A Comparison of Affordable, Self-Assembled Software-Defined Radio Receivers Using Quadrature Sampling Down-Conversion

Caleb Froelich
Dr. Rob Frohne, KL7NA
Konrad McClure
Joshua Silver
Jordyn Watkins, KN6FFS

Walla Walla University

ABSTRACT

The software-defined radio (SDR) is a low-cost radio with the potential to draw attention from both experienced and inexperienced operators. As part of a class project, we designed and built quadrature sampling down-conversion SDR receivers. With the receivers we built and the resulting knowledge from that experience, we wanted to contribute to the impact SDR receivers have on growing the amateur radio community. This paper describes the quadrature sampling detector (QSD) in an easy to understand manner, analyzes the advantages and disadvantages of four different SDR receiver designs, and presents a design guide to building an SDR receiver with relevant equations and considerations.

INTRODUCTION

As the future of the world is uncertain amidst the COVID-19 pandemic, the future of amateur radio is equally uncertain. Noting the majority of ham radio operators are in their 60s and 70s and the growth of the number of new amateur radio licenses has held constant at 1% for the past few years, it appears that the pertinent questions are "How do we attract younger operators?" and "How will this younger generation change the course of amateur radio?[i]" Ham radio has already been changed significantly by the introduction of low-cost radios. Handheld VHF/UHF transceivers like those made by Baofeng are affordable and easy to use. And now software-defined radios (SDRs) are bringing that same budget price tag to HF.[ii]

Low-cost is not the only reason SDRs have become more popular among the amateur radio community. More recently, Guido Ten Dolle's μSDX open source transceiver has generated increasing interest in quadrature sampling down-conversion SDRs in the homebrew QRP community. Guido, PE1NNZ, was able to modify the QCX, QRP transceiver for SSB operation with an efficient class-E amplifier, using only an ATMEGA328 and Arduino code to run the QSD SDR. This groundbreaking work in this type of SDR has inspired various renditions of Guido's radio, fostering a lively groups.io group that can be followed at https://groups.io/g/ucx.

For our Electronics II class at Walla Walla University, we designed, built, and tested SDR receivers with the primary goal of furthering our education in electronic engineering concepts while remaining within a $25 budget. Although we had the privilege of many hours of dedicated class time for researching and designing an SDR receiver, we recognize not everyone has this same

opportunity. We resolved to compose an easy to understand explanation of quadrature sampling detectors (QSDs), (the heart of the type of SDR we all designed), provide a comparison of our SDR receivers with an emphasis on aspects in the designs that enhance or diminish performance, as well as provide a design guide to quadrature sampling down-conversion software-defined receivers with design tools, resources, and tips. This information can help flatten the learning curve in addition to supplementing hobbyist interests in this type of SDR, such as Guido Ten Dolle's µSDX transceiver.

FUNDAMENTAL INFORMATION

The fundamental ideas pertinent to QSD SDR radios, especially the four presented designs, are the operation of mixers and the use of quadrature signals. This section will cover each in an accessible level of detail.

The goal of the mixer is to sample the input from the antenna for information. To start with a simple analogy, have you ever seen a video where a plane propeller starts up? If so, you have likely noticed it appears to spin in one direction, slow down until stationary, and then spin in the reverse direction. This phenomenon is useful to explain mixers. Imagine we have a propeller that is spinning, and a camera is set up to take snapshots of the propeller as it spins.

As a visual example, **Figure 1** shows a plot that traces only the vertical height of the tip of one blade of the propeller over time as it spins clockwise at a rate of 10 rotations/second. The line traces the true position, while the small images of the propeller blade along it shows what would be seen if an image was taken at that moment. In this case, the camera is taking pictures at a rate 8 times faster than the blade rotation rate.

To continue the example, if pictures are taken at the same time the propeller makes one full rotation, as shown in **Figure 2**, it will appear stationary as if the propeller is not spinning at all.

Figure 1. Vertical position of the tip of one propeller blade tracked via pictures taken at 8x the rate of rotation.

Figure 2. If the images are taken at the same rate as the propeller makes a rotation, the propeller would appear stationary.

If pictures are taken at a rate slightly slower than the rotation rate of the propeller, like in **Figure 3**, the images will show the propeller slowly turning in the same direction. Conversely, if pictures were taken faster than the rotation rate of the propeller, like in **Figure 4**, the images would show
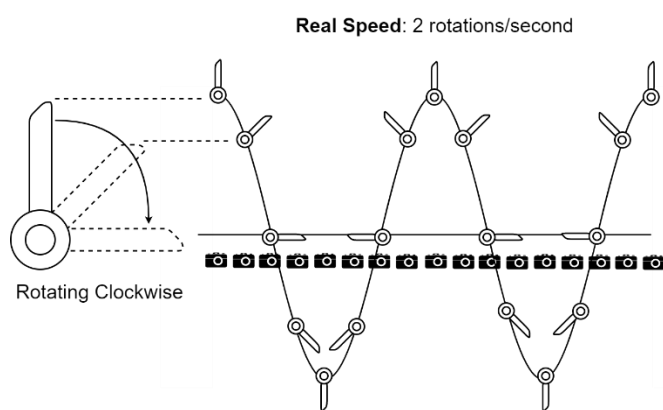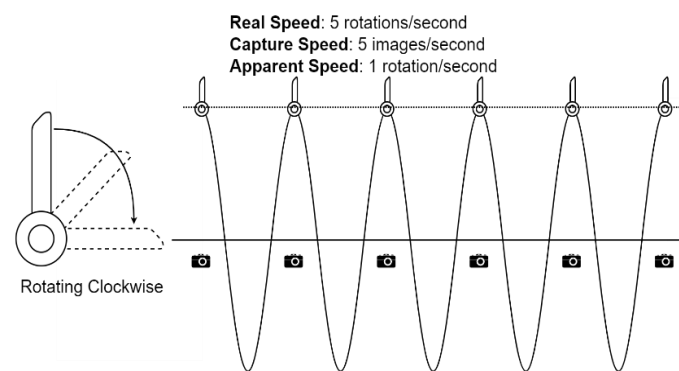
the propeller slowly turning in the opposite direction. Interestingly, the trace of the height of the propeller tip is identical despite the propeller apparently turning in reverse.

By taking snapshots you have effectively changed the apparent frequency of rotation to be the difference between the frequency of rotation of the propeller and the frequency of snapshots taken.

Real Speed: 5 rotations/second
Capture Speed: 6 images/second
Apparent Speed: 1 rotation/second (opposite direction)
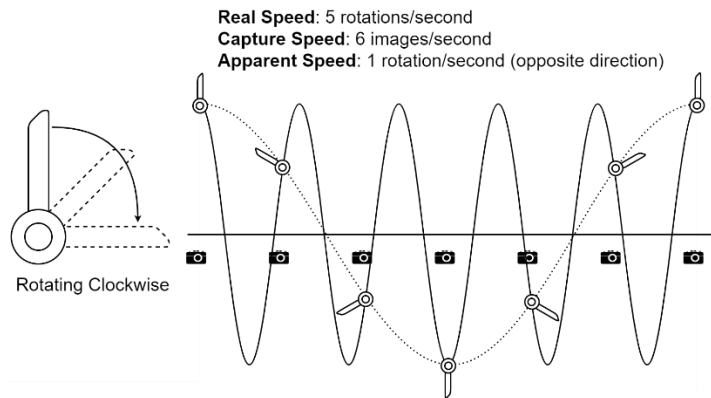
Rotating Clockwise

Figure 3. If the images are taken at a faster rate than the propeller spins, the images display a rotation that is slower than, but in the opposite direction of, the propeller's spin.

As can be seen in the examples above, the frequency of the apparent rotation is equal to *(real speed — capture speed)*. If the capture speed is faster than the real speed, the difference is negative, indicating the propeller will appear to be turning in the opposite direction. All three cases occur when a video is taken of a propeller, though it is the framerate of the video remaining constant while the propeller increases in speed.

Real Speed: 7 rotations/second
Capture Speed: 6 images/second
Apparent Speed: 1 rotation/second (CW)
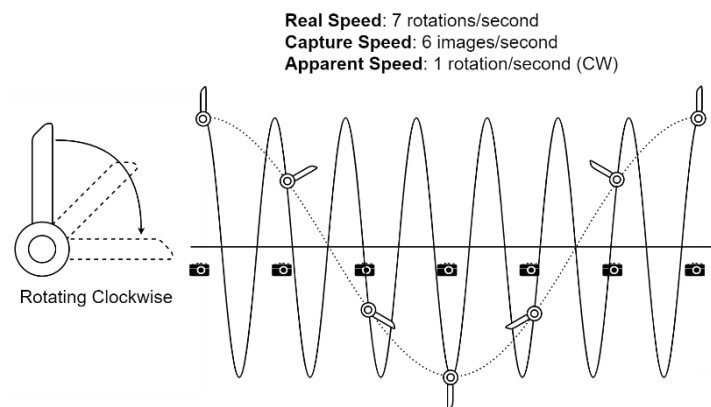
Rotating Clockwise

Figure 4 - If images are taken at a slower rate than the propeller spins, the images display a rotation that is slower than, but in the same direction of, the propeller's spin.

The frequency of rotation of the propeller is like the frequency of the radio signal ("RF signal" with frequency "f_{RF}") you wish to tune to. The frequency of the snapshots is the frequency you can control. This frequency is generated by a clock, known as the local oscillator ("LO signal" with frequency "f_{LO}"). This turns our apparent rotation speed equation from *(real speed — capture speed)* into $(f_{RF} - f_{LO})$. The local oscillator is used to time the switches that take the "snapshots" used to slow down the frequency of the signal you wish to listen to down to a frequency slow enough for your sound card and computer to process. The classic example of an ideal mixer at first appears to be different than the propeller blade analogy, will output both a difference frequency $(f_{RF} - f_{LO})$ as well as a sum frequency $(f_{RF} + f_{LO})$. In the example to the right in **Figure 5**, a 10 MHz RF signal is mixed with a 9.99 MHz LO signal to produce two frequencies: $(f_{RF} - f_{LO}) = 10$kHz and $(f_{RF} + f_{LO}) = 19.99$ MHz. The slow 10kHz signal is the desired
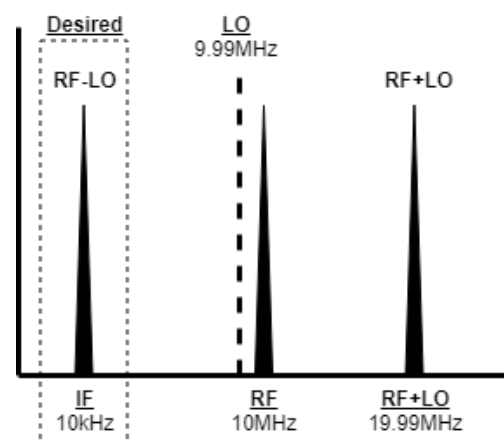

Figure 5. To down-convert a received signal to a slower frequency, the Radio Frequency (RF) and Local Oscillator (LO) frequencies are mixed, resulting in a new Intermittent Frequency (IF)

frequency since the sound card can process it. The 19.99 MHz signal is easily filtered out with a low-pass filter due to its distance from the desired 10kHz frequency. The mixer circuitry used in all the following designs expand on the classic example somewhat by including sampling capacitors on the output, which keep the sampled voltages like the film stores the snapshots of the camera. These capacitors create a low-pass filter that will smooth out the instantaneous points captured into a continuous curve as illustrated in the figures of this section. In the same way, by the nature of the human brain, we assume that between each snapshot is a smooth transition between the two positions, which results in that same continuous "apparent" trace that the low-pass filter creates.

