SANDIA REPORT

SAND 2005-7084
Unlimited Release
Printed November 2005

Chirp Z-Transform Spectral Zoom Optimization with MATLAB®

Grant D. Martin

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND 2005-7084
Unlimited Release
Printed November 2005

Chirp Z-Transform Spectral Zoom Optimization with MATLAB®

Grant D. Martin
Embedded Radar Processing Department

Sandia National Laboratories
PO Box 5800
Albuquerque, NM 87185-1330

ABSTRACT

The MATLAB language has become a standard for rapid prototyping throughout all disciplines of engineering because the environment is easy to understand and use. Many of the basic functions included in MATLAB are those operations that are necessary to carry out larger algorithms such as the chirp z-transform spectral zoom. These functions include, but are not limited to mathematical operators, logical operators, array indexing, and the Fast Fourier Transform (FFT). However, despite its ease of use, MATLAB's technical computing language is interpreted and thus is not always capable of the memory management and performance of a compiled language. There are however, several optimizations that can be made within the chirp z-transform spectral zoom algorithm itself, and also to the MATLAB implementation in order to take full advantage of the computing environment and lower processing time and improve memory usage. To that end, this document's purpose is two-fold. The first demonstrates how to perform a chirp z-transform spectral zoom as well as an optimization within the algorithm that improves performance and memory usage. The second demonstrates a minor MATLAB language usage technique that can reduce overhead memory costs and improve performance.

ACKNOWLEDGEMENTS

This work was funded by the US DOE Office of Nonproliferation & National Security (NNSA), Office of Research and Development, Proliferation Detection Program Office (NA-22), under the Advanced Radar System (ARS) project.

Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Sections of this document include the use of MATLAB to provide documentation on function usage, coding examples, and the chirp z-transform function. MATLAB is a registered trademark of The MathWorks, Inc.; and excerpts from the MATLAB “czt” function and MATLAB documentation are reprinted with permission by The MathWorks Inc.

For MATLAB product information, please contact:

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 USA
Phone: 508-647-7000
E-mail: info@mathworks.com
Web: www.mathworks.com

CONTENTS

Foreword

This research was conducted by the Embedded Radar Processing group at the Sandia National Laboratories site in Albuquerque, New Mexico. This Sandia group develops Synthetic Aperture Radars (SAR's) to produce quality, high-resolution images in real-time for several operating modes. These SAR systems continue to push the envelope for the technology such that real-time processing demands are becoming increasingly difficult. Therefore, any effort to optimize the image formation process is necessary.

One approach to accomplish SAR image formation utilizes a spectral zoom operation with a chirp z-transform. Consequently, the optimizations described within this document resulted from an effort to reduce processing time of the computationally intensive portion of the SAR image formation algorithm, namely the spectral zoom. The optimizations now reside in two different SAR image formation algorithms. The first is a ground processing algorithm that utilizes MATLAB and the second is the recently developed Sandia SAR platform MiniSAR, in which Michael Holzrichter incorporated and adapted the mathematical portion of the optimization technique to demonstrate real-time image formation during flight tests.

1. Introduction & Background

The spectral zoom operation with digital signal processing has many purposes. One such purpose is to look at a signal's peak response within a smaller band of frequencies than the total Nyquist frequency spectrum. This smaller range can be used to measure signal characteristics such as transition slope, resolution, and signal bandwidth. With the standard Discrete Fourier Transform, these measurements critically depend upon the accuracy of the interpolated zoom. Typical frequency interpolation is accomplished by appropriately zero padding the original input sequence. This allows for a finer point distribution over the entire set of Nyquist limited frequencies. However, large amounts of zero padding which are sometimes required for very fine frequency spacing, can increase computational burden considering that the spectrum of interest may only be a fraction of the total nyquist spectrum. Also, in cases where it is necessary to calculate a set of values at precise frequencies, zero padding may not provide the correct output spacing without further post interpolation. There is however another technique that exists to perform a spectral zoom operation and compensate for the deficiencies inherent with zero padding. This technique is the chirp z-transform spectral zoom.

