OREGON STATE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE

ECE 464/564 **DIGITAL SIGNAL PROCESSING** November 13, 2022

COMPUTER PROJECT #3

Due: 5:00 p.m., November 28, 2022

Goal

This is the third in a series of three computing assignments that, together, will take you through the process of developing a simulation system for studying quadrature phase shift keying (QPSK) transmission and digital receivers for QPSK signals. In this part, you will work with and demonstrate your knowledge of

1. DFT-based realization of FIR filters;
2. matched filters;
3. delay estimation using cross correlators;
4. symbol detection using hard limiters; and
5. performance evaluation of the overall communication system.

Project Description

The block diagram for the receiver is shown in Figure 1. In this assignment, you will work on blocks from the point marked **B** in the figure. In addition, you will evaluate the overall performance of the communication system you built during the three assignments.

Filter Realization

For the performance evaluation, you will need to run the complete software system you developed. Recall that you have so far been using the direct form realization of the FIR filters. For this part of the assignment you should rewrite all the filter programs using the overlap-save method. For this, you should code your own FFT algorithm and use it wherever FFT algorithms are needed. You may code a power-of-two FFT, and choose either a decimation-in-time or decimation-in-frequency algorithm.

Equalizer

In practice, the channel will have memory and its characteristics may be linear or nonlinear, in addition to the additive noise introduced by the channel. In such cases, it is necessary to estimate the effects of the channel and then to compensate for such effects using the equalizer. Since we assume that the channel has no memory and does not introduce any distortion other than the additive noise, this project does not need the equalizers shown in the receiver block diagram.

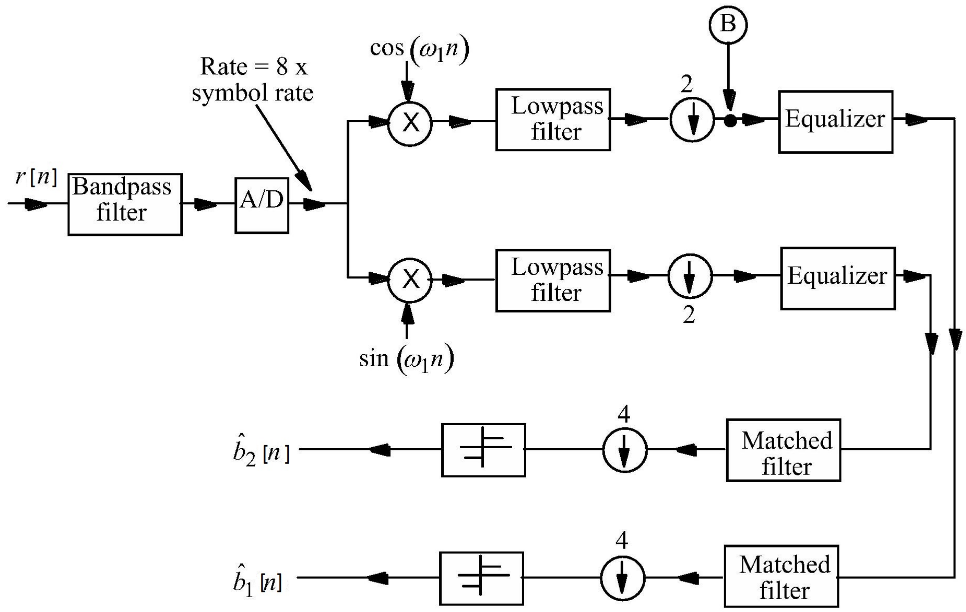

Figure 1: Block diagram of the receiver.

Matched Filtering

The impulse response of the matched filter is given by

$$h_{mf}[n] = \begin{cases} h_{rc}[31-n] & ; \quad n = 0, 1, 2, \cdots, 31 \\ 0 & ; \quad \text{Otherwise.} \end{cases}$$

The objective of the matched filter may be looked upon as that of template matching. As you perform the filtering, when the data aligns most closely with the shape, you will get a large positive spike if the symbol ($b_1[n]$ or $b_2[n]$) is $+1$ and a large negative spike if the symbol is -1.

Delay Estimation

Note that we only need one sample out of four available during each symbol duration. It is important to figure out where exactly to sample, and to do this, we must first determine the delay caused by the channel as well as the signal processing at the transmitter and receiver. I suggest that you estimate the delay experimentally in the following manner.

Let $d_1[n]$ and $d_2[n]$ represent the output of the equalizers before sub-sampling. Let $f_1[n]$ represent the signal at the point marked **A**. Find the lag m at which the correlation between $f_1[n]$ and $d_1[n]$ is the maximum. The correlation function may be computed as

$$c_{fd}[m] = \sum_{n=0}^{P-1} f_1[n] d_1[n-m],$$

where P represents the total number of samples used in the estimate. You should perform the estimation with the noise present in the received signal. The lag m for which $c_{fd}[m]$ is the maximum is the estimate of the delay we are looking for. (Why?) Since the bottom branch is essentially identical (except for the sine and cosine functions), you can use one estimate of the delay for both branches. You should also be able to estimate the delay from knowledge of the various filters and verify that your estimate obtained through use of the correlation function is correct. (In practice, the channel will introduce an unknown delay into the signals, and you will need to estimate the delay. Direct calculation as suggested here from knowledge of the receiver design is not possible.)

Symbol Detection

Once you know the delay, make sure that you choose the sampling instants (which one of the four adjacent samples to keep) such that they match closest to the delay. Once the sampling is done, you decide on the estimate of $b_1[n]$ and $b_2[n]$ by a simple hard limiter (clipping) operation whose output is 1 or -1 depending on whether the input is positive or negative, respectively.

Performance Evaluation

The data for "training" (Training data may be used for estimating the delay for this project. In general you can use training data to estimate the parameters of the equalizer also.) must be different from the data for performance evaluation.

The measure we will use for performance evaluation is the number of symbol errors when 1000 input symbols were transmitted. One issue you will have to remember is that symbol error is different from bit error. You have the correct symbol only when you have both bits associated with a symbol correctly estimated. Find the number of symbol errors that occurred for the cases when the signal-to-noise ratio (SNR) (*i.e.*, the ratio of the variance of the signal arriving at the receiver to the variance of the noise in the received signal.) Use SNR values of 10, 5, and 0 dB for your evaluation. To get the dB values multiply the logarithm (to base 10) of the ratio of the measured variances by 10.

It is possible that, for high signal-to-noise ratios, you will not get many errors with 1000 symbols. In such cases, consider running the test using much longer symbol sequences. Also, there may be errors in the beginning of the data stream because of the transient effects of the signal processing. You should consider only errors that occur in the steady state.

Project Report

Your report must be formally written, comprehensive and must not exceed 12 pages including figures, appendices, and the cover page. (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.
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, write about additional things you could have done or considered to make your project and the report truly professional.