Homework Assignment #3

Baseband Signaling: Physical Signal Mapping

By

Zhening Li

Zuoda Ren

Zeqi Chen

Weimin Zhou

Faculty Advisor:

Dr. Paul S. Min (Electrical and System Engineering)

ESE471

Department of Electrical and System Engineering

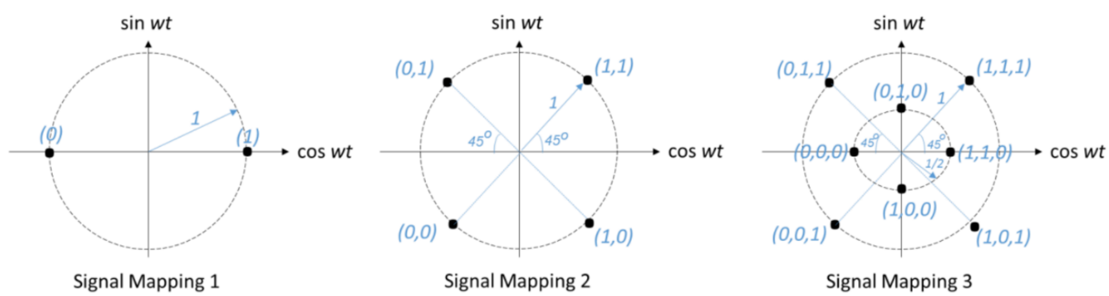
Washington University

Saint Louis, Missouri

Apr 10, 2016

Methods

In this assignment, our group tackled the “baseband signaling”. As professor have explained during the lecture, in the baseband signaling step, the channel-encoded bits are mapped to physical signals that can be transmitted into the channel. A popular way of mapping binary bits to the physical signals is based on what is known as the signal constellation. Three ways of which are shown below.



The binary bits and their corresponding physical signals.

BPSK Mapping	
(0)	$-\cos(\omega t)$
(1)	$\cos(\omega t)$

QPSK Mapping	
(1, 1)	$\frac{1}{\sqrt{2}}\cos(\omega t)+\frac{1}{\sqrt{2}}\sin(\omega t)$
(1, 0)	$\frac{1}{\sqrt{2}}\cos(\omega t)-\frac{1}{\sqrt{2}}\sin(\omega t)$
(0, 0)	$-\frac{1}{\sqrt{2}}\cos(\omega t)-\frac{1}{\sqrt{2}}\sin(\omega t)$
(0, 1)	$-\frac{1}{\sqrt{2}}\cos(\omega t)+\frac{1}{\sqrt{2}}\sin(\omega t)$

8QAM Mapping	
(1, 1, 1)	$\frac{1}{\sqrt{2}}\cos(\omega t)+\frac{1}{\sqrt{2}}\sin(\omega t)$
(1, 0, 1)	$\frac{1}{\sqrt{2}}\cos(\omega t)-\frac{1}{\sqrt{2}}\sin(\omega t)$
(0, 0, 1)	$-\frac{1}{\sqrt{2}}\cos(\omega t)-\frac{1}{\sqrt{2}}\sin(\omega t)$
(0, 1, 1)	$-\frac{1}{\sqrt{2}}\cos(\omega t)+\frac{1}{\sqrt{2}}\sin(\omega t)$
(1, 1, 0)	$\frac{1}{2}\cos(\omega t)$
(0, 1, 0)	$\frac{1}{2}\sin(\omega t)$
(1, 0, 0)	$-\frac{1}{2}\sin(\omega t)$
(0, 0, 0)	$-\frac{1}{2}\cos(\omega t)$

