Lab 3 USRP

3 USRP Hardware Implementation

A *matched filter* is a theoretical framework and not the name of a specific type of filter. It is an ideal filter which processes a received signal to minimize the effect of noise. Hence, it maximizes the signal to noise ratio (SNR) of the filtered signal. It happens that an optimum filter does not exist for each signal shape transmitted and it is a function only of the transmitted pulse shape. Because of its direct relationship to the transmitted pulse shape, it is called a matched filter. The two most common matched filters are integrate and dump and root raised cosine. We have already studied integrate and dump in Section 2.3, so we will focus on root raised cosine in this section.

In this section, we use two models to illustrate a typical setup in which a transmitter uses a square root raised cosine filter to perform pulse shaping and the corresponding receives uses a square root rased cosine filter as a matched filter. The receiver plots an *eye diagram* from the filtered received signal. Based on this diagram, we can learn the effect of matched filter in a communications system.

3.1 Eye Diagram

In telecommunication, an *eye diagram*, also known as an *eye pattern*, is an oscilloscope display in which a digital data signal from a receiver is repetitively sampled and applied to the vertical input, while the data rate is sued to trigger the horizontal sweep. It is so called because, for several types of coding, the pattern looks l8ike a series of eyes between a pair of rails.

Several system performance measures can be derived by analyzing the display. If the signals are too long, too short, poorly synchronized with the system clock, too high, too low, too noisy, or too slow to change, or have too much undershoot or overshoot, this can be observed from the eye diagram. For example, in figure 15, D_A is the range of amplitude differences of the zero corssings and is a measure of distortion caused by *intersymbol interference* (ISI), J_T is the range of amplitude differences of the zero crossing and is a measure of the timing jitter, M_N is a measure of noise margin, and S_T is a measure of sensitivity-to-timing error. In general, the most frequent usage of the eye pattern is for qualitatively assessing the extent of the ISI. As the eye closes, the ISI increases; as the eye opens the ISI decreases.

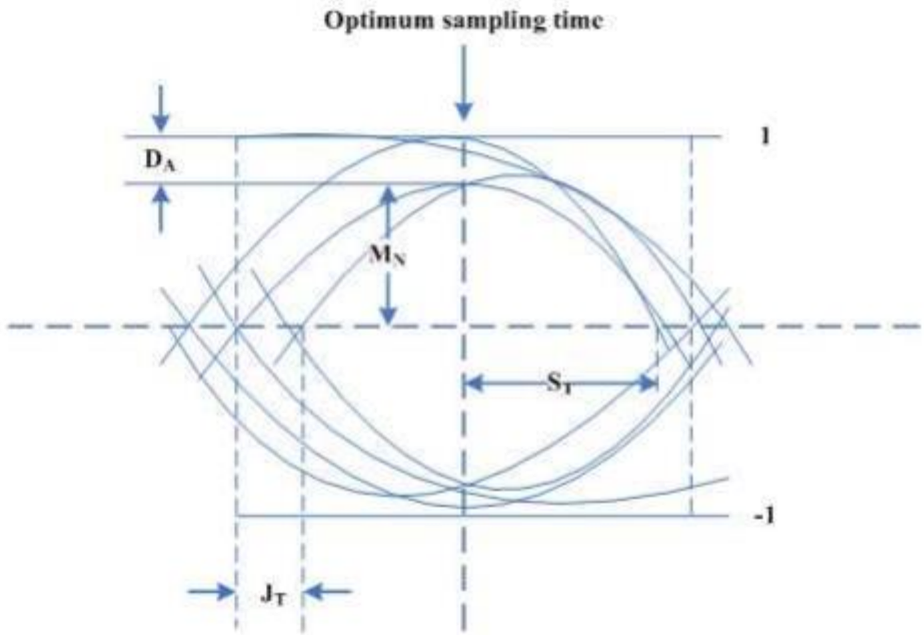

Figure 1: A typical eye pattern. The width of the opening indicates the time over which sampling for detection might be performed. The optimum sampling time corresponds to the maximum eye opening, yielding the greatest protection against noise. If there were no filtering in the system, then the system would look like a box rather than an eye.