The chirp *z*-transform, or CZT, computes the *z*-transform along spiral contours in the *z*-plane for an input sequence.

$$CZT\left(x[n]\right) = \sum_{n=0}^{N-1} x[n] \cdot z_k^{-n} \qquad (1)$$

Unlike the DFT, the CZT is not constrained to operate along the unit circle, but can evaluate the *z*-transform along contours described by

$$z_k = A \cdot W^{-k}, \quad k = 0,\dots, M-1 \qquad (2)$$

where *A* is the complex starting point, *W* is a complex scalar describing the complex ratio between points on the contour, and *M* is the length of the transform. For example, the CZT can be used to evaluate points along the generic spiral given in Figure 1:

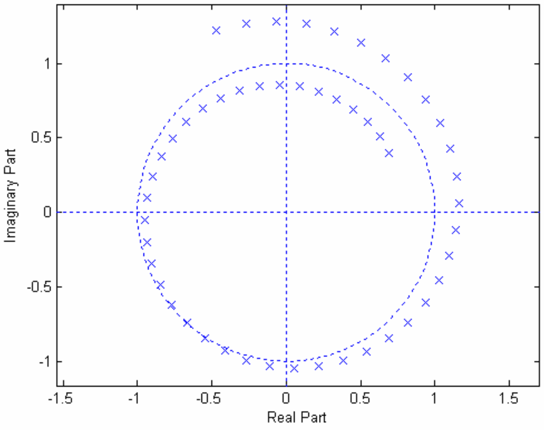

Figure 1: Generic Z-Plane Spiral Contour

For the spectral zoom operation, we limit the chirp z-transform to the region of the unit circle, because we are only interested in some portion of the frequency spectrum. This means that both *A* and *W* are both complex exponential scalars with unity magnitude. In order to better understand how the spectral zoom operation is set up with this respect, all one needs to know is that *A* is the complex starting point (on the unit circle), *W* is a complex scalar describing the complex ratio between points on the contour (point spacing), and *M* is the length of the transform. So, we can define a zoom region that begins at *A*, and ends at (*M*-1)**W*. This zoom region has uniformly spaced points in a subsection of the unit circle, the whole unit circle, or it can even rotate around the unit circle more than 1 time. This is respectively equivalent to a subsection of the Nyquist frequency spectrum, all of the Nyquist frequency spectrum, or a partially duplicated set of the Nyquist frequency spectrum. The first option is the case of interest and is convenient because the spectral zoom can start at any arbitrary point and end at any arbitrary point, with a density defined by the user when the chirp z-transform is used. This is a more direct and exact method of calculating a spectral zoom as opposed to zero padding a DFT and interpolating to desired point locations.

In order to proceed into the MATLAB optimization of the spectral zoom, it is first necessary to look at how the chirp-z transform is evaluated efficiently; and second it is necessary to review how the czt is implemented in MATLAB. Once these are outlined, the discussion of optimization and MATLAB coding optimization can be discussed.

CZT Algorithm

$$X[m] = CZT\left(x[n]\right) = \sum_{n=0}^{N-1} x[n]\cdot z^{-n}\Bigg|_{z=A\cdot W^{-m}} \tag{3}$$

$$\sum_{n=0}^{N-1} x[n]\cdot z^{-n} = \sum_{n=0}^{N-1} x[n]\cdot\left(A\cdot W^{-m}\right)^{-n} \tag{4}$$

$$= \sum_{n=0}^{N-1} x[n]\cdot A^{-n}\cdot W^{mn} \tag{5}$$

Note: $mn = \dfrac{m^2+n^2}{2} - \dfrac{(m-n)^2}{2}$ (6)

$$= \sum_{n=0}^{N-1} x[n]\cdot A^{-n}\cdot W^{\frac{m^2+n^2}{2}-\frac{(m-n)^2}{2}} \tag{7}$$

$$= W^{\frac{m^2}{2}}\cdot\sum_{n=0}^{N-1} x[n]\cdot A^{-n}\cdot W^{\frac{n^2}{2}}\cdot W^{-\frac{(m-n)^2}{2}} \tag{8}$$

$$= W^{\frac{m^2}{2}}\cdot\left[\left(x[n]\cdot A^{-n}\cdot W^{\frac{n^2}{2}}\right) * W^{-\frac{n^2}{2}}\right] \tag{9}$$

