Project 2: OFDM Technology

FAN Qingyuan	Hn Zhuchen	XU Heng	YANG Chenhao
11812418	11813128	11813118	11812413

Dec 2019

Abstract

In order to implement OFDM, we split the input signal into pieces and then transmit them simultaneously by multiplying different sin or cos functions at different frequencies (which are orthogonal to each other) to the receiver. With the help of FFT and IFFT, we can finish the above tasks more easily. In this project, we use 4 blocks to implement OFDM transceiver and use it to transmit a random signal.

Introduction

Block 1 & 2: In block 1 OFDM transmitter, we use IFFT to substitute the way of assigning sub-signals to carriers with different frequencies. In block 2 OFDM receiver, we use FFT to substitute the way of integrating the received signal for the preparation of demodulation. In other words, block 1 converts the input signal from frequency domain to time domain, then after transmitting, block 2 converts the receiving signal from time domain to frequency domain. Then, we add CP to convert convolution to periodic convolution and avoid inter-symbol interference.

Block 3: It includes DAC and transmitter radio frequency front end. DAC transform every impulse in the sample signal into rectangular waves with T extension (T is the sampling period). Next in order to separate the real part and the imaginary part of signal, we multiply the real part by cos(ct) and multiply the imaginary part by sin(ct), and then sum them up for the preparation of transmitting.

Receiver Design and Analysis

Task A

It includes DAC and transmitter radio frequency front end. DAC transform every impulse in the sample signal into rectangular waves with T extension (T is the sampling period). Next in order to separate the real part and the imaginary part of signal, we multiply the real part by $\cos(w_c\ t)$ and multiply the imaginary part by $\sin(w_c\ t)$, and then sum them up for the preparation of transmitting.

Task B

The impulse response of $h(t)$ is shown in the figure below.

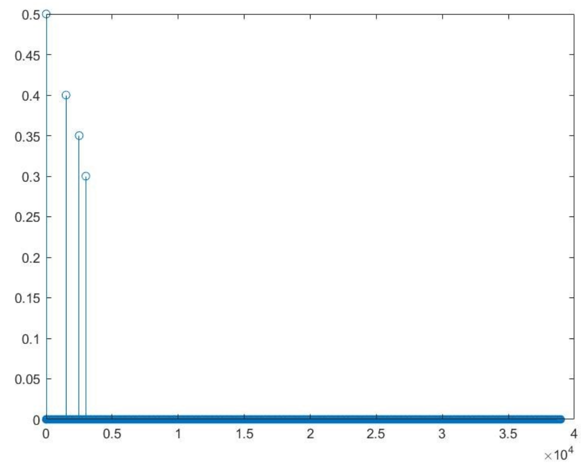

Figure 1: Impulse Response of $h[t]$

And the impulse response of $h[n]$ is shown in the figure below.

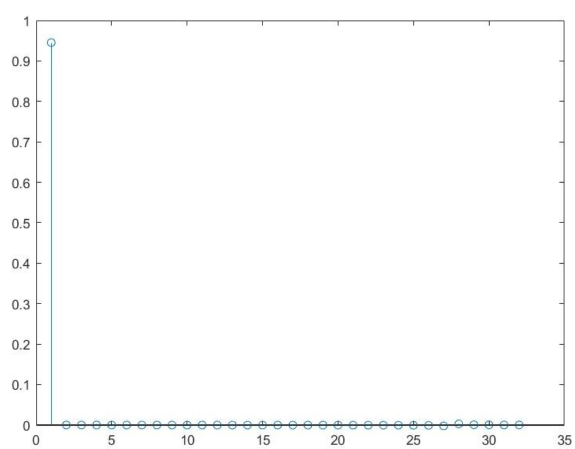

Figure 2: Impulse Response of $h[n]$

Form the process, we can know the relationship between the input signal s and output signal s' is that:

$$f\left[\left[Im\left\{f^{-1}\left\{s\right\}\right\}cos(w_ct)+Im\left\{f^{-1}\left\{s\right\}\right\}sin(w_ct)\right]*h(t)\cdot(isin(w_ct)+cos(w_ct))\right]=s'$$

And the relationship between $h[n]$ and $h(t)$ is shown above. From that equation we can know that the output signal is almost the same as the input signal.

Task C: Plot How the Signal Changes in the Block 4 Step-By-Step

Flow Chart

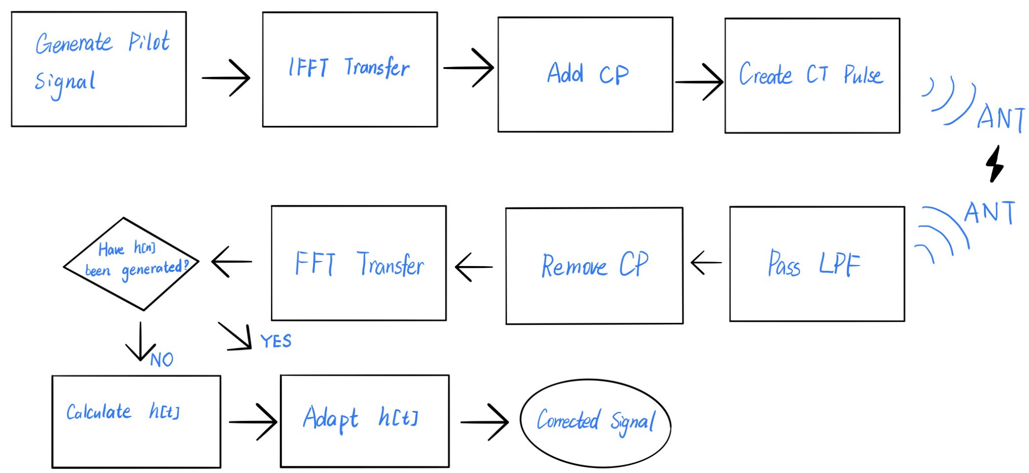

Figure 3: Flow Chart

Step 1

Generate pilot signal randomly.

