OREGON STATE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE

ECE 464/564 **DIGITAL SIGNAL PROCESSING** October 8, 2023

COMPUTER PROJECT #1

Due: 11.59 pm, October 30, 2023

Goal

The overall goal of the series of computer projects in this class is to develop a simulation system for studying quadrature phase shift keying (QPSK) transmission and digital receivers for QPSK signals. In the process, you will demonstrate your knowledge of

1. Filter design and realization
2. Sampling
3. Modulation and demodulation
4. Using DFT and inverse DFT

Project Description

In QPSK systems, each symbol may take one of four possible values, and thus can be represented using two bits. The block diagram in Figure 1 represents the components of the project.

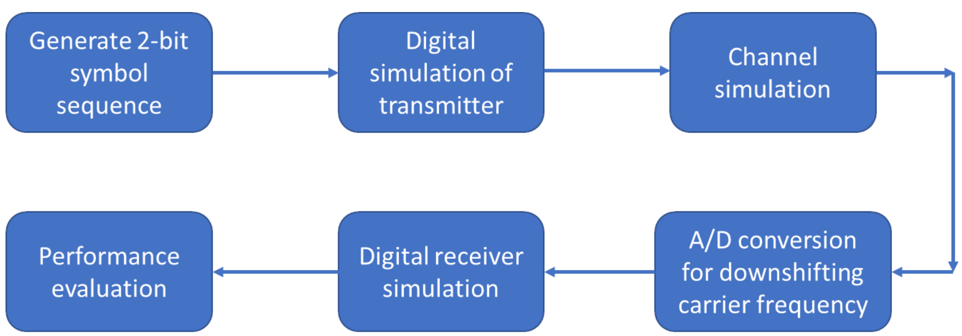

Figure 1: Components of the communication systems simulator.

We have divided the project into 3 modules to be completed during the semester.
The tasks to be accomplished in each module are:

1. Module #1 - Generation of 2-bit symbol sequence, digital simulation of transmitter and channel.

2. Module #2 - Receiver design implementing A/D conversion for down shifting carrier frequency.

3. Module #3 - Digital receiver simulation and performance evaluation.

Module #1

In this module, you will simulate the transmitter section of a QPSK communication system. Upon completion of this module, you would have performed the following tasks:

1. Generation of the symbol sequence.

2. Upsampling the sequence for pulse shaping and modulation.

3. Pulse shaping with raised cosine filter.

4. Low pass filtering.

5. Modulation with carrier frequency.

6. Simulation of the channel and transmission of the message signal.

Generation of Symbol Sequence

We will assume that the information sequence consists of samples that are independent from each other, and takes one of four possible values with equal probability. We will represent these four values with two bits (so that the symbol set is given by $[(-1,-1),(-1,1),(1,-1),(1,1)]$). Perhaps the easiest way to generate such a symbol sequence is to start with two independent and zero-mean pseudo-random sequences $x_1[n]$ and $x_2[n]$ (You may choose Gaussian sequences if you wish and use the `randn` command in Matlab. The only thing that matters is that the distribution is symmetric about zero), and create the sequence $\{(b_1[n], b_2[n]\}$ such that

$$b_1[n] = \begin{cases} 1 & ; \quad x_1[n] \geq 0 \\ -1 & ; \quad x_1[n] < 0 \end{cases}$$

and

$$b_2[n] = \begin{cases} 1 & ; \quad x_2[n] \geq 0 \\ -1 & ; \quad x_2[n] < 0 \end{cases}$$

The symbol given by $(b_1[n], b_2[n])$ at time n takes one of four values and will represent the information we wish to transmit at time n.

Digital Simulation of the Transmitter

Figure 2 shows the block diagram for a system that takes $b_1[n]$ and $b_2[n]$, shapes the sequences into pulses so that the signals have finite bandwidth, and then modulates the signal to a higher frequency for transmission.

Even though we are transmitting information represented digitally, keep in mind the fact that the signal transmission between the transmitter antenna and the receiver antenna is an analog process. We are simply representing the analog signals by a discrete-time signal sampled at a relatively