Real Speed (Black): 7 rotations/second
Image Speed (Grey): 5 rotations/second
Capture Speed: 6 images/second
Apparent Speed (Black): 1 rotation/second (CW)
Apparent Speed (Grey): 1 rotation/second (CCW)
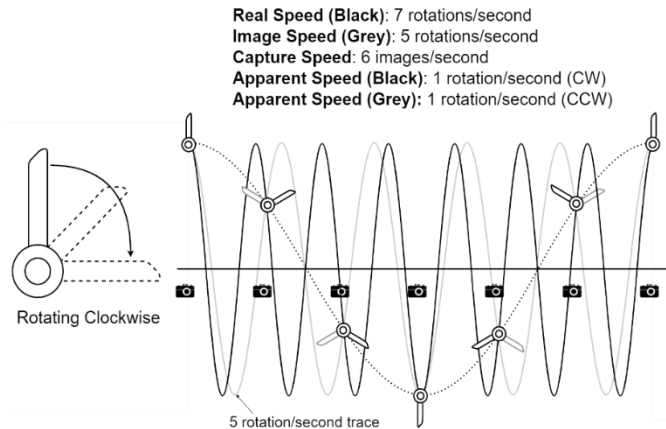
Rotating Clockwise

5 rotation/second trace

Figure 6. The real (desired) and image frequencies create the same plot on the graph when tracking the vertical position. We have no way to tell the difference with just the vertical position of the blade tip.

A key detail of the slower and faster rate examples is that the apparent rates of the propeller both have the same 1 rotation/second rate and produce the exact same trace on the graph. The only difference between them is the direction of rotation. The two are superimposed together in **Figure 6** with the faster rate in black and the slower rate in grey. Unfortunately, there is a key difference between our analogy of taking pictures of rotating propeller and the reality of radio mixers. We are able to identify the direction of apparent rotation from these graphs through the little images of the propeller along the trace of the vertical position. If we only had a plot of the apparent vertical position, however, it would be impossible to tell which direction it was rotating. The two rates, 5 rot/sec and 7 rot/sec are referred to as "images" of each other.

In terms of radio signals and the previous radio example, our desired RF signal would be the 10MHz signal would be down-converted by the 9.99MHz LO frequency to a positive ("CW") 10kHz. However, as shown in **Figure 7**, if there was a second frequency broadcasted at 9.98MHz, that is the same distance from LO as our desired RF (this frequency would be referred to as the Image), the Image would also get down-converted to 10kHz IF frequency, except the Image is negative ("CCW"). As a result, we would hear both broadcasts from the speakers.

RF - LO
or
Image - LO

LO
9.99MHz

Desired

IF
10kHz

Image
9.98MHz

RF
10MHz

The solution in the propeller example is to track both the horizontal and vertical positions of the tip simultaneously, which previously was done artificially by including the small propeller images along the trace. If instead two axes are created to track them simultaneously, the direction of rotation can be determined. **Figure 8** and **Figure 9** display this on the following page.
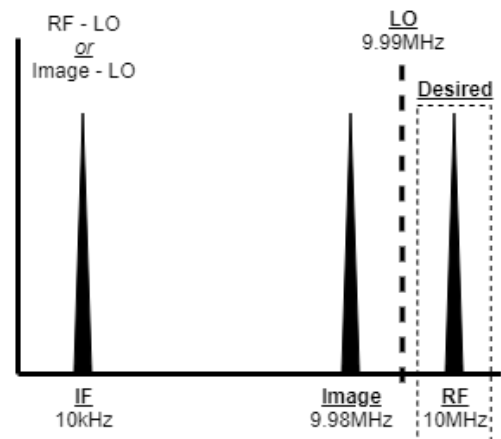
Figure 7. Both the RF and Image frequencies will be down-converted to 10kHz. A method is needed to tell the difference between the positive and negative
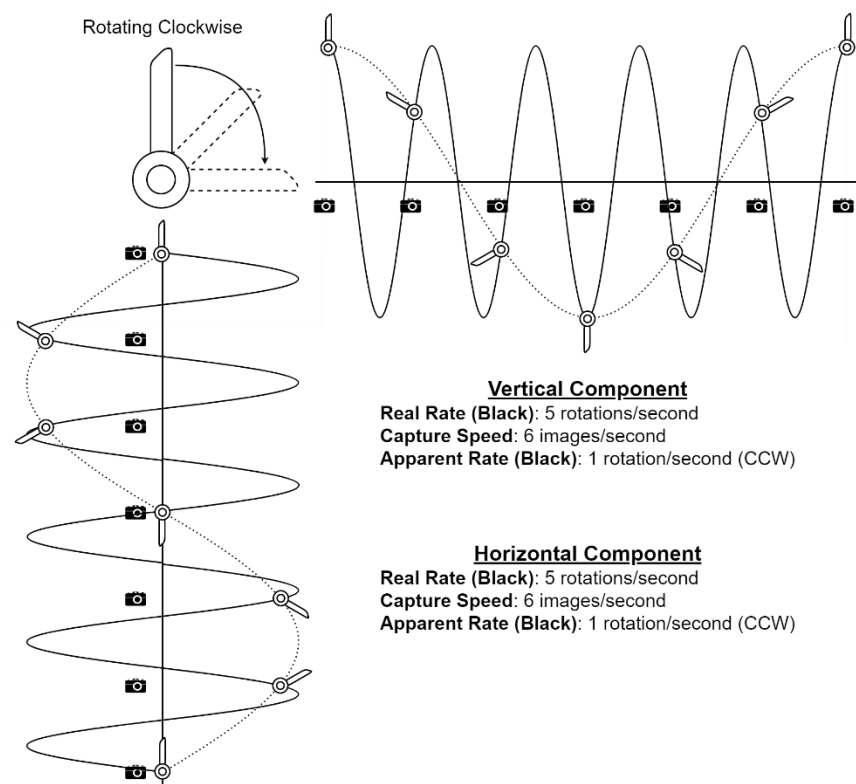
Figure 8. With a rotation rate (RF) slower than the capture rate (LO), the resulting apparent frequency (IF=RF-LO) is negative, resulting in a counter-clockwise (CCW) rotation. This is the undesired Image signal
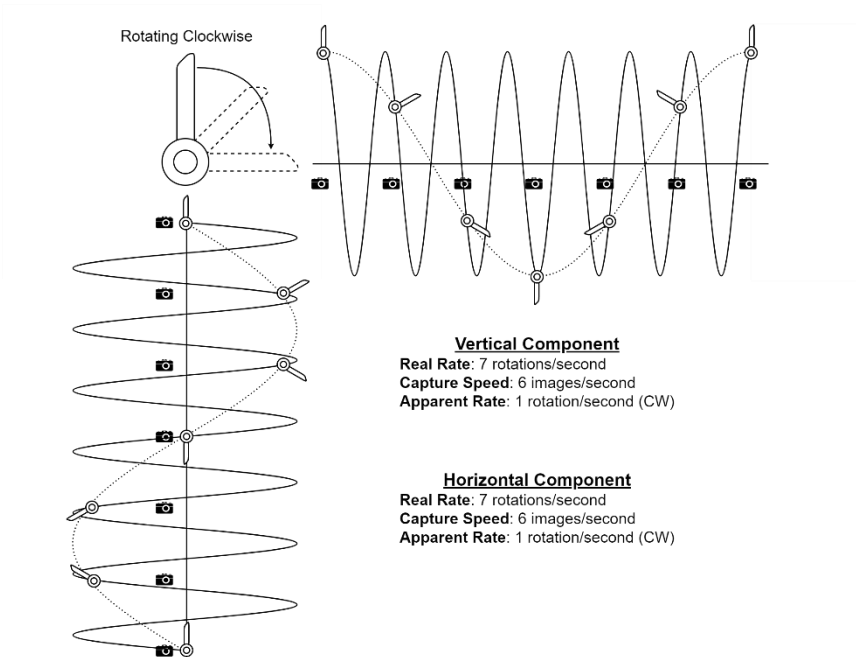
Rotating Clockwise

Vertical Component
Real Rate (Black): 5 rotations/second
Capture Speed: 6 images/second
Apparent Rate (Black): 1 rotation/second (CCW)

Horizontal Component
Real Rate (Black): 5 rotations/second
Capture Speed: 6 images/second
Apparent Rate (Black): 1 rotation/second (CCW)

Figure 9. With a rotation rate (RF) faster than the capture rate (LO), the resulting apparent frequency (IF=RF-LO) is positive, resulting in a clockwise (CW) rotation. This is the desired RF signal.

Rotating Clockwise

Vertical Component
Real Rate: 7 rotations/second
Capture Speed: 6 images/second
Apparent Rate: 1 rotation/second (CW)

Horizontal Component
Real Rate: 7 rotations/second
Capture Speed: 6 images/second
Apparent Rate: 1 rotation/second (CW)

With this, we introduce what are known as Quadrature signals. In their most simple explanation, Quadrature signals are signals that are 90° (quarter of a cycle) out of phase with each other.

Examining **Figures 8** and **9**, we can see that **Figure 8** produces a CCW output while **Figure 9** produces a CW output. Looking closer, we see that while the vertical positions trace the same path as previously observed, the horizontal positions are reflected across the axis. Superimposing these two axes from the vertical and horizontal positions, as shown in **Figure 10**, we can easily see how the "clockwise-ness" is determined. On a positive frequency, the propeller reaches the "upmost position" a quarter of a cycle before the "rightmost position." Conversely, on a negative frequency, the propeller reaches the "rightmost position" a quarter of a cycle before the "upmost position." Since the vertical trace remains constant no matter the sign of the frequency, it is referred to as the In-Phase (I) signal. It is used as a reference point to compare the horizontal trace to, known as the Quadrature (Q) signal. Another illustration of this is seen in **Figure 11**, where each scenario has been superimposed.
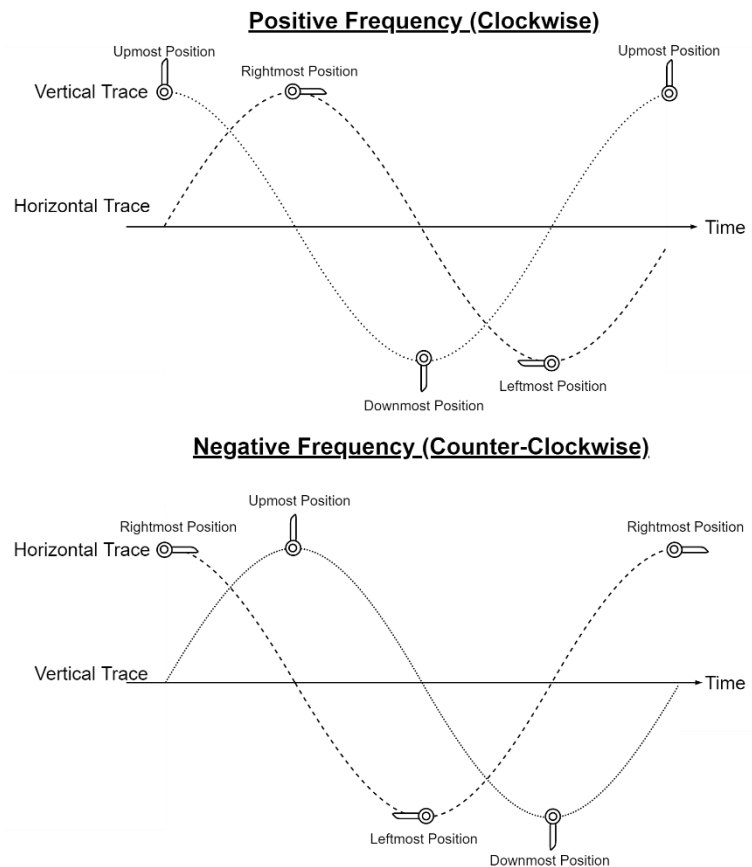
Back to our example in **Figure 7**, the Image signal at negative 10kHz can now be ignored by the software processing the digital signal from the soundcard if it is provided with *both* the I and Q components of the signal. Often, this can be done with two mixers: one that produces the I signal, and another that produces the Q signal. This kind of configuration can be seen in Design 3 using two voltage-controlled switches as mixers. Another method, as can be seen in Designs 1, 2, and 4, is to use
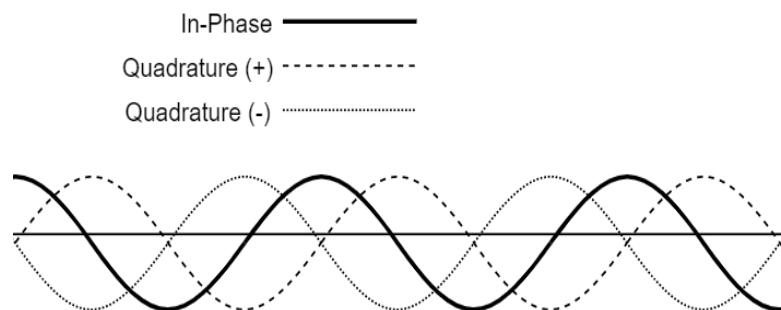

