Contents

ABSTRACT ii

TABLE OF FIGURES iii

TABLE OF TABLES iv

ABBREVATION v

Chapter 1 Related Knowledge 1

1.1. Quadrature Phase-shift Keying Modulation/Demodulation 1

1.1.1. Definition 1

1.1.2. Implementation 2

1.2. MIMO System and OFDM 3

1.2.1. MIMO 3

1.2.2. Introduction of OFDM 5

1.3. Space Time Block Coding (STBC) 6

1.4. Spatial Channel Model in Urban Microcell 7

1.5. Zero Forcing Equalizer 10

Chapter 2 Experiment and Results 11

2.1 . Setup Experiment 11

2.2 . Results 12

Conclusion 14

References 15

Source Code 16

ABSTRACT

In this report, we investigate the performance of space-time block coding (STBC) in non-line-of-sight urban microcell environments using a 2×2 Multiple Input Multiple Output (MIMO) system within the HiperLAN/2 framework. We specifically examine how changing the antenna spacing at the transmitter affects the correlation properties of the Spatial Channel Model (SCM). The SCM, as proposed by the 3rd Generation Partnership Project (3GPP), serves as the channel model for our simulations. Simulation results show that the Symbol Error Rate (SER) performance of STBC with a Zero-Forcing (ZF) equalizer is better than that of Vertical Bell Labs Layered Space-Time (VBLAST) with a Minimum Mean Square Error (MMSE) equalizer. Furthermore, increasing the spacing between transmitter antennas improves system performance.

TABLE OF FIGURES

Figure 1-1. Constellation of QPSK 2

Figure 1-2. QPSK waveform 2

Figure 1-3. Different models: a) Single Input/ Single Output, b) Single Input/Multiple Output, c) Multiple Input/Single Output, d) Multiple Input/Multiple Output 4

Figure 1-4. OFDM Transceiver 6

Figure 2-1. Proposed system 11

Figure 2-2. Our system results 13

TABLE OF TABLES

Table 1.1: Sub-path AoD and AoA offsets 8
Table 1.2: Lists value of those parameters in urban microcell................................. 9
Table 3. Constant variable describe channel model Urban Microcell 12

ABBREVATION

MIMO	Multiple Input Multiple Output
MISO	Multitple Input Single Output
SCM	Spatial Channel Model
SISO	Single Input Single Output
SIMO	Single Input Multiple Output
STBC	Spatial Time Block Coding
OFDM	Orthogonal Frequency Devision Multiplexing
QPSK	Quadrature Phase-shift Keying
ZF	Zero Forcing

Chapter 1 Related Knowledge

In this chapter, we will remind basic knowledge of modulation/demodulation, Multi Input Multi Output System (MIMO), Orthogonal Frequency Division Multiplexing, Space time block coding (STBC), Spatial channel model in Urban Microcell, Zero Forcing Equaliser, and our system.

1.1. Quadrature Phase-shift Keying Modulation/Demodulation

We will introduce the digital modulation QPSK in two parts: definition and implementation.

1.1.1. Definition

Quadrature phase-shift keying (QPSK) [1] is a digital modulation process which conveys data by changing (modulating) the phase of a constant frequency reference signal (the carrier wave). Sometimes this is known as 4-PSK, or 4-QAM (Although the root concepts of QPSK and 4-QAM are different, the resulting modulated radio waves are exactly the same). QPSK uses four points on the constellation diagram, equispaced around a circle. With four phases, QPSK can encode two bits per symbol, shown in the diagram with Gray coding to minimize the bit error rate (BER) – sometimes misperceived as twice the BER of BPSK.

The mathematical analysis shows that QPSK can be used either to double the data rate compared with a BPSK system while maintaining the same bandwidth of the signal, or to maintain the data-rate of BPSK but halving the bandwidth needed. In this latter case, the BER of QPSK is exactly the same as the BER of BPSK – and deciding differently is a common confusion when considering or describing QPSK. The transmitted carrier can undergo numbers of phase changes.

Given that radio communication channels are allocated by agencies such as the Federal Communication Commission giving a prescribed (maximum) bandwidth, the advantage of QPSK over BPSK becomes evident: QPSK transmits twice the data rate in a given bandwidth compared to BPSK - at the same BER. The engineering penalty that is paid is that QPSK transmitters and receivers are more complicated than the ones for BPSK. However, with modern electronics technology, the penalty in cost is very moderate.

As with BPSK, there are phase ambiguity problems at the receiving end, and differentially encoded QPSK is often used in practice.

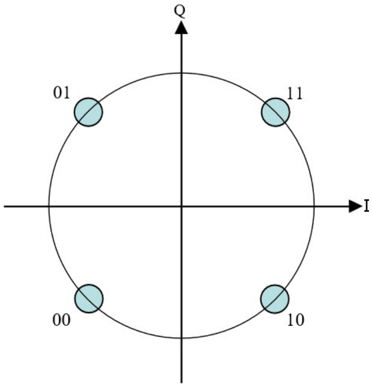

Figure 1-1. Constellation of QPSK

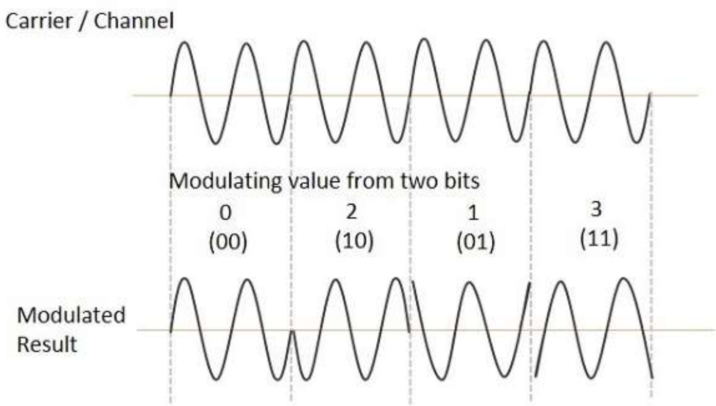

Figure 1-2. QPSK waveform

1.1.2. Implementation

The implementation of QPSK is more general than that of BPSK and also indicates the implementation of higher-order PSK. Writing the symbols in the constellation diagram in terms of the sine and cosine waves used to transmit them:

$$s_n(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f_c t + \frac{(2n-1)\pi}{4}\right), n = 1,2,3,4 \quad (1)$$

Where E_s: the power of symbol, T_s: duration of symbol. This yields the four phases π/4, 3π/4, 5π/4 and 7π/4 as need This yields the four phases π/4, 3π/4, 5π/4 and 7π/4 as needed.

