ELECTRONOTES
1016 Hanshaw Road
Ithaca, NY 14850

APPLICATION NOTE NO. 425

October 10, 2015

INVERTING A DTFT ?

INTRODUCTION

Three popular transforms are the z-Transform of a length-N time sequence x(n) as given by:

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

the Discrete-Time Fourier Transform given by:

$$X\left(e^{j\omega T}\right) = \sum_{n=0}^{N-1} x(n) e^{-jn\omega T} \quad (all\ \omega) \qquad DTFT \qquad (2)$$

and the Discrete Fourier Transform (DFT or FFT) given by:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(\frac{2\pi}{N})nk} \quad (k = 0 \ldots N-1) \quad (T = 1) \qquad DFT = FFT \qquad (3)$$

The development here is that the z-transform is evaluated on the unit circle in the z-plane giving the DTFT, and the DTFT is then evaluated at N equally space frequencies on the unit circle giving he DFT. Note that the DTFT is probably the least known of the three – at least until we consider that if we substitute h for x (H for X) we are talking “FILTER” and we recognize the DTFT as a “frequency response” of the filter. Each of the three transforms has an inversion [1]. Here we will mention the inverse DFT:

$$x(n) = \left(\frac{1}{N}\right) \sum_{k=0}^{N-1} X(k) e^{j\left(\frac{2\pi}{N}\right)nk} \quad (n = 0 \ldots N-1) \qquad Inverse\ DFT = FFT \qquad (4)$$

Our reason for comparing (3) and (4) is their near-perfect symmetry and the fact that both represent N independent linear equations in N unknowns. A matrix inversion will solve these. In the case of the DFT, the FFT is a fast algorithm to the solution. Note that for the DFT, it is the case (and a requirement) that the samples (for forward or inverse) are equally spaced. What happens if we drop this requirement and agree to bring back the matrix?

TWO VIEWS OF THE DTFT EQUATION

The DTFT equation is valid for all ω, but since this is a discrete time device, we think in terms of values of ω between 0 and π as being of the most interest. But even though time is discrete, ω is continuous. Let's suppose we have in mind a particular frequency ω_1. Equation (2) is certainly valid for our choice of ω_1:

$$X\left(e^{j\omega_1 T}\right) = \sum_{n=0}^{N-1} x(n)e^{-jn\omega_1 T} \qquad (5)$$

This shows the computation of the DTFT based on N values of x(n). We can do this for one value of ω or for as many value of ω as we desire. It is a computation of the "spectrum" or frequency-domain description of the time sequence x(n). As said above, this is much more familiar if we substitute h(n) for x(n) and $H(e^{j\omega})$ for $X(e^{j\omega})$. [I guess we should always write out the complete notations $H(e^{j\omega})$ and $X(e^{j\omega})$, although H(ω) and X(ω) are often seen as a shortcut, without confusion.] So $X(e^{j\omega})$ is in a familiar sense the frequency description of x(n) while the "filter case" has $H(e^{j\omega})$ as the frequency response of h(n). Mathematically exactly the same notion. Thus equation (5) is used to compute the DTFT. In Matlab a convenient function ***freqz*** can compute the DTFT. Often a DFT=FFT program can also be used, quite possibly involving zero-padding, as will be discussed below. Thus the forward version of equation (5) is out first of two views.

When we consider "inverting" equation (5), we in turn get two interpretations of the inversion. We want x(n) from $X(e^{j\omega})$. The approach we must choose likely depends on how well (in what form) we know $X(e^{j\omega})$.

For example, if we are doing filters, we might well have in mind something like a low-pass filter where we specify a desired (ideal) passband and stopband. That is, a continuous function on ω=0 to ω=π (quite likely both magnitude and phase). In such a case, the direct mathematical inversion of the continuous frequency function is in order [2].

$$h(n) = \left(\frac{1}{2\pi}\right) \int_{-\pi}^{\pi} H(e^{j\omega})\, e^{jn\omega} d\omega \qquad (6)$$

where we have used the filter notations h and H. This is nothing more than inverting a "Fourier Series" with the roles of time and frequency reversed.

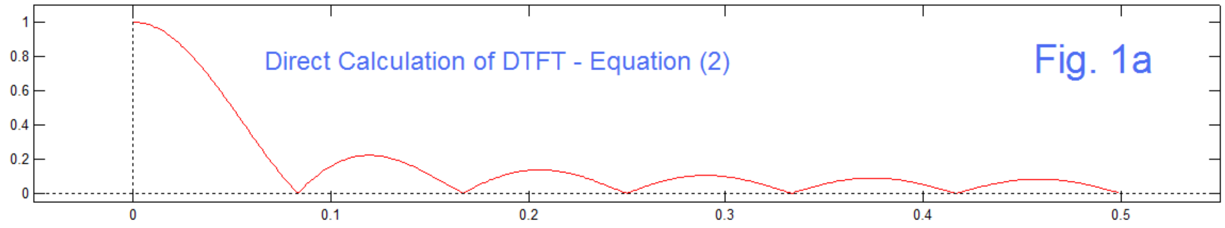
Equation (6) (using x and X) does us no good if we don't have an expression for $X(e^{j\omega})$. In such a case, perhaps we only have some isolated samples of $X(e^{j\omega})$. That is, we have perhaps samples for frequencies ω_0, ω_1, ω_2, …. ω_{N-1}. We can then write N versions of equation (5), each one being one equation in N unknowns [x(n) are unknown] for a total of N equations in N unknowns. If we have fewer equations than we need, or more, some least square fits can be considered. Solving for N unknowns from N equations is the matrix inversion suggested. Although not required, if the N frequency samples are equally spaced on 0 to π, the inversion can be by DFT=FFT.

