(https://www.edx.org)

HKUSTx: ELEC1200.1x A System View of Communications: From Signals to Packets (Part 1)

KarenWest (/dashboard)

Courseware (/courses/HKUSTx/ELEC1200.1x/3T2014/courseware) Course Info (/courses/HKUSTx/ELEC1200.1x/3T2014/info)

Course Outline (/courses/HKUSTx/ELEC1200.1x/3T2014/05fb01b36df14eb99ab54545dabc47f6/)

Grading Scheme (/courses/HKUSTx/ELEC1200.1x/3T2014/6e2be4dac3e44b4d9f812e7b5a5d5a29/)

Instructors (/courses/HKUSTx/ELEC1200.1x/3T2014/674fdd6887fe4f4bb73b984df4a5675b/)

Resources (/courses/HKUSTx/ELEC1200.1x/3T2014/a6a8267fef364cccbccd0128d091f11c/)

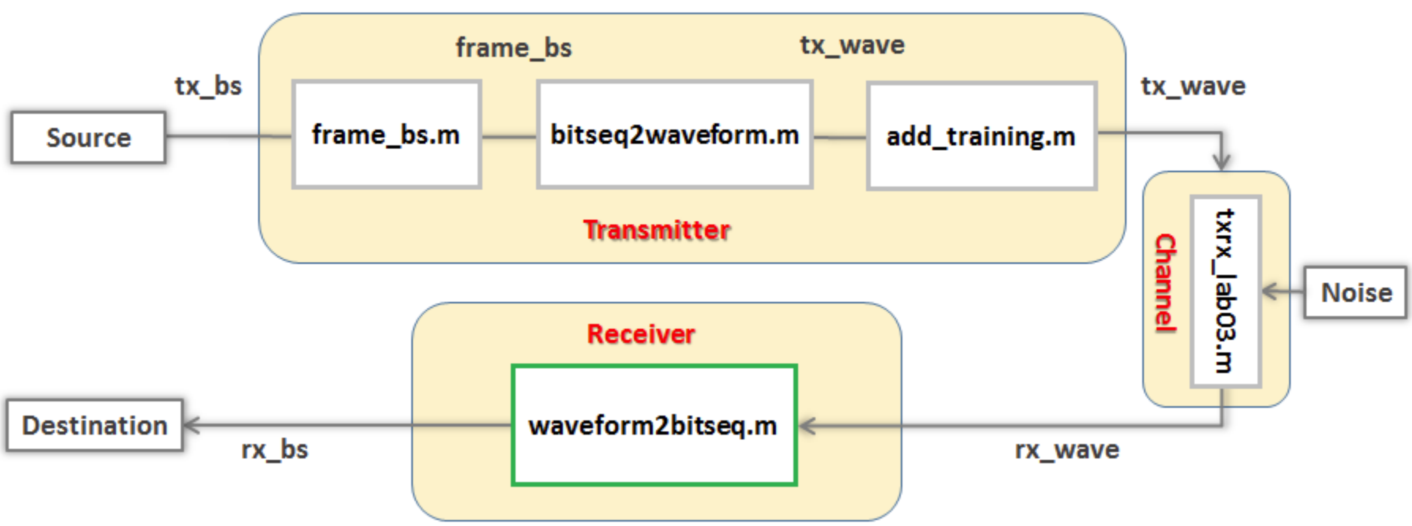
Discussion (/courses/HKUSTx/ELEC1200.1x/3T2014/discussion/forum)

Progress (/courses/HKUSTx/ELEC1200.1x/3T2014/progress)

Help

LAB 3 TASK 3 - DECODE THE RECEIVED WAVEFORM (1 point possible)

In this task, you will implment the process of recovering the bit sequence `rx_bs` from the received waveform `rx_wave`. This is the process that is implemented by the function `waveform2bitseq.m` that we used in Lab 1 and the function `receive` used in the lab demo video.



The MATLAB code in the window below simulates the transmission of a random bit stream over the communication channel under two different scenarios: one where the distance between transmitter and reciever is 10cm and the bit time is 50 SPB and the other where the distance is 15cm and the bit time is 20 SPB. Your goal is write code that can recover the transmitted bits from the received waveform `rx_wave` in both situations.

```
% setup transmission parameters
distance_list = [10 15];
SPB_list = [50 20];
num_dist = length(distance_list);
BER = zeros(1,num_dist);            % store bit error rates
setup_lab0303; % setup other storage variables

tx_bs = rand(1,1280) > 0.5;  % create sequence of 1280 random bits

```

```
for i = 1:num_dist,
    distance = distance_list(i); % transmission distance
    SPB=SPB_list(i);             % bit time in samples per bit

    % transmitter %
    tx_bs_frame = frame_bs(tx_bs);              % generate frame
```

Help

Unanswered

Run Code

Check Save *You have used 0 of 10 submissions*

INSTRUCTIONS

Step 1: Run the code as presented

After you click on the Run Code button, two figures will appear followed by two lines of text indicating the Bit Error Rate (BER) under the two channel scenarios. Because the bit recovery process has been implemented incorrectly, the bit error rates will be quite high (around 0.5 or 50%). This is because the code uses a fixed threshold for determining 0 and 1 bits, and because the subsampling points for bit decisions have been chosen incorrectly.

The two figures show the received waveform in blue for a period around the start bit, whose beginning sample, **`start_ind`**, is shown by the vertical cyan line. The subsampling points used in making bit decisions are shown by green asterisks. The threshold for comparison is shown in red.

Step 2: Correct the code

The code you must correct lies between the comments

`%---MODIFY CODE BETWEEN THESE LINES ONLY---%`

The sub-sampling points at which the received waveform is compared to the threshold for making bit decisions are contained in the vector **sample_ind**. Since there are 1280 bits to decode, this vector should have length 1280. The first element should correspond to the index at which the decision about the first bit should be made. This should happen at the sample where the first bit ends. Remember that each bit lasts for **SPB** samples and that the first bit is preceded by the start bit. The variable **`start_ind`** returned by the function **`find_start`** indicates the beginning of the start bit. Moving forward by **SPB** will bring you to the beginning of the first bit. Subsequent bit decisions should be made every **SPB** samples after that. See the lecture videos for more detail.

In the code given, the threshold for bit decisions is fixed. However, this threshold should vary depending upon the channel conditions. In the previous task, you estimated the threshold from the received training sequence by calculating the **average value** between the minimum and maximum values of the received training sequence. For simplicity, to avoid the need to detect and localize the training sequence, you can estimate the threshold by considering the maximum and minimum values over the entire received waveform **`rx_wave`**. You may find the MATLAB built-in functions **`max`** and **`min`** useful.

If everything works properly, you should see the bit error rate for both channel scenarios drop to zero.

Step 3: Submit your work

Remember to submit your completed script by clicking on the **Check** button.

Help