On Local Interpolation of DFT Outputs

by
Eric Jacobsen
EF Data Corporation

EDICS 3.1.1

Abstract - In many cases frequency estimation of isolated tones can be aided by curve fitting to, or interpolation of interbin energy levels in DFT outputs. Methods of performing DFT magnitude and frequency interpolation are investigated and compared relative to practical real-time applications. Two new methods are proposed which may offer advantages over current approaches: a frequency interpolator which reduces the required number of divide operations with a small increase in MSE, and a new magnitude interpolator which requires no divides.

Eric Jacobsen
EF Data Corp.
2105 W. 5th Pl.
Tempe, AZ 85281

Vox: (602)968-0447
Fax: (602)921-9012

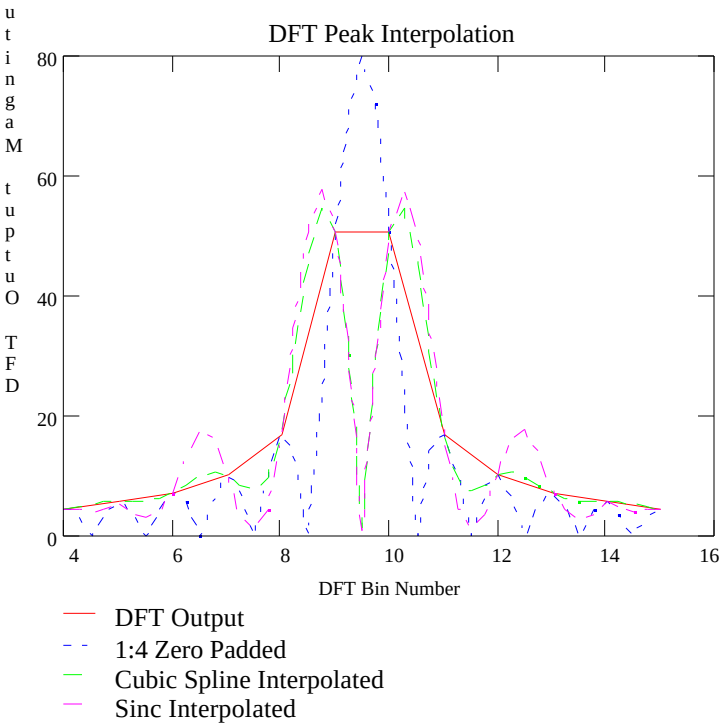
e-mail: ericj@efdata.com, ericj@primenet.com

I. Introduction

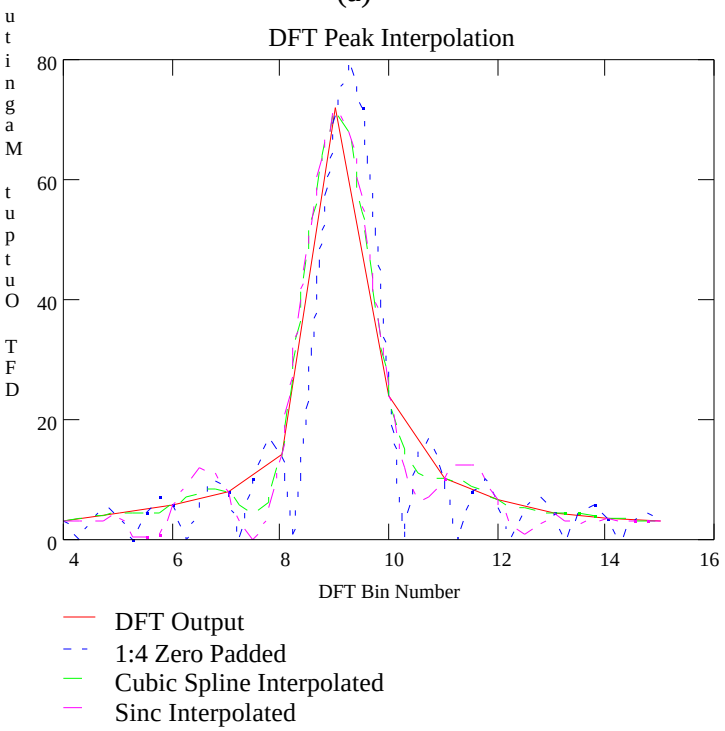
In many applications it is necessary to detect the frequency of a single tone in a noisy environment. Taking a DFT or FFT of collected samples is arguably the most common method of making such frequency estimates. Computational or other limitations often restrict the number of samples which may be processed, which correspondingly restricts the resolution of the estimate provided by the DFT. Quinn [1] developed a simple and efficient method to closely estimate a signal frequency based on the three samples around the DFT output peak. A similar method was shown by Grandke [2] which uses only the DFT output peak and one adjacent sample. Both methods provide efficient frequency estimators which perform well to SNRs as low as 0dB. Neither directly provides a corrected magnitude estimate, and both require division. Jain, et al, [3] proposed a similar frequency estimator, but the operating assumptions are restrictive and, again, division is required for the frequency estimate. Most contemporary DSPs do not have efficient divide instructions, so algorithms which minimize or eliminate the use of divides are advantageous.

Two algorithms presented here offer advantages over the previous algorithms. The first, a quadratic curve-fitting interpolator, offers frequency estimation performance very close to Quinn's method but requires only one divide. In practical DSP-based systems the resolution improvement is then dictated by the number of divide iterations that can be tolerated; A factor of eight improvement in frequency resolution may be achieved with only three divide iterations. This method also does not directly provide a magnitude estimate. The second method requires no divides, but provides a magnitude estimate at only one predetermined frequency. This method is typically used to double the frequency resolution by providing a local interbin magnitude estimate once a DFT output peak has been located.