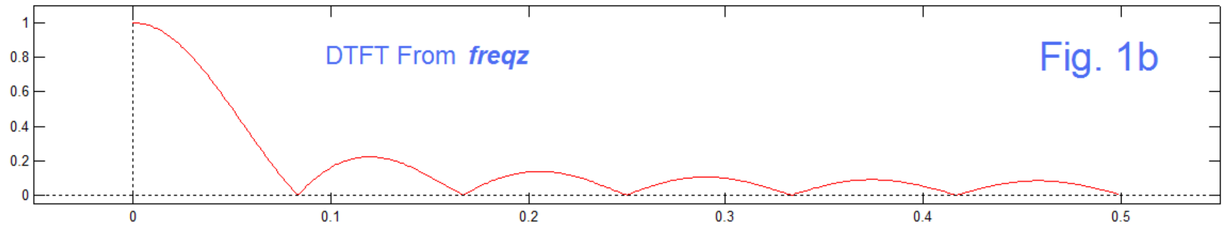
TEST CASE

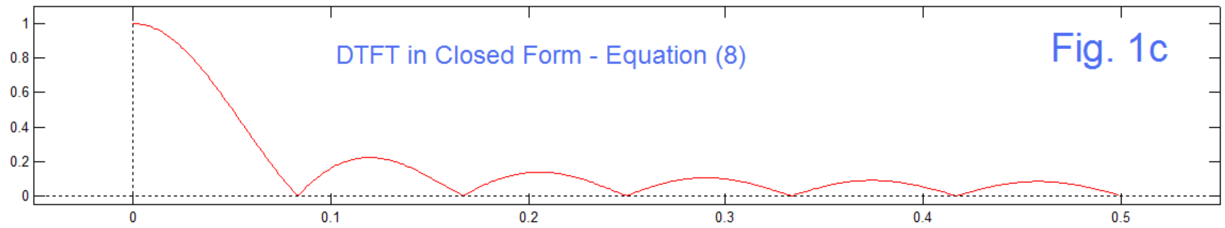
Here we need a test DTFT which we manipulate. We will take this to be the DTFT of a length-12 moving average filter. Thus the impulse response is 12 values of 1/12 in succession: h(n) = (1/12) for n=0 to 11 and is zero everywhere else. We know at least three ways of calculating the DTFT from these h(n). First, we can use equation (2) directly. Second, we can use Matlab's **freqz** function. Thirdly we can write a closed form expression for the result. Fig. 1a and Fig. 1b show the magnitude responses for methods 1 and 2. We need to see the printouts of the real and imaginary parts to be certain the results are identical (see code below).

The closed form (method 3) begins by writing the transfer function using well documented [2, 3] manipulations**:**

$$H(z) = \frac{\left(\frac{1}{12}\right) z^{-\frac{11}{2}} [z^6 - z^{-6}]}{z^{1/2} - z^{-1/2}} \qquad (7)$$







and substitutine z = $e^{j\omega}$ using the Euler relationship to arrive at trig functions – in this case the well-known “periodic sync” or “aliased sinc”**:**

$$H\left(e^{j\omega}\right) = \frac{\left(\frac{1}{12}\right) e^{-(\frac{11}{2})j\omega}\ Sin(6\omega\ )}{Sin(\frac{\omega}{2})} \tag{8}$$

```
% AN425.m
% THREE WAYS OF COMPUTING DTFT
% Direct Computation of H - Equation (1) of AN-425
h=ones(1,12)
Hdc=zeros(1,500);
for kk=1:500
   w=(kk-1)*pi/500;
   for n=0:11
          Hdc(kk)=Hdc(kk)+h(n+1)*exp(-j*n*w);
   end
end
Hdc=(1/12)*Hdc;
%
% Using freqz
Hfz=(1/12)*freqz(h,1,500).';
% Using Closed Form
w=[0:.001:.499]*(2*pi)
Hcf=(1/12)*exp(-(11/2)*j*w).*sin(6*w)./sin(w/2);
Hcf(1)=1;
%display errors
e1=max(abs(Hdc-Hfz))
e2=max(abs(Hdc-Hcf))
e3=max(abs(Hfz-Hcf))
% Plot
figure(1)
subplot(311)
   plot([0:.001:.499],abs(Hdc),'r')
   hold on
   plot([-.05 .55],[0 0],'k:')
   plot([0 0],[-.1 1.2],'k:')
   hold off
   axis([-.05 .55 -0.05 1.1])
   title('Direct Calculation')
   subplot(312)
   plot([0:.001:.499],abs(Hfz),'r')
   hold on
   plot([-.05 .55],[0 0],'k:')
   plot([0 0],[-.1 1.2],'k:')
   hold off
   axis([-.05 .55 -0.05 1.1])
   title('Matlab freqz')
   subplot(313)
   plot([0:.001:.499],abs(Hcf),'r')
   hold on
   plot([-.05 .55],[0 0],'k:')
   plot([0 0],[-.1 1.2],'k:')
   hold off
   axis([-.05 .55 -0.05 1.1])
   title('Closed Form')
```

