Kyle Savidge

EC508 – Mini Project #1

10/10/16

a)

function [ ri ] = pathlengths( t, reflectMat )

%Calculate a vector of pathlengths, between the origin, a given reflector,

%and the reciever

% We pass in the time and a matrix of reflector positions.

% The last row in the output vector will be the direct Rx to Tx distance.

% We are only considering 1 reflection in any given path length, operating under

% the assumption that the amplitude from multiple reflections is sufficiently small

%initialize the velocity vector, Rx initial coordinates, reflector

%positions

v = [2, 2];

dRx0 = [100, 100];

di = reflectMat;

%calculate the new position of the Rx based on elapsed time

dRx = (t \* v) + dRx0;

%generate vector of zeros to use pdist2 function between all reflectors and

%the Tx

origin = zeros(1,2);

%calculate distances between Rx and each reflector, and Tx and each

%reflector. Sum the distances for a given reflector

dRx2di = transpose(pdist2(dRx,di));

ddi2Tx = pdist2(di,origin);

ri = dRx2di + ddi2Tx;

%add in a row at the bottom of the matrix for direct path:

directPath = pdist2(dRx, origin);

ri = [ri;directPath];

end

b)

function [ ai, ti ] = attenuationanddelay( ri )

%Calculate a vector of attenuations and delays, with a row in the vector

%constituing a single path

% We input the pathlength distances calculated using the pathlengths.m

% function

%initialize Tx and Rx antenna gains, carrier frequency, and speed of light

Gtx = 1;

Grx = 1;

Fc = .16;

c = 3 \* 10^8;

%calculate ai for all pathlengths

aidenom = (4\*pi\*ri);

ainum = sqrt(Gtx\*Grx)\*Fc;

ai = ainum./aidenom;

%calculate ti for all pathlengths

ti = ri/c;

end

c)

function [ H , freq ] = powerspectrum( ai, ti )

%Calculate the power distribution across frequencies 0 to 2E9 hz using

%the equation for channel frequency response

freq = zeros(1000,1);

partialH = zeros(size(ai,1),1);

H = freq;

for freqindex = 1:1000

freq(freqindex) = 2000000\*freqindex;

for i = 1:size(ai)

partialH(i) = (ai(i)\*exp(-1i\*2\*pi\*freq(freqindex)\*ti(i)));

end

H(freqindex) = 20\*log10(abs(sum(partialH)));

end

plot(freq,H)

title('Channel power spectrum at time t=0')

xlabel('Frequency')

ylabel('Power gain in dB')

end

Continuous channel impulse response:

Continuous channel frequency response:

d) Discrete-time baseband impulse response:

function [ h ] = discreteimpulse( ai, ti )

%Calculate the impulse response for the channel at m=0, using the formula

%for discrete time. W = 1 MHz, and fc = 1.9 GHz. Trial and error to

%generate the necessary number of taps without having too many that are

%essentially zero.

%initialize necessary variables

fc = 1.9 \* 10 ^ 9;

W = 1 \*10 ^ 6;

%loop over all values of ai and ti

partialh = zeros(size(ai,1),1);

l = zeros(21,1);

h = l;

for j = 1:size(l,1)

l(j) = j-1;

for i = 1:size(ai)

partialh(i) = ai(i)\*exp(-1i\*2\*pi\*fc\*ti(i))\*sinc(l(j) - (ti(i) \* W));

end

h(j) = abs(sum(partialh));

end

stem(l,h)

title('Absolute value of discrete impulse response at m=0')

xlabel('l')

ylabel('abs(h\_l(m))')

end

e)

function [ T\_d , W\_c ] = delayspread( ri )

%T\_d is the delay spread, which is calculated by the maximum difference of

%path lengths, divided by the speed of light. W\_c is the coherence

%bandwidth, and is a simple calculation of 1/(2\*T\_d)

T\_d = (max(ri)-min(ri))/(3 \* 10^8);

W\_c = 1/(2\*T\_d);

end

f)

Appended this code to the existing for pathlengths. Had some issues running after it had run the first time.

%perform Doppler spread calculation

%F is distance as a function of time

Tau\_prime = zeros(size(di, 1),1);

syms time Tau Diffed;

for i = 1:size(di,1)

Tau(time) = (1/c)\*sqrt((dRx0(1,1)+v(1,1)\*time-di(i,1))^2 + (dRx0(1,2)+v(1,2)\*time-di(i,2))^2);

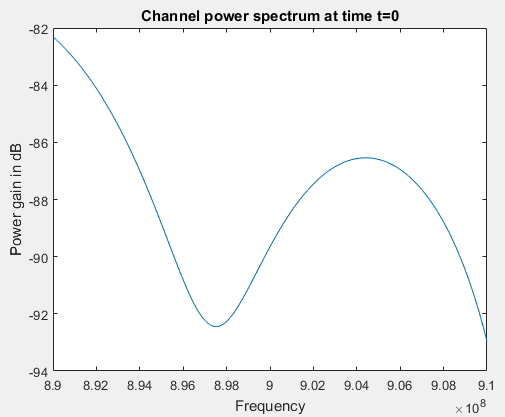
Diffed = diff(Tau,time);

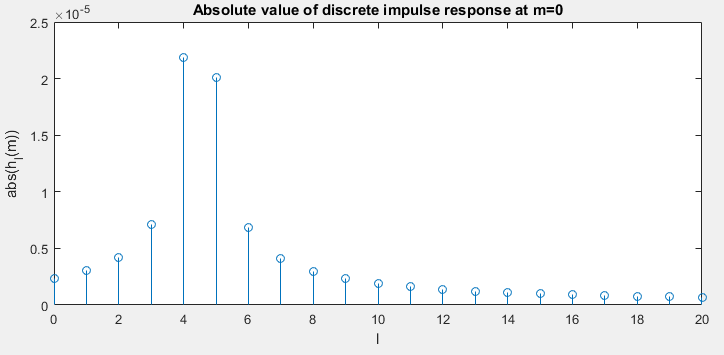
Tau\_prime(i) = Diffed(0);

end

D\_s = (max(Tau\_prime) - min(Tau\_prime))\*Fc;

h) Scenario 1:

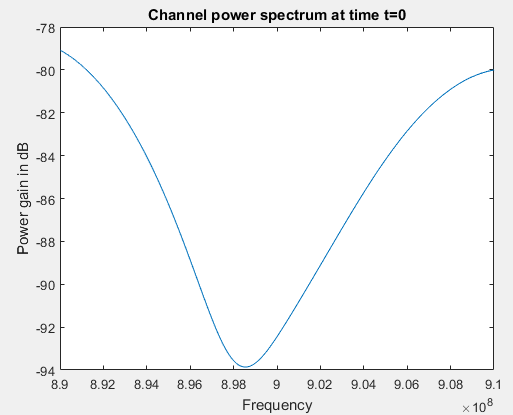


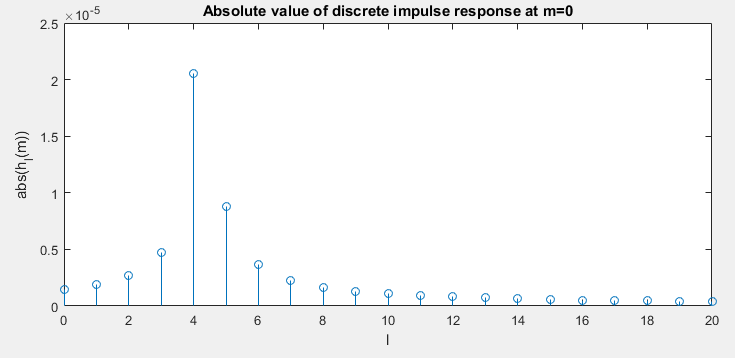


Delay spread: 5.6297e-08; Coherence Bandwidth: 8.8814e+06

Doppler spread: 0.0131; Coherence time: 19.1161

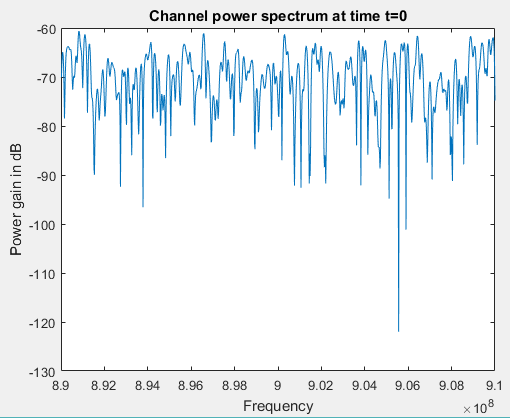
Scenario 2:

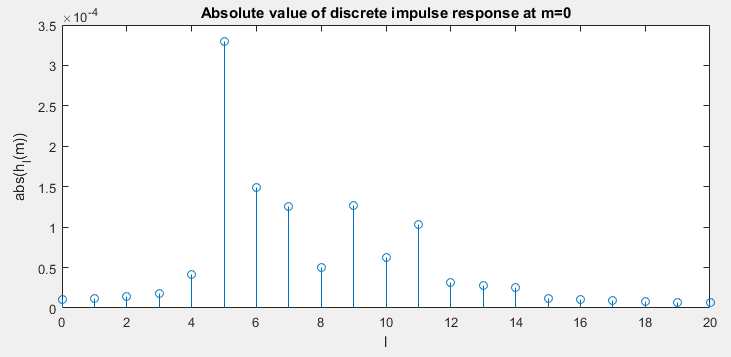




Delay spread: 8.1408e-08; Coherence Bandwidth: 6.1419e+06

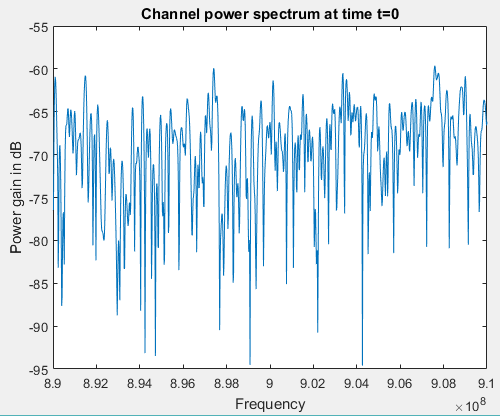
Scenario 3:

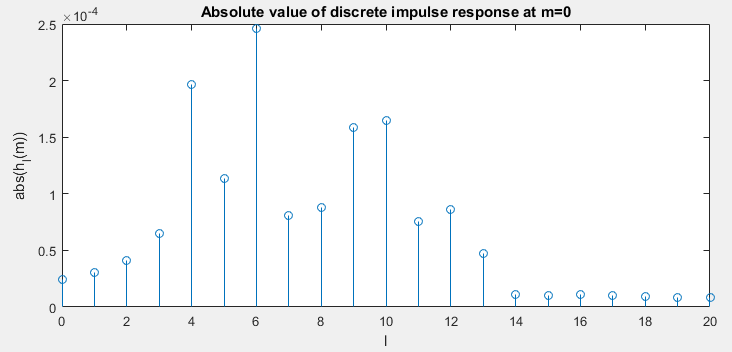




Delay spread: 9.7372e-06; Coherence Bandwidth: 5.1350e+04

Scenario 4:





Delay spread: 1.0585e-05; Coherence Bandwidth: 4.7235e+04