It is prudent to point out that one motivation for studies specific to DFT output interpolation is the failure of traditional interpolation methods for this application. Figure 1a shows a DFT output peak for a monochromatic input, with interpolations performed on the complex data using zero-padding, sinc and cubic spline interpolators. The input frequency is such that the true peak magnitude lies halfway between DFT output bins 9 and 10. Both interpolators generate a false null in the center of the main lobe, which is obviously detrimental to peak detection and location in DFT post processing. Figure 1b shows the detected magnitude of a tone with a frequency of 9.25 (one fourth of the way from bin 9 to bin 10) , and sinc and cubic spline interpolations of the detected magnitude. The non-linearity in the magnitude detection results in a signal which is essentially undersampled, e.g., interpolation will never properly restore the nulls between the sidelobes. It is therefore difficult for the interpolators to properly locate the true peak or deduce the true shape of the main lobe. The failure of these common interpolation methods to yield satisfactory performance when operating on complex DFT outputs or their detected magnitudes helps drive the quest for alternate methods.



(a)



(b)

Figure 1. Raw and interpolated responses for DFT outputs. a) Interpolation of complex DFT output for tone halfway between bins 9 and 10. b) Interpolation of magnitude of DFT output for tone at bin 9.25.

II. DFT Peak Discrimination Performance

The assumed application involves a real-time system which uses a DFT of some kind to generate an estimate of the frequency of a particular monochromatic signal. Often additional tasks are involved, such as detection of the presence of the signal or measurement of the signal power content. Time and processing power constraints typically dictate the length and method of DFT calculation employed, and it will be assumed for this discussion that a length-64 DFT is used, although the principles presented apply to any length of DFT. Many times the required resolution of the frequency estimate cannot be met with the length of DFT which can be practically calculated. Use of an efficient interpolator can easily be seen as a method of reconciling the conflicting requirements. An order of events can also be assumed for the estimation and interpolation process which is as follows:

1) Make an initial, coarse estimate of the signal frequency by locating the maximizer of the DFT output magnitude.
2) Isolate the DFT outputs local to the bin determined in step 1).
3) Apply an algorithm to the isolated samples which increases the resolution of the frequency estimate and possibly improves the magnitude estimate.

It is important to observe that ultimate success depends on the ability of step 1 to provide the correct DFT output bin number. If the tone is actually closest to bin nine and noise or other distortion causes the magnitude maximizer to fall elsewhere, such as bin 10 or 11, the algorithms considered here have little hope of providing useful information. For this reason it is important to consider the limitations of the DFT algorithm itself. Figure 2 shows the results of a simulation performed to determine the probability of correctly determining the bin number of a 64-point DFT output for a rectangular-windowed tone in AWGN. Since the threshold effect [4][5] causes performance to drop rapidly below -3dB SNR, lower SNRs will not be considered when testing

the interpolation algorithms, since the DFT will be the limiting factor in such cases. Although the general shape of the curve will remain the same, performance can be improved by using a longer DFT. The noise-suppression properties of the DFT can be exploited to improve performance by 3dB for every doubling of DFT length.

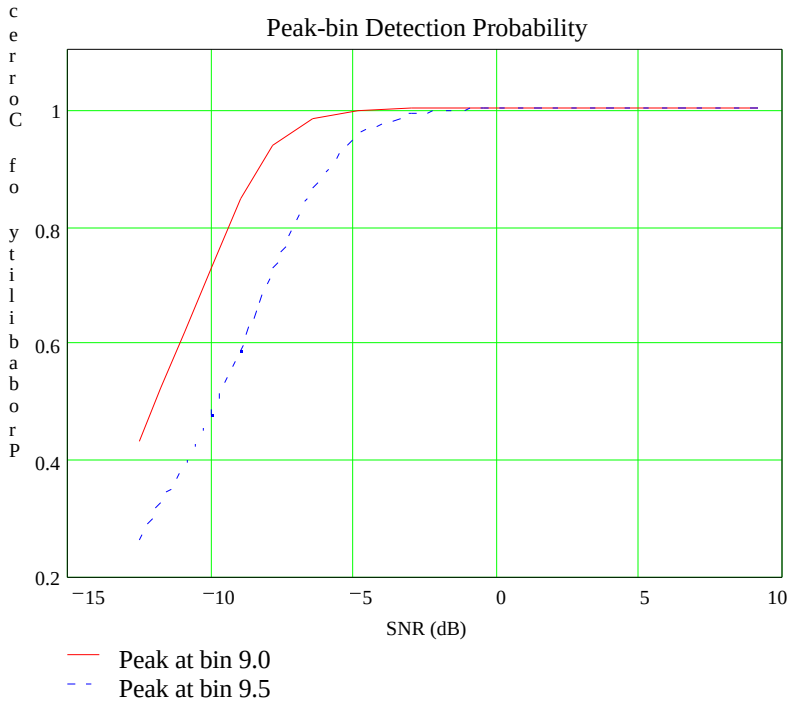


Figure 2. Probability of correctly locating tone in AWGN using 64-pt DFT output.

III. Frequency Interpolating

One class of DFT output interpolators provides an estimate of the frequency of an isolated tone given the complex output samples local to the maximizer of the DFT output magnitude. Quinn and Grandke are among those who have proposed such methods, with Quinn's being perhaps the most well known [1][2]. Both provide a real number, δ, ideally between -0.5 and +0.5, which

indicates the distance from the DFT output maximizer to the true frequency in bin widths. A tone located at bin 9.25 would presumably yield a DFT output magnitude maximizer at bin 9, and the interpolator would then yield +0.25 to be added to the maximizer bin number.

Grandke's method requires that the DFT input be weighted with a Hanning window, which places additional computational load on the system. Once this is done, however, the DFT maximizer and it's largest adjacent can then be used to provide the interpolator inputs. Let G_m and G_{m+1} be the maximizer and it's largest neighbor, where m is the bin number index as appropriate. Then calculating the following ratios provides the frequency correction estimate.

$$\alpha = \frac{|G_{m+1}|}{|G_m|} \qquad (3.1)$$

$$\delta = \frac{2 \cdot \alpha - 1}{\alpha + 1} \qquad (3.2)$$

Quinn's method does not require special weighting of the DFT input, and utilizes the DFT output maximizer and both adjacent bins. Let G_m be the DFT output magnitude maximizer, and then calculate the following ratios for Quinn's estimator.

$$\alpha_1 = \mathrm{Re}\left\{\frac{G_{m-1}}{G_m}\right\} \qquad (3.3)$$

$$\alpha_2 = \mathrm{Re}\left\{\frac{G_{m+1}}{G_m}\right\} \qquad (3.4)$$

$$\delta_1 = \frac{\alpha_1}{1 - \alpha_1} \qquad (3.5)$$

$$\delta_2 = \frac{-\alpha_2}{1-\alpha_2} \qquad (3.6)$$

If $\delta_1>0$ and $\delta_2>0$, then $\delta=\delta_2$, otherwise $\delta=\delta_1$.