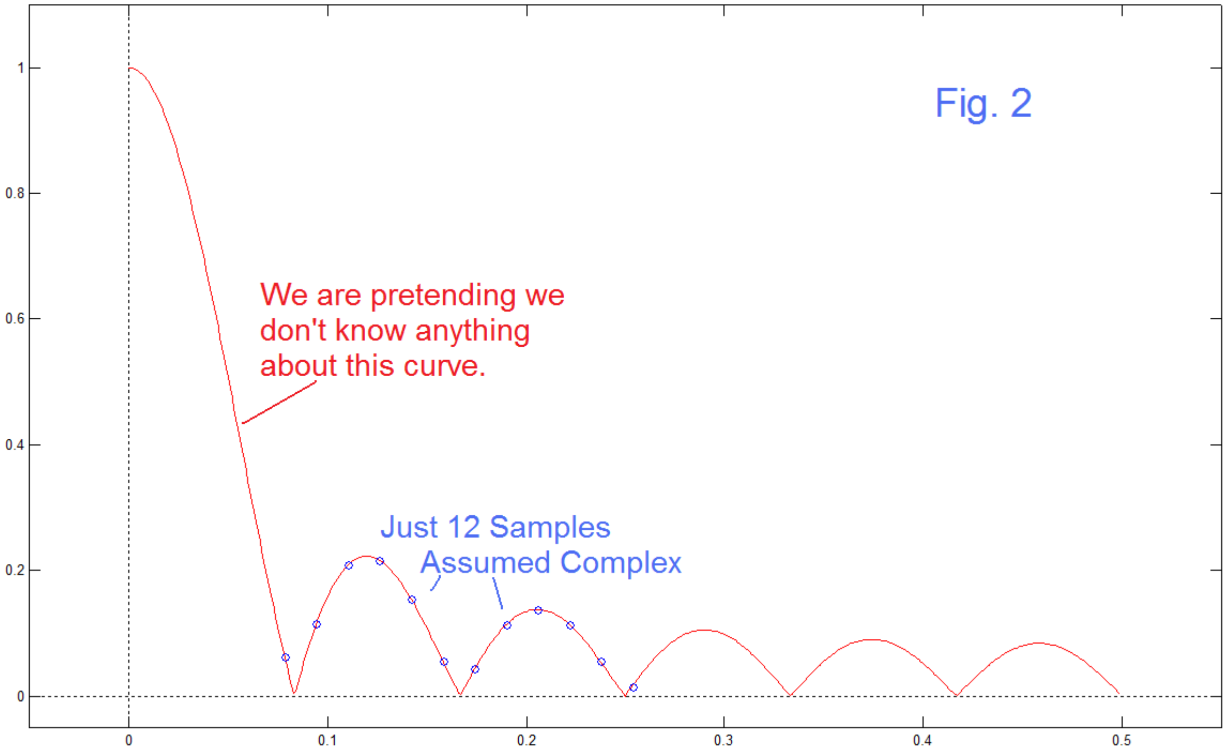
Fig. 1c indeed shows the magnitude plot using equation (8). The code used is printed on page 4 for clarity. Note that the printouts list the three errors e1, e2, and e3 is being in the 10^{-15} range (roundoff only). Thus we have a well defined test object and some practice with our mathematics.

SETTING UP AND INVERTING THE EQUATIONS

Here we are pretending that we have done nothing above that involved actually knowing that we had a length-12 moving average. We just have some data – say 12 frequency samples each of which can be written in terms of equation (5) as**:**

$$X\left(e^{j\omega_k}\right) = \sum_{n=0}^{N-1} x(n)e^{-jn\omega_k} \tag{9}$$

where we have set T=1 so that frequencies ω are on an interval of 0 to π. Here we will take samples starting at π/8 and proceeding at intervals of 0.1. These samples are found using equation (8) and appear as in Fig. 2, but keep in mind that the samples are complex and we are disclaiming we know anything about equation (8) or Fig. 2.



We now have chosen 12 frequencies and have 12 corresponding (complex) values for the DTFT and can write 12 equations in 12 unknown values of x(n). So we can calculate the matrix elements and invert to equations. This is shown in the code on the next page.

```
% Form/Invert Matrix
   k=1:12
   w=pi/8+0.1*k
   X=(1/12)*exp(-(11/2)*j*w).*sin(6*w)./sin(w/2)
   figure(2)
   plot([0:.001:.499],abs(Hcf),'r')
   hold on
   plot(w/(2*pi),abs(X),'bo')
   plot([-.05 .55],[0 0],'k:')
   plot([0 0],[-.1 1.2],'k:')
   hold off
   axis([-.05 .55 -0.05 1.1])
   title('Closed Form with Samples')
   %
   for n=1:12
      for k=1:12
         M(k,n)= exp(-j*(n-1).*w(k));
      end
   end
   M
   x=inv(M)*X.'
```

(Most of the above code just plots Fig. 2.) The frequency samples are “stolen” from equation (8) and the matrix elements M(k,n) are the evaluation of $e^{-jn\omega_k}$ on the frequency vector ω_k for n=0 to 11. The result is that x is 12 values all equal to 1/12:

```
x =
   0.0833+ 0.0000i
   0.0833- 0.0000i
   0.0833
   0.0833+ 0.0000i
   0.0833+ 0.0000i
   0.0833+ 0.0000i
   0.0833- 0.0000i
   0.0833+ 0.0000i
   0.0833+ 0.0000i
   0.0833- 0.0000i
   0.0833+ 0.0000i
   0.0833+ 0.0000i
```

So we see that we can ”invert” from the frequency domain to the time domain in several was. If we have a closed form for $H(e^{j\omega})$, or if we have a data file or even a digitizable graph, we can possibly invert by integration as in equation (6), either analytically or numerically. In the case where we have a limited number of samples, perhaps from an isolated region of frequency, we may be able to solve N equations in N unknowns, or use a least squares procedure.

