UEC1728 - Cognitive Radio
Assignment

Vignesh Mohan
183002181
Electronics and Communication Department
Sri Sivasubramiya Nadar College of Engineering, 603110

1 Energy Detection

Aim:

To detect the unknown PU (Primary user) signal based on energy of the received signal. The Energy detector is optimal to detect the unknown signal if the noise power is known. In the energy detection, CR users sense the presence/absence of the PUs based on the energy of the received signals. As above, the measured signal r(t) is squared and integrated over the observation interval T. Finally, the output of the integrator's compared with a threshold k to decide if a PU is present. Since the energy detection depends only on the SNR of the received signal, its performance is susceptible to uncertainty in noise power. If the noise power is uncertain, the energy detector will not be able to detect the signal reliably as the SNR is less than a certain threshold, called an SNR wall. In addition, the energy detector can only determine the presence of the signal but cannot differentiate signal types.

i) Code for performance of detector (pd) under various values of probability of false alarm (pfa) for SNR = 6dB

```
% energy_detection_pfa.m
clc;
close all;
m = 10;
N = 2 * m;
pf = logspace(-5,0,20);
snr_avgdB = 4;
snr_avg = power(10, snr_avgdB / 10);
for i = 1:length(pf)
    over_num = 0;
    th(i) = gammaincinv(1 - pf(i), 2 * m);
    for kk = 1:1000
        t = 1:N;
        x = sin(pi * t);
        noise = randn(1, N);
        pn = mean(noise.^2);
        amp = sqrt(noise.^2 * snr_avg);
        x = amp .* x ./ abs(x);
```

```
        SNRdB_Sample = 10 * log10(x.^2 ./ (noise.^2));
        signal = x(1:N);
        ps = mean(abs(signal).^2);
        Rev_sig = signal + noise;
        accum_power(i) = sum(abs(Rev_sig)) / pn;
        if accum_power(i) > th(i)
            over_num = over_num + 1;
        end
    end
    pd_sim(i) = over_num / kk;
end
figure;
s = semilogy(pf, pd_sim, 'r-*');
set(s, 'linewidth', 1);
title('Nonfluctuating Coherent ROC');
grid on;
xlabel('Probability of Falsed Alarm (pfa)');
ylabel('Probability of Detection (pd)');
legend(['SNR=' num2str(snr_avgdB) 'dB']);
```

ii) Code for performance of detector under various values of signal to noise ratio (SNR)

```
% energy_detection_snr.m
clc;
close all;
m = 10;
N = 2 * m;
pd_sim = [];
pf = 0.5;
for snr_avgdB = 0:1:30
    snr_avg = power(10, snr_avgdB / 10);
    for i = 1:length(pf)
        over_num = 0;
        th(i) = gammaincinv(1 - pf(i), 2 * m);
        for kk = 1:1000
            t = 1:N;
            x = sin(pi * t);
            noise = randn(1, N);
            pn = mean(noise.^2);
            amp = sqrt(noise.^2 * snr_avg);
            x = amp .* x ./ abs(x);
            SNRdB_Sample = 10 * log10(x.^2 ./ (noise.^2));
            signal = x(1:N);
            ps = mean(abs(signal).^2);
            Rev_sig = signal + noise;
            accum_power(i) = sum(abs(Rev_sig)) / pn;
            if accum_power(i) > th(i)
                over_num = over_num + 1;
```

```
            end
        end
        pd_sim = [pd_sim over_num / kk];
    end
end
figure;
SNR = 0:1:30;
s = plot(SNR, pd_sim, 'r-*');
set(s, 'linewidth', 1);
title('Energy detection method');
grid on;
xlabel('Signal-to-noise ratio (dB)');
ylabel('Probability of Detection (pd)');
legend(['PFA=' num2str(pf)]);
```

Results and Inference:

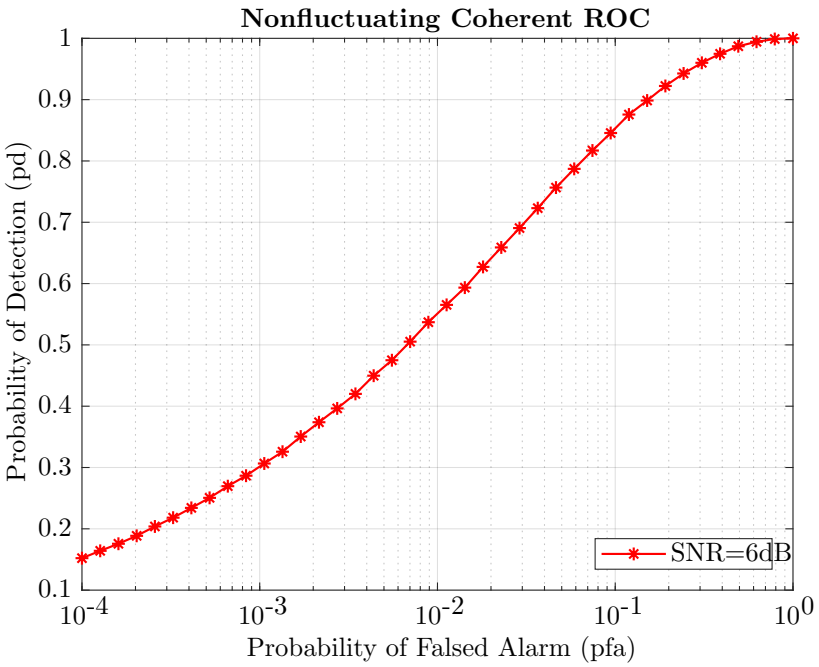

Figure 1.1: Pd vs Pfa for Energy Detection (SNR = 6dB)

From Figure 1.1, we can infer the fact that there is an trade-off/inverse relationship between Pd and Pfa. Since we want Pd value to be high (greater than 0.5), we see that the corresponding Pfa values range from 0.01 to 0.8. After that the detection probability Pd is approximately 1 for SNR = 6 db.