This results in a two-dimensional signal space with unit basis functions:

$$\emptyset_1(t) = \sqrt{\frac{2}{T_s}} cos(2\pi f_c t) \quad (2)$$

$$\emptyset_2(t) = \sqrt{\frac{2}{T_s}} sin(2\pi f_c t) \quad (3)$$

Constellation of QPSK and QPSK waveform are showed *Figure 1-1* and *Figure 1-2* below:

1.2. MIMO System and OFDM

These are the most modern technology which are used in Forth Generation (4G), especially Fifth Generation (5G). In next part, we will briefly introduce two technologies.

1.2.1. MIMO

The multipath propagation is vital characteristic of data transmission in wireless communication systems. Wireless channel contains different impairment to transmitted signal and channel response. It affects the signal to travel in multipath between transmitter and receiver. The receiver gets the reflection of same symbols in delay versions. Delays or fading occurs due to reflection, refractions, diffractions, shadowing etc. Because of buildings, trees, aircrafts, humidity, temperature etc. Delay or fading could be in result of changing phase or magnitude of signals. The multipath affects and delay profile reduce the channel efficiency, through put and cause corrupted information at receiver. Intelligently multipath effect of MIMO [2] is used to increase capacity of system. In Rayleigh fading signal travels through different paths and considered to be follow independent behavior in every path, phase is uniformly distributed between 0 to 2 and magnitude vary randomly**.** While in Rician fading the line of sight (LOS) exists i.e. one of the paths to receiver is much stronger than other one**.** A signal or symbol of delay version have change in phase or differ in phase with line of sight signal phase. Crust and trough of both these signals cause resultant signal to be high average power or attenuated. So in result we may get distorted signal at receiver end. For antennas system .There are four basic models: Single Input Single Output (SISO), Single input Multiple Output (SIMO), Multiple Input Single Output (MISO), Multiple Input Multiple Output (MIMO), which are shown in Figure 1-3:

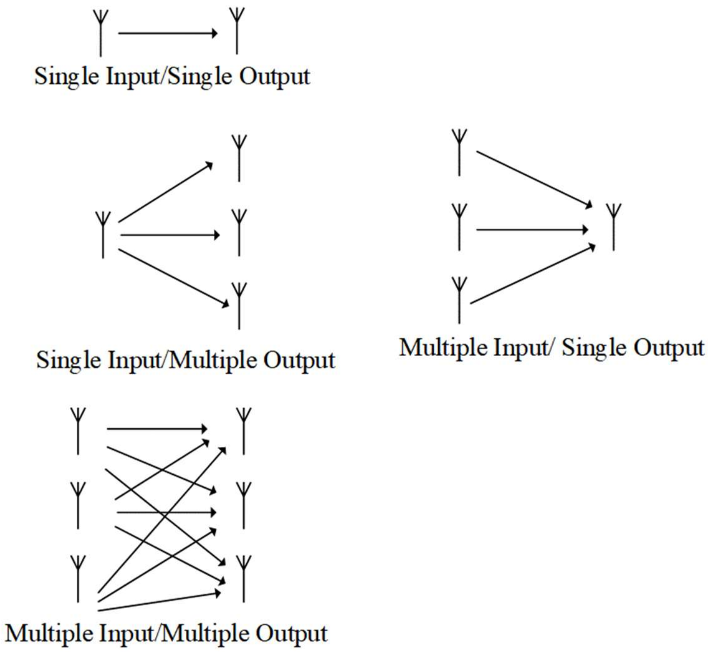

Figure 1-3. Different models: a) Single Input/ Single Output, b) Single Input/Multiple Output, c) Multiple Input/Single Output, d) Multiple Input/Multiple Output

MIMO has many advantages which are listed below:

- It gives array gain which in result of enhance the Quality of Service (QoS) and coverage area
- Higher the multiplexing gain which in result the increase of spectral efficiency
- Higher Diversity Gain, less chances to loss information ,increase QoS service
- Higher the multiplexing gain, which in result the increase in spectral efficiency

- Co–channel interference is minimizes which is helpful in increasing the cellular capacity

1.2.2. Introduction of OFDM

In telecommunications, orthogonal frequency-division multiplexing (OFDM) is a method of encoding digital data on multiple carrier frequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G mobile communications.

OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. OFDM was introduced by Chang of Bell Labs in 1966. Numerous closely spaced orthogonal sub-carrier signals with overlapping spectra are emitted to carry data. Demodulation is based on Fast Fourier Transform algorithms. OFDM was improved by Weinstein and Ebert in 1971 with the introduction of a guard interval, providing better orthogonality in transmission channels affected by multipath propagation.Each sub-carrier (signal) is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate. This maintains total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

The main advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath) without complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapidly modulated wideband signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate intersymbol interference (ISI) and use echoes and time-spreading (in analog television visible as ghosting and blurring, respectively) to achieve a diversity gain, i.e. a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs) where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be re-combined constructively, sparing interference of a traditional single-carrier system.

OFDM transceiver is showed in Figure 1-4

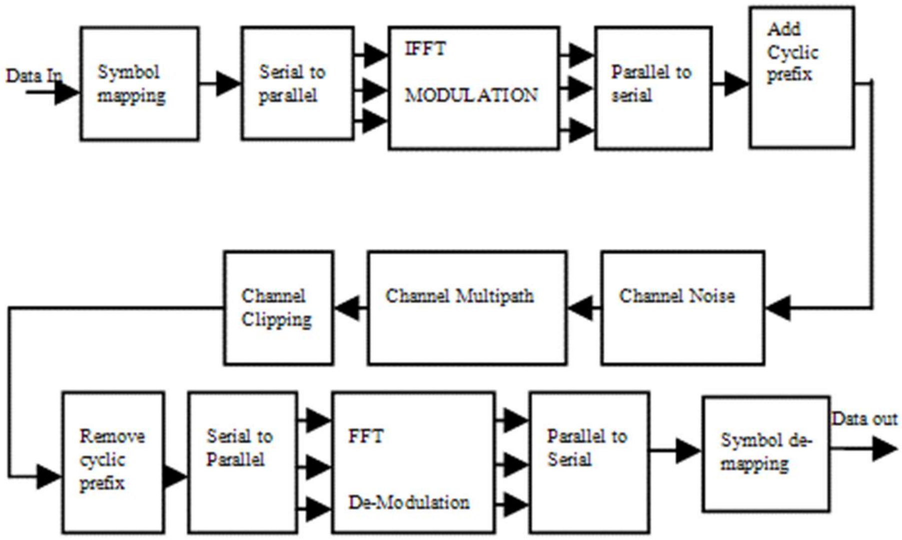

Figure 1-4. OFDM Transceiver

1.3. Space Time Block Coding (STBC)

To achieve maximum channel capacity for MIMO, STBC is better candidate. STBC is designed to achieve transmit diversity and power gain without scarifying any more bandwidth in STBC is performed over two axis spatial (space) and temporal (time) axis for multiple antenna at different time. For example, Alamouti code is show in (4)

$$X = \begin{bmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{bmatrix} \tag{4}$$

Where X is symbol after STBC block. Alamouti code is used frequently in 2x2 MIMO systems (shown in Figure 1-3, draw 2x2 MIMO in this figure). To find transmitted data, we use zero forcing equalizer as we will mention detail below

1.4. Spatial Channel Model in Urban Microcell

The scope of the 3GPP-3GPP2 SCM AHG is to develop and specify parameters and methods associated with the spatial channel modelling [3] that are common to the needs of the 3GPP and 3GPP2 organizations. The scope includes development of specifications for.

