On A Method for Automatic Image Balancing in IQ Mixer Based Software Defined Receivers

By Bob McGwier, N4HY

Submitted to 2007 Digital Communications Conference

Over the past four years while working on a software defined radio core, DttSP, with Frank Brickle, and applying it to real world I/Q mixer based radios (2-5), the problem of find the right algorithm to insure equal amplitude and the proper phase relationship to achieve perfect quadature between the I and Q channels in these radios has been a source of constant work. My favorite author of for sale amateur radio software is Alex Shovkoplyas, VE3NEA(1). He has been doing wonderful work in many areas but specifically his work in software defined radio has been inspirational. He did an automatic image cancellation algorithm for his Rocky SDR as well as other interesting software defined radio and dsp work (6,7). His web page gives a tantalizing hint at the basis for the algorithm but does not express a complete derivation of the detector or show how to apply the results in the end. We will do that here.

A Perfect Tone and Receiver

Consider a tone received in a perfectly balanced I/Q radio and let us digitize this incoming signal. The mathematical representation for this digitized signal will be

$$A_n(\cos(2\pi f t_n + \phi) + j\sin(2\pi f t_n + \phi)) + N_t = A_n \exp(jft_n + \phi) + N_n \qquad (1.1)$$

Here n is the n-th sample, A_n is the amplitude of the received signal at the sampling time $t_{n,}$ and ϕ is the received phase offset. This amplitude modulated tone (the amplitude A_n is allowed to vary over time) is received in perfect quadrature. N_t is the noise in our received signal which we will presume to be additive. We use the complex representation for this signal (j is $\sqrt{-1}$) because the entire mathematical analysis which follows will depend on it and its mathematical properties greatly simplify the calculations used in the derivation (see note 2).

Let us consider an example. Suppose we have a sample rate of 8000 Hz and we are generating 4096 samples of the complex tone with frequency 1000 Hz and that our phase offset ϕ is 30 degrees or 0.5326 radians. Figure 1 shows the first 64 samples of the real or I or cosine channel as it is variously referred to. Figure 2 shows the first 64 samples of the imaginary or Q or sine channel. Notice that each of the two channels, I and Q in these figures have exactly the same amplitude (1.00) and while it may be difficult to tell exactly, they are exactly 90 degrees or $\pi / 2$ radians apart in phase. When the signal in figure one is at maximum amplitude of 1 or -1, the signal in the second figure is at 0. The same is true for the signal in figure 2. When it is at 1 or -1, the signal in figure 1 is 0. These two signals are in perfect amplitude and phase balance. Another way to see this is to take a look at the power spectrum of this complex signal. We will do this by using the fast fourier transform and then displaying the results as power (dB relative)