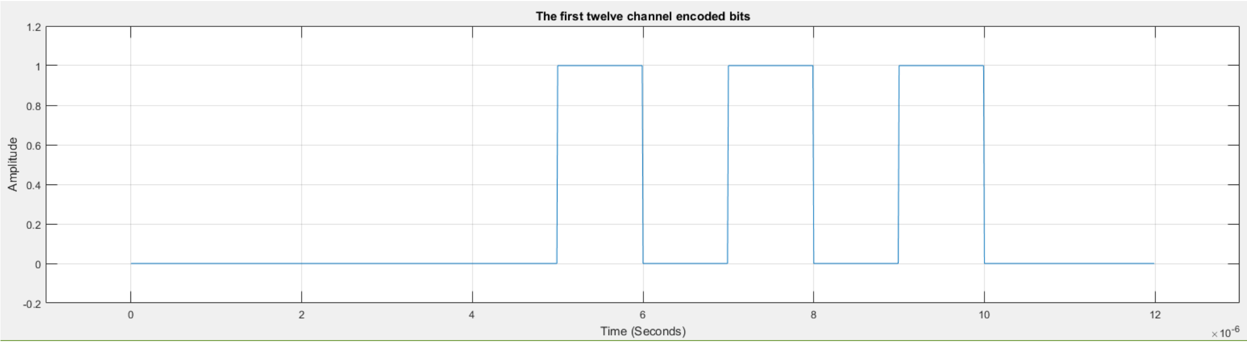
Our group use the NY Times article from the last two assignments and take the first twelve bits from the channel-encoded bit sequence which is [0 0 0 0 0 1 0 1 0 1 0 0]. We assume these twelve bits are to be transmitted in 12 microseconds. Then map these twelve bits to each of the signal mapping methods above. (For the parameter "ω" associated with the constellation use $\omega = 2 \times \pi \times 2{,}000{,}000$. That is the cosines and sines in all of the constellation has frequency of 2 megahertz.) At last, we sketch the resulting signal waveforms in time domain.

Also, to be more realistic and more obvious to their pros and cons, we add white Gaussian noise with SNR = 15 in the physical signaling.

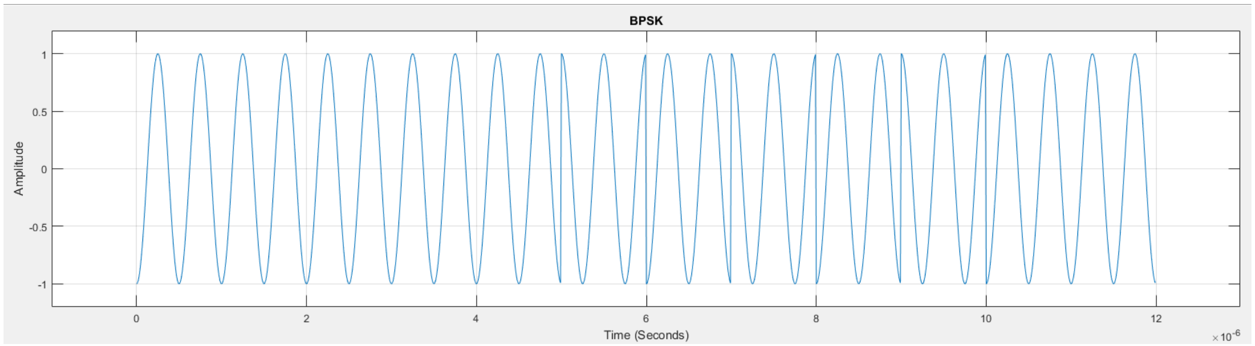
We implemented these steps in Matlab and the m file is attached at the end of this paper.

Results

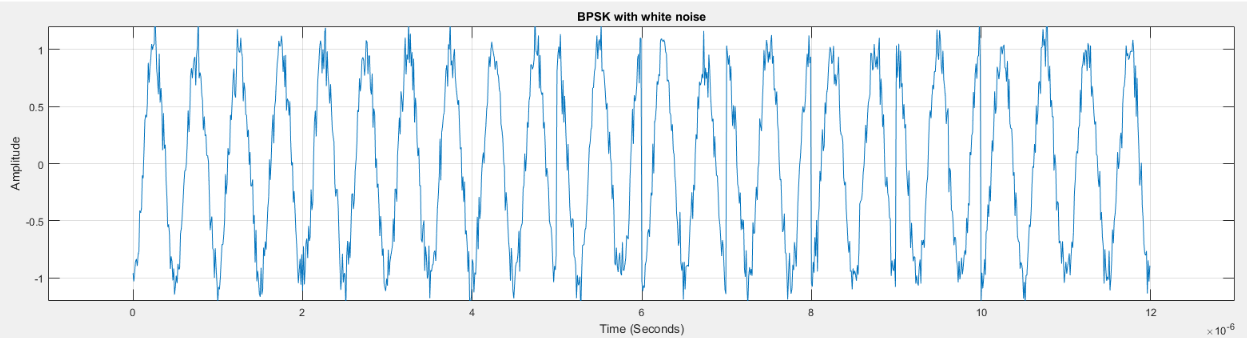
The first twelve bits from the channel-encoded bit sequence. [0 0 0 0 0 1 0 1 0 1 0 0]



