Cosine Filter Theory

R. W. Wall

Fifty years ago, electromechanical protective relays were used almost exclusively. In general, these devices use torque generated by AC currents to magnetically close a set of mechanical contacts. The contacts are held open or “restrained” by a mechanical spring much like the common circuit breaker we use in our homes. In reality, these devices were vastly more complicated. Frequently the phase relationships of currents and voltages allowed the relay to determine the direction of the fault relative to the relay. This makes the relays selective resulting in deenergizing only the parts of the power grid that are absolutely necessary to isolate the faulted section.

a. Computer entry into the power protection

Computer based relaying was experimented with in the 50’s and 60’s but were not commercially viable because of the size, expense, and reliability of early computers. Along came the 70’s and the microcontroller revolution, which of course, changed all of that. Believe it or not, the microprocessor-based relays used those same torque relationships that their predecessor electromechanical relays used. Instead of using magnetic flux to generate the torque, the microprocessor relay computed the torque. Modern microprocessor-based relays still use the magnitude and phase of the 60 Hz (50 Hz in Europe) fundamental power voltage and currents to compute torque like quantities that determine the existence of faults.

One of the challenges then (and continues to be today) is how to reliably and efficiently convert sampled analog signals to magnitude and phase needed for the torque equations. Since the fundamental component is the only signal of interest, all other signals, whether they be harmonics, arcing noise or transients generated by exciting the natural modes of the electrical network, are considered noise that corrupts the signal of interest. Schweitzer and Hou[i] reviewed seven of the more common algorithms used to convert a time sequence to a time varying complex vector. Only two of these algorithms are based on orthogonal basis set decomposition similar to the Fourier transform. Discussion in the article will be limited to only one of these algorithms because of its simplicity, efficiency, and performance, the DFT.

b. Switching domains

The discrete Fourier transform (DFT) is a digital filtering algorithm that computes the magnitude and phase at discrete frequencies of a discrete time sequence. Fast Fourier transforms are computationally efficient algorithms for computing DFTs. FFTs are useful if we need to know the magnitude and/or phase of a number individual or band of frequencies. Jack Crenshaw told us all about Fourier transforms, DFTs and FFT in previous ESP in a series of articles spanning Oct. ’94, through Mar. ’95. But DFTs are simply FIR digital filters and Crenshaw told us “more about filters” (June ‘96), “filters, the *very* last word” (Sept. ’96) and “filters – a few more words” (Nov. ’96). After Jack said all there was to say, Don Morgan told us about the “fundamentals of FIR design” in a sequence of four ESP articles starting in June ’97. With that plethora of background information, we can jump right into the application.
For protective relaying, we don’t really need to extract the magnitude and phase of every signal contained in the sampled data stream, as is the case of an FFT. This is especially true if there is only one signal present to begin with. If we use a DFT for the only signal we’re interested in, we have both a conversion algorithm and a band bass filter.

c. Theory to application

So let’s try out the theory and see how it works. Lets assume that we are dealing with a 60 Hz signal that is sampled synchronously. This means that the sample interval is the inverse of an integer multiple of 60. We need to compute the DFT for the fundamental using equation (1) where *N* equals to the number of samples per fundamental cycle, *k* equal to one for the fundamental, and *n* is the coefficient subscript. Because digital computers (like most of the world) don't really understand the concept of imaginary numbers, two digital filters are required, one to get the real part and one for the imaginary part. Mathematically, the coefficients of these filters are by determined using (2).

$$Ck_n = e^{-j\left(\frac{2\pi nk}{N}\right)}$$

$$Ck_n = \left(\frac{2}{N}\right)\left[\cos\left(\frac{2\pi nk}{N}\right) + j\sin\left(\frac{2\pi nk}{N}\right)\right] = Ak_n + jBk_n$$

