Quadrature (Complex-Valued) Signal Processing
A Technical Overview

Jim Shima

Quadrature Signals Redux

Quadrature, or complex-valued, signals refer to waveforms that are deemed "analytic". In this context, an analytic signal is one which has a frequency response containing only positive frequency terms. In this paper, the terms "quadrature" and "analytic" are sometimes used interchangeably.

Recall from Fourier theory, all real-valued signals have "symmetric" positive and negative frequency terms. For analytic signals, the negative portion of the frequency response disappears. Analytic signals are also related to Hilbert transforms, as they can be represented by:

$$x_{quad}(t) = x(t) + j \cdot H\{x(t)\} \qquad (1.1)$$

Where H{x(t)}is the Hilbert transform of the real signal x(t). By definition, the Hilbert transform shifts the input signal 90 degrees for negative frequencies, and -90 degrees for positive frequencies [2]. If x(t) is a sinusoid, you can easily see that the Hilbert transform turns sines into cosines, and vice versa. As such, passing a real-valued signal through a perfect Hilbert transform creates a phase-shifted version of the input. Combining both real-valued signals into a single complex-valued waveform creates an "analytic" (or quadrature) signal.

As an example, let $x(t) = \cos(2\pi f_o t)$. This real-valued signal has a symmetric Fourier transform: $X(f) = \frac{1}{2}[\delta(f - f_o) + \delta(f + f_o)]$. As stated before, X(f) has a positive and negative frequency term, and this is true for any real-valued signal [6].

The Hilbert transform of x(t) is:

$$H\{x(t)\} = \cos(2\pi f_o t - \frac{\pi}{2}),$$

which is equal to $\sin(2\pi f_o t)$. So, our quadrature signal takes on the form:

$$x_{quad}(t) = \cos(2\pi f_0 t) + j\sin(2\pi f_o t).$$

By using Euler's identity $e^{jx} = \cos(x) + j\sin(x)$, we get:

$$x_{quad}(t) = e^{j2\pi f_o t}$$

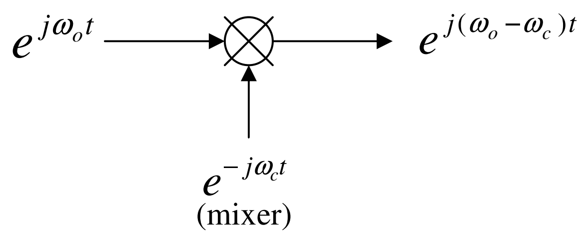
The result is a complex-valued exponential. Recall the Fourier transform of a complex exponential is: $X_{quad}(f) = \delta(f - f_o)$. Hence, our quadrature signal now only has a positive frequency term, the negative frequency term has conveniently "vanished". More on this concept will be presented next.

Quadrature mixing

A common signal processing task is to relocate signals to other frequencies where we can easily process them. Baseband signal processing is the cornerstone of many communication schemes and other DSP algorithms. Here we are attempting to mix the input signals down near DC where we can process them more efficiently.

Quadrature mixing is the process of taking a complex or real-valued discretized input and mixing it with a complex-valued exponential. In this case, we have 2 data streams now, the I (in-phase) stream and the Q (quadrature phase) stream. Each signal stream, taken individually, is a real-valued signal. But when we combine them into a complex-valued signal, we will see some remarkable properties.

Assume we have a complex-valued exponential at frequency $\omega_o = 2\pi f_o$, and a complex exponential mixer at frequency $-\omega_c$. We can show that for a perfect input signal, we are mathematically mixing the quadrature signal to the frequency $(\omega_o - \omega_c)$



(1.1b)

Fig.1 – Complex mixer

We can also assume these signals have already been sampled. So the mixing operation is done after we have digitized each signal.

The mixer is typically implemented as a numerically-controlled oscillator (NCO), which is a fancy name for a type of lookup table that stores the mixer samples we wish to use. NCO analysis is not included in this paper, but suffice it to say that the NCO is simply a table of complex exponential values at frequency ω_c.

A block diagram of a digital complex mixer is shown next.