Code

```
%p1_Initialize.m
%% initialize variable
N = 31; %length of signal
CP_N = 4; %length of CP
dt=1e-9;
T=1e-6;
omega_C=1e8; %Wc

%% generate signal serial (pilot signal)
ss = zeros(1,N);
for n=1:N
rndr = sign(randn(1));
rndi = sign(randn(1));
%rndr = randi([-20000,20000],1,1);
%rndi = randi([-20000,20000],1,1);
%rndr = sin(n/0.5);
%rndi = cos(n/0.5);
ss(n) = rndr + 1i*rndi;
end
figure1 = figure();
```

```
subplot(2,1,1);
stem(real(ss));
grid on;title('Real(x_{pilot})');xlabel('n');ylabel('Magnitude');
subplot(2,1,2);
stem(imag(ss));
grid on;title('Image(x_{pilot})');xlabel('n');ylabel('Magnitude');
saveas(figure1,'../fig/1_pilot_signal.png');
```

Figure

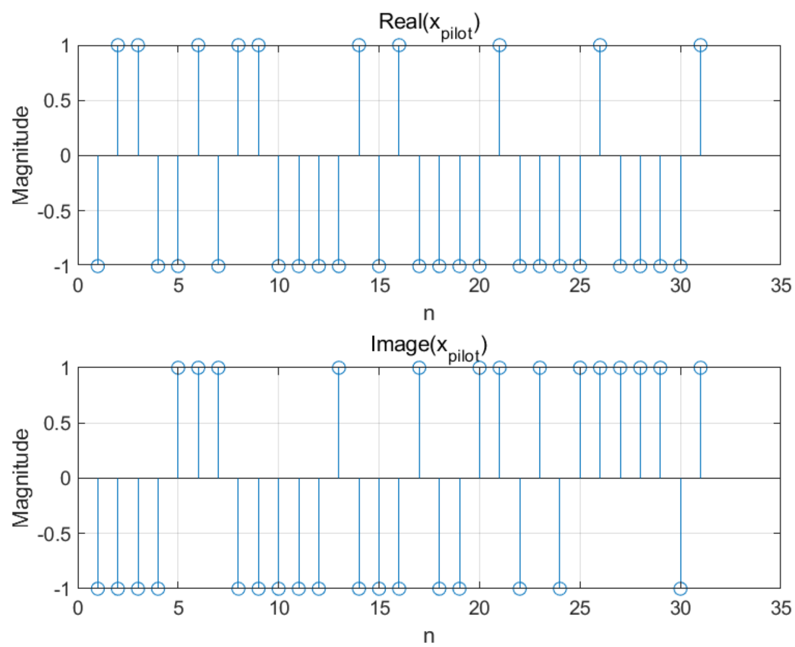

Figure 4: Pilot Signal

Step 2

Using MATLAB function ifft() to transfer it into OFDM symbols.

Code

```
%p2_Block1.m
%% IFFT

IFFT_ss = ifft(ss,N+1);

figure1 = figure();
subplot(2,1,1);
stem(real(IFFT_ss));
grid on;title('Real(Signal After IFFT)');xlabel('n');ylabel('Magnitude');
```

```
    subplot(2,1,2);
    stem(imag(IFFT_ss));
    grid on;title('Real(Signal After IFFT)');xlabel('n');ylabel('Magnitude');
    saveas(figure1,'../fig/2_Signal_After_IFFT.png');

    %% add CP

    ss_Add_CP = [IFFT_ss(N+2-CP_N:N+1) IFFT_ss];

    figure1 = figure();
    subplot(2,1,1);
    stem(real(ss_Add_CP));
    grid on;title('Real(Signal Add CP)');xlabel('n');ylabel('Magnitude');
    subplot(2,1,2);
    stem(imag(ss_Add_CP));
    grid on;title('Real(Signal Add CP)');xlabel('n');ylabel('Magnitude');
    saveas(figure1,'../fig/3_Signal_Add_CP.png');

    %% create CT pulse

    OFDM_Pulse_CT1=[];
    t=0:dt:(T*(N+1+CP_N)-dt);
    for n=1:N+1+CP_N
        OFDM_Pulse_CT1 = [OFDM_Pulse_CT1 ss_Add_CP(n) zeros(1,round(T/dt)-1)];
    end

    % Plot shape
    LTIC = [0 ones(1,round(T/dt))];
    OFDM_Pulse_CT_Transmission = conv(OFDM_Pulse_CT1,LTIC);

    figure1 = figure();
    subplot(2,2,1);
    plot(t,real(OFDM_Pulse_CT1));
    grid on;title('Real(Signal Add CP Pulse)');ylabel('Magnitude');
    subplot(2,2,2);
    plot(t,imag(OFDM_Pulse_CT1));
    grid on;title('Image(Signal Add CP Pulse)');ylabel('Magnitude');
    subplot(2,2,3);
    plot(t,real(OFDM_Pulse_CT_Transmission(1:length(t))));
    grid on;title('Real(Signal Add CP Pulse Convolution)');ylabel('Magnitude');
    subplot(2,2,4);
    plot(t,imag(OFDM_Pulse_CT_Transmission(1:length(t))));
    grid on;title('Image(Signal Add CP Pulse Convolution)');ylabel('Magnitude');
    saveas(figure1,'../fig/4_Signal_Add_CP_Pulse.png');
```