Figure 10. *As time goes on, a positive (CW) frequency will reach the upmost, rightmost, downmost, and leftmost positions in that order. A negative frequency will cycle through the rightmost, upmost, leftmost, and downmost positions in that order. This allows us to identify whether the quadrature signals represent a positive or negative frequency.*

Figure 11. *The In-Phase, Negative Quadrature, and Positive Quadrature signals superimposed to show the phase difference between them. Both quadrature signals are measured relative to the constant In-Phase signal.*

a Quadrature Sampling Detector such as a Tayloe Detector, which converts the input into the I and Q signals directly.

There is one more issue to consider. Returning to the RF equals LO example in **Figure 2** where the propeller appears stationary, consider if the propeller is rotating at twice the rate of the snapshots. It will also appear to be stopped, just as if it had been rotating at the same rate. After some thought, you'd come to realize this is true for any integer multiple (2x, 3x, 4x, …) of the frequency, as shown in **Figure 12**. Evidently, there are an infinite number of rotation rates that can cause any apparent slow rotation rate you might guess by just looking at the snap shots. The same trace will match any frequency of $f = f_{real} + n \times f_{capture}$ where n is any integer, as can be seen in **Figure 13**. Since any value of n will result in the same down-converted trace on both the vertical and horizontal axis.

We only want our system to respond to rotation rates very close to the local oscillator frequency (the 5 rot/sec), not to every harmonic of the local oscillator (the 9 rot/sec, etc.). We want to translate the frequencies very near the local oscillator (around 4 rot/sec) down to near DC and keep only those, not the others (near 8 rot/sec, etc.). If there are no responses to these other frequencies, we can use a low pass filter to smoothly connect the sampling points and keep the desired band of frequencies near the local oscillator, now translated down to DC, and be done. These undesired responses, however, are a problem.

Real Speed: 2, 4, and 8 rotations/second
Capture Speed: 2 images/second
Apparent Speed: 0 rotations/second (stationary)

Rotating Clockwise

Figure 12 - The propeller appears stationary at $f_{real} + 0 \times f_{capture,} = 2\frac{rot}{sec}, f_{real} + 1 \times f_{capture,} = 4\frac{rot}{sec}$, and $f_{real} + 2 \times f_{capture,} = 8\frac{rot}{sec}$

Grey Rate: 5 rotations/second
Black Rate: 5+4 rotations/second
Capture Rate: 4 images/second
Apparent Rate: 1 rotation/second

Rotating Clockwise

Figure 13 - The apparent rate of the propeller can match $f_{real} + 0 \times f_{capture} = 5\frac{rot}{sec}$ or $f_{real} + 1 \times f_{capture} = 9\frac{rot}{sec}$. This will be true for any value of n.

Similar undesired parasitic responses also occur in a radio receiver using instantaneous sampling like in the camera analogy (for both I and Q downconverters). The effect is undesired, because you want to listen to one and only one radio station at once. You want to remove all signals except the desired one near the local oscillator frequency. A key to solving this problem in QSD receivers is realizing the frequency conversion or sampling process is actually weighting (multiplying) the
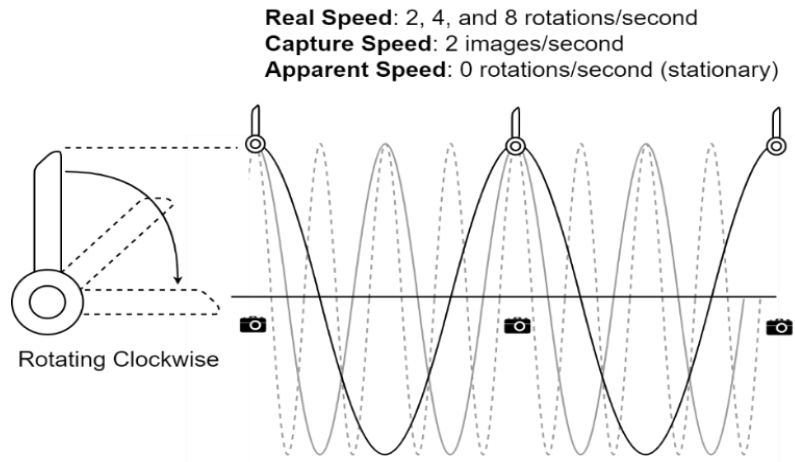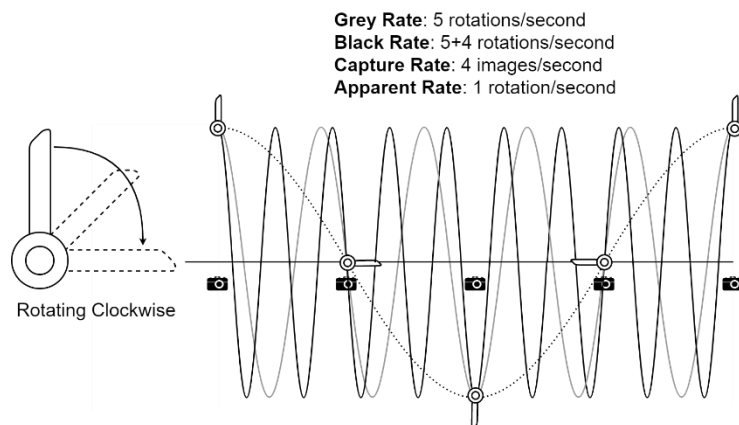
signal by a time varying periodic function. The frequency of this local oscillator weighting function is just the number of Hertz we wish to shift the signal down. Because the weighting function is periodic, it is made up of a fundamental, and usually has harmonics. In the snapshot method, the weighting function is a series of very high short pulses, one at each shutter time when the photos are taken, and zero in between, where the shutter is closed on the camera. Each harmonic component in this weighting function is every bit as big as its fundamental frequency component. The undesired responses come from the harmonics shifting undesired signals down too, right on top of our desired signal. The ideal weighting function would have no harmonics at all, just a single sinusoid at the local oscillator frequency.

Unfortunately, multiplying by a time varying sinusoid in analog electronics is difficult, so it is common to use switches controlled by the local oscillator clock to create a periodic stepped weighting function instead. A simple example of this kind of weighting function is a square wave. The weighting function is selected to have less harmonic content than the snapshot method, thus reducing the undesired responses. It should be noted that sometimes we cannot create a weighting function with a high enough frequency, and in those cases, a harmonic (instead of the fundamental) can be used to down-convert the desired signal to audio frequencies. To really ensure the undesired RF signals do not get through, they are filtered out with bandpass filters before they reach the down conversion stage. If they are not present, we don't have to worry about them interfering with the desired signal, even if using the "snapshot" weighting function.

A single pole single throw (SPST) switch connected to the RF signal multiplies it by one when closed, and zero when open. A center tapped transformer creates, on the secondary, both plus the RF signal and minus the RF signal. A single pole double throw (SPDT) switch controlled by the local oscillator clock accomplishes the multiplication by the square wave weighting function by selecting alternately one or the other. This method is used in Design 3, as shown in **Figure 14**. (Designs 1 through 4 are described in the next major section.) Sometimes amplifiers are used with the switches instead of transformers to set the levels of the segments of the stepped weighting functions. Designs 1, 2 and 4 use this approach.
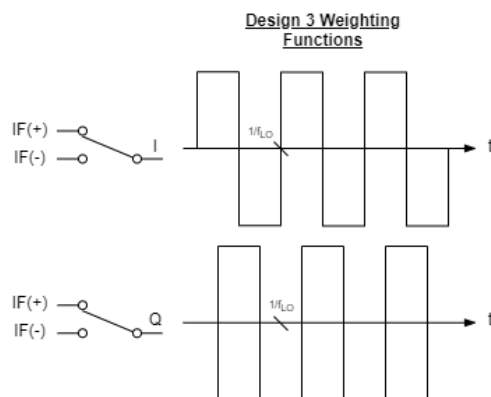

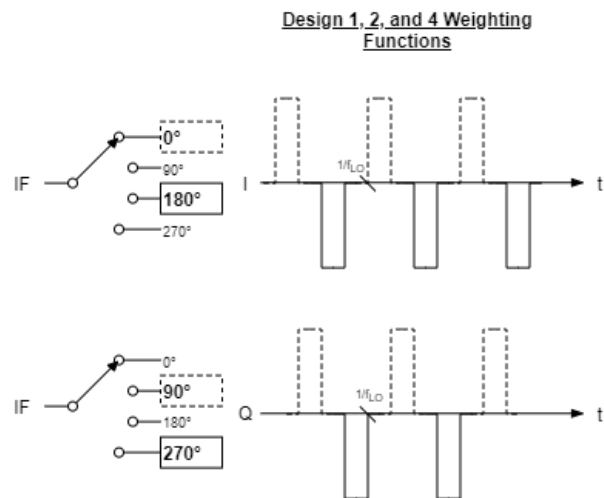Figure 14 - Weighting function used by Design 3.

Figure 15 - Weighting function used by Designs 1 and 2. Design 4 is nearly identical, but the positive pulses are slightly taller than the negative ones.

In the figure above, the weighting functions for the simple mixer and the Tayloe down converter are shown. It should be noted that all these waveforms have no even harmonics, and the odd harmonics all fall off as *1/m*, where *m* is the harmonic number, which means the harmonics are dropped by using them, but not as far as you probably want. To really eliminate the undesired responses, a bandpass filter at the input which only allows the desired band of frequencies in, filtering out the harmonics, is the solution. To ensure only the output connects the smoothest function through the samples and the desired band of low frequencies which was shifted down remain in the result, we filter it with low pass filters, consisting of at least the sampling capacitor, and often further audio filters in subsequent stages.

Designs 1 and 2 use the Tayloe mixer weights shown in **Figure 15**, and Design 3 uses the square wave weighting, while Design 4 uses the Tayloe mixer with a circuit where the 0º and 90º signals are weighted slightly more than the 180º and 270º signals because of the amplifier gains in that circuit. Thus, upon close examination, that circuit would have very slightly higher positive going pulses, but the effect is too small to be visible on this scale.

DESIGN ANALYSIS AND COMPARISONS

We have four SDR receivers from which we will evaluate and compare the designs and draw conclusions. One was designed by Caleb Froelich and Konrad McClure (Design 1), a second by Joshua Silver and Jordyn Watkins (Design 2), and the third and fourth radios were designed by Rob Frohne (Design 3, Design 4).

We explore the differences between our radios with a focus on the aspects in the design that affect performance. We evaluate each of the circuits in the receiver designs, specifically focusing on the following: bandpass filter, local oscillator, mixer, amplifiers, and low-pass filter.