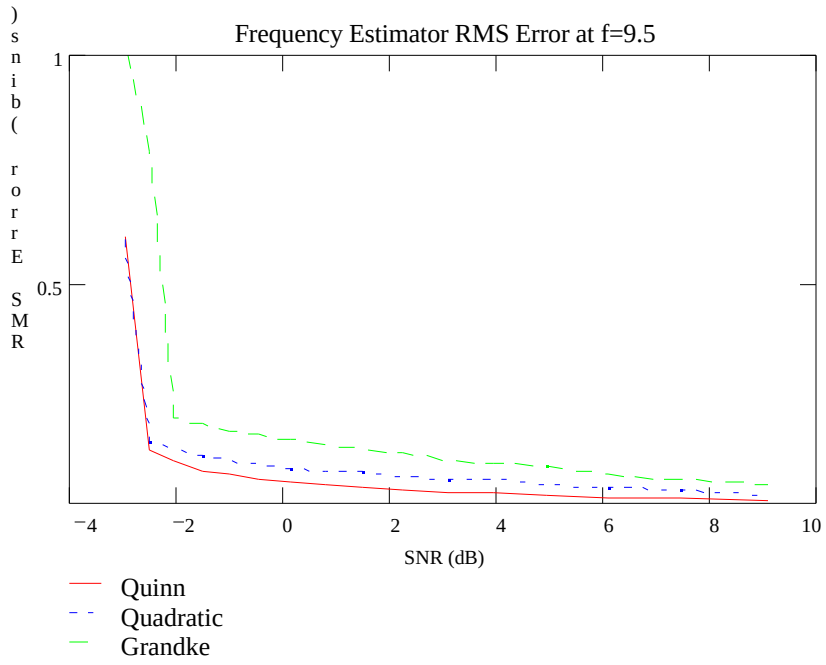
If the real-time system in question utilizes a contemporary fixed-point DSP VLSI processor, a divide instruction will typically not be available. While divides can be hand-coded, an iteration per-output-bit is required which can consume needed processing bandwidth. Grandke's method imposes an additional processing burden with the Hanning window requirement, while Quinn's method requires four divides. A new method suggested by Robert Bristow-Johnson for interpolating cross-correlation sequence peaks and adapted by the author for DFT outputs, requires a single divide and a rectangular DFT input window. This provides a computational advantage and a manageable processing tradeoff: every divide iteration calculated increases the frequency estimate resolution by a factor of two. e.g., a factor of eight improvement over the DFT output resolution can be reached with only three divide iterations.

The new quadratic estimator is calculated from the DFT output magnitude maximizer G_m and its two adjacent samples as follows:

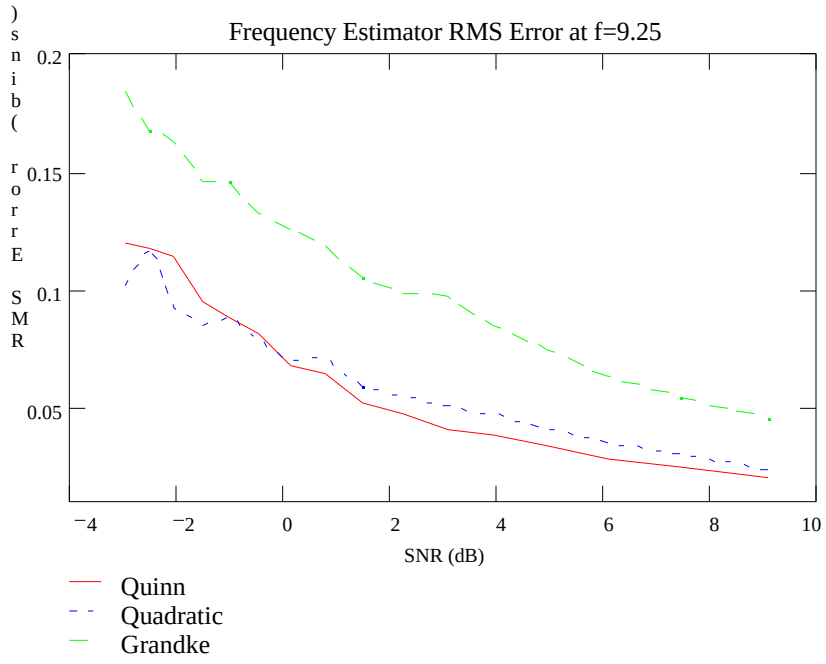
$$\delta = \mathrm{Re}\left\{\frac{G_{m+1}-G_{m-1}}{2\cdot G_m - G_{m+1} - G_{m-1}}\right\} \qquad (3.7)$$

Appendix A contains the derivation of 3.7

Figure 3a shows the relative performance of the three estimators in AWGN. A length-64 DFT was used with one thousand trials performed on each estimator at each SNR tested. The input tone was located at bin 9.5, i.e., halfway between bins 9 and 10, and the phase was varied randomly. Figure 3b shows the results for a repeat of the experiment with the tone located at bin 9.25. It is apparent that while Quinn's estimator appears to provide the lowest rms error, the new estimator provides nearly equal performance with a possible computational benefit in reduced division requirements. Figure 4 shows the rms error of the estimators as a function of frequency. It is seen that the new estimator actually outperforms Quinn's when the frequency is nearly aligned to a DFT bin.



(a)



(b)

Figure 3. Comparative results for frequency estimators for a) input tone at bin 9.5, and b) input tone at bin 9.25.