GETTING MORE VALUES OF $X(e^{j\omega})$

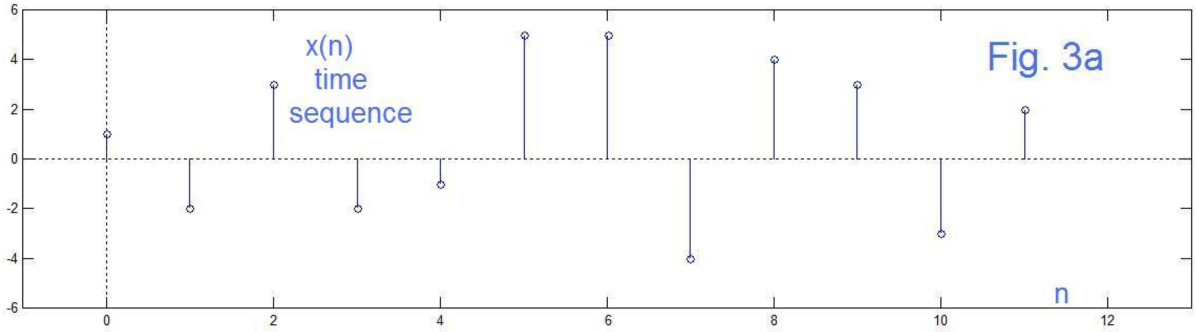
At this point, we assume we already have some length-N time sequence, x(n) so we can thus also easily obtain the DFT=FFT, also length-N, X(k), which is a corresponding frequency-domain description of x(n) as evaluated at frequencies kf_s/N where f_s is the sampling frequency associated with x(n). It may be the case that we want a frequency description at frequencies other than the available values kf_s/N. In such instances, we might think about a general interpolation of X(k) to a somewhat larger number of points by using FFT interpolation in frequency. Then from this larger set we select or reject points we want to use. Or perhaps, we might just think of evaluating the DTFT [equation (2)] at just one frequency, or at relatively few frequencies of interest. Both approaches should (must) give the same correct answers. This we will show.

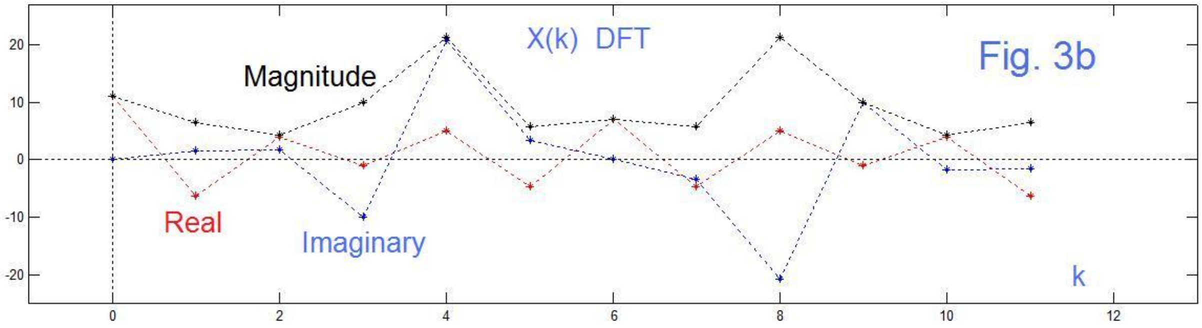
Of importance here is the question of whether or not the new spectral point or points obtained are meaningful. If we are looking for increased spectral "resolution", there will be no improvement because there is no information added. Sometimes we think we are better off, because a plot is more comfortable to understand (smoother). So there can be an issue of viewing comfort. But we don't find a sharper "bump" of spectral energy suddenly appearing inside a more gradual existing bump. Moreover, there can be cases where information other than that in x(n) can be used. For example, if we were specifically told that the samples are, perhaps, those of a single sinusoidal waveform.

On the other hand, perhaps x(n) is an impulse response like h(n) in which case the new frequency point is meaningful. It tells us how the filter will respond to the frequency of interest, even though the frequency is not a DFT frequency. (Typically we measure a frequency response with a function generator which has a continuous frequency dial.)

The conventional wisdom as to how an "FFT-Based" interpolation is achieved may well be confusing [4]. It is well established that in order to interpolate in time, we take the FFT, zero-pad it in the middle, and then take the inverse FFT of the larger sequence [5]. On the other hand, it is less known perhaps but still conventional wisdom to understand that to interpolate in frequency (erroneously sometimes said to increase resolution) you zero pad by adding a string of zeros to the end of x(n) and then take the forward FFT. As we have mentioned [4], while there are differences between the way time and frequency are "appreciated", we really can't make the case that these two, as mathematical "duals", are mathematically different [6 - note]. In fact, the zero padding is always in the middle in both the frequency and the time cases.

How can this be that putting zeros on the end is actually the same as putting them in the middle? Well – because the sequences here are periodic, any insertion of zeros will be in the middle (somewhere). Any portion we think of as above the middle can be thought of as below zero on the negative side.