In the microcell environment, the base station antenna is usually mounted at the same height as the surrounding objects. This implies that the scattering spread of the Angle of Arrival (AoA) of the received signal at the base station is larger than in the macrocell case since the scattering process also happens in the vicinity of the base station. Thus, as the base station antenna is lowered, the tendency is for the multipath AOA spread to increase. This change in the behavior of the received signal is very important as far as antenna array applications are concerned.

Determine random delays for each of the N multipath components. Generate random variables $\tau_1, \dots, \tau_N$ are i.i.d random variables drawn from a uniform distribution from 0 to $1.2\mu s$.

Determine random average powers for each of the N multipath components. The Power Delay Profile (PDP) consists of N=6 distinct paths that are uniformly distributed between 0 and 1.2s. The powers for each path are exponentially decaying in time with the addition of a lognormal randomness, which is independent

$$P_n' = 10^{-\left(\frac{\tau_n + z_n}{10}\right)} \tag{5}$$

where τ_n is the unquantized values and given in units of microseconds, and z_n (n = 1,…,N) are i.i.d. zero mean Gaussian random variables with a standard deviation of 3dB. Average powers are normalized so that total average power for all six paths is equal to one:

$$P_n = \frac{P_n'}{\sum_{j=1}^{6} P_j'} \tag{6}$$

Determine AoDs for each of the N multipath components. The AoDs (with respect to the LOS direction) are i.i.d. random variables drawn from a uniform distribution over –40 to +40 degrees as:

$$\delta_{n,AoD} \sim U(-40\,,+40), \quad n = 1, \dots, N \tag{7}$$

Determine the AoAs for each of the multipath components. The AoAs are i.i.d Gaussian random variables as

$$\delta_{n,AoA} \sim \eta\left(0, \sigma_{n,AoA}^2\right), \qquad n \ = 1, \dots, N \tag{8}$$

Where $\sigma_{n,AoA} = 104.12(1 - \exp(-0.265|10\log_{10}(P_n)|))$ and P_n is the relative power of the nth path

Determine the antenna gains of the BS and MS sub-paths as a function of their respective sub-path AoDs and AoAs. For the nth path, the AoD of the mth sub-path (with respect to the BS antenna array broadside) is

$$\theta_{n,m,AoD} = \ \theta_{BS} + \delta_{n,AoD} + \ \Delta_{n,m,AoD} \tag{9}$$

$$\theta_{n,m,AoA} = \ \theta_{MS} + \delta_{n,AoA} + \ \Delta_{n,m,AoA} \tag{10}$$

$\Delta_{n,m,AoD}$ and $\Delta_{n,m,AoA}$ are given by Table 1.1

Table 1.1: Sub-path AoD and AoA offsets

Sub-path # (m)	Offset for a 5 deg AS at BS (microcell) $\Delta_{n,m,AoD}$ (degrees)	Offset for a 35 deg AS at MS $\Delta_{n,m,AoA}$ (degrees)
1, 2	±0.2236	±1.5649
3, 4	±0.7064	±4.9447
5, 6	±1.2461	±8.7224
7, 8	±1.8578	±13.0045
9, 10	±2.5642	±17.9492
11, 12	±3.3986	±23.7899
13, 14	±4.4220	±30.9538
15, 16	±5.7403	±40.1824
17, 18	±7.5974	±53.1816
19, 20	±10.7753	±75.4274

Table 1.2: Lists value of those parameters in urban microcell

Number of Paths (N)	6
Number of sub-paths (M) per path	20
θ_{MS}, θ_{BS}	0
v	30km/h
Per-path AS at BS (Fixed)	5 deg
Per-path AS at MS (Fixed)	35 deg

The antenna gains are dependent on these sub-path AoDs and AoAs. For the BS and MS, these are given respectively as $G_{BS}(\theta_{n,m,AoD}), G_{MS}(\theta_{n,m,AoA})$. In this report, we use 3 sector antenna.

The channel coefficients for one of N multipath components are given by matrix of U mobile station components ($u = 1, \dots, U$) and S base station components ($s = 1, \dots, S$) based on the Ricean K factor. The channel coefficient of path is given as:

$$h_{us,n} = \sqrt{\frac{P_n \sigma_{SF}}{M}} \sum_{m=1}^{M} (\sqrt{G_{BS}(\theta_{n,m,AoD})} exp\ (jkd_s\ sin(\theta_{n,m,AoD})) \times \sqrt{G_{MS}(\theta_{n,m,AoA})} exp\ (j(kd_u\ sin(\theta_{n,m,AoA}) + \phi_{n,m})) \times exp(jk\|v\|\ cos(\theta_{n,m,AoA} - \theta_v)\ t) \tag{11}$$

Where:

- P_n is the power of nth path
- σ_{SF} is the lognormal shadow fading, applied as a bulk parameter to the n paths for a given drop
- M is the number of subpaths per-path
- θ_{BS} and θ_{MS} are the AoD and the AoA of the LOS component.

- k is the number $2\pi/\lambda$ where λ is the carrier wavelength in meters
- $G(\theta)$ is the antenna gain depending on the value of the AoA and the AoD
- d_s and d_u are the distance between the antenna elements in meters at the BS side and the MS side (as REFERENCE $d_s = 10\lambda$ and $d_u = 0.5\lambda$)

1.5. Zero Forcing Equalizer

Transmitted data symbols detected by using the zero forcing (ZF) equalizer [4]with the matrix is the pseudo inverse of channel coefficients matrix. Since the transmission is done over two periods of time, the decoding will also be done over two periods of time. At the receiver, the received vector Y can be represented by the following

$$Y = \begin{bmatrix} Y_1^1 \\ Y_2^1 \end{bmatrix} = \begin{bmatrix} H_{11}^1 & H_{12}^1 \\ H_{21}^1 & H_{22}^1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} N_1^1 \\ N_2^1 \end{bmatrix} \tag{12}$$

