IOP Conf. Series: Materials Science and Engineering **231** (2017) 012054 doi:10.1088/1757-899X/231/1/012054

# Orthogonal frequency division multiplexing simulation based on MATLAB

**Yuan Qiao**

Department of Biomedical Engineering, Southern University of Science and Technology, Guangdong 518055, China

**Abstract.** OFDM (Orthogonal Frequency Division Multiplexing) is one of the core technologies in the fourth generation mobile communication system. It is a widely-used method of the multi-carrier modulations based on IFFT and FFT transform, it can achieve the lowest complexity and effectively combat frequency selective fading. In this paper, we successfully use MATLAB to do the simulation of OFDM, and obtained good results, in which successful recovery out of the original signal under real channel condition, and error is less than 5% with the original signal.

## 1 Introduction

With the development of electronic technology, computer science and internet gradually come into our daily life, becoming an indispensable part of our life. With the rapid development of science and technology, people's demand of information is growing and growing, promoting wireless communication into a rapid developing period [1]. From the first generation of wireless communication which using frequency division multiple access technology (FDMA) to construct analogue cellular network, to the 2G era of the second generation of digital communication, and then gradually developed to 3G broadband communication which based on circuit switching [2]. Although broadband has become the main development direction in the field of communication at present, there are still many deficiencies which will affect the quality and performance of digital communication system. Such as the frequency selective fading channel, intersymbol interference and noise. On this basis, there are many efficient way of modulation, of which Orthogonal Frequency Division Multiplexing (OFDM) is a very effective method [3].

OFDM, orthogonal frequency division multiplexing technology, is a kind of method widely used in multicarrier modulation of minimum complexity, which based on IFFT (inverse fast Fourier transform) and FFT (fast Fourier transform) technology. As is known to all, in a communication system, the signal bandwidth to transmit a signal is usually much more smaller than the bandwidth we provided, which creates a waste of channel bandwidth. Therefore, in order to reduce the waste of channel bandwidth, make full use of the bandwidth of the channel, gives rise to the method of orthogonal frequency division multiplexing.

The principle of orthogonal frequency division multiplexing is dividing the channel into several sub orthogonal channel, then transfer the data flow through serial to parallel conversion and distribute it to the several orthogonal channel for transmission. In this process, each cycle of sub-channels increases a set period, making cycle be multipath time delay while at the same time guarantee the channel’s orthogonal. This method makes full use of the orthogonality of the signal, eliminates intersymbol interference, let the channel bandwidth have truly effective use. At the same time, because of its use of fast Fourier transform method, it can use digital signal processing methods, making OFDM more simple and feasible.

IOP Conf. Series: Materials Science and Engineering **231** (2017) 012054 doi:10.1088/1757-899X/231/1/012054

In this study, a simulation for the whole process of OFDM and its transmission environment has been done, we successfully simulate the whole process of the generation and propagation of the signal, realizing the signal recovery.

## 2 Methods

*2.1 Selection of signal*

In this experiment, in order to reflect the OFDM system's effect of random signal modulation and recovery, I choose the "randn" function in MATLAB, which can quickly and simply generate a string of specified length of "double" type digital string, and use the array to realize the simulation of the real signal. And because each time when we run the program by using this function, it generate a series of random signal, by comparing the results with the original signal, it can have more obvious contrast with the degree of signal reduction. In this experiment, we selected the 32 digits for signal processing and restoration.

*2.2 Front end signal processing*

In front end processing, we will use a string of random signal whose length is 32 to be converted from series to parallel string in order to transfer one series of signal to 32 series of signal and be transmitted separately [4, 5]. Then use IFFT to handle the 32 channel signal so as to modulate the 32 data to multiple orthogonal sub-carriers. In order to prevent signal multipath propagation and the interference between different carriers, we selected 8 data at the back of the signal after IFFT treatment as the front end of the cycle prefix and added to the signal in order to realize the maximum reduction of original signal information. The front-end processing part is reflected in the code "block.m".