After computing the outputs of two filters using equation (3), we have the desired complex vector shown in (4). Remember from a distant past math class on complex variables that addition of complex numbers is easiest using rectangular notation while multiplication is easiest using polar notation shown in (5). For real-time applications, the conversion back and forth between the two notations usually requires more time than can possibly be gained. Hence, the complex variables are usually exclusively dealt with using rectangular form until such time as a magnitude or phase is explicitly needed. This is particularly true for processors that must use software routines for computing transcendental functions (trig, log, and exponential functions). To further increase speed, magnitude threshold levels are frequently left squared and angles kept as ratios.

$$Y_n(re) = \sum_{0}^{N-1} A_n X_n, \quad Y_n(im) = \sum_{0}^{N-1} B_n X_n$$

$$Y_n = Y_n(re) + jY_n(im)$$

$$Y_n RMS = \sqrt{Y_n(re)^2 + Y_n(im)^2}, \angle Y_n = \arctan\left(\frac{Y_n(im)}{Y_n(re)}\right)$$

d. Transition phase

The conversion process works fine when every thing is at steady state – amplitude, frequency, and phase is held constant or is changing very slowly compared to the frequency of interest. But faulted power systems happen in a flash (pun intended) and these faults can be modeled as step changes as illustrated in Figure 1. During the transition period, the DFT output changes at each sample point until the algorithm processes a complete cycle's worth of steady state data. In this period, the algorithm-generated transient makes the DFT results an inaccurate representation of both the magnitude and the phase (Figure 2) for a signal that has already achieved steady state. Figure 1 also illustrates a point made earlier. That being a whole cycles worth of the steady data must be sampled and processed by the DFT before steady state is achieved. This is true regardless of the sample rate. I will cover more on this later. Another observation from Figure 1 is the magnitude scaling, which for this case, it is not RMS but peak. To obtain the RMS value, you simply make the multiplier in (2) *√2/N* instead of *2/N*.

Looking at the phase output in Figure 2 is sure cause to wonder of what value is it. Very little, in of itself, because phase has no value without a time or phase reference. The difference between the two DFT phase results accurately represents phase relationship of the two signals if they are at the same frequency and sampled simultaneously. The phase steps ahead, as is seen in Figure 2, each time a new sample is processed. The value of N or the size of the DFT determines the size of the phase step as shown in (6). The absolute value of the phase at any time is arbitrary unless the samples are somehow synchronized with a signal's zero crossing.