This is for the first time period. For the second time period, the equation is as:

$$Y = \begin{bmatrix} Y_1^2 \\ Y_2^2 \end{bmatrix} = \begin{bmatrix} H_{11}^2 & H_{12}^2 \\ H_{21}^2 & H_{22}^2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} N_1^2 \\ N_2^2 \end{bmatrix} \tag{13}$$

From (12), (13) we combined and arranged to produce the following:

$$Y = \begin{bmatrix} Y_1^1 \\ Y_2^1 \\ Y_1^2 \\ Y_2^2 \end{bmatrix} = \begin{bmatrix} H_{11}^1 & H_{12}^1 \\ H_{21}^1 & H_{22}^1 \\ conj(H_{11}^2) & conj(H_{12}^2) \\ conj(H_{21}^2) & conj(H_{22}^2) \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} N_1^1 \\ N_2^1 \\ N_1^2 \\ N_2^2 \end{bmatrix} \tag{14}$$

By ZF, we detect transmitted symbols by (15), we ignore noise because it does not matte

$$\hat{X} = [\begin{matrix} \widehat{X_1} \\ \widehat{X_2} \end{matrix}] = (H^H H)^{-1} H^H \begin{bmatrix} Y_1^1 \\ Y_2^1 \\ Y_1^2 \\ Y_2^2 \end{bmatrix} \tag{15}$$

Where $\hat{X}$ is the predicted symbols at n[th] subcarrier, $(.)^H$ designates the Hermitian transpose matrix.

Chapter 2 Experiment and Results

In this chapter, we will proposed our simulation model and results.

2.1.1. Setup Experiment

Our system is shown in Figure 2-1. The system include: random data input, QPSK Modulation/Demodulation, Alamouti Code for Space Time block coding (2x2 Antenna), channel we transfer signal is urban microcell. Besides, we use zero Forcing Equalizer to estimate the received signal and calculate SER.

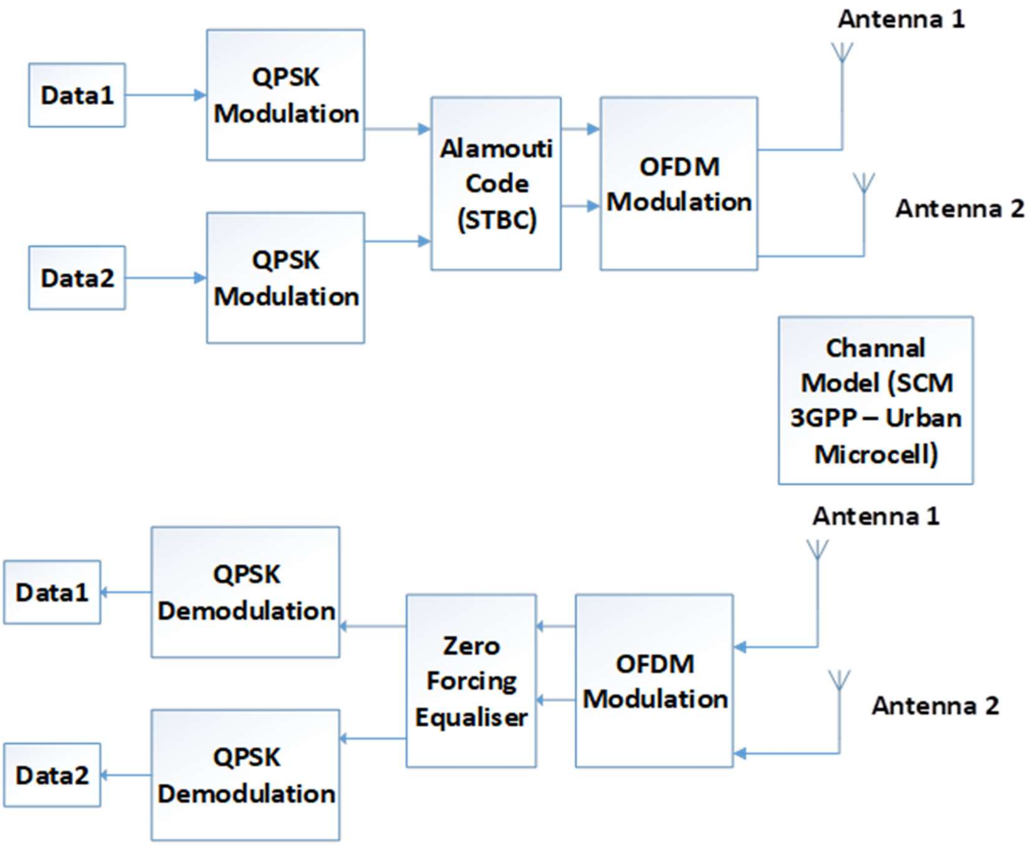

Figure 2-1. Proposed system

Our system is programmed by MATLAB. We will compare 2 SER of ds = 4, du = ½, and ds = ½, du = ½ . SER will be calculated averagely by 10 times. We will summerize here some constant variable's values which are used in our system.

Table 3. Constant variable describe channel model Urban Microcell

	Definition	**Value**
N	Number of Path	6
M	Number of Sub-path	20
omegaBS	The angle between the North and the BS antenna	0
omegaMS	The angle between the North and the MS antenna	0
thetaBS	The angle between the BS antenna and the LOS connection	60
thetaMS	The angle between the MS antenna and the LOS connection	60
V	Veclocity of MS	30 km/h
thetaV	Vector of v	0
sigmaSF	Lognomal shadowing standard deviation	10
A_m	3 Sector antenna maximum attenuation in dB	20
Theta_3dB	3 Sector 3 dB beamwidth in degrees	70

2.2. Results

The results of our system in Urban Microcell with NLOS case are shown in Figure 2-2. Our result is as same plot's trend as from [5]. Obviously, the increasing the spacing in BS antenna gives better SER. Considering case ds = 10λ, du $= \frac{1}{2}\lambda$, SER $= 10^{-1}$, our system (SBTC-ZF) has SNR equal 22.3dB better than the system (VBLAST-MMSE) in [5] equal 26.6dB.

