LAB 8

Frequency-Shift Keying

Prerequisites: Lab 6 – Frequency Modulation, Lab 7 – Amplitude-Shift Keying

8.1 Objective

In frequency-shift keying (FSK), a 1 is represented by a tone at a specific frequency, known traditionally as the "mark" frequency, while a 0 is represented by a tone at a different frequency, known as the "space" frequency. FSK owes part of its popularity to the fact that a tone is always being transmitted, even when the source generates a long string of zeros. This makes it easy for the receiver to distinguish between a transmitter that is idling and a transmitter that has stopped transmitting. FSK also has the property that the transmitted signal has a constant amplitude. This allows a very efficient nonlinear power amplifier to be used for transmission, a very important consideration when the transmitter is battery-powered.

FSK is the digital version of frequency modulation. Just as an ASK system is built around an AM transmitter and receiver, we will see that an FSK system is built around an FM transmitter and receiver. The additional features such as symbol mapping, pulse shaping, matched filtering, threshold detection, and pulse synchronization all apply to FSK as they do to ASK.

8.2 Background

Transmitter

The bandwidth of the transmitted signal is an important property of any modulation method. Bandwidth has two components, the "main lobe" bandwidth determines the channel width needed to carry the transmitted signal, and also the bandwidth of the filter in the receiver front end. The rate of spectral rolloff determines the interference that a signal will cause to signals in adjacent channels. This rolloff determines how close in frequency similarly modulated signals can be placed. The bandwidth of an FSK signal is notoriously difficult to calculate analytically. J.R. Carson, writing in the 1920's, provided a rule of thumb for approximating the bandwidth:

$$B_{FSK} \cong 2\left(\Delta f + B\right), \tag{1}$$

where B_{FSK} is the bandwidth of the FSK signal, Δf is the peak frequency deviation of the signal (see below), and B is the message bandwidth. Carson's rule is simple to apply, but it tends to provide a slight overestimate when applied to FSK signals.

The rolloff rate of an FSK signal is largely governed by the smoothness of the signal at the moments when the frequency changes from mark to space or vice versa. If the FSK signal is allowed to become discontinuous at these moments, the rolloff in the power spectrum will be proportional to $1/f^2$, which is comparable to ASK using rectangular pulses. *Continuous-phase* FSK has a spectral rolloff of $1/f^4$. Consequently, most modern applications only use the continuous-phase version. Even more rapid spectral rolloff rates can be achieved by smoothing the message signal before applying it to the FSK transmitter.

The generation of an analog FM signal is discussed at length in the background section of *Lab 6: Frequency Modulation*. It is strongly suggested that this material be reviewed at this point. Recall that given a message $m(t)$, the instantaneous frequency of an FM signal is defined as

$$f(t) = f_c + k_f m(t), \tag{2}$$

where f_c is the carrier frequency and k_f is the frequency sensitivity. The peak frequency deviation of the FSK signal is given by

$$\Delta f = k_f m_p, \tag{3}$$

where m_p is the peak value of the message. To form a continuous-phase FSK signal, the message signal $m(t)$ is a binary message represented in a polar, non-return-to-zero (NRZ) format, as shown in Figure 1.

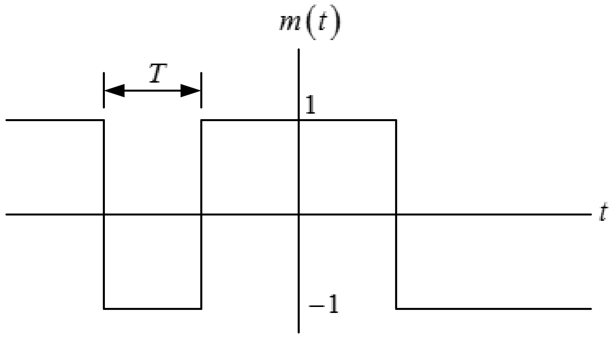

Figure 1. Binary Message Waveform

The pulse duration T that appears in Figure 1 will be called the "symbol time," just as was the case in the ASK lab project. The peak value of the message is $m_p = 1$. Given the polar NRZ message format, we see from Eq. (2) that the mark frequency is $f_c + \Delta f$ and the space frequency is $f_c - \Delta f$. It is important to note, referring to Eq. (1), that the bandwidth of the FSK signal is not $2\Delta f$, but includes a contribution from the message bandwidth. Thus the bandwidth cannot be made arbitrarily small by reducing the peak frequency deviation.