From Figure 1.2, we can infer that the relation is between Pd and SNR is linear for a certain range after which Pd saturates for higher values of SNR (for Pfa = 0.5). For lower values of SNR the detection probability is almost 0 and

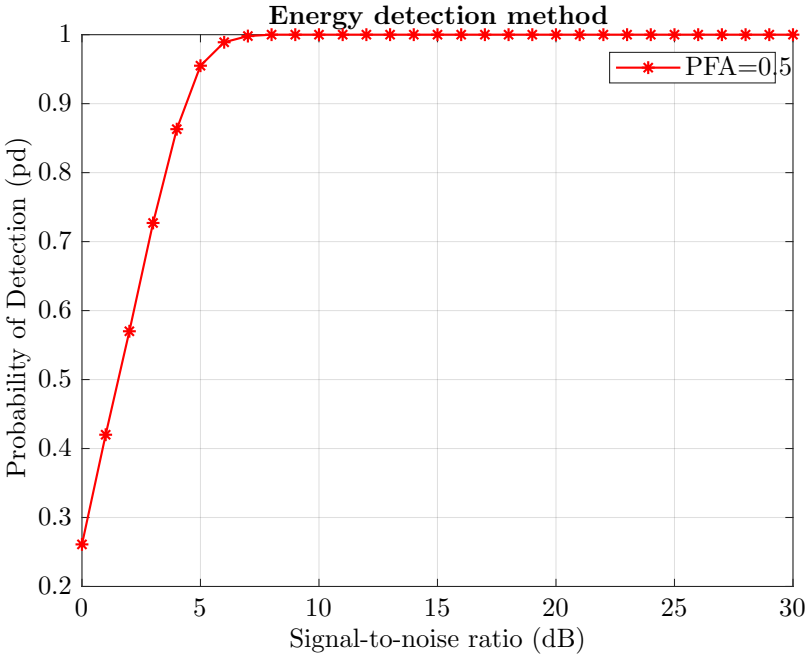

Figure 1.2: Pd vs SNR for Energy Detection (Pfa = 0.5)

increasing at a constant slope for every dB increase in SNR. Above 5 db the detection probability Pd saturates 1.

2 Matched Filter Detection

Aim:

To detect the unknown PU (Primary user) signal based on Prior knowledge (bandwidth, operating frequency, modulation type etc) and to maximize the SNR. When the user has information on the primary user signal, the optimal detector in stationary Gaussian noise happens to be the matched filter as it maximizes the reclosed signal-to-noise ratio (SNR). The palpable advantage of matched filter lies in its requirement of smaller time for achieving high processing gain for the reason of coherence and of a prior knowledge of the basic user signal like the modulation type and order, the shape of the pulse and the packet format. The performance of the matched filter will be poor if this information has no accuracy.

i) Code for performance of detector (pd) under various values of probability of false alarm (pfa) for SNR = 6dB

```
% matched_filter_detection_pfa.m
clc;
close all;
```

```
rstream = RandStream.create('mt19937ar', 'seed', 2009);
Ntrial = 1e5; % number of Monte-Carlo trials
spower = 1; % signal power is 1
pd_graph = [];
for snrdb = 6 % SNR in dB
    snr = db2pow(snrdb); % SNR in linear scale
    npower = spower / snr; % noise power
    namp = sqrt(npower / 2); % noise amplitude in each channel
    s = ones(1, Ntrial); % signal
    n = namp * (randn(rstream, 1, Ntrial) + 1i...
        * randn(rstream, 1, Ntrial)); % noise
    x = s + n;
    mf = 1;
    y = mf' * x;
    z = real(y);
    Pfa = 1e-3;
    snrthreshold = npwgnthresh(Pfa, 1, 'coherent');
    mfgain = mf' * mf;
    % To match the equation in the text above
    % npower - N
    % mfgain - M
    % snrthreshold - SNR
    threshold = sqrt(npower * mfgain * snrthreshold);
    Pd = sum(z > threshold) / Ntrial;
    x = n;
    y = mf' * x;
    z = real(y);
    Pfa = sum(z > threshold) / Ntrial;
    [Y, X] = rocsnr(snrdb, 'SignalType', ...
        'Nonfluctuatingcoherent', 'MinPfa', 1e-4);
    pd_graph = [pd_graph; Y'];
end
figure;
s=semilogx(X,pd_graph(1,:),'r-*');
set(s, 'linewidth', 1);
title('Nonfluctuating Coherent ROC - Matched Filter');
grid on;
xlabel('probability of falsed alarm(pfa)');
ylabel('probability of detection(pd)');
snr_avgdB = 6;
legend(['SNR=' num2str(snr_avgdB(1)) 'dB'])
```

