New SRT Software Radio Notes

Analog Signal Processing:

The signal we are interested in occupies a band of frequencies near 1420.4 MHz. The receiver processes the signal first in analog form then converts it to digital for further processing. The software radio system consists of an Ettus[1] USRP2-N210 with a DBSRX2 daughter board. All of the analog processing occurs on the DBSRX2.

It begins with an Agilent MGA-62563 GaAs preamplifier that has a noise figure of about 1 dB. That feeds a MAX-2112 chip that is designed for use in the 800–2400 MHz band. To discuss the MAX-2112, it is useful to me to review some analog receiver ideas from my ham radio days (the 1950s).

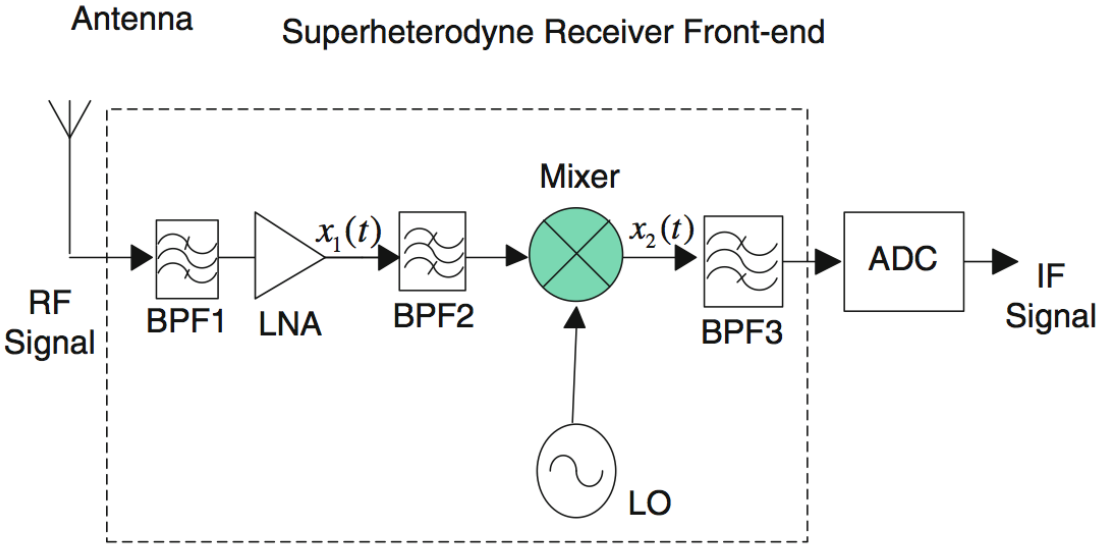
In those days, a state of the art receiver would have been a double conversion superheterodyne receiver. It would begin with an RF preamplifier tuned to a band of frequencies around the desired signal. The amplified RF signal was "mixed" with a local oscillator (LO) signal in a vacuum tube that had two input grids—one for each signal. The output was approximately the product of the two input signals and thus contained sum and difference beats between the LO and the RF input. A narrowly tuned amplifier at a fixed intermediate frequency (IF) selected the difference beat frequency between the LO and the desired RF signal. The IF amplifier provided most of the gain of the receiver and its band pass determined the selectivity of the receiver.

The "superhet" design was invented by Edwin Armstrong in 1918 while he was serving in WW I. The design made it easy to select and separate different RF input frequencies. Of course, there were two RF frequencies that produced an IF beat with the LO. Generally the LO was chosen to be above the desired RF signal by the IF and an RF signal that was the IF above the LO was referred to as an "image". It was the task of the input RF amplifier to reject the image. The signal and its image were separated by twice the IF and to make image rejection easier, a large IF was better. However a high frequency IF amplifier would have a higher bandwidth. To obtain better selectivity, high quality receivers would, after the RF image had been rejected, have a second mixer and LO to convert to a much lower second IF where the amplifier could have a much narrower bandwidth.

Depending on the purpose of the receiver, the amplified IF signal would be processed to recover the information that was originally put on the RF signal. In an amplitude modulated (AM) signal the desired audio information (music, speech, etc.) was added by a process analogous to the mixing in a superhet receiver. The "carrier" frequency of the signal was mixed with the audio frequency signal to produce sum and difference frequencies (called upper and lower "sidebands") that were separated from the carrier by the audio frequency. The carrier and its sidebands passed through a receiver IF amplifier where they could be beat together in a "detector" (essentially a mixer with a zero IF) to recover the original audio frequency modulation. Generally, passing the IF amplifier through a diode worked well; in more sophisticated receivers a detector that took the product of the carrier with the sidebands was sometimes used. In an FM receiver, the frequency of the carrier was varied at the same rate as the audio frequency of the music or speech that was to be transmitted. FM detection was provided by a circuit whose output changed with relatively small changes in the frequency of the output of the IF amplifier.

The AM broadcast band, of course, has signals with AM and the typical analog receiver has an IF of 455 kHz. Analog FM radio (88–108 MHz) receivers typically have an IF of 10.7 MHz. Analog TV sets also used an IF of 10.7 MHz. Each TV channel was allocated a bandwidth of 6 MHz. The audio carrier used FM. The video information was added to its carrier by AM, but one of the sidebands was removed to reduce the bandwidth required.

Using the current diagram style for electronic circuits, here is a superheterodyne receiver.



Software radios follow the concepts mentioned above, but they use digital signal processing and as a result the concepts are implemented differently and are described by different jargon. Many receivers essentially use an IF of zero, and this IF signal is usually described as the "baseband" signal. The Ettus DBSRX2/USRP2 receiver, used in the new SRT, eventually provides a baseband signal which is what we see on the computer screen. However it gets there in much the same way as a double conversion superhet receiver would. In preparation for discussion of software radios, it may be helpful to discuss quadrature mixing and analog signals.

Quadrature Mixing:
Suppose we feed the radio signal into two mixers, each with the same local oscillator but with a 90° phase difference in the LO signal to them. This makes possible some interesting ways to process the signal. The common notation is to call these mixers I (for in phase) and Q (for quadrature, or 90°, phase). Suppose the I mixer is supplied with a LO signal:

$$LO_{\mathrm{I}} = a_{\mathrm{o}} \cos \omega_{\mathrm{o}} t = \frac{a_{\mathrm{o}}}{2} \left[e^{i\omega_{\mathrm{o}} t} + e^{-i\omega_{\mathrm{o}} t} \right]$$

and the Q mixer has the LO signal:

$$LO_{\mathrm{Q}} = a_{\mathrm{o}} \sin \omega_{\mathrm{o}} t = \frac{a_{\mathrm{o}}}{2i} \left[e^{i\omega_{\mathrm{o}} t} - e^{-i\omega_{\mathrm{o}} t} \right]$$

In addition, consider an RF signal at frequency ω_{s}.

$$S(t) = a_{\mathrm{s}} \cos(\omega_{\mathrm{s}} t + \phi_{\mathrm{s}}) = \frac{a_{\mathrm{s}} e^{i\phi_{\mathrm{s}}}}{2} e^{i\omega_{\mathrm{s}} t} + \frac{a_{\mathrm{s}} e^{-i\phi_{\mathrm{s}}}}{2} e^{-i\omega_{\mathrm{s}} t}$$

In each case, the output of the mixer will contain the product of its LO signal with the RF signal.

$$\begin{aligned} I(t) &= \frac{a_\mathrm{o}}{2}\left[e^{i\omega_\mathrm{o}t} + e^{-i\omega_\mathrm{o}t}\right]\frac{a_\mathrm{s}}{2}\left[e^{i\phi_\mathrm{s}}\,e^{i\omega_\mathrm{s}t} + e^{-i\phi_\mathrm{s}}\,e^{-i\omega_\mathrm{s}t}\right] \\ &= \frac{a_\mathrm{o}\,a_\mathrm{s}}{4}\left[e^{i\omega_\mathrm{o}t} + e^{-i\omega_\mathrm{o}t}\right]\left[e^{i\phi_\mathrm{s}}\,e^{i\omega_\mathrm{s}t} + e^{-i\phi_\mathrm{s}}\,e^{-i\omega_\mathrm{s}t}\right] \\ Q(t) &= \frac{a_\mathrm{o}}{2i}\left[e^{i\omega_\mathrm{o}t} - e^{-i\omega_\mathrm{o}t}\right]\frac{a_\mathrm{s}}{2}\left[e^{i\phi_\mathrm{s}}\,e^{i\omega_\mathrm{s}t} + e^{-i\phi_\mathrm{s}}\,e^{-i\omega_\mathrm{s}t}\right] \\ &= \frac{a_\mathrm{o}\,a_\mathrm{s}}{4i}\left[e^{i\omega_\mathrm{o}t} - e^{-i\omega_\mathrm{o}t}\right]\left[e^{i\phi_\mathrm{s}}\,e^{i\omega_\mathrm{s}t} + e^{-i\phi_\mathrm{s}}\,e^{-i\omega_\mathrm{s}t}\right] \end{aligned}$$