3.1.1 Discrete-Time Eye Diagram Scope

The EYE DIAGRAM block displays multiple traces of a modulated signal to produce and eye diagram. You can use the block to reveal the modulation characteristics of the signal, such as pulse shaping or channel distortions. This is the most important block that will be used in this section.

An *open eye* pattern corresponds to minimal signal distortion. Distortion of the signal waveform due to intersymbol interference and noise appears as *closure* of the eye pattern.

3.2 Matched Filter Observation

In this section, for the transmit side download and use eye_tx.grc. On the receiver side, download and open eye_digram.grc. The two flow graphs can be seen below.

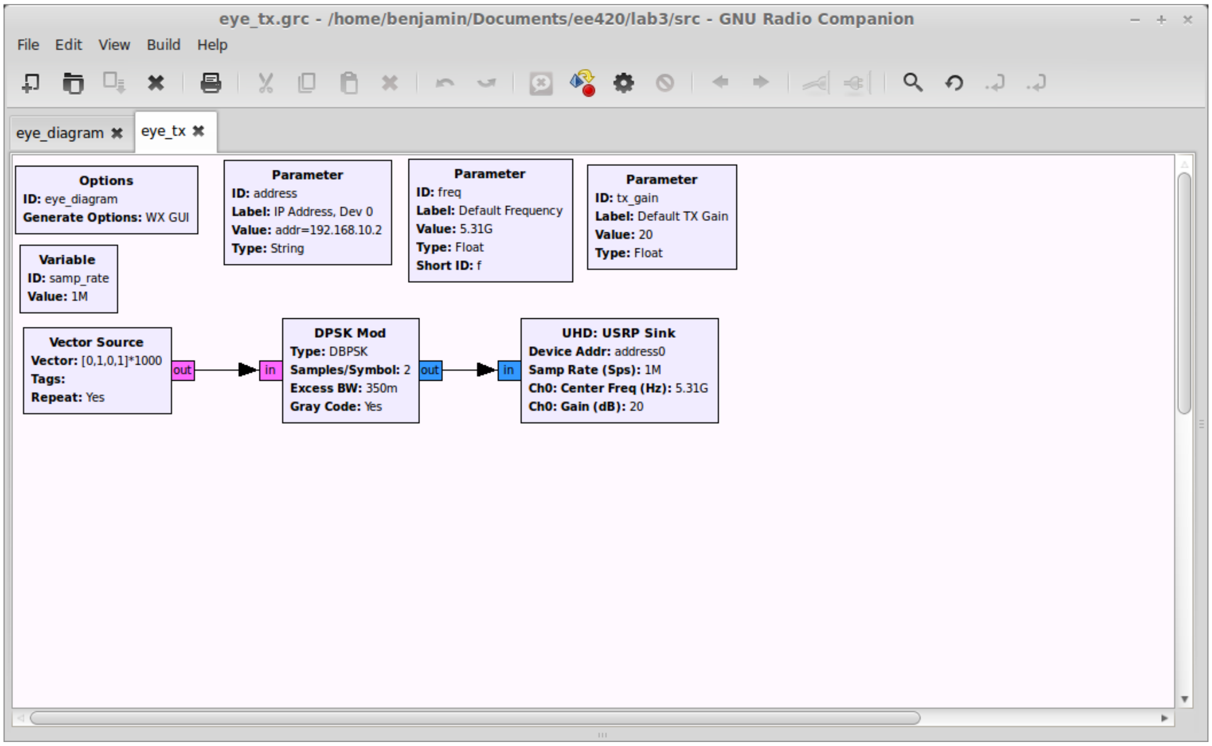

Figure 2: eye_tx.grc

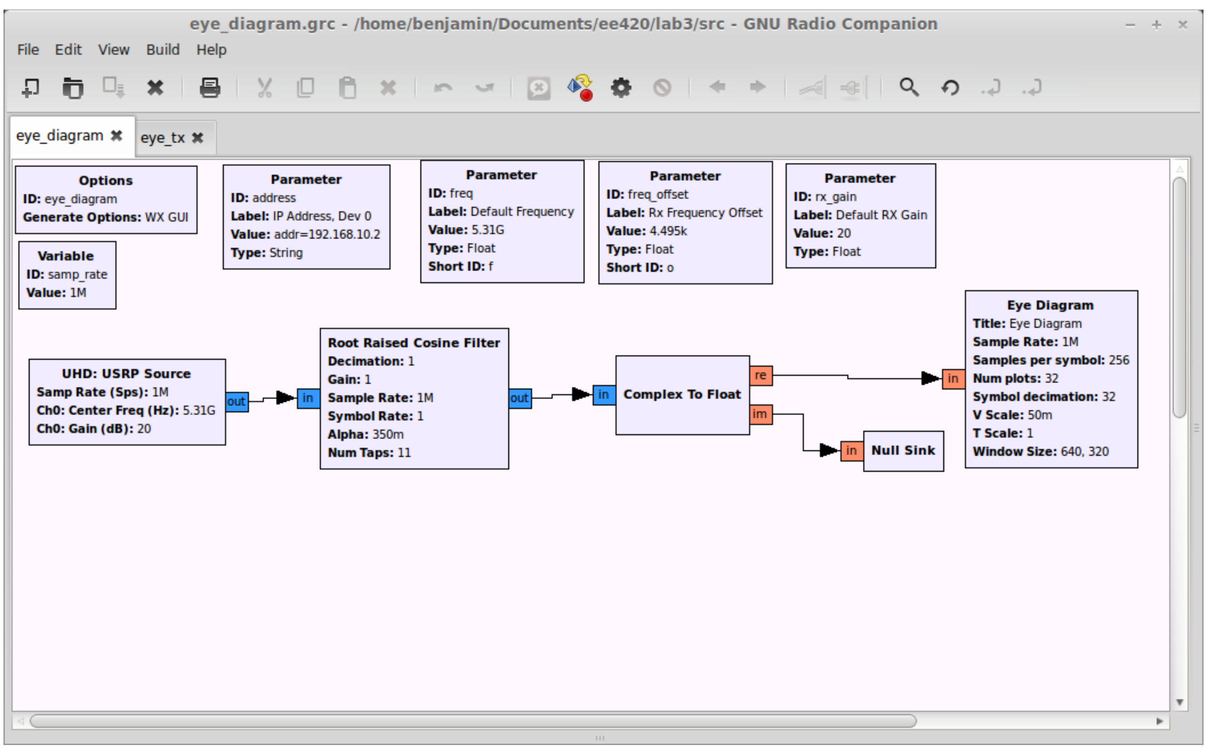

Figure 3: eye_diagram.grc

*Make sure to change the center frequency to your assigned frequency and set the offset to the offset you calculated in lab 1.

Perform the following tasks and plot your observations.

- Compare the **Excess BW** parameter on the DBPSK Mod block on the transmitter side with the **Alpha** parameter on the Root Raised Cosine Filter on the receiver side; see whether they match each other.
- Set these two parameters so they match and run the system. How does the eye diagram look? Capture and describe your observations.
- Change the parameters to make them *mismatch*. How does the eye diagram look? Capture and describe your observations and compare them with your previous observations.
- Delete the Root Raised Cosine Filter block. How does the eye diagram look? Capture and describe your observations and compare them with your previous observations.

4 Open-ended Design Problem: Duplex Communcatino

4.1 Duplex Communication

A duplex communication system is a system composed of two connected parties or devices that can communicate with one another in both directions. Duplex systems are often employed in many communications networks, either to allow for a communication “two-way street” between two connected parties or to provide a “reverse path” for the monitoring and remote adjustment of equipment in the field.

Systems that do not need the duplex capability use instead simplex communication. These include broadcast systems, where one station transmits and the others just “listen”. Several examples of communication systems employing simplex communications include television broadcasting and FM radio transmissions.

