The Chirp z-Transform Algorithm

L. R. RABINER, Member, IEEE
K. W. SCHAEFER, Member, IEEE
Bell Telephone Laboratories, Inc.
Murray Hill, N. J.
C. M. RADER, Member, IEEE
Lincoln Laboratory
Lexington, Mass.

Abstract

A computational algorithm for numerically evaluating the z-transform of a sequence of \( N \) samples is described. This algorithm has been named the chirp z-transform (CZT) algorithm. Using the CZT algorithm one can efficiently evaluate the z-transform at \( M \) points in the \( z \)-plane which lie on circular or spiral contours beginning at any arbitrary point in the \( z \)-plane. The angular spacing of the points is an arbitrary constant, and \( M \) and \( N \) are arbitrary integers.

The algorithm is based on the fact that the values of the z-transform on a circular or spiral contour can be expressed as a discrete convolution. Thus one can use well-known high-speed convolution techniques to evaluate the transform efficiently. For \( M \) and \( N \) moderately large, the computation time is roughly proportional to \( (N+M) \log (N+M) \) as opposed to being proportional to \( N M \) for direct evaluation of the z-transform at \( M \) points.

I. Introduction

In dealing with sampled data the z-transform plays the role which is played by the Laplace transform in continuous time systems. One example of its application is spectrum analysis. We shall see that the computation of sampled z-transforms, which has been greatly facilitated by the fast Fourier transform (FFT) [1], [2] algorithm, is still further facilitated by the chirp z-transform (CZT) algorithm to be described in this paper.

The z-transform of a sequence of numbers \( x_n \), is defined as

\[
X(z) = \sum_{n=0}^{N-1} x_n z^{-n},
\]

a function of the complex variable \( z \). In general, both \( x_n \) and \( X(z) \) could be complex. It is assumed that the sum on the right side of (1) converges for at least some values of \( z \). We restrict ourselves to the z-transform of sequences with only a finite number \( N \) of nonzero points. In this case, we can rewrite (1) without loss of generality as

\[
X(z) = \sum_{n=0}^{N-1} x_n z^{-n},
\]

where the sum in (2) converges for all \( z \) except \( z = 0 \).

Equations (1) and (2) are like the defining expressions for the Laplace transform of a train of equally spaced impulses of magnitudes \( x_n \). Let the spacing of the impulses be \( T \) and let the train of impulses be \( \sum_{k=-\infty}^{\infty} x_k \delta(t-kT) \). Then the z-transform is \( \sum_{k=-\infty}^{\infty} X(z) \) which is the same as \( X(z) \) if we let

\[ z = e^{j2\pi T}. \]

If we are dealing with sampled waveforms the relation between the original waveform and the train of impulses is well understood in terms of the phenomenon of aliasing. Thus the z-transform of sequences of sampled waveforms in the original waveform in a way which is well understood. The Laplace transform of a train of impulses repeats its values taken in a horizontal strip of the \( s \)-plane of width \( 2\pi / T \) in every other strip parallel to it. The z-transform maps each such strip into the entire \( z \)-plane, or conversely, the entire \( z \)-plane corresponds to any horizontal strip of the \( s \)-plane, e.g., the region \( -\infty < \sigma < \infty, -\pi / T < \gamma < \pi / T \) where \( s = \sigma + j\omega \). In the same correspondence, the \( j \omega \) axis of the \( z \)-plane, along which we generally evaluate the Laplace transform with the Fourier transform, is the unit circle in the \( z \)-plane, and the origin of the \( s \)-plane corresponds to \( z = 1 \). The interior of the \( z \)-plane unit circle corresponds to the left half of the \( s \)-plane, and the exterior corresponds to the right half plane. Straight lines in the \( s \)-plane corresponding to \( s = a + j\omega \) with \( a < 0 \) are mapped to the \( z \)-plane. Fig. 1 shows the correspondence of a contour in the \( s \)-plane to a contour in the \( z \)-plane. To evaluate the Laplace transform of the impulse train along the linear contour is to evaluate the z-transform of the sequence along the spiral contour.