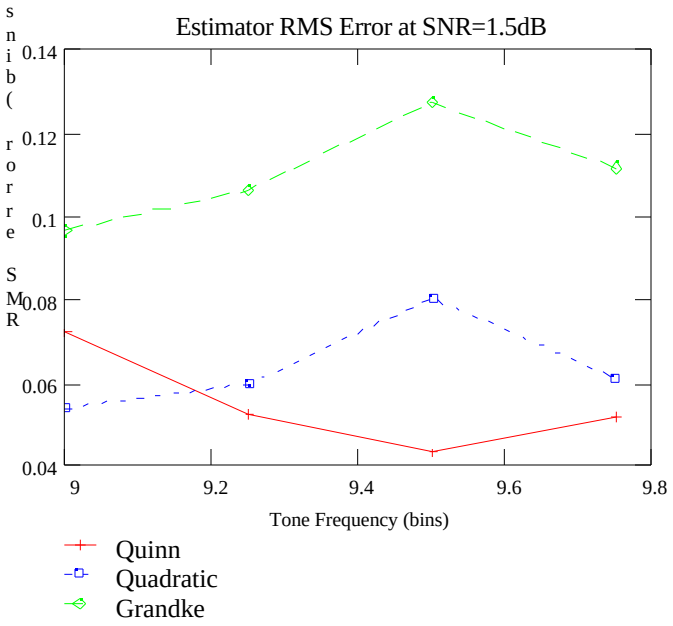


Figure 4. Comparative performance of estimators at SNR=1.5dB.

IV. Magnitude Interpolating

In many applications it is desirable to improve not only the initial DFT frequency estimate, but the magnitude estimate as well. The magnitude estimate can be largely affected by the input frequency, particularly for rectangular input windows, due to the sampling of the main lobe of the sinc function. Grandke and Jain both offered methods of determining an improved magnitude estimate using the interpolated frequency estimate, but both require additional divisions as well as calculation of transcendental functions. In the assumed fixed-point real-time application, this may simply not be possible. The method proposed below, while not offering the frequency resolution of the estimators discussed in section III, provides a tradeoff between computational load, frequency resolution, and magnitude estimation capability that is beneficial in some applications.

One well-known method for increasing the resolution of the frequency estimate involves zero-padding the FFT input. While this method does not reduce the width of the sinusoidal-response sinc function in the frequency domain, it does increase the number of samples in the sinc function. Increasing the number of samples in the sinc improves the performance of a simple peak-detector frequency estimator operating on a DFT output as long as the SNR is not so low that the distortion produces a false peak in the interpolated output. For most applications the SNR gain of the DFT allows this type of interpolation to provide good results in very noisy environments.

Strictly zero-padding the input of the DFT carries a large computational penalty. Similar performance can be obtained by isolating the peak of the DFT output, taking the inverse transform, zero-padding the results, and taking the forward transform of the zero-padded vector. Since this method does not use all of the information contained in the original DFT output, the magnitude of the result generally does not equal the magnitude determined by zero-filling the original time-domain vector. This method does improve the frequency and magnitude resolution over the original DFT output, however, and the computational reduction realized by using a subset of the DFT output provides a useful design tradeoff.

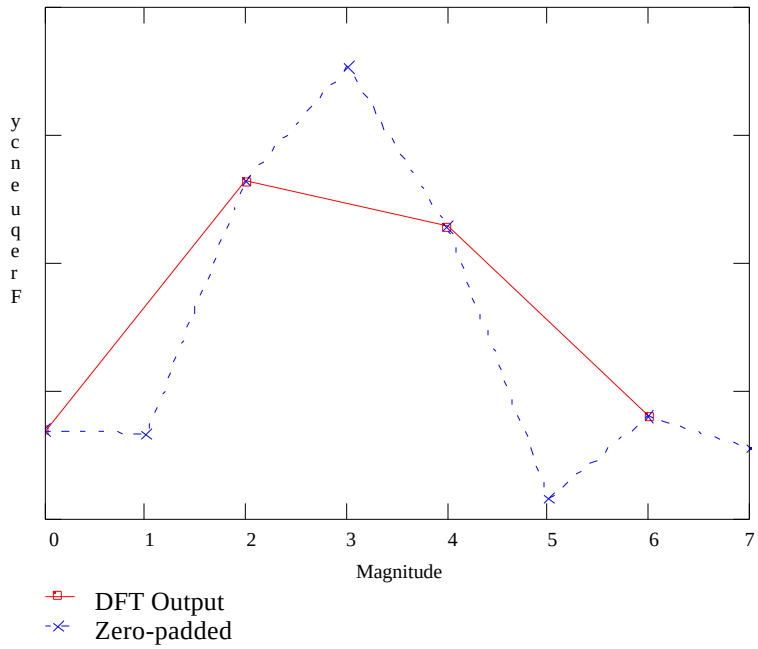


Figure 5. DFT peak output and 1:2 interpolation performed by zero-padding.

Figure 5. shows four samples taken from a 64-point DFT output, taken such that the peak magnitude sample and largest adjacent magnitude sample occupy the center. The interpolated values were created by taking the inverse DFT of the four samples, appending four zeroes to the result, and taking the forward transform of the zero-padded vector. The 1:2 interpolated result shows that the interpolated center sample provides a better estimate of the peak power frequency than the original samples. Computationally, this method of interpolation is not trivial even for such small numbers of samples, especially when compared to Quinn's method or Eqn 3.7.

The interpolation in Figure 5 requires an inverse and forward DFT to produce an interpolated vector that is longer than the input by the number of zeroes that have been inserted. In the example given, the only interpolated output that is of interest is the center sample; the real

question is whether the input frequency corresponded best to the original peak position or the interpolated frequency position. Since only the center sample is of interest, calculation of the others may be skipped. Assuming a DFT is used for the second transform, this reduces the number of computations from $O(NL^2)$ to $O(NL)$, where NL is the number of samples in the long DFT. If NS is the number of samples to be interpolated, then NL-NS samples are padding zeroes and do not contribute to the output of the second DFT. The interpolation process then becomes an NS-point forward transform and an NS-point dot product with the appropriate twiddle factors from the second DFT. These two operations can be combined into a reduced operation that is essentially a single NS-point dot product.