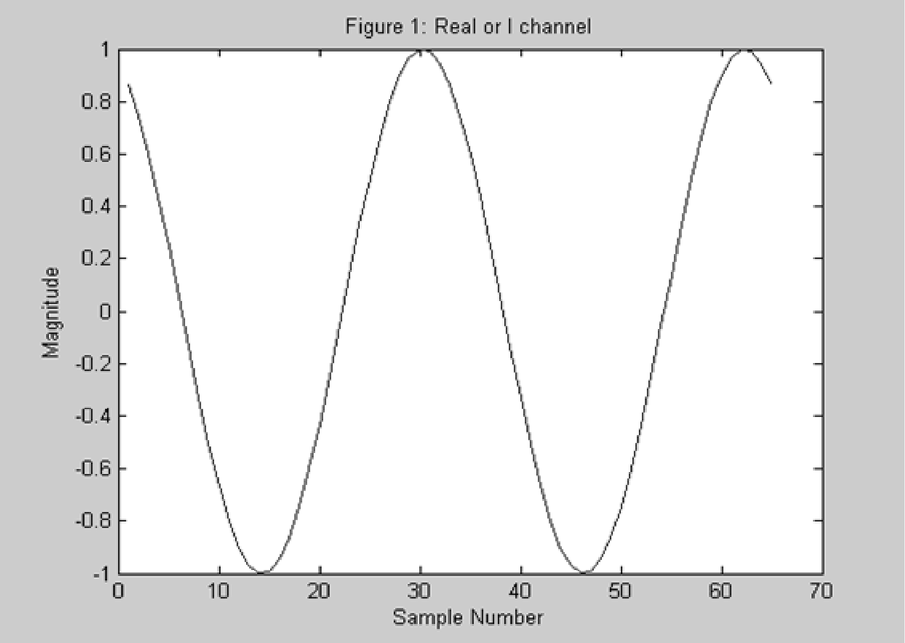


Figure 1: Real or I channel

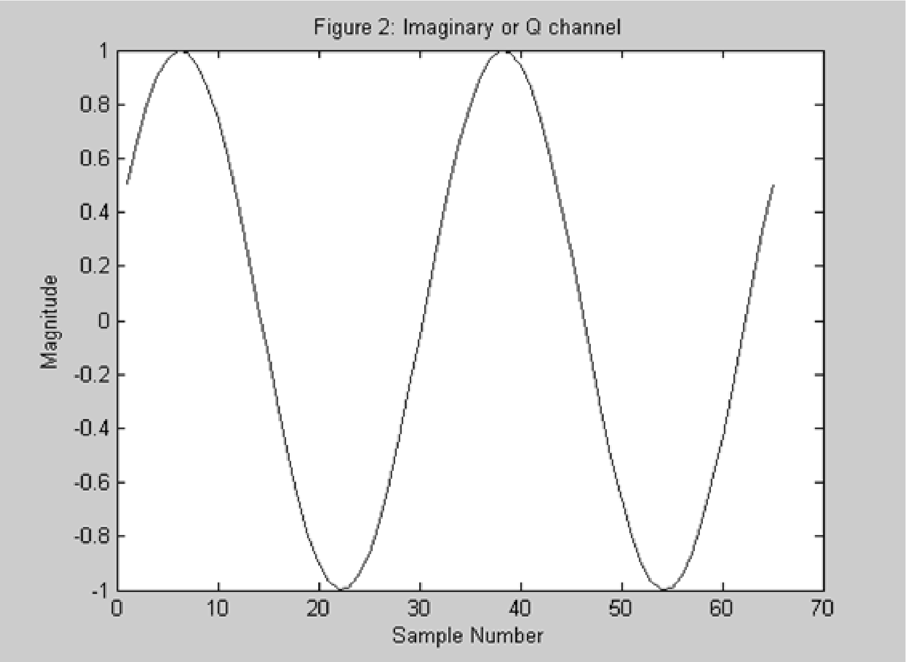
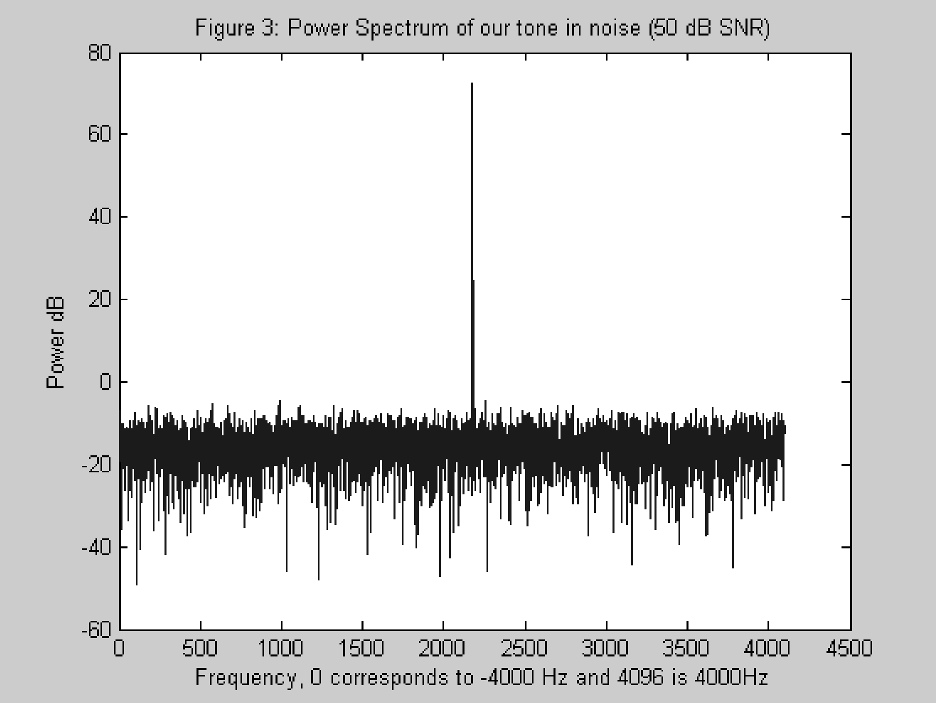


