1/c Cole Gingrich

ENAV HW #3

26FEB14





Delay for Satellite 3: 50

Delay for Satellite 18: 150

Delay for Satellite 19: 75

Delay for Satellite 26: 200

function checkNoise( Noisy )

MaxLag = 500;

LagVector = -MaxLag:MaxLag;

for i = 1:32

PN = PRN\_Generator(i);

XCorr = xcorr(PN,Noisy,MaxLag);

XCorr = abs(XCorr);

figure(i);

plot(LagVector,XCorr); grid on;

title(i);

scale = [-MaxLag,MaxLag,-10,1024]; axis(scale);

end

end

function Gold = PRN\_Generator(Satellite)

%Creates Gold Code for a given satellite

%Initialize shift registers to all 1

G1 = ones(1,10);

G2 = ones(1,10);

%Initialize output array

Gold = zeros(1,1023);

%Select Phase Selector based on satellite

switch Satellite

case 1,

taps = [2 6];

case 2,

taps = [3 7];

case 3,

taps = [4 8];

case 4,

taps = [5 9];

case 5,

taps = [1 9];

case 6,

taps = [2 10];

case 7,

taps = [1 8];

case 8,

taps = [2 9];

case 9,

taps = [3 10];

case 10,

taps = [2 3];

case 11,

taps = [3 4];

case 12,

taps = [5 6];

case 13,

taps = [6 7];

case 14,

taps = [7 8];

case 15,

taps = [8 9];

case 16,

taps = [9 10];

case 17,

taps = [1 4];

case 18,

taps = [2 5];

case 19,

taps = [3 6];

case 20,

taps = [4 7];

case 21,

taps = [5 8];

case 22,

taps = [6 9];

case 23,

taps = [1 3];

case 24,

taps = [4 6];

case 25,

taps = [5 6];

case 26,

taps = [6 8];

case 27,

taps = [7 9];

case 28,

taps = [8 10];

case 29,

taps = [1 6];

case 30,

taps = [2 7];

case 31,

taps = [3 8];

case 32

taps = [4 9];

otherwise,

disp('You lose');

end

for clock=1:1023

%Calculate bit for Gold Code

G21 = bitxor( G2(taps(1)),G2(taps(2)));

Gold(clock) = bitxor(G21,G1(10));

%Shift G1 Register

temp = bitxor(G1(3),G1(10));

G1 = [temp G1(1:9)];

%Shift G2 Register

temp1 = bitxor(G2(2),G2(3));

temp2 = bitxor(temp1,G2(6));

temp3 = bitxor(temp2,G2(8));

temp4 = bitxor(temp3,G2(9));

temp5 = bitxor(temp4,G2(10));

G2 = [temp5 G2(1:9)];

end

Gold = PRN\_Convert(Gold);

end

function New = PRN\_Convert( Incoming )

%Converts the PRN sequence from 0's and 1's to 1's and -1's

New = ((Incoming==0)\*1 + (Incoming==1)\*(-1))\*(-1);

end