Values of the z-transform are usually computed along the path corresponding to \( \text{Re}(s) = 0 \), namely the unit circle. This gives the discrete equivalent of the Fourier transform and has many applications including the estimation of spectra, filtering, interpolation, and correlation. The applications of computing z-transforms off the unit circle are fewer, but one is presented elsewhere [6].

The special case which has received the most attention is the set of points equally spaced around the unit circle,

\[ z_k = e^{j2\pi k / N}, \]

for which

\[ X(z_k) = \sum_{n=0}^{N-1} x_n (z_k)^{-n}. \]

Equation (6) is called the discrete Fourier transform (DFT). The reader may easily verify that, in (5), other values of \( k \) merely repeat the same \( N \) values of \( z_k \), which are the \( N \)th roots of unity. The discrete Fourier transform has assumed considerable importance, partly because of its nice properties, but mainly because since 1965 it has become widely known that the computation of (6) can be achieved, not in the \( N^2 \) complex multiplications and additions called for by direct application of (6), but in something of the order of \( N \log N \) operations if \( N \) is a power of two, or \( N^2 \log N \) operations if the integers \( m_i \) are the prime factors of \( N \). Any algorithm which accomplishes this is called an FFT. Much of the importance of the FFT is that DFT may be used as a stepping stone to computing lagged products such as convolutions, autocorrelations, and cross correlations more rapidly than before [3], [4].

The DFT has, however, some limitations which can be eliminated using the CZT algorithm which we will describe. We shall investigate the computation of the z-transform on a more general contour, of the form

\[ z_k = e^{j2\pi k / N}, \]

and

\[ W = W_m e^{j2\pi m / N}, \]

(See Fig. 2.) The case \( m = 1, N = N, \) and \( W = e^{j2\pi / N} \) corresponds to the DFT. The general z-plane contour begins with the point \( z = 1 \) and, depending on the value of \( W \), spirals in or out with respect to the origin. If \( W \) is 1, the contour is an arc of a circle. The angular spacing of the samples is \( 2\pi / N \). The equivalent \( z \)-plane contour begins with the point

\[ z_0 = 1 + jW_0 = \frac{1}{N} \text{ in } A \]

and the general point on the \( z \)-plane contour is

\[ z_k = z_0 + k(W_0 + j\omega_0) = \frac{1}{N}(1 + k \text{ in } W_0), \]

\[ k = 0, 1, \ldots, M - 1. \]

Since \( A \) and \( W \) are arbitrary complex numbers we see that the points \( z_k \) lie on an arbitrary straight line segment of arbitrary length and sampling density. Clearly the contour indicated in (7a) is not the most general contour but it is considerably more general than that for which the DFT applies. In Fig. 2, an example of this more general contour is shown in both the \( z \)-plane and the \( s \)-plane.

To compute the z-transform along this more general contour would seem to require \( NM \) multiplications and additions as the special symmetries of \( \exp (j2\pi k / N) \) which are exploited in the derivation of the FFT are absent in the more general case. However, we shall see that by using the sequences \( W^k \) in various roles we can apply the FFT to the computation of the z-transform along the contour of (7a). Since for \( W_0 = 1 \), the sequence \( W^k = 1 \) is a complex sinusoid of linearly increasing frequency, and since

\[ k \]
The FFT, although it is also considerably slower. The additional freedoms offered by the CZT include the following:

1) The number of time samples does not have to equal \(N\), but \(N\) is arbitrary. The spectrum samples \(X_n\) are not required to be a function of the computational effort and requires a time roughly proportional to \(N\) samples of \(X_n\) and \(M\) samples of \(X_\nu\) where \(A\) and \(W\) have also been chosen. Begin with a waveform in the form of \(N\) simplex \(x_n\) and \(M\) samples of \(x_\nu\) where \(A\) and \(W\) have also been chosen.

1) Choose \(L\), the smallest integer greater than or equal to \(N+M-1\) which is also compatible with our high-speed FFT program. For most users this will mean \(L\) is a power of two. Note that while many FFT programs will work for arbitrary \(L\), they are not equally efficient for all \(L\). At the very least, \(L\) should be highly composite.

2) Form an \(L\) point sequence \(y_n\) from \(x_n\) by \(y_n = x_n W^{-n\nu^2/2}\). (17)

3) Compute the \(L\) point DFT of \(y_n\) by the FFT. Call this \(Y_k\).