$$\Delta\phi = \left(\frac{2\pi}{N}\right)$$

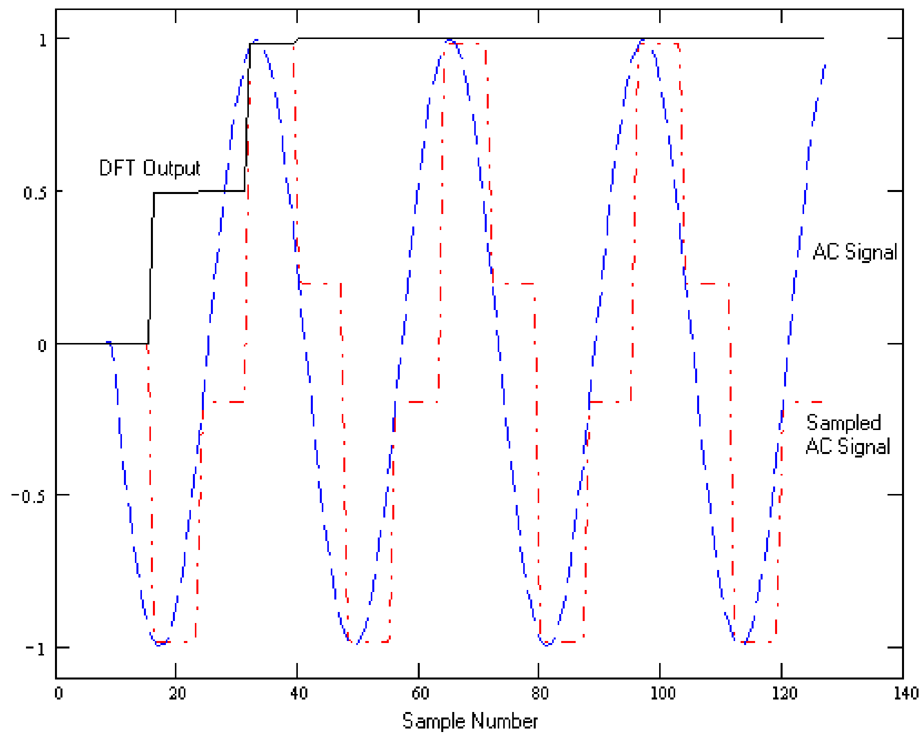


Figure 1. Sampled sine wave with four-point DFT magnitude response to a step change.

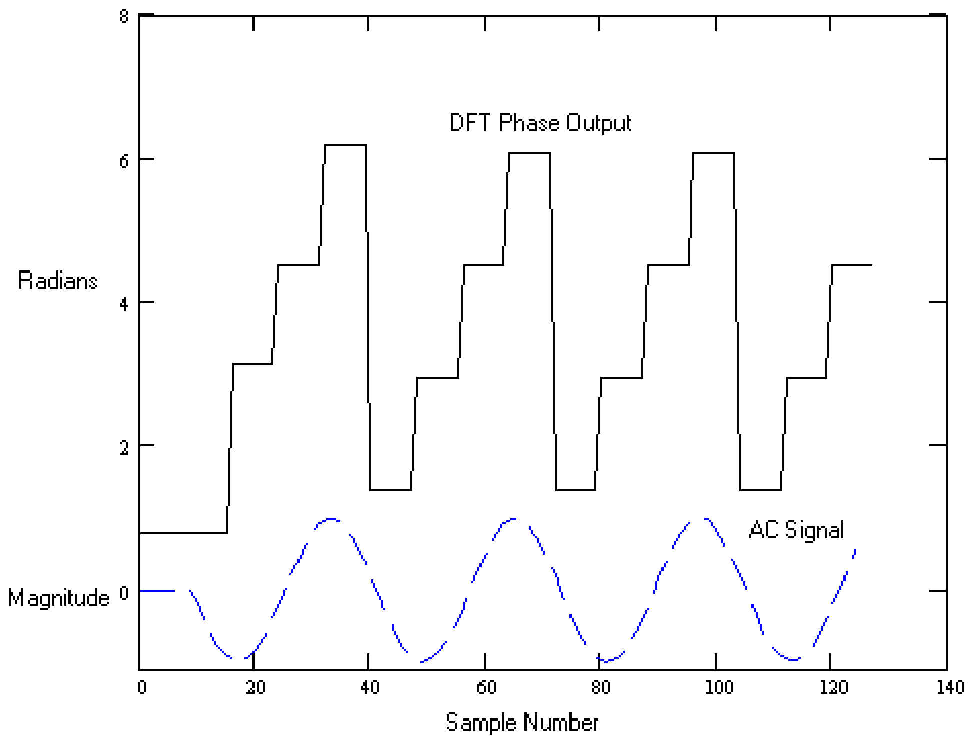
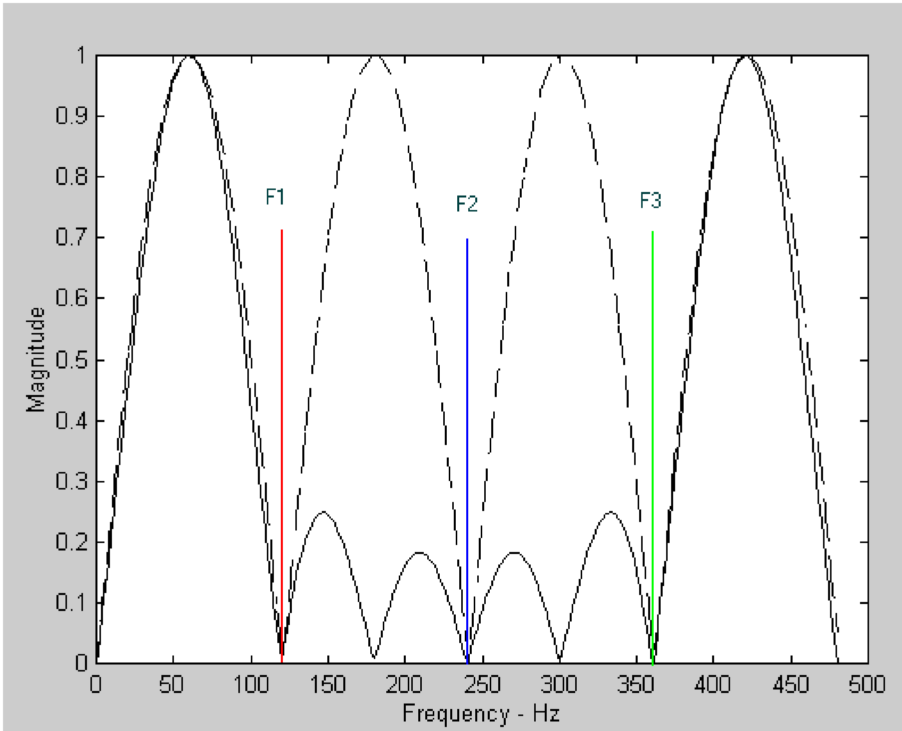