Thus far, the models that we have been using in this course all use simplex communication, such as the DBPSK transmitter and receiver studied in previous laboratory experiments. Since we are using the XCVR2450 transceiver daughterboard, our USRP2 can be either a transmitter or a receiver. However, it cannot be a transmitter and a receiver at the same time. Consequently, in this open-ended design problem, you will need to implement a system that is capable of doing half-duplex communication.

4.2 Half-duplex

A half-duplex (HDX) system provides communication in both directions, but only one direction at a time (not simultaneously). Typically, once a party begins receiving a signal, it must wait for the transmitter to stop transmitting before replying.

An example of a half-duplex system is a two-party system such as a “walkie-talkie” style two-way radio, wherein on must use “Over” or another previously designated command to indicate the end of transmission, and ensure that only one party transmits at a time, because both parties transmit and receive on the same frequency.

A good analogy for a half-duplex system would be a lone-lane road with traffic controllers at each end. Traffic can flow in both directions, but only one direction at a time, regulated by the traffic controllers.

There are several different ways to control the traffic in a half-duplex communication system, and we suggest you to use *time division duplexing* (TDD).

4.3 Time Division Duplexing

Time division duplexing (TDD) refers to a transmission scheme that allows an asymmetric flow for uplink and downlink transmission which is more suited to data transmission. In a time division duplex system, a common carrier is shared between the uplink and downlink, the resource being switched in time. Users are allocated on or more timeslots for uplink and downlink transmission, as shown in the figure below.

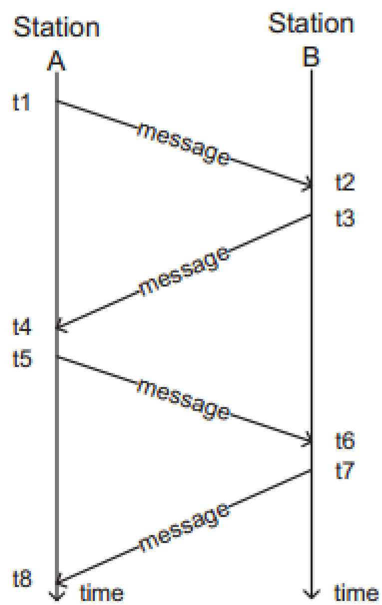

Figure 4: A half-duplex communication system using time division duplexing. A common carrier is shared between Station A and Station B, the resource beign switched in time.

For radio systems that aren't moving quickly, an advantage of TDD is that the uplink and downlink radio paths are likely to be very similar. This means that techniques such as beamforming work well with TDD systems.

Examples of time division duplexing systems are:

- UMTS 3G supplementary air interfaces TD-CDMA for indoor mobile telecommunications
- DECT wireless telephony

- Half-duplex packet mode networks based on carrier sense multiple access, for example 2-wire or hubbed Ethernet, Wireless local area networks and Bluetooth, can be considered as Time Division Duplex systems.
- IEEE 802.16 WiMAX

4.4 Hints

Suppose you have a radio station A and a radio station B, you are required to perform the following tasks in time sequence: Station at transmits “Hello World” to Station B, Station B transmits “Goodbye” to Station A, Station A transmits “Hello World” to Station B, Station B transmits “Goodbye” to station A, etc..

- At the beginning, you may want to try something simple and see whether it works. For example, you can transmit ‘1010…’ instead of “Hello World”, such that frame synchronization is not necessary. You can test one cycle only, that is station A transmits “Hello World” to Station B and station B transmits “Goodbye” to Station A.
- In order to make it an automatic communication system, you need to write GNURadio code for station A and for station B, such that all you need to do is click the “Run” button on the two stations and the beginning of the simulation
- A simple, yet inefficient way to build this system would be to have two different top_blocks, an transmit `top_block` and a receive `top_block` and use the `tb.start()`, `time.sleep()`, `tb.stop()` function calls to switch between them.
- A more efficient way to build the system is to define a separate receive chain and transmit chain flow graphs in the same `top_block` using *message queues* and allow the UHD API (USRP blocks) to handle the switching between transmitting and receiving for you. You can download and use `half_duplex_mac.py` as an outline to get started with.