After addition of cyclic prefix, we want to realize D/A conversion which is converting digital signal to analogue signal. Since the MATLAB is matrix factory, difficult to achieve real analogue signal, so we extend one data to 200 data points to accomplish D/A conversion. After converting digital signals into analogue signals, we will separate the real part and imaginary part of the signal and then carry on carrier Sinusoidal and cosine wave whose frequency is $\omega_c = 1\times10^8$, and by means of addition will combined the two signal into one signal. Through such processing, the signal can be transmitted through real path. This part of the code in the program "block2. m".

*2.3 Simulation of real transmission channel.*

Due to the process of this experiment is intended to simulate the OFDM, and don't have real experiment to do the transmission, so we need to make an estimate of the transmission equation of real channel. In this experiment, we simulate a linear time invariant system. So in this case, I assume that impulse response for the real transmission channel is:

$$\begin{aligned} \mathrm{h}(\mathrm{t}) &= 0.5\delta(\mathrm{t}) + 0.4\delta(\mathrm{t}-1.5\mathrm{T}) \\ &+0.35\delta(\mathrm{t}-2.5\mathrm{T}) + 0.3\delta(\mathrm{t}-3\mathrm{T}) \end{aligned} \tag{1}$$

From using D/A transition to transfer it into analogue signal of certain time interval, then skillfully use the "filter" equation in MATLAB to add the shock response of the real transmission channel into signals we want to transmit. Later in the transmission, we think the signal after "filter" function is the signal which has been through real signal of channel transmission.

*2.4 Signal receiving and later processing.*

As we know, at before we respectively carry on sine and cosine carrier to the real part and imaginary part of the signal and make an addition for transmission, so after receiving the signal, the first step to do is restoring signal's real part and imaginary part. We know that signal after we carry on the carrier signal is:

$$\mathrm{x}(\mathrm{t}) = x_r(t)*\cos(\omega_c t) + x_i(t)*\sin(\omega_c t) \tag{2}$$

So in order to achieve the restoring of the real and imaginary part, we multiply another $\cos(\omega_c)$ and $\sin(w_c)$ as follows:

$$\begin{aligned}&\mathrm{x}(\mathrm{t})*\cos(\omega_c \mathrm{t})=\frac{1}{2}x_r(t)+\frac{1}{2}\cos(2\omega_c \mathrm{t})\\&+\frac{1}{2}x_i(\mathrm{t})*\sin(2\omega_c \mathrm{t})\end{aligned} \tag{3}$$

And:

$$\begin{aligned}&\mathrm{x}(\mathrm{t})*\sin(\omega_c \mathrm{t})=\frac{1}{2}x_i(t)-\frac{1}{2}\cos(2\omega_c \mathrm{t})\\&+\frac{1}{2}x_r(\mathrm{t})*\sin(2\omega_c \mathrm{t})\end{aligned} \tag{4}$$

We can see from the two formula that after combined with the modulation of A and B, from the frequency domain, the signal is divided into two parts, the real part and imaginary part on low frequency and high frequency parts on $w_c$. Easily know that if we filter out the high frequency part and remain the rest part on low frequency, we can restore the original real and imaginary part. Therefore, in this part of the processing, we first make the received signal multiplied by $\cos(\omega_c)$ and $\sin(w_c)$ respectively, and then use FFT functions to transfer two signals to the frequency domain, using a ideal high-pass filter whose gain is 2 to filter out high frequency part and leave only low frequency part, then inverse transform to the time domain. Through these process, we can get the real and imaginary part we want. After doing this, we let our real and imaginary part of signal go through A/D transmission respectively and then make an addition. We can receive the signal. This part of the code in block3. m.

In the process of transmission, while the signal is random or different, but not the channel is, therefore in this experiment, I use another bunch of random Numbers traverse the entire process, through the division of given results and original signal to get the channel's frequency response.

In order to protect the signal from interference, we add cyclic prefix on the signal at before, so after reverting this signal, we will cut off the cyclic prefix and then to do series to parallel transmission. Then do DFT transform to transfer it to the time domain and remove the channel gain and channel frequency response. Then we can get the original signal. This part of the code in "block4. m".

