Chirp z-transform algorithm and its applications

Team-17

Paper- Rabiner, 1969

1 Brief project abstract

In this project we will discuss about an algorithm named chirp z-transform which helps us to calculate the z-transform at M points on any contour lying on the z-plane. This transform is very efficient because we use the principle of discrete convolution which can be implemented very fast using the techniques like fft.
We also discuss the time complexity of chirp z-transform as well as the required conditions, storage and any other additional considerations that we need to take while evaluating the transform.
Then we discuss the applications like enhancement of poles and high resolution narrow band frequency analysis. Later we calculate the chirp z-transform of a signal using MATLAB.

2 Background theory and intuitive idea

The chirp z-transform is obtained by sampling the z-transform at M points. We find that this is analogous to obtaining the DFT by sampling the DTFT at N points.

We know that the formula for Z transform is

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

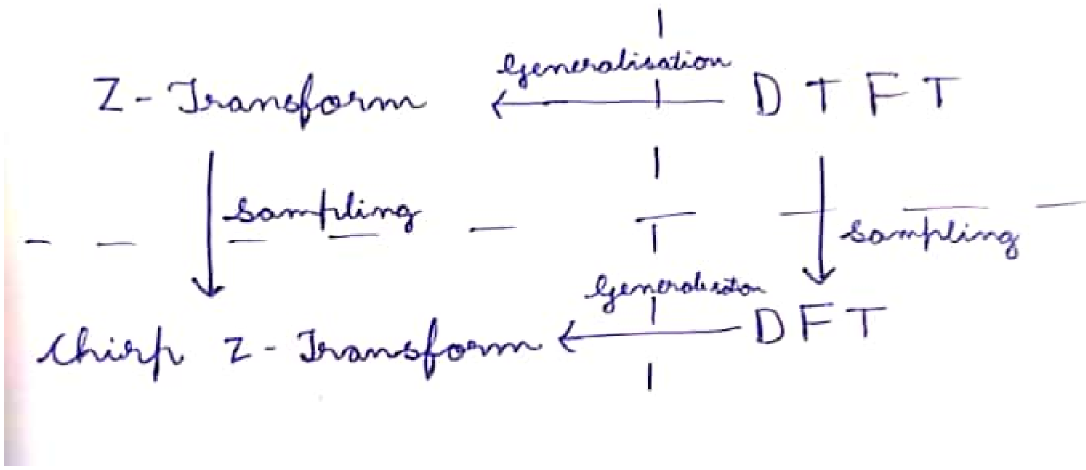
Substituting $z = e^{jw}$ we get the formula for DTFT is

$$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn}$$

So we can say that the z-transform is generalisation of DTFT.

We have DFT formula

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-jw_0kn} \text{ where } w_0 = \frac{2\pi}{N} \text{ and k=0, 1, 2, ...N-1}$$



From this we get the idea that generalised formula(known as the chirp z-transform) will be of the form

$$X[z_k] = \sum_{n=0}^{N-1} x[n] z_k^{-n} \text{ and k=0, 1, 2, ...M-1}$$

3 Derivation

We have the z transform,

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

Since we take the input signal to be a finite signal of length N, without loss of generality we can write the formula as

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

We are interested in sampling the z-transform at a M points. So we get the formula as $X_k = X(z)|_{z=z_k} = \sum_{n=0}^{N-1} x[n] z_k^{-n}$ where k=0, 1, 2, ..M-1.

The values of z_k can be parallel to the z-plane, they can be spiralling with increasing radius and so on. Just as we take a point $s = \sigma + j\omega$ in the s-plane which covers the entire, we need to take a value z_k that is a general point covering the entire z-plane.

To solve this problem, we take the generalised contour, $z_k = A.W^{-k}$ where k=0, 1, ...M-1. M is the number of samples we are interested in and A and W are both complex numbers of the form

$$A = A_0 e^{j2\pi\theta_0}$$

$$W = W_0 e^{j2\pi\phi_0}$$

A Special case $z_k = e^{-jw_0 k_0}$ has got special attention which is set of points on unit circle giving rise to the DFT.

The DFT can be calculated in $Nlog_2 N$ time if N is a power of 2 or in $N \sum m_i$ operations if these integers m_i are prime factors of N.