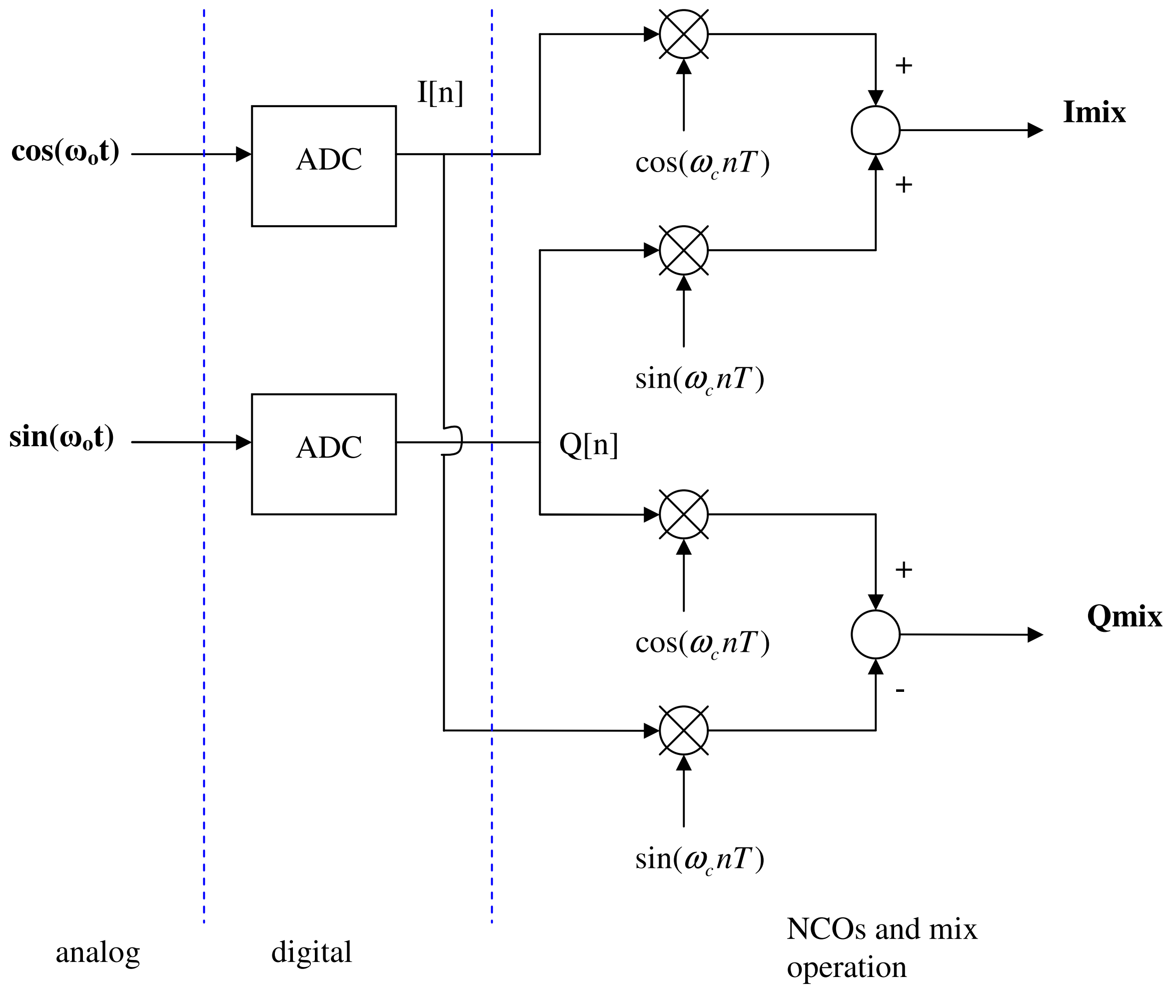


Fig. 2 – Block diagram of quadrature (complex) mixdown

In Fig.2 we have used the notation "nT" as the sampling function, where T = 1/Fs and n is an integer.