4) Define an \(L\) point sequence \(z_n\) by the relation \(W^{-n\nu^2/2} = 0 \leq n \leq M - 1\).

Of course, if \(L\) is exactly equal to \(N+M-1\), the region in which \(z_n\) is arbitrary will not exist. If the region does exist an obvious possibility is to assign \(M\), the desired number of points of the \(z\)-transform we compute, until the region does not exist.

Note that \(z_n\) could be cut into two with a cut between \(n = M - 1\) and \(n = L - N + 1\) and if the two pieces were abutted together differently, the resulting sequence would be a slice out of the indefinite length sequence \(W^{-n\nu^2/2}\). This is illustrated in Fig. 4. The sequence \(z_n\) is defined the way it is in order to force the circular convolution to give us the desired numerical results of an ordinary convolution.

5) Compute the DFT of \(z_n\) and call it \(Z_k\).

6) Multiply \(Y_k\) and \(Z_k\) point by point, giving \(G_k\): \(G_k = Y_k Z_k, \quad 0 \leq k \leq L - 1\).

7) Compute the \(M\) point IDFT of \(G_k\).

8) Multiply \(G_k\) by \(W^{k\nu^2/2}\) to give the desired \(X_k\): \(X_k = W^{k\nu^2/2}, \quad 0 \leq k \leq M - 1\). The \(g_\nu\) for \(k \geq M\) are discarded.

The \(M\) point IDFT of \(x_\nu\) is \(x_n W^{n\nu^2/2}\). The sequence \(x_n\) is formed by multiplying the \(M\) values of the \(z\)-transform.
III. Fine Points of the Computation

Operation Count and Timing Considerations

An operation count can be made, roughly, from the eight steps just presented. We will give it step by step because there are, of course, many possible variations to be considered.

1) We assume that step 1, choosing L, is a negligible operation.

2) Forming $y_k$ from $x_k$ requires $N$ complex multiplications, not counting the generation of the constants $A^{-n/2}$. The constants can be precomputed, as needed, or generated recursively as needed. The recursive computation would require two complex multiplications per point.

3) An $L$ point DFT requires a time $kpcL$ for $L$ a power of two, and a very simple FFT program. More complicated (but faster) programs have more complicated computing time formulas.

4) $v_k$ is computed for either $M$ or $N$ points, whichever is greater. The symmetry in $A^{-n/2}$ preseats the other values of $v_k$ to be obtained without computation. Again, $v_k$ can be computed recursively. The FFT takes the same time as that in step 3.

5) The same contour is used for many sets of data, $L$ need only be computed once, and stored.

6) This step requires $L$ complex multiplications.

7) This is another FFT and requires the same time as that in step 3.

8) This step requires $M$ complex multiplications.

As the number of samples of $x_k$ or $x_k N$ grow large, the computation time for the CZT grows asymptotically as something proportional to $L, \log L$. This is the same sort of asymptotic dependence of the FFT, but the constant of proportionality is bigger for the CZT because two or three FFT's are required instead of one, and because $L$ is greater than $M$ or $N$. Still, the CZT is faster than the direct computation of (10) even for relatively modest values of $M$ and $N$, of the order of 20.

Reduction in Storage

The CZT can be put into a more useful form for computation by redifining the substitution of (11) to read

$$n_k = \frac{(a-M) L}{2} + \frac{k}{L} = (a-M) + \frac{k}{L}.$$  

Equation (12) can now be rewritten as

$$x_n = A^{-n/2} \sum_{k=0}^{L-1} x_k A^{k/n} e^{j2\pi k^2 L^{-1}} A^{-n/2}.$$  

The form of the new equation is similar to (12) in that the input data $x_k$ are re-weighted by a complex sequence $(A^{-n/2})$, convolved with a second sequence $(A^{-n/2} e^{j2\pi k^2 L^{-1}})$, and post-weighted by a third sequence $(A^{-n/2})$ to compute the output sequence $x_n$. However, there are differences in the detailed procedures for realizing the CZT. The input data $x_k$ can be thought of as having been shifted by $N$ samples to the left; e.g., $x_k$ is weighted by $A^{-n/2}$ instead of $A^{-n/2}$. The region over which $y_k$ is weighted must be formed, in order to obtain correct results from the convolution, is