As is shown in Lab 6, the signal that must be sent to the USRP to produce an FM output is

$$\tilde{g}(t) = A_c e^{j2\pi\Delta f \int_0^t m(\alpha)d\alpha}, \tag{4}$$

where A_c is the carrier amplitude. The USRP will produce the FM signal given by

$$g(t) = \mathrm{Re}\left[\tilde{g}(t)e^{j2\pi f_c t}\right] = A_c \cos\left[2\pi f_c t + 2\pi\Delta f \int_0^t m(\alpha)\,d\alpha\right]. \tag{5}$$

It should be noted that the phase of the FM signal given by Eq. (5) is proportional to $\int_0^t m(\alpha)\,d\alpha$. For the polar NRZ message given in Figure 1, this integral yields a continuous function of time.

The steps needed, in addition to frequency modulation, to form an FSK signal should be familiar if you have completed the ASK lab project:

1. <u>Symbol Mapping</u>. The input data arrives as a stream of bits. Recall that the *MT Generate Bits* function produces an array of bytes containing the numbers 1 and 0. In the symbol mapping step, the bits are replaced by numerical values. For FSK we will represent a binary 1 by the real double 1 and a binary 0 by the real double -1. Table 1 shows the FSK symbol mapping.

Table 1 FSK Symbol Mapping

Bit Value	Symbol
0	-1
1	1

2. Upsampling. As a first step toward replacing symbols by pulses, we will place $L-1$ zeros after each symbol. This produces a sample interval of

$$T_x = \frac{T}{L}, \tag{6}$$

or a sample rate of

$$\frac{1}{T_x} = L\frac{1}{T}. \tag{7}$$

A higher upsampling factor L makes the D/A conversion in the transmitter easier, but requires faster digital processing. We will use $L = 40$ in this lab project.

3. Pulse Shaping. If the upsampled signal is applied to a filter whose impulse response $g_{TX}[n]$ is a rectangular pulse of unit amplitude and length L samples, then at the filter output, each symbol will be represented by a rectangular pulse. Steps 1 through 3 convert the input bit stream to a polar NRZ message signal as shown in Figure 1. Note that pulse shapes other than rectangular can be used simply by changing the shape of the impulse response $g_{TX}[n]$.

4. Modulation. Once we have the message signal $m(t)$, Eq. (4) is applied to produce the baseband signal to be sent to the USRP transmitter.

Receiver

An FSK receiver begins with an FM demodulator. First, the phase is extracted from the complex baseband signal received from the USRP. Next, the phase is differentiated and filtered. These steps are described in more detail in the background section of Lab 6. To complete the digital receiver, several additional steps follow the FM demodulator:

1. Pulse Synchronization. The FM receiver output is an analog baseband signal that must be sampled once per symbol time, i.e. once every T seconds. Because of filtering, propagation delays, and distortion caused by the communication channel, it is necessary to determine the optimum time to take these samples. A function called *PulseAlign(real)* has been provided in the *BasicUSRPLabs folder* to align the baseband signal so that the sample at index 0 is the correct first sample.

2. Sampling. The *Decimate* function will sample the aligned baseband waveform at index 0 and every T seconds thereafter.

3. Detection. Once the baseband waveform has been sampled, each sample must be examined to determine whether it represents a symbol of value 1 or a symbol of value 0.

4. Symbol Mapping. The detected symbol values must be converted to bits. For FSK, this step is easily included in the detection step.

8.3 Pre-Lab

Transmitter

1. Create a program to generate a continuous-phase FSK signal using the USRP. A template for the transmitter has been provided in the file FSKTxTemplate.gvi. This template contains the four functions for interfacing with the USRP along with *MT Generate Bits* from the Modulation Toolkit. *MT Generate Bits* will create a pseudorandom sequence of bits that can serve as a data sequence for testing your FSK system. Note that by default, *MT Generate Bits* will produce the same sequence of bits every time you run the program. This is useful for debugging, but if you would like to generate a different sequence of bits every time, wire a random number to the "seed in" input. The steps below contain details about how to create the required transmitter.