Convolution Operator

$$CZT\left(x[n]\right) = W^{\frac{m^2}{2}}\cdot IFFT\left[FFT\left(x[n]\cdot A^{-n}\cdot W^{\frac{n^2}{2}}\right)\cdot FFT\left(W^{-\frac{n^2}{2}}\right)\right] \tag{10}$$

As can be seen from the above algorithm, the CZT can be implemented as a circular convolution or equivalently as 2 FFT's, 1 IFFT and 4 complex multiply's. The documentation and implementation follows:

MATLAB CZT Documentation

`(The following is provided by The MathWorks Inc.)`

czt

Chirp *z*-transform

Syntax

- `y = czt(x,m,w,a)`
- `y = czt(x)`

Description

`y = czt(x,m,w,a)` returns the chirp *z*-transform of signal `x`. The chirp *z*-transform is the *z*-transform of `x` along a spiral contour defined by `w` and a. `m` is a scalar that specifies the length of the transform, `w` is the ratio between points along the *z*-plane spiral contour of interest, and scalar a is the complex starting point on that contour. The contour, a spiral or "chirp" in the *z*-plane, is given by

- `z = a*(w.^-(0:m-1))`

`y = czt(x)` uses the following default values:

- `m = length(x)`
- `w = exp(-j*2*pi/m)`
- `a = 1`

With these defaults, `czt` returns the *z*-transform of `x` at `m` equally spaced points around the unit circle. This is equivalent to the discrete Fourier transform of `x`, or `fft(x)`. The empty matrix `[]` specifies the default value for a parameter.

If `x` is a matrix, `czt(x,m,w,a)` transforms the columns of `x`.

MATLAB Implementation - Signal Processing Toolbox

(The following is provided by The MathWorks Inc.)

```
function g = czt(x, k, w, a)

%   Author(s): C. Denham, 1990.
%              J. McClellan, 7-25-90, revised
%              C. Denham, 8-15-90, revised
%              T. Krauss, 2-16-93, updated help
%   Copyright 1988-2005 The MathWorks, Inc.
%       $Revision: 1.7 $  $Date: 2002/04/15 01:10:37 $

[m, n] = size(x); oldm = m;
if m == 1, x = x(:); [m, n] = size(x); end

if nargin < 2, k = length(x); end
if nargin < 3, w = exp(-i .* 2 .* pi ./ k); end
if nargin < 4, a = 1; end

if any([size(k) size(w) size(a)]~=1),
    error('Inputs M, W and A must be scalars.')
end

%------- Length for power-of-two fft.

nfft = 2^nextpow2(m+k-1);

%------- Premultiply data.

kk = ( (-m+1):max(k-1,m-1) ).';
kk2 = (kk .^ 2) ./ 2;
ww = w .^ (kk2);   % <----- Chirp filter is 1./ww
nn = (0:(m-1))';
aa = a .^ ( -nn );
aa = aa.*ww(m+nn);
y = x .* aa(:,ones(1,n));

%------- Fast convolution via FFT.

fy = fft(  y, nfft );
fv = fft( 1 ./ ww(1:(k-1+m)), nfft );   % <----- Chirp filter.
fy = fy .* fv(:,ones(1, n));
g  = ifft( fy );

%------- Final multiply.

g = g( m:(m+k-1), : ) .* ww( m:(m+k-1),ones(1, n) );


if oldm == 1, g = g.'; end
```

In order to use the CZT as a spectral zoom, the following example is given. This same example will also be performed with the optimization technique presented in the “Detailed Analysis” section.

Example – MATLAB CZT Documentation
`(The following is provided by The MathWorks Inc.)`

This example illustrates the use of `czt` to zoom in on a narrow-band section (100 to 150 Hz) of a filter's frequency response.