Figure 2. DFT Phase response to a step input sine wave.

e. Frequency Response of DFTs

FFTs represent the spectrum of the sampled data with a set of discrete frequencies evenly spaced between zero and half the sampling rate, *FS*, minus one interval. The interval between the discrete frequencies is equal to the *FS/2N* where *N* the size of the DFT or the number of coefficients determined by (1). Figure 3 shows the frequency response of DFTs over the range of zero to 480 Hz for N equal to four and eight. This figure also shows the effects of aliasing around the fold-over frequency F1, F2, and F3 for the four-point filter sampled at 240 Hz and just F2 for the eight-point filter sampled at 480 Hz. The advantage of the eight-point filter is that the filter response is zero at harmonics except the fundamental and the $(N-1)^{th}$ harmonic. Odd harmonics of 60 Hz are of particular concern because they are generated by power transformers saturation and nonlinear loads such as switching power supplies.



D

Figure 3. Frequency response of a 4 and 8 point DFT with sampling rate of 240 Hz and 480 Hz respectively.

A colleague once commented that when your only tool is a hammer, every thing tends to look like a nail. The same is true here. The DFT we are using is looking only for 60 Hz and any energy that is passed by the filter characteristics, regardless of the actual frequency, is aliased to appear that it is energy at 60 Hz. But in this case, aliasing is our friend as well as our nemesis. The zeros at harmonics on the high side of the Nyquist frequency[ii] (*FS/2*) work to our benefit.

f. Points to ponder

At this point we can draw some conclusions. When it comes to DFTs, more is not necessarily better, sometimes it's jut more. Higher order DFTs provide greater harmonic rejection but do not decrease the algorithm transient time. If you don't believe me, look at Figure 4, which also bears out the claim that a complete cycle of the fundamental must be sampled to achieve steady state results. However, if we expect harmonics, then clearly higher order filters will help reduce the both magnitude and phase errors.