2. First do the symbol mapping, as shown in Table 1.

3. Upsample the array of symbols using *Upsample* from the Analysis→Signal Processing→Conditioning subpalette. In this lab project you are given control inputs to set the symbol rate $1/T$ and the IQ rate $1/T_x$. Set the symbol rate to 10,000 symbols/s and the IQ rate to 400×10^3 Sa/s. Use these two inputs to calculate the upsampling factor L.

4. Use *MT Generate Filter Coefficients* from the Modulation Toolkit to generate the pulse shaping filter. (*MT Generate Filter Coefficients* can be found on the Analysis→ Communications→Digital→Utilities subpalette.) Set the modulation type to FSK, and the

pulse-shaping filter "samples per symbol" to your calculated value of L. Create a front-panel control for "pulse shaping filter" and set this initially to "none." As in the ASK lab project, the setting of "none" will generate rectangular pulses. (Remember: "none" does not mean that there is no filter.) Wire the "pulse shaping filter coefficients" output to the "Y" input of a *Convolution*. The *Convolution* is available from the Analysis→Signal Processing→Operation subpalette. Wire the output from your upsampler to the "X" input of the *Convolution*.

5. Normalize the amplitude of your filtered message signal to a maximum absolute value of 1. This step will be important later when we investigate alternative pulse shapes. The *Quick Scale 1D* in the ExternalFiles folder is ideal for this task. (You may use the $Y[i] = X[i]/\mathrm{Max}|X|$ output of *Quick Scale 1D*, since the "X" input is real-valued in this lab project.) Connect the output of *Quick Scale* to the *Build Waveform* function that connects to the Message Signal graph provided in the template.

6. Now implement Eq. (4). To implement the integral use an IIR Filter from the External Files folder. Use a "forward coefficients" array of $[1]$ and a "reverse coefficients" array of $[1 \quad -1]$. Multiply the integrated message by $2\pi\Delta f$ and also by dt. The peak frequency deviation Δf is available from a front panel control; set the deviation initially to 5000 Hz. The sample interval dt is the reciprocal of the "actual IQ rate" (dt and T_x refer to the same quantity). Use a *polar to complex* function to create the complex baseband signal. Let the carrier amplitude A_c be 1. The complex baseband signal connects to the "data" input of *Write Tx Data*.

7. To observe the spectrum of the transmitted signal, wire the complex baseband signal to the *Build Waveform* function that connects to the Baseband Power Spectrum graph provided in the template.

This completes construction of the FSK transmitter. Save your transmitter in a file whose name includes the letters "FSKTx" and your initials (e.g. *FSKTx_BAB.gvi*).

Receiver

1. Create a program to implement an FSK receiver using the USRP. A template for the receiver has been provided in the file *FSKRxTemplate.gvi*. This template contains the six interface functions for interfacing with the USRP.

   Calculate the "number of samples" for the *Fetch Rx Data* to fetch using the message length and symbol rate front panel inputs. Double the number the number of samples in a frame so that the receiver will fetch two frames of data. Since the receiver's starting point is random, this ensures that there will be one complete frame of received data.

   Pass the complex array returned by the *Fetch Rx Data* function through a *Complex to Polar* function to extract the phase. Unwrap the phase using *Unwrap Phase* (Analysis→Signal Processing→Conditioning subpalette) to remove jumps of 2π before taking the derivative. To differentiate the phase, use an *FIR Filter*, with FIR Coefficients set to the array $\begin{bmatrix}1 & -1\end{bmatrix}$. Differentiating the phase will create large spikes wherever there is a phase discontinuity. These occur at the beginning of the signal, and wherever the transmitter begins its transmitted sequence over again. To smooth out the spikes, use a *Median Filter* (Analysis→Signal Processing→Filters→Special Filters). Set the "left rank" input to 5 and leave the "right rank" unwired.

2. To implement the receiver's matched filter, use *MT Generate Filter Coefficients* just as you did for the transmitter. Set the modulation type to FSK, and calculate the "matched samples per symbol" M from the "actual IQ rate" ($1/T_z$) and the symbol rate ($1/T$) obtained from the front-panel control. Create a front-panel control for "pulse shaping filter" and set this initially to "none." (Remember: the setting of "none" will generate a matched filter with a rectangular impulse response.) Wire the "matched filter coefficients" output to the "Y" input of a *Convolution*. The output of your matched filter should be connected to the *Cluster Properties function* provided in the template. The *Cluster Properties function* feeds the Baseband Output graph.