Fig. 2 may look complicated, but it is simply the expanded complex multiplication boiled down into real multiply and add operations. Again using Euler's identity:

$$e^{j(\omega_0-\omega_c)t} = [\cos(\omega_o nT) + j\sin(\omega_o nT)][\cos(\omega_c nT) - j\sin(\omega_c nT)] \quad (1.2)$$

Substituting $I[n] = \cos(\omega_o nT)$ and $Q[n] = \sin(\omega_o nT)$ for the input samples, we get:

$$e^{j(\omega_0-\omega_c)t} = \underbrace{I[n]\cos(\omega_c nT) + Q[n]\sin(\omega_c nT)}_{\text{Real part}} + j\underbrace{\{Q[n]\cos(\omega_c nT) - I[n]\sin(\omega_c nT)\}}_{\text{Imaginary part}}$$

And these are indeed the operations shown in Fig. 2.

$$I_{mix} = I[n]\cos(\omega_c nT) + Q[n]\sin(\omega_c nT) \quad \text{“real part”}$$
$$Q_{mix} = Q[n]\cos(\omega_c nT) - I[n]\sin(\omega_c nT) \quad \text{“imaginary part”}$$

It is interesting to note that you can switch between a down mix and up mix (with or without a phase shift) just by changing the sign of the additions in the above equations.

Quadrature Signaling vs. Nyquist

It is commonly mentioned in DSP literature that utilizing a complex-valued signal relaxes the Nyquist sampling frequency by two. That is, the folding frequency for a complex-valued signal is now Fs, compared to Fs/2 for real-valued signals. Many engineers take this concept at face value, but the underlying concept brings up several issues that usually leaves one scratching their head. Some commonly asked questions include:

1) Why isn't the sampling theorem violated since the two signals are real-valued?
2) What is the sampling rate for my incoming real signals?
3) Why can I run my ADCs at half the sample rate but still represent signals up to Fs?

This paper attempts to answer these questions with satisfaction. Throughout the years I have heard some engineers claim "Using complex signals mean your new sample rate is really 2Fs". Others have said "Complex sampling works because you have twice the data from I and Q, so you have twice the bandwidth". These are misnomers that are not far off the mark, but are not entirely true either. It usually leads to hand waving and a leap of faith in understanding the root issues. Hopefully the following discussion will lead to a more standardized nomenclature and understanding of the material.

Most DSP engineers know that when they are processing complex signals, the I and Q streams go through parallel linear operations, i.e. filters, multipliers, adders, etc. For example, if we are filtering a quadrature signal, then the I and Q real-valued signals are individually processed through identical filters. The outputs are still real-valued, but somewhere in the process we "combine" the two real signals into a complex one. The identical processing paths are there to retain the relative phase and amplitude relationship between the signal pairs. Basically, if you do something to the I channel, you want to do the same to the Q channel in order to uphold the quadrature signal definition.

Firstly, from question 1) the sampling theorem is not violated in the standard case. If we assume both real-valued signals (I and Q) are band-limited to Fs/2 and are sampled at frequency Fs, then we meet the classic sampling theorem we all know and love on the 2 real-valued signals. The complex-valued signal will be able to represent frequencies up to Fs – more on this later.

But let's say that we band-limit the incoming real-signals to Fs and still keep our original sampling frequency. What happens now? We obviously know that the two real-valued signals are now aliased since we have violated the sampling theorem.

Quadrature Alias Cancelation

To understand the above scenario, we introduce the concept of alias cancelation. This term is used in multirate signal processing literature while discussing perfect reconstruction (QMF) type filters [3]. Here we use a similar concept to explain quadrature signaling.

In Fig 2., the Imix and Qmix signal paths are real-valued signals (taken independently). If they contain frequencies above Fs/2, then we expect aliasing to happen in each signal. And indeed it does. If we were to process these 2 signals independently, there is no way we can undo the aliasing. We are stuck as we would be in any other real sampled system where aliasing occurs.

But we have quadrature processing to the rescue! Only when the two real-valued signals are combined together into a complex-valued signal do we see all the underlying alias copies "disappear". This reveals the fact that even though the two real-valued signals are aliased, we can still reconstruct a complex-valued (analytic) signal that has a sample bandwidth up to Fs with no aliasing!

<u>From concept to understanding</u>

Let's take a simple example of a complex-valued analytic sinusoid where I[n] = $\cos(\omega_o nT)$ and Q[n] = $\sin(\omega_o nT)$. Also, lets assume the input frequency of the sinusoid f_o is above Fs/2, in other words we purposefully alias the two real-valued signals.