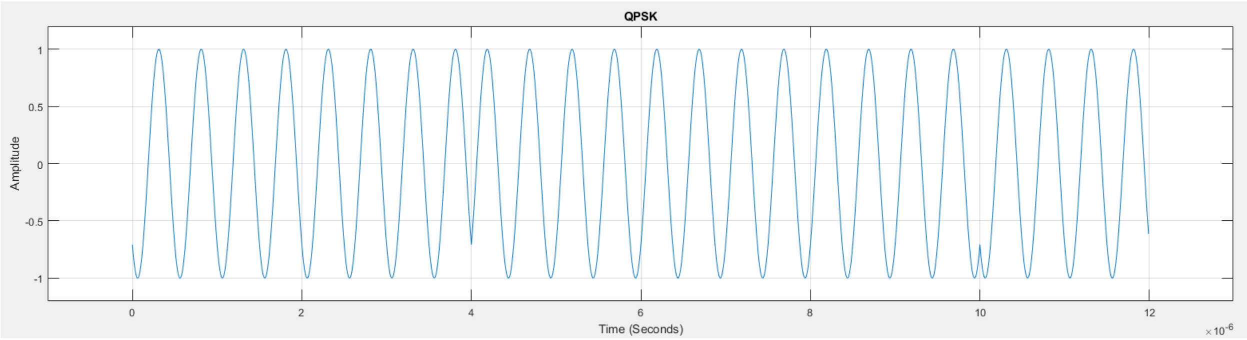
Signal waveforms in time use signal mapping 1 -- binary phase shift key (BPSK).



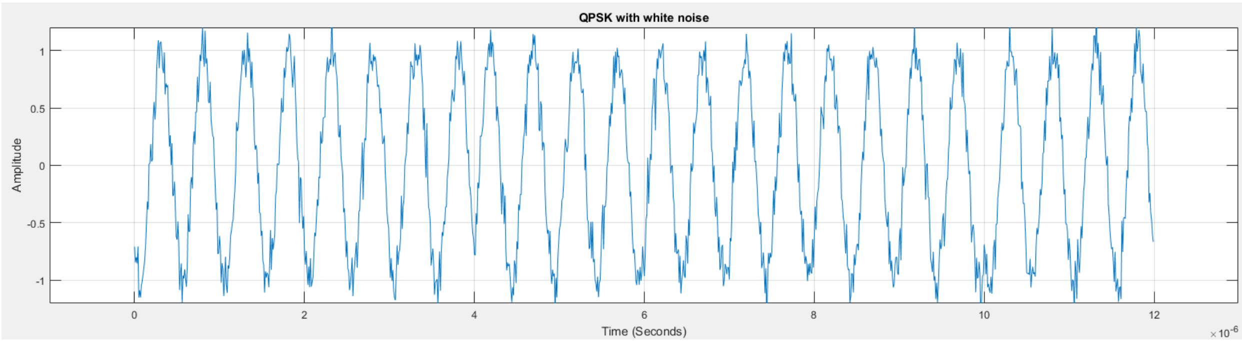
Signal waveforms with white noise in time use signal mapping 1 -- binary phase shift key (BPSK).



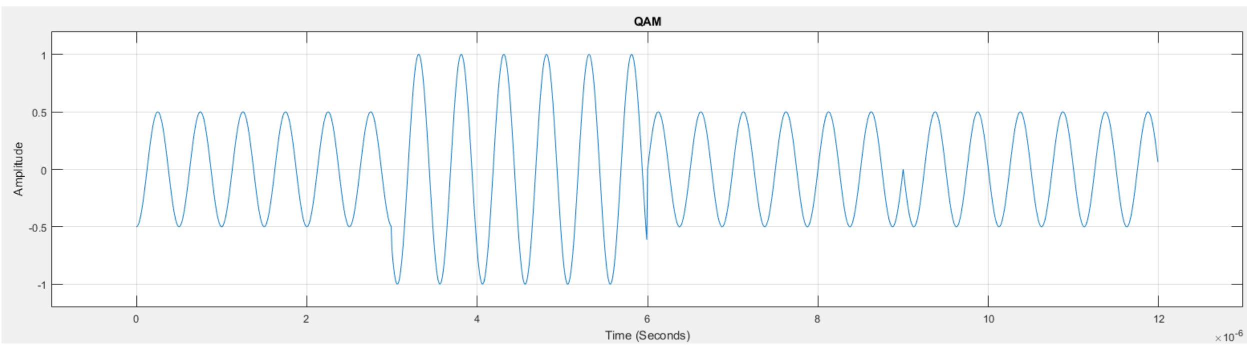
Signal waveforms in time use signal mapping 2 -- quadrature phase shift key (QPSK)