These expressions simplify if we assume that the magnitudes of ω_o and ω_s are too high to get through the baseband filter. Then:

$$\begin{aligned} I(t) &= \frac{a_\mathrm{o}\,a_\mathrm{s}}{4}\left[e^{i\phi_\mathrm{s}}\,e^{i(\omega_\mathrm{s}-\omega_\mathrm{o})t} + e^{-i\phi_\mathrm{s}}\,e^{-i(\omega_\mathrm{s}-\omega_\mathrm{o})t}\right] = \frac{a_\mathrm{o}\,a_\mathrm{s}}{2}\,\cos[(\omega_\mathrm{s}-\omega_\mathrm{o})t + \phi_\mathrm{s}] \\ Q(t) &= i\frac{a_\mathrm{o}\,a_\mathrm{s}}{4}\left[e^{i\phi_\mathrm{s}}\,e^{i(\omega_\mathrm{s}-\omega_\mathrm{o})t} - e^{-i\phi_\mathrm{s}}\,e^{-i(\omega_\mathrm{s}-\omega_\mathrm{o})t}\right] = \frac{a_\mathrm{o}\,a_\mathrm{s}}{2}\,\sin[(\omega_\mathrm{s}-\omega_\mathrm{o})t + \phi_\mathrm{s}] \end{aligned}$$

Next, take the Fourier transform of these, treating $I(t)$ and $Q(t)$ as the real and imaginary inputs to a complex transform algorithm.

$$\begin{aligned} I(\omega) &= \frac{a_\mathrm{o}\,a_\mathrm{s}}{2}\int_{-\infty}^{\infty}\Big\{\cos[(\omega_\mathrm{s}-\omega_\mathrm{o})t + \phi_\mathrm{s}]\Big\}\,e^{-i\omega t}\,dt = I^*(-\omega) = \\ Q(\omega) &= i\frac{a_\mathrm{o}\,a_\mathrm{s}}{2}\int_{-\infty}^{\infty}\Big\{\sin[(\omega_\mathrm{s}-\omega_\mathrm{o})t + \phi_\mathrm{s}]\Big\}\,e^{-i\omega t}\,dt = -Q^*(-\omega) \end{aligned}$$

Using

$$\delta(x-a) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ix\zeta}\,e^{-ia\zeta}\,d\zeta$$

these become

$$\begin{aligned} I(\omega) &= \frac{\pi\,a_\mathrm{o}\,a_\mathrm{s}}{2}\Big\{e^{i\phi_\mathrm{s}}\,\delta(\omega_\mathrm{s}-\omega_\mathrm{o}-\omega) + e^{-i\phi_\mathrm{s}}\,\delta(-\omega_\mathrm{s}+\omega_\mathrm{o}-\omega)\Big\} \\ Q(\omega) &= i\frac{\pi\,a_\mathrm{o}\,a_\mathrm{s}}{2}\Big\{e^{i\phi_\mathrm{s}}\,\delta(\omega_\mathrm{s}-\omega_\mathrm{o}-\omega) - e^{-i\phi_\mathrm{s}}\,\delta(-\omega_\mathrm{s}+\omega_\mathrm{o}-\omega)\Big\} \end{aligned}$$

Consider the case when $\omega_\mathrm{s} > \omega_\mathrm{o}$. Suppose $\omega_\mathrm{s} = \omega_\mathrm{o} + \omega_\mathrm{if}$ and use the subscript 'u' (for upper) to indicate the amplitude and phase. There are two contributions; first, for $\omega > 0$:

$$\begin{aligned} I_\mathrm{u}(\omega>0) &= \frac{\pi\,a_\mathrm{o}\,a_\mathrm{u}}{2}\,e^{i\phi_\mathrm{u}}\,\delta(\omega_\mathrm{u}-\omega_\mathrm{o}-\omega) = \frac{\pi\,a_\mathrm{o}\,a_\mathrm{u}}{2}\,e^{i\phi_\mathrm{u}}\,\delta(\omega_\mathrm{if}-\omega) \\ Q_\mathrm{u}(\omega>0) &= i\frac{\pi\,a_\mathrm{o}\,a_\mathrm{u}}{2}\,e^{i\phi_\mathrm{u}}\,\delta(\omega_\mathrm{u}-\omega_\mathrm{o}-\omega) = i\frac{\pi\,a_\mathrm{o}\,a_\mathrm{u}}{2}\,e^{i\phi_\mathrm{u}}\,\delta(\omega_\mathrm{if}-\omega) \end{aligned}$$

Then for $\omega < 0$

$$I_{\rm u}(\omega<0) = \frac{\pi\, a_{\rm o}\, a_{\rm u}}{2}\, e^{-i\phi_{\rm u}}\, \delta(-\omega_{\rm u}+\omega_{\rm o}-\omega) = \frac{\pi\, a_{\rm o}\, a_{\rm u}}{2}\, e^{-i\phi_{\rm u}}\, \delta(-\omega_{\rm if}-\omega)$$
$$Q_{\rm u}(\omega<0) = -i\frac{\pi\, a_{\rm o}\, a_{\rm u}}{2}\, e^{-i\phi_{\rm u}}\, \delta(-\omega_{\rm u}+\omega_{\rm o}-\omega) = -i\frac{\pi\, a_{\rm o}\, a_{\rm u}}{2}\, e^{-i\phi_{\rm u}}\, \delta(-\omega_{\rm if}-\omega)$$

These are the upper sideband for the LO. For the lower sideband, $\omega_{\rm s} = \omega_{\rm o} - \omega_{\rm if}$ and I will use the subscript ℓ for the amplitude and phase. The contribution for $\omega > 0$ is

$$I_{\ell}(\omega>0) = \frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{-i\phi_{\ell}}\, \delta(\omega_{\rm o}-\omega_{\ell}-\omega) = \frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{-i\phi_{\ell}}\, \delta(\omega_{\rm if}-\omega)$$
$$Q_{\ell}(\omega>0) = -i\frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{-i\phi_{\ell}}\, \delta(\omega_{\rm o}-\omega_{\ell}-\omega) = -i\frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{-i\phi_{\ell}}\, \delta(\omega_{\rm if}-\omega)$$

and for $\omega < 0$

$$I_{\ell}(\omega<0) = \frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{i\phi_{\ell}}\, \delta(-\omega_{\rm o}+\omega_{\ell}-\omega) = \frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{i\phi_{\ell}}\, \delta(-\omega_{\rm if}-\omega)$$
$$Q_{\ell}(\omega<0) = i\frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{i\phi_{\ell}}\, \delta(-\omega_{\rm o}+\omega_{\ell}-\omega) = i\frac{\pi\, a_{\rm o}\, a_{\ell}}{2}\, e^{i\phi_{\ell}}\, \delta(-\omega_{\rm if}-\omega)$$

The two mixer outputs $I(t)$ and $Q(t)$ are real, but the Fourier tramsforms $I(\omega)$ and $Q(\omega)$ are complex. The relations between $I(\omega)$, $I(-\omega)$ and $Q(\omega)$, $Q(-\omega)$ are a consequence of $I(t)$ and $Q(t)$ being real.

If you look at the mixer outputs for a frequency $\omega_{\rm if}$ there are contributions from signals both $\omega_{\rm if} > \omega_{\rm o}$ and $\omega_{\rm if} < \omega_{\rm o}$. That is, both the desired signal and its image are present. However, using two mixers with LO inputs 90° out of phase makes it possible to reject the image without the need for stringent filtering of the RF input to the receiver.