Now we use an algorithm called the Bluestein's algorithm which uses fft to calculate the CZT. We use W^{n^2} many times in this algorithm. For $W_0 = 1$, we get a certain sequence which are called "chirps" in radar systems which are called chirps and hence we get the term "chirp z-transform".

Bluestein's Algorithm

$$X_k = \sum_{n=0}^{N-1} x[n] A^{-n} W^{nk} \text{ where k=0, 1, 2, ...M-1}$$

We have $nk = \frac{n^2+k^2-(k-n)^2}{2}$
Substituting this in the equation of CZT we get

$$X_k = \sum_{n=0}^{N-1} x[n] A^{-n} W^{\frac{n^2}{2}} W^{\frac{k^2}{2}} W^{\frac{-(k-n)^2}{2}} \text{ where k=0, 1, 2, ...M-1}$$

Since $W^{\frac{k^2}{2}}$ is independent of n we write

$$X_k = W^{\frac{k^2}{2}} \sum_{n=0}^{N-1} x[n] A^{-n} W^{\frac{n^2}{2}} W^{\frac{-(k-n)^2}{2}}$$

Now finding X_k can be viewed as a three step process

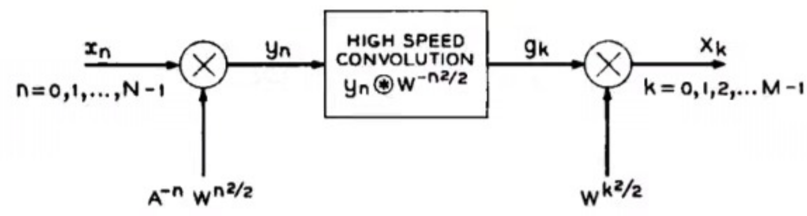
- Forming a new sequence $y_n = x_n A^{-n} W^{\frac{n^2}{2}}$ for n=0, 1, 2, ..N-1
- Convolving y_n with the sequence v_n which is defined as $v_n = W^{\frac{-n^2}{2}}$ which gives us a sequence g_k

  $$g_k = \sum_{n=0}^{N-1} y_n v_{k-n}$$

- Multiply g_k with $W^{\frac{k^2}{2}}$ to give X_k where k=0, 1, 2, .. M-1

Let us focus on the second step which is called the high speed convolution step. We are going to exploit the quickness of the fft algorithm and calculate the required convolution.

i) We know that normal convolution can be converted into cirular convolution by the concept of "zero padding" which is adding of 0's to make the signal have bigger length. The value to which the signal needs to be extended is L which the next power of 2 after N+M-1. Note that this is because the fft algorithm that we use works only for powers of 2.



So the first step is L=nextpow2(N+M-1)

ii) We apply zero padding to y_n which results in changing y_n to a new value which is as follows

$$y_n = \begin{cases} x_n A^{-n} W^{\frac{n^2}{2}}, 0 \le n \le N-1 \\ 0, N \le n \le L-1 \end{cases}$$

iii) Now we calculate the L point DFT (using FFT) of y_n and let us call it Y_r for $r = 0, 1, ...L-1$

iv) We want to perform a circular convolution. We have to keep in mind the circular shift property of the DFT while defining new v_n with zero padding because we are using v_{k-n} where we have circular shift taking place.

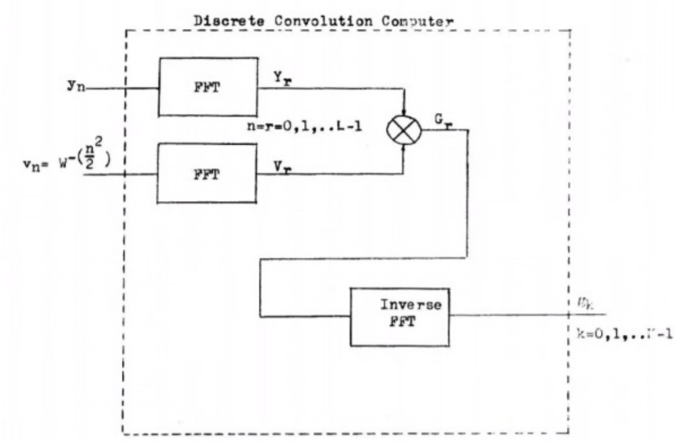
$$v_n = \begin{cases} \mathrm{W}^{\frac{-n^2}{2}}, 0 \le n \le M-1 \\ W^{\frac{-(L-n)^2}{2}} L-N+1 \le n < L \\ arbitrary - value, otherwise \end{cases}$$

