Homework Assignment #4

Receiver Design

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

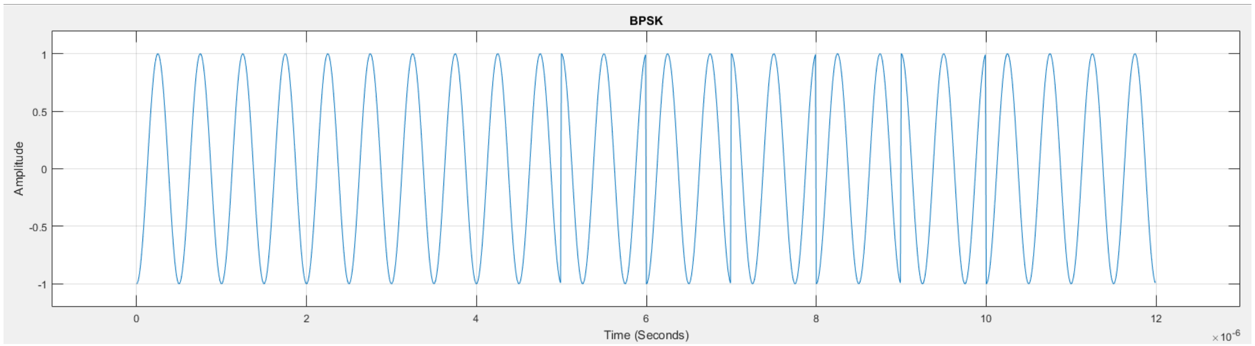
Saint Louis, Missouri

Apr 20, 2016

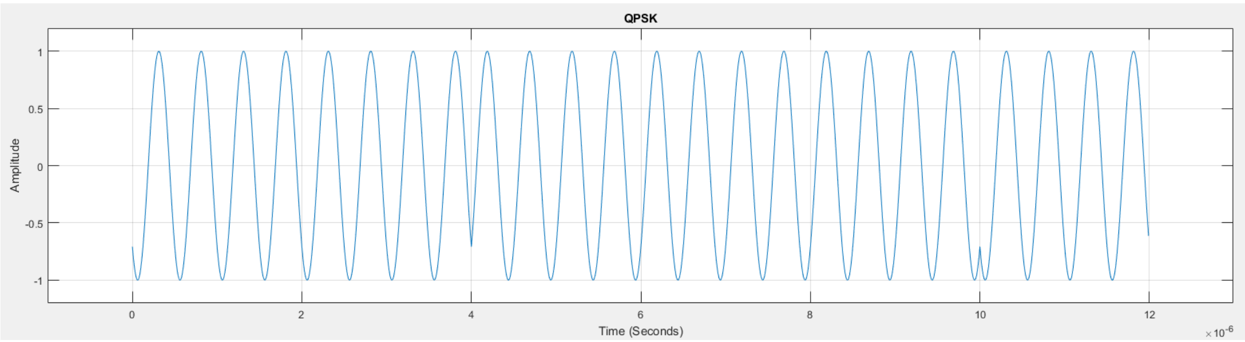
Methods

In assignment 3, our group mapped the channel encoded bit sequence into three different constellations. Each of the three mappings resulted in a 12 microsecond-worth of physical signal comprising sine and cosine waves at 2 million Hz. Now that these 12 microsecond-worth of physical signals were received at the receiver. In this assignment, we assume that the physical signals arrive at the receiver as they were sent by the transmitter, without noise and without attenuation

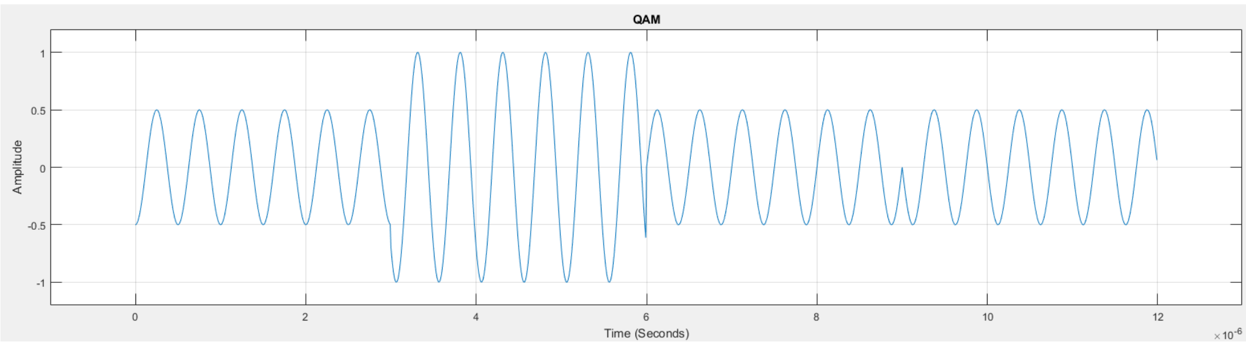
Signal waveforms r(t) in time use signal mapping 1 -- Binary Phase Shift Key (BPSK).