For $\omega > 0$, i.e., $\omega_{\rm s} = \omega_{\rm o} + \omega_{\rm if}$, the FT of the mixer outputs will be

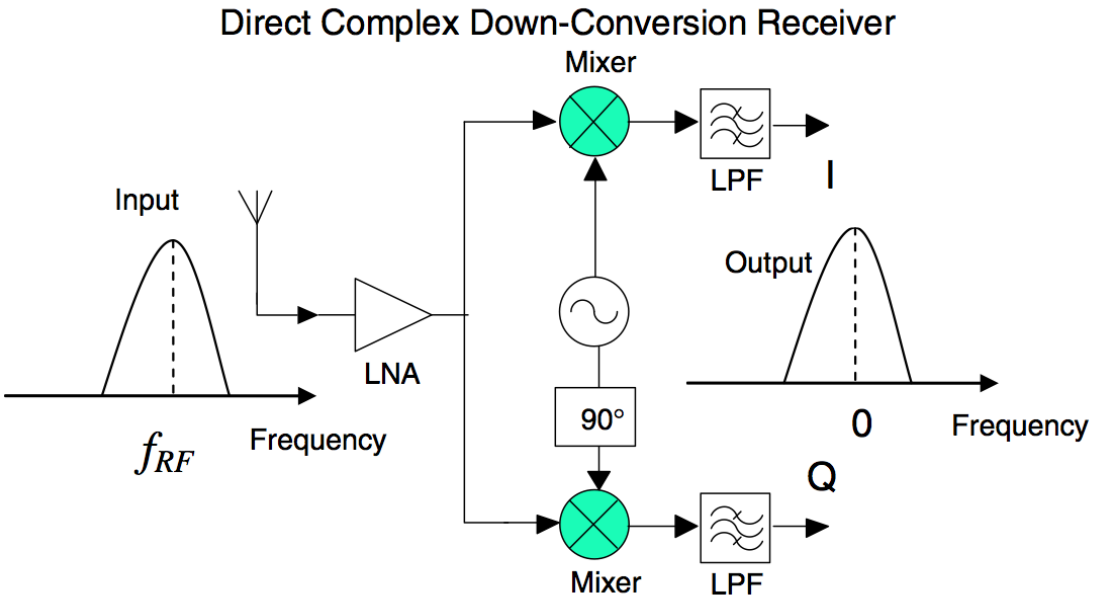
$$I(\omega>0) = I_{\rm u}(\omega>0) + I_{\ell}(\omega>0) = \frac{\pi a_{\rm o}}{2}\Big\{a_{\rm u}e^{i\phi_{\rm u}} + a_{\ell}e^{-i\phi_{\ell}}\Big\}$$
$$Q(\omega>0) = Q_{\rm u}(\omega>0) + Q_{\ell}(\omega>0) = i\frac{\pi a_{\rm o}}{2}\Big\{a_{\rm u}e^{i\phi_{\rm u}} - a_{\ell}e^{-i\phi_{\ell}}\Big\}$$

And for $\omega < 0$, i.e., $\omega_{\rm s} = \omega_{\rm o} - \omega_{\rm if}$, the FT of the mixer outputs will be

$$I(\omega<0) = I_{\rm u}(\omega<0) + I_{\ell}(\omega<0) = \frac{\pi a_{\rm o}}{2}\Big\{a_{\rm u}e^{-i\phi_{\rm u}} + a_{\ell}e^{i\phi_{\ell}}\Big\}$$
$$Q(\omega<0) = Q_{\rm u}(\omega>0) + Q_{\ell}(\omega>0) = -i\frac{\pi a_{\rm o}}{2}\Big\{a_{\rm u}e^{-i\phi_{\rm u}} - a_{\ell}e^{i\phi_{\ell}}\Big\}$$

To reject the lower frequency image, for example, multiply the $Q(\omega > 0)$ output by $-i$ and add it to the $I(\omega > 0)$ output. For negative ω, multiply the $Q(\omega < 0)$ output by i and add it to the $I(\omega < 0)$ output.

You could also build a receiver in which you keep both the upper and lower frequency signals but separate them and work around $\omega_{\rm if} = 0$ so that negative frequencies will be for $\omega_{\rm s} < \omega_{\rm o}$ and positive frequencies are for $\omega_{\rm s} > \omega_{\rm o}$. That way you can measure RF signals in a band with a center frequency that is that of the local oscillator. The jargon for this band of frequencies is the baseband spectrum. Here is a diagram of such a receiver.



That is a common way to build receivers today, and is essentially what the software for the new SRT does. The preceding algebraic discussion of quadrature mixing assumed that the LO amplitude was the same to the I and Q mixers and that the phase difference was exactly 90°. You can imagine that the receiver would not work as well if there were some "IQ imbalance". The Ettus DBSRX2/USRP2 receiver goes to some trouble to deal with this. I will say more about that presently. Now I want to introduce some of the concepts of digital signal processing (DSP). The power of quadrature mixing has been well understood since the early days of vacuum tubes, but it was not until the development of solid state DSP that it became practical to implement it at reasonable cost.

The computer works with discrete samples of the signal equally spaced in time; it uses the fast Fourier transform (FFT) algorithm to carry out a discrete Fourier transform of the I and Q signals in the baseband spectrum. Because of this the algebra is a bit different, but the principle of being able to recover the power in the upper and lower sidebands of the LO from the I and Q baseband signals is valid.

Digital Signal Processing:[2]

Sampling:

If an analog signal is to be processed digitally it must be sampled and converted to digital numbers. That means that the process to be analyzed will be discrete rather than continuous. The analysis will be enormously simplified if the samples are taken separated by equal times. The higher the sampling frequency, the closer the sampled signal approaches a continuous one and the higher the frequencies that can be measured in the signal. There are, of course, practical limitations in the sampling frequency and the resolution of the digital conversion. The sampling is said to be "proper" if the samples uniquely represent the analog signal that has been sampled.

In the 1940s both Shannon and Nyquist wrote papers examining the necessary criterion, and the resulting **sampling theorem** is sometimes named after either of them. The criterion is that the sampling frequency must be at least twice the highest frequency in the analog signal that is to be uniquely represented. Suppose the sampling frequency is 2000 Hz. It is fairly easy to see that if you have a 1000 Hz sine wave and sample it every half period that you can tell if it is 1000 Hz or some lower frequency, but that if you have a sine wave that is 2000 Hz (the sample frequency) it will show up as a constant, i.e., zero frequency, when sampled every 0.5 ms. A sine wave of 3000 Hz will give the same samples as one of 1000 Hz (with a 180° phase shift), and so on. The frequencies outside the 0 to 1000 Hz you are interested in but which appear to be within this band when sampled are called *aliases*. Nyquist and Shannon proved the theorem rigorously and showed, for example, that a 1900 Hz sine wave would sample the same as one of 100 Hz when sampled at 2000 Hz. This aliasing phenomenon can be useful sometimes (when using a strobe light to look at rotating machinery) and you have probably observed wagon wheels slowing down and reversing direction in movies—an alias from the 24 Hz sampling rate.

In the Ettus USRP2 the I and Q analog signals from the DBSRX2 receiver are sampled at 100 MHz and converted to 14-bit integers. The digital outputs can be resampled at lower frequencies, a process called decimation, for further analysis. To avoid aliasing, the low pass filters in the receiver (see the Ettus Receiver section below) should cut off at half the sample rate chosen. However, because a complex FFT (analogous to that on page 4) is applied to the I and Q signals, both positive and negative values of ω result and so the effective receiver bandwidth equals the sample frequency.

The Fast Fourier Transform:

Credit for the FFT is given to J. W. Cooley and J. W. Tukey[3] although Gauss (who also invented complex numbers) reportedly used the method. In computer calculations, we deal with discrete equally spaced signals in time. We also deal with a finite number of samples; for the FFT algorithm the number of samples N should be a power of two. What does the Fourier transform of $N = 2^m$ equally spaced time samples $a(t)$ look like? Well, it depends on what we think the sequence of samples before and after the N samples that interest us looks like. If it were aperiodic, an infinite number of continuously distributed Fourier frequency coefficients would be required to represent it. But we only have a limited number of frequencies, up to half the sampling frequency, available. That means that the Fourier transform of the N discrete time samples will itself be discrete and can have only N terms. It also means that we assume periodic boundary conditions on both the input signal, $a(t) = a(t + Nt_{\text{samp}})$ and the Fourier transform will also be periodic.