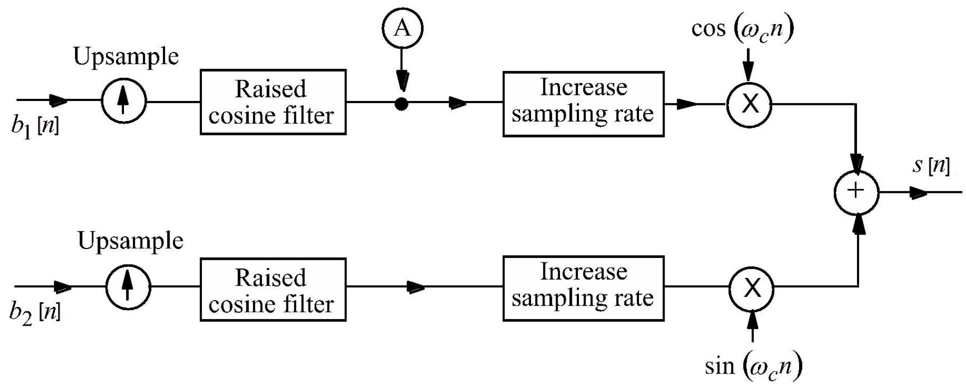

Figure 2: Block for simulating the transmitter.

high rate. Similarly, the analog processing such as transmission through an analog channel will be simulated digitally. Assume that the information rate is 1 symbol per T seconds. Furthermore, assume that the sampling rate is normalized to 1 sample/second. For this project, assume $T = 4$ s. so that each symbol is sampled four times. If we use rectangular pulses to represent the symbols, the signal will have infinite bandwidth and will be distorted by a bandlimited channel. Instead, we will use a pulse shape that has infinite duration (theoretically) in the time domain, but finite bandwidth. The frequency response of the pulse is given by the following equation:

$$H_{rc}(e^{j\omega}) = \begin{cases} \sqrt{T} & ; \quad 0 \leq |\omega| \leq \dfrac{\pi(1-\beta)}{T} \\ \sqrt{\dfrac{T}{2}\left\{1 + \cos\left[\dfrac{T}{2\beta}\left(|\omega| - \dfrac{(1-\beta)\pi}{T}\right)\right]\right\}} & ; \quad \dfrac{\pi(1-\beta)}{T} \leq |\omega| \leq \dfrac{\pi(1+\beta)}{T} \\ 0 & ; \quad \text{Otherwise} \end{cases}$$

This pulse shape is known as the square-root-raised cosine pulse. In the above equations, β is a constant such that $0 \leq \beta \leq 1$, and you may choose $\beta = 0.5$ in your project. Recall that $T = 4$ for this project. You may evaluate the pulse shape $h_{rc}[n]$ as the inverse DFT of

$$H_{rc}(k) = H_{rc}(e^{j\omega})\big|_{\omega = \frac{2\pi}{N}k} \quad ; \quad k = 0, 1, 2, \cdots, N-1.$$

Remember to shift the samples on the second half of the inverse DFT to the beginning of the filter response in your calculation. (See note on the next page.)

(*Hint*: You may use the MATLAB commands `fft`, `ifft` and `fftshift` to compute the DFTs and IDFTs, and for data shifting. In later modules, you will have to write your own FFT routines after the algorithms have been discussed in class.)

Note of Discrete Fourier Transform (DFT): As indicated above, $X(k)$, the N-point DFT of a signal $x[n]$ are the discrete-time Fourier transform (dtFt) of $x[n]$ sampled at frequencies $\omega = 2\pi k/N$ for $k = 0, 1, \cdots, N-1$. The range of frequencies considered here is the first period of the frequency variable in the range 0 to 2π. If you want to find what the dtFt is at some negative frequencies, all we have to do is shift the second half of the frequency sample to the left by N (corresponding to a shift of 2π radians/sample).

For reasons that will become clear when we learn about DFT later in the class, the DFT and inverse DFT operation also assumes that the time domain signal is periodic with period N. Because $H_{rc}(e^{j\omega})$ above is real-valued its inverse dtFt (say, $h_{rc}[n]$) is symmetric about $N = 0$. (Show that this is true.) However, when you take the inverse DFT, you will get a signal that extends from 0 to $N-1$. You can get the symmetric signal by shifting the second half of the inverse DFT to the left by N samples. You can then create a causal pulse shape by moving everything to the right so that all non-zero values occur at $n = 0$ or later.

In the ideal case, $h_{rc}[n]$ has infinite duration. However, we can implement the system digitally only if we have a finite number of coefficients. In our case, I recommend retaining enough samples so that the pulse overlaps eight symbols. This implies choosing a pulse that is 32 samples long in our case. You can take samples in the range $-16 \leq n \leq 15$, and shift the results by 16 samples so as to get a causal sequence. When you compute the inverse DFT, please take a much larger number of samples of $H_{rc}(e^{j\omega})$, so that you can minimize time-domain aliasing. Also, do not forget to shift the inverse DFT appropriately so that there is no discontinuity when you go from $n = -1$ to $n = 0$.

