**Digital Communication Lab**

**E.C.E. DEPARTMENT**

**NIT CALICUT**

CDMA SOFTWARE



**Submitted by-**

D Goutham Varma-B120608EC

P D Anusha – B130204EC

Y Sreenivasulu Reddy- B130867EC

Y Vikas Yadav – B130852EC

**Aim:**

Simulate Spread Spectrum Communication system and implement frequency spreading and de - spreading followed by BPSK modulation and compare the BER performance in each case in the presence of Rayleigh fading channel and AWGN noise respectively.

**Objectives:**

Study of frequency spreading and de - spreading.

Generation of maximal length pseudo noise (PN) sequences of various lengths. Study and implementation of Gold codes.

Comparison of auto correlation and cross correlation properties of the various types of sequences. Generation of Walsh code and plotting the auto correlation and cross correlation of the sequences.

Implementing frequency spreading and de - spreading of data stream an perform BPSK modulation in the presence of Rayleigh fading channel and AWGN channel

Plot the BER performance and compare with the plots obtained without spreading and de - spreading.

**Principles and Theory:**

**Spreading sequences:**

In spread spectrum communications, typically in CDMA, the user data is multiplied with a spreading sequence to achieve spreading . When the signal is received, the spreading is removed from the desired signal by multiplying it by the same sequence that is exactly synchronized to the transmitted PN signal.When a despreading operation is applied to the interferers signals, ideally there is no further contribution to the user of interests signal level. In CDMA, each user is assigned a predetermined spreading sequence which has low cross correlation property with other users spreading sequences.

Spreading sequences are chosen based on their characteristics like autocorrelation, cross correlation properties etc. Some of the spreading sequences are:

Maximum length Pseudo Noise (PN) Sequence Gold Sequences

Kasami Sequences

Walsh - Hadamard Sequences

**Pseudo Noise Sequence:**

These are noise like wide band spread spectrum signals are generated using the PN sequence. Pseudo random noise is a signal similar to noise which satis es one or more of the standard tests for statistical randomness. PN sequences are deterministically generated, however they are almost like random sequences to an observer. Although it seems to lack any de nite pattern, pseudo random noise consists of a deterministic sequence of pulses that will repeat itself after its period.

Various pseudo random codes are generated using LFSR(Linear Feedback Shift Register).To generate a m-sequence, feedback connection of LFSRs are connected according to a primitive polynomial (generator polynomial).

**Primitive Polynomials:**

A generator polynomial is said to be primitive if it cannot be factored (i.e. it is prime), and if it is a factor of (i.e. can evenly divide) XN + 1, where N = 2m-1 (the length of the m-sequence)

**Maximum Length Sequences:**

Shift-register sequences having the maximum possible period for an r-stage shift register are called maximal length sequences or m-sequences. A primitive generator polynomial always yield an m-sequence. The maximum period of an r-stage shift register can be proven to be 2r - 1. The m-sequences has three important properties, i.e., balance property, run-length property and shift-and-add property.

MLSs are spectrally at, with the exception of a near-zero DC term.

An important property of MLS is that its auto - correlation function is essentially an impulse.

It is hence possible to measure the impulse response of linear systems by calculating the cross - correlation between the MLS and the system output signal. If x(k) is a known pseudo random sequence, there exists an e cient and very fast way to calculate the cross correlation function Rxy(k), called the Fast Hadamard Transform, or the FHT.

It is a deterministic signal, but has similar spectral properties as true random white noise. Since it is deter-ministic, it can be repeated precisely.

It is therefore possible to increase the SNR by synchronous averaging of the response sequences. Since it is periodic, the auto spectrum for the sequence consists of lines separated by 1/T.

It can be shown that the lines for frequencies well below the clock frequency are equal in magnitude and that the spectrum thus approximates white band - limited noise.

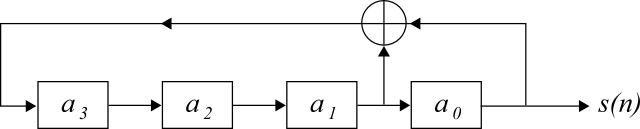


Figure 1: A pseudo random generator

**Walsh Codes:**

The Walsh code is an error-correcting code that is used for error detection and correction when transmitting messages over very noisy or unreliable channels. These codes are also known under Hadamard code and Walsh - Hadamard code.