Figure

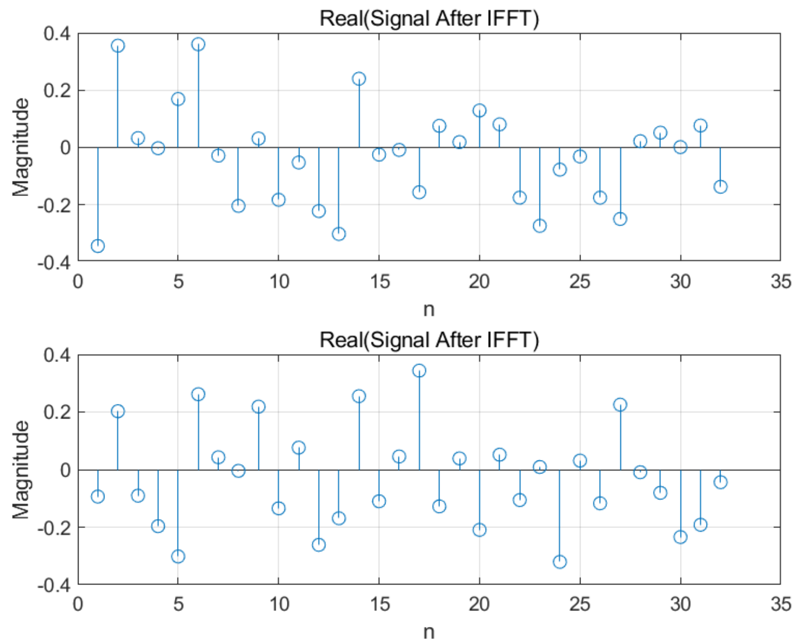

Figure 5: Signal Passed the IFFT Transform

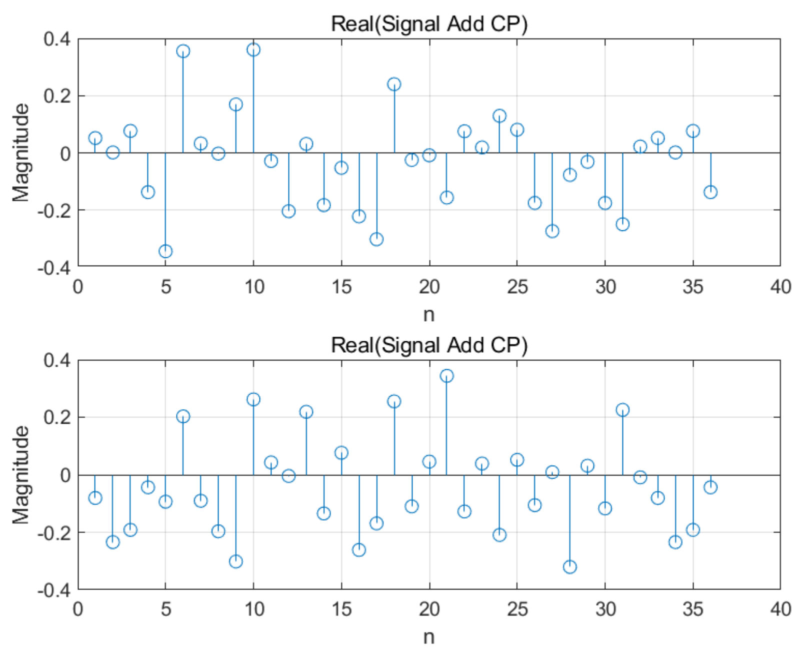

Figure 6: Signal After Added CP

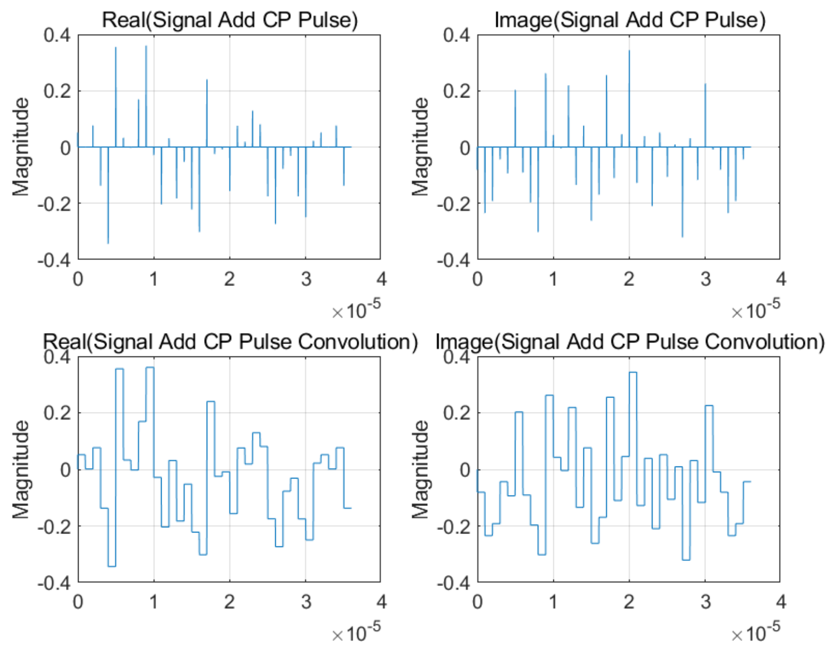

Figure 7: Signal After Added CP's Pulse Shape

Step 3

Add cp signal ahead of the sequence of symbols.

Code

```
%p3_Block3and4.m
%% Construct RF
%multiply sin & cos for real and image part
OFDM_Pulse_TR_TX=real(OFDM_Pulse_CT_Transmission(1:length(t))).*cos(omega_C*t)+imag...
(OFDM_Pulse_CT_Transmission(1:length(t))).*sin(omega_C*t);

figure1 = figure();
subplot(2,1,1);
plot(t,OFDM_Pulse_TR_TX);
grid on;title('Signal');ylabel('Magnitude');
subplot(2,1,2);
f=(-(length(OFDM_Pulse_TR_TX)-1)/2:(length(OFDM_Pulse_TR_TX)-1)/2)*(2*pi/dt/length...
(OFDM_Pulse_TR_TX));
stem(f,abs(fftshift(fft(OFDM_Pulse_TR_TX))/length(OFDM_Pulse_TR_TX)));
grid on;title('Signal Spectrum');xlim([0.8e8 1.2e8]);xlabel('f/Hz');ylabel('Magnitude');
saveas(figure1,'../fig/5_Signal_and_its_Spectrum.png');

%% conv h[t] (pass channel)
%generate h[t]
Ht=[0.5 zeros(1,round(T/dt)*1.5-1) 0.4 zeros(1,round(T/dt)-1) 0.35 zeros(1,round(T/dt)...
*0.5-1) 0.3];
OFDM_Pulse_RF=conv(Ht,OFDM_Pulse_TR_TX); %conv RF with h[t]