To perform the pulse shaping, we have to increase the sampling rate by a factor of 4 (because $T = 4$) by inserting 3 zeros between each symbol and then smoothing the result out by the square-root-raised cosine pulse. (Why does this work?) You should use the FIR filter program that you wrote earlier to perform this filtering operation by considering $h_{rc}[n]$ as the unit impulse response of the FIR filter being implemented. (If your original program was not efficient in its use of multipliers and memory as we discussed in the class, you must rewrite the program to implement it efficiently.) Once the pulse shaping is performed on $b_1[n]$ and $b_2[n]$, we are ready to modulate the information-bearing signal to a much higher frequency. However, we need to realize that so far, we have sampled the signals at only slightly more than five times the bandwidth (note that the bandwidth of the square root raised cosine pulse is $3\pi/8$ radians/sample). In order to simulate the carrier frequency, which is in general several orders of magnitude larger than the bandwidth of the signal to be transmitted, we need to increase the sampling rate appropriately. For our simulations, increase the sampling rate (upsample) by a factor of 20. To do this, you need to insert 19 zeros between adjacent samples and then lowpass filter the resulting high-rate sequence. What should the bandwidth of this filter be? Design a linear phase FIR filter with appropriate passband and stopband characteristics to perform this task. (See note on filter design on the next page.)

Comments on FIR filter design: MATLAB commands such as `fir1`, `remez` are useful for designing FIR filters. One approach to designing linear phase FIR filters is to use the windowing technique. Consider the problem of designing a symmetric low pass linear phase FIR filter having the desired frequency response

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\frac{\omega(M-1)}{2}} & ; \quad 0 \leq |\omega| < \omega_c \\ 0 & ; \quad \text{otherwise} \end{cases}$$

A delay of $(M-1)/2$ units is incorporated into $H_d(\omega)$. In the windowing method, we force the impulse response function to be of length M by truncating the inverse Fourier transform of $H_d(\omega)$ as shown below:

$$h_d[n] = \begin{cases} \dfrac{\sin(\omega_c(n-(M-1)/2)}{\pi(n-(M-1)/2)} & ; \quad 0 \leq n \leq M-1 \\ 0 & ; \quad \text{Otherwise.} \end{cases}$$

One problem with truncation as described above is that it may result in unacceptably low stopband attenuation. Multiplying the truncated window by a window function such as Hamming, Hann, Blackman, etc. can alleviate this problem significantly. You should read about this method of FIR filter design from a text book before performing the filter design.

Perform the modulation digitally using a carrier frequency $\omega_c = 0.44\pi$ radians/sample. (Incidentally, what is the bandwidth of the baseband signal after the sampling rate change in radians/sample?)

Using the cosine and sine functions for modulation in the two branches of Figure 2 allows us to transmit the four levels of the symbols using appropriate phase shifts. The process can be reversed at the receiver to retrieve $b_1[n]$ and $b_2[n]$.

Channel Simulation

An ideal channel will transmit the signal to the receiver without any distortion. However, most channels are not ideal, and they introduce distortions that may be nonlinear and time-varying to the signal. For the purpose of this project, we will assume that the channel introduces only additive noise distortion. The received signal is modeled for the additive noise case as

$$r[n] = s[n] + \eta[n],$$

where $s[n]$ is the transmitted signal (as shown in Figure 2) and $\eta[n]$ is additive white noise that is independent of $s[n]$. You can model this noise as Gaussian with zero mean value. You will study the effect of the noise at different levels (variances) on the performance of the communication receiver.

Project Report

Your report must be *formally written* and must not exceed 8 typed (single-column, single-space) pages including figures, appendices, and the cover page for each module. (Do use a readable font size!) You should also upload any code you wrote to Canvas so that the TA can validate the code while grading your report. Make sure that instructions to run the code is included at the beginning of your code. The report should contain at least the following components:

1. An introduction describing the problem, your goals, and a summary of the results.
2. A section on methods that describes the details of your designs and calculations.
3. A section on performance evaluation describing your experimental techniques, and the results of the experiments. Remember that it is not enough to simply include your results. You should have an explanation for what the results mean as well as discussions of any variation from expected results. Be sure to point out issues you want to highlight. For Project 1, I have not explicitly stated what kind of performance analysis you should do. Make sure that you have verified that each component of your program works, and explain with details and examples as needed, how you performed the verification.
4. A concluding section that summarizes your work and makes your observations. Make sure that you include your thoughts about why the system performed in the way it did. Include a paragraph that describes all the concepts you learned and used to do this project. Also include a statement about things you could have done on this project to make your work better.

Note: Please take the time to write a clear and organized report that discusses your work, results and explanations as described above. 30% of the grade in this assignment will be for the organizational and communication aspects of the report. The other 70% will be for the technical work and its accurate evaluation.