CZT vs FFT: Flexibility vs Speed

Abstract

Bluestein's Fast Fourier Transform (FFT), commonly called the Chirp-Z Transform (CZT), is a little-known algorithm that offers engineers a high-resolution FFT combined with the ability to specify bandwidth. In the field of digital signal processing, engineers are always challenged to detect tones, frequencies, signatures, or some telltale sign that signifies a condition that must be indicated, ignored, or controlled. One of these challenges is to detect specific frequencies, for instance when looking for tones from telephones or detecting 60-Hz noise on power lines. The Goertzel algorithm described in Embedded Systems Programming, September 2002, offered a powerful tool toward finding specific frequencies faster than the FFT.

Another challenge involves analyzing a range of frequencies, such as recording frequency response measurements, matching voice patterns, or displaying spectrum information on the face of an amateur radio. To meet this challenge most engineers use the well-known FFT. The CZT gives the engineer the flexibility to specify bandwidth and outputs real and imaginary frequency components from which the magnitude and phase can be computed.

A description of the CZT and a discussion of the advantages and disadvantages of CZT versus the FFT and Goertzel algorithms will be followed by situations in which the CZT would shine. The reader will find that the CZT is very useful but that flexibility has a price.

Outline

The CZT offers middle ground between the FFT and Goertzel algorithms.

1) General overview of CZT
   a) Users specify the frequency range.
   b) Sampling frequency equal to highest frequency of interest.
   c) Number of samples should be a power of two.

2) CZT vs FFT
   a) Output of FFT ranges from 0 Hz to half the sampling frequency.
   b) For 300 to 1600 Hz and sampling frequency of 12 kHz, information outside this bandwidth is discarded.
   c) The CZT offers better resolution than FFT over the frequency range of interest.
   d) The FFT is fast and while the CZT may offer better resolution and flexibility; it pays the price when it comes to speed.
   e) However, if the number of samples acquired by the CZT is a power of two, the performance loss is minimal.

3) CZT vs Goertzel Algorithm
   a) The outputs of the Goertzel algorithm are the real and imaginary frequency components at a single frequency specified by the user.
   b) The Goertzel is so fast that at specific frequencies like 461 Hz, it can compute the magnitude "in the very same interrupt routine that is gathering the samples." (ESP September 2002).
   c) However, it only computes the magnitude and phase at that one particular frequency. Thus its use is limited unless the engineer calls the algorithm multiple times at different frequencies, which slows down the entire system.
   d) While the CZT is far slower than the Goertzel algorithm, it offers a wider bandwidth than the Goertzel algorithm at almost the same speed as the FFT.

4) Conclusion
   a) Clearly, it can be seen that by offering greater flexibility at the price of being slightly slower the FFT, the CZT is a better all-around candidate for spectrum analysis.
   b) If speed is of utmost importance at a very specific frequency, the Goertzel is a better choice.
   c) Conversely, if the bandwidth to be analyzed is approximately between 0 Hz and half the sampling frequency, then the FFT is a better choice.

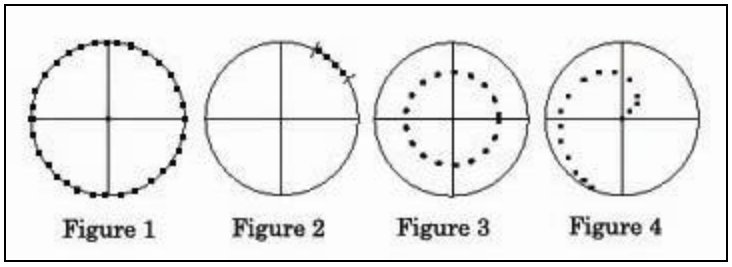
Introduction

Bluestein's Fast Fourier Transform (FFT), commonly called the Chirp-Z Transform (CZT), is a little-known algorithm that offers engineers a high-resolution FFT combined with the ability to specify bandwidth. In the field of digital signal processing, engineers are always challenged to detect tones, frequencies, signatures, or some telltale sign that signifies a condition that must be indicated, ignored, or controlled. One of these challenges is to detect specific frequencies, for instance when looking for tones from telephones or detecting 60-Hz noise on power lines. The Goertzel algorithm described in Embedded Systems Programming, September 2002, offered a powerful tool toward finding specific frequencies faster than the FFT.

Another challenge involves analyzing a range of frequencies, such as recording frequency response measurements, matching voice patterns, or displaying spectrum information on the face of an amateur radio. To meet this challenge most engineers use the well-known FFT. The CZT gives the engineer the flexibility to specify bandwidth and outputs real and imaginary frequency components from which the magnitude and phase can be computed.

