ECE 414 Project

Question 1

e.)

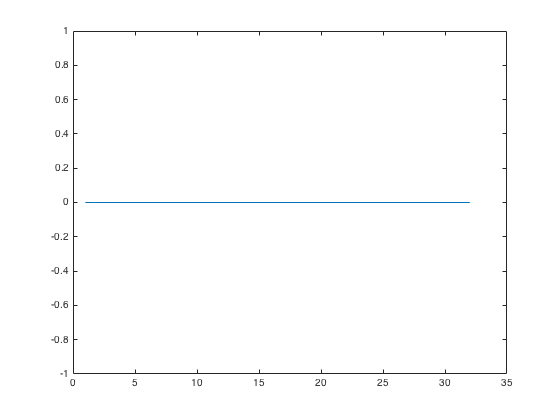


Figure . Graph of X(t) for infinite time duration

In the case where Ts is infinite. X(t) is simply the sum of all 4QAM modulated symbols Xk. The result of this summation is 0 for all time intervals t.

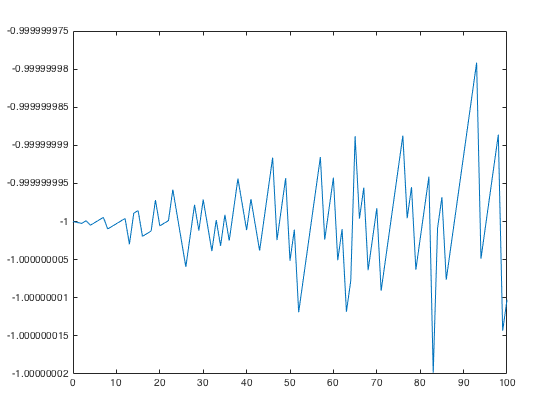


Figure . Graph of X(t) for finite time duration Ts=4µs

The graph shows the response of X(t). X(t) is the discrete Fourier transform of the 4QAM modulated sequence A0.

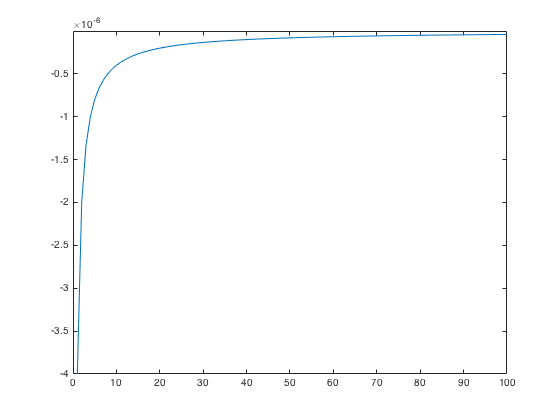


Figure . Graph of X(f) for finite Ts= 4µs

Question 2

1. Adding a cyclic prefix to the symbol makes the linear convolution appear as if it were a circular convolution. For large enough N the cyclic prefix will eliminate inter-symbol interference from the previous symbol.

Question 3

The probability of error was calculated to be 73.125%. This high probability of error may be as a result of the size of the Zero-Forcing matrix used to reconstruct the transmitted signal. Sending the first symbol A0 gave a probability of error of 75%, this means the receiver estimated 24 symbols incorrectly. Thus, the receiver was not successfully constructed to properly retrieve the transmitted symbols

**Project Code**

% ECE 414 Project

% By: Henrietta Odiete

% Student number: 20482444

% Instructor: Professor Guang Gong

clc

clear

close all

% Question 1a

% converts the bit sequence to a decimal number

% A0 even and odd bits

A0\_evenbits=[0 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 1 0 1];

A0\_oddbits= [0 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0];

A0=zeros(1,32);

for b=1:32

for k=1:32

if(A0\_evenbits(b)== 0 && A0\_oddbits(b)==0)

A0(b)=0;

end

if(A0\_evenbits(b)== 0 && A0\_oddbits(b)==1)

A0(b)=1;

end

if(A0\_evenbits(b)== 1 && A0\_oddbits(b)==0)

A0(b)=2;

end

if(A0\_evenbits(b)== 1 && A0\_oddbits(b)==1)

A0(b)=3;

end

end

end

% To generate other symbols A1-A10 circular left shift was applied to A0

% used to compare output at reciver from the transmitter

A1=circshift(A0,[1 -1]);

A2=circshift(A0,[1 -2]);

A3=circshift(A0,[1 -3]);

A4=circshift(A0,[1 -4]);

A5=circshift(A0,[1 -5]);

A6=circshift(A0,[1 -6]);

A7=circshift(A0,[1 -7]);

A8=circshift(A0,[1 -8]);

A9=circshift(A0,[1 -9]);

% modulates the sequence using 4QAM

Xk= zeros(1,32);

for f= 1:32

Xk(f)= qammod(A0(f),4);

end

%Ouestion 1b

% performs inverse fast fourier transform to get xn

xn= ifft(Xk);

% Question 1c

% adding cyclic prefix to the OFDM symbol xn

% gets last eight symbols and adds it to the start

cyclic\_xn=[xn(25:end) xn];

% Question 1d

% for other symbols cyclic prefix A1-A9

% circular shift of cyclic\_xn to generate other OFDM symbols

cyclic\_xn1=circshift(cyclic\_xn, [1 -1]);

cyclic\_xn2=circshift(cyclic\_xn, [1 -2]);

cyclic\_xn3=circshift(cyclic\_xn, [1 -3]);

cyclic\_xn4=circshift(cyclic\_xn, [1 -4]);

cyclic\_xn5=circshift(cyclic\_xn, [1 -5]);

cyclic\_xn6=circshift(cyclic\_xn, [1 -6]);

cyclic\_xn7=circshift(cyclic\_xn, [1 -7]);

cyclic\_xn8=circshift(cyclic\_xn, [1 -8]);

cyclic\_xn9=circshift(cyclic\_xn, [1 -9]);

