Chapter 7

BPSK

In this chapter, the PCI-5640R IF transceiver is used to implement a packet-based digital modulator/demodulator (modem) using *binary phase shift keying* (BPSK). The laboratory employs the matched filter used in Chapter 6 and introduces *pulse shaping*.

In the following exercises, you will implement a BPSK modem. The PCI-5640R IF transceiver will provide wireless transmission and reception; Matlab Script Node is recommended for baseband processing in the exercises.

7.1 Background

7.1.1 Fractional-Rate Baseband Processing

A digital implementation of fractional-rate baseband processing is shown in Figure 7.1. A message sequence $a[n]$ of symbol values (possibly complex-valued) is generated at the symbol rate, $1/T$, upsampled to the fractional sampling rate, P/T, and converted to a baseband waveform via the pulse shaping filter, $g[k]$. Upsampling is merely the process of inserting $P-1$ zeros between each consecutive pair of input samples. The resulting sampling rate of $a_{\uparrow}[k]$ is therefore $1/T_s = P/T$ samples per second, with P as the number of samples per symbol. The output of the convolution of $a_{\uparrow}[k]$ with $g[k]$ is the message signal, $\tilde{m}[k]$, which is then upconverted using the quadrature amplitude modulation and digital-to-analog conversion seen in Chapters 1 and 3.

At the receiver, digital downconversion is followed by a receiver filter, $q[k]$, whose output is downsampled by P to produce the symbol-rate output, $y[n]$. Downsampling is merely the process of keeping only every Pth sample. The goal of the processing chain –and the subject for this chapter – is for $y[n]$ to replicate $a[n]$.

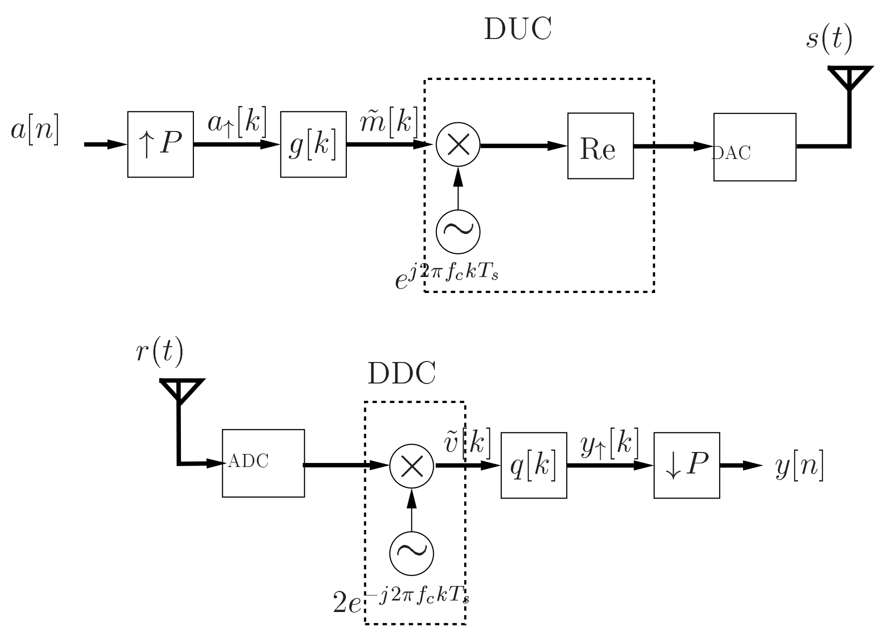

Figure 7.1: Fractionally sampled baseband processing and digital IF upconversion. Top: Transmitter. Bottom: Receiver.