BANDPASS FILTER (BPF)

Bandpass filters are widely used in radio receivers as they allow the designer to change the frequency agility of the radio receiver. Design 1 utilizes a fixed 3rd order Bessel LC bandpass filter. By using a series-first topology, a coupling capacitor can be saved. While a design with a fixed bandpass filter is easier to make, maintain, and fit on a small board, it doesn't offer the flexibility and performance that switchable bandpass filters allow radio enthusiasts.

Designs 2 and 3 both utilize switchable bandpass filters, controlled via two digital output pins on an Arduino Nano v3. With careful planning, the switchable bandpass filters better eliminate spurious signals as discussed above and reduce the overall noise. This increase in performance comes at the price of design complexity. With the introduction of multiple BPFs, the need for a switching circuit (a 4:1 multiplexer is a perfect solution) and software to control the filters are needed. Additional bandpass filters also add more passive components to the board, taking up precious PCB space.

For a designer who has a fixed frequency band with which he would like to communicate, a single bandpass filter is more than adequate. The pure minimalist could even neglect using a bandpass filter. Excluding the BPF exchanges the inclusion of a few spurious signals for a receiver that is

light, compact, cheap, and easy to assemble and debug. Design 4 takes on a composite approach, placing pin headers for an optional bandpass filter to be installed post-facto.
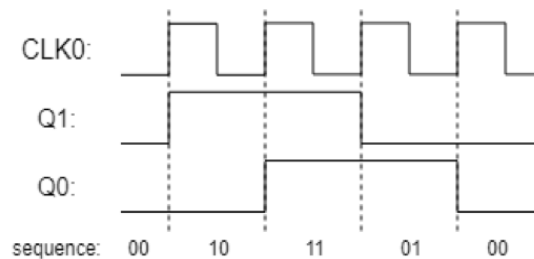
LOCAL OSCILLATOR

The clock generator and local oscillator (LO) configuration is homogeneous for the first three designs. A Si5351a clock oscillator is used to generate the LO signal and dual flip-flops connect as a divide-by-four Johnson counter to produce the I and Q local oscillator signals at one quarter the frequency of the driving clock. This approach offers a large bandwidth, as the LO can vary from the lower limit of the Si5351a of (2.5 kHz)/4 to the maximum frequency of (200 MHz)/4.

Following the design of the µSDX transceiver by Guido Ten Dolle, Design 4 capitalizes on the capabilities of the Si5351a to generate the I and Q signals without the need of flip-flops. Programmable via the I²C protocol, the Si5351a allows you to lock the phase of the two clock outputs, to generate two signals at the same frequency and in quadrature. The Si5351a does, unfortunately, place a limitation on the range of frequencies that can be generated. Utilizing the Arduino Etherkit Si5351 library, written by Jason Milldrum, NT7S, the minimum quadrature frequency that can be generated should theoretically

Figure 16. Square wave visual representation of how the counting sequence differs between the two types of local oscillator configurations.

be 600 MHz/128 = 4.6875 MHz, because the phase register is 7 bits and the minimum PLL frequency is 600 MHz. However, through direct I²C programming, Guido works the minimum quadrature frequency down to 3.2 MHz, fully encompassing the popular 80-meter band (3.5 MHz to 4.0 MHz).[iii]

If the minimum frequency of the quadrature method is not low enough for the desired application, there is an alternative solution. By utilizing two clocks, a sequential counting sequence (00, 01, 10, 11) can be obtained by setting the low order bit to change at two times the desired local oscillator frequency. The higher bit is then set to be half the frequency of the low bit using a divider built into the Si5351a that keeps the bits synchronized. A timing diagram comparing the counting method to the quadrature signals generated via a Johnson counter is shown in **Figure 16.** Interestingly, generating quadrature signals directly from the Si5351a increases image rejection. The quadrature method utilized in Design 4 provides 40 dB of image rejection, and Guido gets 45 dB. Experimentation and testing showed that the Johnson counter LO in Design 2 has 30-35 dB of image rejection, while the well-known SoftRock RXTX we tested only has 25-30 dB depending on the part of the passband you are in. Bottom line: use the quadrature method for frequencies above 3.2 MHz and the counting method if frequencies below 3.2 MHz are desired. Both

approaches will result in more software than utilizing a Johnson counter, but they cheaper solution and save board space.

MIXER

An important consideration in the design of SDR receivers is selection of a mixer topology that combines the signal of interest and the local oscillator output. When evaluating mixers, it is imperative that one compare the conversion losses, or the ratio of the RF input power to the IF SSB output power, expressed in positive dB [iv]. The lower the conversion losses, the more efficient a mixer is at converting energy from the RF to the IF signal. Noise is also a concern and should be considered when designing a mixer.

While there are numerous types of mixers, we utilized two common designs. Design 3 utilizes a doubly balanced sampling mixer, essentially a switch with a capacitor feeding into a summing amplifier. If the amplifier selected has a high impedance input, the capacitor will hold the input voltage when the switch is open. This allows the duty cycle of the switch to be lowered, which results in a lower conversion loss. Design 3 does not have this advantage as the amplifier configuration utilized has a low impedance input, as we'll elaborate on in the next section. The mixer in Design 3 is doubly balanced due to the center tapped transformer that negates the RF input signal for half the period.

An extension of the sampling mixer design well known among radio amateurs is the Tayloe detector. Despite its striking similarities with the sampling mixer, Dan Tayloe emphatically states that his detector is not a mixer since it only produces the difference frequency. Perhaps that means Design 3 uses a sampling detector too. All but one of the designs presented in this paper utilized the Tayloe detector due to its compact, simple design and low conversion loss (less than 1 dB). In a comparison of switch-based frequency converters by Michiel Soer, it was concluded that the "a double balanced Tayloe mixer with 25% duty cycle provides the best balance between noise figure and conversion loss" [v]. The low-cost, high-performance Tayloe detector is one of the better options for SDR receiver designs.
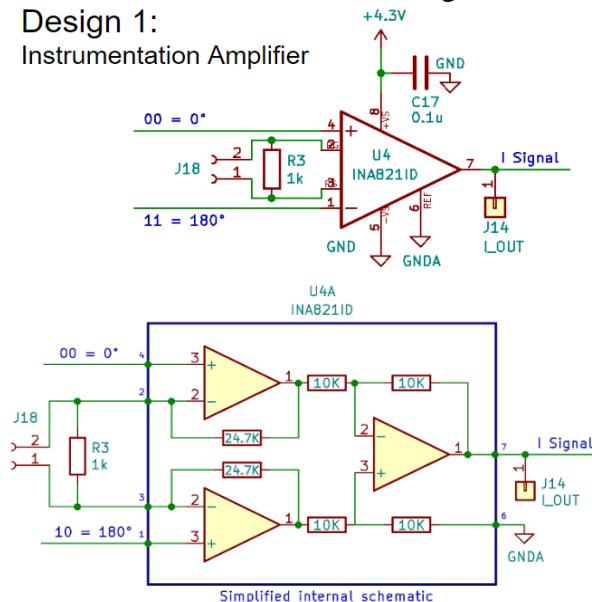
AMPLIFIERS

While the first two designs (Designs 1 and 2) share many fundamental similarities: (they both utilize bandpass filters and a Tayloe (I/Q) mixer to convert the radio spectrum to a low IF which is then demodulated via software) the key difference between the two lies in the amplification of the IF signals. Design 1 utilizes instrumentation amplifiers while Design 2 utilizes a basic differential amplifier. The biggest difference between these two amplifiers is the input impedance. The instrumentation amplifiers provide a very high input impedance which makes the amplifier more like a voltage amplifier, while the differential amplifier more closely resembles a trans-resistance amplifier due to its lowered input impedance. The input impedance is a byproduct of the type of amplifier topology. As seen in **Figure 17**, the instrumentation amplifier used in Design 1 internally utilizes a non-inverting topology which completely isolates the input signal from the output. On the other hand, the inverting topology of the differential amplifiers allows for current to sneak through the feedback loop. This difference in amplifier topology, and subsequently the input impedance of the amplifier introduces a multitude of intriguing effects that influence other aspects of SDR receiver design.

For one, the difference in input impedance affects the way that the mixer sampling capacitor functions. With the high input impedance of the non-inverting topology, the sampling capacitor follows the RF signal when the switch is closed and holds the voltage level when the switch is open, since there is no path for the capacitor to discharge. However, the inverting amplifier with its lower input impedance doesn't charge the sampling capacitor as much.

A voltage amplifier with high input impedance also uncouples the amplifier stage from other sections of the design, discretizing the SDR receiver into a system of easily understood blocks. The many interdependencies experienced with a coupled amplifier design (like Design 2) are eliminated, allowing for easier design. Debugging is also easier in a voltage amplifier like Design 1. The tools that we have at our disposal (oscilloscope, multimeter, etc.) measure voltage much easier than current. Since each designer shared the same primary goal of furthering their
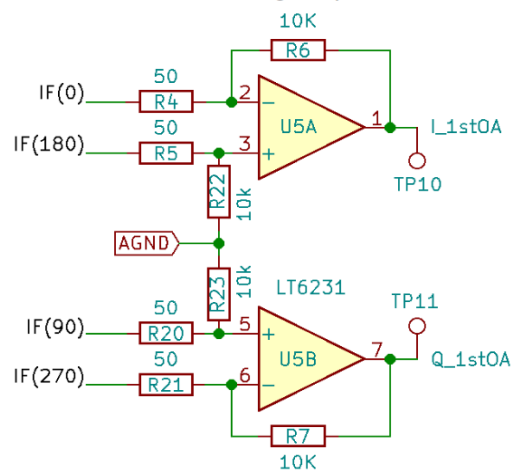

Figure 17 - Comparison between amplifiers used in Designs 1 and 2.

understanding of electronic engineering concepts, the simplicity and understanding that the instrumentation amplifier provides cannot be understated.

Additionally, instrumentation amplifiers remove the ambiguity of antenna impedance from the amplifier gain equation. Design 1 utilized an INA821ID instrumentation amplifier. Since a single resistor sets the gain of this amplifier, it would be trivial to provide two or more gain settings through a relay or switch. This would allow for further optimization and customization of the radio receiver.

The primary advantage of the differential summing op amp is that it can possibly provide a lower noise figure especially at low gain settings, but for HF the atmospheric noise is large enough that if you pick low noise parts, your design should be fine. The voltage noise and noise figure of instrumentation amps increase with decreasing gain[vi]. Because of this, instrumentation amps are less attractive for low gain situations below about 20 dB, though the differential summing amplifiers have similar problems. The price and availability can make differential summing op amps a more appealing option for a Ham radio amateur on a low budget. The single-circuit

INA821IDs utilized in Design 1 were 6% more expensive than the dual-circuit LT6231s utilized in Design 2.

Despite their dissimilarities, both the differential summing amplifier (utilized in Design 2) and the instrumentation amplifier (utilized in Design 1) take the difference between the IF signals ($v_{0°}$ and $v_{180°}$, $v_{90°}$ and $v_{270°}$) In the case where a specific linear combination of the signals is desired (i.e. $v_{0°} - v_{180°}$) and the other combinations are undesirable, then a differential amplifier or instrumentation amplifier is optimal. Differential amps magnify the difference signal and reject the common mode signals through clever balancing of the circuit. But in our case, either $v_{0°}$ or $v_{180°}$ or any other linear combination of these signals can be used. Dan Tayloe's quadrature detector design exploits this fact by eliminating all the input and bias resistors and maintaining just the feedback resistors of the amplifier. This results in the non-inverting path gain not equaling the inverting path gain, and the amplifier does not sum differentially. A side effect of straying from the classic differential amplifier is a larger common mode gain, however in this case the common mode gain is one, just perfect, so the amplifier in Design 4 doesn't need to be DC biased like the amplifiers in designs 1 and 2. This produces a more simplistic design with less parts, reduced power loss and a decreased input noise level.