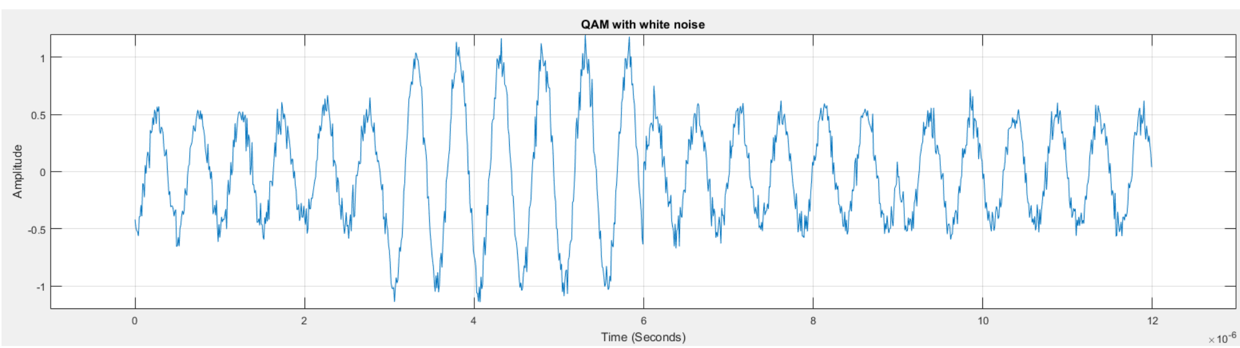
Signal waveforms with white noise in time use signal mapping 2 -- quadrature phase shift key (QPSK)



Signal waveforms in time use signal mapping 3 -- quadrature amplitude modulation (8QAM).



Signal waveforms with white noise in time use signal mapping 3 -- quadrature amplitude modulation (8QAM).



For the pros and cons, BPSK is simplest, most robust and more power efficiency. But it has low bandwidth efficiency. So, BPSK is usually used in low speed communications since it is only able to modulate at 1 bit/symbol. The advantage of QPSK modulation versus BPSK one is well known. It is the possibility to transmit in the same frequency band twice more information, while the number of errors and the SNR are the same. The disadvantage of QPSK relative to BPSK is that it is more sensitive to phase variations.