```

```
    figure1 = figure();
    subplot(2,1,1);
    plot(dt*(1:length(OFDM_Pulse_RF)),OFDM_Pulse_RF);
    grid on;title('Signal (Convolution h[t])');ylabel('Magnitude');
    subplot(2,1,2);
    f=(-(length(OFDM_Pulse_RF)-1)/2:(length(OFDM_Pulse_RF)-1)/2)*(2*pi/dt/length...
    (OFDM_Pulse_RF));
    stem(f,abs(fftshift(fft(OFDM_Pulse_RF))));
    grid on;title('Signal Spectrum (Convolution h[t])');xlim([0.8e8 1.2e8]);xlabel('f/Hz');...
    ylabel('Magnitude');
    saveas(figure1,'../fig/6_Signal_and_its_Spectrum_(conv).png');
    %% Pass the LPF
    T_channel_LPF = dt:dt:(N+CP_N+4)*T; %construct the time axis
    [b,a]=butter(4,0.02); %generate the LPF filter

    %split real & image part using cos & sin function
    OFDM_Pulse_RF_Real = OFDM_Pulse_RF.*cos(omega_C*T_channel_LPF);
    OFDM_Pulse_RF_Image = OFDM_Pulse_RF.*sin(omega_C*T_channel_LPF);
    %pass the LPF and merge
    OFDM_Pulse_RF_LPF_Merge = 2*filter(b,a,OFDM_Pulse_RF_Real) + 2i*filter(b,a,...
    OFDM_Pulse_RF_Image);


    figure1 = figure();
    subplot(2,1,1);
    plot(dt:dt:(N+CP_N+4)*T,real(OFDM_Pulse_RF_LPF_Merge),dt:dt:(N+CP_N+2)*T,real...
    (OFDM_Pulse_CT_Transmission)); %compare real part
    grid on;title('Signal Shape Compare (Real Part)');ylabel('Magnitude');legend...
    ('OFDM Pulse RF LPF Merge','OFDM Pulse CT Transmission');
    subplot(2,1,2);
    plot(dt:dt:(N+CP_N+4)*T,imag(OFDM_Pulse_RF_LPF_Merge),dt:dt:(N+CP_N+2)*T,imag...
    (OFDM_Pulse_CT_Transmission)); %compare iamge part
    grid on;title('Signal Shape Compare (Image Part)');ylabel('Magnitude');legend...
    ('OFDM Pulse RF LPF Merge','OFDM Pulse CT Transmission');
    saveas(figure1,'../fig/7_Signal_Shape_Compare.png');

    %% sample

    Sample_Length = length(ss_Add_CP)+CP_N-1;
    T_sample = round(length(OFDM_Pulse_RF_LPF_Merge)/Sample_Length);
    OFDM_Pulse_RF_Int_Sample_N = 1:1:Sample_Length;
    for n = 1:Sample_Length
        OFDM_Pulse_RF_Int_Sample_N(n) = OFDM_Pulse_RF_LPF_Merge(T_sample*n);
    end

    %OFDM_Pulse_Int_R=cumtrapz(real(OFDM_Pulse_RF_Int_Sample_N));
    %OFDM_Pulse_Int_I=cumtrapz(imag(OFDM_Pulse_RF_Int_Sample_N));
    %OFDM_Pulse_Int = OFDM_Pulse_Int_R + 1i*OFDM_Pulse_Int_I;

    figure1 = figure();
    subplot(2,2,1);
    stem(real(ss_Add_CP));
    grid on;title('Real(Signal add CP)');xlabel('n');ylabel('Magnitude');
    subplot(2,2,2);
    stem(imag(ss_Add_CP));
    grid on;title('Image(Signal add CP)');xlabel('n');ylabel('Magnitude');
```

```
    subplot(2,2,3);
    stem(real(OFDM_Pulse_RF_Int_Sample_N));
    xlabel('n');
    grid on;title('Real(Signal Sampled)');xlabel('n');ylabel('Magnitude');
    subplot(2,2,4);
    stem(imag(OFDM_Pulse_RF_Int_Sample_N));
    grid on;title('Image(Signal Sampled)');xlabel('n');ylabel('Magnitude');
    saveas(figure1,'../fig/8_Signal_After_Integration_Compare.png');
```

