***Abstract*—OFDM (Orthogonal Frequency Division Multiplexing) is an efficient multi-carrier solution over the air interface. Frequency is a limited resource and hence it is regulated. Orthogonality gives the advantage for closely spaced or overlapped sub-channels, can still be resolved. OFDM is an important technology for many developed and developing communications standards which require high throughput and multi-path advantages as much as possible. The air interface is an ever changing channel. Fading and noise, amongst other parameters, affect the efficient use of this channel. Thus, there are developed wireless channel models, and optimized over time, to improve on the efficient use of this channel. On this part of the project we focus on the simulation of an OFDM communications system and its parameters.**

***Index Terms*—OFDM, cyclic prefix, Rayleigh fading, zero forcing, Matlab.**

I. Introduction

THis is the report for group C based on part II of the project. The report is about the design and simulation of an OFDM communication system over an AWGN channel and also over a time-varying frequency selective channel. This report is arranged in sections sequentially, which contains concise explanations and answers to the related questions.
The main task of this project is to demonstrate the simulation of OFDM and how to choose the relevant parameters. The simulation is implemented in MATLAB.

2nd March, 2020

II. Part 2

A. Background information

The essence of the OFDM technology is for a more efficient use of our frequency spectrum in a multi-carrier, multi-access environment. This is modelled with certain constraints and considerations in mind. QPSK modulation is adopted, which offers 4 symbols with two bits per symbol. There is always some noise in communication systems and the effect should be

.
.
.

minimized for accurate detection of the transmitted symbols. Interference should be avoided or greatly minimized between symbols or between sub-carriers. This is managed by using cyclic prefix which acts as a guard-band. Also, the transmission is achieved with the bandwidth of each sub-carrier being lower than the coherent bandwidth. Furthermore, the OFDM symbol duration should be far lower than the coherence time of the channel. As long as there is mobility of the transmitter/receiver or both, there will be a Doppler shift in frequency.
The transmitted symbols have an Energy E. For the energy to be the same value for all symbols, there will be a mathematical relationship.
The communication system we are to design is to operate at $2GHz$ carrier frequency with a bandwidth of $1MHz$. So we need to efficiently use this bandwidth to have usable number of sub-channels. The noise spectral density at the receiver is $N_0 = 2.07 \times 10^{-14} \mu W/Hz$. The wireless link has a path-loss component of $101dB$. The effect of shadowing is negligible. The speed of receiver is $15\frac{m}{s}$ and the delay spread is $5.4\mu s$. Coherence bandwidth is approximately the inverse of the delay spread, which is $185.19KHz$, or $0.18519MHz$. A communication system for a fading channel is designed, built on the assumption that the tap gains $c_l(nT_s)$ are i.i.d. Rayleigh fading with Clarke's spectrum and a flat power delay profile. The simulation will be done over two differently modelled channels: the fading channel with AWGN and the pure AWGN channel.

B. Steps of implementing the OFDM system

1) Transmitter: At this stage the aim is to generate a sequence of bits $2N$, modulate them using QPSK constellation where each two bits correspond to one QPSK symbol with real and imaginary part, and thus form a sequence of QPSK symbols $s_k^{(m)}$ of length N. Each m-th sequence should satisfy an energy constraint $E\{|s_k^{(m)}|^2\} = E$. To get the signal in the time domain an IFFT is applied:

$$z_n^{(m)} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} s_k^{(m)} exp(j2\pi \frac{nk}{N}), n = 0, 1, ..., N-1 \quad (1)$$

After this step. a cyclic prefix (CP) is added to each m-th sequence of QPSK symbols in time domain, which forms an OFDM symbol. The cyclic prefix preserves the periodicity of the transmitted signal, and consequently it eliminates inter-symbol interference between successive symbols, as well as it adds robustness to the OFDM signal. After adding a cyclic prefix of length $N_{cp} \geq 0$, we get the vector of length $N+N_{cp}$ with $z_{-k}^{(m)} = z_{N-k}^{(m)}$, for $k = 1, .., N_{cp}$. Neglecting the D/A and A/D conversion used in an ideal system, we are working with a band-limited signal of a shape of a square pulse of duration T_s, where $T_s = 1/F_s$, so our complex numbers $z_k^{(m)}$ are sent sequentially over the channel at a rate $1/T_s$. Ideally, the higher the number of sub-carriers, the higher the possible system's rate of data transmission, but this also adds additional complexity since we are working in a band-limited regime.

2) Channel: We use a flat fading Rayleigh channel which we generated in the Project Part I. The channel during OFDM symbol m is denoted by $c_l^{(m)}(nT_s)$, $l = 0, 1, ..., L-1$ and $n = -N_{cp}, ..., N-1$. We denote L as a number of channel taps, set such that LT_s is larger than a delay spread τ_{DS} of the underlying physical channel. While designing the OFDM system the following constrains are kept in mind:

- The channel should be constant during an OFDM symbol duration, meaning that an OFDM symbol time duration, T_o, should be much shorter than the coherence time, T_{coh}. This is satisfied when $(N + N_{cp})f_D T_s \ll 1$. Otherwise the channel changes over an OFDM signal which introduces errors.
- The delay spread of the channel, τ_{DS}, should be smaller than the cyclic prefix duration: $N_{cp} \geq L-1$.

