1 Digital messages and pulse shaping

1.1. Square pulses

$m(t)$ is a digital message as shown in Figure 1.

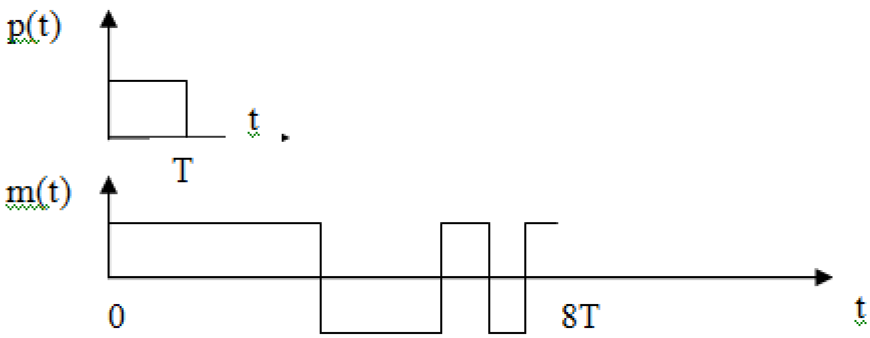


Figure 1 Digital Message

In Figure 1, $p(t)$ is chosen to be a so-called square pulse, where we can write $p(t)=1$ for $0<t<T$ and 0 otherwise . , where T is the symbol time. We send one such pulse for each data bit, a positive pulse $p(t)$ for logic 1 and a negative pulse for logic 0. In the figure, the message is shown in the interval $0<t<8T$ and the data bits are 11100101.

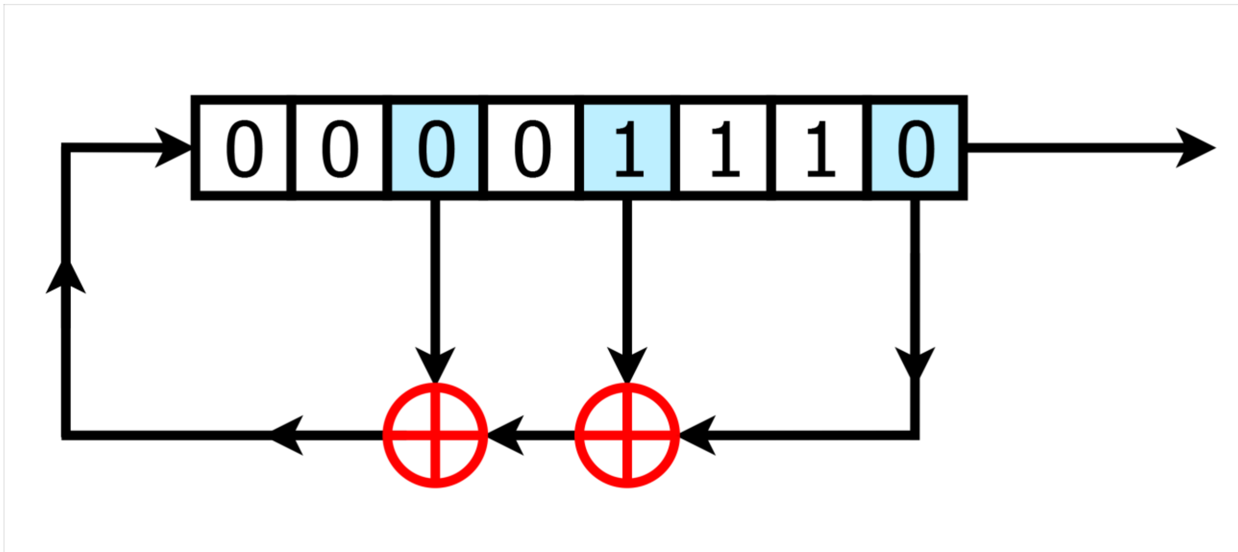
For the period $0<t<T$, $m(t)=A_0 p(t)=+1$ and the data is logic 1. For the next period $T<t<2T$, $m(t)=A_1 p(t-T)=+1$, and the data is logic 1. If $A_k=-1$, the data is logic 0. In the example figure above, the message is shown in the interval $0<t<8T$ and the information bits are 11100101

We can write $m(t)=A_0 p(t)+A_1 p(t-T)+A_2 p(t-2T)+\ldots=\sum_k A_k p(t-kT)$ where the real constants $A_k=\pm 1$ depending on whether a logic 1 or a logic 0 was sent.