AN EXAMPLE

We are badly in need of an example. Let's suppose we make up a length-12 sequence as:

```
x=[1 -2 3 -2 -1 5 5 -4 4 3 -3 2 ]
```

This we suppose is some time indexed data. It might be some actual data sequence of length exactly 12. Of it might be the length-12 impulse response of some filter. Or it might be 12 samples snipped out of a much longer sequence. Note that it is real and has no apparent patterns. We are most familiar with time sequences that are purely real. It is plotted in Fig. 3a.

Now, at the point where we propose to analyze it by a DFT(=FFT), we have given it a particular property: it is periodic with length 12. It might even have been a cycle of some longer sequence that was actually length-12 periodic. In any event, it is now! It overlaps periodically at the ends.

The sequence x(n) has a DFT X(k) as defined by equation (3). This DFT is also length-12. Unlike x(n), X(k) is complex (in general, we expect a complex result – see equation), and it has symmetry about k=0 (constant) and k=6 (half the sampling frequency). In fact, we show in Fig. 3b the real part (red), the imaginary part (blue) and the magnitude (black). It too is periodic. The dashed lines are just to avoid confusion.

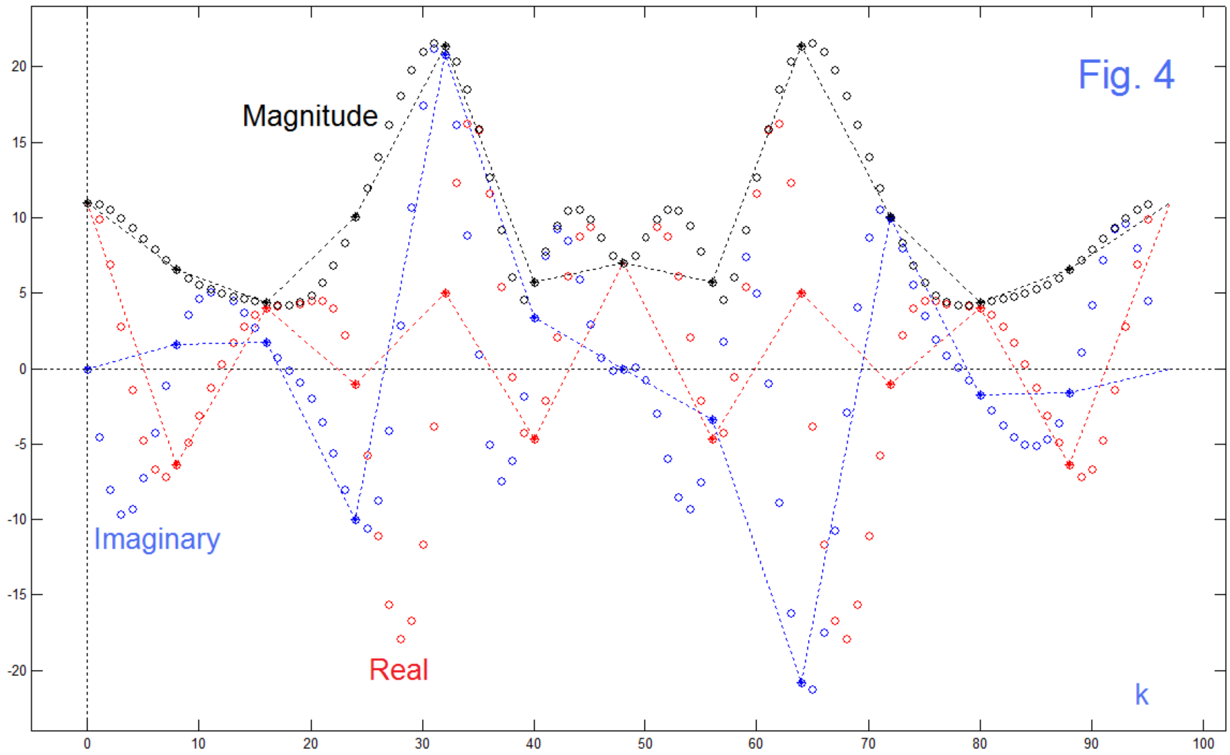


Fig. 4
Magnitude
Imaginary
Real
k
20
15
10
5
0
-5
-10
-15
-20
0
10
20
30
40
50
60
70
80
90
100