When simulating the channel we are adding complex AWGN samples with variance N_o to the received signal. This mimics the noise component of the real channel. We have calculated N_o as multiplication of the noise spectral density and the bandwidth, B, as it shown in equation (9).

3) Receiver: A received signal is a combination of the transmitted signal and the various channel effects, plus noise. In this project we are working with discrete-time signals. First, we are removing the cyclic prefix and then applying FFT to the remaining N samples for each OFDM symbol to receive the data in the frequency domain and obtain samples of the form:

$$y_n^{(m)} = C_n^{(m)} s^{(m)} + w_n^{(m)} \quad (2)$$

where $C_n^{(m)}$ is the FFT of $c_l^{(m)}(nT_s)$ and $w_n^{(m)}$ is the FFT of the noise. In wireless communications there are physical processes such as scattering, diffraction, reflection, etc., which bring distortions to the transmitted signal leading to residual scalings and rotations. Therefore, our aim is to get rid of the channel effect at the receiver side. We can do this by performing zero-forcing equalization in frequency domain. This basically involves multiplying the received signal with the inverse of the channel effect. This is thus represented:

$$Y(f) = H(f)S(f) + W(f) \quad (3)$$

$$\frac{Y(f)}{H(f)} = \frac{H(f)S(f)}{H(f)} + \frac{W(f)}{H(f)} \quad (4)$$

$$\frac{Y(f)}{H(f)} = S(f) + \frac{W(f)}{H(f)}, \quad (5)$$

where $H(f)$ is the channel effect, $S(f)$ is our desired received information, and $\frac{W(f}{H(f)}$ is a new noise component, which in zero-forcing case is larger than $W(f)$. After this, maximum likelihood (ML) detection is used based on the received statistical information to decide on the received symbols and to recover $s_n^{(m)}$. Afterwards, depending on the received SNR, symbol error rate (SER) and bit error rate (BER) are then calculated.

C. Simulation Task

1) Parameters: In our simulation the given bandwidth of $B = 1MHz$ is divided to $N = 500$ sub-carriers, which means that every m-th OFDM symbol is transmitted over 500 sub-carriers each having $2kHz$ bandwidth. Also, the cyclic prefix $N_{cp} = 5$ is added. We have chosen the value of this parameter to satisfy the following constraint: $N_{cp} \geq L-1$, where L is the number of channel taps and calculated as

$$L = \frac{\tau_{DS}}{T_s} = \frac{5.4 \cdot 10^{-6}}{10^{-6}} = 5.4 \quad (6)$$

rounded up to $L = 6$. The derived Doppler frequency is $f_D = 100Hz$. Therefore the second constraint is also satisfied:

$$(N + N_{cp})f_D T_s = 0.0505 \ll 1 \quad (7)$$

The sampling frequency used is $f_s = 1MHz$.
The relation between the average transmit power P and E is given by equation (13). If $P = 0.1W$, then $E = P_t \cdot T_s = 1 \times 10^{-7} Ws$.

2) AWGN channel: In the first part of simulations we are using an AWGN channel in which the only impairment to the transmitted signal is the linear addition of white noise that is Gaussian distributed. The effects of fading, shadowing and multi-path are not taken into consideration. For simulation we set the parameter $c_0(nT_s) = 1, \forall n$ and $c_{l\neq 0}(nT_s)$.

- Transmitter: Create a sequence of QPSK symbols of length N, apply the IFFT to the sequence to obtain the signal in the time domain (1). Then cyclic prefix of length $N_{cp} = 5$ is added and OFDM symbols are formed.
- Receiver: At the receiver side, we add AWGN samples with variance N_0 given in equation (8). The total noise power is given by equation (9).

$$var = \frac{Noise}{2} \quad (8)$$

$$Noise = N_0/2 \cdot B = 2.07 \times 10^{-14} W \quad (9)$$

Then, the cyclic prefix is removed and the FFT is applied to the remaining N samples for the m-th OFDM symbols to obtain signal in the frequency domain. Also, the FFT with the length N of the impulse response h of the AWGN channel is calculated. At the end, the zero-forcing equalization (5) is performed to remove the channel effect.

The obtained scatter plots of the received signal (2) on a few sub-carriers $n \in \{5, 200, 400, 500\}$ are presented on the Figure 5 in appendix A.