Figure 2: Imaginary or Q channel



An Imperfect Receiver (a real one)

An imperfect receiver, in fact all real ones, will have some amplitude or phase imbalance that leads to a failure of perfect image rejection. We will consider a special case and then make an argument that in all practical cases, it applies.

Rather than the perfect tone of (1.1), consider a complex tone

$$A_n(\cos(2\pi f t_n + \varphi) + jG\sin(2\pi f t_n + \varphi + \vartheta)) + N_n . \qquad (1.2)$$

A_n, f, t_n and φ all have the same meanings as in equation (1.1). G represents the amplitude imbalance in that is should be exactly 1.0. ϑ represents the phase shift. We attribute both to imbalance in the I/Q, complex, or SSB mixer (pick your favorite term; they are all the same thing to our math). Let us consider an example of G = 1.2 and ϑ of 5 degrees. The real channel, shown in figure 4 is identical to that of the signal from equation (1.1). The imaginary channel, show in figure 5 is not identical to that of equation (1.1) and is therefore not in perfect quadrature to the real channel. It is clear from figure 5 that the Q or imaginary component of our signal is a larger amplitude sinusoid than the real channel. If we were to zoom in, we could also see that the zeros and full magnitude points no longer correspond exactly as they did with equation (1.1)

Figure 4: Real or I channel

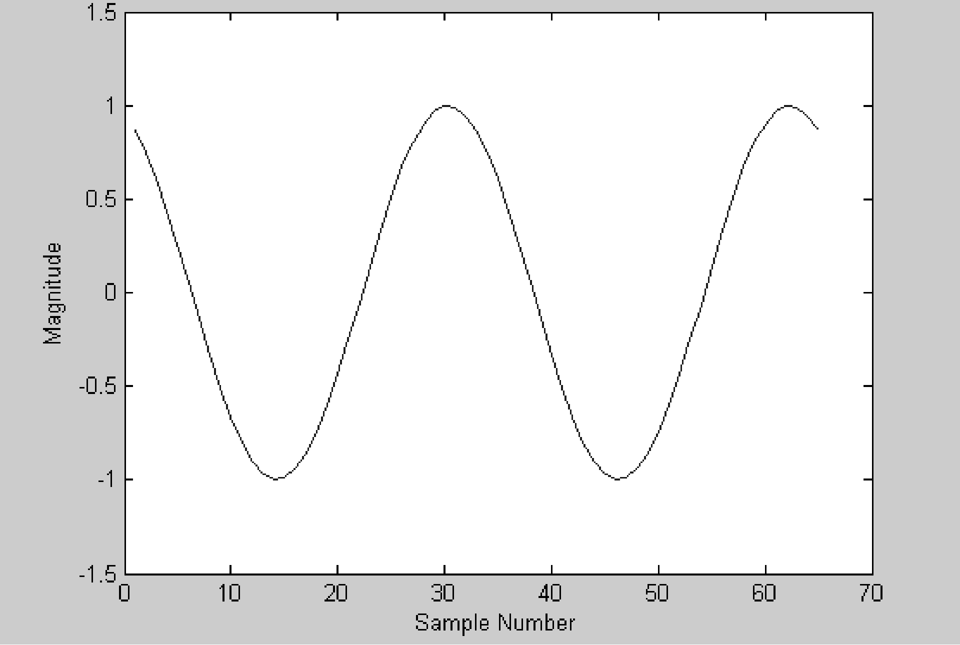


Figure 5: Imaginary or Q channel

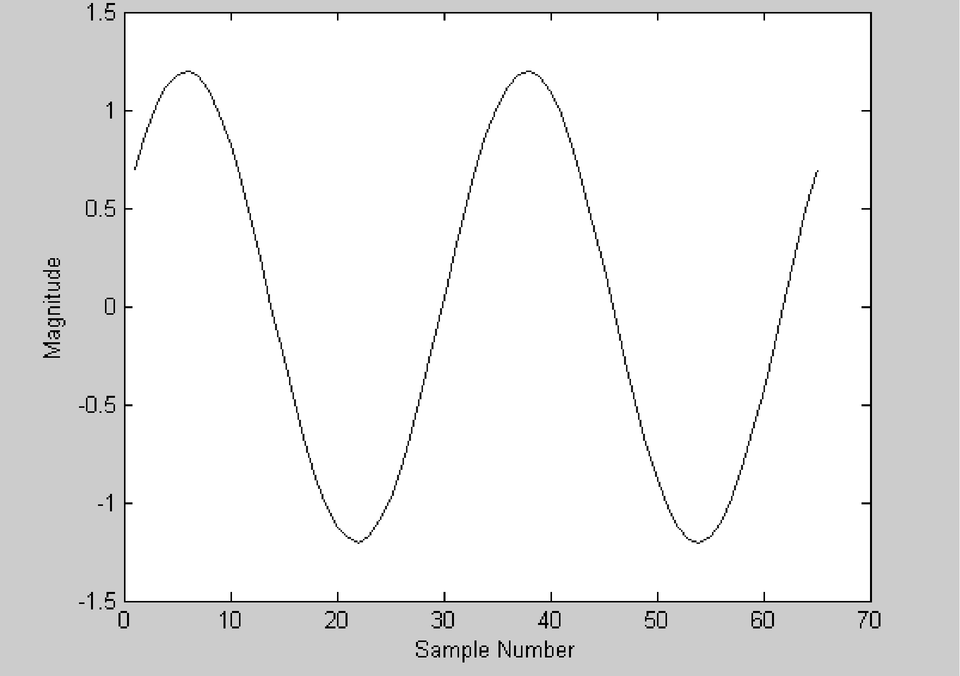
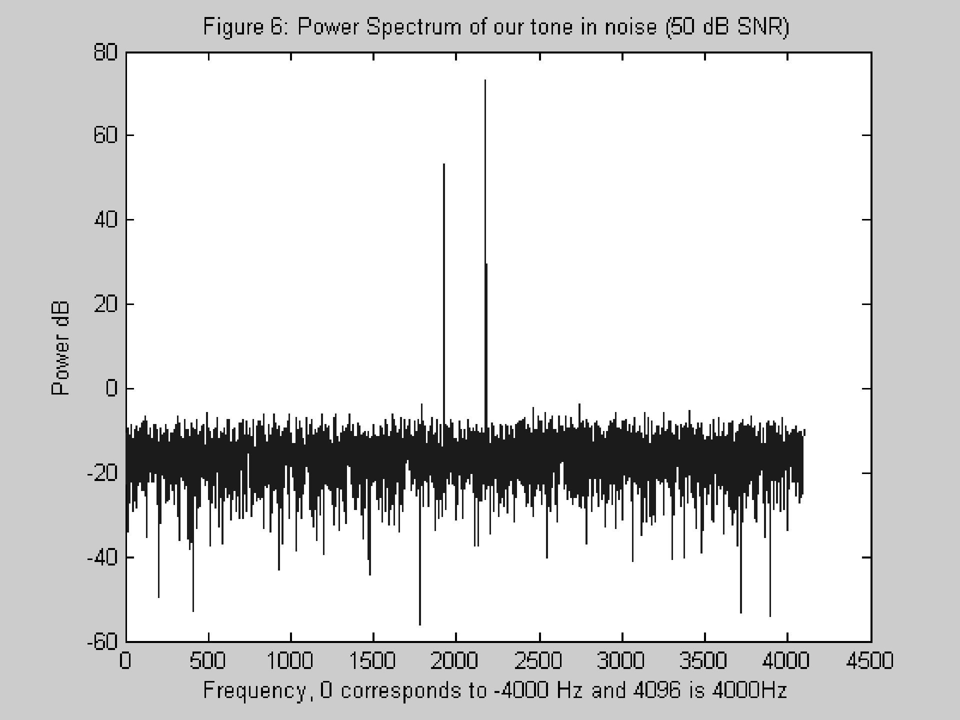