$$-N + 1 \leq n \leq N - 1,$$  

by choosing $N = (N-M)/2$. It can be seen that the limits over which $y_k$ is evaluated are symmetric; i.e., $y_k = y_{-k}$ is a symmetric function in both its real and imaginary parts. Thus the transform of $y_k$ is also symmetric in both its real and imaginary parts. It can be seen that using this special value of $N$, only $(L/2 + 1)$ points of $y_k$ need be calculated and stored, and these points can be transformed using an $L/2$ point transform. Hence the total storage required for the transform of $W^{-n/2}$ is $L/2$ points.

The only additional modifications to the detailed procedures for evaluating the CZT presented in Section II of this paper are: 1) following the $L$ point DFT of step 7, the data of array $y_n$ must be shifted to the left by $N/2$ locations; and 2) the weighting factor of the $y_k$ is $A^{-n/2}$ rather than $W^{-n/2}$. The additional factor $W^n$ represents a data shift of $N$ samples to the right, thus compensating the initial shift and keeping the effective positions of the data invariant to the value of $N$ used.

An estimate of the storage required to perform the CZT can now be made. Assuming that the entire process is done in one piece in core, storage is required for $L/2$ points, for which takes $L/2$ locations; for $y_n$, which takes $L/2$ locations; and perhaps for some other quantities which we wish to save, e.g., the input, or values of $W^{-n/2}$ or $A^{-n/2}$.

Additional Considerations

Since the CZT permits $N=2^L$, it is possible that occasions will arise where $M=N$ or $N=2M$. In these cases, if the smaller number is small enough, the direct method is preferred to the method proposed in (10) even for relatively modest values of $M$ and $N$, of the order of 20.

As the number of samples of $x_k$ or $x_k N$ grow large, the computation time for the CZT grows asymptotically as something proportional to $L, \log L$. This is the same sort of asymptotic dependence of the FFT, but the constant of proportionality is bigger for the CZT because two or three FFT's are required instead of one, and because $L$ is greater than $M$ or $N$. Still, the CZT is faster than the direct computation of (10) even for relatively modest values of $M$ and $N$, of the order of 20.

V. Summary

A computational algorithm for numerically evaluating the z-transform of a sequence of $N$ time samples was presented. This algorithm, entitled the chirp z-transform algorithm, enables the evaluation of the z-transform at $M$ equi-angularly spaced points on contours which spiral in or out (circles being a special case) from an arbitrary starting point in the z-plane. In the $z$-plane the equivalent contour is an arbitrary straight line.

The CZT algorithm has great flexibility in that neither $M$ or $N$ need be composite numbers; the output point spacing is arbitrary; the contour is fairly general and $N$ need not be the same as $M$. The flexibility of the CZT algorithm is due to being able to express the z-transform on the above contours as a convolution, permitting the use of well-known high-speed convolution techniques to evaluate the convolution.

Applications of the CZT algorithm include enhancement of medical data for use in spectral analysis; high-speed, narrowband frequency analysis; and time interpolation of data from one sampling rate to any other sampling rate. These applications are described in detail elsewhere [6].
number of samples. Examples illustrating how the CZT algorithm is used in specific cases are included elsewhere [6]. It is anticipated that other applications of the CZT algorithm will be found.

Appendix

The purpose of this Appendix is to show how the FFT's of two real, symmetric \( L \) point sequences can be obtained using one \( L/2 \) point FFT.

Let \( x_n \) and \( y_n \) be two real, symmetric \( L \) point sequences with corresponding DFT's \( X_k \) and \( Y_k \). By definition,

\[
x_n = y_{L-n}
\]

and it is easily shown that \( X_k \) and \( Y_k \) are real, symmetric \( L \) point sequences, so that

\[
X_k = X_{L-k} = \Re\{X_k\}
\]

\[
Y_k = Y_{L-k} = \Im\{X_k\}
\]

for \( k = 1, 2, \ldots, L/2 - 1 \). The remaining values of \( X_k \) and \( Y_k \) are obtained from the relations

\[
X_k = \sum_{n=0}^{L-1} x_n \exp(-j2\pi nk/L)
\]

\[
Y_k = \sum_{n=0}^{L-1} y_n \exp(-j2\pi nk/L)
\]