Attachment #1 Implementation in Matlab

```
%Author: Zhening Li     zhening.li@wustl.edu
%ESE 471  HW#3  Physical Signal Mapping
clc
clear
close all;
%%
uS = 1e-06;  %1us %twelve bits are to be transmitted in 12 microseconds
fc = 2e06;   %2 megahertz frequency %required physical frequency
Wc = 2*pi*fc; %cosines and sines in all of the constellation has frequency
of 2 megahertz.
SNR = 15; % signal to noise rate %For add white Gaussian noise
sample = [0 0 0 0 0 1 0 1 0 1 0 0];%the first twelve channel encoded bits
%
remat_times = 100; %repeat elements 100 times for plot
t = 0:uS/remat_times: 12*uS - uS/remat_times;  %Time Domain
%%
%original channel encoded bits
sampple_origin = rectpulse(sample,remat_times);
figure(1);plot(t,sampple_origin);
axis([-uS 13*uS-uS/remat_times -0.2 1.2])
title('The first twelve channel encoded bits')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
%%
%BPSK
t_BPSK = 0:uS/remat_times: 1*uS - uS/remat_times;
%time domain for BPSK per uS
for i = 1:12
    if(sample(i)==0)%'0' => -cos(wt)
        physical_signal_BPSK(1,i*100-99:i*100) = -cos(Wc*t_BPSK);
    end

    if(sample(i)==1)%'0' => cos(wt)
        physical_signal_BPSK(1,i*100-99:i*100) = cos(Wc*t_BPSK);
    end
end
figure(2);
plot(t,physical_signal_BPSK);
axis([-uS 13*uS-uS/remat_times -1.2 1.2])
title('BPSK')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
%add white noise
white_noise = awgn(physical_signal_BPSK,SNR,'measured');
figure(3);
plot(t,white_noise);
axis([-uS 13*uS-uS/remat_times -1.2 1.2])
title('BPSK with white noise')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
%%
%QPSK
```

```
t_QPSK = 0:uS/remat_times: 2*uS - uS/remat_times;
%time domain for QPSK per uS
for i = 1:2:12

      %'11' => 1/sqrt(2)*cos(wt)+1/sqrt(2)*sin(wt)
    if(sample(i)==1 && sample(i+1)==1)
        physical_signal_QPSK(1,(i-1)*100+1:(i-1)*100+200) =
                          1/sqrt(2)*cos(Wc*t_QPSK)+1/sqrt(2)*sin(Wc*t_QPSK);
    end

    %'10' => 1/sqrt(2)*cos(wt)-1/sqrt(2)*sin(wt)
    if(sample(i)==1 && sample(i+1)==0)
        physical_signal_QPSK(1,(i-1)*100+1:(i-1)*100+200) =
                          1/sqrt(2)*cos(Wc*t_QPSK)-1/sqrt(2)*sin(Wc*t_QPSK);
    end

      %'00' => -1/sqrt(2)*cos(wt)-1/sqrt(2)*sin(wt)
    if(sample(i)==0 && sample(i+1)==0)
        physical_signal_QPSK(1,(i-1)*100+1:(i-1)*100+200) =
                         -1/sqrt(2)*cos(Wc*t_QPSK)-1/sqrt(2)*sin(Wc*t_QPSK);
    end

    %'01' => -1/sqrt(2)*cos(wt)+1/sqrt(2)*sin(wt)
    if(sample(i)==0 && sample(i+1)==1)
        physical_signal_QPSK(1,(i-1)*100+1:(i-1)*100+200) =
                         -1/sqrt(2)*cos(Wc*t_QPSK)+1/sqrt(2)*sin(Wc*t_QPSK);
    end
end
figure(4);
plot(t,physical_signal_QPSK);
axis([-uS 13*uS-uS/remat_times -1.2 1.2])
title('QPSK')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
%add white noise
white_noise = awgn(physical_signal_QPSK,SNR,'measured');
figure(5);
plot(t,white_noise);
axis([-uS 13*uS-uS/remat_times -1.2 1.2])
title('QPSK with white noise')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
%%
%QAM
t_QAM = 0:uS/remat_times: 3*uS - uS/remat_times; %time domain for QAM per uS
for i = 1:3:12

      %'111' => 1/sqrt(2)*cos(wt)+1/sqrt(2)*sin(wt)
    if(sample(i)==1 && sample(i+1)==1 && sample(i+2)==1)
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                            1/sqrt(2)*cos(Wc*t_QAM)+1/sqrt(2)*sin(Wc*t_QAM);
    end

      %'101' => 1/sqrt(2)*cos(wt)-1/sqrt(2)*sin(wt)
    if(sample(i)==1 && sample(i+1)==0 && sample(i+2)==1)
```

```
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                                    1/sqrt(2)*cos(Wc*t_QAM)-1/sqrt(2)*sin(Wc*t_QAM);
    end

    %'001' => -1/sqrt(2)*cos(wt)-1/sqrt(2)*sin(wt)
    if(sample(i)==0 && sample(i+1)==0 && sample(i+2)==1)
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                                   -1/sqrt(2)*cos(Wc*t_QAM)-1/sqrt(2)*sin(Wc*t_QAM);
    end

    %'011' => -1/sqrt(2)*cos(wt)+1/sqrt(2)*sin(wt)
    if(sample(i)==0 && sample(i+1)==1 && sample(i+2)==1)
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                                   -1/sqrt(2)*cos(Wc*t_QAM)+1/sqrt(2)*sin(Wc*t_QAM);
    end

    %'110' => 1/2*cos(wt)
    if(sample(i)==1 && sample(i+1)==1 && sample(i+2)==0)
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                                                              1/2*cos(Wc*t_QAM);
    end

    %'010' => 1/2*sin(wt)
    if(sample(i)==0 && sample(i+1)==1 && sample(i+2)==0)
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                                                              1/2*sin(Wc*t_QAM);
    end

    %'100' => -1/2*sin(wt)
    if(sample(i)==1 && sample(i+1)==0 && sample(i+2)==0)
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                                                             -1/2*sin(Wc*t_QAM);
    end

    %'000' => -1/2*cos(wt)
    if(sample(i)==0 && sample(i+1)==0 && sample(i+2)==0)
        physical_signal_QAM(1,(i-1)*100+1:(i-1)*100+300) =
                                                             -1/2*cos(Wc*t_QAM);
    end
 end
figure(6);
plot(t,physical_signal_QAM);
axis([-uS 13*uS-uS/remat_times -1.2 1.2])
title('QAM')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
%add white noise
white_noise = awgn(physical_signal_QAM,SNR,'measured');
figure(7);
plot(t,white_noise);
axis([-uS 13*uS-uS/remat_times -1.2 1.2])
title('QAM with white noise')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
```