7.1.2 Pulse Shaping and Intersymbol Interference

Nyquist pulse is ISI-free

For a noiseless, ideal channel and perfect synchronization of phase, frequency, and sample timing at the receiver, we have received samples given by

$$y[n] = \sum_{m} a[m]p(n - mT) \tag{7.1}$$

where $p[k] = g[k] * q[k]$ is the combined effect of transmitter pulse shaping, $g[k]$, and receiver filtering, $q[k]$. Thus, each transmitted symbol $a[m]$ contributes to the received samples across the duration of the pulse, $p[k]$. To ensure that $y[n]$ equals $a[n]$ and thus faithfully reproduces symbols at the receiver, we must prevent a symbol from interfering with other symbols. Thus, the prevention of *intersymbol interference* (ISI) requires

$$p[nT] = \begin{cases} 1, & n = 0 \\ 0 & n \neq 0 \end{cases} . \tag{7.2}$$

This condition is known as the *Nyquist criterion* and is equivalent to the frequency domain condition

$$\frac{1}{T} \sum_{k=-\infty}^{\infty} P\left(f - \frac{k}{T}\right) = 1, \tag{7.3}$$

which is illustrated in Figure 7.2. In other words, the superposition of the frequency response shifted every integer multiple of the symbol rate must sum to T.

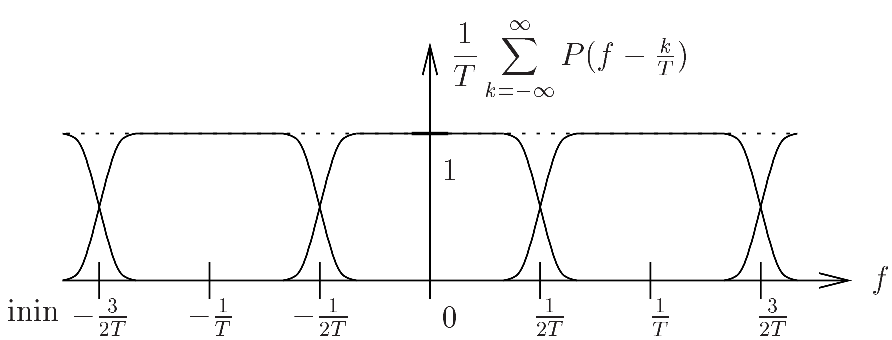

Figure 7.2: Illustration of the frequency-domain expression of the Nyquist pulse criterion.

The raised cosine pulse is a Nyquist pulse, is shown in Figure 7.3, and is given by

$$p_{\mathrm{RC}}(t) = \frac{\cos(\pi\alpha t/T)}{1-(2\alpha t/T)^2}\,\mathrm{sinc}(t/T), \quad \mathrm{sinc}(x) := \frac{\sin(\pi x)}{\pi x}$$

$$P_{\mathrm{RC}}(f) = \begin{cases} T & |f| \leq \frac{(1-\alpha)}{2T} \\ T\cos^2\left(\frac{\pi T}{2\alpha}\left(|f| - \frac{1-\alpha}{2T}\right)\right) & \frac{(1-\alpha)}{2T} \leq |f| \leq \frac{(1+\alpha)}{2T} \\ 0 & \frac{(1+\alpha)}{2T} \leq |f| \end{cases} . \qquad (7.4)$$

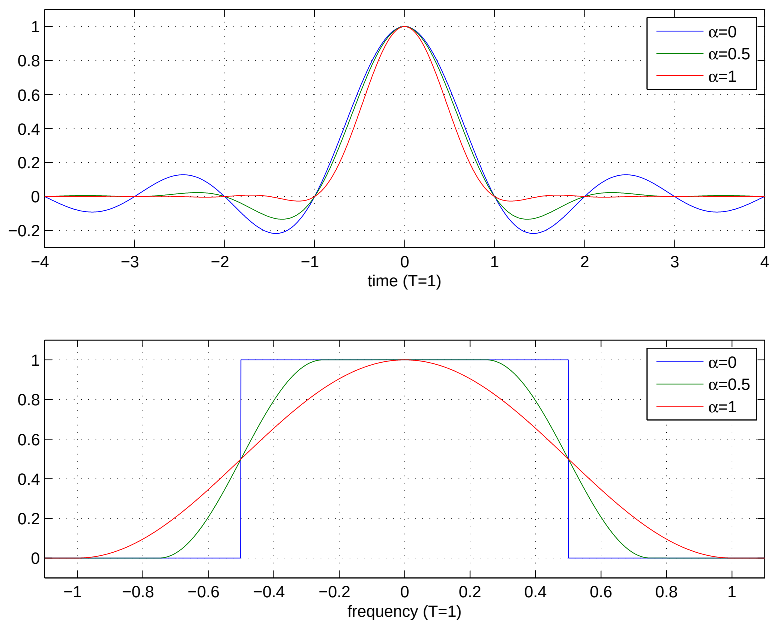

Figure 7.3: Raised cosine pulse provides a Nyquist pulse; the pulse is shown for three values of excess bandwidth, α.