ii) Code for performance of detector under various values of signal to noise ratio (SNR)

```
% matched_filter_detection_snr.m
clc;
close all;
rstream = RandStream.create('mt19937ar', 'seed', 2009);
```

```
spower = 1; % signal power is 1
Ntrial = 1e5; % number of Monte-Carlo trials
s = ones(1, Ntrial); % signal
mf = 1;
Pfa = 0.5;
Pd = [];
for snrdb = 0:1:30 % SNR in dB
    snr = db2pow(snrdb); % SNR in linear scale
    npower = spower / snr; % noise power
    namp = sqrt(npower / 2); % noise amplitude in each channel
    n = namp * (randn(rstream, 1, Ntrial) + 1i ...
        * randn(rstream, 1, Ntrial)); % noise
    x = s + n;
    y = mf' * x; % apply the matched filter
    z = real(y);
    snrthreshold = npwgnthresh(Pfa, 1, 'coherent');
    mfgain = mf' * mf;
    threshold = sqrt(npower * mfgain * snrthreshold);
    Pd = [Pd sum(z > threshold) / Ntrial];
end
figure;
SNR = 0:1:30;
s = plot(SNR, Pd, 'r-*');
set(s, 'linewidth', 1);
title('Matched Filter');
grid on;
xlabel('Signal-to-noise ratio (dB)');
ylabel('probability of detection(pd)');
legend(['PFA=' num2str(Pfa)]);
```

Results and Inference:

From Figure 2.1 we can infer the fact that there is an trade-off/inverse relationship between Pd and Pfa, similar to Energy Detection. Since we want over Pd value to be high (greater than 0.5), we see that the corresponding Pfa values range from 0.005 to 0.6. After that the detection probability Pd is approximately 1 for SNR = 6 db. For the same value of SNR, Matched Filter detection provides better Pd vs Pfa characteristics when compared to Energy Detection.

From Figure 2.2, we can infer that the relation is between Pd and SNR is linear with saturation at a threshold, similar to Energy Detection (for Pfa = 0.5). However, it can be clearly seen that the characteristics of Pd vs SNR for Matched Filter Detection is much better than that of Energy Detection. The detection probability is high even for very low values of SNR. As expected it saturates to maximum probability around 5 dB.

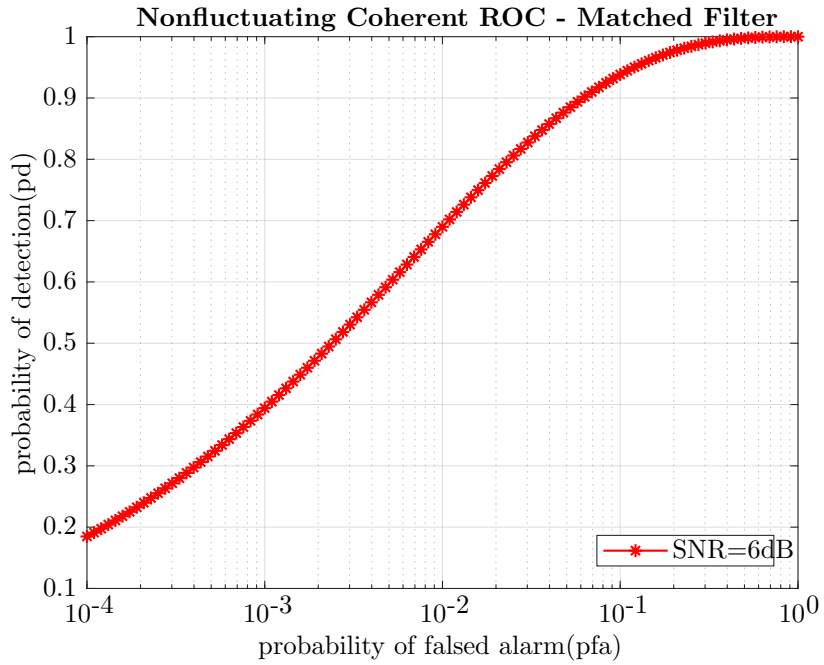

Figure 2.1: Pd vs Pfa for Matched Filter Detection (SNR = 6dB)

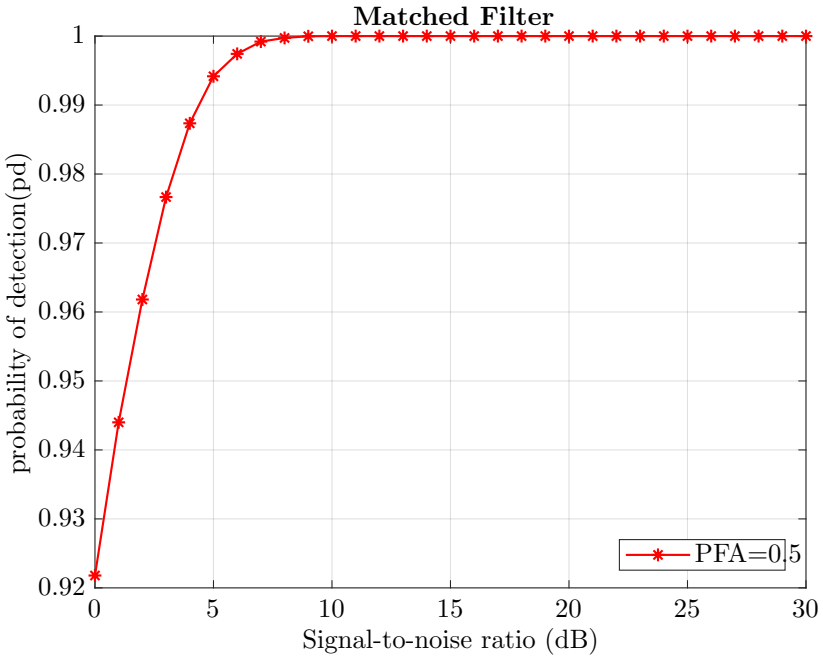

Figure 2.2: Pd vs SNR for Matched Filter Detection (Pfa = 0.5)