The spectrum of each step is shown in Fig. 3. Recall after we have sampled the analog signal, we will see copies of the analog spectrum spaced every +/-k*Fs Hz in the spectrum, where k is an integer.

In Fig. 3(a), the spectral copies of I[n] (due to sampling an analog signal) at f=0,Fs,-Fs are shown. The other spectral images are not included for clarity. Note that since we have guaranteed aliasing by choosing a input freq greater than Fs/2, we will see the alias of the original signal at frequencies ($Fs-f_o$) and $-(Fs-f_o)$ in Fig 3(a)-(b).

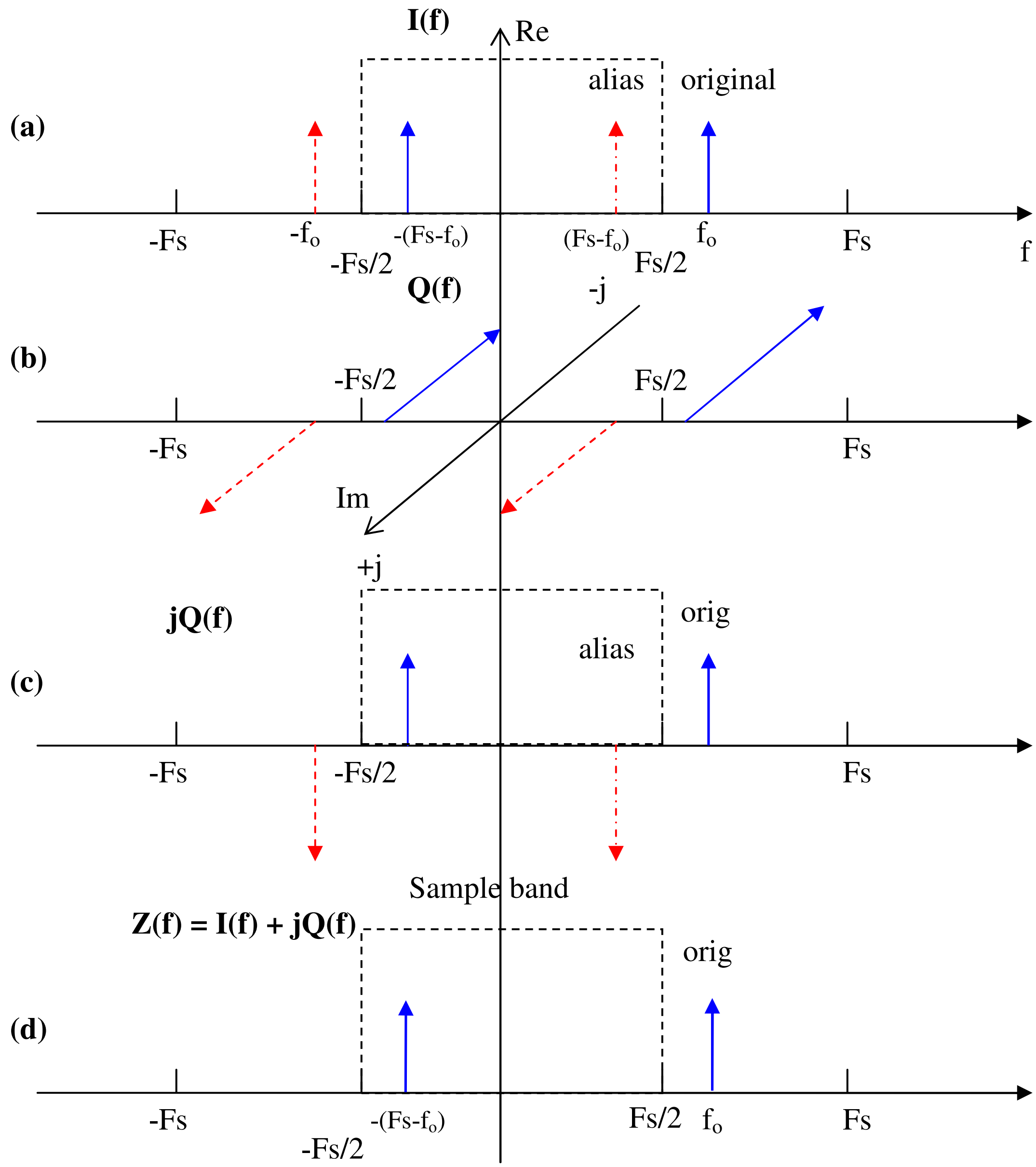


Fig. 3 – Spectral description of quadrature alias cancelation

In Fig 3, step(a)-(b) shows the spectrum of the cos and sin signals, respectively. This is easily verified using the Euler identities [1],[4]:

$$\begin{aligned} \cos(\omega t) &= \frac{1}{2}\left(e^{j\omega t} + e^{-j\omega t}\right) \\ \sin(\omega t) &= \frac{-j}{2}\left(e^{j\omega t} - e^{-j\omega t}\right) \end{aligned} \qquad (1.3)$$

The frequencies in our "sample band" of [-Fs/2, Fs/2] are shown to be alias frequencies of (Fs-fo) and –(Fs-fo). In other words, these are the frequencies we would "see" in the digital samples due to our aliasing. If we chose to process the real-signals separately, we could not tell the true input frequency due to this aliasing.

In Eq (1.3), note that the spectrum of a sin is purely imaginary. Here we have shown this on the imaginary "j" axis in Fig. 3(b).

In Fig. 3(c), the first step of creating the complex-valued signal is shown. Here, we are defining Q(f) to be the imaginary part of the complex signal. Thus, we multiply Q(f) by "j" as shown in 3(c). This multiplication rotates Q(f) by 90 degrees – placing it on the real axis. Note that this is really a manifestation of complex math. The multiplication by "j" re-defines mathematically the Q signal via a rotation. We are not *physically* rotating the signal, but *mathematically* doing so in order to construct a tractable solution. Hence we are choosing to view the two real-valued signals as one complex-valued signal.