First design the filter:

- `h = fir1(30,125/500,rectwin(31)); % filter`

Establish frequency and CZT parameters:

- `fs = 1000; f1 = 100; f2 = 150;  % in hertz`
- `m = 1024;`
- `w = exp(-j*2*pi*(f2-f1)/(m*fs));`
- `a = exp(j*2*pi*f1/fs);`

Compute both the DFT and CZT of the filter:

- `y = fft(h,1000);`
- `z = czt(h,m,w,a);`

Create frequency vectors and compare the results:

- `fy = (0:length(y)-1)'*1000/length(y);`
- `fz = ((0:m-1)'*(f2-f1)/m) + f1;`
- `plot(fy(1:500),abs(y(1:500))); axis([1 500 0 1.2])`
- `title('FFT')`
- `figure`
- `plot(fz,abs(z)); axis([f1 f2 0 1.2])`
- `title('CZT')`

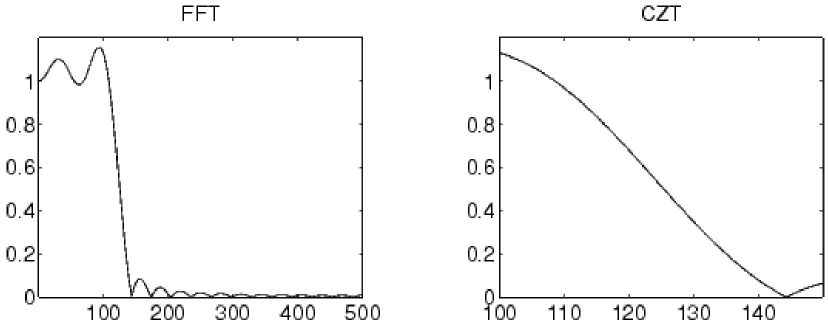

Figure 2: Plot function output for the MATLAB example

2. Overview & Summary

In using the chirp z-transform for a spectral zoom, the *W* and *A* parameters that define the z-plane contour are constrained to be scalars on the unit circle. Therefore there exists an optimization with calculating the necessary complex multiplication factors. Recall that the CZT requires

$$CZT\ \left(x[n]\right) = W^{\frac{m^2}{2}} \cdot IFFT\left[FFT\left(x[n]\cdot A^{-n}\cdot W^{\frac{n^2}{2}}\right)\cdot FFT\left(W^{-\frac{n^2}{2}}\right)\right]$$

where Both *W* and *A* are raised to a power. The optimization then is that instead of raising the passed arguments of *W* and *A* to a power, calculate the exponent as a multiply then perform the regular exponential table look up to evaluate. The benefit is that this can significantly cut down on the operations required to perform a czt. Also, if the czt is to be called repetitiously, most of the exponents can be calculated external to the loop, again significantly reducing the number of operations required. The final optimization reduces some of the overhead memory required and slightly improves performance in evaluating the czt. This comes about from performing direct operation function calls. A more detailed explanation of both optimizations is presented in the following section.

3. Detailed Analysis

CZT Optimization for Spectral Zoom

Note that in the implementation of a spectral zoom, both W and A are defined with the following MATLAB expressions:

```
        w = exp(-j*2*pi*(f2-f1)/(m*fs));
        a = exp(j*2*pi*f1/fs);
```

Both of these are exponentials and scalars, and therefore there exists a simple optimization of the CZT. Recall that the CZT requires

$$CZT\ (x[n]) = W^{\frac{m^2}{2}} \cdot IFFT\left[FFT\left(x[n] \cdot A^{-n} \cdot W^{\frac{n^2}{2}} \right) \cdot FFT\left(W^{-\frac{n^2}{2}} \right) \right]$$

Both W and A are raised to a power. The MATLAB implementation carries this out as follows:

```
        kk = ( (-m+1):max(k-1,m-1) ).';
        kk2 = (kk .^ 2) ./ 2;
        nn = (0:(m-1))';

        ww = w .^ (kk2);
        aa = a .^ ( -nn );
```

`ww` and `aa` are the W and A functions with their associated powers to be used in the CZT algorithm above. The optimization of calculating both ww and aa is a direct result of the property of raising an exponential to a power. Namely:

$$\left(e^{x}\right)^{n} = e^{x \cdot n}$$

Therefore, instead of using a function that has W and A passed as arguments and each is raised to a power, incorporate the CZT function body as part of your spectral zoom code and calculate the exponent as a vector multiplication then evaluate the exponential. The final product is given in the next example.