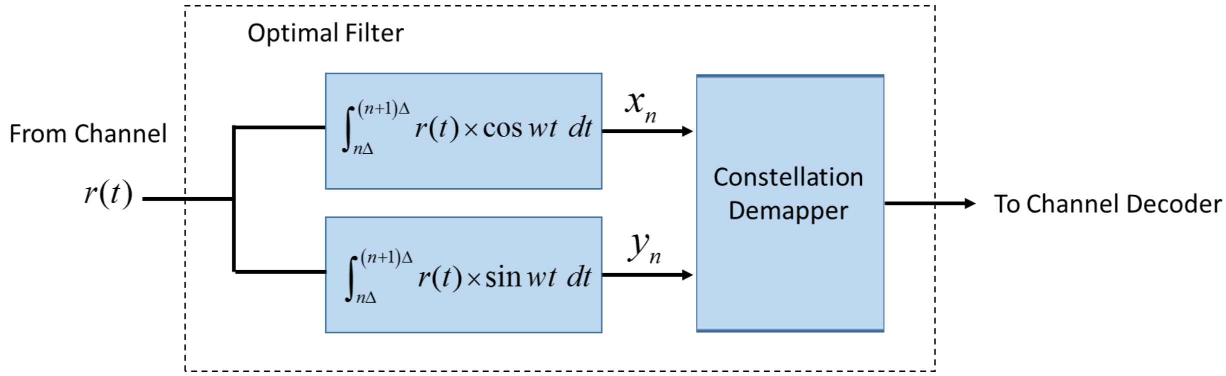
Signal waveforms r(t) in time use signal mapping 2 -- Quadrature Phase Shift Ley (QPSK)



Signal waveforms r(t) in time use signal mapping 3 -- Quadrature Amplitude Modulation (8QAM).



An optimal filter is a filter that extracts the binary signals (i.e., 1s and 0s) from the received physical signals (sines and cosines) in an optimal way. The following corresponds to optimal filter for the signal mapping methods used in my assignment 3.



r (t) is the physical signal received from the channel. For signal mapping method 1 of assignment 3, $n = 0, 1, 2, \ldots, 11$ and $\Delta = 1$ microsecond. In the constellation decapper, the results of the integration, i.e., x_n and y_n are compared with the positions in the BPSK constellation, and determine the position (out of the two in the constellation), which is the closest. Out of the constellation decapper is the channel coded bits to be send to the channel decoder. For signal mapping method 2, $n = 0, 1, 2, \ldots,5$ and $\Delta = 2$ microseconds, and the QPSK constellation is used. For signal mapping method 3, $n = 0, 1, 2, 3,$ and $\Delta = 3$ microseconds, and the 8QAM constellation is used.

Results

For all these three constellation decipher, we get same output and they are same with the channel encoded bit sequence we use in assignment 3 which means our algorithm works perfectly.

