HKUSTx: ELEC1200.2x A System View of Communications: From Signals to...

- Pre-course Materials
- Topic 1: Course Overview
- Topic 2: Lossless Source Coding: Hamming Codes
- Topic 3: The Frequency Domain
- Topic 4: Lossy Source Coding
- Topic 5: Filters and the Frequency Response
- Topic 6: The Discrete Fourier Transform
- Topic 7: Signal Transmission - Modulation
- Topic 8: Signal Transmission - Demodulation
- **Topic 9: IQ**

LAB 5 - TASK 1 (3/3 points)

In this task, you will implement a communication system based on Binary Phase-Shift Keying (BPSK).

```
% initalization
Fs=1e6;                 %sampling frequency
Fc=1e5;                 %carrier frequency for message 1
Fcutoff = 25e3;         %cutoff frequency of low pass filter
Fc2=Fc + 2*Fcutoff;     %carrier frequency for message 2
SPB=18;                 % samples per bit
Ts = 1/Fs;              % sample period

textmsg1 = sprintf('%s\n%s\n%s',...
    'It was the best of times, it was the worst of times',...
    'it was the age of wisdom, it was the age of foolishness,',
    'it was the epoch of belief,');
textmsg2 = sprintf('%s\n%s\n%s',...
    'Lorem ipsum dolor sit amet, at vivamus erat lectus a augue
    'eget a diam aliquam consectetuer, vivamus ad wisi hac posu
    'praesent tincidunt vel.');
```