3) Time-varying frequency selective channel:

- Transmitter: Create a QPSK sequence of length N and apply the IFFT to this sequence. The outcome is given by equation (1). Then add again the cyclic prefix with length $N_{cp} = 5$.
- Receiver: At the receiver the delayed symbols and the cyclic prefix should be removed. Furthermore, AWGN samples with variance given in equation (8) have been added.
  For the channel equalization the FFT with length N of the impulse response h of the time-varying frequency selective channel is calculated. Also the FFT with length N of the received symbols is calculated. The zero forcing equalizer is then given by equation (5). For the maximum likelihood estimation (MLE) we use equation (10) and search for the symbols out of the possible constellations with the smallest euclidean distance to the received symbols. In the case of QPSK symbols, maximum likelihood estimation can be implemented just by looking at the real, or imaginary part of the signal, if it is smaller or larger than zero.

$$arg \min_{s_i} = \sum_{j=1}^{N} (r_j - s_{ij})^2 \tag{10}$$

The power P_r of the received symbols due to path loss is given by equation (12) and (13). Equation (11) is for the computation of the received power P_r in dB.

$$P_r^{dB} = P_t^{dB} + P_L^{dB} \tag{11}$$

$$P_r = P_t \cdot P_L \tag{12}$$

$$P_t = \frac{Energy \cdot N}{Time \cdot N} = \frac{E \cdot N}{T_s \cdot N} = \frac{E}{T_s} \tag{13}$$

For the prediction of the SNR by equation (14), the received symbol energy is divided by the noise per sub-carrier. The behavior of the simulated channel compared to the prediction of the SNR is shown in Fig. 2. The peaks in the curves for the simulated SNR for low transmitted power are caused by the channel effect and the channel estimation and its noise increase. If we average over all sub-carrier this effect is a little flattened out.

$$SNR = \frac{E \cdot P_L}{Noise} \tag{14}$$

Fig. 1 shows the measured symbol error rate. The effective date rate can be calculated with equation (15).

$$R = \frac{N}{N + N_{cp}} \cdot f_c \cdot 2bit \tag{15}$$

The scatter plots for the time-varying frequency selective channel are attached to the appendix in Fig. 6. The figure shows obtained scatter plots of the received signal for some sub-carriers.

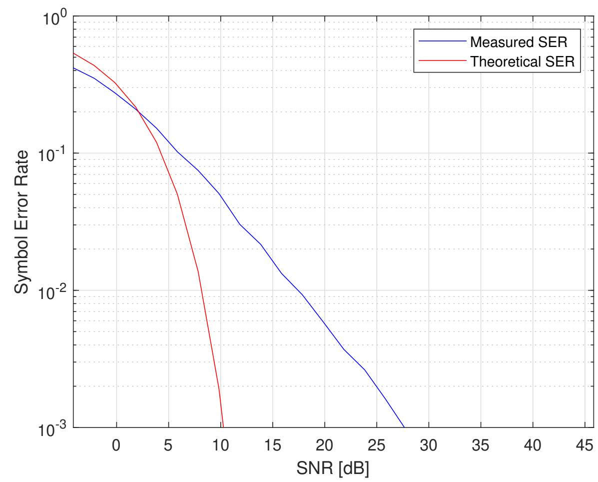

Fig. 1: Comparison of theoretical symbol error rate (SER) to measured SER

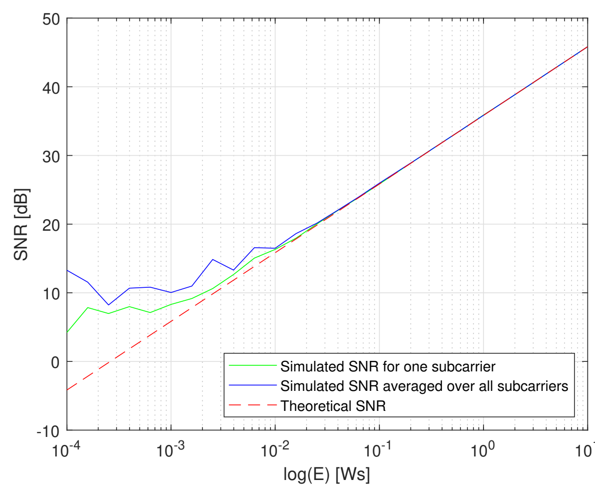

Fig. 2: Comparison of theoretical SNR to measured SNR

4) Ncp: In Fig. 3 different lengths for the cyclic prefix is tested for the same time-varying frequency selective channel with AWGN. When the cyclic prefix is shorter than the delay spread $N_{cp}T_s < \tau_{DS}$, the transmitted symbols experience ISI. Which can be seen in Fig. 3 as the symbols obtain phase shift and amplitude scaling. Fig. 4 shows the SER for $N_{cp} = 2$ which is much worse in comparison to $N_{cp} = 5$ in Fig. 1

