A SHORT PRESENTATION OF FFT (DFT) INTERPOLATION

It is possible to interpolate time-domain samples using the FFT.   The procedure is to first take the FFT of the time sequence, zero-pad the middle of this FFT, and then take the inverse FFT of the larger FFT to obtain the longer time-sequence. For example, a length 20 time signal could be interpolated to a length 100 time sequence by inserting 80 zeros in the middle of its FFT. In such a case, while the total length of the sequence changes, we usually think of the new samples (4 or every 5) as now being <u>between</u> the originals, which remain in the same relative positions in time. As such, we can also think of the sampling rate as increasing by a factor of 5.

An example (Fig. A1) will serve as an intuitive explanation of the validity of the method. Consider a sequence of length 6 that is [0 1 1 0 -1 -1]. Clearly these samples might have come from anywhere. They could be, as one obvious example, samples of a sine of amplitude $2/\sqrt{3} = 1.1547$, taken at 0, 60°, 120°, 180°, 240°, and 300°

We desire to interpolate additional samples within this single cycle. Perhaps we want to be more convinced that the six given samples do correspond to the sinewave suggested. The first step is to take the FFT of the six time samples. This gives us two non-zero FFT values, both purely imaginary (because of the sine symmetry we start with). We assume the reader is familiar with the FFT.

The tricky part is to now "zero pad" the FFT up to a size corresponding to the number of time samples we want to end up with. Here we have inserted 18 zeros so that we will end up with 24 time samples.   Note that the zeros are inserted in the exact middle of the FFT.   Keep in mind that k=0 is a unique point (dc) so the middle of this length N=6 FFT is at N/2=3. The center of the zero-padded FFT is at 12, since N becomes 24.

FIG A1
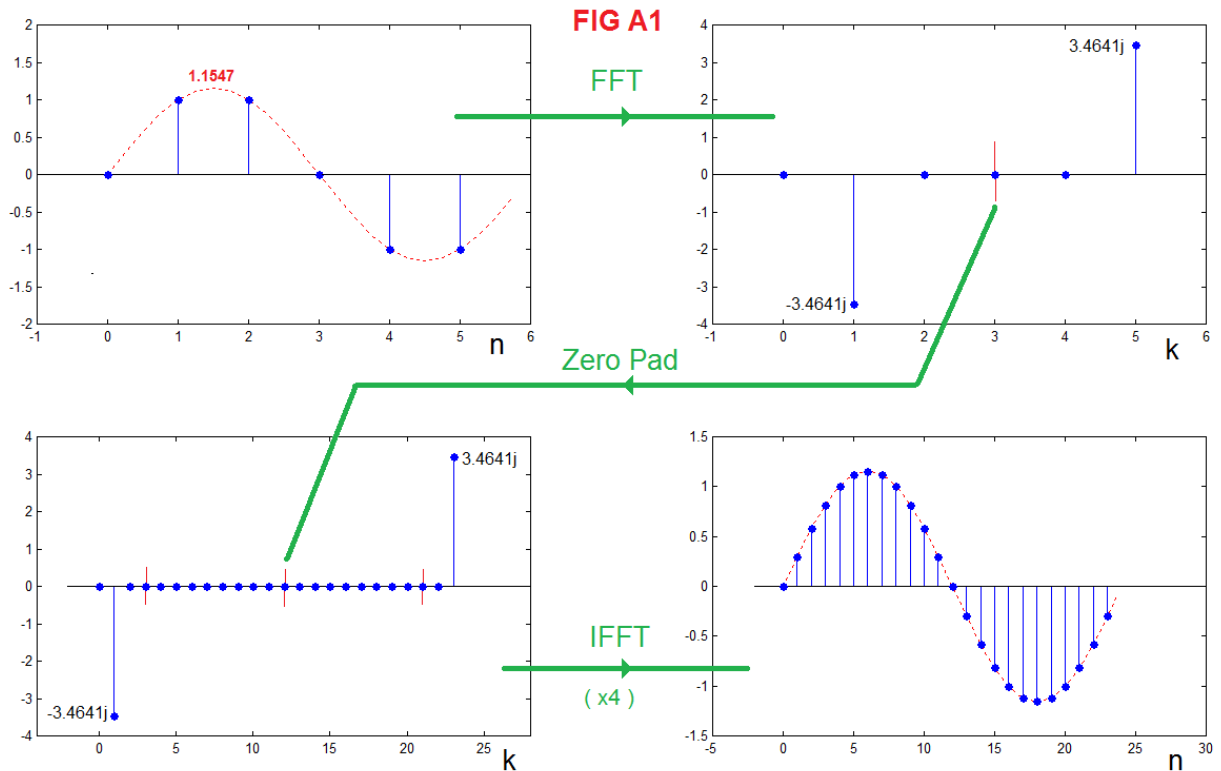
The final step is to take the inverse FFT of the zero-padded FFT. Adding the zeros in the center pushes the non-zero portions of the original FFT apart, but maintained the symmetry, and the values are still purely imaginary. Thus we end up getting a sine back. There are now 24 samples, up from just 6, in the final result. Further, we note that the zero-padded FFT is exactly what we expect from 24 samples of a sinewave cycle.