It is an example of linear code over a binary alphabet that maps messages of length k to code words of length 2k.

In CDMA, it is used to de ne individual communication channels. Since Walsh code words are mathematically orthogonal, a Walsh-encoded signal appears as random noise to a CDMA capable mobile terminal, unless that terminal uses the same codeword as the one used to encode the incoming signal.

Hadamard codes are based on Hadamard matrices. A Hadamard matrix H of order n is an n x n matrix of 1s and -1s in which HHT = nIn (In is the n x n identity matrix).

**Gold Codes:**

Some pairs of m-sequences with the same degree can be used to generate Gold codes by linearly combining two m-sequences with di erent o set in Galois eld. All pairs of m-sequences do not yield Gold codes and those which yield Gold codes are called preferred pairs.

The generation of Gold codes is very simple. Using two preferred m-sequence generators of degree r, with a xed non-zero seed in the rst generator, 2r Gold codes are obtained by changing the seed of the second generator from 0 to 2r - 1. Another Gold sequence can obtained by setting all zero to the rst generator, which is the second

m-sequence itself. In total, 2r + 1 Gold codes are available.

**MATLAB CODES:**

%PN Sequence

function [pnseq] = pnsq(G,m) S = [1 zeros(1,m-1)];

N = 2^m-1; pnseq = []; for i=1:3\*N

pnseq = [pnseq S(1)];

k = mod( sum(S.\*G(1:m)),2);

S = circshift(S',-1)';%circular left shift S(m) = k;

end

pnseq = 2\*pnseq-1; end

%Correlation

function [corr] = cor(C1,C2) C1 = [C1 C1];

corr = [];

N = length(C2); for i=1:N

corr = [corr sum( C1(i:N+i-1).\*C2 )/N];

end stem(corr);

%Gold codes clc;

clear all; m = [5 6];

G = [1 0 0 1 0 1 0 1 1 1 1 0 1 0; 1 0 0 0 0 1 1 1 1 0 0 1 1 1];

for i=1:length(m)

C11 = (pnsq(G(i,1:7),m(i))==1);

C12 = []; k = 3;

V11 = [C11 C11 C11 C11 C11]; for j=1:3\*(2^m(i)-1)

C12 = [C12 V11(k)]; k = k+5;

end

X1 = 2\*xor(C11,circshift(C12',1)')-1;

X2 = 2\*xor(C11,circshift(C12',2)')-1; figure;

subplot(211);

cor(X1,X1);

title(['Auto-correlation of ',num2str(2^m(i)-1),' length goldcode']); subplot(212);

cor(X1,X2);

title(['Cross-correlation of ',num2str(2^m(i)-1),' length goldcode']);

end

%Hadamard codes clc;

clear all; N=32;

H32 = ones(N); k = N/2; while k>=1

for i=1:N/k for j=1:N/k

if mod(i,2)+mod(j,2)==0 H32((i-1)\*k+1:i\*k,(j-1)\*k+1:j\*k)=~H32((i-1)\*k+1:i\*k,(j-1)\*k+1:j\*k);

end

end

end k=k/2;

end

H32 = 2\*H32-1; N=64;

H64 = ones(N); k = N/2; while k>=1

for i=1:N/k for j=1:N/k

if mod(i,2)+mod(j,2)==0 H64((i-1)\*k+1:i\*k,(j-1)\*k+1:j\*k)=~H64((i-1)\*k+1:i\*k,(j-1)\*k+1:j\*k);

end

end

end k=k/2;

end

H64 = 2\*H64-1; figure; subplot(211);

cor([H32(3,:) H32(3,:) H32(3,:) H32(3,:)],[H32(3,:) H32(3,:) H32(3,:) H32(3,:)]); title('Auto-correlation of 32 length Walsh-Hadamard code');

subplot(212);

cor([H32(3,:) H32(3,:) H32(3,:) H32(3,:)],[H32(15,:) H32(15,:) H32(15,:) H32(15,:)]); title('Cross-correlation of 32 length Walsh-Hadamard code');

figure;

subplot(211);

cor(H64(8,:),H64(8,:));

title('Auto-correlation of 64 length Walsh-Hadamard code'); subplot(212);

cor(H64(8,:),H64(15,:));

title('cross-correlation of 64 length Walsh-Hadamard code');