If the sampling frequency is $f_{\rm samp} = 1/t_{\rm samp}$, the highest frequency will be $f_{\rm max} = \omega_{\rm max}/2\pi = f_{\rm samp}/2$. Since we are taking a complex Fourier transform, the Fourier coefficients will, in general, be complex numbers even if $a(t)$ is real. The frequencies will be discrete, spaced by $f_{\rm sp} = f_{\rm samp}/N$ or $\omega_{\rm sp} = 2\pi f_{\rm samp}/N$ between $-\omega_{\rm max} < \omega < \omega_{\rm max}$. The Fourier transform will be

$$A_m = \sum_{n=0}^{n=N-1} a_n e^{-i\,m\omega_{\rm sp} t_{\rm samp}\,n} = \sum_{n=0}^{n=N-1} a_n e^{-i2\pi(mn/N)}$$

Often this is written showing the separate real and imaginary parts

$$\mathrm{Re}\,A_m - i\,\mathrm{Im}\,A_m = \sum_{n=0}^{n=N-1} a_n \cos\left(\frac{2\pi mn}{N}\right) - i \sum_{n=0}^{n=N-1} a_n \sin\left(\frac{2\pi mn}{N}\right)$$

The power at frequency $m\omega_{\rm sp} = m2\pi/(Nt_{\rm samp})$ for $-N/2 < m < N/2$ will be proportional to $(\mathrm{Re}\,A_m)^2 + (\mathrm{Im}\,A_m)^2$. **Note** that $\mathrm{Re}\,A_m = \mathrm{Re}\,A_{-m}$ while $\mathrm{Im}\,A_m = -\mathrm{Im}\,A_{-m}$. This means that there are only $N/2$ independent values for $\mathrm{Re}\,A_m$ and also only $N/2$ independent values for $\mathrm{Im}\,A_m$. Thus the inverse will be

$$\begin{aligned}
a_n &= \frac{1}{N} \sum_{m=-N/2}^{m=N/2} A_m e^{i2\pi(nm/N)} \\
&= \frac{1}{N} \sum_{m=-N/2}^{m=N/2} \{\mathrm{Re}\,A_m - i\,\mathrm{Im}\,A_m\} \left\{\cos\left(\frac{2\pi nm}{N}\right) + i\sin\left(2\frac{\pi nm}{N}\right)\right\} \\
&= \frac{2}{N} \sum_{m=0}^{m=N/2} \mathrm{Re}\,A_m \cos\left(\frac{2\pi nm}{N}\right) + \frac{2}{N} \sum_{m=0}^{m=N/2} \mathrm{Im}\,A_m \sin\left(\frac{2\pi nm}{N}\right)
\end{aligned}$$

So far, this discussion has considered only real samples $a(t)$ subject to the discrete Fourier transform. Now suppose the samples are complex: $a(t) + i\,b(t)$, where $b(t)$ is a real number. The preceding calculation can give the results for the coefficients B_m if we simply replace a_n by $i\,b_n$. This means

$$B_m = i \sum_{n=0}^{n=N-1} b_n e^{-i2\pi(mn/N)} = \sum_{n=0}^{n=N-1} b_n \sin\left(\frac{2\pi mn}{N}\right) + i \sum_{n=0}^{n=N-1} b_n \cos\left(\frac{2\pi mn}{N}\right)$$

$$\mathrm{Re}\,A_m + i\,\mathrm{Im}\,A_m = \sum_{n=0}^{n=N-1} a_n \cos\left(\frac{2\pi mn}{N}\right) + i \sum_{n=0}^{n=N-1} a_n \sin\left(\frac{2\pi mn}{N}\right)$$

$$\mathrm{Re}\,B_m + i\,\mathrm{Im}\,B_m = \sum_{n=0}^{n=N-1} b_n \sin\left(\frac{2\pi mn}{N}\right) + i \sum_{n=0}^{n=N-1} b_n \cos\left(\frac{2\pi mn}{N}\right)$$

Now we see that $\mathrm{Re}\,B_m = -\mathrm{Re}\,B_{-m}$ while $\mathrm{Im}\,B_m = \mathrm{Im}\,B_{-m}$. So, there is a way to separate the FFTs of the real and imaginary inputs. Remember, too, that while we usually consider $0 \le n \le N-1$ and $-N/2 \le m \le N/2$ the periodicity means that $a_{n\pm pN} = a_n$ and $A_{m\pm pN} = A_m$ where p is any integer.

The Gnuradio[4] software takes the I and Q outputs of the USRP2 and produces a vector of complex single precision floating point numbers. Code from the *fftw3*[5] package is used to compute the Fourier transform. For N complex time samples, the *fftw3* returns an N component complex vector $\mathcal{V}$ containing the complex Fourier coefficients. The initial component of the vector is the zero frequency coefficient

$$\mathcal{V}[0] = (\mathrm{Re}\, A_0 + \mathrm{Re}\, B_0) + i(\mathrm{Im}\, A_0 + \mathrm{Im}\, B_0) = \sum_{n=0}^{n=N-1} (a_n + ib_n)$$

Now, let's consider a component k, where $k < N/2$

$$\begin{aligned}\mathcal{V}[k] = &\sum_{n=0}^{n=N-1} \left\{ a_n \cos\left(\frac{2\pi kn}{N}\right) + b_n \sin\left(\frac{2\pi kn}{N}\right) \right\} \\ &+ i \sum_{n=0}^{n=N-1} \left\{ a_n \sin\left(\frac{2\pi kn}{N}\right) + b_n \cos\left(\frac{2\pi kn}{N}\right) \right\} = \mathcal{V}[N+k]\end{aligned}$$

and a component $N - k$, where $k < N/2$

$$\begin{aligned}\mathcal{V}[N-k] = &\sum_{n=0}^{n=N-1} \left\{ a_n \cos\left(\frac{2\pi (N-k)n}{N}\right) + b_n \sin\left(\frac{2\pi (N-k)n}{N}\right) \right\} \\ &+ i \sum_{n=0}^{n=N-1} \left\{ +a_n \sin\left(\frac{2\pi (N-k)n}{N}\right) + b_n \cos\left(\frac{2\pi (N-k)n}{N}\right) \right\} \\ = &\sum_{n=0}^{n=N-1} \left\{ a_n \cos\left(\frac{2\pi kn}{N}\right) - b_n \sin\left(\frac{2\pi kn}{N}\right) \right\} \\ &+ i \sum_{n=0}^{n=N-1} \left\{ -a_n \sin\left(\frac{2\pi kn}{N}\right) + b_n \cos\left(\frac{2\pi kn}{N}\right) \right\} = \mathcal{V}[-k]\end{aligned}$$

So, in the complex vector $\mathcal{V}$ the components $\mathcal{V}[0]$ through $\mathcal{V}[N/2-1]$ correspond to the positive frequencies (upper sideband) and $\mathcal{V}[N-1]$ through $\mathcal{V}[N/2]$ are the lower sideband of the LO in the DBSRX2 receiver. The Gnuradio software module `gr.fft_vcc()` makes the appropriate translations before computing the power at each frequency. I put code in the server so either $\log_{10}$ or linear FFT power data can be sent back to the client on the computer.

Windowing:
In order to calculate the FFT of the radio signal, N complex data points spaced by the sample time are sent to the *fftw3* function. Before they are sent the points may multiplied by a set of N scale factors. These N scale factors are called a **window**. The simplest window would be for all N scale factors to be 1.0. This is sometimes referred to as having no window or sometimes as a Rectangular window—to recognize that there is always a window just because a finite number of data points is chosen.

The significance of the window can be understood if you recall that multiplication in the time domain is equivalent to convolution in the frequency domain. Thus the Fourier transform of the window function is the resolution function for the FFT results.

A rectangular window of width $Nt_{\rm samp}$ has the Fourier transform $Nt_{\rm samp}(\sin X)/X$ where $X = (Nt_{\rm samp}/2)\omega = \pi\omega/\omega_{\rm sp}$ and $\omega_{\rm sp}$ is the spacing between the data points in the FFT result.