Good performance can be obtained by letting NS=4 and calculating the interpolating vector from:

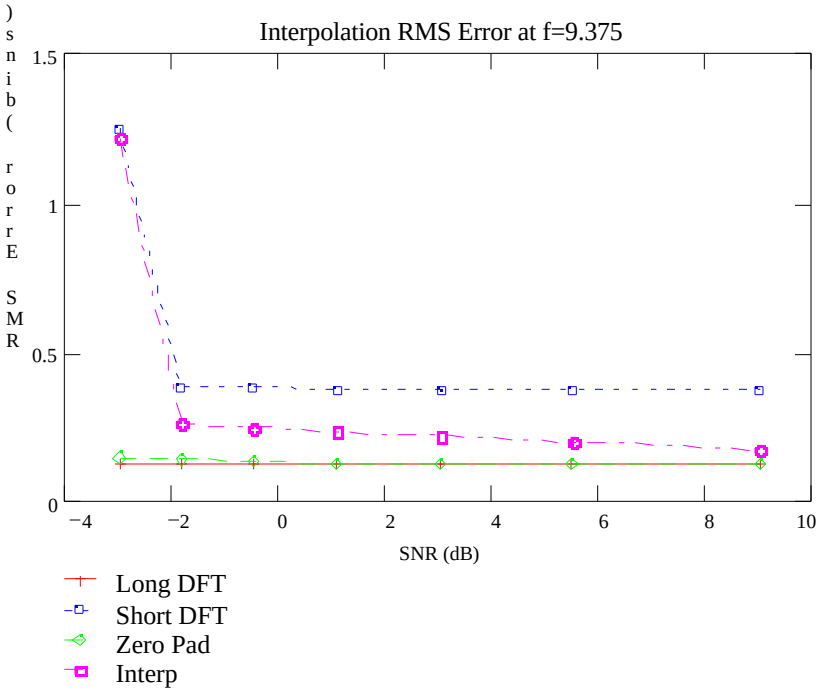
$$M_n = \sum_{i=0}^{3} e^{\frac{j \cdot 2 \cdot \pi \cdot i \cdot 3}{8}} \qquad (4.1)$$

where n runs from 0 to 3. M is then a 4-element interpolating vector which can be used with G, the four elements around the DFT output maximizer, to form a dot product. The dot product of M and G is the interpolated sample halfway between the center samples of G. Appendix B contains the rationale and derivations for 4.1.

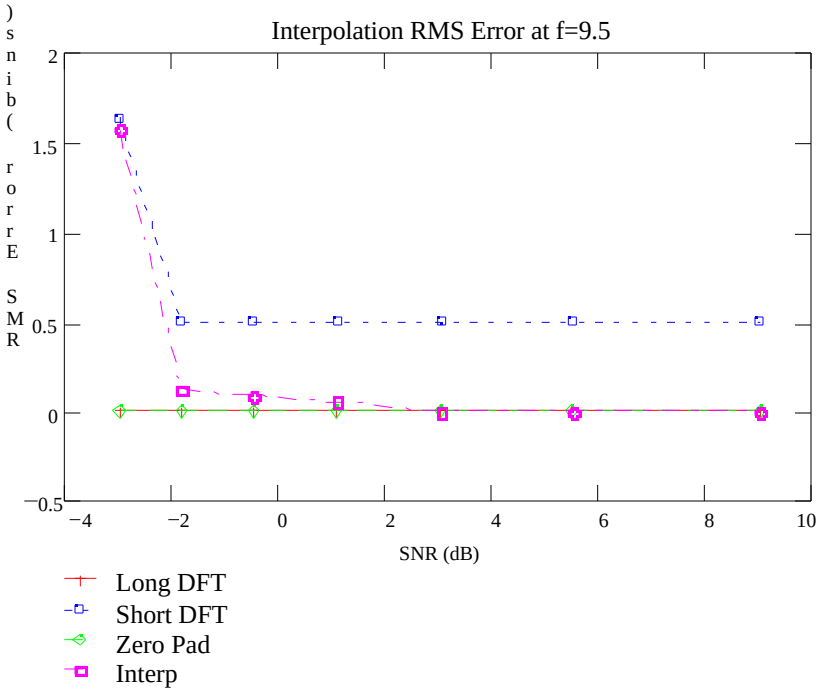
In order to evaluate the performance of the interpolator in 4.1, an experimental simulation was created very similar to that used in section III. Four frequency estimation methods were compared: a 64-pt DFT, a 128-pt DFT, a 128-pt DFT of 64-pt data zero-padded to 128-pts, and a 64-pt DFT with the interpolator of 4.1 used on the four points about the maximizer. In all cases the DFT output maximizer was assumed to indicate the signal frequency. The four element vector G used for the interpolator were chosen such that the maximizer and its largest adjacent sample formed the center two elements. If the interpolator output had a larger magnitude than

the maximizer in G, then it was assumed that the signal tone location was between the center samples of G.

Figure 6a shows the rms error for one thousand trials each for each estimation method across a range of Signal-to-Noise ratios. The frequency of the input tone corresponded to bin 9.375, which avoids any symmetries in the DFT performance in order to more clearly demonstrate the performance of the estimators. While the long (128-pt) DFT and the zero-padded DFT have very similar performance, the interpolator clearly offers a performance improvement over the short (64-pt) DFT, and approaches the performance of the long DFT as SNR increases. Figure 6b shows the results for a tone at bin 9.5. In this case the long DFT and zero-padded DFT have identical performance, and the interpolator performance matches that of the long DFT even more closely. Figure 7 shows the mean magnitude estimates versus tone frequency at SNR = -0.5dB. Although the interpolator does provide a magnitude estimate that is superior to the short DFT alone, the error is still large for frequencies between DFT bins.