A small point to note here is that if L=M+N-1, then we donot get any arbitrary values.
If we want to avoid the arbitrary values, then one obvious possibility is increasing M so that the region does not exist.

v) Now we calculate the L point DFT (using FFT) of v_n and let us call it V_r for $r = 0, 1, ...L-1$

vi) Now let us multiply Y_r and V_r to get $G_r = Y_r V_r$ for $r = 0, 1, ...L-1$.

vii)Now find inverse DFT of G_r to get g_k which gives us the required answer in the step 2 mentioned above.
To summarise,
$y[n] * v[k-n] = y_{new}[n] \circledast v_{new}[n] = IFFT(FFT(y_{new}[n])FFT(v_{new}[n]))$



4 Few comments

4.1 Advantages

- The number of time samples need not be equal to the number of samples of z-transform. Meaning we can have M samples in z domain and N samples in time domain unlike the N-DFT where we have both the samples as N.
- Neither M, nor N need to be composite integer. This freedom is given by the czt which is not provided by fft using $n\Sigma m_i$ method.
- The angular spacing of z_k is an arbitrary *constant*[1] which is not the case in DFT.
- The contour need not be a circle but can spiral in or out or any other contour.

4.2 Time complexity

- In this section we try to compute the time complexity of the process mentioned above.
- The step for calculation y_n roughly requires N complex multiplications (without including the computation of A.W)
- For this the L-point DFT requires $Llog_2L$

1) We have stated wrongly in the viva that angular spacing is arbitrary due to ϕ.It is an **arbitrary constant where ϕ is arbitrary. Once we fix it,the angular spacing between each point is $2\pi\phi$ i.e. constant**

- For calculating v_n, we require max(N, M) multiplications as there is a bit of symmetry due to n^2.

- For fft of v_n, we again get $Llog_2L$

- Multiplication of these two vectors requires L multiplications. The inverse fourier transform again requires $Llog_2L$

- Finally the multiplications requires M multiplications.

Clearly we notice that the maximum time complexity is $Llog_2L$, upon adding the values in order of L and $Llog_2L$, the $Llog_2L$ value prevails for large L and can hence be regarded as the time complexity

4.3 Reduction in storage

We can define $nk = \frac{(n-N_0)^2+k^2-(k-n+N_0)^2+2N_0k}{2}$. Now we notice that there are changes in the constant that we are going to multiply. Also the terms for which we need to calculate the fft have powers $(n - N_0)^2$ instead of n^2.

So inorder to obtain the convolution, the limits will be $-N + 1 + N_0 \leq n \leq M - N_0 + 1$. If we choose the value of $N_0 = \frac{N-M}{2}$, we can see that there is a symmetry over the limits in which we need to evaluate the values. So we can find that in $W^{\frac{n^2}{2}}$ there is symmetry in both real and imaginary parts.

This symmetry ensures that instead of L points only $\frac{L}{2} + 1$ values are needed as calculating the values half of the times is sufficient due to the symmetry and in the z domain as well, we need $\frac{L}{2} + 1$ instead of L. So we can find that L+2 space is required for calculating the fft

