WARP WIRELESS OPEN-ACCESS RESEARCH PLATFORM

Lab 3: Introduction to WARPLab

Melissa Duarte & Siddharth Gupta

Rice University

WARP Project

Document Revision 8

April 9, 2008

1 Introduction

Lab 3 is an introduction to the WARPLab framework. The WARPLab framework will be used to transmit signals created in MATLAB over a wireless channel using the WARP boards. This lab is divided into the following five exercises.

1. In the first exercise you will learn how to use the WARPLab GUI (a MATLAB Graphical User Interface) and the functions in the WARPLab framework. You will create in MATLAB a vector of samples to transmit and transmit the samples over a wireless channel using the WARP boards and WARPLab framework.

2. In the second exercise you will use the WARPLab framework to estimate a narrowband flat fading wireless channel.

3. In the third exercise you will use MATLAB to design a transmitter and a receiver, and use the WARPLab framework to test the transmitter and the receiver by transmitting over a wireless channel using the WARP boards.

4. In the fourth exercise you will learn how to use the continuous transmission option of the WARPLab framework.

5. In the fifth exercise you will use the WARPLab framework to implement two-way transmission and reception of data.

Note: All files are stored in `C:\workshop\userN\` where `userN` is your user login location. This location will be referred to as `.\` for the rest of the lab.

Note: To avoid conflict with other groups using the boards, please test the code you write in any of the following three ways:
OPTION 1. Run the code's script from MATLAB's Command Window by entering the name of the script.
OPTION 2. In the menu bar go to Debug and select Run. If there are errors in the code, error messages will appear in the Command Window.
OPTION 3. Press F5. If the are errors in the code, error messages will appear in the Command Window.
DO NOT USE the Evaluate Selection option and DO NOT run the script by sections. To test any change, always run the whole script by following any of the three options above.

2 Basic transmission and reception of signals using WARPLab

In this first part of the lab you will learn how to transmit and receive signals using the WARPLab GUI and the functions defined in the WARPLab framework.

1. Open MATLAB and then open the WARPLab GUI by entering **warplab_siso_Gui** on the MATLAB command line. Click the **Go** button to transmit the default signal in the 'TxVector' field using the default Gains, Capture Offset, and Channel. The GUI will show a plot of the received In phase signal (Rx I), the received Quadrature signal (Rx Q), the spectrum of the transmitted signal (Tx Spectrum), and the spectrum of the received signal (Rx Spectrum), for both of the spectrums the x axis is in Hertz.

2. The signal to transmit is specified in the 'TxVector' field, modify this field and transmit different signals. You can transmit real or complex signals, the real and imaginary parts must be inside the interval [-1,1].

3. Modify the values of the Capture Offset and Gains and observe the effects on the received signal. Choose values for the Gains such that the received signal is not saturated and the I and Q signals have a peak of around 0.8. Write down these Gains, you will use them later.

4. Set the MATLAB Current Directory to `.\Lab3_WARPLab`.
5. Open the file **warplab_example_TxRx_WorkshopExercise** and follow the instructions given at the beginning of the file. You will learn how to transmit and receive a signal using the MATLAB functions in the WARPLab framework . You can view these functions in MATLAB by entering **edit FunctionName** in the MATLAB command line. For example, to open the function **warplab_initialize** enter **edit warplab_initialize**. The functions are also available in the repository, click here to access the repository.

3 Channel Estimation using WARPLab

1. Open the file **warplab_example_ChannelEstim_WorkshopExercise** and follow the instructions given at the beginning of the file. By following these instructions you will be able to write a MATLAB script that transmits and receives data using WARPLab and computes an estimate of the amplitude and phase of the channel by comparing the transmitted and received data.

4 Transmitting Bits over a Wireless Channel using WARPLab

1. Open the file **warplab_example_Comm_WorkshopExercise** and follow the instructions given at the beginning of the file. By following these instructions you will be able to write a MATLAB script that generates a stream of bits, modulates them using DQPSK, transmits the modulated symbols over a wireless channel using WARPLab, and demodulates the received signal to obtain the transmitted bits. Bit error rate (BER) is computed by comparing the transmitted bits with the bits recovered at the receiver.

5 Using WARPLab in continuous transmission mode

1. Open the file **warplab_example_ContinuousTx_WorkshopExercise** and follow the instructions given at the beginning of the file. By following these instructions you will learn how to use the contninuous transmission option avalaible in WARPLab. In continuous transmitter mode, the transmitter board will continue transmitting the samples stored in the transmit buffer until the user manually disables the transmitter.

6 Using WARPLab to implement two-way transmission and reception of data

1. Open the file **warplab_example_TxRxTwoWay_WorkshopExercise** and follow the instructions given at the beginning of the file. By following these instructions you will learn how to use WARPLab for two-way communication between nodes. First node A will transmit to node B and then node B will transmit to node A.

7 Optional Exercises

If you finish the lab with extra time, here are a few other exercises to try.

1. **BER Test.** Modify the **warplab_example_Comm_WorkshopExercise** file you created to compute the BER at different SNRs. You can change the SNR by changing the gains at the transmitter. Plot BER vs. SNR.