In the figure above, $A_0=+1, A_1=+1, A_2=+1, A_3=-1, A_4=-1, A_5=+1, A_6=-1, A_7=+1$

1.2. Data source

The data source produces the data symbols $A_k=\pm 1$ that may be stored in memory or generated on the fly. For testing purposes, it is convenient to have a random data source, with an equal number of logic 1s and logic 0s, or an equal number of positive and negative pulses. This can be achieved with a so-called linear feedback shift register.



At each time step (symbol time) T, the input on the left is determined by the XOR gate output and all the bits slide along one step and the output is the bit on the right. If the feedback shift register is of length M and the feedback taps (XOR gates) are selected correctly, the output sequence will have length $2^M - 1$ before it repeats.

1.3. Implementation in sampled system

For a sampling frequency f_s and symbol time T, there are $f_s T$ samples per symbol. Each square pulse contains $f_s T$ samples of identical value.

The data symbols $A_k = \pm 1$ are represented by one sample per symbol and come from the data source at a rate $1/T$ symbols per second. To generate the square pulse, each sample of data must be repeated $f_s T$ times, so that the square pulse is represented by $f_s T$ samples running at sampling frequency f_s to make up a symbol that takes time T to transmit. We also refer to the symbol time as the symbol length. A sequence of such symbols will look like the waveform shown in the figure above with $f_s T$ samples per symbol.

1.4. Pulse shaping to eliminate the sharp edges of a square pulse

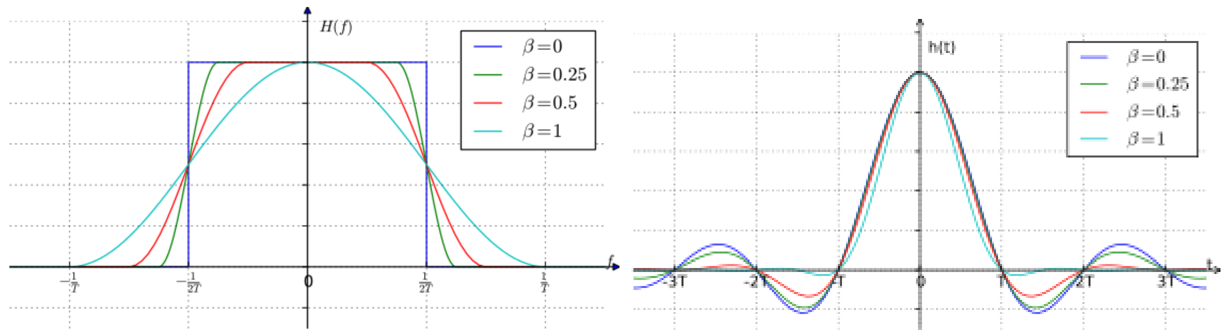
The square pulse can be low pass filtered with a cutoff frequency of $1/(2T)$ since this is the highest frequency in the data stream for an alternating data sequence 101010… $A_k = 1, k \text{ odd}, A_k = -1, k \text{ even}$