Figure 18. Amplifier used in Design 4 which follows Guido's uSDX design.

Design 4 follows Tayloe's approach for amplification yet maintains the input resistors before both inverting outputs so that the gain of the amplifiers is not solely dependent on antenna/system resistance. Resistors before the non-inverting capacitors (R18 and R19) are included in some designs to make the system more symmetrical. However, they have no effect on the operation of the amplifier, other than adding a little noise, so the present plan is to replace them with wire. The two capacitors on each feedback path allow for slightly different cutoff frequencies and eliminate the need of a separate low-pass filter. Two filters are necessary in Design 4, with the bandwidth set at about 6 kHz, as all the digital signal processing (DSP) can optionally be done on the board itself and without the horsepower of a PC using Guido's great code. This allows the sampling to be conducted at a slower pace so that the processor has more time to filter the signals, but no spectral display is possible using only the MCU.

LOW PASS FILTER

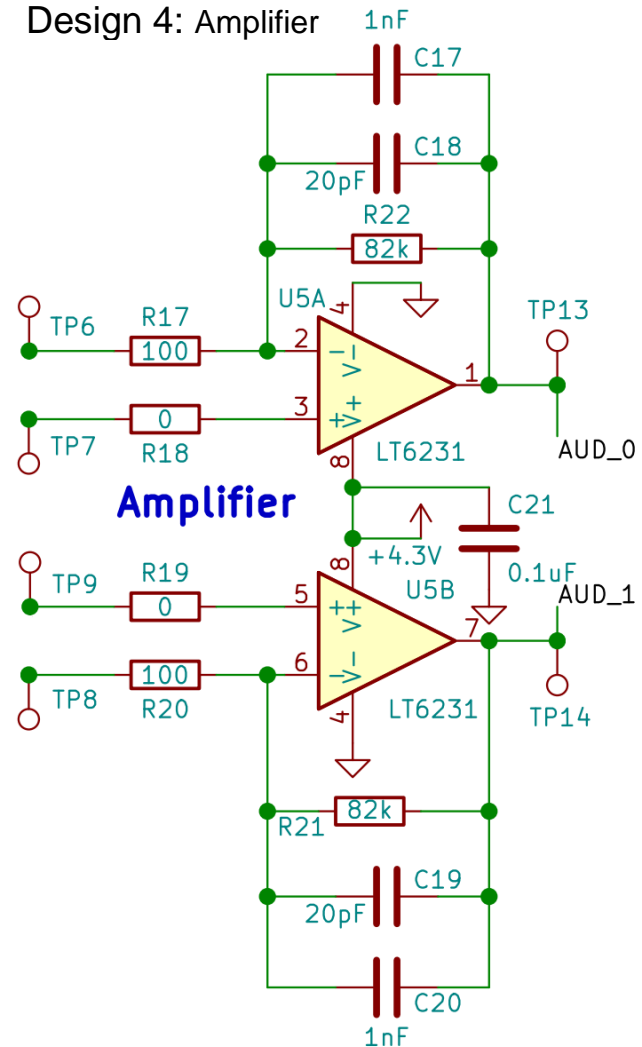Separate low-pass filters are utilized in the first two designs. Design 2 uses the low-pass filter to distribute the overall gain of the receiver across two different amplifier stages. The differential summing amplifiers have a theoretical gain of 46 dB (200 V/V) and the low-pass filters contribute another 20 dB (10 V/V) of gain. By placing more gain in the preamplifier stage before the signals get too big, the amplifiers in the low-pass filter don't have to be as robust. Design 2 could have avoided the use of a low-pass filter altogether by simply placing a capacitor in parallel with the feedback resistors of the summing amplifiers, however, the 2nd order Butterworth filter boasts an extra pole of attenuation over Designs 3 and 4. Design 1 must use a secondary low-pass filter since the negative feedback loop occurs internally inside the INA821's. Some instrumentation amplifiers can be purchased with access to the feedback loop; however, these amplifiers are far too expensive for a low-budget radio project. Design 1 utilizes an LPF with no gain since the gain is already controlled by instrumentation amps. Additionally, the noise figure for the INA821s gets better with increasing gain, thus, minimizing the gain on the low pass filter helps improve overall system noise.

Design 1: 2nd Order LPF
- no gain
- 100kHz cutoff frequency
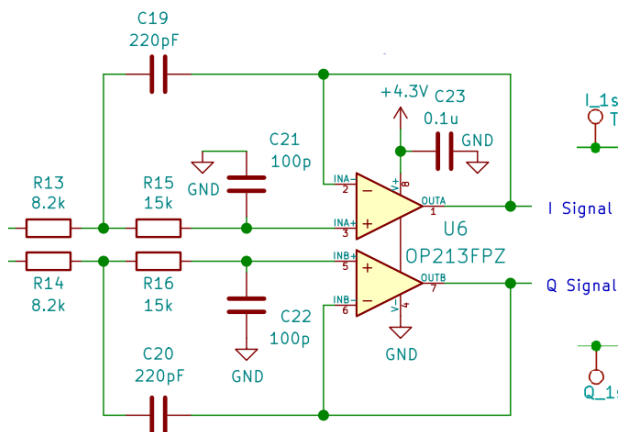
Design 2: 2nd Order Butterworth LPF
- 10 V/V gain
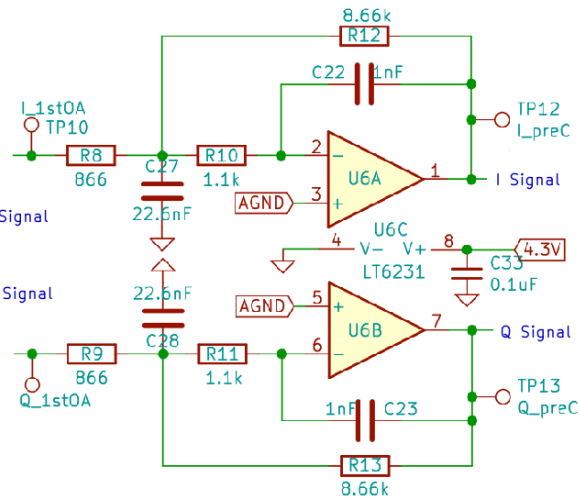- 100kHz cutoff frequency

Figure 19 - Comparison of 2nd order filters used in Designs 1 and 2.

OTHER DESIGN NOTES

Ground loop noise is minimized in Design 2 using a transformer to isolate the radio antenna from the receiver. Experimentation and testing found that the noise level was reduced by nearly 20dB (from –90dB down to –110dB) thanks to the inclusion of the transformer. The transformer is directly responsible for removing ground loops.

Grounding problems can arise when there are multiple different paths to ground. A ground loop is formed when there are two ground connections between a component and the ground plane. This loop acts as a single-turn inductor and can infuse substantial noise into the audio signal, corrupting the signal. Since SDR radios utilize extremely sensitive audio cards, it is imperative to take

precautions to avoid the devastating effects of ground loops (see the design guide for more details) [vii].

Maintaining a clean power source is also important towards minimizing noise. When powered via a USB, the SDR can experience a lot of noise due to the rapid switching that happens in computers. The solution utilized in all the designs is a common emitter (CE) amplifier that regulates and smooths the output voltage at the emitter. In the first three designs, an RC low pass filter with a cutoff frequency of 16 Hz is used. Design 4 utilizes an LC low pass filter with a cutoff frequency of 90 Hz. Even though this is a second order filter, it did not provide appreciable improvement in the noise levels, whereas the RC network in Design 3 did. It is not recommended to copy this circuit as it was not carefully designed. The only downside to the CE amplifier is it comes with an approximate 0.7V drop across the transistor. Depending on the IC's selected in the design, this 0.7V loss could be significant. Including a bypass jumper allows flexibility to test operation on 4.3V or 5V to determine which is preferred.

	MDS (µV)	Image Rejection(dB)	Power Consumption (Watts)	Cost ($)	Size (cm x cm)
Design 1	0.5	30	—	$30	10.0 x 10.0
Design 2	0.5	30	—	$25	10.0 x 10.0
Design 3	0.1	15	5V · 0.027A	$25	5 x 10
Design 4	0.5	33-40	5V · 0.035A	$25	5 x 10
SoftRock RXTX	0.2	25-30	12V · 0.028A	$75	6.35 x 12.7

Table 1. Quantitative Design Comparison performance summary.

DESIGN GUIDE

After spending many hours researching quadrature sampling down-conversion receivers, we decided that creating a condensed design guide to creating a QSD SDR receiver would be beneficial to the amateur radio community and draw interest from a less experienced audience that may have otherwise become lost in the process. The QSD receiver is best for simplicity, power consumption, cost and size. However, if better image rejection and the ability to tune many receivers with the same SDR are desired however, a digital down conversion (DDC) receiver is more appropriate.

The first thing to consider when designing this type of SDR receiver is, "What is your goal for building one?" Is your aim to experiment with electronics? Have a working radio for cheap? Take the radio backpacking? Is there a specific range of frequencies you are hoping to listen to? Do you want it to be as small and lightweight as possible? Or are you shooting for building the highest quality radio possible? These are all questions you should ask yourself beforehand, as the answers can greatly affect your design choices.

In this design guide, we try to break down the design process into a step by step manual. At each step, we provide tables with several design choice options and explanations of when using each option may be appropriate based on your goal with your radio. We also provide relevant equations, links to online design tools, and general board construction and testing tips we found helpful when designing our own radios. We specifically address the documentation and design preparation,

hardware design, circuit simulation, board layout and build, and software with regards to software-defined radio design and construction.

DOCUMENTATION AND DESIGN PREPARATION

The documentation and design preparation are foundational for any project. Keeping careful documentation is important because it can not only help remind yourself of the design and test procedures conducted, but it can also be useful to others who are looking at your project for reference. Documentation can take many forms, from paper and pencil to a GitHub repository. Keeping careful record of corrections needed, mistakes made, concepts learned and more can be of great benefit in the long run. We found Kicad's[viii] schematic capture tool, EEschema, was a real help in documentation and design.

Design Planning: When designing electronics, it is helpful to create a block diagram as shown in **Figure 20**. Each block can then be designed separately. In addition to designing the project in segments, try to anticipate problems and incorporate possible solutions in your design. In concordance with anticipating issues, make trouble shooting the issues easier by adding many test points and pins for bypassing entire sections. Being able to identify the problem can sometimes be even more challenging than resolving it.
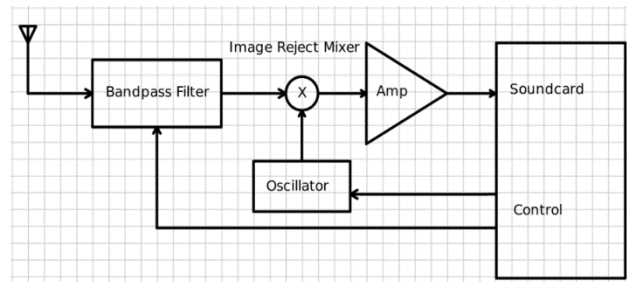

Figure 20. Example Block Diagram

Circuit Simulation: It is crucial to confirm that the design will work using a simulation software before building a prototype. For our receivers, LTspice was used to simulate each component. This will not ensure that the final design will work. However, this will prevent many mistakes that could have been made. It is recommended to start simulating with "ideal components" as this will speed up runtime and makes simulating easier. Once a design is complete and you are ready to pick parts then you can simulate real-world components to find parts that will work for your application. The circuit simulation is discussed in depth in a later section.