CZT Description

To describe the CZT, first we need to visualize the FFT. Imagine when calculating the FFT that 0 Hz to the sampling frequency (f_s) is equal to 0 thru 2p around the unit circle with samples taken equal distance around it (Figure 1). The CZT is capable of calculating the spectrum of a signal over an arc of the unit circle (Figure 2), in other words between two arbitrary frequencies below the sampling frequency such as 255 Hz to 1234 Hz. Although beyond the scope of this article, the CZT is not restricted to calculating the spectrum on the unit circle (Figures 3 and 4).



Figures 1-4. Unit circle representation of spectrum

The details of the CZT can be comprehended as the convolution of the sampled input and the unit circle arc coefficients defined by the user. As seen in Figure 5, two FFTs and one inverse FFT are used to calculate the CZT. Therefore, in order to quickly determine the CZT, the number of points sampled must be a power of two. However, the CZT code provided will zero-pad the input samples to a power of two; therefore, the user is not restricted to this rule. If the CZT is being used in real-time spectrum analysis, prior knowledge of the following will aid in the speed of computing the CZT:

1) What is the sampling frequency of the input?
2) How many input samples are there?
3) What is the start frequency of the bandwidth of interest?

4) What is the stop frequency of the bandwidth of interest?
5) What kind of resolution is desired between the start and stop frequency?

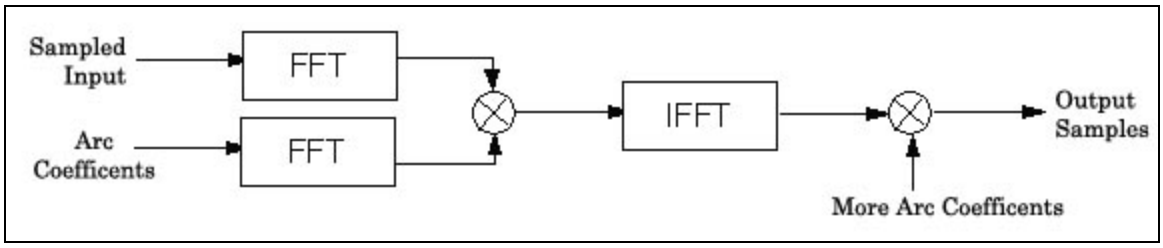


Figure 5. Data flow through CZT

Most likely the sampling frequency and number of input samples will not change during the use of the application. However, if the start and stop bands and output resolution were fixed, the arc coefficients could be precomputed at initialization of the program, thereby speeding up the computation of the CZT at the expense of flexibility.

Sampling Frequency and Input Samples

The sampling frequency usually depends on the application.
For example:

- 8000 Hz – telephony standard
- 16000 Hz – G.722 audio compression standard for video teleconferencing
- 32000 Hz – used in digital radio
- 44100 Hz – CD quality audio

The CZT uses the sampling frequency as a reference to determine where the start and stop bands are located on the unit circle. The resolution, also known as bin size or block size, is determined by dividing the sampling frequency bandwidth by the number of input samples. For instance, if the sample frequency bandwidth is 44.1 kHz and 1024 samples were recorded in 23 milliseconds, then the resolution would be 44100/1024 = 43 Hz. Increasing the number of samples to 2048 recorded over 46 milliseconds would offer a resolution of 44100/2048 = 21.5 Hz.

Start and Stop Frequency and Frequency Resolution

Many times engineers are interested in a small range of frequencies and over-sample to satisfy the nyquist criteria. Having over-sampled, the bins below and above the bandwidth of interest do not aid in creating a clear picture of the desired frequencies (Figure 6).

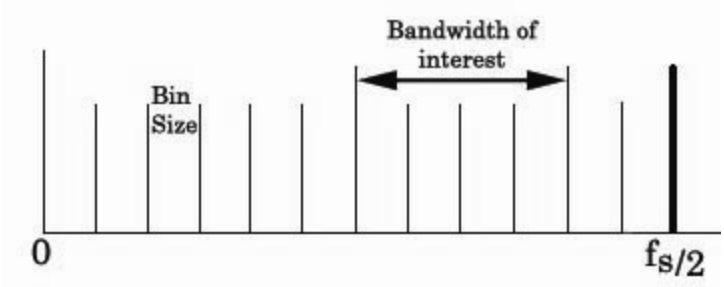


Figure 6. Resolution using FFT

The CZT has the advantage that the user cannot only define the start and stop frequencies but also set the number of bins contained by that bandwidth.