Example

First design the filter:

- `h = fir1(30,125/500,rectwin(31)); % filter`

Establish frequency and CZT parameters:

- `fs = 1000; f1 = 100; f2 = 150;  % in hertz`
- `m = 1024;`

```
% Begin Spectral Zoom Optimization CZT

[k, n] = size(h); oldk = k;
if k == 1, h = h(:); [k, n] = size(h); end

%------- Length for power-of-two fft.

nfft = 2^nextpow2(k+m-1);

%------- Premultiply data.

kk = ( (-k+1):max(m-1,k-1) ).';
kk2 = (kk .^ 2) ./ 2;
ww = exp(-j*2*pi*(f2-f1)/(m*fs)*kk2);
nn = (0:(k-1))';
aa = exp(-j*2*pi*f1/fs .* nn );
aa = aa.*ww(k+nn);
y = h .* aa(:,ones(1,n));

%------- Fast convolution via FFT.

fy = fft(  y, nfft );
fv = fft( 1 ./ ww(1:(m-1+k)), nfft );   % <----- Chirp filter.
fy = fy .* fv(:,ones(1, n));
z  = ifft( fy );

%------- Final multiply.

z = z( k:(k+m-1), : ) .* ww( k:(k+m-1),ones(1, n) );


if oldk == 1, z = z.'; end

% End Spectral Zoom Optimization CZT
```

Compute the DFT of the filter:

- `y = fft(h,1000);`

Create frequency vectors and compare the results:

- `fy = (0:length(y)-1)'*1000/length(y);`
- `fz = ((0:m-1)'*(f2-f1)/m) + f1;`
- `plot(fy(1:500),abs(y(1:500))); axis([1 500 0 1.2])`
- `title('FFT')`
- `figure`
- `plot(fz,abs(z)); axis([f1 f2 0 1.2])`
- `title('CZT')`

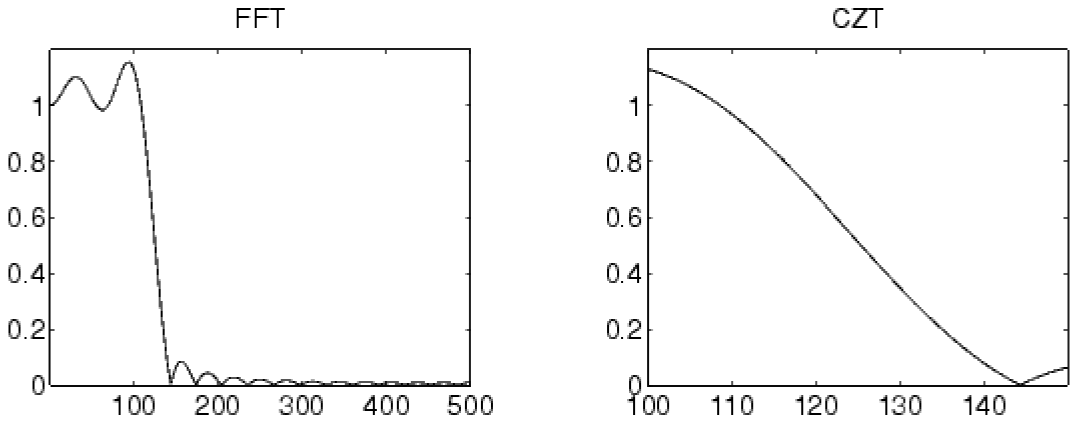

Figure 3: Plot function output for the MATLAB example with czt optimization

The optimization for this example has a 61% reduction in the total time required to run as seen from 10000 trials. However, for simple examples, like the one above, the optimization is not overtly noticeable since the running time of the code is sub second. Therefore, for simple cases it is much easier to evoke the MATLAB czt function. As stated earlier though, if the czt is to be called repetitiously, the exponents can be calculated external to the loop, again significantly reducing the number of operations required and the reduction will accumulate in each repetition making this optimization indispensable.

