The What, How and Why of Complex Sampling for SDR Transceivers

Bruce E. Dunne
School of Engineering, Padnos College of Engineering and Computing
Grand Valley State University
Grand Rapids, MI 49504
Email: dunneb@gvsu.edu

Abstract

Software Defined Radio (SDR) transceivers are widely employed as powerful and low-cost platforms for classroom laboratory experimentation in the field of analog and digital communications. These SDR systems have also found application in student project work when significant signal processing horsepower is required for minimal cost. SDR systems typically utilize complex sampling, where instead of a single real-valued set of signal samples, a two-dimensional set of real and imaginary-valued samples are made available. Presenting the sampled data in this complex format has the benefit of allowing for unaliased signal processing right up to the sampling rate (instead of the usual Nyquist limit of half of the sampling rate). Hence, the constraints on the system sampling rate are somewhat lessened, and the requirements on the Analog to Digital Converter (ADC) are likewise lessened.

While available in most SDRs, complex sampling is often mysterious, even to those with significant signal processing background. Increasingly, instructors and students are using complex sampling in their SDR processing without an understanding of the consequences of the implied signal processing, with the potential for poor results. In this paper, the principle behind complex sampling and its benefits are derived, along with the associated cost. Next, the practical implementation of complex sampling for low-cost SDRs is described. A comparison of conventional sampling as compared to complex sampling is presented via a detailed example. An understanding of the difference will more clearly inform those instructors and students who choose to take advantage of the remarkable processing capabilities of SDR platforms.

Introduction

Software Defined Radio[1,2] (SDR) offers a powerful alternative to conventional communication system design. In conventional design, specially-built hardware is implemented to perform communication for a particular, radio-specific modulation scheme, usually over a limited frequency range. In contrast, SDR systems offer much more flexibility by implementing the modulation/demodulation functionality in software.

The literature has widely discussed and promoted the advantages of SDR for use in an academic laboratory setting for instructional purposes. Such papers present an overview of various experiments and projects[3,4,5,6,7] including discussions of both analog and digital communications laboratories, typically implemented via GNU Radio Companion[8] (GRC). GRC is the graphical user interface for GNU Radio, where users place functional blocks into a processing chain known as a flowgraph. Blocks exist for the vast majority of communication system functions, requiring users to simply configure the block with a handful of parameters particular to their system, compile and run.

One advantage to SDR processing is the ability to digitize and process large sections of the spectrum of interest at once. For example, consider an FM radio receiver. In a conventional, non-SDR approach, such an FM radio makes use of a superheterodyne architecture that precisely tunes to the channel of interest with a bandwidth equal to that of the FM signal. Conversely, in SDR, one approach might be to capture the entire FM spectrum (from 88 MHz to 108 MHz) and then selectively demodulate channels from that overall spectrum. Complex sampling lessens the requirements on the digitizing hardware such that this wide spectrum approach is more feasible and economical. Complex sampling (aka quadrature sampling) also finds application in digital demodulation since the signal naturally contains complex I/Q components.

Complex sampling is widely used for most SDR platforms (one recently found reference can be found here[9] that covers many of the ideas presented herein). Curiously, however, a precise description of what is meant by complex sampling as well as what the developer must consider when using such sampling is seemingly not well documented. It seems that instructors and students are employing SDR devices using complex sampling without a thorough understand of the consequences. Furthermore, the SDR manufacturers, while mentioning that their devices support complex sampling, seem to omit the fine details if the method as well. While arguably the theory behind complex sampling is not essential to its use, it is essential to at least understand the implications of using this sampling method. Hopefully, this paper will be a source of useful information to the SDR developer.

In this paper, an attempt is made to correct the shortcomings mentioned above and educate those who use SDR devices. Here, the method of complex sampling is explained by comparison of the capture of an example spectral band using conventional sampling (real-valued samples) versus complex sampling (complex-valued samples). Once understood, some properties of complex sampling that should be considered by any SDR developer are examined.