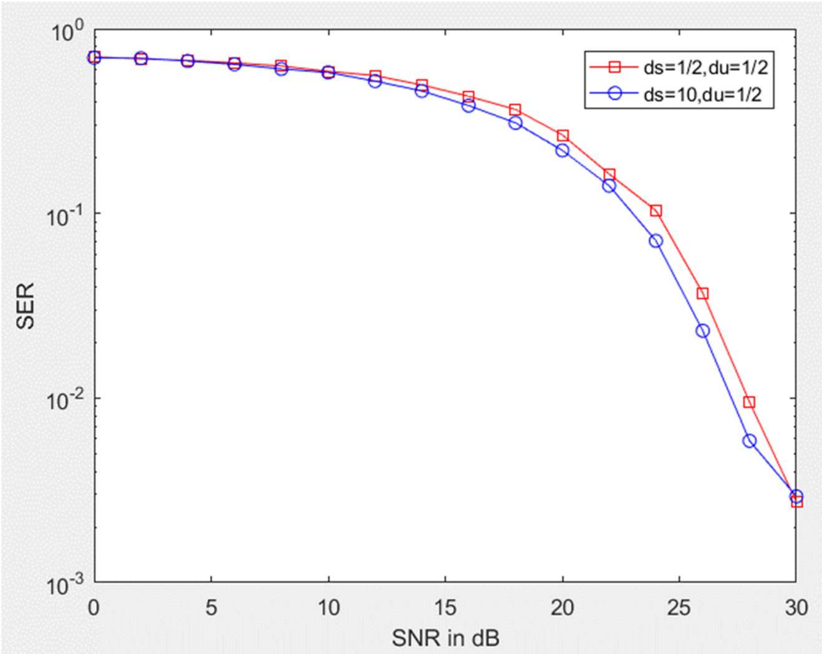

Figure 2-2. Our system results

Conclusion

In this study, we demonstrated that STBC with a ZF equalizer outperforms VBLAST with a MMSE equalizer in NLOS urban microcell environments using a 2×2 MIMO system within the HiperLAN/2 system. Our simulations, based on the SCM proposed by the 3GPP, showed that increasing the spacing between transmitter antennas improves system performance. For future work, we plan to implement space-frequency block coding (SFBC) and explore advanced equalizer algorithms such as MMSE and Maximum Likelihood (ML) to further investigate and enhance system performance in these environments.

References

[1] S. Bernard, "BandPass Modulation and Demodulation," in *Digital Communications. Fundamentals and Applications*, 2014.

[2] H. Lipfert, " MIMO OFDM Space Time Coding – Spatial Multiplexing, Increasing Performance and Spectral Efficiency in Wireless Systems, Part I Technical Basis," Institut für Rundfunktechnik, 2007.

[3] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Spatial channel model for Multiple Input Multiple Output (MIMO) simulations (Release 9)," Valbonne - FRANCE , 2009.

[4] "nutaq," [Online]. Available: https://www.nutaq.com/blog/alamouti-space-time-block-coding. [Accessed 07 04 2019].

[5] N. T. Nga, N. Van Duc, J. Byeungwoo and N. Quy Sy, "An investigation of the spatial correlation influence on coded MIMO-OFDMA system performance," in *The 12th International Conference*, 2018.