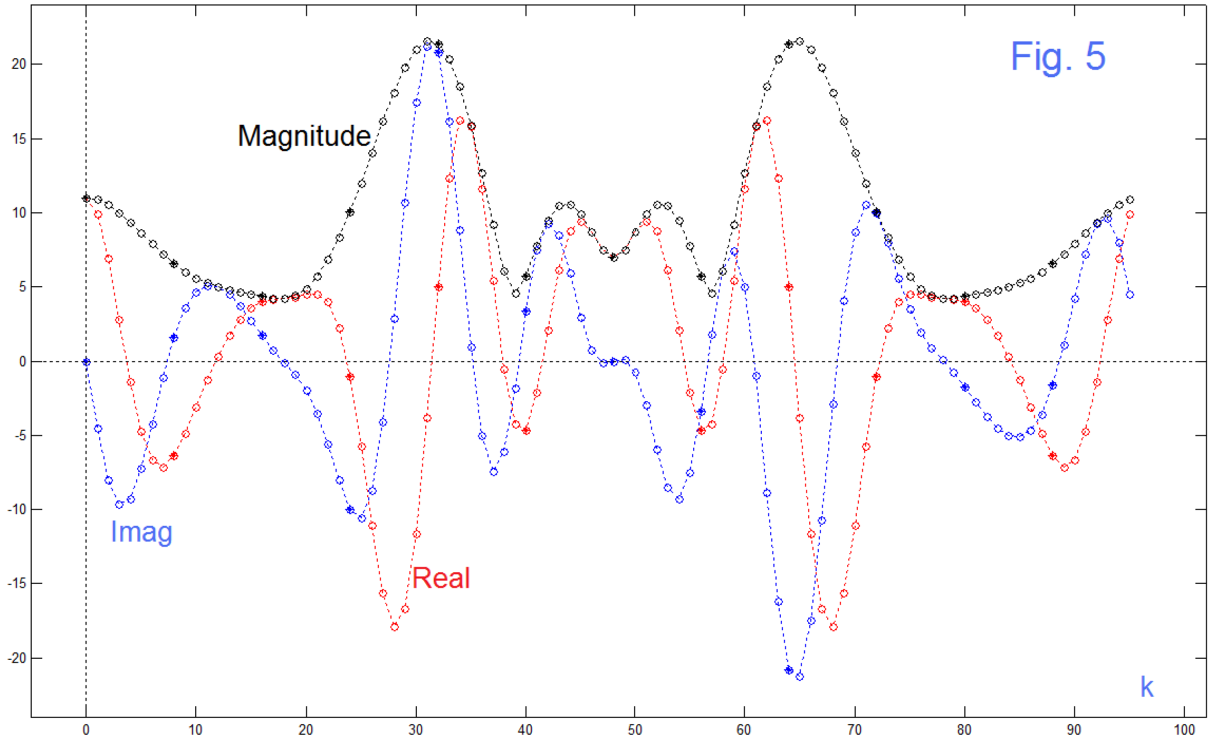
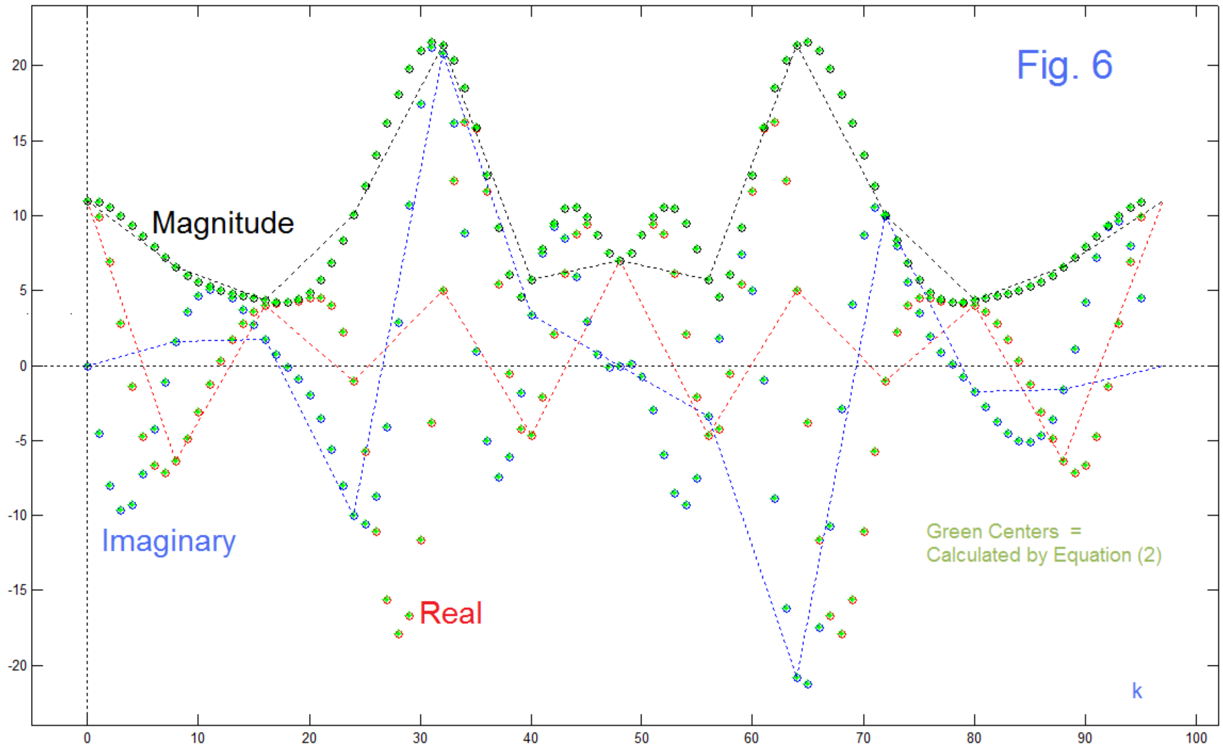
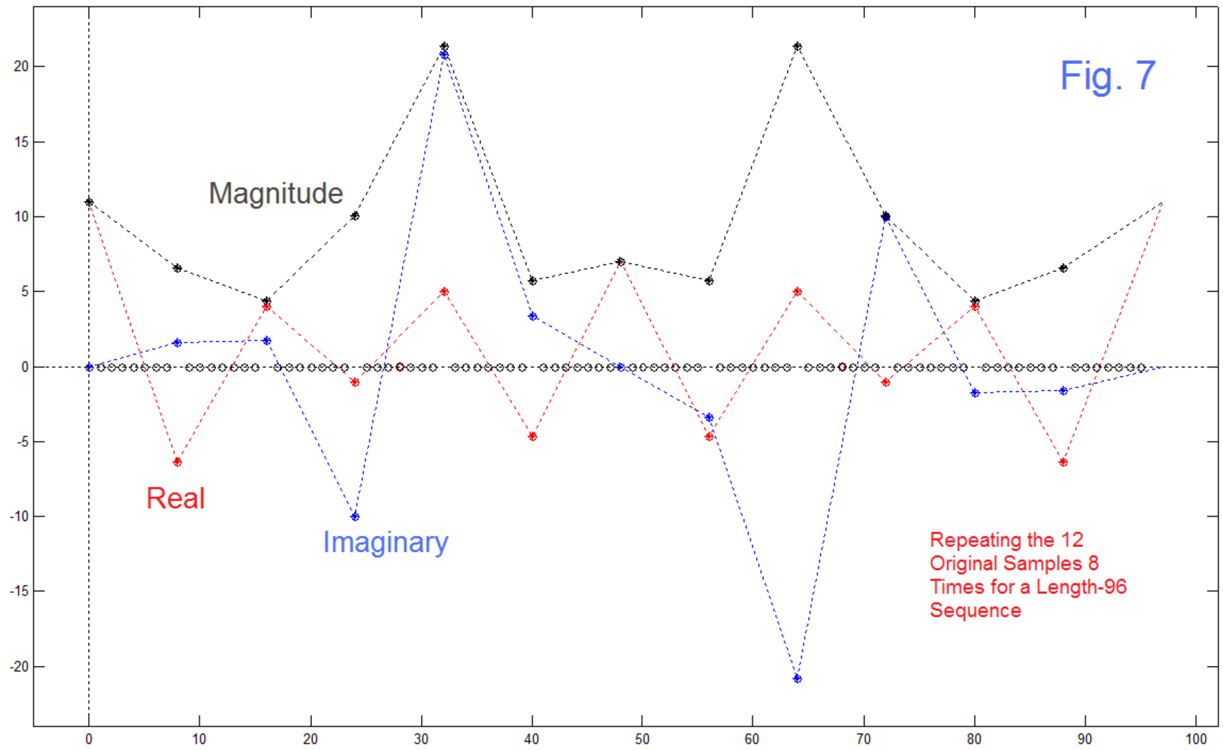