Goal: Capturing Spectrum

For illustrative purposes, let us suppose that we are trying to capture and digitize a portion of the RF spectrum and move it to baseband. A illustrative, simplified RF spectrum (magnitude) is shown below in Figure 1, with a center frequency of f_o Hz and lower and upper edge frequencies of f_1 and f_2 Hz, respectively (denote our desired signal $X(f)$ – positive frequencies shown; assume that the real-valued signal has a conjugate symmetric spectrum). Using the earlier example of FM radio, the lower edge frequency would be 88 MHz, the upper edge would be 108 MHz and the center frequency would then be 98 MHz.

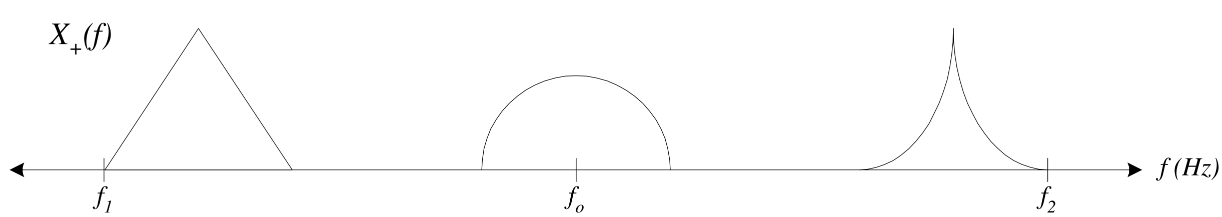

Figure 1: Example Spectrum for Digitization

To illustrate the complex sampling method, first we will explain the requirements to digitize the above spectrum using conventional real-valued sampling. We will then contrast that method with acquiring the digitized spectrum via complex sampling. Comparing the two methods will illustrate the advantages and costs of complex sampling.

Conventional Sampling

To digitize the spectrum of Figure 1, we first must translate it down to baseband. Note that for real-value signals, the spectrum will by conjugate symmetric; hence, we need to be careful such that "negative" spectral copies do not overwrite "positive" spectral copies when the spectrum is shifted up or down, respectively. Care must also be taken to ensure that out-of-band energy does not corrupt our resulting baseband signal.

To that end, a first step is careful bandpass filtering (BPF), particularly to remove any energy from DC to f_1 Hz as signals in this range would overlap the desired spectrum following frequency translation. Note that it is also possible to take a superheterodyne approach and apply the tight filtering at an IF frequency, but conceptually, the requirements are the same. Our BPF has a lower passband edge of f_1 Hz (tight) and an upper passband edge of f_2 Hz (not as tight).

Following filtering, the spectrum is frequency translated using down-conversion with an LO frequency of f_1 Hz. The resulting baseband portion of the spectrum is shown below in Figure 2 (here, the height of the spectral copies will be reduced by half, but this scaling is immaterial for our current discussion).

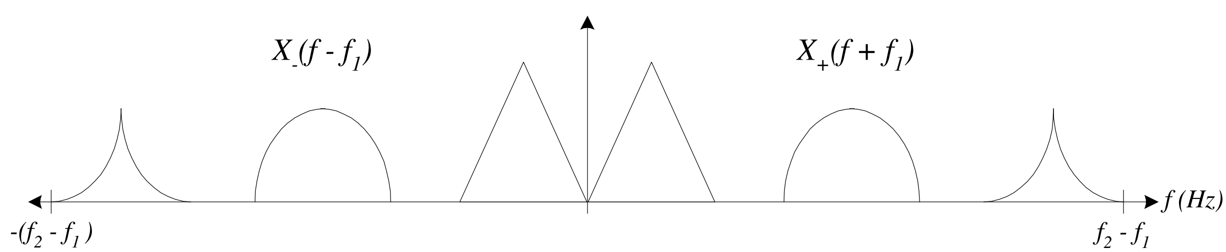

Figure 2: Down-converted Example Spectrum

Note that the resulting (magnitude) spectrum is even (the phase would likewise remain odd) and hence, our corresponding signal is real-valued.

To digitize the corresponding signal of Figure 2 without aliasing, we must choose a sampling frequency f_s larger than twice the highest frequency component of our signal, or

$$f_s > 2(f_2 - f_1) \text{ Hz.} \tag{1}$$

Prior to digitization, a lowpass filter (LPF) should be applied above $f_2 - f_1$ Hz to remove any potential for aliasing. Following digitization, and choosing f_s at its lowest (impractical) value, the resulting discrete spectrum $X_d(f)$ will look something like that as shown in Figure 3.

