18-758 Project Logistics Handout

Professor: Rohit Negi (negi@ece.cmu.edu)

Overview

The 18-758 project is designed to give you the opportunity to use a real digital communication system. In class you were given a block diagram of a communication system and diagram of a passband modulator and demodulator. These block diagrams are shown in the figure below.

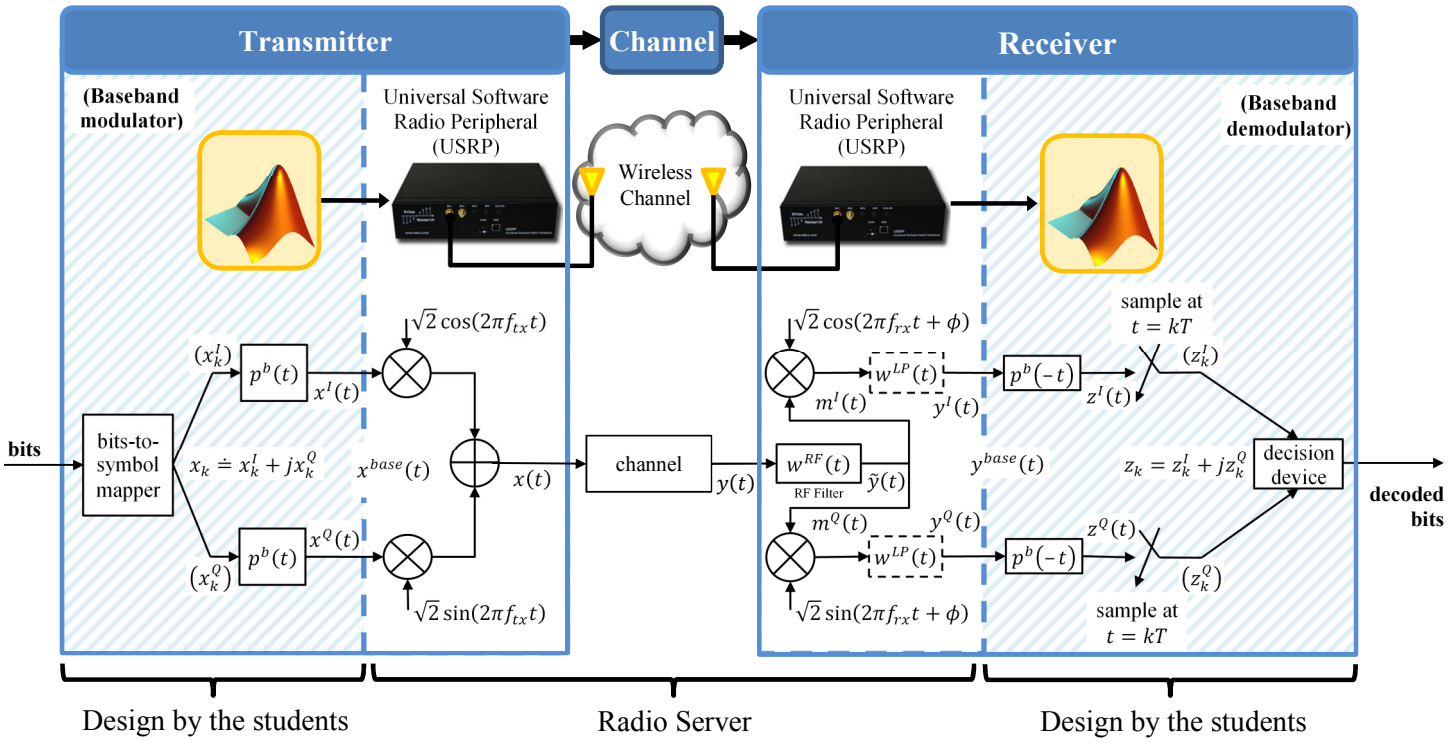


Figure 1: Block diagram of the 18-758 Project

As seen in the figure, you will design the baseband signal, $x^{base}(t)$, in MATLAB®. You will upload the signal that you design to the radio server. The signal you upload will be transmitted by a radio through a wireless channel to a receiving radio. Then you will be able to download the received baseband signal, $y^{base}(t)$, from the radio server. You will be able to load this signal into MATLAB®, where you will do all the necessary processing to obtain the decoded bits. Shown in Figure 2 is a picture of the system. Users can connect to the radio server via an internet connection. The radio server allows users to upload signals to be transmitted and received by a couple of Universal Software Radio Peripherals (USRPs) designed by Matt Ettus, a CMU alum.