1.5. Representing data sequence with Impulses

The data sequence can be represented by the sequence of impulses $m_\delta(t) = A_0\delta(t) + A_1\delta(t-T) + A_2\delta(t-2T) + \ldots = \sum_k A_k \delta(t-kT)$ convolved with a pulse shaping filter with impulse response $p(t)$ and frequency response $P(f)$. To represent this sequence of impulses $m_\delta(t)$ as a waveform sampled at f_s, the impulses must be filtered by an interpolating FIR filter that yields one sample $A_k = \pm 1$ followed by $f_s T - 1$ samples set to zero. This waveform $m_\delta(t)$ is then in turn convolved with an FIR filter with impulse response $p(t)$ to yield the message waveform $m(t)$ The filter coefficients of the FIR filter will be the pulse shape $p(t)$ sampled at f_s. $p(t)$ can be a square pulse $p(t) = 1 \text{ for } 0 < t < T \text{ and } 0 \text{ otherwise}$, but can also be a longer pulse, for example a so-called raised cosine pulse $p(t) = 0.5(1-\cos(\pi t / T), 0 < t < 2T, 0 \text{ otherwise}$ Note that the raised cosine pulse extends over 2 symbol periods, so that adjacent symbol pulses will overlap.

1.6. Raised Cosine filter

$p(t)$ can be chosen to obtain a particular frequency response $P(f) = H(f)$. One example choice is a raised cosine shape in the frequency domain, as per the figure on the left



The impulse response is a sinc shape extending for a time usually truncated to $6T$ as shown in the figure on the right.

1.7. Raised Root Cosine filter

A receiver is optimum if the receiver filter matches the transmit filter. Thus we often use a Root Raised Cosine (RRC) pulse shaping filter at the transmitter and another RRC filter at the receiver, so that the combination of the two filters yields a Raised Cosine (RC) pulse shape at the receiver output. The RRC filter impulse response looks similar to the RC impulse response but with lower sidelobes.

1.8. Summary

Pulse shaping can be done in multiple ways. Here we consider three methods:

1. Generate square pulses and low pass filter, commonly used in analog systems.
2. Generate impulses and low pass filter with time domain RC FIR filter
3. Generate impulses and filter with frequency domain RRC filter

2 BPSK Transmitter

A general signal

$$s(t) = \mathrm{Re}[a(t)e^{j\phi}e^{j2\pi f_c t}] = \mathrm{Re}[\tilde{s}(t)e^{j2\pi f_c t}],$$

where the complex envelope

$$\tilde{s}(t) = a(t)e^{j\phi} = a(t)\cos\phi + ja(t)\sin\phi$$
$$= i(t) + jq(t)$$

and $i(t), q(t)$ are obtained from the message signal.

Recall that for DSB-SC, we choose $i(t) = m(t)$ and $q(t) = 0$, so that

$$s(t) = \mathrm{Re}\{\tilde{s}(t)e^{j2\pi f_c t}\} = \mathrm{Re}\{[i(t) + jq(t)]e^{j2\pi f_c t}\}$$
$$= \mathrm{Re}\{m(t)e^{j2\pi f_c t}\} = m(t)\cos 2\pi f_c t$$

For DSB-SC $\tilde{s}(t) = m(t)$, i.e. the complex envelope is equal to the (real) message.

For binary phase shift keying (BPSK), we also choose $m(t) = i(t)$ and $q(t) = 0$ and we choose

$$m(t) = A_0 p(t) + A_1 p(t-T) + A_2 p(t-2T) + ... = \sum_k A_k p(t-kT)$$

The BPSK signal is written

$$s(t) = \mathrm{Re}[m(t)e^{j2\pi f_c t}] = m(t)\cos 2\pi f_c t = \sum_k A_k p(t-kT)\cos 2\pi f_c t$$

The signal constellation for BPSK consists of two points located at $A_k = \pm 1$, one at +1 and one at -1, as shown in Figure 2.

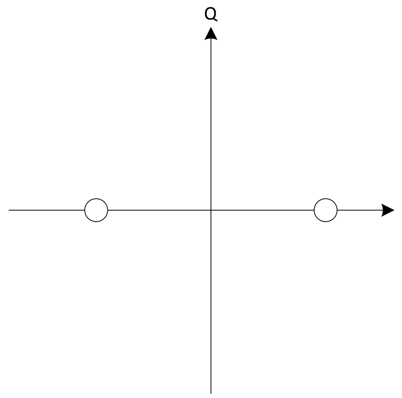


Figure 2 BPSK Constellation

For BPSK, $\tilde{s}(t) = m(t)$, i.e. the complex envelope is equal to the message.