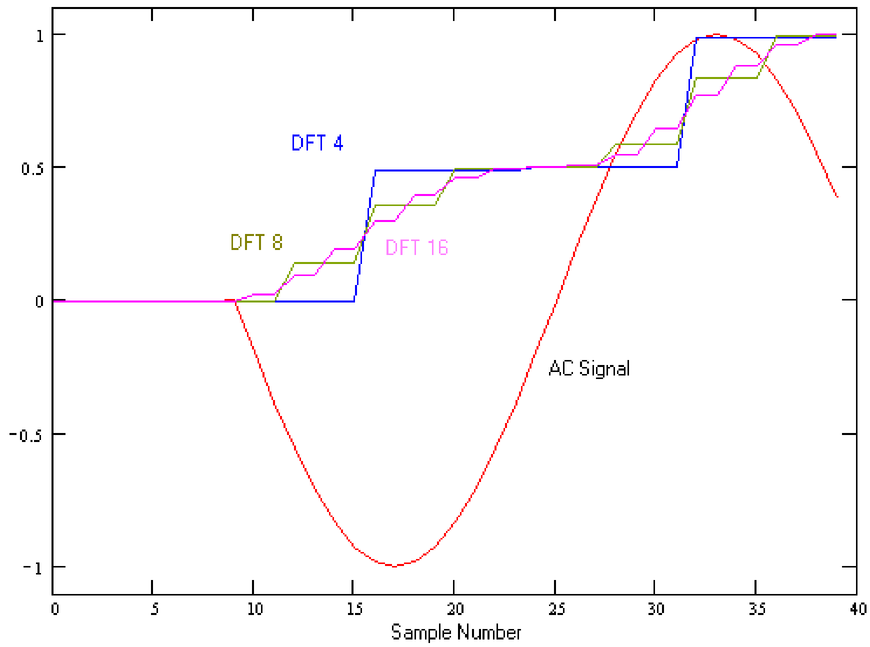


Figure 4. Magnitude responses for a 4, 8, and 16 point DFT to step change of sine wave amplitude.

What if the signal being sampled is not exactly at 60 Hz? Well, the magnitude will change according to the filter response attenuation for that frequency as shown in Figure 3. DFT magnitude will not change much if the frequency is not too far off. We will notice that the frequency increments no longer adhere to (6). In fact, the difference in phase increment is exactly proportional to the frequency difference. Say that the actual frequency is 59 Hz. This means that the frequency difference is 1 Hz or 2π radians / second. If the sampling rate is 240 Hz, then the phase shift will be off by (2π/240) radians / sample.

We can use this algorithm another way too and then it becomes a frequency meter! Say we know that the system is operating in steady state and we calculate the measured phase step (from the DFT output) and the expected phase step according to (6). Then the actual frequency of the sampled signal is the fundamental frequency ± the difference frequency. Mathematically, if it works as shown in (7) through (9). When calculating the phase difference in (7), be sure to consider the case when successive iterations are on opposite sides of the 2π / zero radian boundary. Do not expect the accuracy of such approach to compare favorably with conventional zero crossing detectors.

$$\delta\phi = \left(\frac{2\pi}{N}\right) - \left(\phi_n - \phi_{n-1}\right)$$

$$\delta f = \delta\phi\left(\frac{Fs}{2\pi}\right), \text{ where } Fs \text{ is the sample rate.}$$

$$f_{ACTUAL} = \delta f + f_0$$

g. The fly in the ointment

With the transient response time fixed by the fundamental frequency and the errors resulting from harmonics eliminated by the zeros of the DFT filter response, what's left? Usually higher harmonics and high frequency oscillations due to exciting natural resonance modes in the power system network are removed by analog filtering prior to sampling. Even though the DSP filter has a zero at DC, power system faults frequently generate other low frequency components commonly called DC offset. It is not really DC but a slowly decaying exponential superimposed on the AC signal as shown in Figure 5. Also shown in this figure are the DFT magnitudes for the signal without the superimposed exponential and the offset AC signal. If the DFT for the non-offset AC signal is considered optimal then the other DFT is what we are stuck with.