Other General MATLAB Optimizations

Most of the additional optimizations involve un-nesting vector operations and using the direct function call for the mathematical operators as opposed to the symbolic notation. By un-nesting vector operators the interpreter can more readily identify the operations to be carried out when moving from line to line. This will not make a large performance improvement in a function that is only called once, but when operations are continuously performed, the benefit of un-nesting commands becomes noticeable. The benefit in using direct function calls for operators as opposed to their symbolic notation is that some of the program overhead is removed for the interpreter. Again, this will not be largely noticeable in a single function call, but becomes noticeable when a function is repeatedly called.

The following example incorporates the algorithm optimization from the preceding section, as well as all the general MATLAB optimizations of un-nesting vector operations and using direct function calls.

Example

First design the filter:

- `h = fir1(30,125/500,rectwin(31)); % filter`

Establish frequency and CZT parameters:

- `fs = 1000; f1 = 100; f2 = 150;  % in hertz`
- `m = 1024;`

```
% Begin Spectral Zoom Optimization CZT
[k, n] = size(h); oldk = k;
if k == 1, h = h(:); [k, n] = size(h); end

%------- Length for power-of-two fft.
nfft = power(2,nextpow2(k+m-1));

%------- Premultiply data.

kk = transpose([(-k+1):max(m-1,k-1)]);
kk2 = times(kk,kk)./2;

wPow = times(-j*2*pi*(f2-f1)/(m*fs) , [kk2]);
ww = exp(wPow);

nn = transpose([0:(k-1)]);

aPow = times(-j*2*pi*f1/fs , [nn] );
aa = exp(aPow);

aa = times(aa , ww(k+nn));
y = times(h , aa);

%------- Fast convolution via FFT.

fy = fft(y,nfft);
fv = fft(1./[ww(1:(m-1+k))],nfft);   % <----- Chirp filter.
fy = times(fy,fv);
z  = ifft(fy);
```

```
%------- Final multiply.

z = times(z(k:(k+m-1)),ww(k:(k+m-1)));

if oldk == 1, z = transpose(z); end
% End Spectral Zoom Optimization CZT
```

Compute the DFT of the filter:

- `y = fft(h,1000);`

Create frequency vectors and compare the results:

- `fy = (0:length(y)-1)'*1000/length(y);`
- `fz = ((0:m-1)'*(f2-f1)/m) + f1;`
- `plot(fy(1:500),abs(y(1:500))); axis([1 500 0 1.2])`
- `title('FFT')`
- `figure`
- `plot(fz,abs(z)); axis([f1 f2 0 1.2])`
- `title('CZT')`

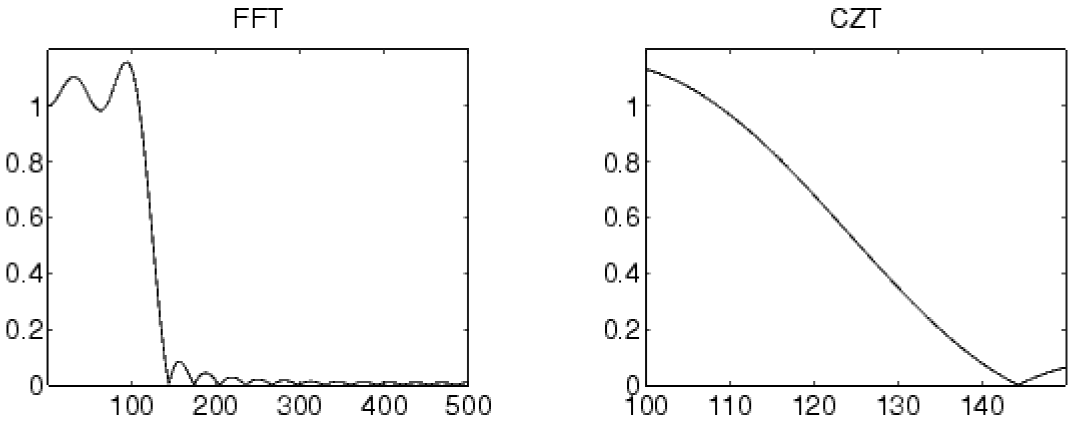

Figure 4: Plot function output for the MATLAB example with czt and general MATLAB optimizations

The optimization in this example has a 71% reduction in the total time required to run as seen from 10000 trials. This is an additional 23% improvement over the algorithm optimization alone and is noticeable as a 10% overall performance boost.