2.1 Real BPSK signal

To create a real BPSK signal in GRC at intermediate frequency f_1 using these equations, we:

1. Generate the real baseband signal $i(t) = m(t)$
2. Upconvert it to the desired carrier frequency by multiplying by

$\cos 2\pi f_1 t = 0.5\{e^{j2\pi f_c t} + e^{-j2\pi f_c t}\}$ to yield $s(t) = \mathrm{Re}[m(t)e^{j2\pi f_1 t}] = m(t)\cos 2\pi f_1 t$

The spectrum of $s(t)$ will be centered at $\pm f_1$ and has both positive and negative frequency components.

2.2 Complex BPSK signal

To create a complex (analytic) BPSK signal in GRC at intermediate frequency f_1 using these equations, we:

1. Generate the real baseband signal $i(t)$
2. Upconvert it to the desired carrier frequency by multiplying by $e^{j2\pi f_1 t}$. The results is a complex (analytic) signal .

$$\begin{aligned}
&\tilde{s}_1(t) = i(t)e^{j2\pi f_1 t} \\
&= i(t)\cos 2\pi f_1 t + j[i(t) sin 2\pi f_1 t] \\
&= i_1(t) + jq_1(t)
\end{aligned}$$

The spectrum of $\tilde{s}_1(t)$ will be centered at f_1 . Since the signal is analytic, there is no negative frequency component at $-f_1$.

2.3 BPSK at radio frequency (RF)

The USRP sink block with complex input $\tilde{s}_1(t)$ is used to transmit the BPSK signal near a radio frequency (RF) f_c. The USRP transmitter multiplies $\tilde{s}_1(t)$ by $e^{j2\pi f_c t}$ and takes the real part to generate a real RF signal centered at $f_1 + f_c$ and $-f_1 - f_c$.

$$
\begin{aligned}
s(t) &= \mathrm{Re}\{\tilde{s}(t)e^{j2\pi f_c t}\} = \mathrm{Re}\{[i_1(t) + jq_1(t)]e^{j2\pi f_c t}\} \\
&= i_1(t)\cos 2\pi f_c t - q_1(t)\sin 2\pi f_c t \\
&= i(t)\cos 2\pi f_1 t \cos 2\pi f_c t - i(t)\sin 2\pi f_1 t \sin 2\pi f_c t \\
&= i(t)\cos 2\pi(f_c + f_1)t
\end{aligned}.
$$

We can get the same result using the exponentials instead of the sine and cosines

$$
\begin{aligned}
s(t) &= \mathrm{Re}\{\tilde{s}(t)e^{j2\pi f_c t}\} = \mathrm{Re}\{i(t)e^{j2\pi f_1 t}e^{j2\pi f_c t}\} \\
&= \mathrm{Re}\{i(t)e^{j2\pi(f_1+f_c)t}\} = i(t)\cos 2\pi(f_c + f_1)t
\end{aligned}
$$

The real signal $s(t)$ is centered at $f_1 + f_c$ and $-f_1 - f_c$. Since the signal is real, there are both positive and negative frequency components.

The USRP sink block could also have used the complex input $\tilde{s}(t) = i(t) = m(t)$ from section 2.2.

with the imaginary part set to zero to transmit the BPSK signal at f_c In this case the USRP transmitter multiplies $\tilde{s}(t)$ by $e^{j2\pi f_c t}$ and takes the real part to generate a real RF signal centered at f_c and $-f_c$. However, this practice is not recommended, since $\tilde{s}(t) = m(t)$ may contain DC, and the USRP does not pass DC.