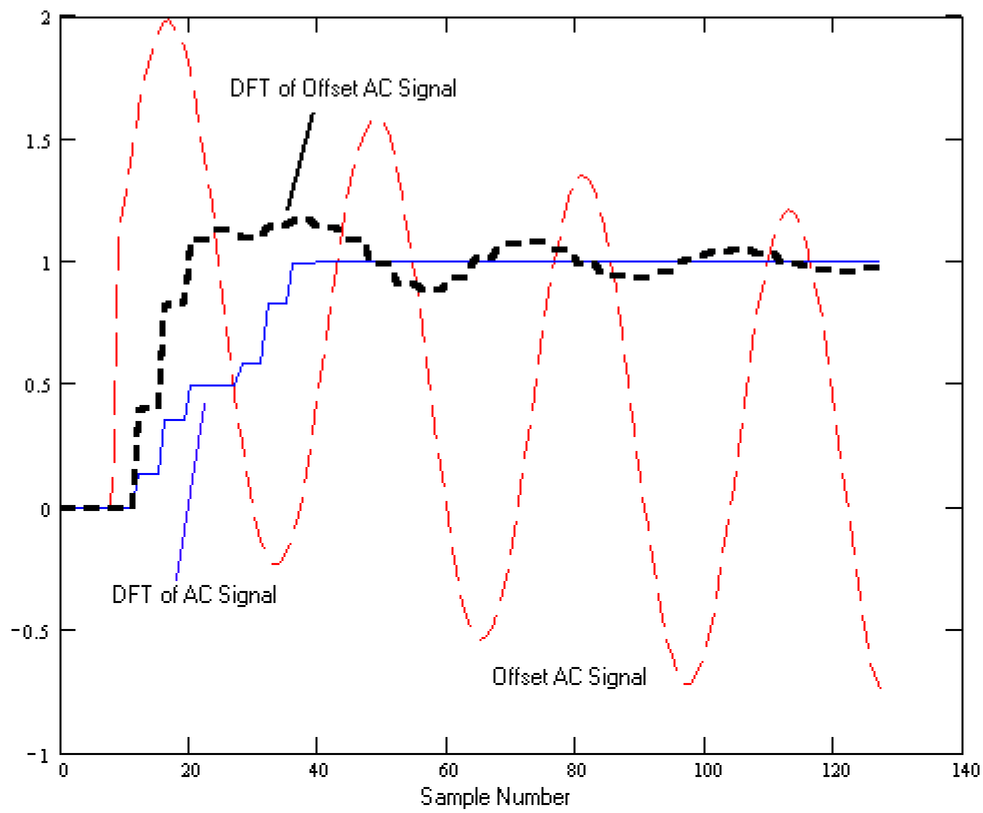


Figure 5. DFT filtering of a fully offset sine wave.

One readily observes that the response of the DFT to the offset sine wave approaches the steady state value sooner and could concluded that the offset is to our benefit. Unfortunately, protective relays could interpret the higher magnitude as a fault that results in an incorrect operation. Today's power systems are operated so closely to designed capability that frequently a slow correct operation is preferable to fast but possibly incorrect operations.   As the engineer, you have a choice to make.

h. Tricks of the trade

For the power industry, it is certainly in their best interest to reduce the response of the DFT due to the offset. One trick frequently employed is to use only the coefficients to compute the real part of the DFT shown in (3) that are generated by the cosine function. This is sometimes called a Cosine filter. Figure 7 shows the frequency response of the Cosine filter compared to the DFT filter.

Note that the Cosine filter favors higher frequencies and attenuates the frequencies close to zero. This is good when trying to filter out a slowly decaying exponential. There is also a computational advantage to eliminating the multiply and accumulate instructions associated with imaginary term.

Note also from Figure 7 that the Cosine filter matches the response on the DFT at 60Hz so there is no amplitude compensation required. However, off-frequency signals will be more affected by the Cosine filter frequency response than for DFT filters. One solution is to adjust the sampling rate to be an integer number of the fundamental. This can accomplished by measuring the period with a zero-crossing detector. Adjustments to the sampling period should be slow so to track only the power system frequency changes and not frequencies generated by transients.[iii]