Requirements for Project

- MATLAB®
- SSH Tectia – File Transfer (you need SFTP)

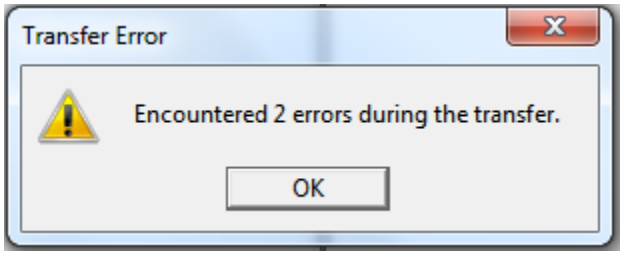
Figure 2: Picture of the Radio Server Setup

Transmit Signal Design

1. Using MATLAB, Create the baseband signal, $x^{base}(t) = x^I(t) + jx^Q(t)$
   - The signal must be in vector form (i.e., one-dimensional complex sequence).
   - This sequence represents a $f_s = 2\ MHz$ signal, which is the sampling rate of the USRP.
   - The signal length must be less than or equal to $L = 10000$ samples. Thus, the length in time is L / f_s seconds.
   - The signal must be double-valued and its I and Q values must lie between ± 1. (i.e., $|x^I(t)| < 1$ and $|x^Q(t)| < 1$).
2. Set the MATLAB variable `transmitsignal` to equal your created baseband signal vector.
3. Save this variable into a .mat file using `save('transmitsignal.mat','transmitsignal')`.

Transmission and Reception

1. Start SSH – Tectia File Transfer program.
2. Click on "Quick Connect".
3. You should be prompted with another window asking for the Host Name, User Name, and Port Number
   Host Name: CMU-796353.WV.CC.CMU.EDU
   User Name: *Your Project User Name* (default: Andrew ID)
   Port Number: 22
4. Click the connect button once you have provided the correct information.
5. For the first time use, follow the on-screen instructions for the next windows that will pop-up.
6. You will be prompted to enter your password. Once entered, you should receive the welcome dialog box.
7. Now you are ready to transmit your signal, in the right pane navigate to the current folder noted above and locate your `transmitsignal.mat` file.
8. Right click on the `transmitsignal.mat` file you want to transmit and click upload.
9. Ignore the following error message if it appears.



10. The received signal obtained by the USRP in response to your transmit signal should appear in the right pane as the file `receivedsignal.mat`. It should typically take about 10 seconds for our software to complete the processing needed, which includes TDMA transmission of all active users. Refresh the pane a couple of times to check that the file has appeared.
11. If our software detects that your transmit signal does not adhere to prescribed limits, your signal will not be transmitted, so that you will not receive the `receivedsignal.mat` file. Instead, an error message will be placed in your folder in the `errormessage.txt` file.
12. Right click on the `receivedsignal.mat` signal file or the `errormessage.txt` file, as the case may be, and download it to the noted current folder.
13. In case of error: Open the `errormessage.txt` file and correct the error noted by it regarding your signal.
14. In case of correct reception: Return to MATLAB to process the `receivedsignal.mat` file.

Processing the Received Signal File

1. Type `load receivedsignal.mat` in Matlab. This loads the vector variable `receivedsignal` into Matlab workspace. This is the baseband receive signal $y^{base}(t)$.
   - Its sample rate is 2 MHz
   - The received signal is longer than your transmit signal due to zero padding that our software applied to your transmit signal.
   - The signal is double-valued and complex.
2. Finish your receiver processing to get the decoded bits.

Troubleshooting and Assistance

Real radios are being used in this project. By using real radios you will get a feel of what it's like to process signals using actual hardware. The system is designed to be robust but errors may occur, typically due to interference from WiFi, or (rarely) hardware failure. Errors should go away when you retransmit the signal.

Try to get a basic design working properly, and then incrementally push the design limits. This means that you can initially use some or all of the following simplifications:

1. Large symbol period, say, $T = 10$ samples.
2. Small packet length, say, $L = 200$ samples.
3. Long pilot sequence.
4. Rectangular pulse (easier to avoid ISI).
5. Lower constellation order (e.g., BPSK).
6. No channel coding.

Once a basic design works, you can then gradually add more sophisticated features to your system, to push the system limits.

To get you started, we have included a Matlab file `studentdemo.m`. Read the instructions in that file to demonstrate basic USRP transmission and reception. This file uses the included `transmitsignal.mat` file for transmission. It also needs a `receivedsignal.mat` file that you must supply by transmitting the `transmitsignal` signal. The program will then display the transmitted and received signals. For reference, we have also included a typical received file (`receivedsignalEXAMPLE.mat`) to show you what to expect from the USRP.