3 BPSK Receiver

To receive the BPSK signal, the USRP source block is used to downconvert the RF signal to complex baseband $i(t) + jq(t)$

We can also simulate the BPSK receiver within GNU Radio without using the USRP source block.

The simulation is done by using the BPSK signal generated at intermediate frequency f_1

3.1 Real BPSK signal

We first work with the real BPSK signal $\tilde{s}(t) = \mathrm{Re}[m(t)e^{j2\pi f_1 t}] = m(t)\cos 2\pi f_1 t$

We use a standard IQ receiver set up for f_1 with complex output $i(t) + jq(t)$

If there is no frequency or phase offset, then $q(t) = 0$

The IQ receiver can be implemented with real cosines and sines and low pass filters.

With a suitable low pass filter, the received data $i(t)+jq(t)$ at sampling times $t=kT+\epsilon$ is $A_k=\pm 1$ (real), where ϵ is adjusted to sample in the middle of the pulse $p(t)$.

ϵ is a timing offset to sample the data away from the transitions between different bits.

3.2 Complex BPSK signal

We can also work with the complex BPSK signal

$$\tilde{s}_1(t)=i(t)e^{j2\pi f_1 t}$$
$$=i(t)\cos 2\pi f_1 t + j[i(t)sin2\pi f_1 t]$$
$$=i_1(t)+jq_1(t)$$

Again, we use a standard IQ receiver. In this case, we can use the complex IQ receiver that multiplies $\tilde{s}_1(t)$ by $e^{-j2\pi f_1 t}$ (note the minus sign for downconversion).

4 QPSK Transmitter

Recall that we can send separate messages $m_1(t), m_2(t)$ on the I and Q channels, i.e.

$i(t)=m_1(t), q(t)=m_2(t)$

We can choose $m_1(t)$ and $m_2(t)$ to be separate digital messages with the same kind of waveform as shown in Section 1 above. When we use both the I and Q channels, the modulation is called Quadrature Phase Shift Keying (QPSK).

To write an expression for QPSK, we can use the same general form as for BPSK above, except that the constants representing the data are now complex instead of real.

A digital message can be written as a complex envelope

$$\tilde{s}(t)=\sum_k C_k p(t-kT)=C_0 p(t)+C_1 p(t-T)+C_2 p(t-2T)+.$$

where

$\tilde{s}(t)=i(t)+jq(t)$ is a complex message waveform, $C_k=A_k+jB_k=a_k e^{j\phi_k}$ is the digital data, where $A_k=a_k\cos\phi_k$ and $B_k=a_k\sin\phi_k$

From the expression for $\tilde{s}(t)$, we see that for the period $0<t<T$, $\tilde{s}(t)=A_0+jB_0=a_0 e^{j\phi_0}$ is a complex constant. For the next period $T<t<2T$, $\tilde{s}(t)=A_1+jB_1=a_1 e^{j\phi_1}$, etc. In general, for the time period $kT<t<(k+1)T$ $\tilde{s}(t)=A_k+jB_k=a_k e^{j\phi_k}$ is a fixed amplitude and phase.

$$i(t) = \sum_k A_k p(t - kT) = A_0 p(t) + A_1 p(t-T) + A_2 p(t-2T) + .$$

and

$$q(t) = \sum_k B_k p(t - kT) = B_0 p(t) + B_1 p(t-T) + B_2 p(t-2T) + .$$

For QPSK, we choose $A_k = \pm 1$ and $B_k = \pm 1$, so that $C_k = \pm 1 \pm j$.

There are 4 possible values of C_k depending on whether a logic 00, 01, 10 or 11 was sent.