(a)



(b)

Figure 6. Comparative results for frequency interpolators for a) tone at bin 9.375, b) tone at bin 9.5

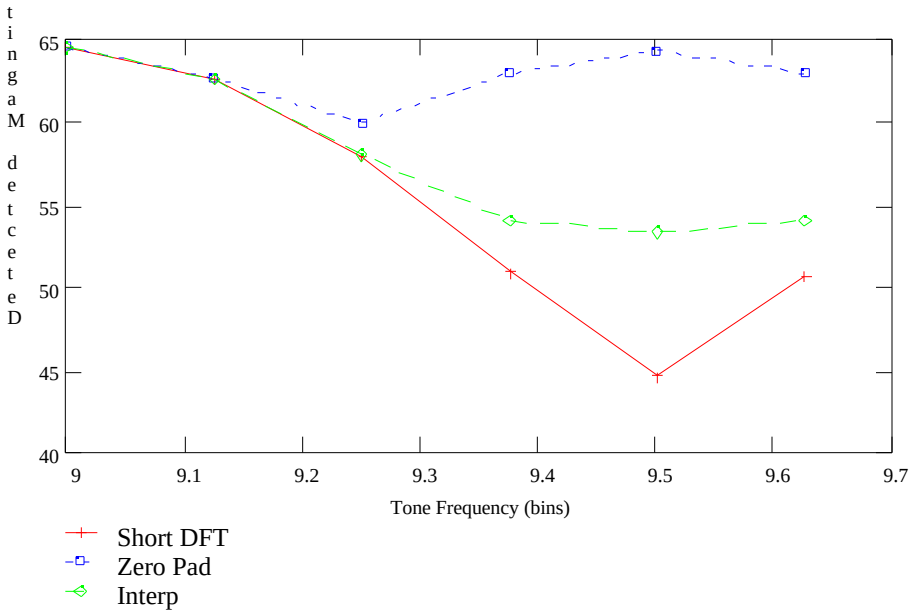
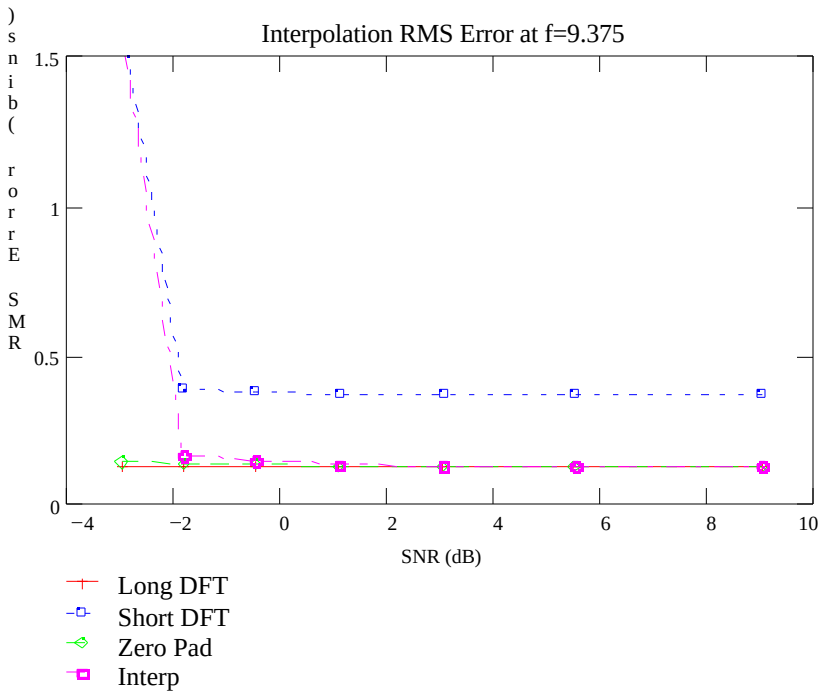
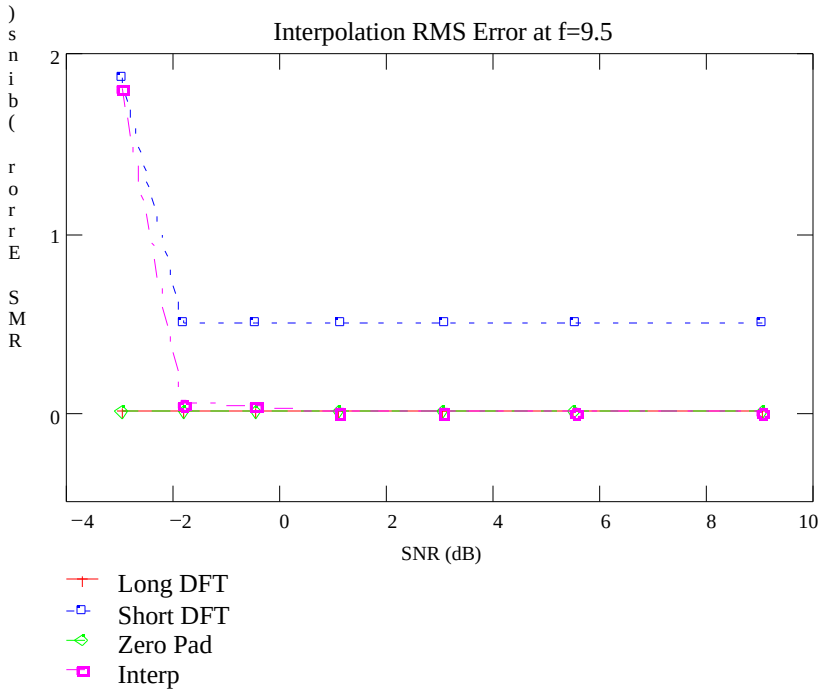


Figure 7. Peak detected magnitude using interpolators.

The experimental data indicated that the ratio of the magnitude estimates from the zero-padded DFT and the interpolator was a nearly constant 1.2 over all SNRs at bin 9.5. Since scaling the output would affect the frequency estimation decision in the algorithm, the experiment was re-run with the 1.2 correction factor in place. Figure 8 shows the results of the simulations including the scale factor. The frequency estimation performance of the interpolator is clearly improved over that shown in Figure 6, and rivals the performance of the long DFT. Figure 9 shows the magnitude estimation performance over frequency at SNR = -0.5dB, with the corrected interpolator. The performance of the interpolator very closely matches that of the zero-padded DFT.