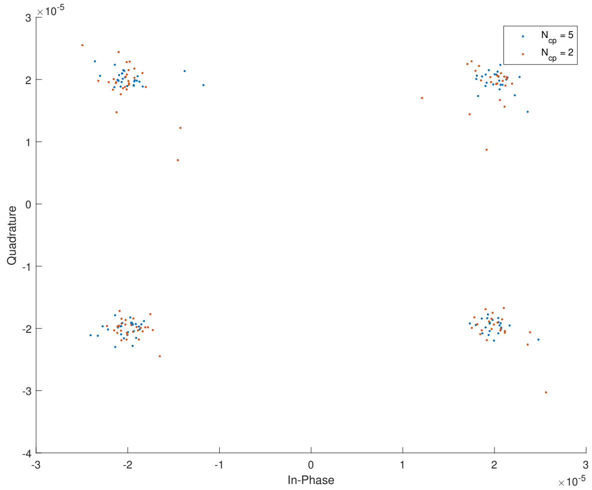

Fig. 3: Scatterplot for different N_{cp}

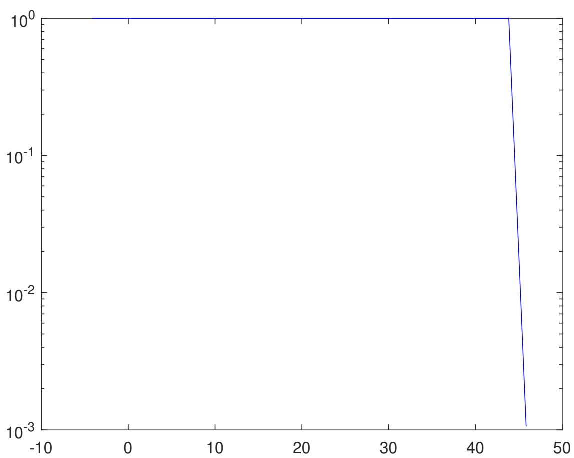

Fig. 4: SER for $N_{cp} = 2$

III. Conclusion

In this project we used the Orthogonal Frequency Division Multiplexing (OFDM) to enable communication over a time-varying frequency selective channel. The simulation was built up on the transformation of a sequential sequence of QPSK symbols in the frequency domain into a parallel sequence in the time domain using the $IFFT$. We learnt the role of the cyclic prefix in OFDM and its impact on the communication when it gets too short. For the channel, the given fading channel was used. The concluding transformation back into the frequency domain was done by the FFT of the received symbols. To get rid of the residual interference caused by the fading channel, we used zero forcing estimation with the drawback of increasing the noise. The simulated data was then compared with theory to create different plots such as SNR and SER.