Matched filter

Thus, to be free from intersymbol interference, we have $G(f)Q(f) = P(f)$, where $P(f)$ satisfies the Nyquist criterion. And, the raised cosine, $P_{\mathrm{RC}}(f)$ is one such pulse function. To maximize signal-to-noise ratio at the receiver output, we further have the matched filtering criterion, $Q(f) = G^*(f)$, as seen in Chapter 6. Thus, taking these two design criteria jointly, we can choose the pulsing shaping filter, $G(f)$, to be the square-root of the raised cosine,

$$G(f) = \sqrt{P_{\text{RC}}(f)} \tag{7.5}$$

where $Q(f) = G(f)$ satisfies the matched filtering criterion because $P_{\text{RC}}(f)$ is real-valued and positive. This choice for the pulse shaping filter is called the *square-root raised cosine* (SRRC) pulse and is given by

$$g_{\text{SRRC}}(t) = \frac{(1-\alpha)\,\text{sinc}(\frac{t}{T}(1-\alpha))}{1-(4\alpha\frac{t}{T})^2} + \frac{4\alpha\cos(\pi\frac{t}{T}(1+\alpha))}{\pi(1-(4\alpha\frac{t}{T})^2)}. \tag{7.6}$$

Observe that $q(t) = g^*_{\text{SRRC}}(-t) = g_{\text{SRRC}}(t)$ because $g_{\text{SRRC}}(t)$ is real-valued and symmetric about $t = 0$.

For implementation, the MATLAB command `g=srrc(D,alpha,P)` creates a sampled-data SRRC pulse truncated to ±`D` symbol intervals in length, with `P` samples per symbol and excess bandwidth parameter `alpha`. Examples are shown in Figure 7.4. Note that the truncation implies that the resulting filter only approximately satisfies the Nyquist criterion. Larger `D` and larger `alpha` create a more exact approximation; lower values degrade the approximation, causing ISI at the receiver.

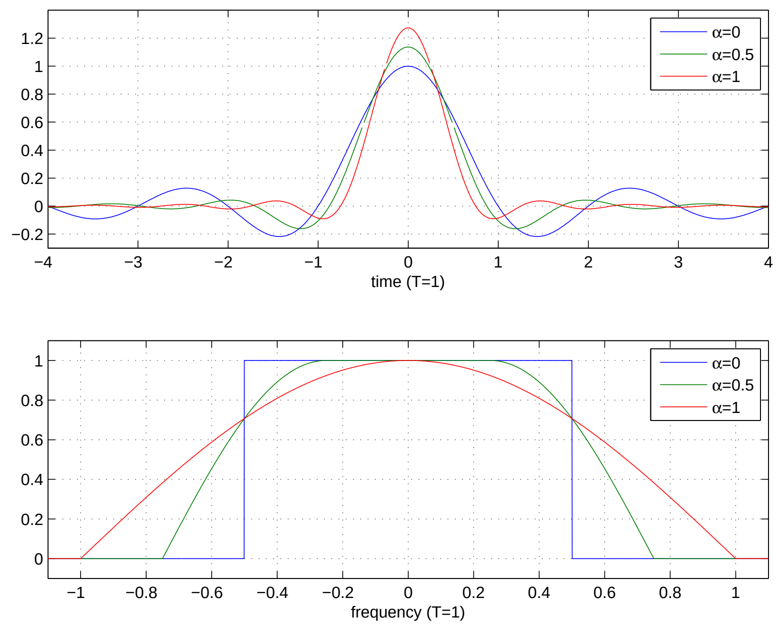

Figure 7.4: Square-root raised cosine filter; the filter impulse response is shown for $D = 4$ and three values of excess bandwidth, α.