Figure

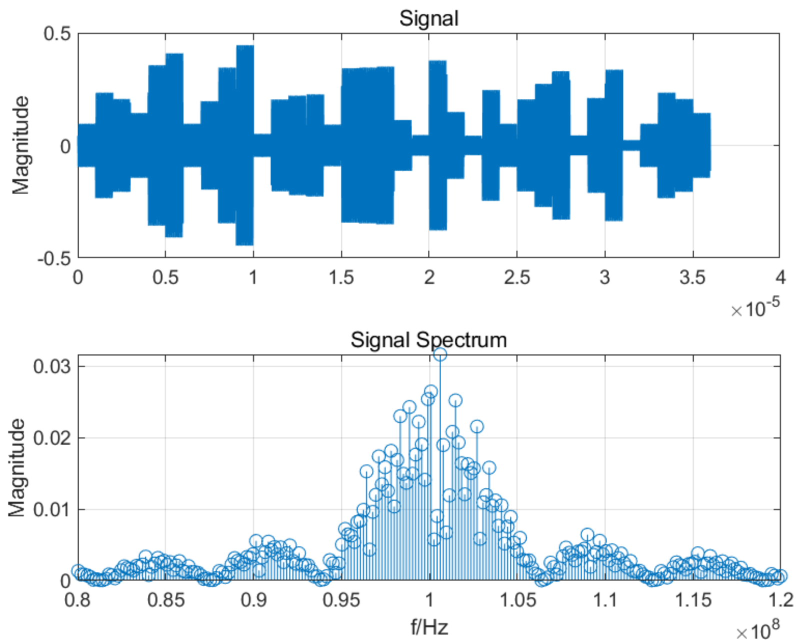

Figure 8: Signal and Its Spectrum

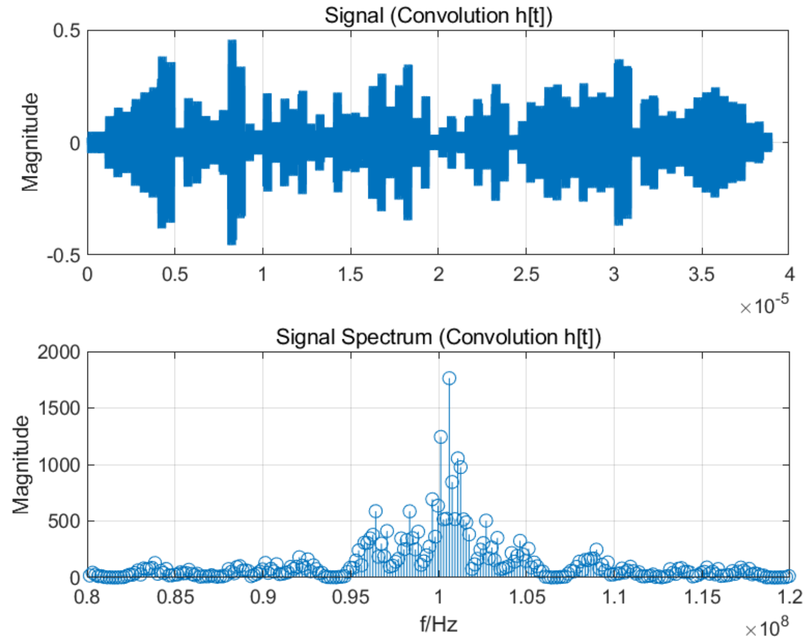

Figure 9: Signal and Its Spectrum After Convolution

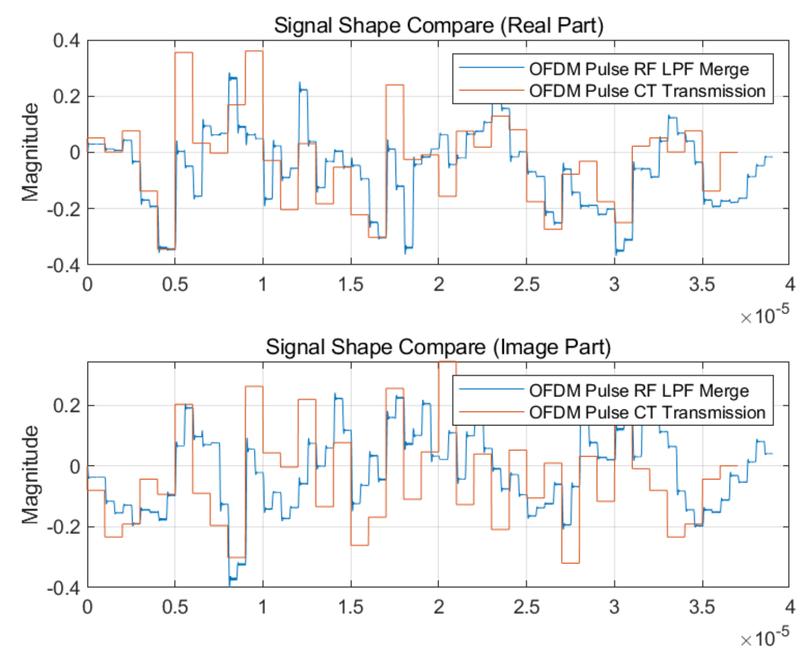

Figure 10: Comparation of the shape of the signal

Step 4

Create and plot CT pulse and the pulse shaping.

Code

```
%p4_Block2.m
%% remove CP
OFDM_Pulse_Int_Remove_CP=OFDM_Pulse_RF_Int_Sample_N((CP_N+1):(N+1+CP_N));

figure1 = figure();
subplot(2,2,1);
stem(real(IFFT_ss));
grid on;title('Real(IFFT of Origin Signal)');xlabel('n');ylabel('Magnitude');
subplot(2,2,2);
stem(imag(IFFT_ss));
grid on;title('Real(IFFT of Origin Signal)');xlabel('n');ylabel('Magnitude');
subplot(2,2,3);
stem(real(OFDM_Pulse_Int_Remove_CP));
grid on;title('Real(RF removed CP)');xlabel('n');ylabel('Magnitude');
subplot(2,2,4);
stem(imag(OFDM_Pulse_Int_Remove_CP));
grid on;title('Real(RF removed CP)');xlabel('n');ylabel('Magnitude');
saveas(figure1,'../fig/9_RF_Signal_Removed_CP.png');


%% FFT
OFDM_RF_fft = fft(OFDM_Pulse_Int_Remove_CP);


figure1 = figure();
subplot(2,2,1);
stem(real(ss));
grid on;title('Real(Origin Signal)');xlabel('n');ylabel('Magnitude');
subplot(2,2,2);
stem(imag(ss));
grid on;title('Image(Origin Signal)');xlabel('n');ylabel('Magnitude');
subplot(2,2,3);
stem(real(OFDM_RF_fft));
grid on;title('Real(RF After FFT)');xlabel('n');ylabel('Magnitude');
subplot(2,2,4);
stem(imag(OFDM_RF_fft));
grid on;title('Image(RF After FFT)');xlabel('n');ylabel('Magnitude');
saveas(figure1,'../fig/10_RF_Signal_FFT.png');
```