## 3 Results

### *3.1 Front end signal processing*

We start with the front-end signal processing part of the results, in Fig. 1, 1 a is our randomly generated original signal, we can see the MATLAB generated a signal which length is 32 in the distribution of the random number between -3 to 2, this is the original signal we're going to undertake transfer and reduction. Then, we do IFFT transformation to the signal and combined with the cyclic prefix in order to avoid the loss of signal. We can see 1 b is the signal after processing, compared to 1 a, 1 b has passed from 32-bit array to 40 array. Later, in 1 c, we respectively carry on sine wave and do digital analogue conversion to the real part and imaginary part of the signal. 1c for analogue signals we get. 1d is the performance of the signal being transformed to the frequency domain. We can see the data has been carried on the sine wave.

### *3.2 Simulation of the real environment transmission and post-processing*

In this part, according to our previous method, we use the "filter" function in MATLAB to simulate the real channel. After this, in order to demodulate the signal and get the real part and imaginary part we transported, we take on its carrier again. 2a is the frequency domain response after they take on

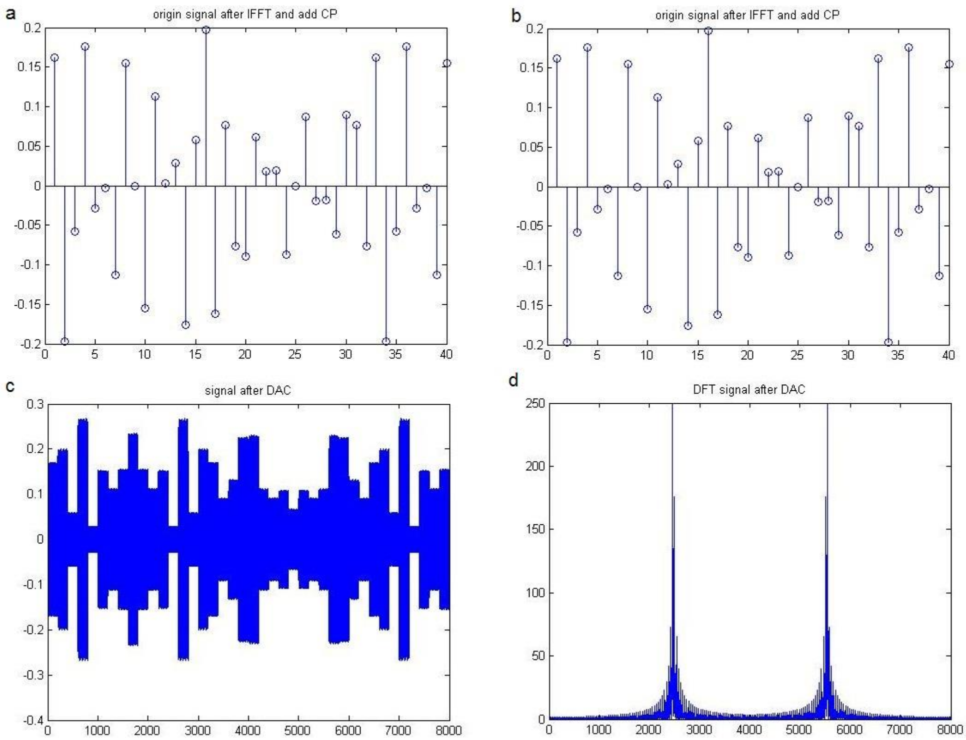


Figure 1 Front end signal processing. a). Original random signal produced by MATLAB. b). Signal after IFFT transform and addition of cyclic prefix. c). Signal after D/A transmission. d). Frequency performance of signal after D/A transmission.

the carrier. Here we use a function “fftshift” to move it to center position, so the frequency of 4000 in this picture actually is zero frequency. We can see in 2a, signals in real and imaginary part are all have a high peak both in the low frequency and high frequency, which we have proved before. Here, we just need to filter out the portion of the high frequency can we get the real part and imaginary part of the original signal what we want. Here, we directly use the method of setting the data to 0 to make an ideal low-pass filter to filter out the portion of the high frequency directly. 2b is the real and imaginary part of the signal we get after an ideal low-pass filter. We can see that although at this time, the signal still has many spikes, the reduction of the real part and imaginary part of the signal can be seen clearly.

Later, we will add the real part and imaginary part together and do analogue-digital conversion to convert it to digital signal. 2c is the digital signal after conversion. we can see that at this point, the signal has changed to 40 number of digital signal again. At past, because of the need to protect the signal from being damaged, we added the cyclic prefix, so here, we remove the cyclic prefix, 2d is the signal after removing cyclic prefix.