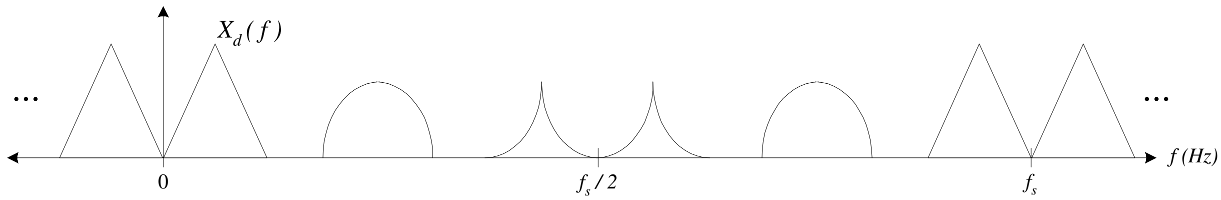

Figure 3: Discrete Spectrum using Conventional Sampling

For our practical example of digitizing the entire FM spectrum, we would require

- BPF from 88 MHz to 108 MHz;
- Down-conversion mixing with an LO frequency of 88 MHz;
- LPF with passband edge of 20 MHz;
- Digitize with a sampling frequency of at least 40 MSPS.

The resulting discrete spectrum is conjugate symmetric and repeats every f_s Hz. For someone well-versed in manipulating sampled signals in the frequency domain, Figure 3 presents no surprises.

Complex Sampling

In complex sampling, two down-converted representations of the RF spectrum are created. Since the overlap of lower frequency energy will effectively be removed algebraically in the formation of the digital signal, no BPF is required.

The first representation is created by down-converting using the in-phase or cosine function, this time at an LO frequency of f_o Hz (instead of f_1 Hz as done above conventionally). That is, we construct

$$x_I(t) = \mathrm{x}(t)\cos(2\pi f_o t). \qquad (2)$$

The resulting baseband spectrum is shown below in Figure 4 in two parts, being the result of downshifting and upshifting each of the positive and negative spectrum, respectively (where $f_x = f_2 - f_1$). Note that the images are “flipped” versions of each other.

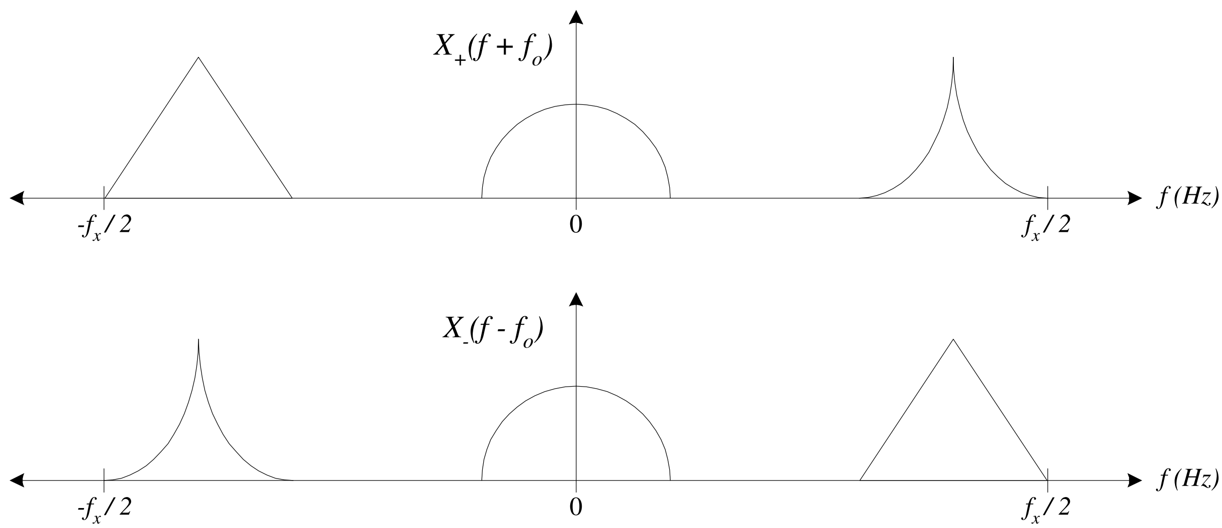

Figure 4: In-Phase Down-converted Frequency Components

Similarly, a second quadrature representation is created by down-converting using the sine function, also at an LO frequency of f_o Hz. Here, we construct