%main code clc;

close all; m = [4 5 6];

G = [1 0 0 1 1 0 0 1 1 0 0 1 0 0; 1 0 0 1 0 1 0 1 1 1 1 0 1 0; 1 0 0 0 0 1 1 1 1 0 0 1 1 1];

for i=1:3

C1 = pnsq(G(i,1:7),m(i));

C2 = pnsq(G(i,8:14),m(i)); figure;

subplot(211)

cor(C1,C1);

title(['Auto-correlation of ',num2str(2^m(i)-1),'-sequence']); subplot(212)

cor(C1,C2);

title(['Cross-correlation of ',num2str(2^m(i)-1),'-sequence']);

end goldcodes; hadam;

%BPSK modulation clear all; N=1024;

s1= rand(1,N)>0.5; ip1=2\*s1-1; t=0:0.00001:0.001; c1= sin (2\* pi\* 1 \*t); c=c1/norm(c1); ebyn0db=0:14;

ebyn0=10.^(ebyn0db/10);

PSK1=ip1'\*c; %msequence length=31 nBits=5;MLS5=[];

weights = [zeros(1, nBits - 1), 1]; for i = 1 : 2^nBits - 1

MLS5(i, :) = weights;

tapVal = rem((weights(5) + weights(2)), 2); weights = circshift(weights, [0 1]); weights(1) = tapVal;

end

%msequence length=63 nBits=6;MLS6=[];

weights = [zeros(1, nBits - 1), 1]; for i = 1 : 2^nBits - 1

MLS6(i, :) = weights;

tapVal = rem((weights(6) + weights(1)), 2); weights = circshift(weights, [0 1]); weights(1) = tapVal;

end

for n=1:2 if n==1

l=31;pn=2\*MLS5(:,5)'-1;

else

l=63;pn=2\*MLS6(:,6)'-1;

end ip=[];

for i=1:length(s1)

temp=ip1(i)\*pn; ip=[ip temp];

end PSK=ip'\*c;

% AWGN channel

n1 = randn(length(ip),length(t));

n11 = randn(length(ip1),length(t)); BER = []; BERn=[];

for k = 1:length(ebyn0db) n0 = l/ebyn0(k); n01=1/ebyn0(k);

rcv1=PSK+sqrt(n0/2)\*n1; rcv1\_1=PSK1+sqrt(n01/2)\*n11; rcv=rcv1\*c';

rcv=rcv'; rcv\_1=rcv1\_1\*c'; rcv\_1=rcv\_1'; rx=[];

for i=1:length(rcv)/l

d= sum(rcv(l\*(i-1)+1:l\*(i-1)+l).\*pn)/l; rx=[rx d];

end

y= rx > 0; err=xor(y,s1); BER(k)=mean(err); y=rcv\_1>0; err=xor(y,s1); BERn(k)=mean(err);

end

thber = 0.5\*erfc(sqrt(10.^(ebyn0db/10)));

%Rayleigh channel

n2 = randn(length(ip),length(t))+1i\*randn(length(ip),length(t));

n21 = randn(length(ip1),length(t))+1i\*randn(length(ip1),length(t)); BERr=[]; BERrn=[];

for k = 1:length(ebyn0db) n0 = l/ebyn0(k); n01=1/ebyn0(k);

rcv11=n2.\*PSK+sqrt(n0/2)\*n1; rcv11\_1=n21.\*PSK1+sqrt(n01/2)\*n11; rcv1=rcv11./n2;

rcv=rcv1\*c';

rcv=rcv'; rcv1\_1=rcv11\_1./n21; rcv\_1=rcv1\_1\*c'; rcv\_1=rcv\_1'; rx=[];

for i=1:length(rcv)/l

d= sum(rcv(l\*(i-1)+1:l\*(i-1)+l).\*pn)/l; rx=[rx d];

end

y= rx > 0; err=xor(y,s1); BERr(k)=mean(err); y=rcv\_1>0;

err=xor(y,s1);