3. Place the *PulseAlign(real)* on your block diagram and wire the baseband output waveform to the "input waveform" input and wire the M samples/symbol to the "receiver sampling factor" input.

Once the baseband waveform is aligned, it can be sampled. *Decimate (single shot)* can be obtained from the Analysis→Signal Processing→Conditioning subpalette. The "decimating factor" is M.

4. The Modulation Toolkit function *MT Format Eye Diagram* has been provided in the receiver template. Wire the baseband output waveform to the "waveform" eye-diagram input. The "symbol rate (Hz)" input value is available from the front panel control. Set the "eye length" parameter to 2.

5. To determine whether each received sample is more likely to represent a 1 or a 0, the sample must be compared with a threshold. Because the message $m(t)$ is a polar signal, the threshold can be taken as zero. The result of the comparison is the receiver's digital output.

This completes construction of the FSK receiver. Save your receiver in a file whose name includes the letters "FSKRx" and your initials (e.g. *FSKRx_BAB.gvi*).

Questions

1. In *Transmitter* step 3 you are given $1/T = 10{,}000 \text{ symbols/s}$ and $1/T_x = 400 \times 10^3 \text{ Sa/s}$. Find the corresponding value for the number of samples per symbol L.

2. Explain how an IIR filter having a forward coefficients array of $[1]$ and a reverse coefficients array of $[1 \quad -1]$ implements an integrator.

3. Explain how an FIR filter having a coefficients array of $[1 \quad -1]$ implements a differentiator, as used in *Receiver* step 1.

4. Explain what a median filter does. Refer to the LabVIEW 2014 online help for the *Median Filter* function for information.

5. In *Receiver* Step 2 the "actual IQ rate" $1/T_z$ may be different than the rate $1/T_x$ that was used at the transmitter. (Note that the symbol rate $1/T$ must be the same at the transmitter and receiver.) The value of the receiver's IQ rate determines the receiver sampling factor M . What is the advantage to using a higher value of M ? What is the advantage of using a lower value of M ?

6. The FM receiver uses a differentiator to undo the integral in Eq. (4). What is the effect of the differentiator on any noise that might be present along with the signal? To answer this, consider what a differentiator does in the frequency domain. What would be the effect of omitting the filter that follows the differentiator?

8.4 Lab Procedure

1. Connect a loopback cable and attenuator between the TX 1 and RX 2 connectors of the USRP. Connect the USRP to your computer and plug in the power to the USRP. Run LabVIEW and open the transmitter that you created in the prelab.

2. Ensure that the transmitter is set up to use

   Carrier Frequency: 915.0 MHz

   IQ Rate: 400 kHz. Note: This sets the value of $1/T_x$.

   Gain: 0 dB.

   Active Antenna: TX1

   Symbol rate: 10,000 symbols/s

   Message Length: 1000 bits

   Pulse shaping filter: None

   Peak frequency deviation: 5000 Hz

   Run the transmitter. Use the large STOP button on the front panel to stop transmission connectors.

3. After running the transmitter, observe the spectrum of the transmitted signal. The two significant features are the bandwidth and the rate of spectral rolloff. Measure the null-to-null bandwidth of the transmitted signal. Compare the measured bandwidth to the bandwidth predicted using Carson's rule, Eq. (1). The rate of spectral rolloff is a measure of the interference that your signal will cause to signals using nearby carrier frequencies. Set the vertical scale of your spectrum plot to the range -100 dB to 0 dB, and print a copy of the spectrum for comparison in Step 4 below.

4. The rate of spectral rolloff of an FSK signal is determined primarily by the smoothness of the transmitted signal. Continuous-phase FSK has no discontinuities when the frequency of the transmitted signal changes, but there can be "corners" where the slope of the transmitted signal changes abruptly. To smooth out these corners, the message $m(t)$ can be filtered before it is passed to the FM modulator. Change the pulse-shaping filter to "Gaussian" to create a very smooth pulse transition. Run the transmitter and observe the smoothed message signal and the power spectrum. Measure the bandwidth of the transmitted signal 60 dB below the spectral peak. Set the vertical scale of your spectrum plot to the range -100 dB to 0 dB and print a copy of the spectrum. Compare the rolloff rate with that of the spectrum you obtained using rectangular pulses.