3 Cyclostationary Feature Detection

Aim:

To detect the PU (Primary user) signal exploiting the cyclostationary features (spreading code, cyclic prefixes etc). Cyclostationary feature detection is an alternative detection method. Modulated signals are combined with sine wave carriers, pulse trains, repeating spreading, hopping sequences or cyclic prefixes which end up in built in periodicity. These modulated for the reason that their mean and autocorrelation display periodicity feature. Analysis of the spectral correlation function helps detection of such periodicity feature. The spectral correlation function can distinguish between noise energy and modulated signal energy. This arises from the feature of noise as a wide-sense stationary signal and no correlation. On the other hand, modulated signals have the feature of being cyclostationary with special correlation in view of the embedded redundancy of signal periodicity. Therefore, it is possible for a detector of cyclostationary feature top perform better than the energy detector for discrimination against noise. This is because of its robustness to uncertainty seen in noise power. But, there is the computational complexity entailing a long duration for observation.

Code for Cyclostationary Feature Detection

```
% cyclostationary_feature_detection.m
clc;
close all;
r = randi([0 1], 100, 1);
stem(r, 'DisplayName', 'r');
hold on;
l = length(r);
r(l + 1) = 0;
n = 1;
L = 1000;
fs = 1000;
T = 1 / fs;
fc = 3 / T;
t1 = (0:L);
while n <= l
    t = (n - 1):.001:n;
    if r(n) == 1
        if r(n + 1) == r(n)
            y = (t <= n);
        else
            y = (t < n);
        end
    else
        if r(n + 1) == r(n)
            y = (t > n);
        else
            y = (t >= n);
```

```
        end
    end
    plot(t, y, 'r');
    title('NRZ encoding');
    grid on;
    xlabel('time');
    ylabel('amplitude');
    hold on;
    axis([0 100 -1.5 1.5]);
    n = n + 1;
end
b = (2 * 3.142 * fc * t1) + (3.142 * (1 - y));
s = sqrt(2) * cos(b);
figure;
plot(t1, s, 'r');
axis([0 1000 -1.5 1.5]);
title('BPSK - Transmitted');
xlabel('Time');
ylabel('Amplitude');
axis([0 1000 -1.5 1.5])
hold on;
noise = awgn(s, 20, 'measured');
snr_db = 10 * log10(s.^2 / (noise.^2));
snr_avg = power(10, snr_db / 10);
rs1 = (s + noise)/2;
vr = var(rs1);
figure;
plot(t1, rs1, 'r');
axis([0 100 -1.5 1.5]);
title('BPSK - Received');
xlabel('Time');
ylabel('Amplitude');
axis([0 1000 -1.5 1.5])
hold on;
nfft = 2^nextpow2(L);
f1 = fft(rs1, nfft) / L;
fs1 = fs / 2 * linspace(0, 1, nfft / 2 + 1);
figure;
plot(fs1, 2 * abs(f1(1:nfft / 2 + 1)), 'r');
title('FFT of received signal');
xlabel('Frequency (HZ)');
ylabel('|rs1|');
grid on;
hold on;
fc=1:3/T;
s1=cos(2*3.14*fc*T);
l1=length(s1);
sc=s1+sin(2*3.14*fc*T);
l2=length(sc);
```

```
cr1=xcorr2(rs1,s1);
figure;
plot(cr1, 'r');
title('Cross Correlation of Rx signal with cos');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
axis([0 500 -5.5 5.5]);
hold on;
cr2=xcorr2(rs1,sc);
l2=length(cr2);
figure;
plot(cr2,'r');
title('Cross Correlation of Rx signal with with sin+cos');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
axis([0 500 -5.5 5.5]);
hold on;
for i = 1:500
    if s1(i) > 0.5 || sc(i) > 0.5
        disp('PU PRESENT');
    end
end
L = 2:1:30;
snr1 = 2:1:30;
q1 = (2 * L + 1) * (0.5)^2;
Pf = exp(-q1);
del = (2 * snr1 + 1) * vr / (2 * L + 1);
q2 = sqrt(2 * snr1 / vr);
q3 = 0.5 / del;
Pd = marcumq(q1, q2);
figure;
P = plot(snr1, Pd, 'r-*');
set(P, 'linewidth', 1);
title('Pd Vs SNR');
xlabel('SNR');
ylabel('Probability of detection');
hold on;
grid on;
figure;
P2 = semilogx(Pf, Pd, 'r-*');
set(P2, 'linewidth', 1);
title('Pfa Vs Pd');
xlabel('Probability of false alarm');
ylabel('Probability of detection');
hold on;
grid on;
```

Results and Inference:

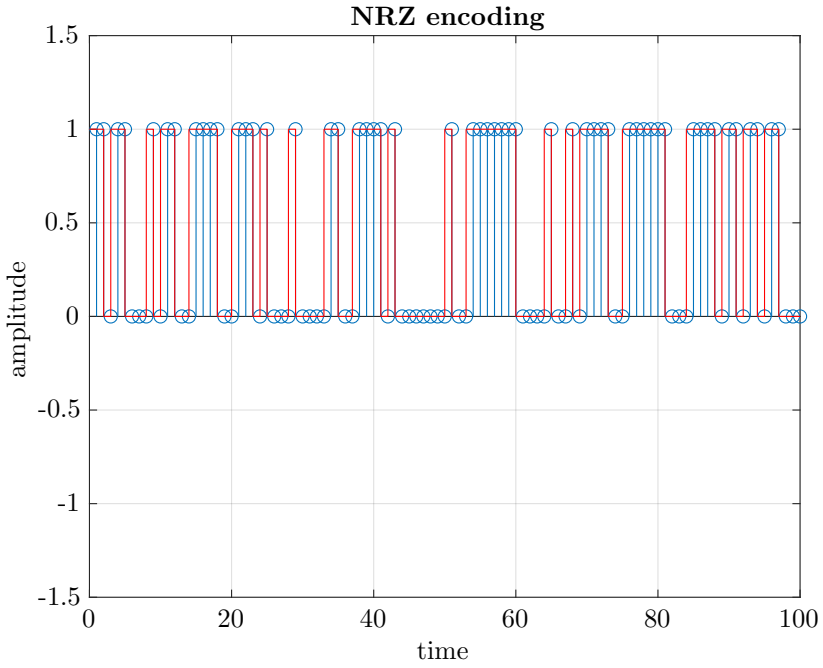

Figure 3.1: NRZ Encoding of input data

The figures 3.1, 3.2, 3.3 and 3.4 show the type of line coding (NRZ), signal with modulation type (BPSK) transmitted and received with AWGN noise, and the spectrum of the received signal respectively. Figures 3.5 and 3.6 show the cross correlation of the received signal with cosine signal and sine+cosine signal respectively. The amplitude of cross-correlated output signal is a measure of how much the received signal resembles the target signal. The correlation peak specifies the location of the target.

The main inference from figure 3.7 is the presence of decreasing relation between Pd and Pfa, contrary to what was seen in Energy Detection and Matched Filter Detection.For the lower values of Pf up to 0.0001, probability of detection is nearly 1 and after that, detection of PU is less for more probability of false alarm.

From figure 3.8, we can infer the presence of a linear increase in the probability of detection Pd for increasing values of SNR. For SNR = 2 dB, probability of detection is increased compared to other methods of detection. Finally, is that the detection probability reaches 1 at SNR = 15 dB.