Component Selection: There are multiple options when selecting components to use on your circuit board. One of the more important things to keep in mind is to select components you are comfortable soldering. Many individual components (resistors, capacitors, ICs) come in different packages. A summary of the two primary types of packages described in **Table 2.**

TYPE	DESCRIPTION
SMD/SMT	Surface Mount Device/Technology is smaller and is generally soldered onto the surface of the board using solder paste. An oven reflows (melts) the paste which then cools and hardens onto the solder pad. If you do not have access to an industrial reflow oven, a $20 toaster oven can work just fine. These components tend to behave more ideally than their THT counterparts because they have no leads. Leads have unwanted parasitic inductance and capacitance. Another advantage is this package type can be cheaper than THT. Furthermore, if you

	have the equipment and steady hands, it is easier and faster to assemble boards using this package type. However, we recommend not using anything smaller than the 0805 package size, especially if you are inexperienced placing components by hand with tweezers.
THT/THD	Through-Hole Technology/Device components, an older technology, are generally larger and the pins of the device stick through the circuit board which can then be secured using a standard soldering iron. This is by far the simpler method for those with little to no experience. It is also cheaper to begin with as you do not need to purchase any additional equipment to attach these components. If you anticipate needing to switch out component values, designing plug-in sockets for THT components can serve as a potential solution.

Table 2. Comparison of component package types.

It is also important to choose components that are not obsolete and are easier to find. The simplest way to do this is to go to your favorite electronics part's website (Mouser or DigiKey work great) and sort the parts you are looking for by availability. Manufacturers will stock the parts most used in industry. When in doubt, check the datasheets for the components to confirm the specs. It's also important to choose these parts with high availability as the lead-time to order parts not in stock can take at least 6-8 weeks.

HARDWARE DESIGN

General Good Hardware Design Practices: Use a bypass capacitor at every IC and transistor. A bypass capacitor is a capacitor from the DC power pin to ground. It needs to be as close as possible to the IC or transistor it is bypassing. Often 0.1uF ceramic capacitors are used. The purpose is to ensure that the DC power supplies look like short circuits for AC currents. If they are not used the parasitic impedance of conductors connecting them to the actual DC source cause problems the designers were not anticipating, because they designed as if DC sources were true AC shorts.

Bandpass Filter Selection: As discussed previously in the design comparisons, there are several options when considering using bandpass filters. The type of bandpass filter selected is dependent on the type of radio you desire to build. The first question to consider when selecting a bandpass filter is the preferred filter complexity. **Table 3** summarizes reasons for selecting to omit the use of a bandpass filter, use a single fixed filter, frequency specific filter, or opt for a switchable bandpass filter.

FILTER COMPLEXITY	PURPOSE
No Bandpass Filter	Desired if the user prioritizes simplicity over a clearer signal, or perhaps wants to add an external filter later.
Single Fixed Bandpass Filter	Ideal for receivers where the user knows in advance a specific range of frequencies they wish to listen to.
Switchable Bandpass Filter	Preferable for users who wish to eliminate as many spurious signals as possible while maintaining the ability to listen to a large range of frequencies. One octave (the lower cutoff frequency is half the upper) is a good design choice when using multiple filters.

Frequency Specific Filter	If certain signals are known to be problematic, filtering out only the problematic signals such as a local AM station may be adequate.

Table 3. Different bandpass filter complexity selections and suggested application for selection each type.

After determining the complexity of the filter, the next step is to decide the *type* of filter. LC filters are the best choice for most HF radio applications, so a list of different types of LC filter configurations are listed in **Table 4** along with their advantages and disadvantages. It should assist in determining which type of filter is appropriate for your application.

The final few things to consider when designing the bandpass filters are selecting the frequency range, input and output impedance, the order, and choosing a series-first or shunt-first configuration. Once these characteristics are determined, the next step is to calculate the inductor and capacitor values. These values can be easily calculated through an online calculator such as that at RF Tools (https://rf-tools.com/lc-filter/).

The center frequency and bandwidth are selected based on your interest. Note that the wider the bandwidth, the lower the quality. Likewise, a narrow bandwidth increases quality. It is also important to match impedances. When the input impedance does not match the antenna and the output impedance does not match the load, the filter will not have the desired response. When matching impedances, the input impedance does not necessarily have to equal the output impedance. Rather, the filter tools give us the freedom to choose different input and output impedances. This is advantageous when designing the filter because we can have the input impedance equal to that of the antenna, then choose the output impedance to match that of the remaining circuitry, specifically the impedance of the amplifiers. For example, if we were using an instrumentation amplifier which has a theoretical input impedance of infinity, it would be appropriate for us to choose an output impedance of $10M\Omega$ and input impedance of 50Ω for our bandpass filter. The order of the filter determines the effectiveness of the filter. The higher the order, the steeper the roll-off, and therefore the more effective the filter. However, as the order is increased, the magnitude by which the effectiveness is increased decreases, because nominal filter component values are never exact. Typically, we find a sweet spot around the 3rd order. Lastly, one can choose between series-first (T) or shunt-first (Pi) configurations. If you are using 50Ω and $10 M\Omega$ as in the previous example, shunt is preferable over series because the series filter will require component values that are very hard to make or procure.

FILTER TYPE	ADVANTAGE	DISADVANTAGE
Butterworth	Very flat response within passband with essentially no ripple. Low level of overshoot. Linear. Tolerant of component variation.	Reaches ultimate roll-off rate more slowly thus the performance around the cut-off frequency is poor.
Chebyshev	Provides faster transition from passband to stop-band. Steep roll-off provides significant attenuation of unwanted out of band spurious signals and limiting harmonics.	This fast transition comes at the cost of in-band ripple. Less tolerant of component variation.

Elliptic	Produces the fastest transition from passband to stop band of any type of filter.	Presence of gain ripple in both passband and stop-band.
Bessel	Very flat group or linear phase delay. Therefore, it is ideal for applications where the wave shape of signals within the passband must be preserved. No overshoot. Linear.	Slower transition from passband to stopband than other filters in this table, of the same order (slow cut-off).

Table 4. Comparison between the different types of bandpass filter topologies and their advantages and disadvantages [ix].

With the design of the bandpass filters complete, the last task is to gather components. You should strive to get inductor and capacitor values as close to the calculated ones as possible or simulate the filter in LTspice or similar circuit simulator to visualize the effect of variations in component values. Some components, such as bypass capacitors, are not very sensitive to the value, but filter components are. For the capacitors, surface mount capacitors work slightly better, because of their low parasitics, but really either works perfectly fine for most HF filters. The inductors can either be purchased pre-wound as a through hole or surface mount part (caution: make sure that it is rated for the appropriate frequencies), or you can make them yourself using powdered iron toroids and wire. Some inductors are lossy. These are less desirable, and if used, you need to simulate the losses, and make sure you are okay with the effects. There are several online winding calculators, two that we've found to work well are:

- http://www.66pacific.com/calculators/toroid-coil-winding-calculator.aspx
- https://toroids.info

Hand-wound transformers provide a high-quality solution by minimizing losses. However, they are not the cheapest option, and take time, effort, and testing to ensure proper operation. Because winding inductors by hand tends to be time consuming and tedious, generally we try to select bandpass filters with central frequencies and bandwidths that enable us to limit the number of inductors needed in a circuit and keep their values low.

An important note when doing the board layout for the bandpass filters is to recognize the influence the components may have on each other. Generally, iron powder or ferrite toroid cores will keep their magnetic field relatively close, allowing the designer to place the components close to each other without any issues. However, if the designer chooses to use an air core, then mutual inductance may become an issue. It's also good practice to add a surplus of test points, with at least one before the filter and one after for each band. For further ease of testing, we recommend connecting the bandpass filter to the rest of the circuit using jumpers. This way, the user can easily bypass the bandpass filters for troubleshooting purposes.

Ground-Loop Avoidance: Simple solutions can often be used to solve the ground loop problem. Design 1 places a jumper wire from a component to the ground plane that helps break ground loops. While this is simpler and cheaper than winding a transformer, it's not as universal. Jumper wires can be finicky and require readjusting when the receiver is moved to a new QTH. We recommend using a 1:1 transformer between ground and the antenna. This solution is reliable and rugged in mitigating ground loops. If hand-winding a transformer using wire and a toroid core, the

following equation can be used to calculate the inductance of the transformer given an impedance of 50 Ω:

$$wL_m \gg Rs \quad \Rightarrow \quad 2\pi f L_m = 50\Omega(20) \quad \Rightarrow \quad L_m = \frac{50\Omega(20)}{2\pi f}$$

With the inductance calculated, you can now use an online calculator like the ones listed previously to determine the number of windings. Another method used to help avoid ground loops is to have a single ground reference plane on the circuit board and use flood fill with ground on both sides of the printed circuit board. Some radios may not need any ground loop mitigation. For example, when using a battery connected very close to the PCB, with a self contained SDR like Guido's or Design 4. In general, if there is more than one connection to ground, then watch out for ground loops.

Voltage Smoothing: If you use one of these circuits to remove noise coming from the computer's USB port, use the one from Design 3, rather than Design 4. The latest one was not carefully designed and did not improve the noise, while the voltage smoother from Design 3 did in a significant way.

Local Oscillator: There are various options for local oscillators, but all four designs from this paper used the same chip, the Si5351A. As mentioned previously, three different configurations were used and are summarized in **Table 5.**

TYPE	PARTS REQUIRED	COUNTING ORDER	FREQUENCY RANGE
Johnson Counter Circuit	2 (Dual D Flip-Flops, Si5351a)	00, 01, 11, 10	2.5/4 kHz to 200/4 MHz
Counting Method	1 (Si5351a)	00, 01, 10, 11	< 3.2 MHz
Quadrature Method	1 (Si5351a)	00, 01, 10, 11	> 3.2 MHz

Table 5. A comparison of different local oscillator configurations and important characteristics.

When constructing the local oscillator of the SDR receiver, be aware of the voltages of different devices. For instance, the Si5351a is inherently a 3.3V device whereas an Arduino Nano is a 5V device. We need to be careful when connecting different voltage devices. We can connect devices with different logic levels by the use of a pull-up resistor to the lower voltage device. In the case of this example, we would have pull-up resistors to 3.3V as recommended in the Si5351a datasheet. However, in our experimentation, we found this method inconsistent. A safer approach would be to implement the use of a level shifter such as that used by Etherkit. The schematic by Etherkit is shown in **Figure 21[x]** where the pull up resistor and MOSFETS that are part of the level shifter are connected to the SCL and SDA pins of the Si5351a clock generator. More information about level shifters can be found at:

- https://etherkit.github.io/si5351abb_landing_page
- https://www.nxp.com/docs/en/application-note/AN10441.pdf
- https://www.digikey.com/product-detail/en/diodes-incorporated/2N7002DW-7-F/2N7002DW-FDICT-ND/750003

Figure 21 - Etherkit Schematic of the Si5351a with level shifter. [x]

Mixer Selection: Based on the discussion on mixers described in an earlier section, we recommend you use a Tayloe Mixer. An SN74CBT3253 or a similar IC can be used as the 4:1 multiplexer. The schematic for the Tayloe Mixer will typically take the form shown in **Figure 22[xi]**. The sampling capacitors, as part of the Tayloe Mixer configuration, can be selected based on the following equation:

$$C = \frac{1}{n \times \pi \times f_{ReceiverBandwidth} \times R}$$

R is the antenna impedance plus any other impedances in series before the sampling capacitor, n is the number of capacitors being charged, and $f_{ReceiverBandwidth}$ is selected by the designer. The sampling capacitors essentially become a simple RC low-pass filter when analyzed with the impedance, R.

