SPREAD SPECTRUM CHANNEL SOUNDER IMPLEMENTATION WITH USRP PLATFORMS

A Project Report

submitted by

PESEKE ANUSHA

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

JUNE 2016

THESIS CERTIFICATE

This is to certify that the thesis titled **SPREAD SPECTRUM CHANNEL SOUNDER IMPLEMENTATION WITH USRP PLATFORMS**, submitted by **Peseke Anusha**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology(Electrical Engineering)**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Dr.Radha Krishna Ganti
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 10th June 2016

ACKNOWLEDGEMENTS

I would like to thank my guide Dr. Radha Krishna Ganti for giving me the opportunity to work under his guidance.

ABSTRACT

KEYWORDS: CDMA Technique, USRP-N210, GNU radio, spread spectrum.

This paper shows the simulation and implementation of a spread spectrum channel sounder using USRP platforms and GNU Radio environment.Imperfections like Multiple path propagation are considered for the model.Two USRP-N210 devices are used,one for transmission and the other for receiving. GNU radio companion (GRC) is used as a SDR(Software defined radio) for real implementation through the USRPâĂŹs.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CDMA Code Division Multiplexing

USRP Universal Software Radio Peripheral

CHAPTER 1

Simulation in MATLAB

CDMA technique is used for transmission.Initially a channel model is assumed and the received bits are calculated. Using these received bits and the transmitted ones the channel is recovered. A k tap TDL (Tapped Delay line model) is assumed .LSE estimation is used for recovering the channel impulse responses at the k taps.

Channel imperfections such as multiple paths and Gaussian noise can be considered. The channel sounder allows to get the impulse response of the channel.

Mean square error is calculated for different SNR values and plotted to estimate the accuracy of the model.

1.1 LSE estimation

The method of least squares is about estimating parameters by minimizing the squared discrepancies between observed data, on the one hand, and their expected values on the other

Initially a channel model is assumed and the transmitted bits are sent .The received bits are used to recover back the channel coefficients using LSE. Mean square error is calculated for different Assumed SNR values for the channel model. The MSE is used to estimate the accuracy of the estimation algorithm.