BERrn(k)=mean(err);

end

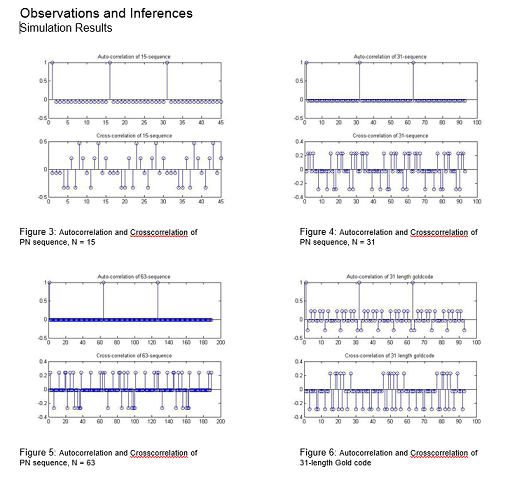
thray = 0.5.\*(1-sqrt(ebyn0./(ebyn0+1))); figure; semilogy(ebyn0db,thber,'b-');hold on; semilogy(ebyn0db,BER,'rx-');hold on; semilogy(ebyn0db,BERn,'g.-');hold on; semilogy(ebyn0db,thray,'b-');hold on; semilogy(ebyn0db,BERr,'rx-');grid on; semilogy(ebyn0db,BERrn,'g.-');grid on;

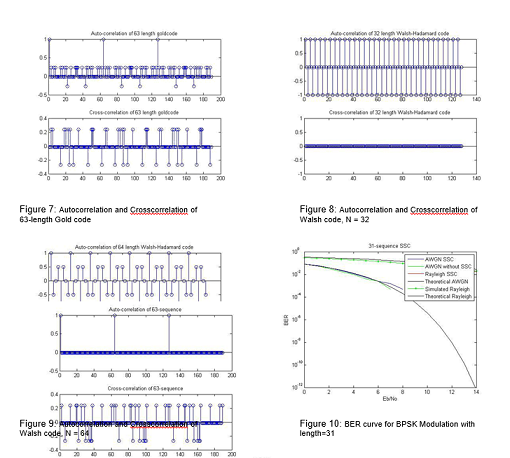
legend('theory-AWGN', 'simulation-AWGN with Spreading','simulation-AWGN without Spreading','theory-Ray xlabel('Eb/No, dB');

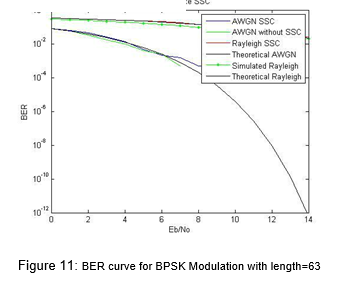
ylabel('Bit Error Rate');

str=sprintf('Bit error probability curve for BPSK modulation for m-sequence length=%d',l); title(str);

end







Inferences:

Walsh codes posses excellent cross-correlation property ( cross correlation of one Walsh code with another is always zero) therefore possess excellent orthogonality property.

The receiver utilizes the auto correlation property of the gold code to depict the message.

The auto-correlation property of Walsh code is very poor and so it is used only in synchronous CDMA networks, which maintains a synchronizing mechanism to identify the starting of the codeword.

Gold codes were constructed from a modulo 2 addition of two maximum length sequences, which helps in generating a large number of codes.

A PN generator will produce a periodic sequence that eventually repeats, but that appears to be random.

The m - sequence provide the best periodic auto correlation in terms of minimizing value of the out - of phase auto correlation.

The cyclic shift of the m - sequence was also observed to be an m - sequence.

The m - sequence is best utilized if the synchronization window is longer than the on period.

Results

The simulation of spread spectrum communication system was performed.

The auto correlation and cross correlation properties of Gold codes, Walsh code and PN sequences were determined and analyzed

BER performance of BPSK scheme with and without frequency spreading was plotted and analyzed.