Define a complex \( L/2 \) point sequence \( z_n \), whose real and imaginary parts are

\[
\Re\{z_n\} = x_n - y_{L-n}
\]

\[
\Im\{z_n\} = x_n + y_{L-n}
\]

The \( L/2 \) point DFT of \( z_n \), is denoted \( U_k \), and is calculated by the FFT. The values of \( X_k \) and \( Y_k \) may be computed from \( U_k \) using the relations

\[
X_k = 1/2 \left( \Re\{U_k\} + \Re\{U_{L-k}\} \right)
\]

\[
Y_k = 1/2 \left( \Im\{U_k\} - \Im\{U_{L-k}\} \right)
\]

\[
x_k = X_k + Y_k
\]

\[
y_k = X_k - Y_k
\]

Lawrence R. Rabiner (S'62-M'67), for a photograph and biography, please see page 13 of the March, 1969, issue of this TRANSACTIONS.

Ronald W. Schafer (S'62-M'67) was born in Tecumseh, Neb., on February 17, 1938. He received the B.S.E.E. and M.S.E.E. degrees from the University of Nebraska, Lincoln, in 1961 and 1962, respectively, and the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in February, 1968.

From 1962 to 1965 he served as an Instructor in the Department of Electrical Engineering at the University of Nebraska, and from 1964 until 1968 he was an Instructor in the Department of Electrical Engineering at M.I.T., where he received a departmental award for teaching. During 1965 he was with the M.I.T. Electronic Systems Laboratory, where he worked on digital computer simulations of guidance and control systems, and from 1966 to 1968 he was with the M.I.T. Research Laboratory of Electronics. At present he is engaged in research on digital speech processing techniques at Bell Telephone Laboratories, Inc., Murray Hill, N. J.

Dr. Schafer is a member of the Iota Kappa Nu, Sigma Xi, and the Acoustical Society of America.

C. M. Rader (S'59-M'62), for a photograph and biography, please see this issue, page 76.
A Bound on the Output of a Circular Convolution with Application to Digital Filtering

ALAN V. OPPENHEIM, Member, IEEE
CLIFFORD VERSTEYNEN, Student Member, IEEE
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, Mass. 02173

Abstract
When implementing a digital filter, either in hardware or on a computer, it is important to utilize in the design a bound or estimate of the largest output value which will be obtained. Such a bound is particularly useful when fixed point arithmetic is to be used, since it assists in determining register lengths necessary to prevent overflow. This paper considers the class of digital filters which have an impulse response of finite duration and are discrete Fourier transform. A least upper bound is obtained for the maximum possible output of a circular convolution for the general case of complex input sequences. For the case of real input sequences, a lower bound on the least upper bound is obtained. The use of these bounds on the ratio max [|Y|/|X|] allows maximum energy transfer through the filter, consistent with the requirement that Y does not overflow the register length.

I. Introduction
When implementing a digital filter, either in hardware or on a computer, it is important to utilize in the design a bound or estimate of the largest output value which will be obtained. Such a bound is particularly useful when fixed point arithmetic is to be used, since it assists in determining register lengths necessary to prevent overflow. In this paper we consider the class of digital filters which have an impulse response of finite duration and are discrete Fourier transform (DFT). The output samples of such a filter are obtained from the results of N-point circular convolutions of the filter impulse response (kernel) with sections of the input. These convolutions are obtained by computing the DFT of the input section, multiplying by the DFT of the impulse response, and inverse transforming the result. Stockham [1] has discussed procedures for utilizing the results of these circular convolutions to perform linear convolutions, rationalizations for choosing the transform length N, and speed advantages to be gained by using the fast Fourier transform (FFT) to implement the DFT. We concern ourselves here only with bounding the output of the N-point circular convolutions.