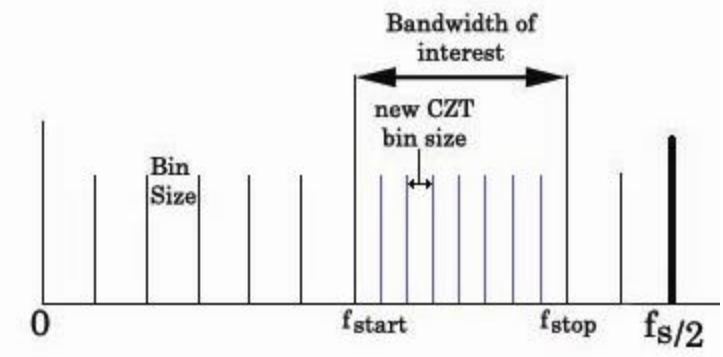


Figure 7. CZT resolution superimposed over FFT

The significance of resolution can be seen if two frequencies appear between the set bin sizes. Suppose 128 samples at 8000 Hz are taken of an audio signal. The resolution is calculated to be 8000/128 = 62.5 Hz, and three of the bins are calculated to be 437.5 Hz, 500 Hz, and 562.5 Hz respectively.

$$62.5 * N = 437.5, 500, 562.5 \text{ Hz} \quad (\text{where } N = 7,8,9)$$

If two tones were acquired at 470 Hz and 530 Hz between the established bins, the spectrum output would appear similar to the shaded area in Figure 8. Analysis of this spectrum would indicate a tone at 500 Hz has been detected.

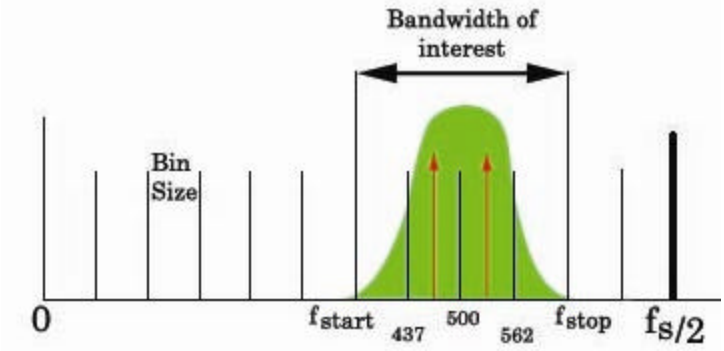


Figure 8. Sample spectrum using FFT

Using the CZT with a start and stop frequencies of 100 Hz and 1000 Hz respectively, 128 output samples would give a resolution of 7 Hz.

$$\frac{(1000\,Hz - 100\,Hz)}{128} = 7.03\,Hz$$

With a higher resolution of 7 Hz per bin, the two tones at 470 Hz and 530 Hz can clearly be resolved from each other with little chance of error, unless the user desires to resolve two tones closer than 7 Hz apart.

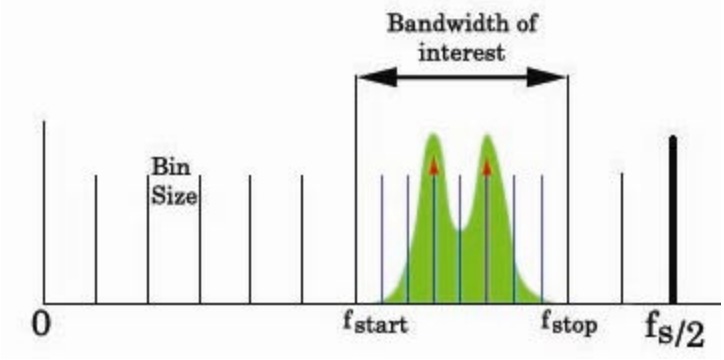


Figure 9. Sample spectrum using CZT

To summarize, the CZT requires the sampling frequency and number of input samples for reference. This reference coupled with the start and stop frequencies are used to determine the arc along the unit circle that the sampled inputs will be convolved with. The CZT also requires the number of output samples to establish the resolution between the start and stop frequencies.

CZT Algorithm Compared to FFT

Having expended so much effort on increasing the speed and accuracy of the FFT, why would there be a need for anything else? The answer lies in the sacrifices made to the FFT to achieve speed. One such limitation is the power-of-two rule, requiring the number of input samples to be a power of two (i.e., 128, 256, 512). While in itself this may not seem important, coupled with the fact that sampling frequencies are often dictated by the sampling hardware to common sampling frequencies, such as 44.1 kHz, 22.050 kHz, 16 kHz, 11.025 kHz, and 8 kHz, resolution, choices start becoming severely limited.

Table 1. Possible bins sizes with common sampling frequencies and power-of-two number of samples.

44100 / 1024 = 43 Hz	22050 / 1024 = 21 Hz	16000 / 1024 = 16 Hz	11025 / 1024 = 11 Hz	8000 / 1024 = 8 Hz
44100 / 512 = 86 Hz	22050 / 512 = 42 Hz	16000 / 512 = 32 Hz	11025 / 512 = 22 Hz	8000 / 512 = 16 Hz
44100 / 256 = 172 Hz	22050 / 256 = 84 Hz	16000 / 256 = 64 Hz	11025 / 256 = 44 Hz	8000 / 256 = 32 Hz

Add to this scenario the nyquist criteria, requiring the sampling frequency to be twice the highest frequency being sampled. Then, choosing to lower sampling frequencies for

better resolution is no longer a viable option. A clever engineer would simply increase the number of samples being taken. However, this solution quickly gets out of hand.