(a)



(b)

Figure 8. Comparative results for frequency interpolators for a) tone at bin 9.375, b) tone at bin 9.5 The dot-product interpolator includes a scale factor determined from the previous experiment.

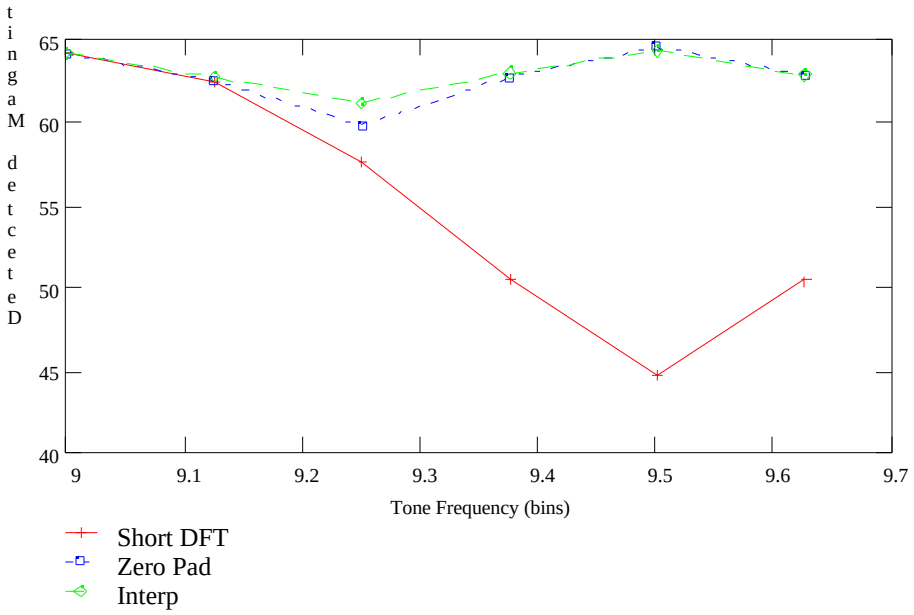


Figure 9. Peak detected magnitude using interpolators. The dot product interpolator includes a scale factor determined from the previous experiment. Inclusion of the scale factor produces performance nearly equal to zero-padding the full length of the transform.

The performances of the estimators must be evaluated against the computational complexity in real-time systems. While the long DFT will clearly always provide superior performance, it does so with high computational complexity and a longer data collection time. The zero-padded DFT halves the data collection time with very little loss of performance, but with the same computational complexity as the long DFT. The short DFT realizes reduced computational complexity and reduced data collection time, but also reduces the resolution of the frequency estimate. The proposed interpolator requires an N-point dot product with a precomputed vector and provides performance close to the long DFT with N as small as 4, particularly if a gain correction factor is used. Computation of the gain correction factor is dependent on N as well as the length of the DFT, and is subject to further research.

V. Conclusion

When interpolating DFT outputs for improved performance in time or processor constrained systems, tradeoffs may be performed between the system constraints and the algorithms presented. If magnitude information is inconsequential, Quinn's method or Eqn. 3.7 may be used to more accurately determine the true signal frequency with reasonable computational load. If division calculation is an issue, as with fixed-point DSP systems, then Eqn. 3.7 offers a computational improvement over Quinn's method with little loss of performance in the estimator.

An additional method which completely eliminates divides and also provides an improved magnitude estimate was presented in Section IV. While this method would typically be used to calculate only one to three interpolated values, each interpolated point is determined from an N-point complex dot product where good performance is achieved with N as low as 4.

ACKNOWLEDGMENTS

Credit is due Robert Bristow-Johnson of Crescent Engineering for providing the derivation for the quadratic interpolator as well as for suggesting its use for this application. Thanks also to Peter Kootsookos of Australia National University, for valuable feedback from reviewing an early draft.

APPENDIX A

The peak position of a sampled quadratic can be located using the sampled peak and the two adjacent samples. For a sampled quadratic function y with the sampled peak at y_m, and the true peak δ samples from m, the three samples about the peak are:

$$y_{m-1} = a \cdot (m - 1 - \delta)^2 + b$$

$$y_m = a \cdot (m - \delta)^2 + b$$

$$y_{m+1} = a \cdot (m + 1 - \delta)^2 + b$$

Setting m = 0 and solving for δ yields

$$\delta = \frac{0.5 \cdot \{y_{m+1} - y_{m-1}\}}{2 \cdot y_m - y_{m+1} - y_{m-1}} \qquad \text{A.1}$$

so that m-δ provides an accurate estimate of the true peak location of a sampled real quadratic.

A simple adaptation provides an estimator suitable for estimating tone frequencies from complex DFT outputs. Given y_m is the maximizer of the DFT output magnitude

$$\delta = \mathrm{Re}\left\{\frac{y_{m+1} - y_{m-1}}{2 \cdot y_m - y_{m+1} - y_{m-1}}\right\} \qquad \text{A.2}$$

then m-δ provides an accurate estimate of the true tone frequency, in real bin numbers.

APPENDIX B

Frequency-domain interpolation with zero-padded DFTs can be reduced to a dot product operation if only a single interpolation point is needed. For a length NS complex frequency domain vector, a length NL vector can be generated by taking an inverse DFT, zero padding to length NL, and then taking a forward DFT of the zero-padded time-domain vector. Looking first at the second DFT, an expression for the reduced transform can be written in terms of the input samples. For simplicity, the DFT scale factors will be ignored for the moment. Assuming NL is 8, the index, k, of the center output is 3 (numbering from 0-7). The zeroes do not contribute to the output, so the index, i, does not need to exceed the length of the data, NS. Letting SI be the interpolated sample and SZ be the zero-padded input to the second transform and ignoring DFT scale factors,

$$SI_k = \sum_{i=0}^{NS-1} SZ_i \cdot e^{\frac{j \cdot 2 \cdot \pi i \cdot k}{NL}} \qquad \text{(B.1)}$$