Channel_Decoder_BPSK

1x12 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	1	0	1	0	1	0	0

Channel_Decoder_QPSK

1x12 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	1	0	1	0	1	0	0

Channel_Decoder_QAM

1x12 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	1	0	1	0	1	0	0

Attachment: Implementation in Matlab

```
clc
clear
close all;
%%
uS = 1e-06;  %1us %twelve bits are to be transmitted in 12 microseconds
fc = 2e06;   %2 mega hertz frequency %required physical frequency
Wc = 2*pi*fc; %cosines and sines in all of the constellation has frequency
of 2 mega hertz.
remat_times = 100; %repeat elements 100 times for plot
%%
[BPSK,QPSK,QAM]=hw_3();%get the physical signals of BPSK,QPSK,QAM from HW#3
%%
%BPSK
t_BPSK = 0:uS/remat_times: 1*uS - uS/remat_times;%set the time domin delta
t_delta = remat_times;%set the delta for BPSK signal vector
%for the discrete time solution in Matlab, instead of doing integral, we do
sum
for i = 1:12
    sign = sum(BPSK(i*t_delta-99:i*t_delta).*cos(Wc*t_BPSK));
    if(sign>0)
        Channel_Decoder_BPSK(i) = 1;
    elseif(sign<0)
        Channel_Decoder_BPSK(i) = 0;
    end
end
%%
%QPSK
t_QPSK = 0:uS/remat_times: 2*uS - uS/remat_times;%set the time domin delta
t_delta = 2*remat_times;%set the delta for QPSK signal vector
%for the discrete time solution in Matlab, instead of doing integral, we do
sum
for i = 1:6
    sign_1 = sum(QPSK(i*t_delta-199:i*t_delta).*cos(Wc*t_QPSK));
    sign_2 = sum(QPSK(i*t_delta-199:i*t_delta).*sin(Wc*t_QPSK));
```

```
    if(sign_1>0&&sign_2>0)
        Channel_Decoder_QPSK(2*i-1) = 1;
        Channel_Decoder_QPSK(2*i) = 1;
    elseif(sign_1>0&&sign_2<0)
        Channel_Decoder_QPSK(2*i-1) = 1;
        Channel_Decoder_QPSK(2*i) = 0;
    elseif(sign_1<0&&sign_2>0)
        Channel_Decoder_QPSK(2*i-1) = 0;
        Channel_Decoder_QPSK(2*i) = 1;
    elseif(sign_1<0&&sign_2<0)
        Channel_Decoder_QPSK(2*i-1) = 0;
        Channel_Decoder_QPSK(2*i) = 0;
    end
end
%%
%QAM
t_QAM = 0:uS/remat_times: 3*uS - uS/remat_times;%set the time domin delta
t_delta = 3*remat_times;%set the delta for QAM signal vector
%for the discrete time solution in Matlab, instead of doing integral, we do
sum
for i = 1:4
    sign_1 = sum(QAM(i*t_delta-299:i*t_delta).*cos(Wc*t_QAM));
    sign_2 = sum(QAM(i*t_delta-299:i*t_delta).*sin(Wc*t_QAM));
    Amplitude = sqrt(sign_1^2+sign_2^2); %for QAM, we need to take amplitude
into consideration
   if (Amplitude < 150)
    if(sign_1<0&&sign_2>0&&sign_2<1)
        Channel_Decoder_QAM(3*i-2) = 0;
        Channel_Decoder_QAM(3*i-1) = 0;
        Channel_Decoder_QAM(3*i) = 0;
    elseif(sign_1>-1&&sign_1<0&&sign_2>0)
        Channel_Decoder_QAM(3*i-2) = 0;
        Channel_Decoder_QAM(3*i-1) = 1;
        Channel_Decoder_QAM(3*i) = 0;
    elseif(sign_1>0&&sign_1<1&&sign_2<0)
        Channel_Decoder_QAM(3*i-2) = 1;
        Channel_Decoder_QAM(3*i-1) = 0;
        Channel_Decoder_QAM(3*i) = 0;
    elseif(sign_1>0&&sign_2<0&&sign_2>-1)
        Channel_Decoder_QAM(3*i-2) = 1;
        Channel_Decoder_QAM(3*i-1) = 1;
        Channel_Decoder_QAM(3*i) = 0;
    end
   elseif (Amplitude > 75)
    if(sign_1>0&&sign_2>0)
        Channel_Decoder_QAM(3*i-2) = 1;
        Channel_Decoder_QAM(3*i-1) = 1;
        Channel_Decoder_QAM(3*i) = 1;
    elseif(sign_1>0&&sign_2<0)
        Channel_Decoder_QAM(3*i-2) = 1;
        Channel_Decoder_QAM(3*i-1) = 0;
        Channel_Decoder_QAM(3*i) = 1;
    elseif(sign_1<0&&sign_2>0)
        Channel_Decoder_QAM(3*i-2) = 0;
        Channel_Decoder_QAM(3*i-1) = 1;
        Channel_Decoder_QAM(3*i) = 1;
    elseif(sign_1<0&&sign_2<0)
```

```
        Channel_Decoder_QAM(3*i-2) = 0;
        Channel_Decoder_QAM(3*i-1) = 0;
        Channel_Decoder_QAM(3*i) = 1;
    end
   end
end
```