IV. Contributions

APPENDIX A
SCATTERPLOTS

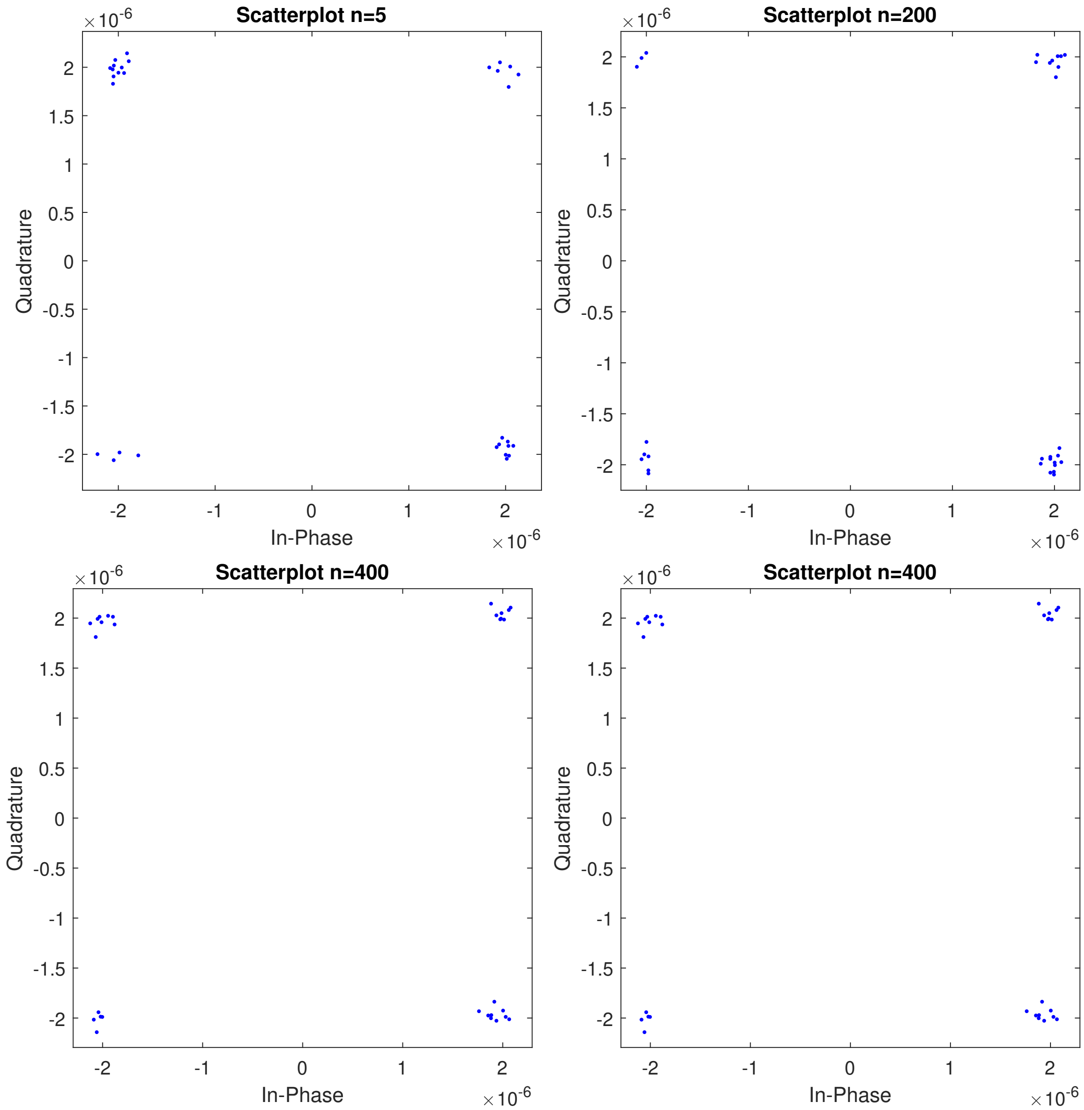

Fig. 5: Scatterplots of received symbols over the AWGN channel for different sub-carriers $n \in \{5, 200, 400, 500\}$