% Question 1e

% Case 1: The time duration Ts is infinite

s=zeros(1,32);

s(1:32)=sum(Xk);

q=1:32;

% plots the response for infinte Ts

% plot(q,s);

% Case 2:The time duration Ts is finite

% OFDM transmit time Ts= 4e-6

Xt = 0;

Ts = 4\*10^(-6); % Ts represents OFDM signal transmit time

t=0:100;

for r = 1:32:32

Xt = Xt + Xk(r)\*exp(2\*1j\*t\*pi/Ts);

end

% plots the response for finite Ts

plot(t,Xt);

% for X(f): spectrum of X(t) when Ts is not infinite

% ?f=k/Ts

Xf=0;

for k = 1:32:32

Xf = Xf + Xk(k)\*sinc(Ts\*(t.^-1-(k\*k/Ts)));

end

%plots the frequency spectrum X(f)

figure();plot (t,Xf);

% impulse response of channel

% used to form ZF matrix

hn= [1 0.5 0.6 0.2 0.1 0 0 0 0 0 0 0 0 0 0];

% Question 3

% Ouestion 3a- received signal yn

yn= conv(hn,cyclic\_xn);

% Question 3b- spectrum of received signal Yk

yk= fft(yn);

% Question 3c- Recover Xk

% creates zero forcing matrix from the channel hn

% for A0 symbol

mn= tril(toeplitz(hn));

v= [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]';

bn= (mn\v)';

A0\_recov=cconv(yn,bn,32);

A0\_recov=fft(A0\_recov);

% Question 3e- Demodulate Xk to get data

A0\_recov=qamdemod(A0\_recov,4);

% Question 3d- Convert to serial stream

% for A1 symbol

yn1= conv(hn,cyclic\_xn1);

yk1= fft(yn1);

A1\_recov=cconv(yk1,bn,32);

A1\_recov=qamdemod(A1\_recov,4);

% for A2 symbol

yn2= conv(hn,cyclic\_xn2);

yk2= fft(yn2);

A2\_recov=cconv(yk2,bn,32);

A2\_recov=qamdemod(A2\_recov,4);

% for A3 symbol

yn3= conv(hn,cyclic\_xn3);

yk3= fft(yn3);

A3\_recov=cconv(yk3,bn,32);

A3\_recov=qamdemod(A3\_recov,4);

% for A4 symbol

yn4= conv(hn,cyclic\_xn4);

yk4= fft(yn4);

A4\_recov=cconv(yk4,bn,32);

A4\_recov=qamdemod(A4\_recov,4);

% for A5 symbol

yn5= conv(hn,cyclic\_xn5);

yk5= fft(yn5);

A5\_recov=cconv(yk5,bn,32);

A5\_recov=qamdemod(A5\_recov,4);

% for A6 symbol

yn6= conv(hn,cyclic\_xn6);

yk6= fft(yn6);

A6\_recov=cconv(yk6,bn,32);

A6\_recov=qamdemod(A6\_recov,4);

% for A7 symbol

yn7= conv(hn,cyclic\_xn7);

yk7= fft(yn7);

A7\_recov=cconv(yk7,bn,32);

A7\_recov=qamdemod(A7\_recov,4);

% for A8 symbol

yn8= conv(hn,cyclic\_xn8);

yk8= fft(yn8);

A8\_recov=cconv(yk8,bn,32);

A8\_recov=qamdemod(A8\_recov,4);

% for A9 symbol

yn9= conv(hn,cyclic\_xn9);

yk9= fft(yn9);

A9\_recov=cconv(yk9,bn,32);

A9\_recov=qamdemod(A9\_recov,4);

% Question 3f Probablity of error

number\_symbols=320; % total number of decimal symbols sent

symbol\_error=zeros(1,10); %number of error for each sequence A0-A9

for d=1:32

if A0(d)~=A0\_recov(d)

symbol\_error(1)=symbol\_error(1)+1;

end

end

for d=1:32

if A1(d)~=A1\_recov(d)

symbol\_error(2)=symbol\_error(2)+1;

end

end

for d=1:32

if A2(d)~=A2\_recov(d)

symbol\_error(3)=symbol\_error(3)+1;

end

end

for d=1:32

if A3(d)~=A3\_recov(d)

symbol\_error(4)=symbol\_error(4)+1;

end

end

for d=1:32

if A4(d)~=A4\_recov(d)

symbol\_error(5)=symbol\_error(5)+1;

end

end

for d=1:32

if A5(d)~=A5\_recov(d)

symbol\_error(6)=symbol\_error(6)+1;

end

end

for d=1:32

if A6(d)~=A6\_recov(d)

symbol\_error(7)=symbol\_error(7)+1;

end

end

for d=1:32

if A7(d)~=A7\_recov(d)

symbol\_error(8)=symbol\_error(8)+1;

end

end

for d=1:32

if A8(d)~=A8\_recov(d)

symbol\_error(9)=symbol\_error(9)+1;

end

end

for d=1:32

if A9(d)~=A9\_recov(d)

symbol\_error(10)=symbol\_error(10)+1;

end

end

P\_e= 100\*sum(symbol\_error)/(number\_symbols);