Expanding the summation and formalizing k, NL, and NS yields, after simplification

$$SI_3 = SZ_0 \cdot e^{j \cdot 0} + SZ_1 \cdot e^{\frac{j \cdot 3 \cdot \pi}{4}} + SZ_2 \cdot e^{\frac{j \cdot 3 \cdot \pi}{2}} + SZ_3 \cdot e^{\frac{j \cdot \pi}{4}} \qquad \text{(B.2)}$$

Letting SP be the original peak samples to be interpolated, B.2 can be rewritten in terms of SP as

$$SI_3 = e^{j\cdot 0}\cdot\sum_n SP_n\cdot e^{-j\cdot 0} + e^{\frac{j\cdot 3\cdot\pi}{4}}\cdot\sum_n SP_n\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot n}{4}} + e^{\frac{j\cdot 3\cdot\pi}{2}}\cdot\sum_n SP_n\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot 2\cdot n}{4}} + e^{\frac{j\cdot\pi}{4}}\cdot\sum_n SP_n\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot 3\cdot n}{4}} \quad \text{(B.3)}$$

The exponentials may be combined in an expanded form of B.3 to provide a clear picture of the estimation process. Expanding B.3 and simplifying yields

$$\begin{aligned} SI_3 = {} & SP_0\cdot\left\{e^{j\cdot 0} + e^{\frac{j\cdot 3\cdot\pi}{4}} + e^{\frac{-j\cdot\pi}{2}} + e^{\frac{j\cdot\pi}{4}}\right\}\ldots \\ & + SP_1\cdot\left\{e^{j\cdot 0} + e^{\frac{j\cdot\pi}{4}} + e^{\frac{j\cdot\pi}{2}} + e^{\frac{j\cdot 3\cdot\pi}{4}}\right\}\ldots \\ & + SP_2\cdot\left\{e^{j\cdot 0} + e^{\frac{-j\cdot\pi}{4}} + e^{\frac{-j\cdot\pi}{2}} + e^{\frac{-j\cdot 3\cdot\pi}{4}}\right\}\ldots \\ & + SP_3\cdot\left\{e^{j\cdot 0} + e^{\frac{-j\cdot 3\cdot\pi}{4}} + e^{\frac{j\cdot\pi}{2}} + e^{\frac{-j\cdot\pi}{4}}\right\} \end{aligned} \quad \text{(B.4)}$$

It can be seen by the simplified expression for SI that the process is reduced to an NS-point dot product of SP with an interpolating vector. The interpolating vector is determined by combining the exponential terms from the two DFTs used with the zero-fill method.

The example case deals with a 1:2 interpolation from a four-point input vector. Generalization for arbitrary NS, arbitrary NL, and arbitrary k is easily accomplished. Freedom in selection of NL and k allows essentially infinite freedom in selection of the location of a point to be interpolated within the input vector. Practically, granularity of output sample location is driven by the numerical precision of the implementation.

$$SI_k = e^{j\cdot 0}\cdot\sum_n SP_n\cdot e^{-j0} + e^{\frac{j\cdot 2\cdot\pi\cdot k}{NL}}\cdot\sum_n SP_n\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot n}{NS}} + e^{\frac{j\cdot 2\cdot\pi\cdot k\cdot 2}{NL}}\cdot\sum_n SP_n\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot n\cdot 2}{NS}} + \ldots + e^{\frac{j\cdot 2\cdot\pi\cdot(NS-1)\cdot k}{NL}}\cdot\sum_n SP_n\cdot e^{\frac{-j2\cdot\pi\cdot n\cdot(NS-1)}{NS}}$$

(B.5)

Equation B.5 is the generalization of equation B.3. Repeating the example procedure for the general case, combining exponentials for like SP_n produces an expression for each element of the interpolating vector M, equation B.6 . It can be seen that there are always NS exponentials contributing to each element of M, and M contains NS elements. For the dot product of the original vector and M to be a unity-gain operation, the coefficients must be normalized. The normalization replaces the scale factors used in the zero-fill interpolation method. Since there are NS unity magnitude elements contributing to M_n, scaling M_n by 1/NS will produce an interpolating vector with unity gain. Equation B.7 shows the reduced expression for M_n with the scale factor.

$$M_n = \frac{1}{NS} \cdot \left[e^{j0} + e^{j \cdot 2 \cdot \pi \cdot \left(\frac{k}{NL} - \frac{n}{NS} \right)} + e^{j \cdot 2 \cdot \pi \cdot \left(\frac{k2}{NL} - \frac{n \cdot 2}{NS} \right)} + \ldots + e^{j \cdot 2 \cdot \pi \cdot \left[\frac{k(NS-1)}{NL} - \frac{n \cdot (NS-1)}{NS} \right]} \right] \tag{B.6}$$

$$M_n = \frac{1}{NS} \cdot \sum_{i=0}^{NS-1} e^{j \cdot 2 \cdot \pi \cdot i \cdot \left(\frac{k}{NL} - \frac{n}{NS} \right)} \tag{B.7}$$

REFERENCES

[1] B. G. Quinn, "Estimating Frequency by Interpolation Using Fourier Coefficients," *IEEE Trans. Signal Processing*, Vol. 42, pp. 1264-1268, May 1994.

[2] Thomas Grandke, "Interpolation Algorithms for Discrete Fourier Transforms of Weighted Signals," *IEEE Trans. Instrumentation and Measurement*, Vol. IM-32, pp 350-355, June 1983.

[3] V.K. Jain et al, "High-Accuracy Analog Measurements via Interpolated FFT," *IEEE Trans. Instrumentation and Measurement*, Vol. IM-28, pp 113-122, June 1979.

[4] D. C. Rife and R. R. Boorstyn, "Single-Tone Parameter Estimation from Discrete-Time Observations," *IEEE Trans. Information Theory*, Vol. IT-20, pp 591-598, September 1974.

[5] B. G. Quinn and P. J. Kootsookos, "Threshold Behaviour of the Maximum Likelihood Estimator of Frequency," *IEEE Trans. Signal Processing*, Vol. 42, pp 3291-3294, November 1994.