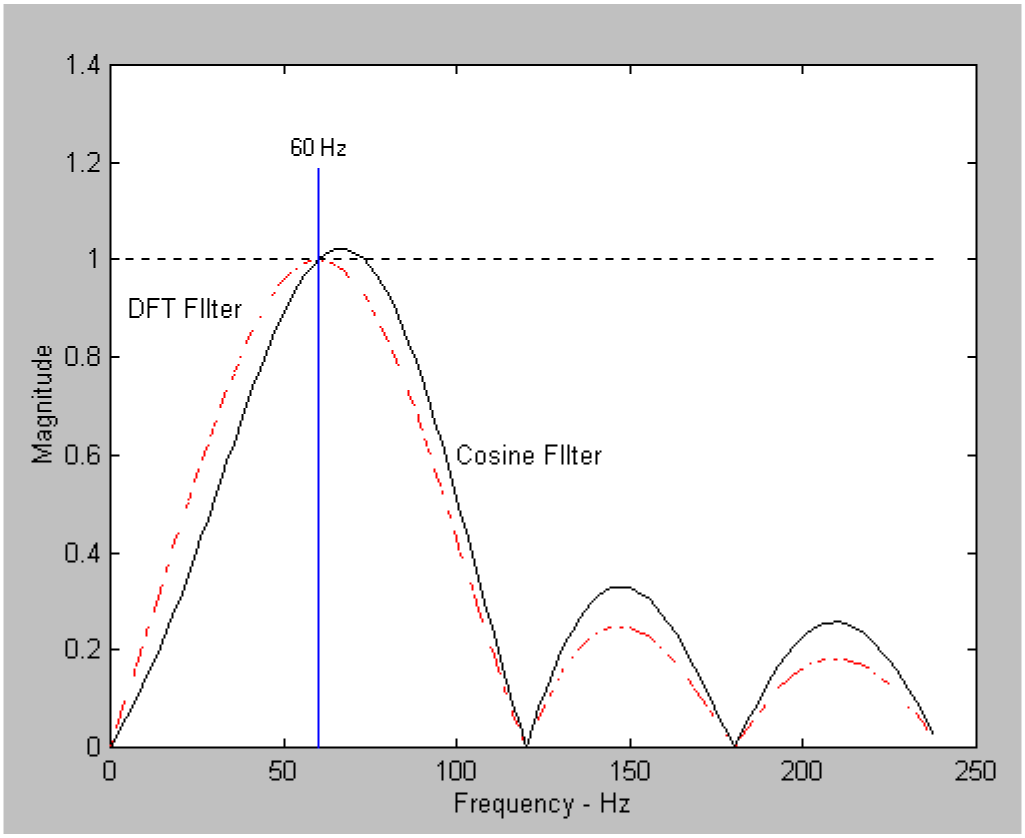


Figure 6. Frequency response of an eight-point Cosine DFT filter

The trick is to use the result generated by the Cosine filter for both the real and imaginary parts of the complex vector. This is accomplished by making the most recent Cosine filter output the real term and the output that has been delayed a quarter of the period of the fundamental the imaginary term as shown in (10) and (11). Both the real and the imaginary terms now have identical frequency responses.

$$Yc_n = \sum_{0}^{N-1} A_n X_n,\ A_n = \left(\frac{2}{N}\right)\cos\left(\frac{2\pi n}{N}\right)$$

$$Y_n = Yc_n + jYc_{n-N/4}$$

Since we know that the DFT of the pure sine wave is the desired output we can make it our evaluation reference. By computing the absolute difference between reference output and the outputs of the DFT filter and the Cosine filter, we can see the improvement. This is done in Figure 7 labeled D1 and D2 respectively. The difference for the Cosine filter response has reduced overshoot and achieves an overall smaller difference. The cost of the improved offset rejection is that the filter transient is extended by the time equal to one quarter of the period of the fundamental. This is not obvious from Figure 5 because it is difficult to differentiate the signal transient from the algorithm transient.