Every increase in samples collected must be twice that previously sampled in order to satisfy the power-of-two rule (Table 2). The sampling frequency and number of samples acquired define the sampling time. For example, to capture 1024 samples with a sampling rate of 8000 Hz requires (1/8000)*1024 = 0.128 seconds. Stepping up to the next power of two, the sampling time required is (1/8000)*2048 = 0.256 seconds. The pattern is: to attain twice the resolution, twice the number of samples is required, which takes twice the time to acquire.

Table 2. Valid Power of Two Samples

2^{10} =	1024
2^{11} =	2048
2^{12} =	4056
2^{13} =	8192
2^{14} =	16384
2^{15} =	32768
2^{16} =	65536

Compared to the FFT, the CZT is much more flexible. Given the sampling rate and the number of samples taken, the resolution can be tailor made by means of adjusting the start and stop frequencies, and the number of output samples (bin size). However, it is ironic to find that to make spectrum analysis more flexible, the CZT uses the FFT itself. Therefore, the faster the FFT becomes, the more efficient the CZT becomes. In spite of this, the CZT will never be faster than the FFT.

CVT Algorithm Compared to Goertzel Algorithm

In the September 2002 issue of Embedded Systems Programming, the Goertzel algorithm was featured as a faster method of pitch detection than the FFT. The amazing speed of the Goertzel comes from focusing on detecting the amplitude and phase of a single frequency. However, although the Goertzel is not limited by the power-of-two rule, the same sampling and bin size considerations of the FFT apply. Not bounded by the power-of-two rule, the Goertzel is more flexible at adjusting bin sizes. Assuming that sampling rates are dictated by hardware (i.e., 44.1 kHz, 8000 Hz), the user needs only change the number samples to achieve the bin size desired.

$$\#\,samples = \frac{k * sampling\_frequency}{target\_frequency} \qquad k = \{1, 2, 3, 4\ldots.\}$$

To determine the number of samples to obtain in the Goertzel algorithm, find the smallest integer value of *k* that provides an integer value number of samples (i.e., 205, 301). If an integer value of *k* cannot be found and an approximate resolution is used, be aware that an accurate magnitude measurement may be questionable. If the Goertzel algorithm is

used for tone detection, this may not be an issue. Be aware that a larger number of samples requires a longer time to sample. Therefore, for real-time applications the sampling duration should be minimized.

Although the Goertzel algorithm is better suited to tone detection, it is possible to use it for spectrum analysis by looking at a range of tones to create a spectrum. To do this, the user should change the target frequency and sweep it across the spectrum of interest. This method is acceptable for very small bandwidths but becomes much slower as the number of frequencies being swept across grows. The FFT and CZT, on the other hand, are much more time efficient at calculating the spectrum of larger bandwidths.

Compared to the FFT, the Goertzel algorithm is more flexible. Given the sampling rate and the target frequency, the number of samples acquired can easily be adjusted to obtain the ideal bin size. The advantages of speed and flexibility are spoiled by its singular focus on a solitary frequency. The CZT in contrast, offer more choices than the Goertzel algorithm in terms of resolution selection, but at the price of speed. In addition, the CZT output resolution can be tailor made by adjusting the start and stop frequencies, and the number of output samples (bin size) over a range of frequencies.

Conclusion

With a better understanding of Bluestein's Fast Fourier Transform (FFT), commonly called the Chirp-Z Transform (CZT), the widespread and enduring use of the FFT begs the question, "Is the need for more resolution and flexibility worth the computational time delay?" The CZT is a new tool in the engineer's arsenal, when engineers are challenged to detect tones, frequencies, signatures, or some telltale sign that signifies a condition. Similarly, the Goertzel algorithm, described in Embedded Systems Programming, September 2002, is a powerful tool, which finds specific frequencies faster than the FFT. The pertinent design question at this point is, "Is the single frequency detection of the Goertzel algorithm too narrow and the FFT too wide for the application in question?"

The CZT is capable of analyzing a range of frequencies in order to record frequency response measurements, match voice patterns, or display spectrum information. It gives the engineer the flexibility to specify bandwidth and resolution, and outputs real and imaginary frequency components from which the magnitude and phase can be computed.

This work was supported by the U.S. Department of Energy, National Nuclear Security Administration Nevada Site Office, under Contract No. DE-AC08-96NV11718.
DOE/NV/11718--833.