Figure 22 - Tayloe Mixer Schematic with CLK 1 and CLK 2 as select line inputs to 4:1 multiplexer and sampling capacitors, C, on the 00, 11, 01, and 10 outputs of the multiplexer. The RF signal comes directly from the antenna to 1Y and 2Y. [xi]

Amplifiers and Low Pass Filter Selection: In order to design the amplifiers and LPFs properly a few things must be considered. Op-Amps can be expensive, so the model chosen has to balance cost and efficiency. The best way to do that is to compare the datasheets of the amplifiers to the specifications required. From the designs shown in this paper, a common IC that is used in Design 2 is the LT6321, chosen for its low noise performance and built-in model for LTspice simulations. The INA821 instrumentation amps used in Design 1 have the benefit of low noise, effectively infinite impedance on their inputs, and gain set by a single resistor, but are quite costly. The instrumentation amplifiers appear to be the simplest to analyze since they act purely as a voltage amplifier whereas the differential amplifiers act as trans-resistance amplifiers. The amount of gain on the amplifiers implemented post-mixer should strive to keep the signal to noise ratio within reason for the given application while also taking into account the sensitivity of the ADC of the soundcard to minimum voltages. Choosing an amplifier that has a pre-made or built-in spice model can be very useful for quality simulation. If there are no pre-made models for one you've selected, you can use the universal op amp model, and fill in the appropriate specifications for the one selected. The NE5532 and SA5532 are op amps with not quite the noise performance of the LT6231, but a much lower price. They can be had in the same package as the LT6321, so could be substituted for a less expensive receiver, especially for the lower bands using the same bands with the same PCB.

For low-pass filters, we utilized the online Filter Design Tool from Texas Instruments: https://webench.ti.com/filter-design-tool/filter-type. This saves time as you can get a complete filter design in minutes if you know the parameters needed. More in-depth discussion on the amplifiers and low-pass filters are presented earlier in this paper in the Design Comparison section.

CIRCUIT SIMULATION

Simulating a circuit before building the hardware, as mentioned briefly before, can save the designer many hours of troubleshooting. Often, simulation illuminates potential issues that weren't easily visible before. When an issue is identified in simulation, it is much easier to change the simulation component than it is to solder and de-solder a component from a physical board. In this section, we show several circuit examples with advice on how to simulate them properly. Images are included here, but the LTspice .asc files are available for download at http://fweb.wallawalla.edu/~frohro/ClassHandouts/?dir=Electronics/LTspice_simulations_of_QSD_SDR_Receivers.

LTspice Introduction: We primarily used LTspice for our circuit simulations. LTspice is a "high performance SPICE simulation software, schematic capture and waveform viewer with enhancements and models for easing the simulation of analog circuits." It can be downloaded for free at https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html and runs on Linux, OSX and Windows. LTspice also has many readily available help resources including help forums and instructional YouTube videos.

Circuit Simulation Examples: The portions of the SDR receiver circuit that are essential to simulate beforehand are shown with some examples here. Figures are paired with descriptions of what each circuit simulates and how to use appropriate *run* commands for each.

The bandpass filter hardware design can easily be simulated in LTspice using an AC analysis, though the design tools also give simulated results. **Figure 24** demonstrates the topology of a Bessel 3rd order bandpass filter generated through the RF tools website. When simulating the bandpass filter, it is important to include the input and output impedances. Here, the output impedance was assumed to be a 50Ω load as shown by R12. The input 50Ω impedance was included intrinsically in the voltage source V6. To implement an ac sweep, first select a small signal amplitude. For this example, the amplitude was set as 0.5mV. We selected a decade sweep
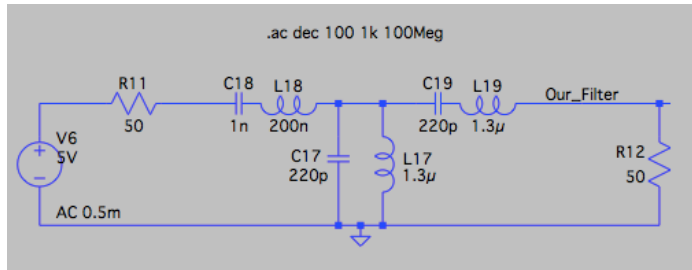
Figure24. LTspice Bandpass Filter Simulation
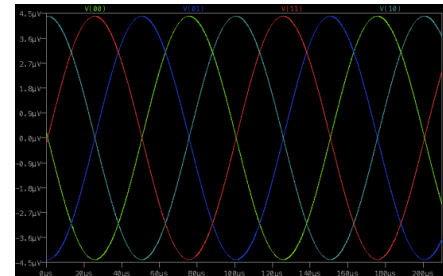
Figure 23. Bandpass Filter Simulation Results

with 100 points per decade spanning a frequency range from 1kHz to 100MHz. The results of running this simulation are shown in **Figure 23**.

Simulating the mixer in LTspice can potentially save hours of time in the long run. For this example, we will show you the simulation schematic for a Tayloe Mixer. Since LTspice is an analog circuit simulation tool, the 4:1 multiplexer used in the circuit must be illustrated by voltage-controlled switches. Recall that the Tayloe Mixer uses a local oscillator to drive the select pins of the FST5351 or similar multiplexer to "mix" the local oscillator frequency with the RF from the antenna. We use voltage-controlled switches and specific signals for the voltage sources to simulate this "mixing" phenomenon. There are two primary ways of specifying the signals in the voltage sources for the voltage-controlled switches.

The first way is by using the PULSE voltage source style. **Figure 26** shows the LTspice edit menu for inputting the settings for the pulse waveform. The pulse waveform is used to generate a square wave at 4 times the switching period to turn the voltage-controlled switch on when the wave is above a specified voltage. We use 4 separate switches controlled by 4 separate pulse waveforms.
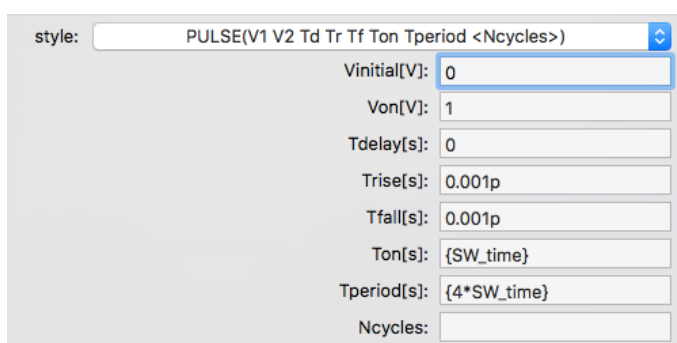

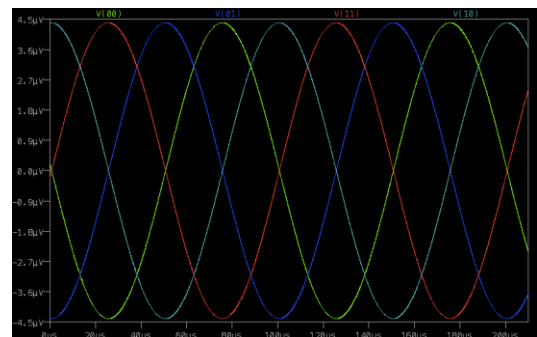*Figure 26. Settings for the PULSE wave 0-degree voltage control*

Figure 23. Output of the Tayloe Mixer Simulation with no load attached.

The only difference between each pulse wave is the delay time (*Tdelay[s]*). This delay is essential in generating 4 signals that are out each 90 degrees out of phase with each other.
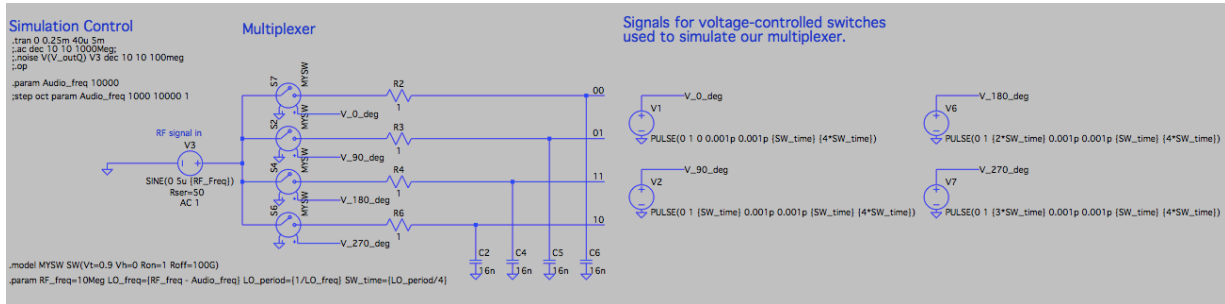

Figure 27. Tayloe mixer LTspice simulation utilizing voltage controlled switches with a PULSE wave control.

The second way is by using four voltage sources as sine waves that are phase shifted 90 degrees from each other. We phase shift them by editing the sine function, as shown in **Figure 29**, such that Φ is the degree (0, 90, 180, or 270), the frequency is 10kHz less than the RF signal phase, and the amplitude is at least 50V. Another important thing to note is that in order for this circuit to work as a multiplexer, we must change the switch model. The model must be adjusted such that Vt is the appropriate value, as shown in **Figure 28**. The rest of the circuit topology is the same as the previous illustration.
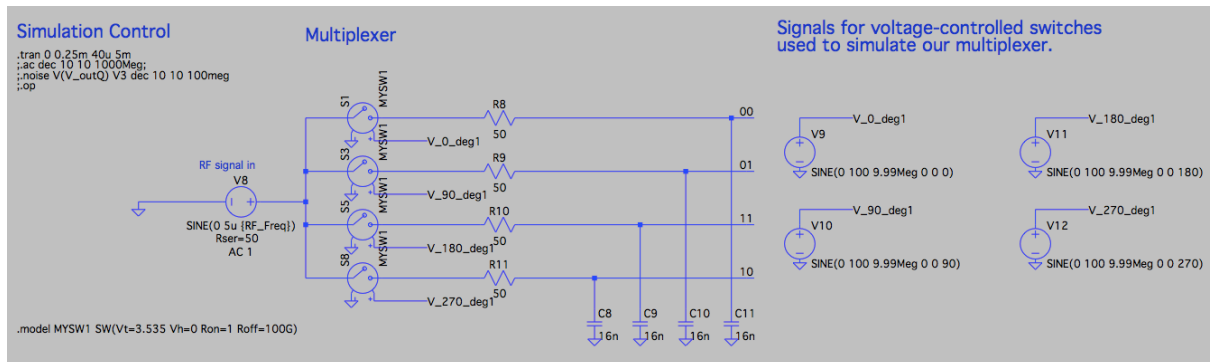

Figure 28. Tayloe Mixer LTspice simulation utilizing voltage controlled switches with SINE wave control

It is important to note that in order to have a simulation output that looks like **Figure 25**, we must

Figure 29. Settings for 90-degree SINE wave voltage output.

include the sampling capacitors. The sampling capacitors in this example are C2, C4, C5, and C6. We must also test this mixer without the amplifiers attached. When the amplifiers are attached to the outputs of the mixer, we see a loaded effect as a result of the characteristics of the op-amps, which may not be the result expected. This emphasizes the importance of testing the receiver in

compartmentalized sections rather than as a whole. In this way, we can verify each component individually making troubleshooting much easier and faster.

The first method of simulating the mixer is more intuitive and tends to operate more cleanly than the second. The second method offers an easier way of changing the local oscillator frequency. However, with some variables like the local oscillator frequency, period, and switch time coded as *.param* variables, it becomes just as easy to change the local oscillator values with the first method.