4.4 Limitations

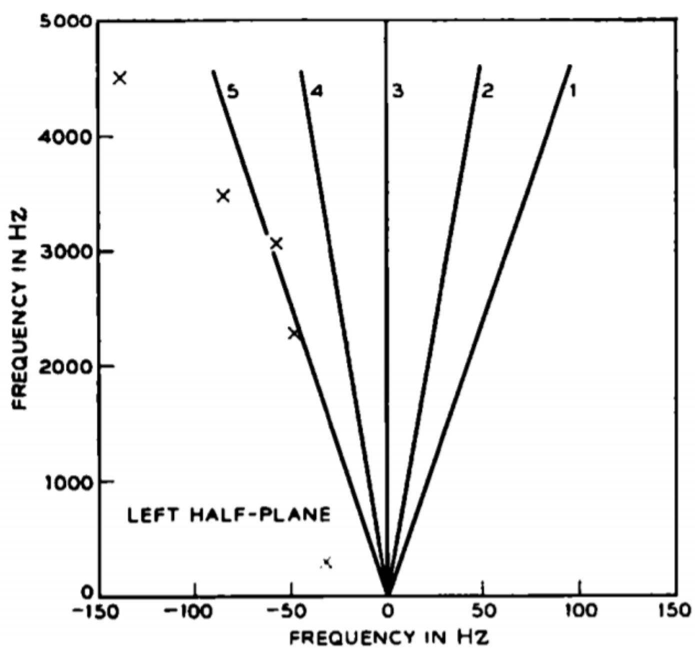
- One of the major limitation is that when the value of N, M is higher, the value of $W_0 = 0.72$(the value deviates from 1). Then we need to calculate the value of $W_0^{\frac{n^2}{2}}$ for values of $N > 1000$ we get huge error as this exceeds the floating precision of computers. So it is better if we have w_0 to be nearer to 1.

- Another limitation is that when M, N are large, the L calculation is also a bit difficult as L is the next power of 2. Since L is also a large number, for calculation of fft we need storage space in the order of L which is huge.

5 Applications

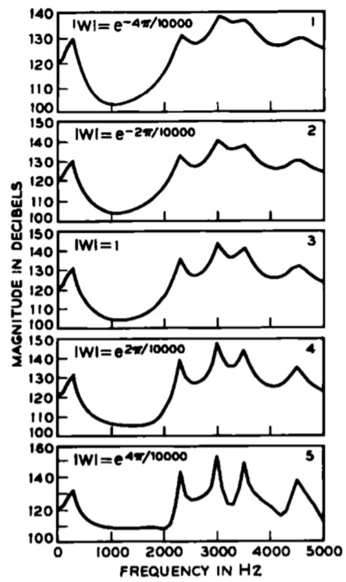
5.1 Enhancement of poles

We have discussed that the czt helps us to find the values of z-transform in spiral contours which maybe lying inside or outside the unit circle. One of the main applications of this property is the enhancement of poles.
Say we are given a transfer function and from plotting the magnitude response, we are asked to find the location of poles. Even in DFT we have dealt with such kind of problems where we were able to locate the poles based on the location of the peaks.
While evaluating the transform off the unit circle, we can see that the contour passes near the poles and hence it effectively reduces the bandwidth and sharpens the peak.
We know that $z = e^{sT}$ is the relation that is between the $s-plane$[2] and z-plane. Also we know that points on the imaginary axis in the s-plane correspond to circles whereas other lines which are parallel to imaginary axis or inclined at a particular angle give rise to different contours.
Now from this relation we can get a transition between z-plane and s-plane. Now say we have the following contours in the s-plane.



2) We are dealing with spectral analysis which involves the s-plane. So we get the poles of interest in s-plane only. We convert them into z-plane and apply czt. So transition is necessary and s-plane plots are important.

The points marked with x are the values of poles in the s-plane. We are given example contours out of which the third one is the DFT corresponding contour. Clearly we see that the contour 5 is the nearest and the contour 1 is the farthest. Now we take the corresponding contours of these lines in the z-plane and find the magnitude response.



Clearly we can see that the contour which is nearer in the s-plane (which is 5) has the better looking curve with sharp peaks and less bandwidth at the poles.

5.2 High resolution, narrow band frequency analysis

Another application of the czt is being able to display high resolution over a certain narrow band of frequency.

Say we have fft, to obtain the resolution $\leq \Delta F$, and sampling frequency at $\frac{1}{T}$, we require more than $\frac{1}{T.\Delta F}$ points. So this value N attains a very high value which is undesirable.

In practical applications, we are required to have high resolution over a certain narrow band which is accomplished by the czt due to its flexibility in allowing selection of initial frequency and angular spacing.

The below figure depicts this perfectly. We used 1600 point dft on a contour. We gave the czt a resolution of 1.25Hz in the interval 500-1500 Hz. To actually acheive this high resolution we need 8000 point DFT for which calculation takes a lot more time than CZT. For the same time complexity we see that the resolution offered by CZT is far more better.