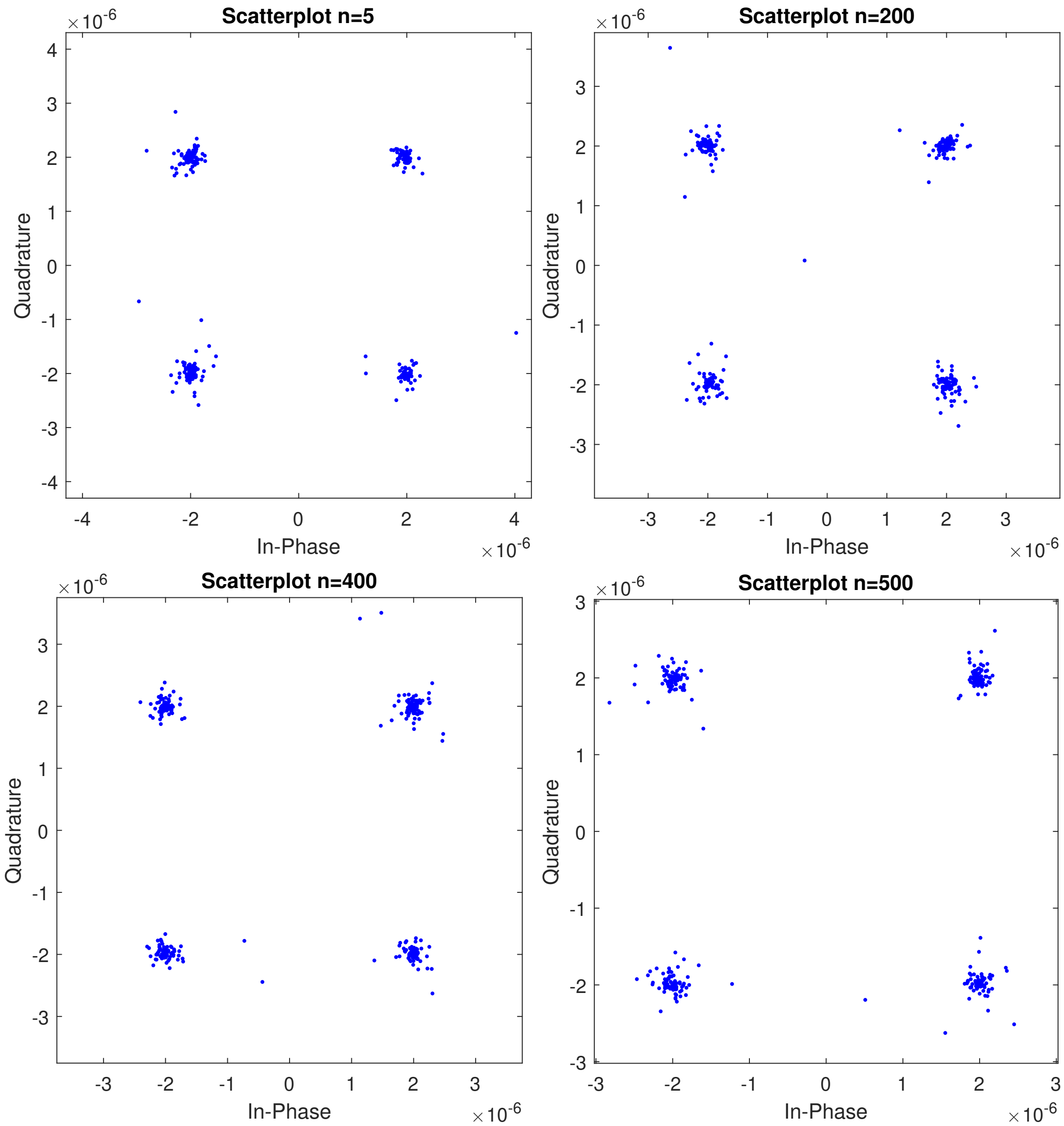

Fig. 6: Scatterplots of received symbols over the time-varying frequency selective channel for different subcarriers $n \in \{5, 200, 400, 500\}$

Appendix B
Matlab code

AWGN channel

```
%% Project 2
clc, clear all, close all
% == Parameters == %
delay_spread = 5.4e-6;
fc = 2e9;
v = 15; % speed of receiver
fD = (v/(physconst('LightSpeed')))*fc;
B = 1e6;
Noise = 2.07e-20*B; % in [W]
fs = 1e6;
Ts = 1/fs; % Sample time
fDTs = fD*Ts; % Normalized Doppler frequency
E = 0.1;
N = 500;
L = ceil(delay_spread/Ts);
Ncp = (L-1);
PL = 10^(-10.1); %path loss (linear)
M = 30; % Number of time samples
if (N + Ncp)*fDTs*10 > 1 % Check constraint
    error('N or Ncp to large')
end

% == Transmitter == %
symbs = zeros(N,M);
y = zeros(N,M);
for m = 1:M
    [symbs(:,m), ~] = qpsk_sequence(N,E);
    symbs_ifft = sqrt(N)*ifft(symbs(:,m),N);
    z = [symbs_ifft(end-Ncp+1:end);symbs_ifft];

    % == Channel == %
    h = 1;
    z = z*sqrt(PL); % path loss
    c_output = z + (randn(length(z),1) +1i*rand(length(z),1))*sqrt(Noise/2);  % add noise

    % == Reciever == %
    r2 = c_output(Ncp+1:end); %remove cp

    ch = fft(h,N); % FFT of channel

    y(:,m) = fft(r2,N)/sqrt(N); % FFT of received symbols
    y(:,m) =y(:,m)./ch.';  % zero forcing equalizer with the first tap of the channel frequency response.
end

scatterplot(y(5,:)), hold on; % scatterplot for the first subcarrier
title('Scatterplot n=5');
print -depsc awgn_n=5;

scatterplot(y(200,:));
title('Scatterplot n=200');
print -depsc awgn_n=200;
```

```
scatterplot(y(400,:));
title('Scatterplot n=400');
print -depsc awgn_n=400;

scatterplot(y(N,:));
title('Scatterplot n=500');
print -depsc awgn_n=500;
```

Time-varying frequency selective channel

```
%% Project 2
clc, clear all, close all
% == Parameters == %
delay_spread = 5.4e-6;
fc = 2e9;
v = 15; % speed of receiver
fD = (v/(physconst('LightSpeed')))*fc;
B = 1e6;
Noise = 2.07e-20*B; % in [W]
fs = 1e6;
Ts = 1/fs; % Sample time
fDTs = fD*Ts; % Normalized Doppler frequency
E = 0.1;
N = 500;
L = ceil(delay_spread/Ts);
Ncp = L-1;
tau = (0:1:L-1); % Path delays in samples
P = (1/length(tau))*ones(1,length(tau)); % Flat power delay profile
M = 300; % Number of time samples

if (N + Ncp)*fDTs*10 > 1 % Check constraint
    error('N or Ncp to large')
end

% == Transmitter == %
symbs = zeros(N,M);
y = zeros(N,M);

for m = 1:M
    [symbs(:,m), ~] = qpsk_sequence(N,E);
    symbs_ifft = sqrt(N)*ifft(symbs(:,m),N);
    z = [symbs_ifft(end-Ncp+1:end);symbs_ifft];

    % == Channel == %
    [z, h] = Fading_Channel(z, tau, fDTs, P);
    z = z(1:end-L+1); % Remove delayed symbols

    %c_output = awgn(z,i,'Measured');
    z = z*sqrt(10^(-10.1)); % path loss
    c_output = z + (randn(length(z),1) +1i*rand(length(z),1))*sqrt(Noise/2);   % add
        noise

    % == Reciever == %
    r2 = c_output(Ncp+1:end); %remove cp

    ch = fft(h(1,:),N); % FFT of channel

    y(:,m) = fft(r2,N)/sqrt(N); % FFT of received symbols
```

```
    y(:,m) =y(:,m)./ch.';  % zero forcing equalizer with the first tap of the channel frequency response.

end

scatterplot(y(5,:)), hold on; % scatterplot for the first subcarrier
title('Scatterplot n=5');
print -depsc scatter_freq_sel_n=5;

scatterplot(y(200,:));
title('Scatterplot n=200');
print -depsc scatter_freq_sel_n=200;

scatterplot(y(400,:));
title('Scatterplot n=400');
print -depsc scatter_freq_sel_n=400;

scatterplot(y(N,:));
title('Scatterplot n=500');
print -depsc scatter_freq_sel_n=500;
```

