Development of Orthogonal frequency division multiplexing system in MATLAB

Student:
Al-Nussairi Ahmed kateb jumaah

Supervisor:
PhD., Assoc. Prof.
Tatyana Makovetskaya
Goal

Development and analyzing of orthogonal frequency division multiplexing system in MATLAB.

Objectives

1. Analyze the subject area.
2. Structure of Project.
3. Design the system.
4. Analyze of tool that used for development OFDM.
5. Implement model(coding).
6. Test of system.
7. Result of project.
Subject area

Subject area we can send packages from information's on the multiple levels without losses in data, OFDM used in applications such as digital television and wireless networks, audio broadcasting, internet access, 4G mobile communications.
Structure of Project

- 11 Files
- 18 Interfaces
- 588 Lines of source code
Functions

1) Analyze the data (video, audio);
2) Send packages from information into the receiver;
3) Add a new data on the same channel;
4) Detection and correction of error in data;
5) Searches for data if it happens to be lost;
6) Convert the data to other expression;
7) Matching between the sender and receiver;
8) Recovery original data;
9) Display the status of sender and receiver;
10) Remove the noise and attenuation from signal.
We used MATLAB because it has built-in function “ifft()” during this function I develop this project.
Generation and analyzing original data

clear all;
close all;

% Initiation
% no_of_data_bits = 64; %Number of bits per channel extended to 128
M = 4; %Number of subcarrier channel
n = 256; %Total number of bits to be transmitted at the transmitter
block_size = 16; %Size of each OFDM block to add cyclic prefix
cp_len = floor(0.1 * block_size); %Length of the cyclic prefix

% Transmitter
%
% Source generation and modulation
%
% Generate random data source to be transmitted of length 64

originaldata.m

1 - figure(1), stem(data); grid on; xlabel('Data Points'); ylabel('Amplitude')
2 - title('Original Data')
3 - psk_modulated_data = pskmod(data, M);

Result code

Analysis table

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>block_size</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>cp_len</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data</td>
<td>1x64 double</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>256</td>
<td></td>
</tr>
<tr>
<td>no_of_data_bits</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>
QPSK Modulation (Shifting in data)

```matlab
figure(2), stem(qpsk_modulated_data); title('QPSK Modulation ')
%
%
% Converting the series data stream into four parallel data stream to form
% four sub carriers
S2P = reshape(qpsk_modulated_data, no_of_data_bits/M,M)
Sub_carrier1 = S2P(:,1)
Sub_carrier2 = S2P(:,2)
Sub_carrier3 = S2P(:,3)
Sub_carrier4 = S2P(:,4)
```

**Result**

**QPSK**

**Analyze**

**QPAS Curve**
### Example: OFDM sender with BPSK

Input data is 1010111010001010011

<table>
<thead>
<tr>
<th>Time</th>
<th>D0</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>T4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Data shifting

<table>
<thead>
<tr>
<th>Time</th>
<th>S0</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>T2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>T3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>T4</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Note:** for calculating data in BPSK we should use \( \Delta = \sin(2\pi FT) \)

Where:
- \( \Delta \) is variable between two data;
- T is Time;
- F is frequency;
- \( F = \frac{1}{T} \)

\( (2\pi FT) \) is like \( \Theta \)
Dividing data by sub-carrier

```matlab
figure(3), subplot(4,1,1), stem(Sub_carrier1), title('Subcarrier1'), grid on;
subplot(4,1,2), stem(Sub_carrier2), title('Subcarrier2'), grid on;
subplot(4,1,3), stem(Sub_carrier3), title('Subcarrier3'), grid on;
subplot(4,1,4), stem(Sub_carrier4), title('Subcarrier4'), grid on;

\% IFFT OF FOUR SUB_CARRIERS

number_of_subcarriers = 4;
op_start = block_size - op_len;
i.fft_Subcarrier1 = i.fft(Sub_carrier1);
i.fft_Subcarrier2 = i.fft(Sub_carrier2);
i.fft_Subcarrier3 = i.fft(Sub_carrier3);
i.fft_Subcarrier4 = i.fft(Sub_carrier4)
```

Output

- `Subcarrier1`
- `Subcarrier2`
- `Subcarrier3`
- `Subcarrier4`

Analyzing

- Workspace
  - `n`: 256
  - `no_of_data_bits`: 64
  - `number_of_subcarriers`: 4
  - `qpsk_modulated_S2P`: 1x64 complex double
  - `S2P`: 16x4 complex double
  - `Sub_carrier1`: 16x1 complex double
  - `Sub_carrier2`: 16x1 complex double
  - `Sub_carrier3`: 16x1 complex double
  - `Sub_carrier4`: 16x1 complex double
IFFT On all subcarrier

```matlab
figure(4), subplot(4,1,1), plot(real(fft_Subcarrier1),'r'),
title('IFFT on all the sub-carriers')
subplot(4,1,2), plot(real(fft_Subcarrier2),'c')
subplot(4,1,3), plot(real(fft_Subcarrier3),'b')
subplot(4,1,4), plot(real(fft_Subcarrier4),'g')

% ADD CYCLIC PREFIX %

for i=1:number_of_subcarriers,
    ifft_Subcarrier(:,i) = ifft(S2P(:,i)),16) % 16 is the ifft point
end
for j=1:cp_len,
cyclic_prefix(:,i) = ifft_Subcarrier(j+cp_start,1)
end
Append_prefix(:,i) = vertcat( cyclic_prefix(:,i), ifft_Subcarrier(:,i))
% Appends prefix to each subcarriers
end
A1=Append_prefix(:,1);
A2=Append_prefix(:,2);
A3=Append_prefix(:,3);
A4=Append_prefix(:,4);
```
Recovery original data

```
1 - figure(10)
2 - stem(data)
3 - hold on
4 - stem(qpsk_demodulated_data,'rx');
5 - grid on;xlabel('Data Points');ylabel('Amplitude');
6 - title('Received Signal with error')
7 -
```

Some error
Publications

Link To verify this book
https://www.facebook.com/327198190760666/photos/a.755401181273696.1073741828.327198190760666/840996569380823/?type=3&theater
We make testing for company ZAIN in Iraq for mobile cells about send and receive data and we found that all data depend on technology OFDM so that we knew.

1. We sent all information sent on the one channel without overlap.
2. High efficiency and less losses.
3. Each user he has specific code and this code can be divider to other code at any time.
The results

1. Analysis the performance codes for OFDM technology in MATLAB.
2. We measured the percentage of errors in codes.
3. The simulation is able to simulate bit error rate for research purpose as well as education purpose.
4. Multiple levels of OFDM simulation to characterize signal to noise ratio performance.