Generalizing the inverse FFT off the unit circle

2019112021

April 24, 2021

1 Introduction

I would be discussing about inverse chirp Z-transform (ICZT) algorithm which is a generalization of inverse fast Fourier transform (IFFT) off the unit circle in the complex plane.

1.1 What is chirp Z-transform?

The chirp Z-transform(CZT) is the generalization of the discrete Fourier transform (DFT) or Fast Fourier transform (FFT). The differences between them are :-

- DFT samples the Z plane at uniformly-spaced points along the unit circle.
- CZT samples the Z plane not only from the unit circle but from any point on the Z-plane.
- The DFT transforms a sequence of N complex numbers $\{x_n\}$: $= x_0, x_1, x_2, \ldots, x_{N-1}$ into another sequence given by $\{X_n\}$: $=X_0, X_1, X_2, \ldots, X_{N-1}$, which is defined by:-

  X(k) $= \sum_{n=0}^{N-1} x(n) e^{-\frac{i2\pi}{N}kn}$

  for k = 0,1,2,3,4,...,N-1.
- The CFT transforms a sequence of N complex numbers $\{x_n\}$: $= x_0, x_1, x_2, \ldots, x_{N-1}$ into another sequence given by $\{X_n\}$: $=X_0, X_1, X_2, \ldots, X_{M-1}$, which is defined by:-

  X(k) $= \sum_{n=0}^{N-1} x(n) z_k^{-n}$

  $z_k = A.W^{-k}$, for k = 0,1,2,3,4,...,M-1.

  where A is the complex starting point, W is the complex ratio between points, and M is the number of points to calculate.

CZT distributes the samples along a logarithmic spiral contour (called chirp contour) that is defined by the formula $A^{-j}.W^{jk}$ where j denotes a zero-based input sample index and k denotes a zero-based output sample index.

An efficient algorithm for computing CZT has the complexity O(n.log(n)) which is same as the complexity of FFT.

1.2 What is inverse chirp Z-transform?

The ICZT is the inverse of the CZT. It is the generalised version of IDFT/IFFT, just like CZT is the generalised version of DFT/FFT. The ICZT maps the output of the CZT back to the input.

The IDFT transforms the sequence $\{X_n\}$: $=x_0,x_1,x_2,\ldots,X_{N-1}$ back to the sequence $\{x_n\}$: $= x_0,x_1,x_2,\ldots,x_{N-1}$ by the formula:-

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{\frac{i2\pi}{N}kn}$$

Similarly, ICZT transforms the sequence $\{X_n\}$: $=X_0,X_1,X_2,\ldots,X_{M-1}$ back to the sequence $\{x_n\}$: $= x_0,x_1,x_2,\ldots,x_{N-1}$.

2 Algorithms for calculating ICZT and CZT

As stated before, CZT is defined as:

$$X(k) = \sum_{n=0}^{N-1} X(n)A^{-n}W^{nk}$$

Let A = diag(A^0,A^{-1},A^{-2},A^{-3},...,A^{N-1})

The CZT can be written as $\mathbf{X} = \mathbf{W}\ \mathbf{A}\ \mathbf{x}$ where $\mathbf{W}$ is a Vandermonde matrix defined as:-

$$\mathbf{W} = \begin{bmatrix} W^{0.0} & W^{0.1} & \ldots & W^{(N-1).0} \\ W^{0.1} & W^{1.1} & \ldots & W^{(N-1).1} \\ \vdots & \vdots & \ldots & \vdots \\ W^{0.(M-1)} & W^{1.(M-1)} & \ldots & W^{(N-1).(M-1)} \end{bmatrix}$$

Note, we can express W as a product of a diagonal matrix, a Toeplitz matrix and another diagonal matrix. Using Bluestein's substitution, we can write nk = $\frac{n^2+k^2-(n-k)^2}{2}$

Thus, $X(k) = \sum_{n=0}^{N-1} X(n)A^{-n}W^{\frac{n^2}{2}}W^{\frac{k^2}{2}}W^{-\frac{(n-k)^2}{2}}$

$= W^{\frac{k^2}{2}}\sum_{n=0}^{N-1} X(n)A^{-n}W^{\frac{n^2}{2}}W^{-\frac{(n-k)^2}{2}}$

We can write

$\mathbf{P} = \mathrm{diag}(W^{\frac{0^2}{2}}, W^{\frac{1^2}{2}}, W^{\frac{2^2}{2}},\ldots,W^{\frac{(M-1)^2}{2}})$

$\mathbf{Q} = \mathrm{diag}(W^{\frac{0^2}{2}}, W^{\frac{1^2}{2}}, W^{\frac{2^2}{2}},\ldots,W^{\frac{(N-1)^2}{2}})$

and let $\mathbf{V}$ be a M by N Toeplitz matrix with element in ith column and jth row as $W^{-\frac{(i-j)^2}{2}}$

As a result, we can write $\mathbf{X} = \mathbf{P}\ \mathbf{V}\ \mathbf{Q}\ \mathbf{A}\ \mathbf{x}$.

To compute the product of **P,Q,A**, we require O(n) time as they all are diagonal matrices. And the product of a Toeplitz matrix with a vector can be calculated in O(n.log(n)) time. Thus, the overall complexity for calculating CZT is O(n log(n)).

We can derive the formula for ICZT using the above expression if M = N.

$\mathbf{x} = A^{-1}\ Q^{-1}\ V^{-1}\ P^{-1}\ X$

The time complexity of inverting all the diagonal matrices is O(n). We only need to find an efficient algorithm for inverting **V**.

After applying Gohberg–Semencul formula, we can find V^{-1} in O(n.log(n)) time.

3 References

https://www.nature.com/articles/s41598-019-50234-9
https://www.nature.com/articles/s41598-020-60878-7
Mersereau, R. An algorithm for performing an inverse chirp z-transform. IEEE Transactions on Acoustics, Speech and Signal Processing 22, 387–388 (1974).