The last section of the circuit that is beneficial to simulate beforehand are the amplifiers and low pass filters with the final I and Q signals that will be entering the audio jack. It's important to remember that the voltage output from the Tayloe Mixer will not be the same as discussed previously due to the characteristics of the filters and amplifier chosen for the design. There are multiple amplifier topologies that can work for the radio. As mentioned in the design comparison section of this paper, the amplifiers have different characteristics that can be advantageous or disadvantageous in depending on the application. **Figures 30** and **31** illustrate two different amplifier circuits and how they can be simulated in LTspice as a starting point for your design. These represent only the I signal. A duplicate of this circuit can be used for the Q signal.
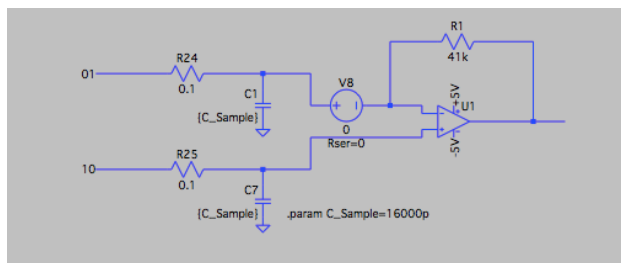

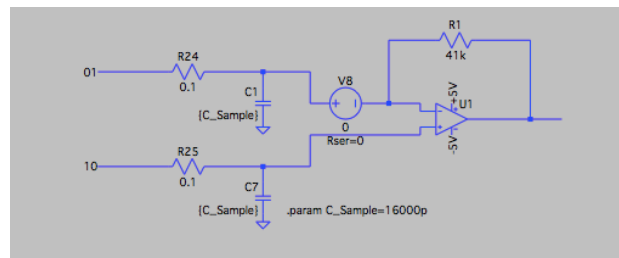*Figure 30. Instrumentation amplifier LTspice simulation topology*

Figure 31. Softrock amplifier LTspice simulation topology

BOARD LAYOUT AND BUILD
The layout of a PCB board can potentially have extreme effects on the performance of the circuit. A checklist of good design practices to keep in mind when designing the board follows:

- Include a prototype area
- Use sockets for components, such as capacitors or resistors, that will likely need adjustment.
- Include add-in areas with jumpers that will allow you to try multiple options on a single board
- Add test points everywhere! This makes testing and troubleshooting a lot easier.
- Strive to make the I and Q signal paths as similar as possible.
- Match component values and paths as closely as possible, as this can impact image rejection.
- Think about the types of connectors you will use on your board: audio jack, SMA, Arduino connector, etc.

- Contemplate the mechanical aspect of the design: how will it fit in a case? Is the display readable? Are the controls or test points difficult to reach?
- Anticipate which traces will be carrying large amounts of current and adjust for this with the trace sizes. There are multiple online tools to calculate trace impedance (and therefore size) for different current values, but we found the KiCad trace impedance calculator to be convenient.
- KiCad provides a bug check in the schematic capture and the board layout tools. Use them! The PCBnew bug check will look for unconnected traces and other violations. Ask a qualified peer to review your schematic especially, and your PCB. Utilize the board manufacturer's bug checker and their design rules.
- Label the PCB using the silkscreen with important information such as what it is and where the relevant documentation is located.
- Use the 3D viewer on PCBnew to ensure silk screen text is readable and placed correctly.
- Use a flood fill ground plane on both the front and back of the board to avoid ground loops, and to help bypass capacitors do their job.
- Try to keep the analog and digital components separate, because digital components cause analog noise.
- Refrain from having the input and output signals close to each other on the board, as it can lead to oscillation.

Board Bring-Up Plan: Before ordering a printed circuit board, or assembling the electronics, it is helpful to create a board bring-up plan. This is a guide to how one plans to assemble and test the different parts of the radio. By going through each sub-circuit or block as described early, one can troubleshoot sub-circuits and ensure that they work properly before moving on. Before applying power, check each sub-circuit and verify that there are no shorts between power and ground. It is not uncommon for parts to be damaged by electrostatic discharge, or accidents. This is one of several reasons to purchase extra components, especially the smaller package components as one lost or damaged component can set you back a week waiting for new parts.

As part of the board bring-up plan, include debugging strategies. Some debugging tips to keep in mind are:

- Verify jumpers are in the correct places.
- Carefully inspect the board with a magnifier for soldering problems.
- Use an ohmmeter to verify connections
- Check for oscillator signals, as we found sometimes you think they are there, but something happened in your last tests.
- Look for saturated op amps. Saturated op amps have their output voltages close to the power supply rails. They are not amplifying your signal if they are saturated.
- If more noise appears to be present than you like, try checking for ground loops.
  - The ground loop may be caused by the computer or soundcard attached to your board.
  - Experiment with running the receiver on batteries instead of through a laptop USB port

- o Try disconnecting your laptop from ground by running it on batteries instead of the charger connected to the wall.
- o Play around with the ground loop jumpers you included to prevent ground loops in anticipated problematic areas (if applicable).
- o Add transformers to isolate the antenna input and the audio output.

A good example of a board bring-up plan can be found here:

- https://github.com/froeca/Software-Defined-Radio/blob/master/Milestone_Submissions/Board%20Construction%20and%20Testing%20Plan.pdf

Ensure that the designer has an adequate test bench to assemble and troubleshoot the device. Ideally one should work in a well-lit space at a large desk with an anti-static mat. For the SDR radio we required a multimeter, signal generator, and an oscilloscope for a basic test bench. For an all-in-one solution, many students used a Digilent Analog Discovery 2, a device that plugs into a USB port and can perform the testing requirements for a project like this. The only drawback to this handy piece of equipment, is the limited frequency coverage it affords. This can be somewhat addressed by using harmonics of the signal generator for frequencies above 10 MHz. In addition, be sure to have a good soldering set up. A good kit may include the following.

- Soldering Iron (Wedge tip if possible)
- Solder
- Solder-Wick
- Flux
- Helping Hands
- Magnifying glasses
- Small Fume-Extractor Fan
- Wire Strippers

Remember, always turn off the soldering iron after use to prevent oxidizing the tip. For more soldering tips check here https://www.jameco.com/Jameco/workshop/techtip/soldering-tips.html

SOFTWARE

The software we used was Quisk, a Python program running on the PC, and Arduino code running on the SDR board. The details of installing and setting up the software will not be discussed directly in the paper. However, the following resources cover the software set up and bugs more in detail:

- http://james.ahlstrom.name/quisk/docs.html
- https://groups.io/g/n2adr-sdr/topics
- https://github.com/frohro/IQ_SDR/tree/master/Quisk/Arduino
- https://github.com/KonradMcClure/SDR_Receiver
- https://github.com/greenjacketgirl/SDR_Receiver/wiki/9.-Software
- https://github.com/threeme3/QCX-SSB

A great idea when testing the software is to write simple test programs. For instance, try writing an Arduino program that directly controls the Si5351a rather than one that communicates with Quisk to do so. An example of a simple test program is found in **Figure 32**.

When selecting the soundcard, be aware of the effect it will have on your radio. A sound card with a poor signal to noise ratio can degrade the quality of your receiver. Fortunately, nowadays there are inexpensive USB sounds cards with 24-bit ADCs and at least 90 dB of signal to noise ratio. Another characteristic to check on the sound cards is whether it is dual channel. It must have both a right and a left channel for there to be any image rejection.

```
#include "si5351.h"
#include "Wire.h"

Si5351 si5351;

void setup()
{
  bool i2c_found;
  i2c_found = si5351.init(SI5351_CRYSTAL_LOAD_8PF, 0, 0);
  si5351.set_freq(4*900000000ULL, SI5351_CLK0); // set local oscillator frequency
  Serial.begin(115200);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  digitalWrite(3, HIGH); // D3 (s0) select line for bandpass filter selection
  digitalWrite(4, LOW);  // D4 (s1) select line for bandpass filter selection
}


void loop()
{
  si5351.update_status();
  Serial.print("SYS_INIT: ");
  Serial.print(si5351.dev_status.SYS_INIT);
  Serial.print("  LOL_A: ");
  Serial.print(si5351.dev_status.LOL_A);
  Serial.print("  LOL_B: ");
  Serial.print(si5351.dev_status.LOL_B);
  Serial.print("  LOS: ");
  Serial.print(si5351.dev_status.LOS);
  Serial.print("  REVID: ");
  Serial.println(si5351.dev_status.REVID);
}
```

Figure 32. Arduino simple test program code to experiment with the local oscillator and bandpass filters. The local oscillator is set for 9MHz and multiplexer select lines to 01.

CONCLUSIONS

In this paper, we have discussed the theory behind quadrature signals, drew comparisons between four different self-assembled software-defined radio receivers, and addressed design tips for those interested in building their own SDR receiver. The interest Guido Ten Dolle's µSDX drew from the community illustrates the relevance of studying and presenting the knowledge we have on similar types of SDR receivers.

We broke down the theory of quadrature signals and mixers into an analogy of plane propeller rotations with visuals to illustrate the concepts. We hope this explanation will help extend understanding about this topic to a larger audience.

Through our design comparison analysis, we discovered that depending on the desired application of your receiver, different types of local oscillators may be selected. In addition, we noted the Tayloe Detector is the best selection for a mixer due to its low conversion loss and simplicity in design. The amplifiers can be difficult to analyze, even in a simulation program, due to some configurations acting as current or transresistance amplifiers rather than solely a voltage amplifier. The instrumentation amplifier presented itself as the simplest amplifier to analyze since it has infinite input impedance and acts as a voltage amplifier.

The design guide will be helpful to those unsure of where to start in the design process of an SDR receiver. We included a few issues that we encountered and possible solutions to them so that others can avoid the problems we had.

The details and instructions compiled in this paper are catered towards making the study and design of SDR receivers more accessible to a wide audience. With the information and time barrier lowered, we hope ham radio operators of a wider range of ages and expertise can discover the joys of designing and building a self-tailored software-defined radio with a thin wallet.

[i] "US Amateur Radio Population Grows Slightly in 2018." *ARRL: The National Association for Amateur Radio*, 14 Feb. 2019, www.arrl.org/news/us-amateur-radio-population-grows-slightly-in-2018.

[ii] Pepitone, Julianne. "The Uncertain Future of Ham Radio." *IEEE Spectrum: Technology, Engineering, and Science News*, 10 July 2020, spectrum.ieee.org/telecom/wireless/the-uncertain-future-of-ham-radio.

[iii] "Si5351a CLK0 and CLK1 Signals." *Groups.io*, 12 Aug. 2020, https://groups.io/g/ucx/topic/76162126#

[iv] Kebede, Zenaneh Ashebir. "Low frequency Quadrature detector design, simulation and implementation for use in underground communication." 2014. *Graduate Theses, Dissertations, and Problem Reports*. 219. https://researchrepository.wvu.edu/etd/219.

[v] Soer, Michiel. "Analysis and comparison of switch-based frequency converters." *University of Twente: Faculty of Electrical Engineering, Mathematics & Computer Science*, Sept. 2007, https://essay.utwente.nl/58276/1/scriptie_Soer.pdf.

[vi] Youngberg, Gerald. "A Software Defined Radio for the Masses, Part 4." *QEX,* Mar./Apr. 2003, http://www.arrl.org/files/file/Technology/tis/info/pdf/030304qex020.pdf.

[vii] "More on Ground Loops and Audio Settings - g4zfqradio." *Google Sites*, Aug. 2011, sites.google.com/site/g4zfqradio/more-on-ground-loops-and-audio-settings.

[viii] Kicad is an EE schematic design tool, https://kicad-pcb.org/

[ix] Notes, Electronics. "RF & Microwave Filters: the Basics." *Electronics Notes*, www.electronics-notes.com/articles/radio/rf-filters/understanding-rf-filters-basics-tutorial.php.

[x] "Si5351A Breakout Board." *Si5351A Breakout Board | Etherkit Documentation*, etherkit.github.io/si5351abb_landing_page.

[xi] Soer, Michiel. "Analysis and Comparison of Switch-Based Frequency Converters." *University of Twente, Faculty of Electrical Engineering, Mathematics & Computer Science*, 2007.