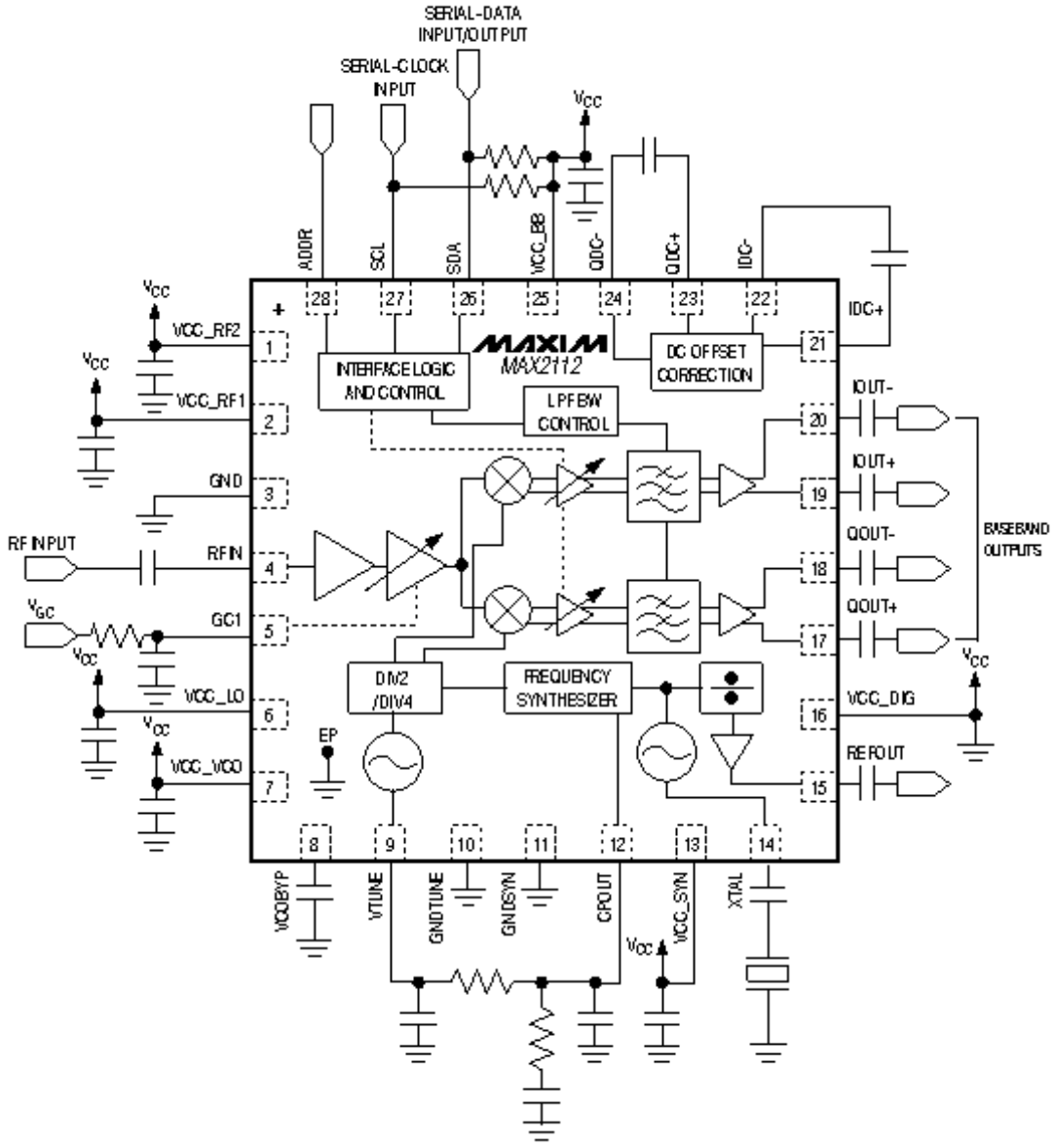
There are several commonly used window functions, usually named after the people who proposed them. Useful windows are almost always symmetric about the center of the packet of N time data points and they usually reduce the amplitude at the beginning and end of the packet. Thus the window is effectively narrower in the time domain and gives a broader frequency resolution. Windowing is most useful if you are seeking to identify rather narrow features in the frequency domain, especially if they happen to fall about midway between two of the discrete frequencies of the FFT. It can also help reduce noise, as with broader resolution there is a greater tendency for noise to average out between adjacent bins in the FFT.

In radio astronomy studies of the 21 cm line we are not usually looking for sharp features, but you should keep in mind that the measured power at each FFT frequency can be affected by the window used. For that reason the telescope should be calibrated with the same window you intend to use for your measurements. I wrote the software to use a rectangular window.

It will not surprise you that windows are important in the design of digital filters. The FFT, after all, is a comb of N equally spaced frequency filters. You might think that a suitable time domain window could produce nice frequency filters. For example, an appropriate $(\sin X)/X$ time window might produce nice rectangular FFT bins. Unfortunately a time window to produce FFT bins with steep sides and sharp corners, would have to extend well outside the $Nt_{\rm samp}$ length of the N input data points. However, the idea is often a useful one and digital low pass, high pass and band pass filters often make use of windows to improve their performance.

An important technique is called "decimation", which is the term for keeping only 1 in every n of the samples you take of the signal. As first thought that may seem silly, as you are discarding information about the signal. That is true, but you are not discarding information about any changes in the signal whose frequency is low enough that it changes little from sample to sample. So you lose no information about the low frequency parts of the signal and this can be a way to implement a low pass filter.

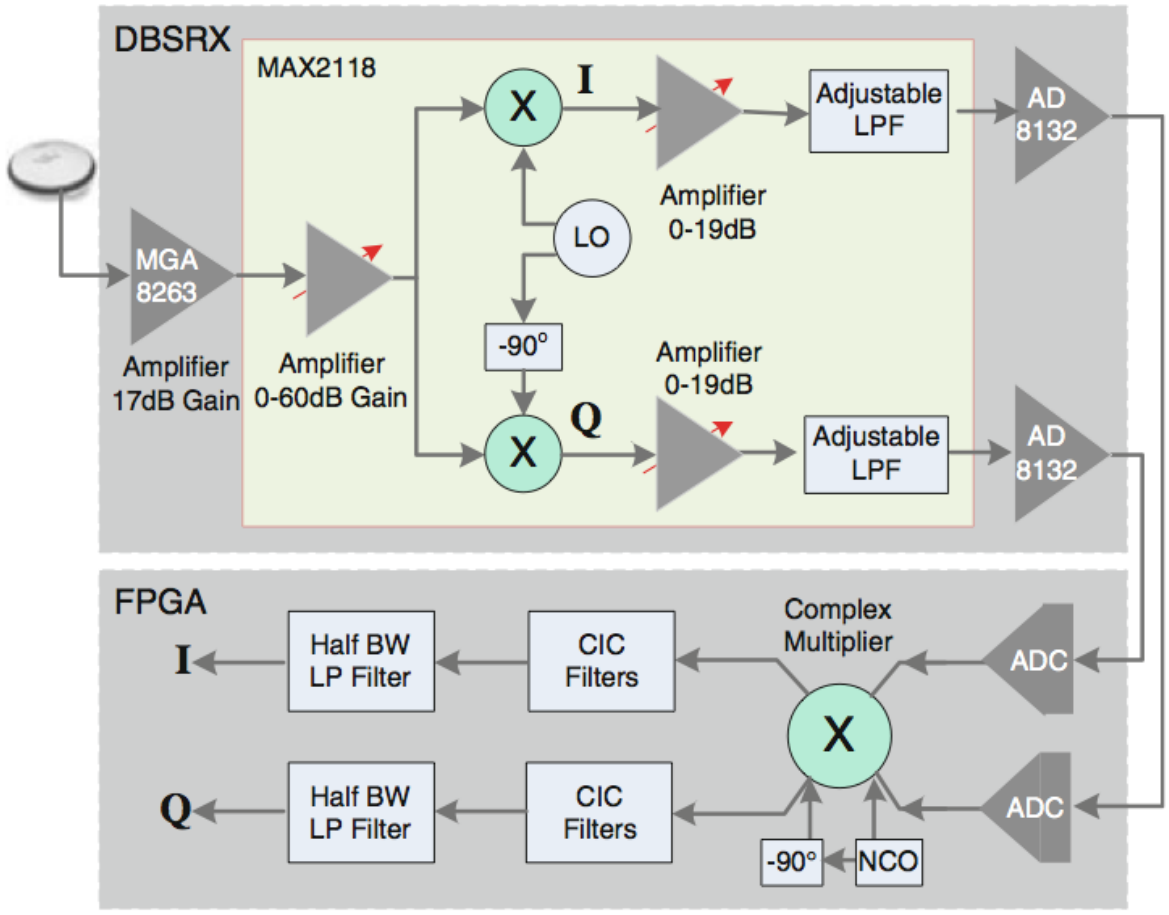
The Ettus DBSRX2/USRP2 Receiver:
This is the receiver used in the new SRT. The front end is on the DBSRX2 daughter board and is built around the MAX-2112 receiver chip.