Figure

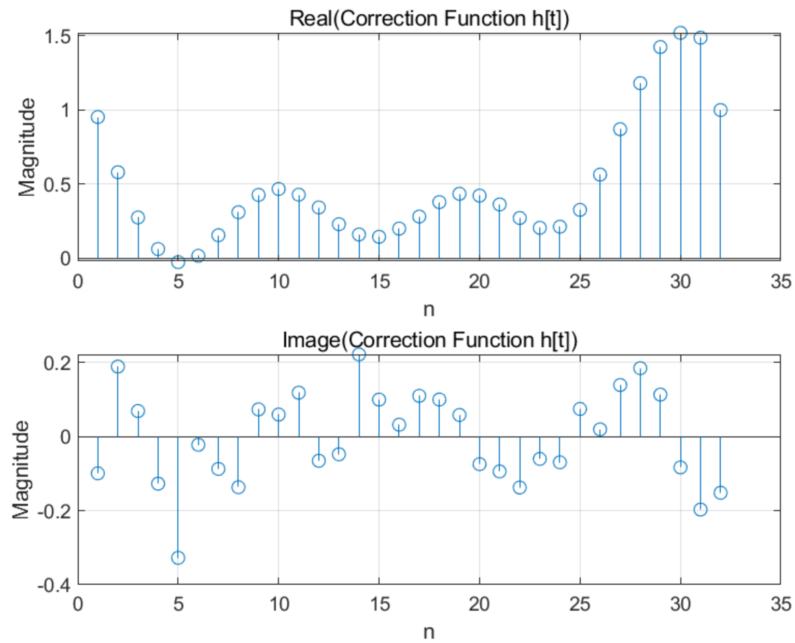

Figure 11: Received Signal Removed CP

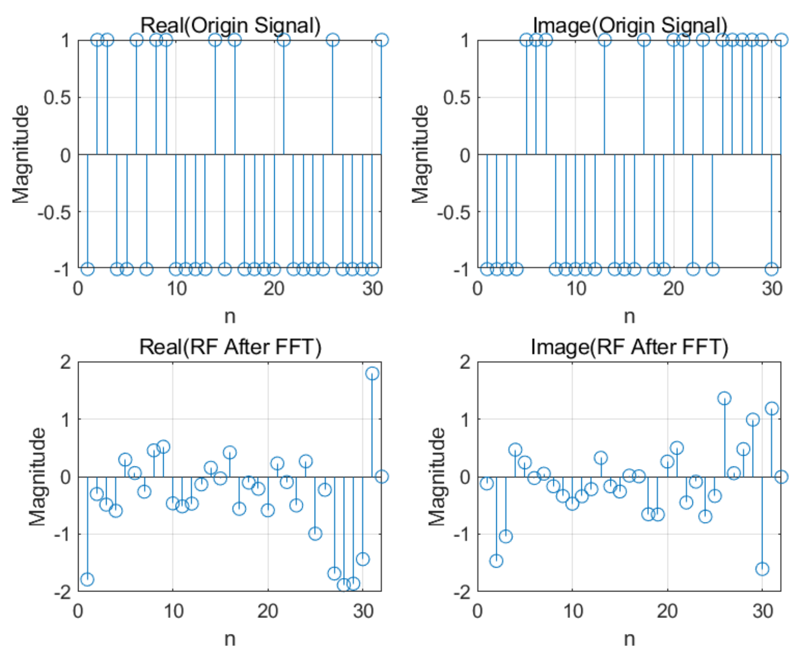

Figure 12: Received Signal After FFT

Step 5

Calculate the Signal Correction Function $h[n]$

```
%get_hn.m
function hn = get_hn(OFDM_RF_fft,Serial_Signal,N)
hn = OFDM_RF_fft(1:N)./Serial_Signal;
    hn = [hn 1];
```

As the figures above shows, there're still slightly deferences between the original pilot signal and the RF signal after FFT, so a function is needed to adapt the correction to the RF signal to accurately recover the original signal. The function `get_hn.m` could find the relation between the RF signal after FFT and the original signal.

```
%% Cal ht
% RUN THIS PROGRAM ONLY ONCE!

hn = get_hn(OFDM_RF_fft,ss,N);
figure1 = figure();
subplot(2,1,1);
stem(real(hn));
grid on;title('Real(Correction Function h[t])');xlabel('n');ylabel('Magnitude');
subplot(2,1,2);
stem(imag(OFDM_Pulse_Int_Remove_CP));
grid on;title('Image(Correction Function h[t])');xlabel('n');ylabel('Magnitude');
saveas(figure1,'../fig/9_RF_Signal_Removed_CP.png');
saveas(figure1,'../fig/11_ht.png');
```

Then, we can use `get_hn.m` to get and plot the correction function $h[t]$. We only need to run this code ONCE.

Figure

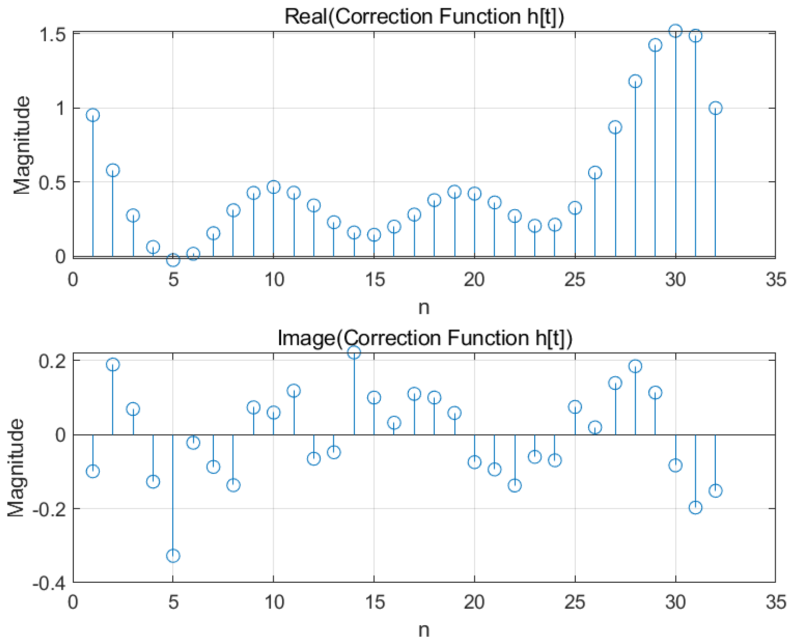

Figure 13: Correction Function $h[t]$

The shape of $h[t]$ dows not have any relation to number of N.

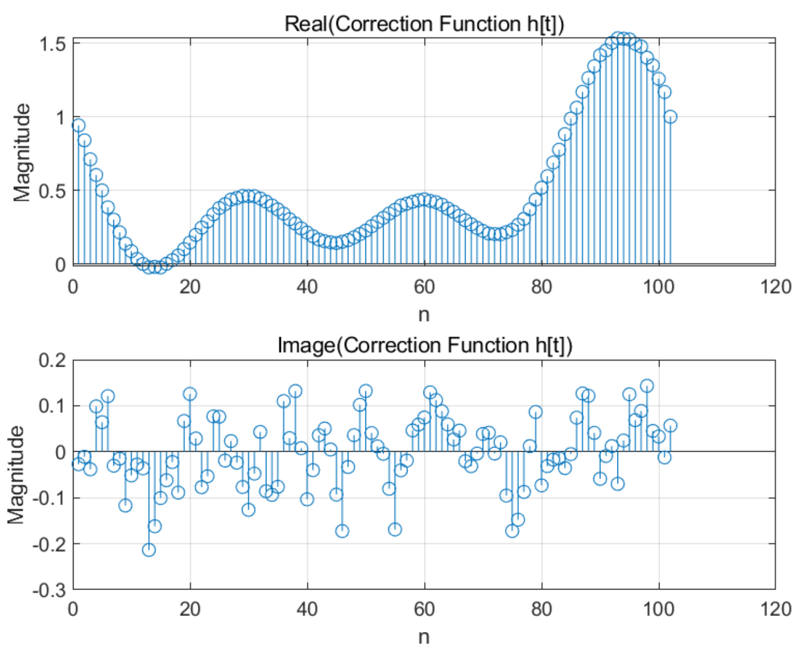

Figure 14: Correction Function $h[t]$ while N=101

Step 6

Adapt the previous correction function, recover the signal and find the difference between them.