Figure 6 clearly shows why this imbalance is important to the performance of transmit and receive systems which use these mixers. If this were the output of a mixer in our system, we would have very poor image rejection. The graph shows as little as 20 db of image rejection. Yet we only had a phase deviation of 5 degrees and an amplitude difference of 1.5 dB! It is clear why it is imperative this be addressed and successfully.



In order to see how we might address this problem, we first need to rewrite our equation (1.2) in a more useful form. Many have called Euler's formula

$$\exp(j\omega) = \cos(\omega) + j\sin(\omega) \qquad (1.3)$$

the most important formula in mathematics. This might be debatable, but it is almost certainly the single most important formula or identity in all of signal processing.

There are important formulae derived from this identity as well:

$$\begin{aligned} \cos(\omega) &= \frac{\left(\exp(j\omega) + \exp(-j\omega)\right)}{2} \\ \sin(\omega) &= \frac{\left(\exp(j\omega) - \exp(-j\omega)\right)}{2j} \end{aligned} \qquad (1.4)$$

If we use (1.4) in (1.2) and collect all like terms with the complex sinusoids, we get a much more useful form of (1.2)

$$A_n\left(\frac{(1+G\exp(j\theta))}{2}\exp\left(j\left(2\pi f t_n + \phi\right)\right) + \frac{(1-G\exp(-j\theta))}{2}\exp\left(-j\left(2\pi f t_n + \phi\right)\right)\right) \quad (1.5)$$

Please note that the first complex exponential term is the primary signal and the second complex exponential is the image term. This formulation completely explains the main and image spikes in the power spectrum in Figure 6.

In his web page description of Rocky SDR (7), Shovkoplyas gives us a cryptic hint that a certain formula provides the phase and gain imbalance information needed to equalize or “fix” the channel. (1.5) makes this transparent and even suggests the algorithm to follow. But first, we consider a real life example in the Flex Radio 5000A and see what we find. The Flex 5000A has a quadrature sampling detector (1-4) and it is known to have small phase and gain imbalance that varies with frequency. Let us consider the 20 meter band at 14.100 MHz and with the sample rate set to 192,000 samples per second.

In figure 7 we display the phase offset which has been observed between I and Q channel which would be used in equation (1.2) to model the received signal across the IF. Notice that at DC (0 Hz) there are issues in the data. The data there is to be ignored. The nonlinearities involved in the detector become numerically difficult and noise is a serious factor there, but as can be seen from the rest of the graph, finding a formula to interpolate the points in the graph, especially through DC, will not be difficult.