7.2 Exercise: Pulse Shaping

In this exercise, you will generate and use the square-root raised cosine pulse.

1. Download the MATLAB programs `srrc.m` and `plottf.m` from the course webpage.

2. For $P = 9$, $D = 3$, and $\alpha = 0.8$, create and plot the SRRC pulse shaping filter and its autocorrelation using `g = srrc(D,alpha,P)` and `xcorr`. What is the length of the filter impulse response? For each plot, identify the zero crossings. If the autocorrelation is in the variable `p`, then the following overlay highlights the approximation to the Nyquist criterion:

```
%overlay plot with red circles at sampling times
figure;plot(p);
[pmax,Imax] = max(abs(p));%find peak and its index
hold on; t = Imax+ [-2*D*P : P : 2*D*P];%downsampling times
plot(t,p(t),'ro');grid on;%plot
```

Question 7.1: Pulse shape.

3. Repeat for $P = 9$, $D = 2$, and $\alpha = 0.05$. Record your observations and offer an explanation for the differences from the previously observed pulse shape.

Question 7.2: Distorted pulse shape.

7.3 Exercise: Implement BSPK modem

1. Download the MATLAB program `eyeplot.m` from the course webpage.

2. Copy the subdirectory

```
C:\508 New templates\TxRxPacket
```

to your personal workspace. Open and read the `Readme` file contained in that folder.

3. Following Figure 7.1, write MATLAB code for the MATLAB SCRIPT NODE at the transmitter.

   (a) Use an upsampling and downsampling factor of $P = 9$. The commands `upsample` and `downsample` are convenient (although directly indexing using the colon syntax `(1 : P : end)` to count indices by steps of `P` works well, too).

   (b) For the pulse shape, use a square-root raised cosine, which may be designed using `srrc(D,alpha,P)`. Use $P = 9$, $D = 3$, and $\alpha = 0.8$.

   (c) For the message signal, use a length 63 Kasami sequence as a preamble, followed by a BPSK data sequence of length 200. The BPSK alphabet is $\{1, -1\}$, resulting in one bit per symbol. (The name comes from two phases, 0 and 180°, of the symbols.)

   (d) In LabVIEW set `IQrate` to 1.5625M and the carrier frequency to 38 MHz.

4. Following Figure 7.1, write MATLAB code for the MATLAB SCRIPT NODE at the receiver.

   (a) Employ a matched filter.

   (b) Process the matched filter output, $y_{\uparrow}[k]$, to locate the Kasami sequence preamble, thereby recovering the packet timing at the receiver. Use correlation as explored in Chapter 6.

   (c) Recall that the digital upconversion and digital downconversion are not phase synchronous. Therefore, use the correlation peak not only for timing recovery but also for phase recovery. Adjust the phase of $y_{\uparrow}[k]$.

   (d) Use downsampling factor of $P = 9$ to obtain $y[m]$.

   (e) In LabVIEW set `IQrate` to 1.5625M and the carrier frequency to 38 MHz.

5. Use the phase-adjusted $y_{\uparrow}[k]$ and the script `eyeplot.m` to create an eye diagram for your received BPSK message signal.

Question 7.3: Tx/Rx baseband code. Report your MATLAB code as an appendix in your lab report.

7.4 Exercise: ISI

Use your BPSK modem to explore the effects of intersymbol interference. Use RF cables and antennas to transmit and receive data packets.

1. Plot the eye diagram for your received packet and compare the data bits to the received bits.

Question 7.4: Eye diagram and error count.

2. Introduce error to the estimated timing recovery by adding K samples to the estimate start time. For $K = 3$, plot the eye diagram and compare the data bits to the received bits. Mark your observations on the eye diagram and explain the observed behavior. Repeat for a higher value of K that results in a high bit error rate.

Question 7.5: Timing errors.

3. Introduce distortion by changing D and α in your square-root raised cosine filter. Use the same values at both transmitter and receiver. For the following three settings, plot the autocorrelation of the srrc pulse and plot the eye diagram for the received data.

Setting	P	D	α
#1	9	2	0.05
#2	9	2	0.8
#3	9	3	0.01

Record your observations and offer an explanation.

Question 7.6: Pulse shape distortion.