Source Code

Main.file

```
clc;
snrMin=0.0;
snrMax = 30.0;
step = 2.0;
NT=10;   % change number of loop, number of transmission
%Frequency Carrier
lambda = 3*10^8/(5*10^9);
%Sampling rate of HiperLAN2 - frequency 5GHz
ta = 50*10^-9;
%QPSK
M_ary = 4;
%Initialize constant variable of OFDM
NFFT = 512;         % FFT length
G = 212;            % Guard interval length
%intialize constant variable for baseband data
lengthData = NFFT;  %The total data's length
%OFDM Symbol Duration
% To avoid conflict in reality system
symbolDuration = NFFT*ta;%TS = t_a*NFFT: formular in page 29
%Value of Channel NLOS Microcell
load ParametersForSpatialChannelModel_UrbanMicroCell.mat;
%Note: in future maybe change this code into 1 variable and seperate it
%later
%Source
sourceBitDataAntenna1 = randi([0 1],lengthData,log2(M_ary));
sourceBitDataAntenna2 = randi([0 1],lengthData,log2(M_ary));
%bit to decimal to use qammod
sourceAntenna1 = bi2de(sourceBitDataAntenna1);
sourceAntenna2 = bi2de(sourceBitDataAntenna2);
%QPSK Block: Because 4QAM = QPSK
QAMSymbolAntenna1 = qammod(sourceAntenna1, M_ary);
QAMSymbolAntenna2 = qammod(sourceAntenna2, M_ary);
% STBC Encoder and Preparing data pattern
% x1: data Antenna 1, x2: data Antenna 2
% Alamouti Code: x = [x1 -x2*;x2 x1*]
% dataPatternAntenna1v1 = [x1, -x2*];
dataPatternAntenna1 = [QAMSymbolAntenna1,-conj(QAMSymbolAntenna2)]; %512x2
dataPatternAntenna1 = dataPatternAntenna1.'; %dataPatternAntenna1: [x1
                                    %                    -x2*] ->
                                    %                    2x512
% dataPatternAntenna2 = [x2 x1*]
dataPatternAntenna2 = [QAMSymbolAntenna2, conj(QAMSymbolAntenna1)];
dataPatternAntenna2 = dataPatternAntenna2.';
SER_R = [];
%Get Value from Paper 54
SER=[];
ds = 1/2;
du = 1/2;
for snr = snrMin:step:snrMax
   SER_t = 0;
%     snr = snr - 10*log((NFFT-G)/NFFT);
   SNR = 10^(snr/10);%dB -> times (l?n)
   for t_i = 1:NT
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % Transmitter
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      t = rand(1)*1000;
      t_init = t;
      %saving value of channel into matrix size have 2 rows (2 time slot)
      rs11TimeSlot = [];
      rs12TimeSlot = [];
      rs21TimeSlot = [];
      rs22TimeSlot = [];
```

```
h11TimeSlot = [];
h12TimeSlot = [];
h21TimeSlot = [];
h22TimeSlot = [];
%number of  timeslot is 2
NoTimeSlot = 2;
for i=1:NoTimeSlot
   % OFDM Block
   OFDMSignalAntenna1 = OFDM_Modulator(dataPatternAntenna1(i,:),NFFT,G);
   OFDMSignalAntenna2 = OFDM_Modulator(dataPatternAntenna2(i,:),NFFT,G);
   % Find channel Model
   h11 = SpatialChannelModelBMC(1,1,du,ds,t,N,M,Pn,...
                         sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                         theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
   h12 = SpatialChannelModelBMC(1,2,du,ds,t,N,M,Pn,...
                         sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                         theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
   h21 = SpatialChannelModelBMC(2,1,du,ds,t,N,M,Pn,...
                         sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                         theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
   h22 = SpatialChannelModelBMC(2,2,du,ds,t,N,M,Pn,...
                         sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                         theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
   %Singal pass channel
   % Receive at antenna 1
   RxS11 = conv(OFDMSignalAntenna1, h11);
   RxS12 = conv(OFDMSignalAntenna2, h12);
   % Receive at antenna 2
   RxS21 = conv(OFDMSignalAntenna1, h21);
   RxS22 = conv(OFDMSignalAntenna2, h22);
   % The received signal over white gauss noise
   RxS11 = awgn(RxS11,snr-20,'measured','dB');
   RxS12 = awgn(RxS12,snr-20,'measured','dB');
   RxS21 = awgn(RxS21,snr-20,'measured','dB');
   RxS22 = awgn(RxS22,snr-20,'measured','dB');
   %save Value of Rxsignal and h channel
   h11TimeSlot = [h11TimeSlot; h11];
   h12TimeSlot = [h12TimeSlot; h12];
   h21TimeSlot = [h21TimeSlot; h21];
   h22TimeSlot = [h22TimeSlot; h22];
   rs11TimeSlot = [rs11TimeSlot; RxS11];
   rs12TimeSlot = [rs12TimeSlot; RxS12];
   rs21TimeSlot = [rs21TimeSlot; RxS21];
   rs22TimeSlot = [rs22TimeSlot; RxS22];
   t = t + 1000*symbolDuration;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Receiver
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% At first Time slot
RxSAntenna1TimeSlot1 = rs11TimeSlot(1,:)+ rs12TimeSlot(1,:);% Rx:Antenna1, Tx: A1, A2
RxSAntenna2TimeSlot1 = rs21TimeSlot(1,:)+ rs22TimeSlot(1,:);% Rx:A2,       Tx: A1, A2
% At second Time slot
RxSAntenna1TimeSlot2 = rs11TimeSlot(2,:)+ rs12TimeSlot(2,:);% Rx:Antenna1, Tx: A1, A2
RxSAntenna2TimeSlot2 = rs21TimeSlot(2,:)+ rs22TimeSlot(2,:);% Rx:A2,       Tx: A1, A2
% OFDM Demodulator
DemodulatedSignalAntenna1TimeSlot1 = OFDM_Demodulator(RxSAntenna1TimeSlot1,NFFT,NFFT,G);
DemodulatedSignalAntenna2TimeSlot1 = OFDM_Demodulator(RxSAntenna2TimeSlot1,NFFT,NFFT,G);
DemodulatedSignalAntenna1TimeSlot2 = OFDM_Demodulator(RxSAntenna1TimeSlot2,NFFT,NFFT,G);
DemodulatedSignalAntenna2TimeSlot2 = OFDM_Demodulator(RxSAntenna2TimeSlot2,NFFT,NFFT,G);
%Zero Forcing Equalizer
% Because we need to convert from time to frequency by fft
% (convolution to multiplication)
H11TimeSlot1 = fft([h11TimeSlot(1,:),zeros(1,NFFT-length(h11TimeSlot(1,:)))]);
H12TimeSlot1 = fft([h12TimeSlot(1,:),zeros(1,NFFT-length(h12TimeSlot(1,:)))]);
H21TimeSlot1 = fft([h21TimeSlot(1,:),zeros(1,NFFT-length(h21TimeSlot(1,:)))]);
H22TimeSlot1 = fft([h22TimeSlot(1,:),zeros(1,NFFT-length(h22TimeSlot(1,:)))]);
H11TimeSlot2 = fft([h11TimeSlot(2,:),zeros(1,NFFT-length(h11TimeSlot(2,:)))]);
H12TimeSlot2 = fft([h12TimeSlot(2,:),zeros(1,NFFT-length(h12TimeSlot(2,:)))]);
H21TimeSlot2 = fft([h21TimeSlot(2,:),zeros(1,NFFT-length(h21TimeSlot(2,:)))]);
```

```
        H22TimeSlot2 = fft([h22TimeSlot(2,:),zeros(1,NFFT-length(h22TimeSlot(2,:)))]);
        signalEst = [];
        for m = 1:NFFT
            H_tmp = [H11TimeSlot1(m), H12TimeSlot1(m);...
                     H21TimeSlot1(m), H22TimeSlot1(m);...
                     conj(H12TimeSlot2(m)), -conj(H11TimeSlot2(m)); ...
                     conj(H22TimeSlot2(m)), -conj(H21TimeSlot2(m))];
            W = inv(transpose(conj(H_tmp))*H_tmp)*transpose(conj(H_tmp));%2x4
            r = [DemodulatedSignalAntenna1TimeSlot1(m); DemodulatedSignalAntenna2TimeSlot1(m);...
                conj(DemodulatedSignalAntenna1TimeSlot2(m)); conj(DemodulatedSignalAntenna2TimeSlot2(m))];%4x1
            signalEstTemp = W*r;
            signalEst = [signalEst, signalEstTemp];
        end
        s1 = qamdemod(signalEst(1,:),M_ary);
        s2 = qamdemod(signalEst(2,:),M_ary);
        [number1, ratio1] = symerr(sourceAntenna1,s1');