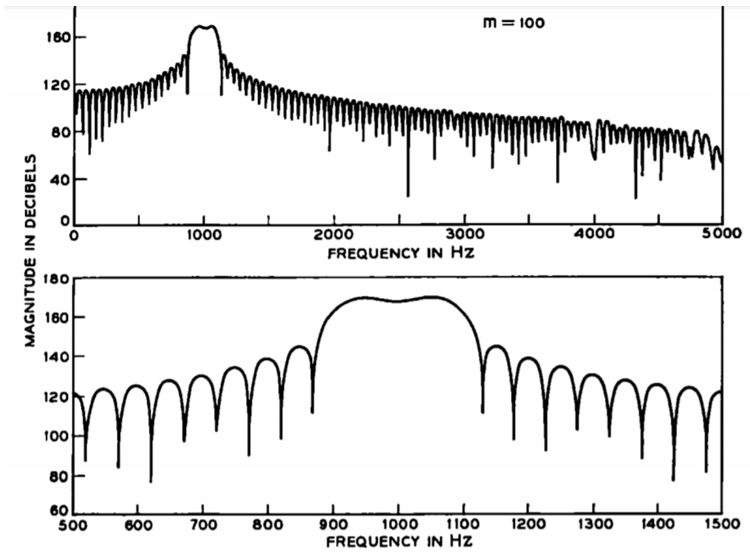
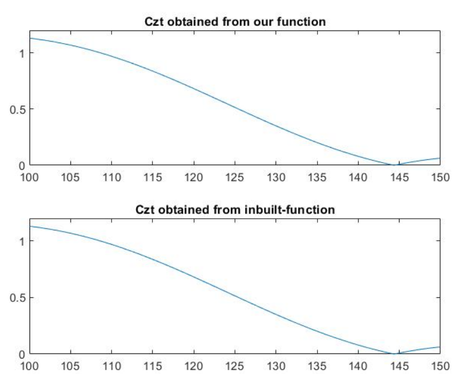
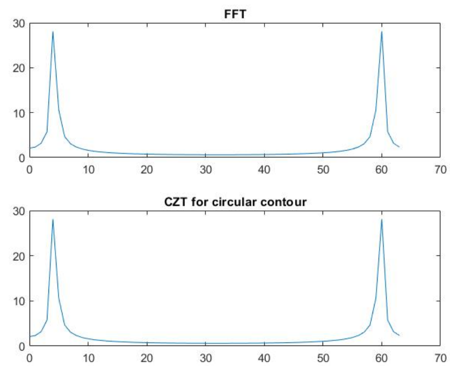


Fig. 21 — Frequency response curves for upper impulse response (200 samples) in Fig. 20. Upper curve obtained with 1,600 point fast Fourier transform (resolution 6.25 Hz). Lower curve obtained with chirp z-transform algorithm (1.25 Hz resolution).

6 MATLAB results and inferences



This figure displays the chirp z-transform graph of a signal which we calculated using inbuilt function and the function that we have coded. Both the



graphs are almost the same.
The second figure, depicts that when we take a circular contour the values obtained in DFT and CZT are same.

7 Summary

We are able to give a computational algorithm for the chirp z-transform, which has complexity $Llog_2L$. Also we were able to discuss the advantages of this transform. We also discussed a method for reducing the storage even further by a better method. Then we talked about the limitations. After that we discussed two applications which are very helpful. Finally we verified that we can code the matlab function for chirp z-transform.

8 List of references

i) Given paper Rabiner, 1969 https://ieeexplore.ieee.org/document/6772159
ii) A study of chirp z-transform, Alan Shilling https://krex.kstate.edu/dspace/bitstream/handle/2097/7844 LD2668R41972S43.pdf
iii) Enhancement of Poles in Spectral Analysis- Fung-I Tseng; Tapan K. Sarkar https://apps.dtic.mil/dtic/tr/fulltext/u2/a107739.pdf
iv) Wikipedia https://en.wikipedia.org/wiki/Chirp$_Z - transform$
$v)Matlab : https : //in.mathworks.com/help/signal/ref/czt.html$