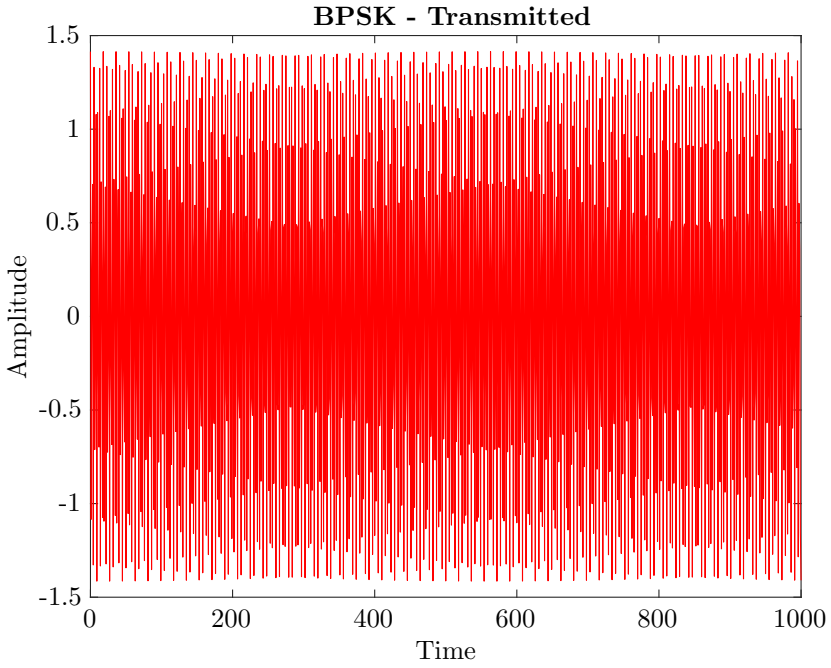

Figure 3.2: BPSK - Transmitted signal

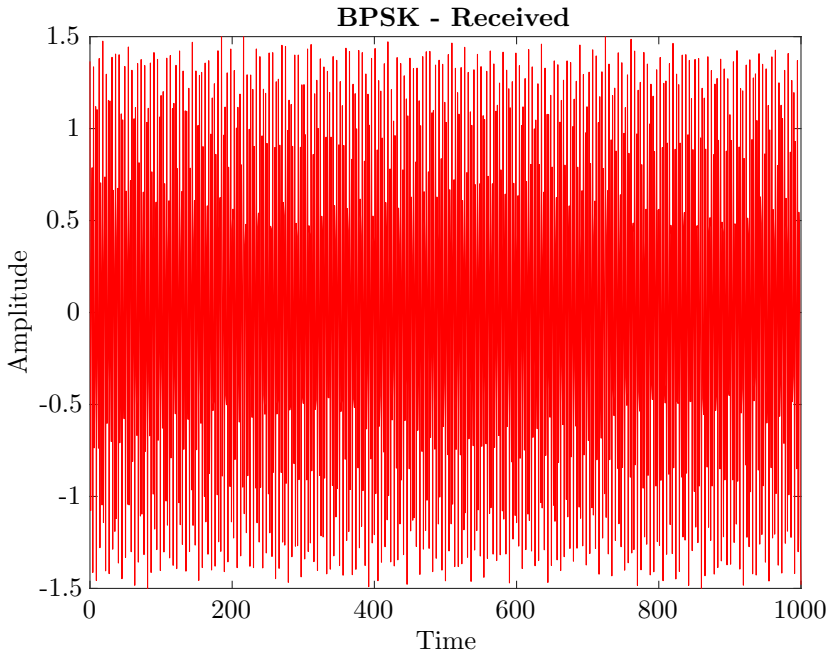

Figure 3.3: BPSK - Received signal with AWGN noise

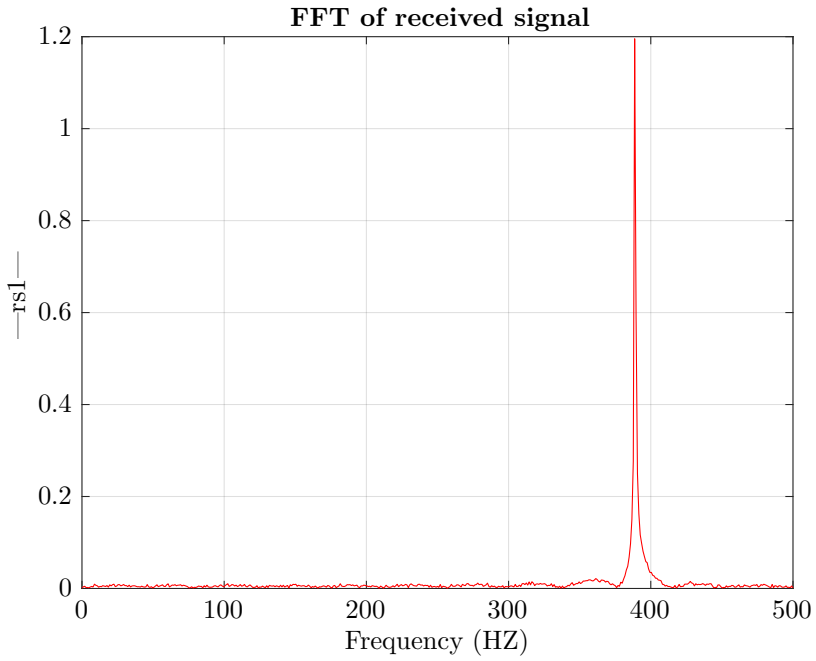

Figure 3.4: FFT of Received signal showing main frequency component

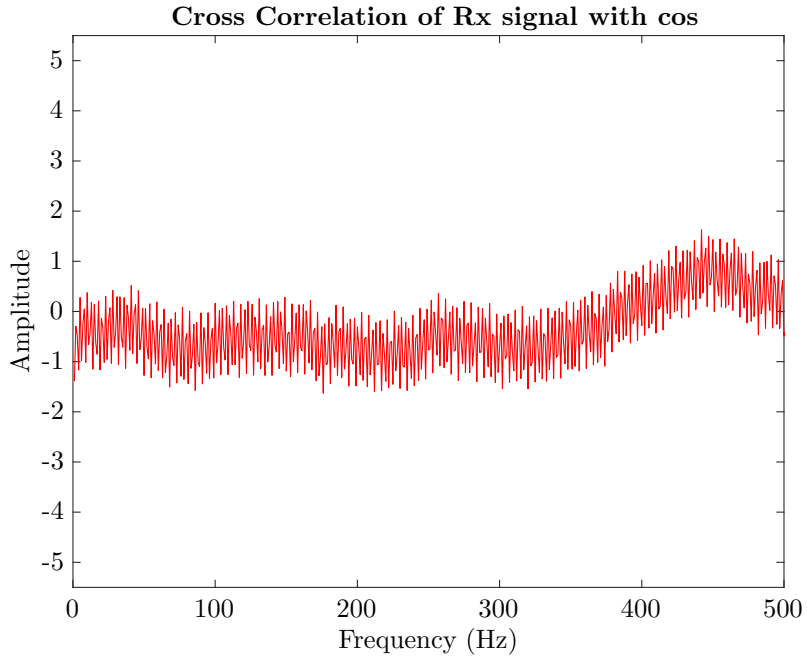

Figure 3.5: Cross Correlation of received signal with cosine

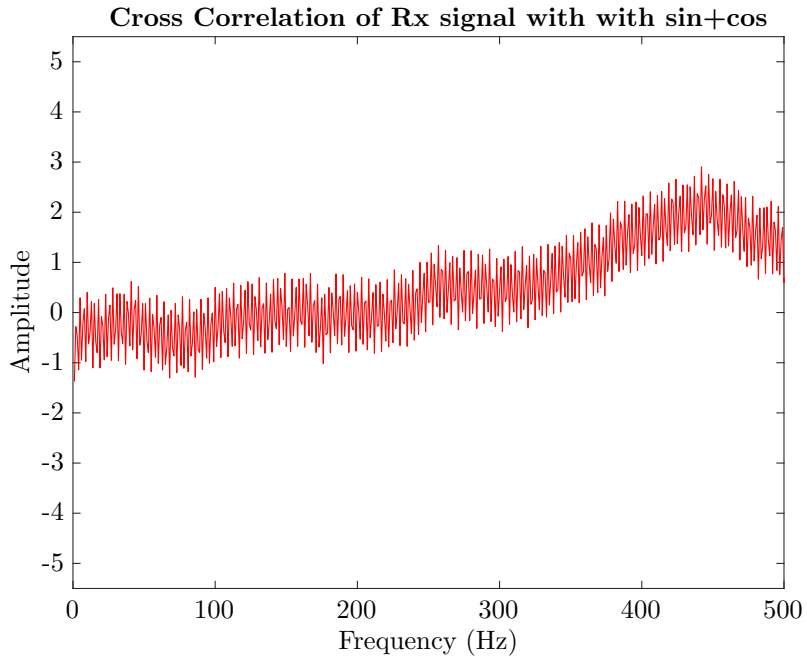

Figure 3.6: Cross Correlation of received signal with sine + cosine

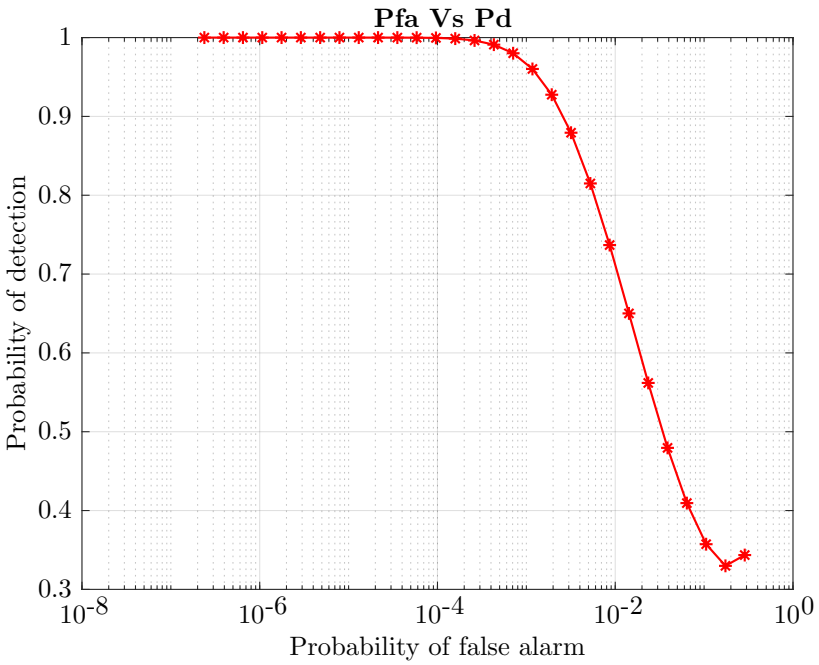

Figure 3.7: Pd Vs Pfa for Cyclostationary Feature Detection

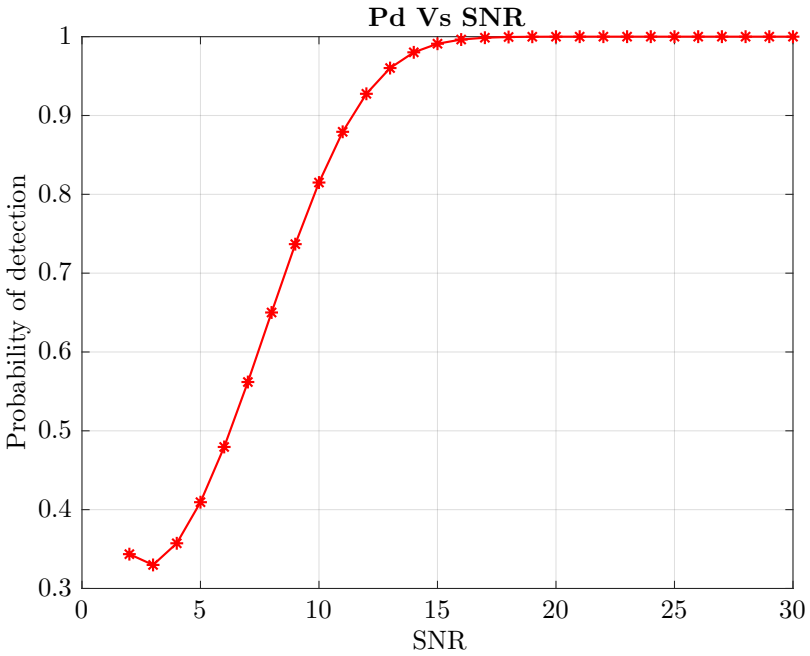

Figure 3.8: Pd Vs SNR for Cyclostationary Feature Detection

4 Co-Operative Spectrum Sensing

Aim:

To enhance the sensing performance by exploiting the spatial diversity in the observations of spatially located CR users. Theoretically, there is greater accuracy seen in cooperative detection among unlicensed users with the possibility of uncertainty in a single user's detection getting minimized. The features of multipath fading and shadowing effect undermine the performance of primary user detection methods. However, cooperative detection schemes permit mitigation of multipath fading and shadowing effects, thereby improving the detection probability in a heavily shadowed ambience.