The RF signal passes through a fixed gain preamplifier and then a variable gain amplifier before it is split and input to two mixers that produce the product of the signal with a LO. The variable gain RF amplifier has a voltage gain that can be varied from 1 to 4500 (0 to 73 dB). The two mixers are fed by the same local oscillator, but with a 90° phase difference, *en principe*, between the LO signals to the two mixers. The output of each mixer is then passed through a variable gain "baseband" amplifier into a low pass filter, whose cut-off frequency can be adjusted from 4 MHz to $\simeq$ 40 MHz, and then into differential amplifiers, that roll off at 6 dB per octave and are down by 6 dB at 33 MHz, before going to 14-bit analog to digital (ADC) converters on the USRP2 motherboard. I think the chip is commonly used as a direct down conversion receiver (page 5) but that is not quite the way Ettus uses it.

One of the weaknesses of this kind of receiver is a DC offset (zero frequency signal) caused by LO leakage into the signal input[6]. The USRP2 uses a LO offset (in the MAX2112) from the desired center frequency the receiver is tuned to, so that LO leakage will fall outside the desired bandwidth. I don't know what this offset is, but the bandwidth of the amplifiers in the MAX2112 means that the USRP2 uses an IF frequency that is not zero, but is probably also significantly less than 30 Mhz. The baseband sampling frequency would be a sensible guess.

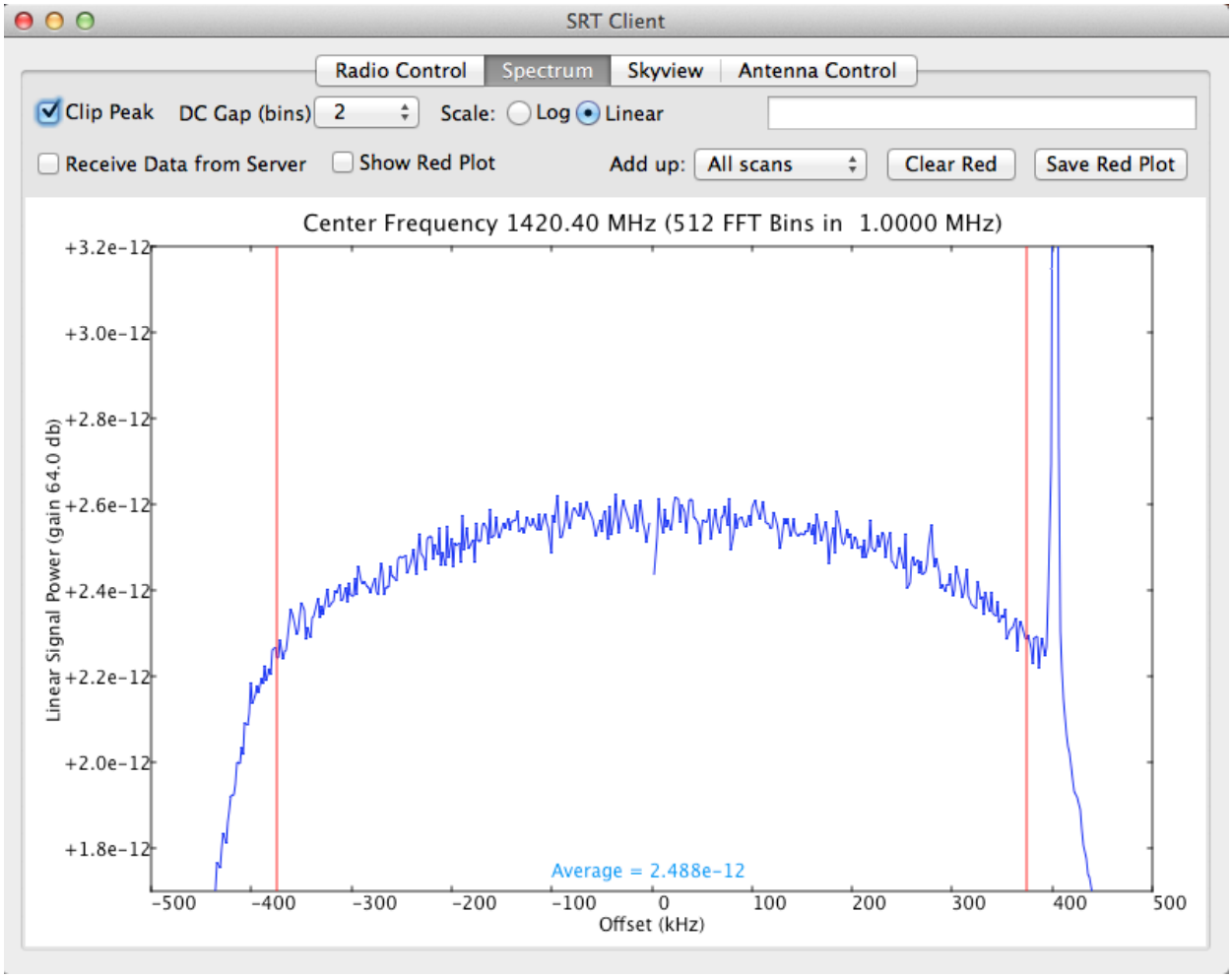
In any case, the USRP2 acts like a double conversion superhet. The motherboard digitizes the I and Q signals using a TI ADS62P4X which takes differential inputs; I think the USRP2 runs it with a 100 MHz sampling frequency. The outputs go into the Xilinx FPGA. There, according to this figure stolen (along with those on pages 2 and 5) from an article by Senlin Peng and Yu Morton,[7] it is converted to zero baseband frequency using a digital version of a quadrature mixer.



The figure is for a DBRSX daughterboard (with a MAX2118) but I think is otherwise the same as ours. The output from the Half BW filters is sent by GHz Ethernet UDP as a stream of complex integers (16-bit signed real and imaginary parts) to the computer in 26-630 at the sample rate selected by the server software on the computer. Usually that is 1.00 or 1.25 MHz, but can be as high as 5.0 MHz–which requires a 160 Mb/s Ethernet data rate.

Filter Correction:
As I discussed above, Nyquist proved that sampling will accurately represent a signal that has no components above half the sampling frequency. Since we use both upper and lower sidebands, our total bandwidth equals the sampling frequency. The initial sampling frequency in the USRP2 is 100 MHz but this is decimated to produce the sampling frequency that we choose to determine the bandwidth we want for the receiver. That is done digitally with the CIC filters[8] followed by the Half BW filters shown on page 11. To work properly, this requires the decimation (i.e., the ratio of 100 MHz to the sample rate) to be an even number.