In Fig 3(d). we combine both signals into a complex-valued signal Z(f) = I(f) + jQ(f). Here is where the magic begins. By adding Fig. 3(a) with Fig. 3(c), we combine the two real-valued signals into a complex representation. The result - we see the red "dotted" alias copies subtract away! And the new complex-valued signal is analytic.

One can appreciate the remarkable result here. Even though we started out with two aliased real-valued signals, we were able to combine them into a complex-valued signal that canceled the alias images. The beneficial side effect of this cancelation is that we are able to represent complex-valued signals up to frequencies of Fs. All of this is possible because the complex signal is analytic and is no longer symmetric. One can see the negative frequency term has vanished, thus it is not taking up bandwidth from [-Fs/2,0] as it would in a real-valued signal.

<u>Viewing the signal in the sample band</u>

In this explanation the original input freq was > Fs/2. In Fig 3(d), we can see in our sample band [-Fs/2, Fs/2] that we will detect a negative frequency. In other words, the frequency we see in our discrete-time samples is actually a sinusoid at $-(Fs–f_o)$ Hz. Why?

Recall that we can only see into the spectral "window" (known as the sample band) between [-Fs/2, Fs/2]. This is an artifact of the sampling theorem (everything outside of this folds back down into the sampling band). Here we are "seeing" the analytic signal copy from –Fs, which shows up in our sample band. The original frequency at f_o is outside our sample band, but we still get one of its spectral copies that falls into our sample band. Plus the effective bandwidth of the signals we can uniquely determine is still Fs Hz.

Now, this is only of concern if we were attempting to determine the frequency of the complex-valued signal. This has no effect on the Nyquist relaxation criteria and is strictly an artifact of complex signaling. We still get one frequency only in the sample band (instead of 2 in the real signal case), and we know it to be $–(Fs-f_o)$. Thus we can still solve for f_o since we inherently know Fs.