II. Problem Statement
According to the above discussion, we would like to determine an upper bound on the maximum modulus of an output value that can result from an N-point circular convolution. With \( x_k \) denoting the input sequence, \( h_k \) denoting the kernel, and \( y_k \) denoting the output sequence, we have
\[
Y_k = H_k X_k \tag{1}
\]
where it is understood that, in general, each of the three sequences may be complex. The circular convolution is accomplished by forming the product
\[
Y_k = H_k X_k \tag{2}
\]
where
\[
X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n \exp\left[-j2\pi n k / N\right] \quad k = 0, 1, \ldots, N - 1 \tag{3}
\]
and
\[
Y_k = \frac{1}{N} \sum_{n=0}^{N-1} y_n \exp\left[-j2\pi n k / N\right] \quad k = 0, 1, \ldots, N - 1 \tag{4}
\]
and
\[
H_k = \frac{1}{N} \sum_{n=0}^{N-1} h_n \exp\left[-j2\pi n k / N\right] \quad k = 0, 1, \ldots, N - 1 \tag{5}
\]
with \( W = \exp\left[-j2\pi / N\right] \).

For convenience in notation, we imagine the computation to be carried out on fixed point fractions. Thus we bound the input values so that
\[
|X_k| \leq 1 \tag{6}
\]
By virtue of (3) we are then assured that
\[
|X_k| \leq 1 \tag{7}
\]

III. Derivation of Results

Proof of Result A
Parseval's relation requires that
\[
\sum_{k=0}^{N-1} |Y_k|^2 = \sum_{k=0}^{N-1} |X_k|^2 |H_k|^2 \tag{8}
\]
and
\[
|Y_k|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |x_n|^2 |H_k|^2 \tag{9}
\]
Substituting (9) into (8) and using (7),
\[
\sum_{k=0}^{N-1} |Y_k|^2 = \frac{N}{N} \sum_{k=0}^{N-1} |X_k|^2 \tag{10}
\]
or, using (9),
\[
\sum_{k=0}^{N-1} |Y_k|^2 \leq \frac{N}{N} \sum_{k=0}^{N-1} |x_n|^2 \tag{11}
\]
with equality if and only if \( |H_k| = 1 \). However, (6) requires that
\[
\sum_{k=0}^{N-1} |x_n|^2 \leq N \tag{12}
\]
with equality if and only if \( |H_k| = 1 \). Combining (11) and (12),
\[
\sum_{k=0}^{N-1} |y_n|^2 \leq N \tag{13}
\]
But
\[
|y_n|^2 \leq \frac{N}{N} |x_n|^2 \tag{14}
\]
and therefore
\[
|y_n|^2 \leq \sqrt{N}. \tag{15}
\]

Proof of Result B
To show that \( \sqrt{N} \) is a least upper bound on \( |y_n| \), we review the conditions for equality in the inequalities used above. We observe that for equality to be satisfied in (15), it must be satisfied in (11), (12), and (14), requiring that
1. \( |H_k| = 1 \)
2. \( |x_n| = 1 \)
3. Any output sequence \( |y_n| \) which has a point whose modulus is equal to \( \sqrt{N} \) can contain only one nonzero point.

The third requirement can be rephrased as a requirement on the input sequence and on the sequence \( H_k \). Specifically, if the output sequence contains only one nonzero point then \( Y_k \) for this sequence must be of the form
\[
Y_k = \alpha W^{\epsilon} \tag{16}
\]
declaring \( \alpha \) a real constant and \( \epsilon \) an integer so that, from (12),
\[
H_k X_k = |\alpha| \exp\left[\frac{2\pi \epsilon k}{N}\right] \tag{17}
\]
We can express \( H_k \) and \( X_k \) as
\[
H_k = e^{\epsilon} \quad \text{and} \quad X_k = |\alpha| \exp\left[\frac{2\pi \epsilon k}{N}\right] \tag{18}
\]
where we have used the fact that \( |H_k| = 1 \). For (16) to be satisfied, then
\[
|X_k| = |\alpha| \tag{19}
\]
and
\[
\alpha = -\alpha + \frac{2\pi \epsilon k}{N} \tag{20}
\]
Therefore, requirement 3) can be replaced by the requirement that:
3'). \( |X_k| = \text{constant} \) and the phase of \( H_k \) be chosen to satisfy (18).
As an additional observation, we note that for any input sequence \( \{x_n\} \),
\[
|y_n| \leq \sum_{k=0}^{N-1} |H_k| |x_k|
\]
with equality for some value of \( n \) if and only if \( |H_k| = 1 \) and the phase of \( H_k \) is chosen on the basis of (18). Therefore, for any \( \{x_n\} \) the output modulus is maximized when \( H_k \) is chosen in this manner. This maximum value will only equal \( \sqrt{N} \), however, if, in addition, \( |x_n| = 1 \) and \( x_k = \text{constant} \).