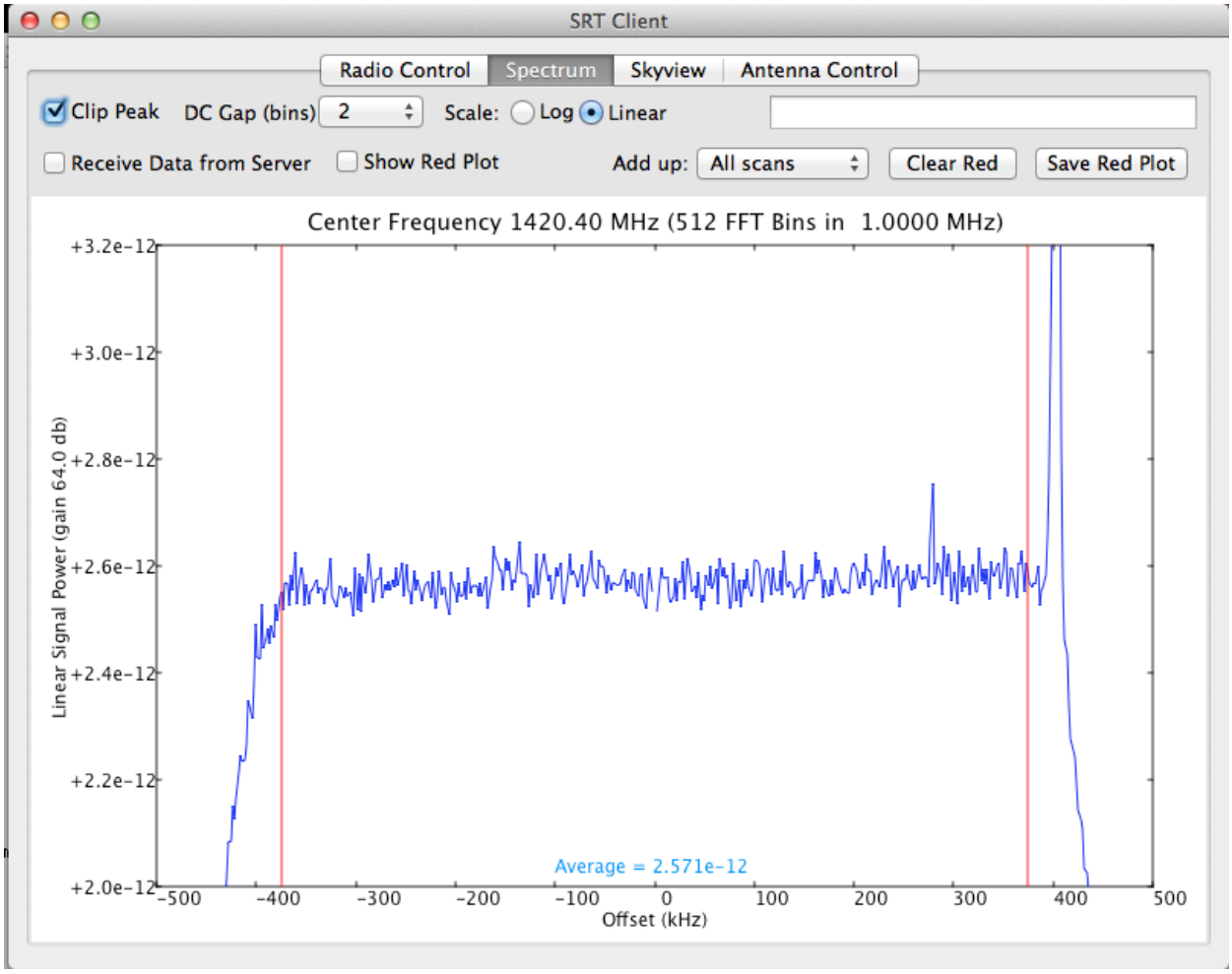
The purpose of the low pass filter is to eliminate aliasing from higher frequencies. But no low pass filter can be perfectly sharp and so it will have some effect on the signal analyzed by the FFT. The power of frequencies further away from the baseband will be reduced more. That can be seen in the raw FFT output from one scan in the figure below. The redlines mark the ±192 bins in the FFT. I think signals between these bins are reliable and can be averaged to get the signal power. The large peak in the blue area to the right is probably an artifact of the software radio and the neighboring cell phone transmitters, which I would like to understand better.



To correct for this pre-filter roll-off, I measured carefully the FFT signal (with an integration time of about 1000 s) from a white source (the Sun) and folded the upper and lower sideband signals on top of each other. Then I fit the average for bin numbers, ≤ 192. The fit was very good and gave the normalised result

$$1.0 - 1.903 \times 10^{-6} n^2 - 3.411 \times 10^{-11} n^4$$

where n was the bin number from the baseband frequency. This was used to calculate a list of 256 correction factors to achieve a flat frequency response for the central 384 bins in the FFT. For the side bins ($|n| > 192$), the factor for $n = 192$ was used. The plot below shows a subsequent scan after the correction was applied.

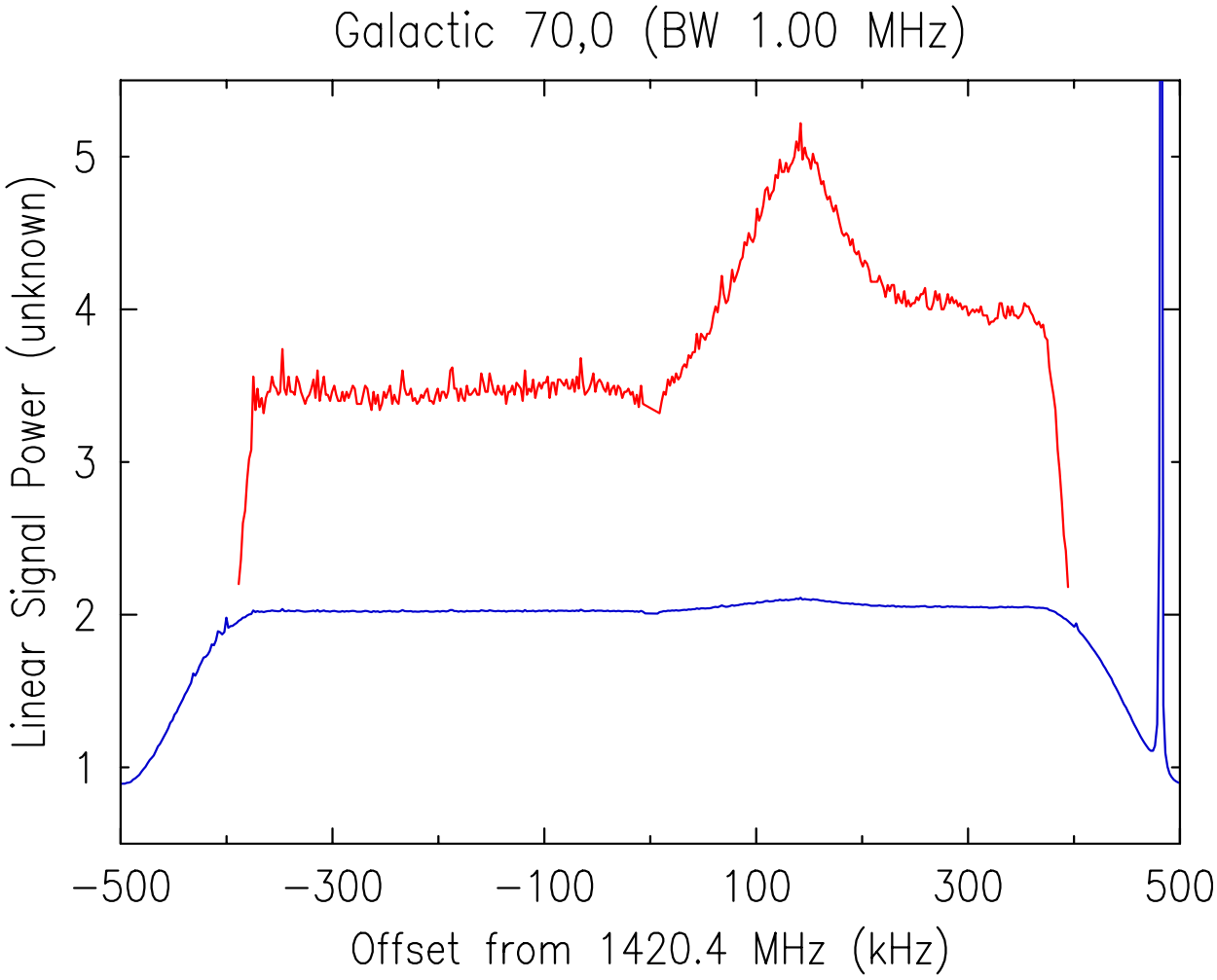


I did this for sample rates of 1.00 MHz and 2.00 MHz. I wrote the software in `bernieClientG.py` to restrict sample rates to 11 values from 416 kHz to 5.00 MHz that gave even decimation values with the 100 MHz DSP. It turned out that the 1.00 MHz corrections worked very well for sample rates 1.25 MHz and below as well as 2.5 MHz and above. The 2.00 MHz correction had more curvature and worked well for 1.5152 MHz and 2.00 MHz. The equation for it was

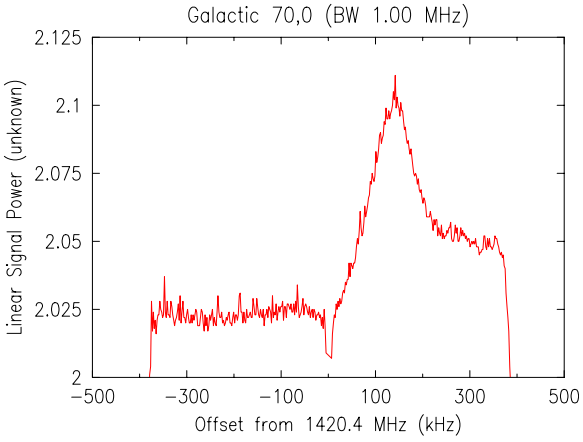
$$1.0 - 1.533 \times 10^{-5} n^2 + 6.027 \times 10^{-11} n^4$$

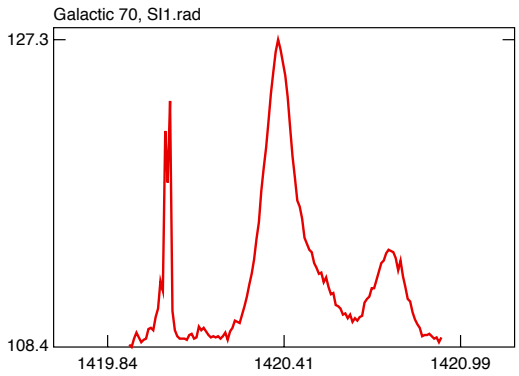
I suppose that tells us something about how the DSP in the USRP2 works.