Fig. 5
Magnitude
Imag
Real
k
20
15
10
5
0
-5
-10
-15
-20
0
10
20
30
40
50
60
70
80
90
100



At this point, we are going to do some very simple things and if we are careful, we should get good results quite promptly. First we will add zeros to the end of x(n) (the conventional wisdom). In fact, we add 84 zeros making it length 96 in all. Then we will take the FFT of the padded sequence. We anticipate a frequency-interpolated version of Fig. 3b. It is composed of sinusoidal components, so should be smooth and not very much like the dashed lines of Fig. 3b. Fig. 4 shows the results. Fig. 5 shows the same results with dashed lines between the interpolated set of points. As a check, we will also compute the intermediate spectral values (one at a time) using equation (3). Fig. 6 is just a re-plot of Fig. 4 with the calculated values shown as green stars. The agreement is perfect. The methods are equivalent. (No doubt from the equations, but the surprise is always when you do get it work!)

So Fig. 4, Fig. 5, and Fig. 6 are pretty much the same graphs and constitute the elements of a “slide show” presentation. Fig. 4 relates best to the original length-12 plot while Fig. 6 shows essentially a verification that the calculation works (all dots turn green). This leaves Fig. 5 as the plot that is easiest to follow. Note that we see that interpolation adds a lot to the understanding of the sinusoidal components, and we understand how much the dashed lines help us follow the swings. For example, the red dashed line between k=24 and k=32 as seen in Fig. 4 (straight!) and then in reality, Fig. 5! In going from 12 points to 96 points (Fig. 4), a lot of definition is added by the additional circles. However, the dashed line linear interpolations (Fig. 5) clear up a lot of uncertainty. A better interpolation, perhaps to 1200 points using the DFT, would likely be better than the dashed lines. A lot to examine.



One final variation here is shown in Fig. 7. In as much as we did interpolation by taking a zero-padded time sequence, thus adding no additional information, we might suppose that we could pad with something other than zeros, and still argue that no new information is added. In particular, we might simply repeat the original 12 samples seven additional times and look at the length 96 FFT. One new feature is that we are adding energy in this case, so we need to divide by 8 at some point. The result is the DFT of the original length-12 sequence with 7 zeros between each of the original frequency points. This is no surprise. The DFT harmonics of the length 96 sequence are just 0, 8, 16, … . 96.