Code

```
%p6_Signal_Correction.m
%% Signal Correction
OFDM_RF_fft = OFDM_RF_fft./hn;
OFDM_RF_fft_Scale_Length = OFDM_RF_fft(1:N);

%% Plot Signal After Correction
figure1 = figure();
subplot(2,2,1);
stem(real(ss));
grid on;title('Real(Origin Signal)');xlabel('n');ylabel('Magnitude');
subplot(2,2,2);
stem(imag(ss));
grid on;title('Image(Origin Signal)');xlabel('n');ylabel('Magnitude');
subplot(2,2,3);
stem(real(OFDM_RF_fft_Scale_Length));
grid on;title('Real(RF Signal After Correction)');xlabel('n');ylabel('Magnitude');
subplot(2,2,4);
stem(imag(OFDM_RF_fft_Scale_Length));
grid on;title('Image(RF Signal After Correction)');xlabel('n');ylabel('Magnitude');
saveas(figure1,'../fig/12_RF_Signal_After_Correction.png');

%% Calculate Difference
figure1 = figure();
subplot(2,1,1);
stem(real(ss)-real(OFDM_RF_fft_Scale_Length));
grid on;title('Real(Difference)');xlabel('n');ylabel('Magnitude');
subplot(2,1,2);
stem(imag(ss)-imag(OFDM_RF_fft_Scale_Length));
grid on;title('Image(Difference)');xlabel('n');ylabel('Magnitude');
saveas(figure1,'../fig/13_Signal_Difference.png');
```

Figure

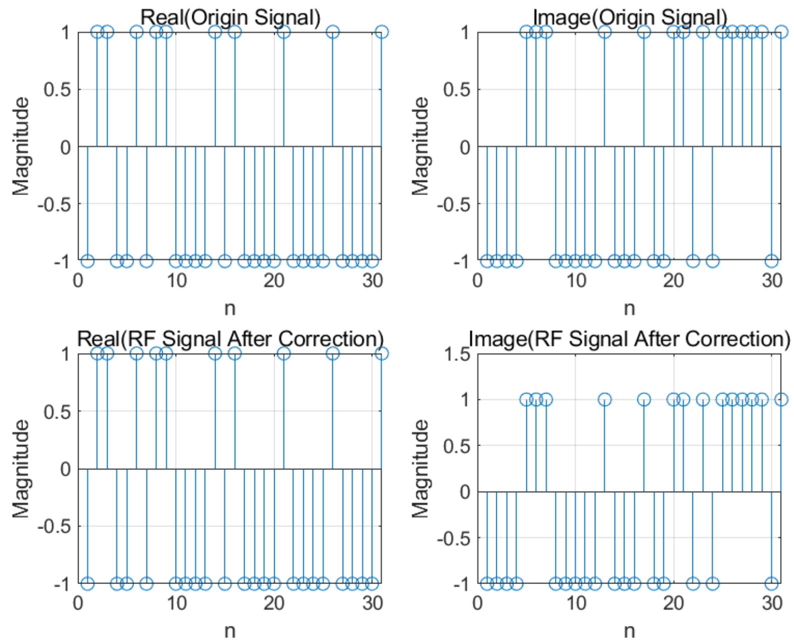

Figure 15: Recovered Signal

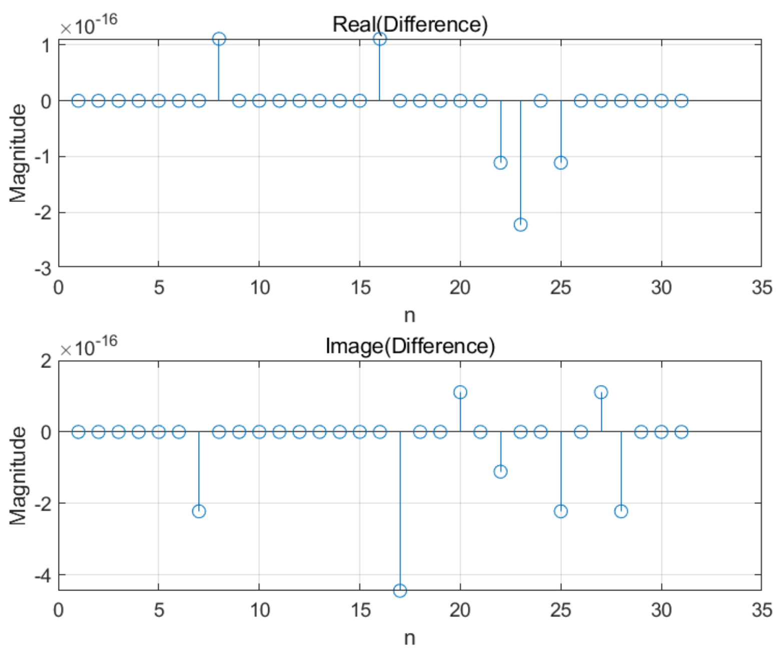

Figure 16: Difference Between Original Signal and Recovered Signal

It can be seen that the recovered signal is close enough to the original signal which only has the difference about 10^{-16} on magnitude.

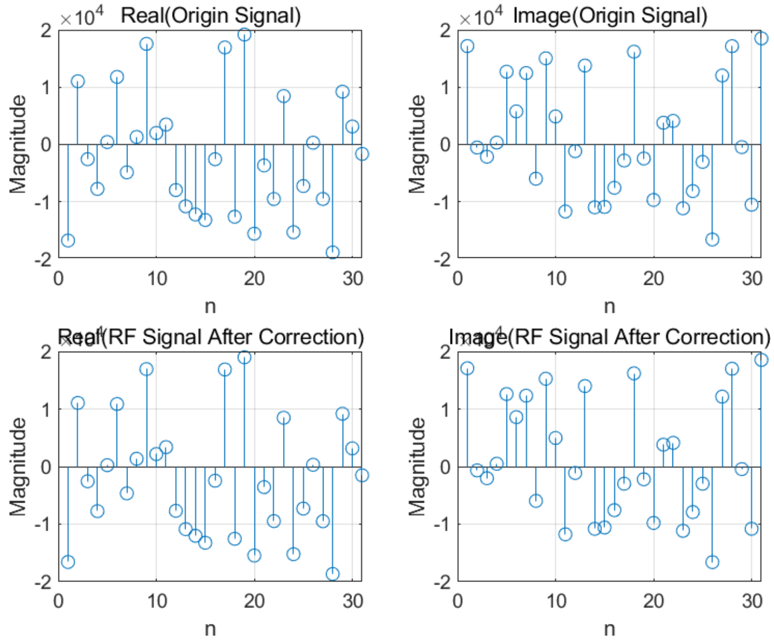

Figure 17: Recovered Signal(Random Generated)

We have also put some random signal into this transmit system, and the recovered signal are also got close enough to the original signal.