In the DttSP software radio core for PowerSDR, the primary tool for doing the DSP work is the fast Fourier transform. It will be our tool in this endeavor as well. In our case, let us have the FFT “listen” to 4096 contiguous samples of the incoming IF signal. Furthermore, let us carefully tune the signal we are having the FFT listen to. Since the FFT is 4096 long, it may be considered to be 4096 parallel filters on these samples. It allows us to easily separate the two components of formula (1.5). The feature allowing us to do the separation is clearly evident in figure 6. They are not on the same frequency. In formula (1.5), we notice further that the two visible components in Figure 6 have a mathematical relationship we will exploit. One is at frequency f, and the other is at frequency –f. By varying f, we can map out the transfer function through our IQ or sideband mixer.

In order to successfully apply the FFT to our problem, we must carefully pick the f of the signal we use to do our mapping. We must be aware of a feature of the FFT and avoid the pitfalls. If the frequency f we listen to does not sit in the middle of one of the 4096 parallel filters provide by the FFT, the energy from that tone at frequency f begins to splatter out amongst adjacent filters. This would not be good for us because it is critical that we get an accurate reading on the relative power in the two tones visible in figure 6 so we can accurately determine the gain and phase imbalance G and ϕ respectively in equation (1.5). The allowed frequency settings for f are easily derived.

$$f = \frac{n\pi F_s}{\text{FFT length}}, \qquad (1.6)$$

where F_s is the sample rate, in this case, 192000 samples per second, and n is an integer from -2048 to 2047 and in our case the FFT length is 4096. Fortunately for the intended application, applying DttSP to the Flex 5000A, the radio contains an internal generator which can insert the tone at an almost arbitrary frequency and a combiner which will allow the signal to be injected into the front end of the receiver.

So when our intended signal f has a signal very close to one of the allowed frequencies f, almost all of the energy for that signal and its image is two DIFFERENT parallel filters in the FFT. Notice that the multipliers of each of the positive and negative frequency f tones will be assumed to be a constant that does not vary with frequency or time. Measurements show this is so in the Flex 5000 in particular and will be in most practical receivers used with IQ mixers (even the $25 Softrocks!).

Let us take the two FFT bins or filters of interest and multiply their contents. A very good thing happens if we multiply the two components in (1.5). The time varying components with their arbitrary but unknown phase φ are eliminated! This is easy to do in the frequency domain or the Fourier transform domain by taking the two bins and multiplying them.

$$A_n^2\left(\left(1-G^2\right)/4+j\frac{G}{2}\sin\vartheta\right) \tag{1.7}$$

is the resultant signal from this mixing. It will be a complex phasor or DC tone. We need to eliminate A_n^2 if we can. In our case, we can. Notice the strength of the main tone and image in figure 6. Even if the image were much smaller, the y axis on these graphs is in dB. The power in both the desired tone and image are over a hundred thousand times more powerful than the noise. So we can figure out the power of the original signal by simply summing the powers of the two FFT bins or parallel filters we used to get our signals for (1.6) and (1.7). Once we have determined this original signal power from the sum of the powers in the two signals, we can just eliminate it from (1.7) by division. This leaves us with only that part of the original problem that is due solely to gain and phase imbalance.

The real part of the power normalized (1.7) is

$$\left(1-G^2\right)/4 \tag{1.8}$$

and the imaginary part is

$$\frac{G}{2}\sin\vartheta. \tag{1.9}$$

So if R is the real part and I is the imaginary part of our product in (1.7) and setting R equal to (1.8) we may solve for *G* to get

$$G=\sqrt{1-4R} \tag{1.10}$$

From (1.9) we get

$$\sin\vartheta = 2I/G \tag{1.11}$$

From this we get

$$\vartheta=\sin^{-1}\left(2I/G\right) \tag{1.12}$$

Let us rewrite (1.1) as follows and assume the noise is neglible

$$\begin{aligned}&\cos(2\pi f t_n + \varphi) + jG\sin(2\pi f t_n + \varphi + \vartheta) = \\ &\cos(2\pi f t_n + \varphi) + jG[\sin(2\pi f t_n + \varphi)\cos\vartheta + \cos(2\pi f t_n + \varphi)\sin\vartheta]\end{aligned} \tag{1.13}$$

We know the real part (the cosine term) and we have derived both G and ϑ so if the original signal is $A + jB$ then the corrected signal is

$$A + j[B / G - A * \sin\vartheta] / \cos\vartheta \tag{1.14}$$

To build an algorithm that is usable in a software defined radio, we need to be able to measure across the entire IF passband of the IQ mixer and possibly on several bands and maybe even several places in the bands. This is straight forward to do but it could be horrendously expensive unless the phase and gain imbalance we measure is easily described.