$$x_Q(t) = \mathrm{x}(t)\sin(2\pi f_o t) = x(t)\frac{1}{2j}\left\{e^{j2\pi f_o t} - e^{-j2\pi f_o t}\right\}. \quad (3)$$

Further, assume that we multiply the signal of (3) by $-j$. Clearly, what results in frequency is then

$$-j \times x_Q(t) \overset{FT}{\Leftrightarrow} \frac{1}{2}[X(f + f_o) - X(f - f_o)]\,. \quad (4)$$

The resulting baseband spectrum of (4) is shown below in Figure 5. Comparing Figure 4 and Figure 5, it would seem that simply adding these two spectra would result in the cancellation of the negative baseband spectrum from the original RF negative spectrum as well as the constructive addition of the positive baseband spectrum from the original RF positive spectrum. That is, if we form

$$x_{IQ}(t) = \mathrm{x}(t)\cos(2\pi f_o t) - j\,\mathrm{x}(t)\sin(2\pi f_o t), \quad (5)$$

the resulting spectrum is now expressed as

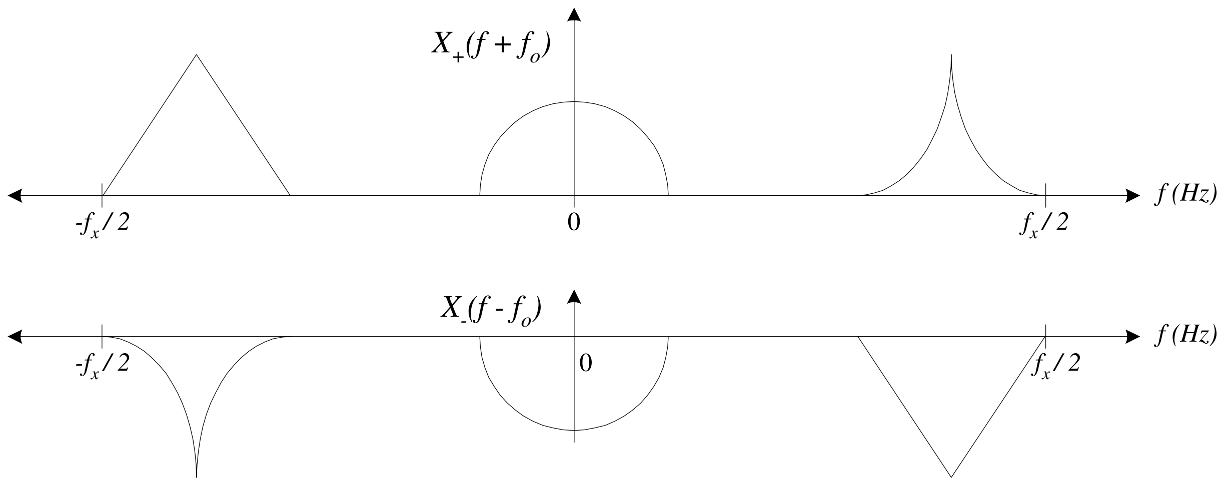

Figure 5: Quadrature Down-converted and Imaginary Scaled Frequency Components

$$X_{IQ}(f) = X(f + f_o), \tag{6}$$

with the associated baseband spectrum given in Figure 6. Comparison with Figure 1 shows that what we accomplished is similar to re-centering the positive frequencies around DC.