For \( N \) even, a sequence having the property that \( |x_n| = 1 \) and \( |x_k| = \text{constant} \) is (see Appendix) the sequence
\[
x_n = \exp \left( j \pi k^2 \right) \quad |H_k| = \text{constant}
\]
and
\[
H_k = \begin{cases} 
1 & R_k > 0 \\
-1 & R_k \leq 0,
\end{cases}
\]
then
\[
y_n = \sum_{k=0}^{N-1} |H_k|.
\]

Similarly, if we choose \( x'_n = \sin(\pi k^2/N) \), then we can choose \( |H_k| \) in such a way that
\[
y_n = \sum_{k=0}^{N-1} |H_k|.
\]

We note that since \( |x_n| \) and \( |x'_n| \) are both real, the values \( x_n \) and \( x'_n \) will be obtained with \( |H_k| \) having even magnitude and odd phase, corresponding to real and/or imaginary sequences, having even and odd phase, respectively.

Adding (23) and (26b) and using (21),
\[
\beta \geq \frac{N}{2}.
\]
Since we argued previously that \( \beta \leq \sqrt{N} \), Result C is proved.

IV. Discussion

The bound obtained in the previous sections can be utilized in several ways. If the DFT computation is carried out using a block floating-point strategy so that arrays are rescaled only when overflows occur, then a final rescaling must be carried out after each section is processed so that it is compatible with the results from previous sections. For general input and filter characteristics, the final rescaling can be chosen based on the bounds given here to ensure that the output will not exceed the available register length.

The use of block floating-point computation requires the incorporation of an overflow test. In some cases we may wish to incorporate this scaling in the computation in such a way that we are guaranteed never to overflow. For example, when we realize the DFT with a power of two algorithm, overflows in the FFT computation of \( \{|x_n|\} \) will be prevented by including a scaling of \( 1/\sqrt{N} \) at each stage, since the maximum modulus of an array in the computation is nondecreasing and depends at most a factor of two as we proceed from one stage to the next. With this scaling, the bound derived in this paper guarantees that with a power of two computation, scaling is not required in more than half the arrays in the inverse FFT computation. Therefore, including a scaling of \( 1/\sqrt{N} \) in the first half of the stages in the inverse FFT will guarantee that there are no overflows in the remainder of the computation. The fact that \( \beta \leq \sqrt{N} / 2 \) indicates that we restrict ourselves to only real input data, at most one rescaling could be eliminated for some values of \( N \).

The bounds derived and method of scaling mentioned above apply to the general case that is, except for the normalization of (1), they do not depend on the filter characteristics. This is useful when we wish to fix the scaling strategy with respect to any particular filter. For specific filter characteristics, the bound can be reduced. Specifically, it can be verified from (1) and (6) that in terms of \( |A_k| \),
\[
\beta \geq \sqrt{N} |A_k|.
\]

To determine the modulus of \( B \), Parseval's relation requires that
\[
\sum_{k=0}^{N-1} |x_k|^2 = N \sum_{k=0}^{N-1} |H_k|^2.
\]

Adding (27) and (26a) and using (21),
\[
x_n = \exp \left[ j \pi n^2 / N \right] \quad n = 0, 1, \ldots, N - 1
\]
has a discrete Fourier transform with constant modulus and that for \( N \) odd, the sequence
\[
x_n = \exp \left[ j \pi n^2 / N \right] \quad n = 0, 1, \ldots, N - 1
\]
has a discrete Fourier transform with constant modulus. We consider first the case of (29). Letting \( S \) denote the DFT of \( x_n \),
\[
S_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n \exp \left[ j 2\pi nk / N \right]
\]
and
\[
S_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n \exp \left[ j 2\pi nk / N \right]
\]
and
\[
S_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n \exp \left[ j 2\pi nk / N \right]
\]