Shown in figures 7 and 8 are the phase and gain imbalance profile for the IQ mixer when the radio is tuned to 14.100 MHz. Notice that the phase is a straight line almost everywhere and the gain profile looks pretty easily described as well. This is in fact true. In the algorithms we will choose to apply, we are going to be approximating these curves, while leaving out the area near DC with cubic polynomials in frequency.

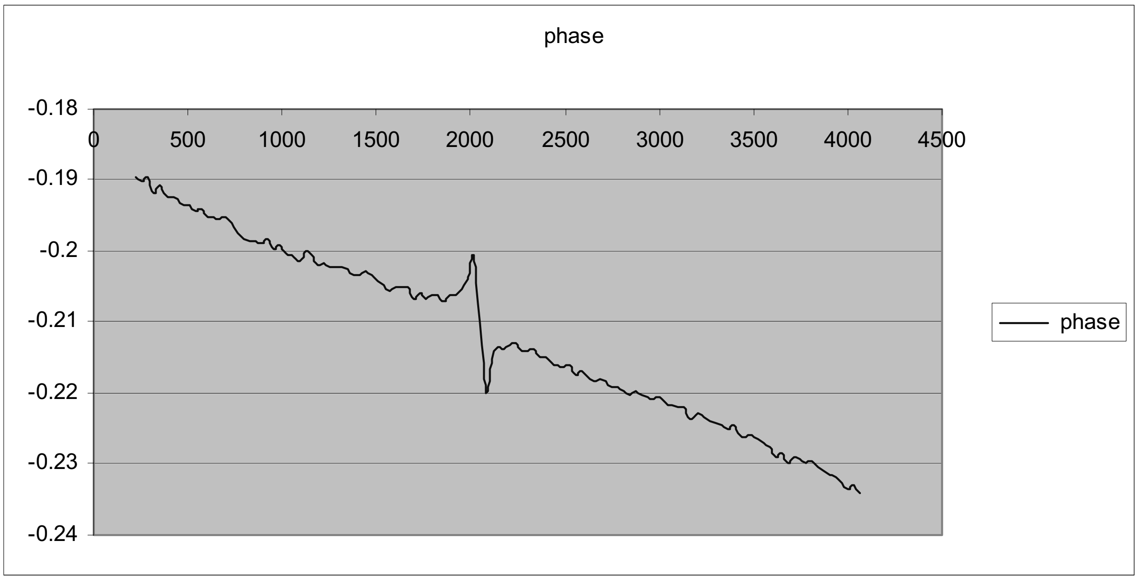


Figure 7: measured phase imbalance at 14.100 in Flex 5000A

The measured phase imbalance is in radians so the offset is on average 12 degrees. Figure 8 shows the gain imbalance in the same equation (1.2) used to model the received signal across the entire 192000 Hz visible in the IF.