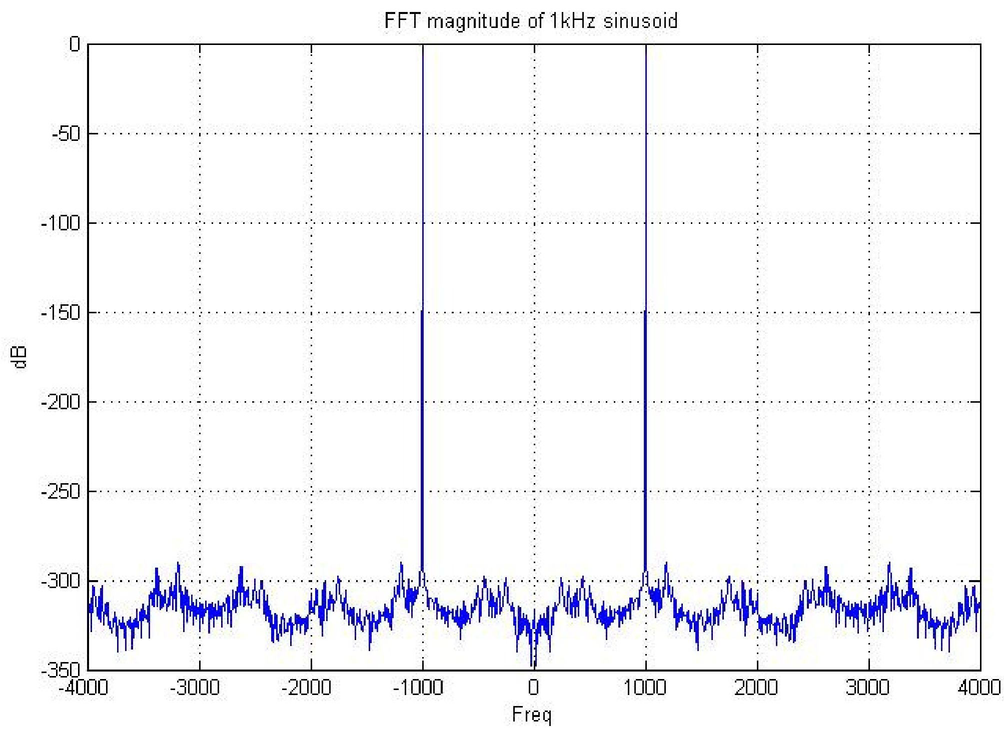
The net effect is this: For any two real signals (in quadrature) with frequencies between [0, Fs/2], the complex-valued spectrum will have a positive frequency impulse equal to f_o. For any two real signals (in quadrature) with frequencies between [Fs/2, Fs], the complex-valued spectrum will have a negative frequency impulse equal to $–(Fs-f_o)$. The important fact here is that we can still uniquely determine f_o up to frequencies equal to the sampling frequency of Fs.

Simple example of a powerful concept

To drive the above explanation further, let's try an example. From DSP theory we know the DFT of any signal returns discrete frequency coefficients that represent in the input signal frequency content from [-Fs/2, Fs/2].

Let's take a sinusoid at f_o=1kHz sampled at Fs=8kHz. We perform an FFT on this sampled signal using MATLAB and take a look at the frequency response:

```
>> x=cos(2*pi*1000/8000*[0:N-1]);
>> X=fft(x);
>> f=[0:N-1]*8000/N - 4000;
>> plot(f,db(1/N*abs(fftshift(X))))
```

