor and input word lengths vary from 8 to 32 bits. Observe that as predicted by theoretical results the modified real-factor FFT enjoys a 3 dB gain in terms of SQNR (Note that SQNR is inverse of NSR).

IV. Conclusion

In this paper we have shown that the arithmetic complexity of real-factor FFT's is significantly less than that of radix-2 Cooley Tukey FFT. We have also shown that this FFT has a significant advantage as compared to Cooley-Tukey FFT when it comes to fixed point implementation as it has lower number of noise sources (real multiplications) and hence lower NSR. This algorithm is of practical importance as radix-2 FFTs are generally preferred in ASIC implementation and for software defined radio applications due to ease of reconfigurability. Overall these factors are in favor of real factor FFT's. However, the disadvantage is additional memory for storage of C_0 for each stage (additional memory access) but as the cost of memory decreases this might not be significant.

References

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computation of the complex Fourier series," Math. Computation, vol. 19, pp. 297301, Apr. 1965.

[2] S. Winograd, "On computing the discrete Fourier transform," Proc. Nat. Acad. Sci. USA, Vol. 73, April 1976, pp. 1005–1006.

[3] C. M. Rader and N. M. Brenner, "A new principle for fast Fourier transformation," IEEE Trans. Acoust., Speech, Signal Processing, vol. 24, pp. 264-265, 1976.

[4] K.M. Cho and G.C. Temes, "Real-factor FFT algorithms," Proc. ICASSP 78, Tulsa, OK, April 1978, pp. 634–637.

[5] A. V. Oppenheim and C. J. Weinstein, "Effects of finite register length in digital filtering and the fast Fourier transform," Proc. of IEEE. 1972, pp. 957–976

[6] T. Tran, B. Liu, "Fixed-point fast Fourier transform error analysis", IEEE Trans. on ASSP, 1976, vol.24(6), pp. 563–573.

[7] W. M. Gentleman and G. Sande, "Fast Fourier transform for fun and profit," in 1966 Fall Joint Cornput. Conf., AFIPS Con5 Proc.,v ol. 29. New York: Spartan, pp. 563–578.

[8] T. Kaneko and B. Liu, "Accumulation of roundoff errors in fast Fourier transforms", J. Ass. Comput. Mach., vol. 17, pp. 537–654, Oct. 1970.

[9] C. J. Weinstein, "Roundoff noise in floating point fast Fourier transform computation", IEEE Trans. Audio Electroacoust ., vol. AU-17,pp. 209–215, Sept. 1969.

[10] G. U. Ramos, "Roundoff error analysis of the fast Fourier transforrn", Math. Cornput., vol. 25, pp. 757–768, Oct. 1971.

[11] C. J.Weinstein, "Qunatization effects in digital filters", MIT Lincoln Lab. Tech. Rep. 468, ASTIA Doc. DDC AD-706862, Nov. 21,1969.

[12] Wade Lowdermilk and Fred Harris, "Finite Arithmetic Considerations for the FFT Implemented in FPGA-Based Embedded Processors in Synthetic Instruments", In IEEE Instrumentation and Measurement Magazine, August 2007.

[13] W.-H. Chang and T. Nguyen, On the Fixed-Point Accuracy Analysis of FFT Algorithms, IEEE Transactions On Signal Processing, Vol. 56, No. 10, pp. 4673-4682, October 2008.

[14] P. D. Welch, "A fixed point fast Fourier transform error analysis", IEEE Trans. Audio Electroacoust., vol. AU-17, pp. 151–157, June 1969.

[15] D V. James, "Quantization errors in fast Fourier transform", IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-23, pp. 277–283, June 1975.

[16] A. V. Oppenheim and R. Schafer," Digital Signal Processing". Pearson Education, 2004.

[17] Sanjit Mitra, "Digital Signal Processing: A Computer Approach" 3rd edition, Chapter 2, McGraw- Hill, 2006.