Fig. 2.1CDMA block diagram

Fig. 1.2MSE with different SNR

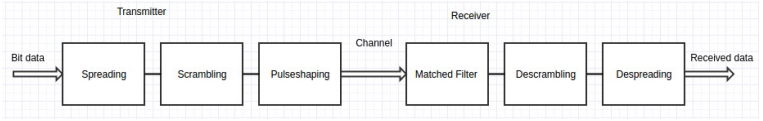

Figure 1.1: CDMA block diagram.Basic structure of the technique

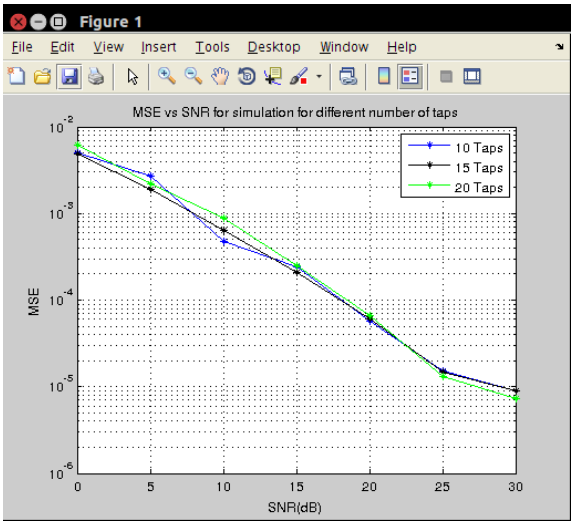

Figure 1.2: MSE vs SNR plot for different number of taps

1.2 MATLAB code

```
%Estimator run
function channel=Estimator_run(TAPS)
% k=50;%Number of taps
% Taps=[1:k];
disp('Reading');
k=TAPS;
txchips=read_complex_binary('test_transmit');
rxchips=read_complex_binary('test_receive');
disp('Written');
[X,Y,H]=channel_estimation(txchips,rxchips,k);
channel=H(1:k);
% yaxis=abs(channel(1:k));
% plot(Taps,yaxis);
end
```

```
function [X,Y,H]=channel_estimation(txchips,rxchips,k)
% R=fft(rxchips);
% T=fft(txchips);
% R=rxchips;
% T=txchips;
L=length(rxchips);
n=length(txchips);

R=rxchips(1:n);
T=txchips;

%Pseudo inverse
```

```
H=ifft(fft(rxchips,L)./fft(txchips,L));
%Finding txchips from rxbits
%Recovering the channel coefficients

%LS estimation
%Y=X*H
%H=(X'X)^-1.X'Y

%Matrix formation
%Assuming k taps
%H=zeros(k,1);
Y=zeros(n,1);
X=zeros(n,k);

%Construction of Y matrix
for r=1:n
   Y(r,1)=R(r,1);
end

%Construction of X matrix
for c=1:k
   X(:,c)=circshift(T,c-1);
end

%Channel
%H=((X.'*X)^-1)*(X.')*Y;

% XH=ctranspose(X);
% h=XH*X;
```

```
% H=h\(XH*Y)
disp('Done');

% %Finding PDP
% C=complex(randn(k,1),randn(k,1))*sqrt(0.5);
% %channel_coef=H.*PROFILE;
% %H=C.*PROFILE
% PROFILE=H./C
end
```

```
clc;
close all;
clear all;

SF=16;
profile=[0:-3:12].';
K=256; % No of symbols;
 X=hadamard(SF);
% [Gigcode]=gigcode(SF,3);
% [U D V]=svd(Gigcode);
% X=V
CH_CODE1=2;

code1=X(CH_CODE1,:)';
AWGN=1;
COUNT=1;
vSNR=0:5:30;
% vSNR=0;
TAPS=15;
```

```
profile=[0:-3:-3*(TAPS-1)].'; % Exponential decay
% profile=ones(TAPS,1);

PROFILE=10.^(profile/10); % Converted to LinearTAPS=5;
PROFILE=PROFILE/sqrt(sum(abs(PROFILE).^2)); % Normalizing
   the Profile

Eb=1/2;
Eu=1;
disp('start..');
for SNR=vSNR;
  snr=10^(SNR/10);
  No=Eb*Eu/(snr);

  tempmse=0;
  BLOCKS=10;
%  if(SNR>15)
%    BLOCKS=1000;
%  end
  for blk=1:BLOCKS
    % Tx Model goes here
    txbits=randsrc(2*K,1,[0
       1]);txsymbols=modulatebits(txbits);
    tmchips=spread(code1,txsymbols);

      % Scrambling Generator
      scrcode=randsrc(SF*K+2*SF,1,[1 -1]);
      tmchips=scrambler(scrcode,tmchips);
```

```
    % CHANNEL will GO here
    H=complex(randn(TAPS,1),randn(TAPS,1))*sqrt(0.5);
    channel_coef=H.*PROFILE;

    txchips=conv(channel_coef,tmchips);
    if(AWGN==1)
      rxchips=txchips+sqrt(No)*complex(randn(length(txchips),1),randn(lengt
    else
      rxchips=txchips;
    end
```

```
%Estimator run
function channel=Estimator_run(TAPS)
% k=50;%Number of taps
% Taps=[1:k];
disp('Reading');
k=TAPS;
txchips=read_complex_binary('test_transmit');
rxchips=read_complex_binary('test_receive');
disp('Written');
[X,Y,H]=channel_estimation(txchips,rxchips,k);
channel=H(1:k);
% yaxis=abs(channel(1:k));
% plot(Taps,yaxis);
end
```

CHAPTER 2

Real Implementation in USRP using GRC

GNU Radio Companion (GRC) is a free development environment for software radio. It has been first checked that transposed in GRC environment, without connecting to radio platforms, the simulations give similar results to MATLAB simulation.

2.1 Real implementation on USRP: transmitter

Two USRPâĂŹs are required in order to implement the channel sounder on USRP platforms: a transmitter capable of generating m-sequences and a receiver capable of correlating the received signal with the m-sequence used at Tx. Each USRP is connected to a host computer equipped with one USRP1 platform (via Gigabit ethernet connection). GNU Radio is used through GRC to conceive the designs. In GRC, components are sent to the Tx USRP platform through the imaginary and real parts of the USRP sink block input .

2.2 Real implementation on USRP: receiver

The receiver includes a USRP source block. The working frequency is 2.4 GHz ; however the actual frequency of Tx and Rx local oscillators in USRP devices can deviate from that frequency (the difference is up to a few kHz).

The observation of the impulse response provides useful information about the imperfections introduced by the channel.We can see that in these conditions of experimentation, the signal to noise ratio is high ; indeed, USRP platforms are side by side and signal amplification is within its maximal range. Multiple paths are absent, because of short-range transmission.

In the GRC block diagram : File source: The transmitted bits in binary format are given as the source anf file source reads them and transmits them to USRP sink

USRP sink:The bits are communicated to the USRP-N210 device through USRP sinks. The device address is specified here.Gain,BAndwidth,Centre Frequency can be adjusted in this block

USRP source: The data received after transmission from the USRP-N210 is obtained using USRP source. Again the Gain,Bandwidth,Centre frequency are specified here.

File sink: File sink writes the received data into binary file which is used for analysing and recovering the transmitted bits.

Fig. **??** GRC block diagram

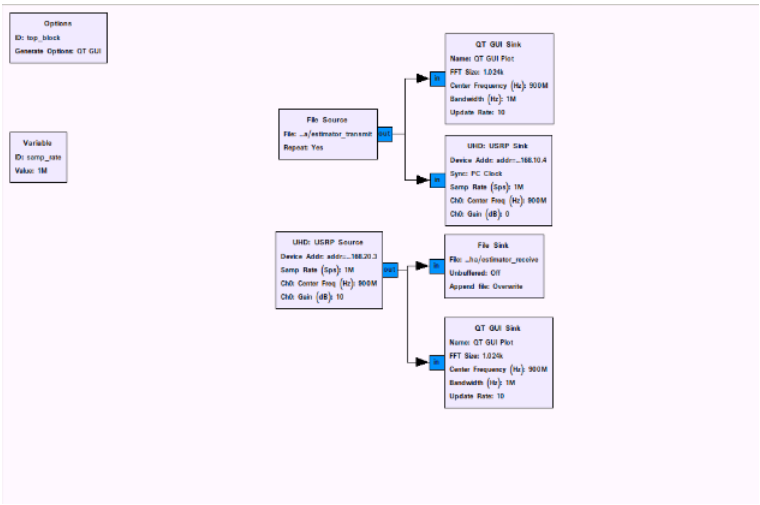

Figure 2.1: GRC block diagram

2.3 REFERENCES

1. Adrien Le Naour, Olivier Goubet, Christophe Moy, Pierre Leray:. *Spread Spectrum Channel Sounder Implementation with USRP Platforms*