Code for Co-Operative Spectrum Sensing

```
% cooperative_spectrum_sensing.m
clc;
close all;
u = 1000; % time bandwidth factor
N = 2 * u; % samples
a = 2; % path loss exponent
C = 2; % constant losses
Crs = 10; % Number of cognitive radio users
PdAnd = 0;
% ----------Pfa------------%
```

```
Pf = 0:0.005:1;
Pfa = Pf.^2;
% ---------signal-----%
t = 1:N;
s1 = cos(pi * t);
% s1power=var(s1);
% -------- SNR ----------%
% Snrdb=-15:1:15;
Snrdb = 15;
Snreal = power(10, Snrdb / 10); % Linear Snr
% while Snrdb<15
for i = 1:length(Pfa)
    lamda(i) = gammaincinv(1 - Pfa(i), u) * 2; % thershold
    % lamdadB=10*log10(lamda);
    % ---------Local spectrum sensing---------%
    for j = 1:Crs % for each node
        detect = 0;
        % d(j)=7+1.1*rand(); %random distanse
        d = 7:0.1:8;
        PL = C * (d(j)^-a); % path loss
        for sim = 1:10
        % Monte Carlo Simulation for 100 noise realisation
            % -------------AWGN channel--------------------%
            noise = randn(1, N);
            % Noise production with zero mean and s^2 var
            noise_power = mean(noise.^2); % noise average power
            amp = sqrt(noise.^2 * Snreal);
            s1 = amp .* s1 ./ abs(s1);
            % SNRdB_Sample=10*log10(s1.^2./(noise.^2));
            Rec_signal = s1 + noise; % received signal
            localSNR(j) = mean(abs(s1).^2) * PL / noise_power;
            Pdth(j, i) = marcumq(sqrt(2 * localSNR(j))...
                , sqrt(lamda(i)), u); % Pd for j node
            % Computation of Test statistic for energy detection
            Sum = abs(Rec_signal) * PL;
            Test(j, sim) = sum(Sum);
            if Test(j, sim) > lamda(i)
                detect = detect + 1;
            end
        end % END Monte Carlo
        Pdsim(j) = detect / sim;
        % Pd of simulation for the j-th CRuser
    end
    PdAND(i) = prod(Pdsim);
    PdOR(i) = 1 - prod(1 - Pdsim);
end
PdAND5 = (Pdth(5, :)).^5;
Pmd5 = 1 - PdAND5;
PdANDth = (Pdth(Crs, :)).^Crs;
```

```
PmdANDth = 1 - PdANDth; % Probability of miss detection
Pmdsim = 1 - PdAND;
figure(1);
semilogy(Pfa, Pmdsim, 'r-x', ...
    Pfa, PmdANDth, 'k-o', Pfa, Pmd5, 'g-*');
title('Complementary ROC Cooperative sensing - AND rule (AWGN)');
grid on;
ylim([0,1]);
xlabel('Probability of False alarm (Pfa)');
ylabel('Probability of Missed Detection (Pmd)');
legend('Simulation', 'Theory n=10', 'Theory n=5');
```

Results and Inference:

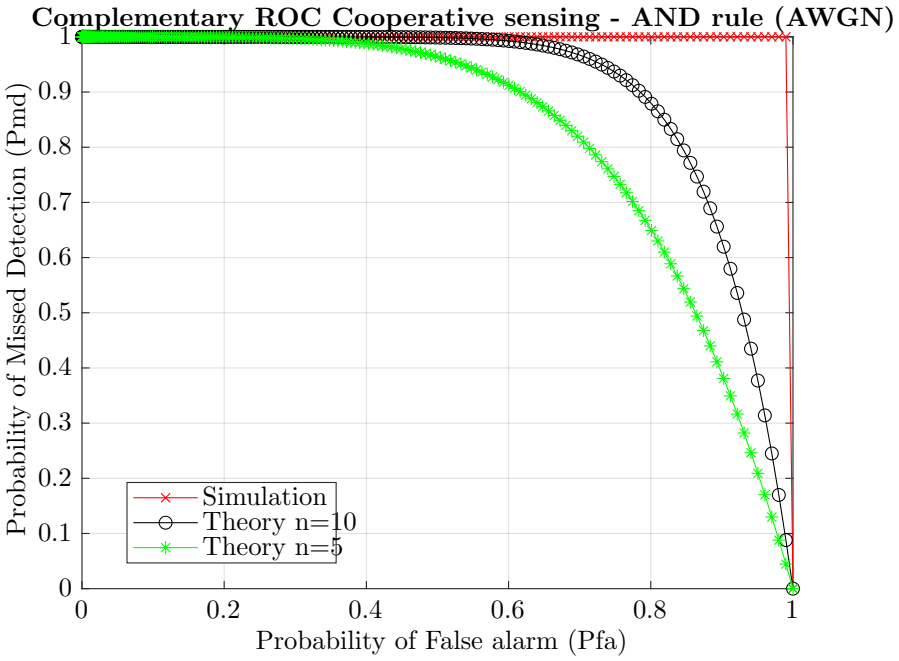

Figure 4.1: Complementary ROC of Cooperative sensing - AND rule under AWGN channel

From Figure 4.1 we can infer that for lower values of Pfa, probability of missed detection (Pmd) is almost 1 and as the value of Pfa increases, Pmd value gradually decreases.

When the number of users in the cognitive radio system is increased from 5 to 10, the performance of missed detection is increased. In simulation results, Pmd value remains 1 for Pfa values close to 1 (0.99) and later suddenly reaches 0 when Pfa = 1.