Chirp-z Transform*

Douglas L. Jones

Abstract

Efficient scheme for computing samples of the z-transform. (Important special case: DFTs)

Let $z^k = AW^{-k}$, where $A = A_o e^{i\theta_o}$, $W = W_o e^{-(i\phi_o)}$.
We wish to compute M samples, $k = [0, 1, 2, \dots, M-1]$ of

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

*Version 1.4: Jun 21, 2004 12:37 pm -0500
†http://creativecommons.org/licenses/by/1.0

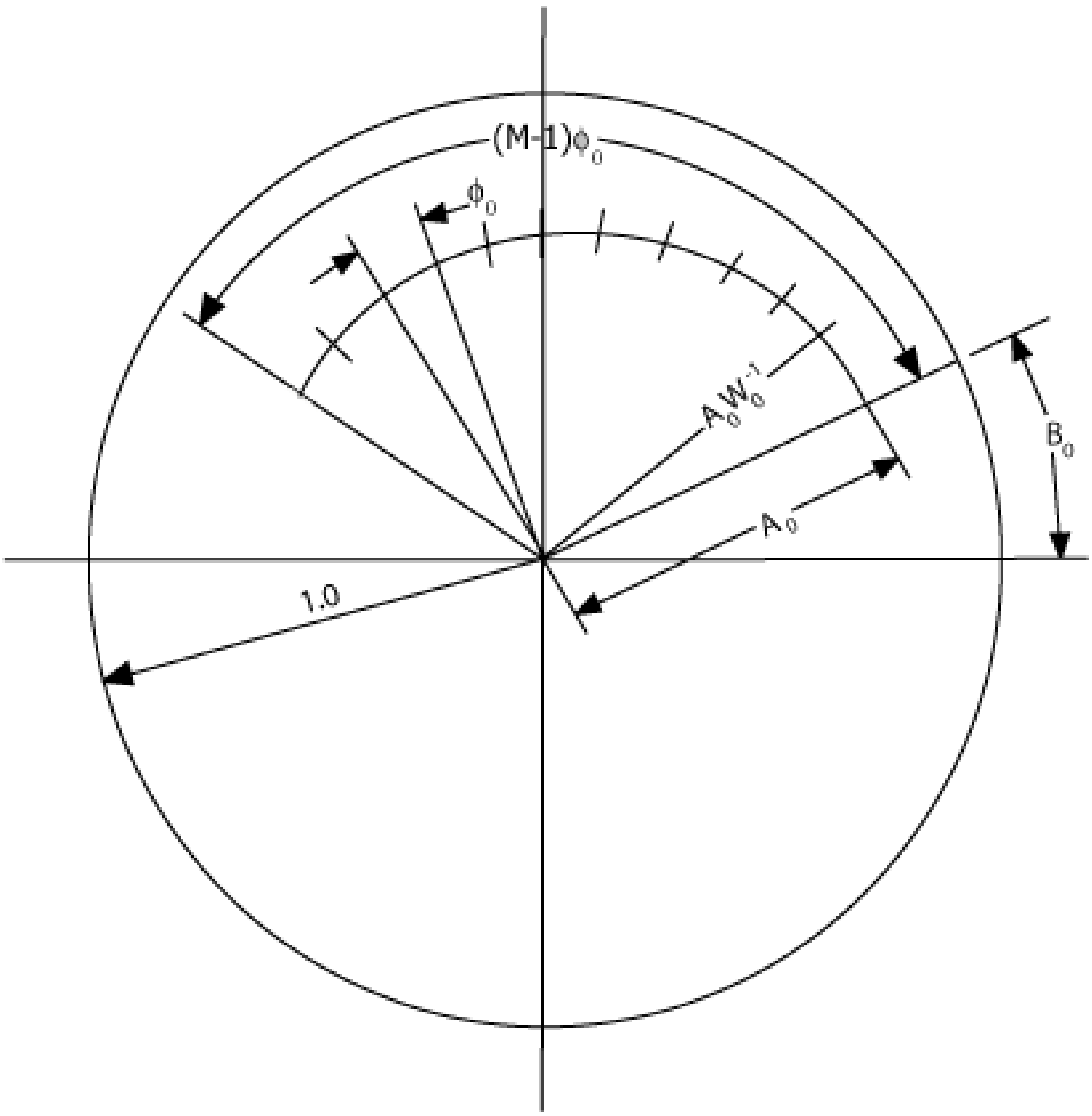

Figure 1

Note that $\left((k-n)^2 = n^2 - 2nk + k^2\right) \Rightarrow \left(nk = \frac{1}{2}\left(n^2 + k^2 - (k-n)^2\right)\right)$, So

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

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

Thus, $X(z_k)$ can be compared by

1. Premultiply $x(n)$ by $A^n W^{\frac{n^2}{2}}$, $n = [0, 1, \dots, N-1]$ to make $y(n)$
2. Linearly convolve with $W^{\frac{-(k-n)^2}{2}}$
3. Post multiply by to get $W^{\frac{k^2}{2}}$ to get $X(z_k)$.

1. (list, item 1, p. 2) and 3. (list, item 3, p. 3) require N and M operations respectively. 2. (list, item 2, p. 3) can be performed efficiently using fast convolution.

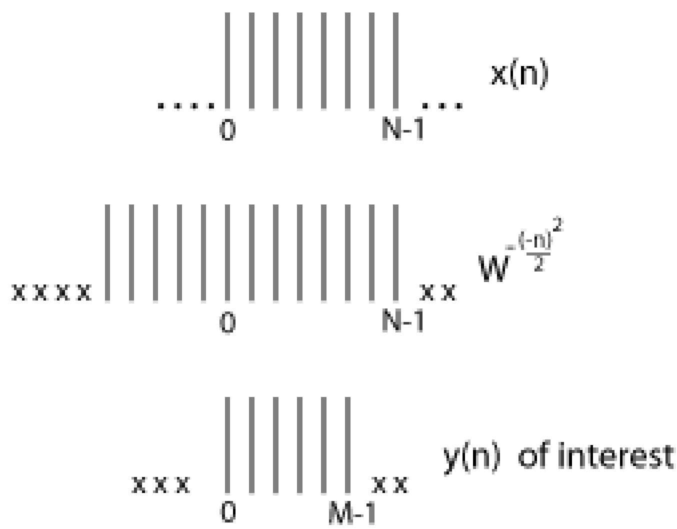

Figure 2

$W^{-\frac{n^2}{2}}$ is required only for $-(N-1) \le n \le M-1$, so this linear convolution can be implemented with $L \ge N + M - 1$ FFTs.

NOTE: Wrap $W^{-\frac{n^2}{2}}$ around L when implementing with circular convolution.

So, a weird-length DFT can be implemented relatively efficiently using power-of-two algorithms via the chirp-z transform.

Also useful for "zoom-FFTs".