Because we don't know the channel frequency response in the process, we use the method we recommend before, first use the string of sequence to go through the whole process, and then in turn get frequency response, 2e is the whole channel frequency response we get. After we do the DFT and then removing the gain, we get the reduction of signal. 2f is the signal we restore.

### *3.3 Comparison of results and original signal.*

In order to make the result looks more clearly, here we made a comparison of the results and the original signal. 3a was the figure of the original signal, 3b for reduction of signal, 3c for the differences between them. We can see that under the same coordinate system, the difference is very small, can say, we are still good reduction results.

## 4 Conclusion

In this paper, we successfully use MATLAB to do the simulation of OFDM, and obtained good results, in which successful recovery out of the original signal under real channel condition, and error is less than 5% with the original signal. In this experiment, we use MATLAB to realize the transform of series and parallel, analogue and digital conversion and the signal modulation and demodulation. All of the function we used in this experiment is the basic function of MATLAB, the overall programming is very concise and effective. In the process of simulation, the functions provided MATLAB, including the properties of matrix operations, both provided a lot of convenient for our experiment. Orthogonal frequency division multiplexing technology is a very important part of communication engineering, its strong anti-interference ability, high spectrum efficiency, strong antifading ability and flexible allocation of resources can achieve very high efficient and good results. In this experiment, we still have a lot of deficiencies. For example, in the real channel simulation, we simulate it by multiplying a shock response equation to it and

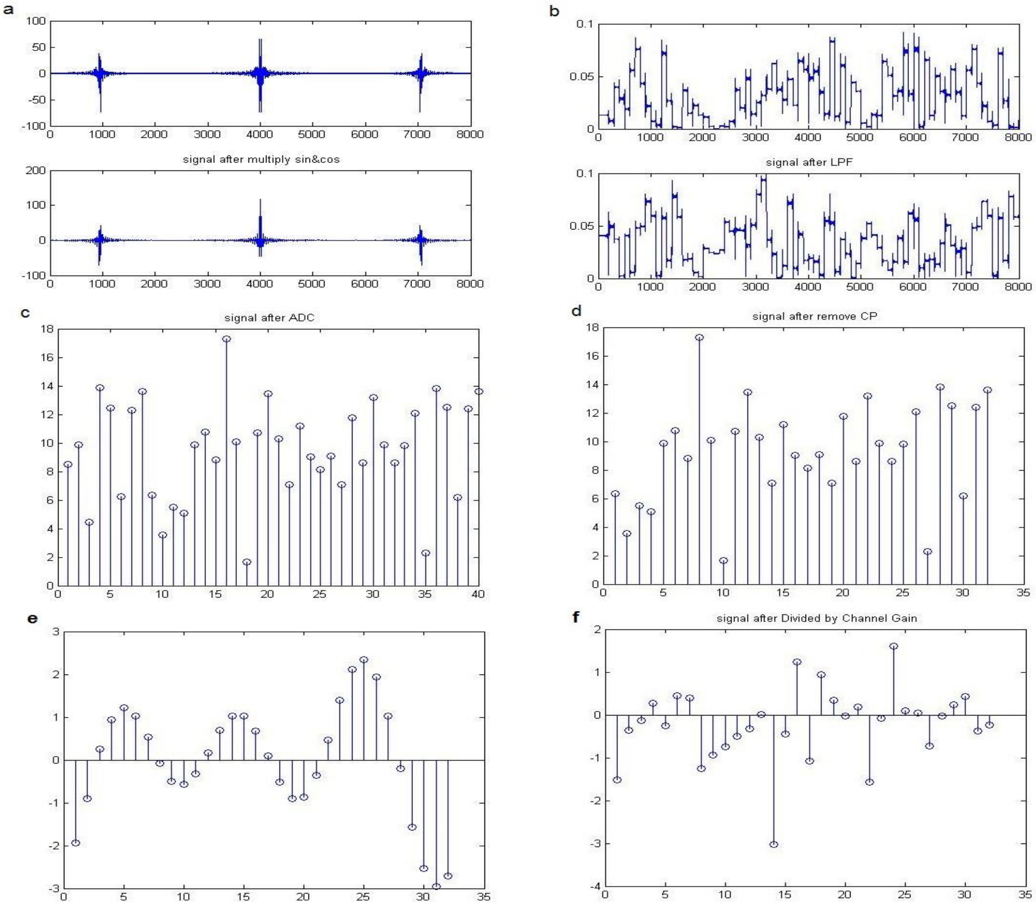


Figure 2 Receiving of the signal and post processing. 2a). Frequency response of received signal. 2b). Real and imaginary part after low-pass filter. 2c). Digital signal after ADC. 2d). Signal after cutting off the cyclic prefix. 2e). Frequency response from pi