Time-varying frequency selective channel with varying E

```
%% Project 2
clc, clear all, close all
% == Parameters == %
delay_spread = 5.4e-6;
fc = 2e9;
v = 15; % speed of receiver
fD = (v/(physconst('LightSpeed')))*fc;
B = 1e6;
Noise = 2.07e-20*B; % in [W]
fs = 1e6;
Ts = 1/fs; % Sample time
fDTs = fD*Ts; % Normalized Doppler frequency
P_transmitted  = 20:2:70; %Transmitted power in dB.
E = 10.^(P_transmitted/10)*Ts; % Transmitted Energy converted to linear.
%E = linspace(1e-7,1e-2,100);%/Ts;
N = 500;
L = ceil(delay_spread/Ts);
Ncp = L-1;
tau = (0:1:L-1); % Path delays in samples
P = (1/length(tau))*ones(1,length(tau)); % Flat power delay profile
M = 600; % Number of OFDM symbols
PL = 10^(-10.1); %Path loss (linear)

if (N + Ncp)*fDTs*10 > 1 % Check constraint
    error('N or Ncp to large')
end
%% == Simulation Parameters == %%
SNR_N = nan(N,length(E));
SER = nan(length(E),1);
% == Simulation  == %
for i=1:length(E)
    % == Transmitter  == %
    symbs = zeros(N,M);
    y = zeros(N,M);

    for m = 1:M
```

```
            [symbs(:,m), ~] = qpsk_sequence(N,E(i));
            symbs_ifft = sqrt(N)*ifft(symbs(:,m),N);
            z = [symbs_ifft(end-Ncp+1:end);symbs_ifft]; %Add cyclic prefix

            % == Channel  == %
            [z, h] = Fading_Channel(z, tau, fDTs, P);
            z = z(1:end-L+1); % Remove delayed symbols

            z = z*sqrt(PL); % apply path loss
            c_output = z + (randn(length(z),1) +1i*rand(length(z),1))*sqrt(Noise/2);   % add noise

            % == Reciever  == %
            r2 = c_output(Ncp+1:end); %remove cp

            ch = fft(h(1,:),N); % FFT of channel

            y(:,m) = fft(r2,N)/sqrt(N); % FFT of received symbols
            y(:,m) =y(:,m)./ch.';  % zero forcing equalizer with the first tap of the channel frequency response.
        end
        const = [1+1j,1-1j,-1+1j,-1-1j]*sqrt(0.5*E(i));
        received_ofdm_syms = reshape(y,[N*M 1]);
        metric = abs(repmat(received_ofdm_syms,1,4) - repmat(const, length(received_ofdm_syms), 1)).^2; % compute the distance to each possible symbol
        [~, indx_r] = min(metric, [], 2); % find the closest for each received symbol
        msg_r = nan(length(indx_r),1); % All ofdm symbs in one vector
        for j=1:length(indx_r)
            msg_r(j) = const(indx_r(j));
        end
        msg_r = reshape(msg_r,[N M]);
        %  == Calculate symbol error rate == %
        [number,ratio] = symerr(round(msg_r,4),round(symbs,4));
        SER(i) = ratio;
        % == Power of signal == %
        SNR_N(:,i) = mean(abs(y).^2,2)/Noise-1; %averaging over all subcarriers, for large number of OFDM symbols is it the same than for single sucarriers.
end
SNR_avg = mean(SNR_N,1);
Snr = @(x)((x)*PL/Noise);     %x or x^2 depends on implementation of QPSK_sequence:
                                %x for sqrt(E)*symbol; x^2 for E*symbol;

SER_Theory = @(x) 2*qfunc(sqrt(x));
SNR = E*PL/Noise;
figure
semilogy(10*log10(SNR),SER,'b-'), hold on
% semilogy(10*log10(SNR),SER2,'g*'), hold on
semilogy(10*log10(SNR),SER_Theory(SNR),'r-')
xlabel('SNR [dB]'), ylabel('Symbol Error Rate')
xlim([min(10*log10(SNR)) max(10*log10(SNR))]);
ylim([1e-3 1]);
grid on;
legend('Measured SER','Theoretical SER','location','northeast')
print -depsc SymbolErrorRate;
figure
semilogx(E,10*log10(SNR_N(2,:)),'g-'), hold on
semilogx(E,10*log10(SNR_avg),'b-')
semilogx(E,10*log10(Snr(E)),'r--');
```

```
%xlim([0 0.5]);
grid on
xlabel('log(E) [Ws]'), ylabel('SNR [dB]')
legend('Simulated SNR for one subcarrier','Simulated SNR averaged over all
    subcarriers','Theoretical SNR','location','southeast')
print -depsc SNR;
```