NOTES: The example above is the basic idea, and note well that this method applies to much more general cases. Here are a few things you may bump into.

(1) Well, we did multiply the IFFT by 4. This is because of the different (1/N) factors in the different size FFT's here. In general, multiply by N_2/N_1 where N_1 is the original number of points and N_2 is the final number. This ratio need not be an integer (here is 4, but it could be, for example 5/2 if we added 9 zeros, and so on).

AN-398 (2)

(2) As long as we start with a real sequence, the procedure returns a real sequence – in theory. (We can do complex sequences equally well if necessary.) However, the matter of taking an FFT and eventually an IFFT in general results in a tiny tiny imaginary part due to arithmetic roundoff that is not supposed to be there. It's an accident. As a result, you usually need to take the real part of the IFFT to avoid confusion and/or warning messages while plotting.

(3) In the example, N was even. If N is odd, we still put zeros in the middle about N/2 – there just won't be an original point there. Probably this is in fact easier. This does however brings up the question of what we would have done at N/2, for the <u>even</u> case, if X(N/2) is not zero. How do we handle this! Well, in the first place, perhaps it will be zero. If the sequence is sampled according to the sampling theorem, the waveform it came from should have been bandlimited such that there was no energy at half the sampling frequency (k = N/2). If there is however, nonetheless a non-zero FFT value at N/2, it should be split (divided by 2) with half moving downward, and half upward, as the zeros are inserted. This is best understood by considering that the inverse of interpolation is a process that discards samples (resamples) and any energy at N/2 would be the result of the overlap of two identical (real) contributions (the insight here was from Tom Parks).   See: B. Hutchins, "Interpolating Samples with the DFT/FFT," <u>Electronotes</u>, Vol. 16, No. 172, July 1988.

THE MATLAB CODE THAT MADE THE BASIC FIGURE

```
% FFTINTERP.m

x=[0 1 1 0 -1 -1 ];
X=fft(x)
Y=[X(1:3) zeros(1,18) X(4:6)];
y=4*real(ifft(Y));

figure(1)
subplot(221)
stem([0:5],x)
hold on
plot([-2 30],[0 0],'k')
xx=(2/sqrt(3))*sin(2*pi*15*[0:23]/360);
plot([0:1/4:5.75],xx,':r')
axis([-1 6 -2 2])
hold off

subplot(222)
stem([0:5],imag(X))
hold on
plot([-2 30],[0 0],'k')
axis([-1 6 -4 4])
hold off

subplot(223)
stem([0:23],imag(Y))
hold on
plot([-2 30],[0 0],'k')
hold off
axis([-4 28 -4 4])
subplot(224)
stem([0:23],y)
hold on
plot([-2 30],[0 0],'k')
xx=(2/sqrt(3))*sin(2*pi*15*[0:1/4:23.75]/360);
plot([0:1/4:23.75],xx,':r')
hold off

figure(1)
```