Some Discussion

We've also tried to let the audio file in the project 1 of the MATLAB as the input of the system(the audio which has a voice saying "How much it will cost to send a registered letter to Beijing?"). In this process, we also need to calculate the h[n] of the signal of such length, which is shown below. To reduce the calculation time and the memory space(the matrix with $(522 \cdot 1010^3) \cdot 1$ will comsume more than 10 minutes on the calculation on Block2), we adjusted the variable `dt` to 10^{-7} to reduce the size of the signal matrix.

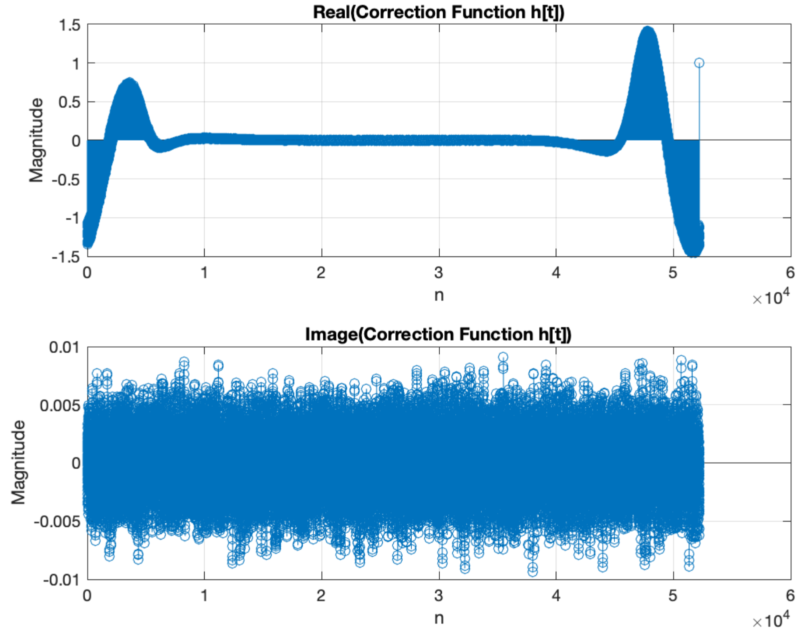

Figure 18: h[t] while N=52210

After correction, we exported the audio file.(Could be heard below) Comparing the specturm of two audio files, we found that though the audibility of the two files are basically the same, but there were many high frequancy noise which have mixed into the recovered signal, which maybe imported on the sampling process of the RF signal.

Original Audio (Click to Play)	Audio Recovered (Click to Play)

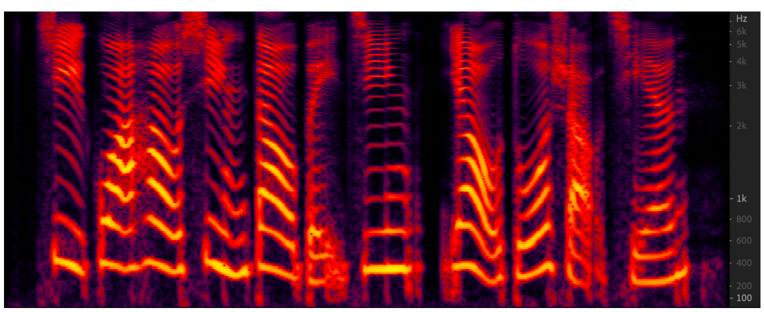

Figure 19: Spectrum of the Original Audio

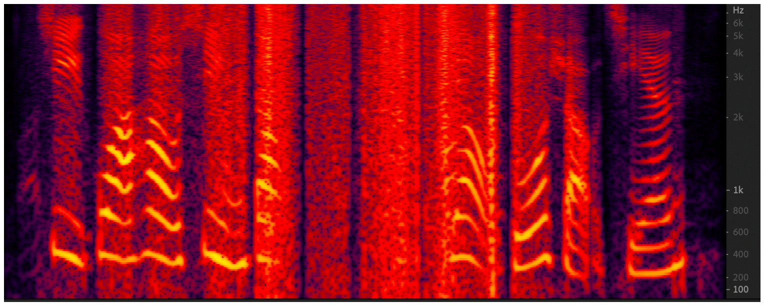

Figure 20: Spectrum of the Recovered Audio

Task D: Elaborate on the advantages of OFDM systems

Theory Part

1. Sharply increase spectral efficiency:
   In a traditional FDM system, each channel is separated apart by about 25% of the channel width, aiming to guarantee that adjacent channels do not interfere with each other. However, OFDM system allows channels to overlap, and the channel bandwidth can approach Nyquist bandwidth (The ideal spectrum utilization is 2 Baud/Hz). Hence, if the number of sub-carriers is quite large, OFDM system will nearly double the spectral efficiency than traditional FDM system.

2. Avoid ISI (Inter Symbol Interference):
   By using cyclic prefix (cp), OFDM system eliminate ISI in multipath propagation.

3. More resistant to frequency selective fading:
   In OFDM system, the channel is divided into smaller narrowband subchannels, which makes it more resistant to frequency selective fading than traditional single carrier system. Furthermore, using adequate subchannels in coding and decoding make it more possible to recover the original signal from frequency selective fading.

4. Simpler channel equalization:
   In OFDM system, the channel equalizer has the known pattern and is simpler to approach than single carrier system.

5. Simpler calculation:
   The modulation and demodulation calculation of OFDM system achieved by IFFT and FFT, which is much simpler and more efficient than traditional FDM system.

6. Less sensitive to sample timing offsets than single carrier systems.

7. Provides good protection against co-channel interference and impulsive parasitic noise.

Application Part

1. Usually, the wireless data services have the feature of asymmetry: the amount of data in downloading is much larger than that in uploading. OFDM system can easily satisfy the need of asymmetric transmission rate by changing the number of sub-channels in downloading and uploading.

2. OFDM system can be easily combined with other access technologies, resulting in OFDMA system, such as MC-CDMA, FH-OFDM and OFDM-TDMA. The OFDMA system allows multiple users to use OFDM technology simultaneously.

3. The transmitting ability of communication path varies with time, while OFDM system can change the related carriers to adapt the variation dynamically.

4. Since OFDM system can eliminate ISI in multipath propagation, it is quite suitable to be applied in the transmission of data and radio signals in the tall buildings and residential area, where the transmission measure that can reduce multipath propagation effect is highly needed.

Task Assignment

- Receiver design and analysis by Yang
- Matlab Code and document composing by Fan
- Abstract, introduction and Task D Description by Xu
- Task C Description by Han

END OF THE REPORT