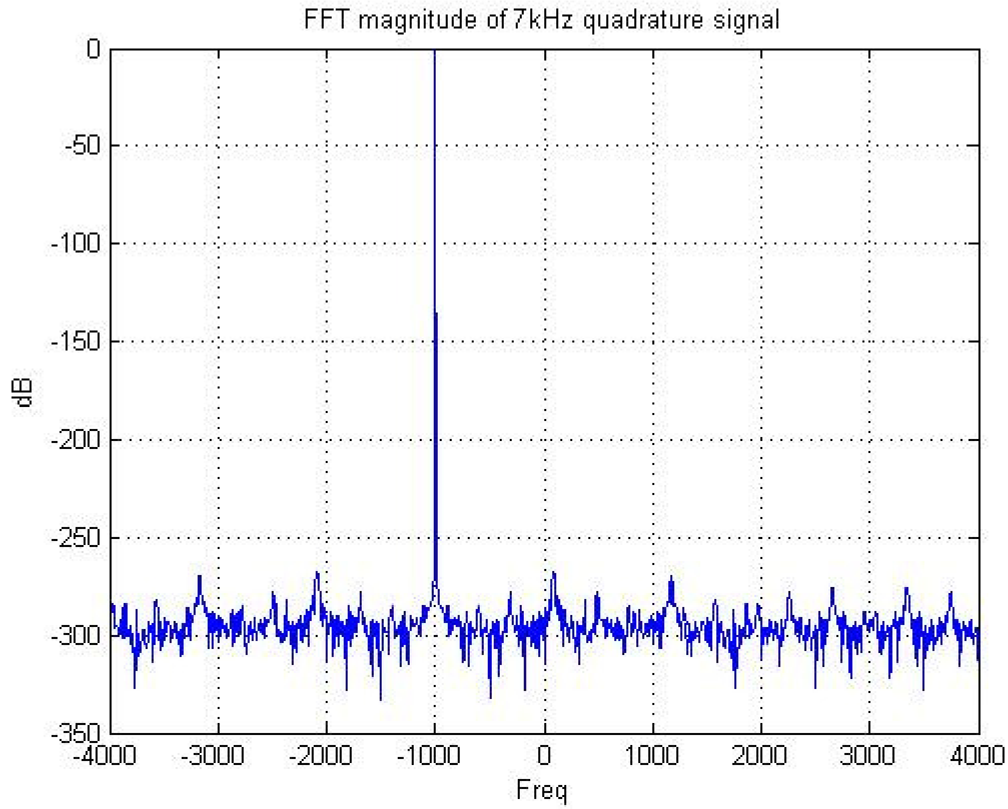


As expected, we see two frequency impulses at -1kHz and +1kHz for our real-valued sinusoid. These are the only frequencies present in our [-4kHz, 4kHz] sample band. The signal is also not aliased since we readily meet the sampling criteria.

Now consider a quadrature signal at f_o=7kHz sampled at Fs=8kHz. Obviously, each real signal is aliased since f_o >Fs/2. As demonstrated in Fig. 3, we should expect to see an analytic signal show up in our sample band at –(Fs-f_o), which is equal to -1kHz.

```
>> x=cos(2*pi*7000/8000*[0:N-1]);
>> y=sin(2*pi*7000/8000*[0:N-1]);
```

```
>> z=x+sqrt(-1)*y;
>> Z=fft(z);
>> plot(f,db(1/N*abs(fftshift(Z))))
```



In the above figure, we can indeed see we have a single complex exponential at -1kHz.

The above MATLAB commands look simple, but the underlying alias cancelation that happened in Fig.3 was never truly seen. It all took place automatically! By referring to Fig. 3, we can appreciate the mathematical steps that took place to create the final analytic signal at -1kHz. And this is typically something that DSP engineers overlook (and for good reason!).

Once you understand the concepts presented, then creating more complicated and advanced quadrature systems is not very hard to accomplish.

Given the basis of quadrature signaling then we should be able to answer the original questions:

1) Quadrature signaling effectively allows us to relax the Nyquist folding frequency to Fs for analytic signals compared to the typical Fs/2 for real-valued signals.
2) This means if you are processing complex signals that have frequency content up to Fs, the ADCs only have to sample at Fs rather than 2Fs (since we can cancel the aliasing in the 2 real signals). In other words, the ADCs only have to run at half the rate of a commensurate single real-valued sampling system.

Conclusion

This article focused on quadrature signaling and the definition of complex-valued signals with respect to real-world inputs. It was shown how real signals are combined into complex signals, and the folklore behind the advantages of quadrature signaling were methodically proven.

References

[1] R. Lyons, *Understanding Digital Signal Processing,* Prentice-Hall PTR, Upper Saddle River, NJ, 2001, Appendix C
[2] S. Mitra and J. Kaiser, *Handbook for Digital Signal Processing*, John Wiley & Sons, New York, 1993, Ch. 13
[3] P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Upper Saddle River, NJ, 1993
[4] R. Churchill and J. Brown, *Complex Variables and Applications*, 5th edition, McGraw-Hill, 1990