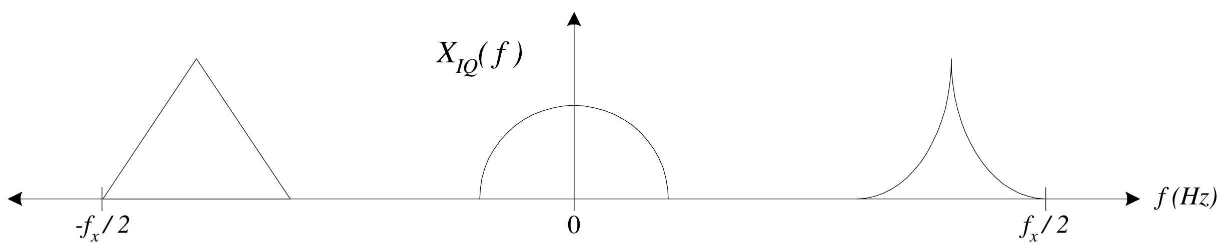

Figure 6: Resulting I/Q Baseband Spectrum

In practice, the multiplication by $-j$ in the time-domain can be tricky, since it corresponds to introducing a phase shift of exactly $-90°$ across a wide band of frequencies. However, since the intent is to sample the signal, there is an easier approach. Essentially, continuous time signals $x_I(t)$ and $x_Q(t)$ are formed according to (2) and (3), respectively. Each signal is subject to a LPF with a passband edge of $f_x/2 = (f_2 - f_1)/2$, followed by sampling at a rate of

$$f_s > (f_2 - f_1) \text{ Hz}. \tag{7}$$

Once each signal is digitized, the complex discrete signal

$$x_{IQ}[n] = x_I[n] - j\, x_Q[n] \tag{8}$$

is formed. Doing so will constructively add the discrete translated positive frequency spectrum while deleting the discrete translated negative frequency spectrum. With a sample rate equal to that of (7) (for illustrative purposes), the resulting discrete spectrum will similar to that shown in Figure 7.

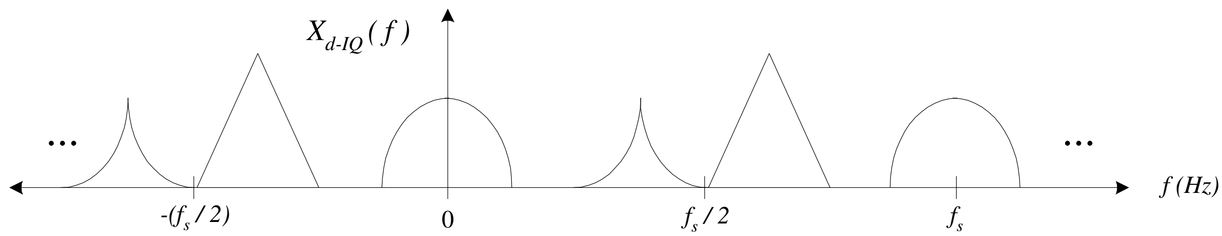

Figure 7: Discrete I/Q Spectrum using Complex Sampling

Comparison of Figure 7 with Figure 3 (and with Figure 1 as a point of reference) illustrates the difference between complex sampling and conventional sampling. Note the more efficient spectral representation without the redundant symmetry, and effectively, more efficient use of the capabilities of the ADC sampler. The same content was captured but at half the rate (albeit with two time-domain data streams).

For our practical example of digitizing the entire FM spectrum, we would require

- No BPF needed;
- Down-conversion mixing with an LO frequency of 98 MHz using both a cosine and sine carrier;
- LPF each of in-phase and quadrature signals with passband edge of 10 MHz;
- Digitize both in-phase and quadrature signals with a sampling frequency of at least 20 MSPS.
- Form a complex time signal in the discrete domain by scaling the quadrature component by $-j$ and adding it to the in-phase component.

The resulting discrete spectrum is not conjugate symmetric but it still repeats every f_s Hz.

Processing

Complex sampling has been shown to be an efficient method to capture a particular band of signals. The developer must be aware of the differences in the spectrum layout as compared to conventional sampling when processing signals. To highlight this point, consider the example of capturing the FM spectrum, and the desire to process a particular channel. A first step might be to isolate this channel with a BPF, but due to the fact that the spectrum is not symmetric, this filtering would include out-of-band spectral energy since the filter itself is usually symmetric. Instead, the approach should be to shift the desired channel to be centered about DC, and then applying a LPF to isolate the spectrum of interest. Note that this approach is doubly recommended: due to various design tradeoffs, the SDR spectrum often contains spurious content at DC. Isolating the signal of interest offset from DC removes the influence of this spurious content.