4. Conclusions

We have seen that the chirp z-transform spectral zoom is a simple algorithm to implement when done as a fast convolution with 2 FFT's, 1 IFFT and 4 complex multiply's. There are also several advantages of the algorithm over a standard zero-padded DFT. One advantage is that the czt eliminates the need for interpolation, while providing the desired linear point spacing and density over any range of the nyquist frequency spectrum. This makes it an excellent algorithm for many applications that utilize a spectral zoom such as spectrum measurements, computed tomography (CT) image reconstruction, and synthetic aperture radar (SAR) image formation. This report acknowledges the usefulness of the chirp z-transform spectral zoom and shows that a mathematical optimization can be performed to increase performance by ~61% (MATLAB algorithm optimization) over the standard chirp z-transform spectral zoom. This paper further demonstrates that additional optimizations can be performed within the MATLAB language that additionally boost performance over the original czt by 10%. The mathematical optimization, which is not limited to MATLAB, is incredibly useful when deployed into larger applications of the czt spectral zoom such as SAR image formation. However, when used in conjunction with the MATLAB language optimizations, the algorithm performance enhancements make MATLAB a practical tool in cases where it previously may not have been feasible.

This page left intentionally blank.

References

[1] Alan V. Oppenheim, Ronald W. Schafer, *Digital Signal Processing*, Prentice-Hall (1975).

[2] Oppenheim, A.V. & R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, pp. 623-628, 1989.

[3] Rabiner, L.R. and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, pp. 393-399, 1975.

[4] MATLAB Documentation and original Example is provided by The Mathworks Inc. and is available on the web at: http://www.mathworks.com/access/helpdesk/help/toolbox/signal/czt.html

DISTRIBUTION

Unlimited Release

1	MS 0509	M. W. Callahan	5300
1	MS 0529	B. L. Remund	5340
1	MS 0519	B. L. Burns	5340
1	MS 0503	T. J. Mirabal	5341
1	MS 0519	W. H. Hensley	5342
1	MS 0519	T. P. Bielek	5342
1	MS 1330	A. W. Doerry	5342
1	MS 0519	D. Harmony	5342
1	MS 0519	J. A. Hollowell	5342
1	MS 0529	S. S. Kawka	5342
1	MS 0519	M. S. Murray	5342
1	MS 0519	B. G. Rush	5342
1	MS 0519	D. G. Thompson	5342
1	MS 1330	K. W. Sorensen	5345
1	MS 1330	J. A. Bach	5345
1	MS 0529	B. C. Brock	5345
1	MS 1330	D. F. Dubbert	5345
1	MS 1330	G. R. Sloan	5345
1	MS 1330	S. M. Becker	5348
1	MS 0519	S. M. Devonshire	5348
1	MS 1330	M. W. Holzrichter	5348
1	MS 1330	O. M. Jeromin	5348
1	MS 1330	G. D. Martin	5348
1	MS 0519	B. E. Mills	2348
1	MS 1330	P. G. Ortiz	5348
1	MS 1330	D. C. Sprauer	5348
1	MS 1330	A. D. Sweet	5348
1	MS 0519	M. E. Thompson	5348
1	MS 0519	L. M. Wells	5354
1	MS 0519	D. L. Bickel	5354
1	MS 0519	J. T. Cordaro	5354
1	MS 0519	J. DeLaurentis	5354
1	MS 1330	A. E. Mihalik	5354
1	MS 0311	F. M. Dickey	2616
2	MS 0899	Technical Library	4536
1	MS 9018	Central Technical Files	8945