        [number2, ratio2] = symerr(sourceAntenna2,s2');
        SER_R = (ratio1+ratio2)/2;
        SER_t=SER_t+SER_R;
    end
    tE=SER_t/t_i; %NT = t_i
    SER = [SER, tE];
end
%save and plot
psnr = snrMin:step:snrMax;
semilogy(psnr, SER,'rs-');
hold on
%Get Value from Paper 54
SER=[];
ds = 4;
du = 1/2;
%Value of Channel LOS Microcell
load ParametersForSpatialChannelModel_UrbanMicroCell.mat;
N = 6;
for snr = snrMin:step:snrMax
    SER_t = 0;
%     snr = snr - 10*log((NFFT-G)/NFFT);
    SNR = 10^(snr/10);
    for t_i = 1:NT
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Transmitter
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%
        t = rand(1)*1000;
        t_init = t;
        %saving value of channel into matrix size have 2 rows (2 time slot)
        rs11TimeSlot = [];
        rs12TimeSlot = [];
        rs21TimeSlot = [];
        rs22TimeSlot = [];
        h11TimeSlot = [];
        h12TimeSlot = [];
        h21TimeSlot = [];
        h22TimeSlot = [];
        %number of  timeslot is 2
        NoTimeSlot = 2;
        for i=1:NoTimeSlot
            OFDMSignalAntenna1 = OFDM_Modulator(dataPatternAntenna1(i,:),NFFT,G);
            OFDMSignalAntenna2 = OFDM_Modulator(dataPatternAntenna2(i,:),NFFT,G);
            % h11 = SpatialChannelModel(1,1,du,ds,t,N,M,Pn,...
            %                         sigma_SF,G_BS,G_MS,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
            %                         theta_BS,theta_MS,theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,Phi_LOS,v,theta_v,K,lambda);
            % h12 = SpatialChannelModel(1,2,du,ds,t,N,M,Pn,...
            %                         sigma_SF,G_BS,G_MS,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
            %                         theta_BS,theta_MS,theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,Phi_LOS,v,theta_v,K,lambda);
            % h21 = SpatialChannelModel(2,1,du,ds,t,N,M,Pn,...
            %                         sigma_SF,G_BS,G_MS,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
            %                         theta_BS,theta_MS,theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,Phi_LOS,v,theta_v,K,lambda);
            % h22 = SpatialChannelModel(2,2,du,ds,t,N,M,Pn,...
            %                         sigma_SF,G_BS,G_MS,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
            %                         theta_BS,theta_MS,theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,Phi_LOS,v,theta_v,K,lambda);
```

```
    h11 = SpatialChannelModelBMC(1,1,du,ds,t,N,M,Pn,...
                    sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                    theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
    h12 = SpatialChannelModelBMC(1,2,du,ds,t,N,M,Pn,...
                    sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                    theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
    h21 = SpatialChannelModelBMC(2,1,du,ds,t,N,M,Pn,...
                    sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                    theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
    h22 = SpatialChannelModelBMC(2,2,du,ds,t,N,M,Pn,...
                    sigmaSF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,...
                    theta_n_m_AoD,theta_n_m_AoA,Phi_n_m,v,thetaV,lambda);
    %Singal pass channel
    RxS11 = conv(OFDMSignalAntenna1, h11);
    RxS12 = conv(OFDMSignalAntenna2, h12);
    RxS21 = conv(OFDMSignalAntenna1, h21);
    RxS22 = conv(OFDMSignalAntenna2, h22);
    % The received signal over multhipath channel is created
    RxS11 = awgn(RxS11,snr-20,'measured','dB');
    RxS12 = awgn(RxS12,snr-20,'measured','dB');
    RxS21 = awgn(RxS21,snr-20,'measured','dB');
    RxS22 = awgn(RxS22,snr-20,'measured','dB');
    %save Value of Rxsignal and h channel
    h11TimeSlot = [h11TimeSlot; h11];
    h12TimeSlot = [h12TimeSlot; h12];
    h21TimeSlot = [h21TimeSlot; h21];
    h22TimeSlot = [h22TimeSlot; h22];
    rs11TimeSlot = [rs11TimeSlot; RxS11];
    rs12TimeSlot = [rs12TimeSlot; RxS12];
    rs21TimeSlot = [rs21TimeSlot; RxS21];
    rs22TimeSlot = [rs22TimeSlot; RxS22];
    t = t + 1000*symbolDuration;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Receiver
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% At first Time slot
RxSAntenna1TimeSlot1 = rs11TimeSlot(1,:)+ rs12TimeSlot(1,:);% Rx:Antenna1, Tx: A1, A2
RxSAntenna2TimeSlot1 = rs21TimeSlot(1,:)+ rs22TimeSlot(1,:);% Rx:A2,       Tx: A1, A2
% At second Time slot
RxSAntenna1TimeSlot2 = rs11TimeSlot(2,:)+ rs12TimeSlot(2,:);% Rx:Antenna1, Tx: A1, A2
RxSAntenna2TimeSlot2 = rs21TimeSlot(2,:)+ rs22TimeSlot(2,:);% Rx:A2,       Tx: A1, A2
% OFDM Demodulator
DemodulatedSignalAntenna1TimeSlot1 = OFDM_Demodulator(RxSAntenna1TimeSlot1,NFFT,NFFT,G);
DemodulatedSignalAntenna2TimeSlot1 = OFDM_Demodulator(RxSAntenna2TimeSlot1,NFFT,NFFT,G);
DemodulatedSignalAntenna1TimeSlot2 = OFDM_Demodulator(RxSAntenna1TimeSlot2,NFFT,NFFT,G);
DemodulatedSignalAntenna2TimeSlot2 = OFDM_Demodulator(RxSAntenna2TimeSlot2,NFFT,NFFT,G);
%Zero Forcing Equalizer
H11TimeSlot1 = fft([h11TimeSlot(1,:),zeros(1,NFFT-length(h11TimeSlot(1,:)))]);
H12TimeSlot1 = fft([h12TimeSlot(1,:),zeros(1,NFFT-length(h12TimeSlot(1,:)))]);
H21TimeSlot1 = fft([h21TimeSlot(1,:),zeros(1,NFFT-length(h21TimeSlot(1,:)))]);
H22TimeSlot1 = fft([h22TimeSlot(1,:),zeros(1,NFFT-length(h22TimeSlot(1,:)))]);
H11TimeSlot2 = fft([h11TimeSlot(2,:),zeros(1,NFFT-length(h11TimeSlot(2,:)))]);
H12TimeSlot2 = fft([h12TimeSlot(2,:),zeros(1,NFFT-length(h12TimeSlot(2,:)))]);
H21TimeSlot2 = fft([h21TimeSlot(2,:),zeros(1,NFFT-length(h21TimeSlot(2,:)))]);
H22TimeSlot2 = fft([h22TimeSlot(2,:),zeros(1,NFFT-length(h22TimeSlot(2,:)))]);
signalEst = [];
for m = 1:NFFT
    H_tmp = [H11TimeSlot1(m), H12TimeSlot1(m);...
           H21TimeSlot1(m), H22TimeSlot1(m);...
           conj(H12TimeSlot2(m)), -conj(H11TimeSlot2(m)); ...
           conj(H22TimeSlot2(m)), -conj(H21TimeSlot2(m))];
    W = inv(transpose(conj(H_tmp))*H_tmp)*transpose(conj(H_tmp));%2x4
    r = [DemodulatedSignalAntenna1TimeSlot1(m); DemodulatedSignalAntenna2TimeSlot1(m);...
        conj(DemodulatedSignalAntenna1TimeSlot2(m)); conj(DemodulatedSignalAntenna2TimeSlot2(m))];%4x1
    signalEstTemp = W*r;
    signalEst = [signalEst, signalEstTemp];
end
s1 = qamdemod(signalEst(1,:),M_ary);
s2 = qamdemod(signalEst(2,:),M_ary);
```

```
        [number1, ratio1] = symerr(sourceAntenna1,s1');
        [number2, ratio2] = symerr(sourceAntenna2,s2');
        SER_R = (ratio1+ratio2)/2;
        SER_t=SER_t+SER_R;
    end
    tE=SER_t/t_i;  %NT = t_i
    SER = [SER, tE];
end
psnr = snrMin:step:snrMax;
semilogy(psnr, SER,'bo-');
xlabel('SNR in dB');
ylabel('SER');
legend('ds=1/2,du=1/2','ds=4,du=1/2');
```