Often, someone new to working with SDR platforms may be confused by the complex sampling and simply use a GRC block to convert the signal format from complex to real-valued. Doing so is potentially disastrous as revealed by considering Figure 4. As can be envisioned from the figure, the resulting spectrum is now the overlapping sum of two versions of the spectrum where one is rotated about the center frequency in relation to the other. The spectral sum is effectively corrupted and useless, other than the portion centered about DC (assuming that it was symmetric to start with). Therefore, it is essential that the developer is aware of this issue before proceeding further.

Finally, complex sampling can actually be quite beneficial in some cases, beyond the obvious efficiencies. In many M-ary digital communication schemes, the signal is composed of in-phase and quadrature components, or equivalently, a phases and/or magnitude. Representing the signal with complex samples simplifies the processing necessary to work with these naturally complex-valued signals.

Conclusion

The theory of complex sampling was explained via a comparison with conventional sampling, using spectral plots to illustrate the main concepts. Essentially, to digitize the same spectral content, complex sampling halves the ADC rate as compared to conventional sampling at the cost of doubling the ADC operation and storage. Furthermore, complex sampling does not exhibit the same conjugate symmetry as obtain via conventional sampling, a point which requires consideration when processing signals. Given these observations, complex sampling naturally fits with SDR processing when wide sections of an RF band are to be collected (as opposed to a single channel).

A comparison between conventional and complex sampling is summarized below in Table 1.

Table 1. Comparison of Conventional Sampling to Complex Sampling

Parameter	**Conventional Sampling**	**Complex Sampling**
Frequency support	Unique frequency range $0 \leq f \leq f_s/2$	Unique frequency range $-f_s/2 \leq f \leq f_s/2$
Symmetry	Conjugate Symmetric	None
Minimum Sampling Frequency	Twice the signal bandwidth	One times the signal bandwidth
Time-domain data format	Real-valued	Complex-valued
Digitization	Single path	In-phase and quadrature paths
Analog Processing	Quality BPF and LPF	LPF only

Bibliography

1. Bard, J. & Kovarik, V., "*Software Defined Radio: The Software Communications Architecture*," Wiley Series in Software Radio, 2007.
2. Reed, J., "*Software Radio: A Modern Approach to Radio Engineering*," Prentice Hall, 2005.
3. Mao, S., & Huang, Y., & Li, Y. (2014, June), *On Developing a Software Defined Radio Laboratory Course for Undergraduate Wireless Engineering Curriculum* Paper presented at 2014 ASEE Annual Conference, Indianapolis, Indiana. https://peer.asee.org/22880

4. Wu, Z., & Wang, B., & Cheng, C., & Cao, D., & Yaseen, A. (2014, June), *Software Defined Radio Laboratory Platform for Enhancing Undergraduate Communication and Networking Curricula* Paper presented at 2014 ASEE Annual Conference, Indianapolis, Indiana. https://peer.asee.org/23023
5. Wyglinski, A. M., & Cullen, D. J. (2011, June), *Digital Communication Systems Education via Software-Defined Radio Experimentation* Paper presented at 2011 ASEE Annual Conference & Exposition, Vancouver, BC. https://peer.asee.org/17783
6. Cao, D., & Wu, Z., & Wang, B., & Cheng, C. (2018, June), *Undergraduate Research: Adaptation and Evaluation of Software-defined Radio-based Laboratories* Paper presented at 2018 ASEE Annual Conference & Exposition , Salt Lake City, Utah. https://peer.asee.org/31170
7. Zhang, Z., & Wu, Z., & Wang, B., & Cheng, C., & Cao, D. (2016, June), *Software Defined Radio-based General Modulation/Demodulation Platform for Enhancing Undergraduate Communication and Networking Curricula* Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. https://peer.asee.org/25833
8. URL: www.gnuradio.org last visited January 31, 2019.
9. Lyons, R.G., (2011) *Understanding Digital Signal Processing*, 3rd edition, Prentice Hall Pearson Education, Boston MA, Chapter 8, pp. 451-462.