The signal constellation for QPSK is shown in Figure 3 and consists of the 4 points located at $C_k = \pm 1 \pm j$

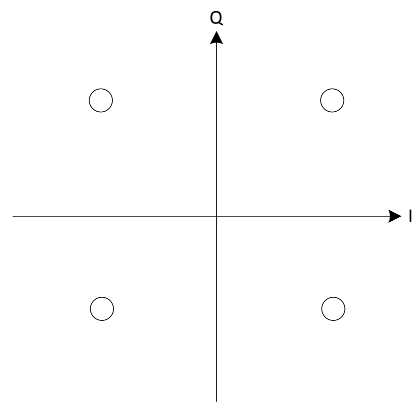


Figure 3 QPSK Constellation

In polar form $C_k = A_k + jB_k = a_k e^{j\phi_k}$, so that with $A_k = \pm 1$ and $B_k = \pm 1$, $a_k = \sqrt{2}$ and $\phi_k = \dfrac{(2n-1)\pi}{4}$ are the 4 possible phases of a QPSK signal.

4.1 Real QPSK signal

To create a real QPSK signal in GRC at intermediate frequency f_2 using these equations, we (1) generate the complex baseband signal $\tilde{s}(t) = i(t) + jq(t)$ and (2) upconvert it to the desired carrier frequency with a standard IQ transmitter to yield $s(t) = i(t)\cos 2\pi f_2 t - q(t) sin 2\pi f_2 t$

The spectrum of $s(t)$ will be centered at $\pm f_1$ and has both positive and negative frequency components.

4.2 Complex QPSK signal

To create an complex QPSK signal in GRC at intermediate frequency f_2 using these equations, we (1) create the complex baseband signal $\tilde{s}(t) = i(t) + jq(t)$ and (2) upconvert it to the desired carrier frequency by multiplying by $e^{j2\pi f_2 t}$. The results is another complex signal

$$\tilde{s}_2(t) = [i(t) + jq(t)]e^{j2\pi f_2 t}$$
$$= i(t)\cos 2\pi f_2 t - q(t)\sin 2\pi f_2 t + j[q(t)\cos 2\pi f_2 t + i(t)\sin 2\pi f_2 t]$$
$$= i_2(t) + jq_2(t)$$

We can create a second QPSK signal at a different intermediate frequency f_3 in the same way to yield

$$\tilde{s}_3(t) = i_3(t) + jq_3(t)$$

We can create a composite signal consisting of several QPSK signals at f_2, f_3 etc.

4.3 QPSK at radio frequency (RF)

The upconversion of a QPSK signal to a radio frequency (RF) wave at f_c is the function of the USRP sink block (standard IQ transmitter) with inputs $i_2(t)$ and $q_2(t)$ .The USRP transmitter accepts a complex baseband signal $\tilde{s}_2(t)$ and multiplies it by $e^{j2\pi f_c t}$ and takes the real part to generate a real RF signal $s(t)$.

The entire group of QPSK signals $\tilde{s}_2(t) + \tilde{s}_3(t)$ can be upconverted to RF by the USRP sink block.

The USRP sink block could also have used the complex input $\tilde{s}(t) = i(t) + jq(t)$ from section 3.1.

to transmit the QPSK signal at f_c In this case The USRP transmitter multiplies $\tilde{s}(t)$ by $e^{j2\pi f_c t}$ and takes the real part to generate a real RF signal centered at f_c and $-f_c$. However, this practice is not recommended, since $\tilde{s}(t) = i(t) + jq(t)$ may contain DC, and the USRP does not pass DC.

5 QPSK Receiver

To receive the QPSK signal, the USRP source block is used to downconvert the RF signal to complex baseband $i(t) + jq(t)$

We can also simulate the QPSK receiver within GNU Radio without using the USRP source block.

The simulation is done by using the QPSK signal generated at intermediate frequency f_2

5.1 Real QPSK signal

We first work with the real QPSK signal

$$s(t) = i(t)\cos 2\pi f_2 t - q(t) sin 2\pi f_2 t$$

We use a standard IQ receiver set up for f_2 with complex output $i(t) + jq(t)$

This IQ receiver can be implemented with real cosines and sines and low pass filters.