SpatialChannelModel File

```
function [h] =
SpatialChannelModelBMC(u,s,d_u,d_s,t,N,M,Pn,sigma_SF,G_BS_theta_n_m_AoD,G_MS_theta_n_m_AoA,theta_n_m_AoD,theta_n_m_Ao
A,Phi_n_m,v,theta_v,lambda)
    k = 2*pi/lambda;
    h = [];
    if(u==1)
        d_u = 0;
    end
    if(s==1)
        d_s = 0;
    end
    for n = 1:N
        sum = 0;
                    for m = 1:M
                            sum = sum + ...
            sqrt(G_BS_theta_n_m_AoD(n,m))*...
                                    sqrt(G_MS_theta_n_m_AoA(n,m))*...
            exp(1j*k*d_s*sind(theta_n_m_AoD(n,m)))*...
            exp(1j*k*d_u*sind(theta_n_m_AoA(n,m)))*...
            exp(1j*Phi_n_m(n,m))*...
                                    exp(1j*k*abs(v)*cosd(theta_n_m_AoA(n,m)-theta_v)*t);
                    end
                    h(n) = sqrt(Pn(n)*sigma_SF/M)*sum;
    end
end
```

defineParameterChannel File

```
% This script file will generate all the parameters
% used in generating Channel Coefficients
clc;
clear all;
%Number of Path N and sub-path M:
N = 6;        % number of paths
M = 20;        % number of subpaths per path
omegaBS = 0;% The angle between the North and the BS antenna
omegaMS = 0;% The angle between the North and the MS antenna
thetaBS = 60;% angle between the BS antenna and the LOS connection
thetaMS = 60;% angle between the MS antenna and the LOS connection
%velocity of MS
v = 30*1000/3600;%km/h->m/s
thetaV = 0;
%lognormal shadowing standard deviation
sigmaSF = 10; %dB (NLOS)
% Determine the random delays for each of the N multipath components
% Read Page 20 file SCM 3GPP
```

```
tau_n = 1.2*rand([1,N]);
tau_n = sort(tau_n);
Tc = 1/(3.84*10^6); %Tc-second
tau_n = (Tc/16)*floor((tau_n-tau_n(1))/(Tc/16)+0.5);
% Determine random average powers for each of the N multipath components
standardVariationZn = 3;%3dB
zn = standardVariationZn.*randn(1,N);% size zn:1x6, with standard variation 3dB
Pnprime = 10.^((-tau_n+zn)/10);
Pn = Pnprime/(sum(Pnprime));
% Determine  AoDs for each of the N multipath components
standardVariationdelta_n_AoD = 40;
delta_n_AoD = standardVariationdelta_n_AoD.*randn(1,N);
% Determine the AoAs for each of the multipath components
sigma_n_AoA = 104.12*(1-exp(-0.2175*abs(10*log(Pn))));
delta_n_AoA = sigma_n_AoA.^2.*rand(1,N);
% Associate the multipath delays with AoDs
% Phase of the mth subpath of the nth path
Phi_n_m = 2*pi*rand(N,M);
Delta_n_m_AoD_5deg = [0.2236,-0.2236,0.7064,-0.7064,1.2461,-1.2461,1.8578,-1.8578,2.5642,-2.5642,3.3986,-3.3986,4.4220,-
4.4220,5.7403,-5.7403,7.5974,-7.5974,10.7753,-10.7753];
Delta_n_m_AoA_35deg = [1.5649,-1.5649,4.9447,-4.9447,8.7224,-8.7224,13.0045,-13.0045,17.9492,-17.9492,23.7899,-23.7899,30.9538,-
30.9538,40.1824,-40.1824,53.1816,-53.1816,75.4274,-75.4274];
% Determine the antenna gains of
% the BS and MS sub-paths as a function of their
% respective sub-path AoDs and AoAs
for n=1:1:N
        for m = 1:1:M
                theta_n_m_AoD(n,m) = thetaBS+delta_n_AoD(n)+Delta_n_m_AoD_5deg(m);
                theta_n_m_AoA(n,m) = thetaMS+delta_n_AoA(n)+Delta_n_m_AoA_35deg(m);
    A_m=20;       %3 Sector antenna maximum attenuation in dB
        theta_3dB=70;   %3 Sector 3 dB beamwidth in degrees
    if (theta_n_m_AoD(n,m)>180)
      theta_n_m_AoDTemp(n,m) = 360-theta_n_m_AoD(n,m);
    elseif (theta_n_m_AoD(n,m)<-180)
      theta_n_m_AoDTemp(n,m) = 360+theta_n_m_AoD(n,m);
    else
      theta_n_m_AoDTemp(n,m) = theta_n_m_AoD(n,m);
    end
    A_theta_n_m_AoD(n,m) = -min((12*(theta_n_m_AoDTemp(n,m)/theta_3dB)^2),A_m);
    G_BS_theta_n_m_AoD(n,m) = 10^(A_theta_n_m_AoD(n,m)/10);
    if theta_n_m_AoA(n,m)>180
      theta_n_m_AoATemp(n,m) = 360-theta_n_m_AoA(n,m);
    elseif theta_n_m_AoA(n,m)<-180
      theta_n_m_AoATemp(n,m) = 360+theta_n_m_AoA(n,m);
    else
      theta_n_m_AoATemp(n,m) = theta_n_m_AoA(n,m);
    end
    A_theta_n_m_AoA(n,m) = -min((12*(theta_n_m_AoATemp(n,m)/theta_3dB)^2),A_m);
        G_MS_theta_n_m_AoA(n,m) = 10^(A_theta_n_m_AoA(n,m)/10);
        end
end
save ParametersForSpatialChannelModel_UrbanMicroCell.mat v N M sigmaSF Pn thetaV Phi_n_m theta_n_m_AoD theta_n_m_AoA
G_BS_theta_n_m_AoD G_MS_theta_n_m_AoA
```

Ofdm_modulator file

```
%-----------------------------------------------------------------------
%  OFDM modulator
%  NFFT: FFT length
%  chnr: number of subcarrier
%  G: guard length
%------------------------------------------------------------------------
function [y] = OFDM_Modulator(data,NFFT,G)
chnr = length(data);
x = [data,zeros(1,NFFT - chnr)]; %Zero padding
```

```
a = ifft(x); % should be: fft a = sqrt(N)*ifft(x);
y = [a(NFFT-G+1:NFFT),a]; % insert the guard interval
```

Ofdm_demodulator file

```
%--------------------------------------------------------------------------
%  OFDM modulator
%  NFFT: FFT length
%  chnr: number of subcarrier
%  G: guard length
%  N_P: channel impulse response length
%--------------------------------------------------------------------------
function [y] = OFDM_Demodulator(data,chnr,NFFT,G)
x_remove_guard_interval = data(G+1:NFFT+G); % remove the guard interval
x = fft(x_remove_guard_interval); % should be: x = (1/sqrt(N))*fft(x_remove_guard_interval);
y = x(1:chnr); %Zero removing
```