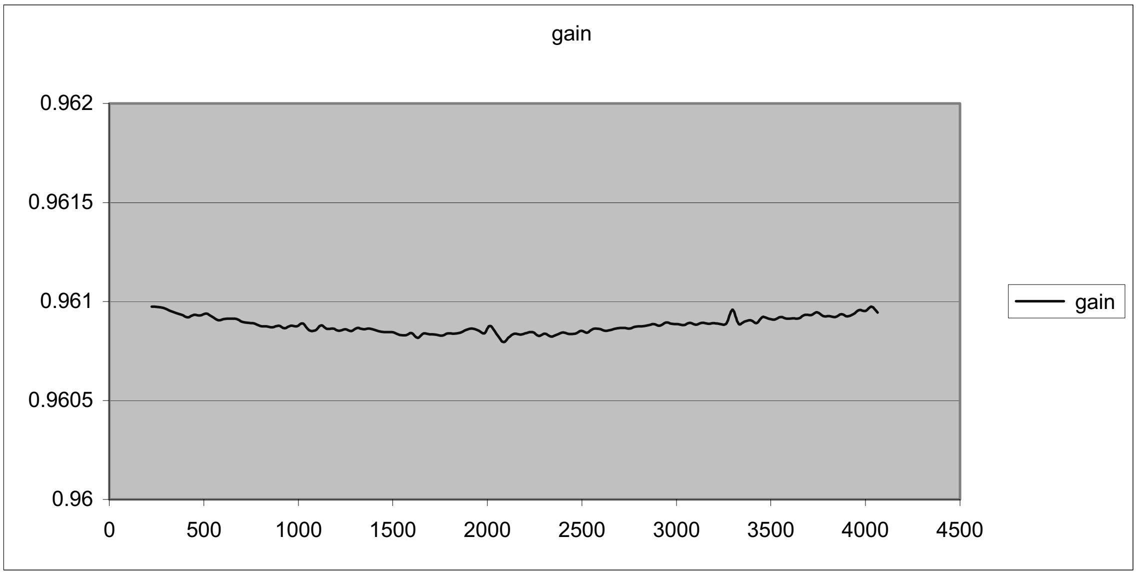


Figure 8: measured gain imbalance at 14.100 in Flex 5000A

We will only need to measure the pass band in a few places and then determine the coefficients of the polynomial. The polynomial will then be evaluated to tell us what gain and phase correction to apply at each bin in the FFT which is involved in the filtering system (ovsv.[ch]) in DttSP.

While we have primarily described the application to the 5000, it also applies to the SDR-1000 and other software radios.

Notes

Note 1: Bob McGwier has been a licensed radio amateur continuously for over 40 years. He earned his Ph.D. in Mathematics at Brown University in the area of stochastic processes and wrote his thesis on Nonlinear Filtering. Bob was cofounder of the AMSAT/TAPR DSP project with Tom Clark K3IO (ex-W3IWI). He was a designer and a builder of the AMSAT Microsats and the principle investigator on the DOVE Voice module. He is currently AMSAT's V.P. for Engineering, chairman of the ARRL SDR Working Group, co-author with Frank Brickle, AB2KT of the DttSP software defined radio library in use in many places but most widely known as the heart of the Flex Radio software: PowerSDR. He is married to Shann, N2HPE, his high school sweetheart and lives and works in New Jersey. He has been employed at the Center for Communications Research of I.D.A. for over 20 years and is the author of over 100 papers for the Department of Defense.

Note 2: See http://mathworld.wolfram.com/ComplexNumber.html and the references contained in this web page.

References

1) Shovkoplyas, Alex, VE3NEA, author of DXAtlas and other offerings at http://www.dxatlas.com .
2) Youngblood, Gerald, K5SDR, *A Software Defined Radio for the Masses, Part 1,* QEX, July/August 2002.

3) Youngblood, Gerald, K5SDR, *A Software Defined Radio for the Masses, Part 2,* QEX, September/October 2002.
4) Youngblood, Gerald, K5SDR, *A Software Defined Radio for the Masses, Part 3,* QEX, November/December 2002.
5) Youngblood, Gerald, K5SDR, *A Software Defined Radio for the Masses, Part 4,* QEX, March/April 2003.
6) Shovkoplyas, Alex, VE3NEA*, CW Shaping in DSP Software,* QEX, May/June 2006.
7) Shovkoplyas, Alex, VE3NEA, http://www.dxatlas.com/rocky/, software defined radio intended for use with the Softrock series of radios.
8) Softrock 40, see http://groups.yahoo.com/group/softrock40 , QSD mixer based SDR designed by Tony Parks, KB9YIG.