did not really restore the transmission environment. In the process of signal demodulation, we don't use other filters, but directly use the MATLAB matrix factory’s feature and produced an ideal low-pass filter for filtering and demodulation. But taken together, the experiment successfully simulated the whole process of OFDM, and complete the reduction of the original signal, completed the whole process of the signal from production to early treatment, to real channel transmission and to demodulation.

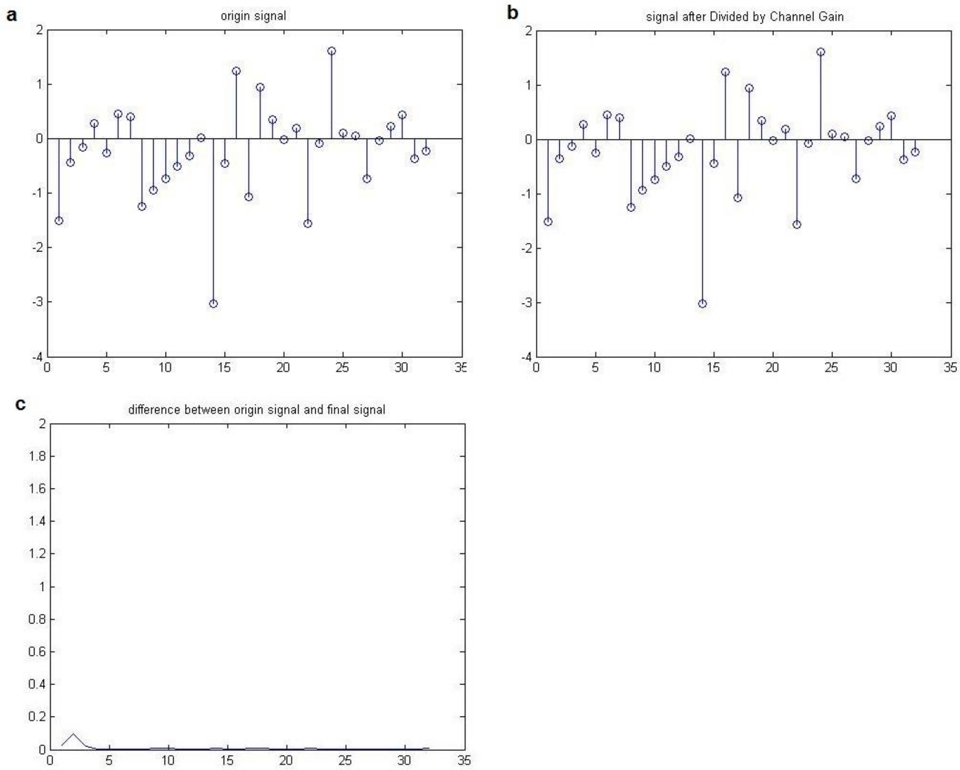


Figure 3 Comparison of results and original signal. 3a). Original signal. 3b). Results signal we get. 3c). Differences between original signal and results signal.

## References

[1] Moore, G. C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation.Information systems research, 2(3), 192-222.
[2] Holma, H., & Toskala, A. (Eds.). (2009). LTE for UMTS-OFDMA and SC-FDMA based radio access. John Wiley & Sons.
[3] Nee, R. V., & Prasad, R. (2000). OFDM for wireless multimedia communications. Artech House, Inc..
[4] Cho, Y. S., Kim, J., Yang, W. Y., & Kang, C. G. (2010). MIMO-OFDM wireless communications with MATLAB. John Wiley & Sons.
[5] Acosta, G. (2000). OFDM simulation using matlab. Smart Antenna Research Laboratory report.