The figure below shows some data taken at Galactic (90, 0) on April 9, 2013. That blue curve shows the full range of the data, except the top of the peak is clipped. It is the result of integrating the FFT output for about 1000 s. The red curve shows the same data scaled up and clipped in the plot to show only the interesting frequency range and intensity.



I wanted to try to compare the NSRT sensitivity with the SRT, and I had some data taken by students Illan Halpern and Sarah Geller in October, 2011. I plot them below beside my April 9 data.





The peaks look similar. The SRT data have an interference peak at about 1420 and the NSRT data notch at zero offset is, I think, the result of an imbalance in the USRP2. It can be either a notch or a peak depending on the receiver gain and is usually about two FFT bins wide.

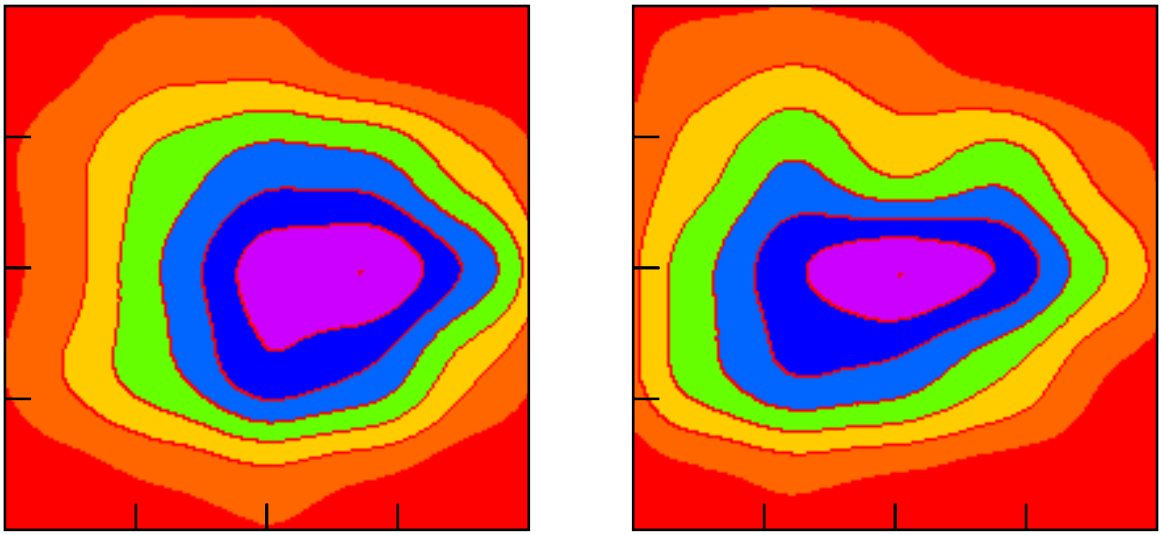
A comparison needs to be taken with a grain of salt as my telescope aiming had not yet been calibrated. My peak appears to be 50–100 kHz higher. The interesting comparison is the ratio of peak height to background (system noise). My peak is about 4% of the background, while Halpern/Geller's is about 15%. This could be just because the NSRT antenna was pointed somewhat out of the galactic plane. However I am concerned about the insertion loss of the old 1420 MHz filter as well as some apparent corrosion of the probe in the NSRT antenna feed horn.

On April 26 the new 1420 MHz bandpass filter we had ordered over a year ago arrived, and we installed it. For reasons I don't understand yet, the low pass filter corrections were slightly asymmetric and I decided to use 512 correction factors instead of the same 256 applied to both positive and negative baseband frequencies. When I fit both positive and negative frequencies to the same expression, χ^2 was nearly 10 times greater. The separate fits to the data were as follows.

$1.0 - 2.48 \times 10^{-6} n^2 - 2.38 \times 10^{-11} n^4$	1.0 MHz sample rate, negative frequencies
$1.0 - 1.67 \times 10^{-6} n^2 - 3.23 \times 10^{-11} n^4$	1.0 MHz sample rate, positive frequencies
$1.0 - 1.31 \times 10^{-5} n^2 + 7.09 \times 10^{-11} n^4$	2.0 MHz sample rate, negative frequencies
$1.0 - 1.17 \times 10^{-5} n^2 + 4.77 \times 10^{-11} n^4$	2.0 MHz sample rate, positive frequencies

Using a script `usrp_corr.py`, the reciprocals were put in the files `USRPcorr1nf.dat` and `USRPcorr2nf.dat` to use as correction factors in the `bernieClientG.py` program.

Telescope Pointing:
After adding software to do a 5x5 npoint grid scan I tested the telescope aiming on the Sun. The original aim was set by taking the EQ zero to be at the lock position and the DC zero to be 34° above the point where lower limit switch closed. Below at the left is the scan. The peak appears very close to the EL offset of 0 and the AZ off is about 2° (the half beamwidth for the antenna, whose diameter is 96 in or 240 cm, is 3°.) At the coordinates of the Sun for this scan, the EQ-DC and AZ-EL axes were at about a 45° angle to each other. So, to make the correction I subtracted 1° from each to make the EQ lock position −1° and the DC zero to be 35° from the point where the EQA limit closed.



Two days later I repeated the Sun npoint scan and got the result above on the right. So the pointing calibration seems to be good.

On April 16 I did Sun npoint scans with both the SRT and the NSRT close to the same time (about 1/2 hour apart). With the NSRT I found the ratio of the peak in the scan to the background signal when the telescope was pointed 20° off the sun direction was 1.3; the same number for the SRT was 2.0. So, the S/N ratio is not as good for the NSRT. The SRT indicated an interference peak was $\simeq$ 400 kHz below the 1420.4 MHz central frequency, that we often observe. The NSRT did not show this same peak, but had a strong one $\simeq$ 400 kHz above, which I believe to be spurious. I want to investigate whether the SRT peak might also be an artifact, as it uses a similar DSP technology in the receiver. With either telescope, one can sit at constant receiver parameters and watch the peaks move around and their strength change; they are the result of some external influence.

I am inclined to think they are an effect of the strong cell tower signals. The US cell bands are GSM-850 and GSM-1900. For GSM-850 the towers use 869–894 MHz to transmit and GS-1900 uses 1930–1990 MHz. Mobile devices transmit from 824–849 MHz and 1850–1910 MHz, respectively. I'm not sure what the 849–869 and 1910–1930 gaps are used for; I'd guess for administrative activity between the mobile units and the towers, so that towers probably transmit in them. I found a very strong signal at 850.0 MHz—but perhaps spurious—with the NSRT receiver.

[1] `http://www.ettus.com/`

[2] See, for example, *The Scientist and Engineer's Guide to Digital Signal Processing* by Steven W. Smith. This book is on line at `http://www.dspguide.com/`. An outstanding text, with an MIT author, is *Discrete-Time Signal Processing* by Alan Oppenheim and Ronald Schafer (Prentice Hall, 2010)

[3] "An algorithm for the machine calculation of complex Fourier Series," *Mathematics Computation*, **19**, 279–301 (1965).

[4] http://gnuradio.org/redmine/projects/gnuradio/wiki/

[5] http://www.fftw.org/doc/

[6] B. Razavi, IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing, **44**, 428-35, (1997)

[7] Published on line. Try `http://www.users.muohio.edu/mortonyt/` or DOI 10.1007/s10291-012-0263-y

[8] Eugene B. Hogenauer, IEEE Transactions on Acoustics, Speech and Signal Processing **Vol. ASSP-29**, (Issue 2, April) 155-162 (1981)