CODE FOR FIG. 4 TO FIG. 7

```
%
%  Now zero pad x to xe
xe=[x zeros(1,84)];
Xe=fft(xe) ;
figure(4)
plot([0:95],real(Xe),'ro')
hold on
plot([0:95],imag(Xe),'bo')
plot([0:95],abs(Xe),'ko')
% Plot originals with *
plot([0:8:95],real(X),'r*')
plot([0:8:95],imag(X),'b*')
plot([0:8:95],abs(X),'k*')
```

```
% Plot dashed lines between orig
plot([0:8:95],real(X),'r:')
plot([0:8:95],imag(X),'b:')
plot([0:8:95],abs(X),'k:')
% fudge in dashed lines to starting values for clarity
plot([88 97],[real(X(12)),real(X(1))],'r:')
plot([88 97],[imag(X(12)),imag(X(1))],'b:')
plot([88 97],[abs(X(12)),abs(X(1))],'k:')
%
plot([0 0],[-25 25],'k:')
plot([-5 105],[0 0],'k:')
axis([-5 102 -24 24])
hold off
%
%
% replot Fig. 4 with dashed lines
figure(5)
plot([0:95],real(Xe),'ro')
hold on
plot([0:95],imag(Xe),'bo')
plot([0:95],abs(Xe),'ko')
%Plot originals with *
plot([0:8:95],real(X),'r*')
plot([0:8:95],imag(X),'b*')
plot([0:8:95],abs(X),'k*')
% Plot dashed lines between all
plot([0:95],real(Xe),'r:')
plot([0:95],imag(Xe),'b:')
plot([0:95],abs(Xe),'k:')
%
plot([0 0],[-25 25],'k:')
plot([-5 105],[0 0],'k:')
axis([-5 102 -24 24])
hold off
%
%
% Calculate with equation (2)
Xdc=zeros(1,96);
for kk=1:96
   w=(kk-1)*2*pi/96;
   for n=0:11
          Xdc(kk)=Xdc(kk)+xe(n+1)*exp(-j*n*w);
   end
end
figure(6)
% Interpolated by FFT
plot([0:95],real(Xe),'ro')
hold on
plot([0:95],imag(Xe),'bo')
plot([0:95],abs(Xe),'ko')
```

```
% Originals from length 12
plot([0:8:95],real(X),'r*')
plot([0:8:95],imag(X),'b*')
plot([0:8:95],abs(X),'k*')
% Direct Calculations as green *
plot([0:95],real(Xdc),'g*')
plot([0:95],imag(Xdc),'g*')
plot([0:95],abs(Xdc),'g*')
%  dashed lines for original length 12
plot([0:8:95],real(X),'r:')
plot([0:8:95],imag(X),'b:')
plot([0:8:95],abs(X),'k:')
%  fudge connection
plot([88 97],[real(X(12)),real(X(1))],'r:')
plot([88 97],[imag(X(12)),imag(X(1))],'b:')
plot([88 97],[abs(X(12)),abs(X(1))],'k:')
%
plot([0 0],[-25 25],'k:')
plot([-5 105],[0 0],'k:')
axis([-5 102 -24 24])
hold off
%
%
%
%  Repeat the length-12 sequence 8 times
xe=[x x x x x x x x ]/8;
Xe=fft(xe) ;
figure(7)
plot([0:95],real(Xe),'ro')
hold on
plot([0:95],imag(Xe),'bo')
plot([0:95],abs(Xe),'ko')
% Plot originals with *
plot([0:8:95],real(X),'r*')
plot([0:8:95],imag(X),'b*')
plot([0:8:95],abs(X),'k*')
% Plot dashed lines between orig
plot([0:8:95],real(X),'r:')
plot([0:8:95],imag(X),'b:')
plot([0:8:95],abs(X),'k:')
plot([88 97],[real(X(12)),real(X(1))],'r:')
plot([88 97],[imag(X(12)),imag(X(1))],'b:')
plot([88 97],[abs(X(12)),abs(X(1))],'k:')
%
plot([0 0],[-25 25],'k:')
plot([-5 105],[0 0],'k:')
axis([-5 102 -24 24])
hold off
```

REFERENCES (All from EN Publications)

[1] “Fourier Map”, Electronotes Application Note No. 410, May 6, 2014 http://electronotes.netfirms.com/AN410.pdf

[2] “Basic Elements of Digital Signal Processing: Filter Elements – Part 1”, ***Electronotes***, Vol. 20, No. 197, pp 3-6; http://electronotes.netfirms.com/EN197.pdf “Basic Elements of Digital Signal Processing: Filter Elements – Part 2”, ***Electronotes***, Vol. 20, No. 198 , pp 12-13 http://electronotes.netfirms.com/EN198.pdf

[3] “Linear Phase - The Term Taken Outside”, Electronotes Application Note No. 350, November 1998 http://electronotes.netfirms.com/AN350.PDF

[4] “FFT – Interpolation In Time And Frequency “, ***Electronotes***, Volume 23, Number 222 June 2015 http://electronotes.netfirms.com/EN222.pdf

[5] “A Short Presentation Of FFT (DFT) Interpolation”, Electronotes Application Note No. 398, July 20, 2013 http://electronotes.netfirms.com/AN398.pdf See also: “Interpolating Samples with the DFT/FFT,” ***Electronotes***, Vol. 16, No. 172, July 1988. See also: Interpretation of DFT as One Frequency Sinusoidal Waveforms”, Electronotes Application Note No. 373, February 2009 http://electronotes.netfirms.com/AN373.pdf and “Calculating the DFT and the Fourier Series: Each with the Other”, ***Electronotes***, Vol. 19, No. 188, February 1997

[6] Note: This relates to the “distress” admitted to in [4] as to why there is a well-defined zero in frequency (zero frequency = direct current) with no corresponding zero in time. This, and the tendency to make time functions and sequence be purely real (instilling certain symmetries in the transformed frequency description) tends to prejudice our expectations about how time and frequency relate.