Simulations for short N_{cp}

```
%% Project 2
clc,
% == Parameters == %
delay_spread = 5.4e-6;
fc = 2e9;
v = 15; % speed of receiver
fD = (v/(physconst('LightSpeed')))*fc;
B = 1e6;
Noise = 2.07e-20*B; % in [W]
fs = 1e6;
Ts = 1/fs; % Sample time
fDTs = fD*Ts; % Normalized Doppler frequency
P_transmitted  = 20:2:70; %Transmitted power in dB.
E = 10.^(P_transmitted/10)*Ts; % Transmitted Energy converted to linear.
N = 500;
L = ceil(delay_spread/Ts);
Ncp = L-1-3;
tau = (0:1:L-1); % Path delays in samples
P = (1/length(tau))*ones(1,length(tau)); % Flat power delay profile
M = 100; % Number of time samples
PL = 10^(-10.1);
if (N + Ncp)*fDTs*10 > 1 % Check constraint
    error('N or Ncp to large')
end
To = (N+Ncp)*Ts;
n = 1:N+Ncp;
%z = (m-1)*To;
%% == Simulation Parameters == %%
pow = nan(length(E),1);
SER = nan(length(E),1);
% == Simulation  == %
for i=1:length(E)
    % == Transmitter  == %
    %symbs = zeros(N,M);
    bits_s = zeros(2*N,M);
    y = zeros(N,M);

    for m = 1:M
        %[symbs(:,m), bits_s(:,m)] = qpsk_sequence(N,E(i));
        symbs_ifft = sqrt(N)*ifft(symbs(:,m),N);
        z = [symbs_ifft(end-Ncp+1:end);symbs_ifft];

        % == Channel  == %
        [z, h] = Fading_Channel(z, tau, fDTs, P);
        z = z(1:end-L+1); % Remove delayed symbols

        z = z*sqrt(10^(-10.1)); % Pathloss
        c_output = z + (randn(length(z),1) +1i*rand(length(z),1))*sqrt(Noise/2);   %
            add noise

```

```
        % == Reciever  == %
        r2 = c_output(Ncp+1:end); %remove cp

        ch = fft(h(1,:),N); % FFT of channel

        y(:,m) = fft(r2,N)/sqrt(N); % FFT of received symbols
        y(:,m) =y(:,m)./ch.';  % zero forcing equalizer with the first tap of the channel frequency response.
    end
    const = [1+1j,1-1j,-1+1j,-1-1j]*sqrt(0.5*E(i));
    received_ofdm_syms = reshape(y,[N*M 1]);
    metric = abs(repmat(received_ofdm_syms,1,4) - repmat(const, length(received_ofdm_syms), 1)).^2; % compute the distance to each possible symbol
    [~, indx_r] = min(metric, [], 2); % find the closest for each received symbol
    msg_r = nan(length(indx_r),1); % All ofdm symbs in one vector
    for j=1:length(indx_r)
        msg_r(j) = const(indx_r(j));
    end
    msg_r = reshape(msg_r,[N M]);
    %  == Calculate symbol error rate == %
    [number,ratio] = symerr(round(msg_r,4),round(symbs,4));
    SER(i) = ratio;
    % == Power of signal == %
    pow(i) = mean(mean(abs(y).^2))*N/Noise; %averaging over all subcarriers, for large number of OFDM symbols is it the same than for single sucarriers.
    %pow(i) = qfuncinv(ratio/2)^2;
end

Snr = @(x)((x)*(10^(-10.1)))*N/Noise;   %x or x^2 depends on implementation of QPSK_sequence:
                                        %x for sqrt(E)*symbol; x^2 for E*symbol
SER_Theory = @(x) 2*qfunc(sqrt(x));
SNR = E*PL/Noise;
figure
semilogy(10*log10(SNR),SER,'b-'), hold on
figure
plot(E,10*log10(pow),'b-'), hold on
plot(E,10*log10(Snr(E)),'r--');
legend('Simulated SNR','Theory SNR','location','northwest')
figure(11)
scatter(real(y(1,:)),imag(y(1,:)),'.'), hold on % scatterplot for the first subcarrie
xlabel('In-Phase'), ylabel('Quadrature'), legend('N_{cp} = 5','N_{cp} = 2')
```

Function to generate QPSK symbols

```
function [symbols, data]= qpsk_sequence(N,E)
%N: lenght of the QPSK sequence
%E: energie per symbol
data = randi([0 1],2*N,1,'double');        %random bitstream of lengt 2*N
symbols = (sqrt(E))*nrSymbolModulate(data,'QPSK','OutputDataType','double');        % QPSK sequence of length N
end
```