With a suitable low pass filter, the received data $i(t)$ at sampling times $t = kT + \epsilon$ is $A_k = \pm 1$ and the received data $q(t)$ is $B_k = \pm 1$, where ϵ is adjusted to sample in the middle of the pulse $p(t)$

In the complex domain, the received data $i(t) + jq(t)$ at sampling times $t = kT + \epsilon$ is $C_k = \pm 1 \pm j$

5.2 Complex QPSK signal

We can also work with the complex QPSK signal

$$
\begin{aligned}
&\tilde{s}_2(t) = [i(t) + jq(t)]e^{j2\pi f_2 t} \\
&= i(t)\cos 2\pi f_2 t - q(t)\sin 2\pi f_2 t + j[q(t)\cos 2\pi f_2 t + i(t)\sin 2\pi f_2 t] \\
&= i_2(t) + jq_2(t)
\end{aligned}
$$

Again, we use a standard IQ receiver. In this case, we can use the complex IQ receiver that multiplies $\tilde{s}_2(t)$ by $e^{-j2\pi f_2 t}$ (note the minus sign for downconversion). The received data $i(t) + jq(t)$ at sampling times $t = kT + \epsilon$ is $C_k = \pm 1 \pm j$

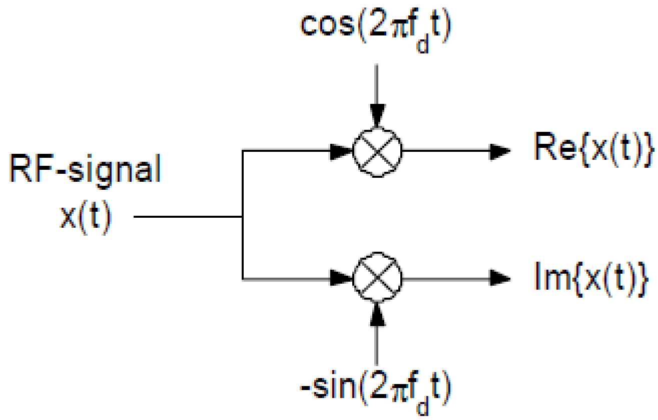
5.3 QPSK signal at radio frequency (RF)

To receive a QPSK signal at RF, we use the USRP source block. The USRP receiver multiplies the real valued radio frequency signal $s(t)$ by $e^{j2\pi f_d t}$ to generate $i(t) + jq(t)$. This is called *complex downmixing* and is equivalent to the standard IQ receiver, where in the figure below $i(t) = \mathrm{Re}\{x(t)\}$ and $q(t) = \mathrm{Im}\, x(t)$ and f_d is the frequency of the USRP source block.

The USRP source block has a complex output with real and imaginary components $i(t)$ and $q(t)$,

Thus the USRP source block with frequency set to f_c will have outputs

$i(t) = a(t)\cos\phi,\ q(t) = a(t)\sin\phi.$



The complex signal output from the USRP source block $i(t)+jq(t)$ is bandlimited to the sampling rate of the USRP source block. The USRP source block output can be recorded to a file and used again at a later time. This file source will have the same sampling rate and bandwidth as the USRP source block used to record it.

With a sampling rate of 200 kHz and complex samples, the bandwidth will be 200 kHz (because the complex signal spectrum is not symmetric and does not have redundant mirror-image positive and negative frequencies)

We can "tune in" (receive) any of the QPSK signals in this bandwidth by shifting the spectrum of the USRP source output. We shift by e.g f_2 Hz by multiplying the USRP source output (complex signal) $i(t)+jq(t)$ by $e^{-j2\pi n f_2 t} = \cos 2\pi f_2 t - j \sin 2\pi f_2 t$, so that the signal that first appeared at f_2 Hz now appears at zero Hz. With a suitable low pass filter, the received data $i(t)+jq(t)$ at sampling times $t = kT + \epsilon$ is $C_k = \pm 1 \pm j$, where ϵ is adjusted to sample in the middle of the pulse $p(t)$