Correct

Figure 1

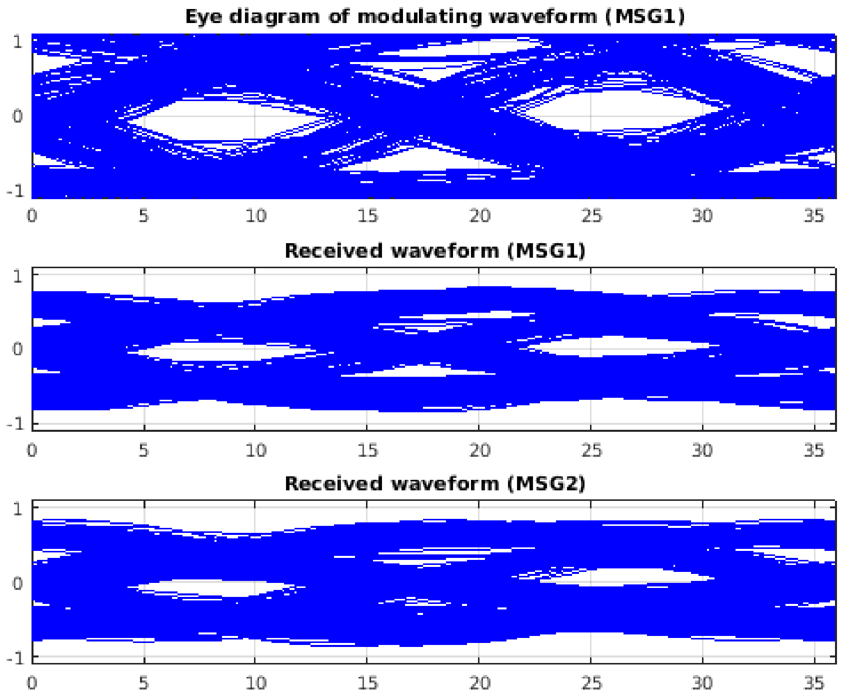


Figure 2

Modulation

9.1 Binary Phase Shift Keying
Week 5 Quiz due Nov 30, 2015 at 15:30 UTC

9.2 I/Q Modulation
Week 5 Quiz due Nov 30, 2015 at 15:30 UTC

9.3 Quadrature Phase Shift Keying
Week 5 Quiz due Nov 30, 2015 at 15:30 UTC

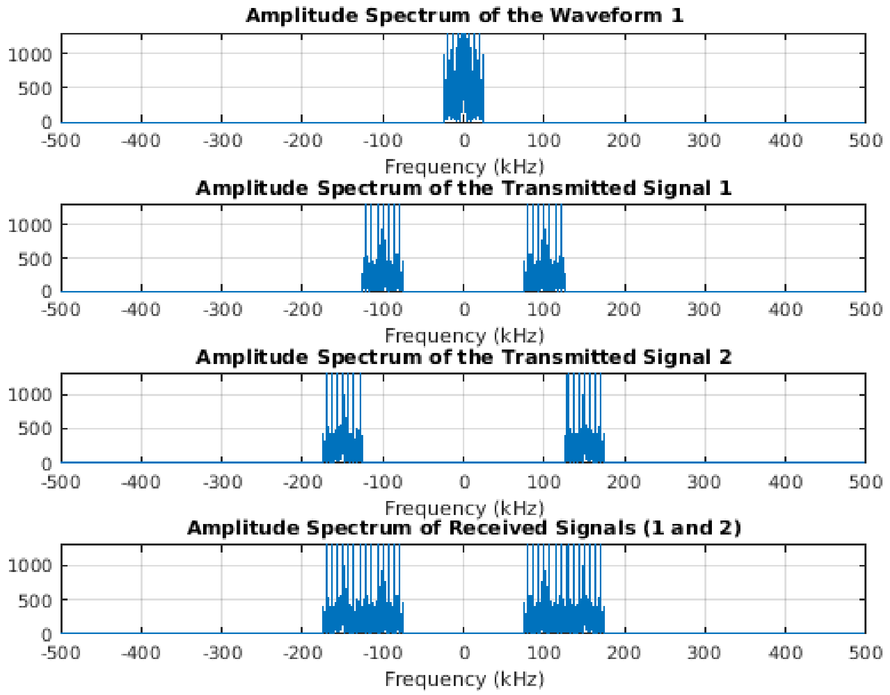
9.4 Constellation Diagrams
Week 5 Quiz due Nov 30, 2015 at 15:30 UTC

9.5 Lab 5 - BPSK and QPSK
Lab due Nov 30, 2015 at 15:30 UTC

Topic 10: Summary and Review

MATLAB download and tutorials

MATLAB Sandbox



```
Original Message  1:
It was the best of times, it was the worst of times
it was the age of wisdom, it was the age of foolishness,
it was the epoch of belief,
Recovered Message 1:
It was the best of times, it was the worst of times
it was the age of wisdom, it was the age of foolishness,
it was the epoch of belief,
Number of Bit Errors: 0

Original Message 2:
Lorem ipsum dolor sit amet, at vivamus erat lectus a augue,
eget a diam aliquam consectetuer, vivamus ad wisi hac posuere,
praesent tincidunt vel.
Recovered Message 2:
Lorem ipsum dolor sit amet, at vivamus erat lectus a augue,
eget a diam aliquam consectetuer, vivamus ad wisi hac posuere,
praesent tincidunt vel.
Number of Bit Errors: 0
```

You have used 1 of 10 submissions

INSTRUCTIONS

The MATLAB code in the above window shows an example of digital communication using binary phase-shift keying (**BPSK**), which is the simplest and more robust form of phase-shift keying. It uses waveforms with two phases separated by 180° to transmit 0 and 1 bits. If we use the two phases, 0 and π, to represent bits 0 and 1, then if the bit at time t is given by $d(t)$, then during one bit period, BPSK transmits

$$x(t) = \cos(2\pi F_c t + \pi[1 - d(t)])$$

where t represents time and F_c represents the carrier frequency.

Since $\cos(2\pi F_c t + \pi[1 - d(t)]) = [2d(t) - 1] \cdot \cos(2\pi F_c t)$, BPSK can also be implemented using amplitude modulation, where the signal $2d(t) - 1$ modulates the amplitude of a cosinusoidal carrier wave: $\cos(2\pi F_c t)$.

In the above example, there are two users, each transmitting a text message (**textmsg1** or **textmsg2**) using BPSK modulation. As in the previous lab, the users share the same channel using Frequency Division Multiplexing (FDM). We assign the frequency band **Fc±Fcutoff** kHz to the first user and the frequency band **Fc2±Fcutoff** kHz to the second user. Since the frequency bands do not overlap, there should be no interference between users. Thus, each user should be able to send and receive messages correctly.

If you run the initial code, you will observe that message 1 is recovered correctly, while message 2 is not. This is due to an error in the implementation of the BPSK transmitter for message 1. It is your task to correct this error.