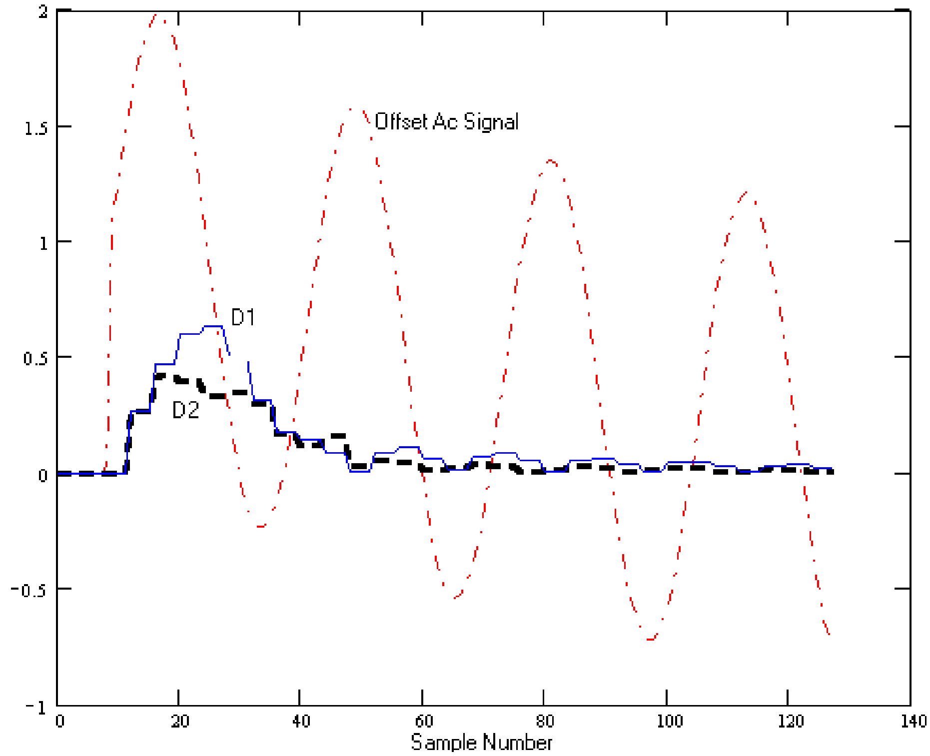


Figure 7. Differences of a DFT and Cosine filter for an offset sine wave compared to a pure DFT for a sine wave without the offset.

i. Wrap up

Remember the leader of the Jodi Foster movie, “Contact”. The camera is supposedly starts at planet earth and the sound track plays what seems to be the simultaneous audio from all radio and TV broadcasts. The camera take a path through inter stellar space that leads further from our reality and all the while the audio becomes more focused on fewer and fewer broadcasts. Finally, we’re left with a single radio transmission of a young girl on a ham radio. I feel that this article has taken a similar path.

The idea here is that we’re interested signals at one frequency only and we needed an algorithm that quickly and accurately computes the magnitude and phase of that signal. We can take advantage of aliasing to cancel harmonics if we don’t expect that the signal will contain energy that is also passed by the aliasing. DSP tricks can improve performance but always come at a price. It is the responsibility of the designer to understand the application sufficiently to know where compromises are tolerable to achieve the desired performance.

j. Final words of caution

The voltages on lines that deliver power make them lethal. Relays cannot operate fast enough to prevent serious injury or death to someone coming in contact with an energized power line. In an emergency situation, never assume that relays have operated and the lines are deenergized. One of the most difficult conditions to detect is a distribution line that broken and fallen to the ground. The fault current is so small that most relays cannot sense the fault. Always assume power lines are energized and treat them accordingly.

[i] E.O. Schweitzer and D. Hou, "Filtering for Protective Relays", 47th Annual Georgia Tech Protective Relay Conference, Atlanta GA. April 28-30, 1993. This article is available for download in PDF format at http://www.selinc.com/techpprs.htm.

[ii] Digital Filtering: an Introduction, Edward P. Cunningham, Houghton Mifflin Co., 1992, ISBN 0-395-53989-7

[iii] R.W. Wall and H.L. Hess, "Design of Microcontroller Implementation of a Three Phase SCR Power Converter", Journal of Circuits, Systems, and Computers, Vol. 6. No. 6. Mar. 1997, pp. 619-633.