5. Change the pulse-shaping filter back to "none." Set the peak frequency deviation to 20,000 Hz, 10,000 Hz, 5000 Hz, and 2500 Hz. In each case, measure the null-to-null bandwidth of the transmitted signal and compare to the values predicted by Carson's rule.

   For the given symbol rate, 2500 Hz is the "minimum" frequency deviation. Below this deviation, performance will suffer, as the mark and space signals are not sufficiently different.

6. Ensure that the receiver is set up to use

   Carrier Frequency: 915.0 MHz

   IQ Rate: 400 kHz. Note: This sets the value of $1/T_z$.

   Gain: 0 dB

   Active Antenna: RX2

   Symbol rate: 10,000 symbols/s

   Message Length: 1000 bits

Pulse shaping filter: None

7. Run the transmitter, then run the receiver. Once the receiver has acquired a block of data, you may stop the transmitter.

8. Use the horizontal zoom feature on the Baseband Output graph palette to expand the demodulated waveform so that you can see individual pulses. Ideally, rectangular pulses passed through the receiver's matched filter should produce triangular output pulses. Note whether the demodulated pulses have the expected shape. (You can also right click a graph and choose Capture Data then analyze the data from the Data pane)

9. Observe the eye diagram. Make note of the optimum sampling time and the presence of intersymbol interference. To see the effect of pulse synchronization, move the waveform input of *MT Format Eye Diagram* to the "aligned waveform" output of *Pulse_align(real)*. Run the transmitter and receiver again. Observe the eye diagram. What is the optimum sampling time now?

10. Set the transmitter's peak frequency deviation to 20,000 Hz, 10,000 Hz, 5000 Hz, and 2500 Hz. For each case, run the transmitter and then the receiver. Record the peak value of the baseband output.

11. Return the peak frequency deviation to 5000 Hz. Change the "pulse shaping filter" control at both the transmitter and the receiver to "Gaussian." In the Gaussian setting, the receiver filter is a wire, i.e., no filter at all. Run the transmitter and then run the receiver. Once the receiver has acquired a block of data, you may stop the transmitter. Observe the baseband output signal and the eye diagram.

Questions

1. With the peak frequency deviation set at 5000 Hz, compare the rate of spectral rolloff with rectangular pulses and with Gaussian filtering. Also, by examining the plot of the transmitted message signal, can you find evidence of intersymbol interference when Gaussian filtering is used?

2. For each value of peak frequency deviation listed in Step 5, compare the null-to-null bandwidth of the transmitted signal with the bandwidth predicted by Carson's rule, Eq. (1). Compute the percentage difference in each case. Is Carson's rule more accurate for large peak frequency deviation or for small peak frequency deviation?

3. Observe the eye diagram shown on the receiver front panel. Compare the display when the eye diagram shows the baseband output waveform and when the eye diagram shows the aligned baseband output waveform. Describe what function *PulseAlign(real)* is performing.

4. Make a plot of the amplitude of the receiver's baseband output waveform vs. the peak frequency deviation. What is the relationship between these two quantities?

5. As described in Lab 7: Amplitude-Shift Keying, one of the quantities easily seen on the eye diagram is the optimum decision threshold location. In the present lab project, you were instructed to set the decision threshold to zero. Run the transmitter and receiver several times for different values of peak frequency deviation and for filtering set to "none" and "Gaussian." Observe the eye diagram and comment on the appropriateness of using zero for the decision threshold.

Optional

To verify the correctness of the received bit sequence, you can add *AddFrameHeader(real)* to the transmitter and add *FrameSync(real)* and *MT Calculate BER* to the receiver. Follow the instructions given in Lab 7: Amplitude-Shift Keying. The *AddFrameHeader(real)* and *FrameSync(real)* can be found in the *BasicUSRPLabs folder*.

8.5 Report

Prelab

Hand in documentation for the programs you created for the transmitter and receiver. Also include documentation for any functions you created. To obtain documentation, print out legible screenshots of the front panel and block diagram.

Lab

Submit the programs you created for the transmitter and receiver. Also submit any functions you created. Be sure your files adhere to the naming convention described in the instructions above.

Resubmit documentation for any functions you modified during the lab.

Answer all of the questions in each of the sections marked *Questions* above.