To help you understand the source of this error, the code plots several figures. Figure 1 shows the eye diagrams of signals at three points in the communication system as three subplots. The first subplot shows the eye diagram of the waveform that modulates the cosinusoidal carrier with frequency **Fc** at the transmitter for message 1. This waveform is referred to as **wave1** in the code. This eye diagram is very open. Subplots 2 and 3 show the eye diagrams of the received waveform for messages 1 and 2 after demodulation by the carriers with frequencies **Fc** and **Fc2**, respectively. These waveforms are referred to as **wave_r1** and **wave_r2** in the code. The eye diagram of **wave_r1**, which encodes

textmsg1, is fairly open. However, the eye diagram of **wave_r2**, which encodes **textmsg2**, is closed. This suggests that there is some interference in the frequency band assigned to message 2.

Figure 2 plots the amplitude spectra of the waveforms at four points in the communication system as four subplots. The first subplot shows the amplitude spectrum of the modulating signal for message 1, **wave1**. The second subplot shows the amplitude spectrum of the transmitted signal for message 1, **tx_wave1**. The third subplot is the amplitude spectrum for the transmitted signal of message 2, **tx_wave2**. The final subplot is the amplitude spectrum for the received signal, **rx_wave**. We can observe from subplot 2 that the transmission for message 1 is outside of it assigned range of **Fc±Fcutoff** kHz (75-125 kHz). This is what is generating the interference seen in message 2. On the other hand, subplot 3 shows that the transmission for message 2 is contained within its assigned frequency band, so it generates no interference for other users.

In order to help you to fix the error, we describe the operation of the code in more detail below.

After initializing some variables, the code first implements the BPSK tranmitter for **textmsg1**. The first step is to convert the text message into a waveform. To do that, we first convert the text message into a binary sequence by using the function **text2bitseq**. Then, we frame the bit sequence, encode it as a discrete time waveform using **SPB** samples per bit, and add the training sequence using the function **format_bitseq**. These are the same functions that we studied in detail during Part 1 of this course. Finally, we multiply the waveform by 2 and we subtract 1 so that the waveform varies between -1 and +1, rather than between 0 and +1. This waveform is stored as **wave1** in the code. It makes quick transitions between the high and low values, which means that it contains much high frequency content, as can be observed initially in the amplitude spectrum of **wave1**. The waveform **wave1** then modulates a cosinusoidal carrier with frequency **Fc** in order to shift its spectrum so that it is centered around **Fc** for transmission. The modulated carrier is represented by the variable **tx_wave1** in the code. There is an error in this implementation of BPSK.

The signal transmitted by the second user, **tx_wave2**, is generated using a correct implementation of BPSK, which has been encapsulated into the function **tx_BPSK**. Then, the two modulated signals are summed together and are sent through an ideal channel (function **txrx**).

The received signal (**rx_wave**) is then demodulated with two carrier waves of frequencies **Fc** and **Fc2**, in order to recover the two text messages. The demodulation processes are the same for two messages. We show it explicitly for the first user, and use the function **rx_BPSK**, which encapsulates the same operations, for the second user. For this first user, the received waveform is first mixed with a copy of the carrier wave used during modulation. This copy must have the same frequency (**Fc** or **Fc2**), the same sampling rate **Fs**, and the same phase shift (0). The mixed signal is then low pass filtered by the function **lowpass**. This function takes three arguments **wave_r1** (the signal to be filtered), **Fs** (the sampling frequency), and **Fcutoff** (the cutoff frequency of the filter). This function may be useful in your modifications of the existing code. Finally, we convert the demodulated waveforms into a text message using the function **text2waveform**. This function is the same as that studied in more detail during Part 1 of this course.

The last part of the code is used to show the original and recovered messages, the number of erroneous bits, and some useful plots: the amplitude spectrum of some signal in Fig. 2 and the eye diagram in Fig. 1.

Your task is to correct the initial code in order to avoid the interference from message 1 to message 2. Please revise the code between the lines

% % % % Revise the following code % % % %

and

% % % % Do not change the code below % % % %

Do not change other parts of the code. Also, do not use the function **tx_BPSK**.