CHAPTER – 1

INTRODUCTION

1.1 Background and Motivation

In present situations, medical images (mainly of radiographs - e.g. X-rays) become compulsory in all the health centers to diagnose the diseases in the human body. Tele radiology (the transmission of Radiographs using communication networks) is useful in places where the population is widely scattered over a large geographical area so that access to health centers is difficult. This technique is also useful in situations when the time of diagnosis is critical as an emergency in hospitals.

The demand for medical image interpretation services in radiology is developing rapidly all over the world. This is because of the lack of adequate staff for providing specialty expertise. These problems can be overcome by utilizing communication to draw on the expertise of distantly located radiologists

Some of the current goals of tele-radiology include making radiological consultations available in medical facilities without on-site radiological support, enhancing educational opportunities for practicing radiologists and providing timely availability of radiological images and its interpretations in emergent and non-emergent clinical care areas

There is a necessity of enable the proper interpretation and advice of the radiologists about the disease to the remote, rural and unserved population. Medical image transmission to the specialized doctors is one of the approaches in the form of improved access and reduced cost to the rural patients. This enables the specialist doctor to assess the physical and psychological state of the patient and suggest treatment. This remote consultation and treatment is much more valuable in case of post operation (Post Surgery) follow up since the patient is not required to travel unnecessarily and hence saving money and time.

The different medical imaging modalities are X-ray radiography, magnetic resonance imaging, ultrasonography, endoscopy, elastography, tactile imaging, thermography etc. These images can be utilized for timely interpretation and also provide greater access to Physicians about the patient condition. Also allows specialists to view medical images in various locations and they provide service. The standard technologies such as LTE, Raspberry PI processor are used. LTE is a third party dependent mode of wireless communication. It has a recurring cost and depends on public wireless communication domain though it is less cost. By utilizing LTE technology we can achieve wireless communication up to far greater distances as compared to point to point communication with a very low competitive cost. Quality of the medical image should be good otherwise it may end up with erroneous result. The necessity of secure transmission is vital in the medical world.

1.2 Statements of Problems:

In case of remote communication, when a physician and a patient are located in different geographical locations, it is a major problem to communicate between them specially communication with medical image. To communicate between physician and patient present technology used OFDM but upcoming technologies will be use SC-FDMA for both uplink, and OFDMA downlink. SC-FDMA reduces PAPR, which reduces power consumption. Very valuable in the UE, but less so in the eNB where the PA is carrying multiple users and multiple signals (which might make it harder) and power is "cheaper". Contrarily, SC-FDMA needs a lot of extra computation at the receiver (more power, more complex DSP). For the UL that is fine, as the complexity is in base station and a mortised over lots of users, but would be a bad thing in the UE. SC-FDMA transfers costs & power from the sender to the receiver; that is a good thing to shift them from UE to eNB but so far there has been no motivation to do the reverse. One more reason on why SC-FDMA is not used on the downlink is, it would limit the resource allocation flexibility on the downlink. As the name implies SC (Single Carrier) meaning that on the UL the radio resource allocation has to be contiguous. But on the other hand on the downlink with OFDMA, we do not have this limitation

1.3. Objectives

This research is transmitting a medical image using Single-Carrier Frequency Division Multiple Access (SCFDMA) system.

Our first objective is to transmit a medical image through SCFDMA and reconstructing it.

Second objective is performance analyze using different DCT based SCFDMA and DFT based SCFMA through which will try to understand which technique gives more enhanced image.

Finally we will try to visualize whole research through graph by Peak Signal to Noise Ratio (PSNR) values in the received images.

1.4. Organization of the Thesis

The following is a detailed outline of the remaining chapters of the thesis.

In chapter 2, Literature reviews are given. In literature review, we have showed related work on existing system.

In chapter 3 this chapter is for an introductory information about image and medical image, the usefulness of medical image transmission.

In chapter 4, we have discussed Long Term Evolution (LTE) system. And discussed Single-Carrier Frequency Division Multiple Access (SCFDMA) system with necessary figure and tables.

In chapter 5, we have discussed over simulation techniques and results.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction:

The wireless devices have advances the medical world with a wide range of capability by providing the quality of life of patients. Wireless technology enables clinicians to monitor patients remotely and give them timely health information, reminders, and support potentially extending the reach of health care by making it available anywhere, anytime. The use of wireless technologies in medical environments is bringing major advantages to the existing healthcare services.

75% of doctors are practice in the urban areas as mentioned above so that in rural area people will not get timely consultation and advice of especially radiologists. X-rays scanning facility can be provided but the consultation of radiologist is quite difficult task. If rural and remote area people want to consult a doctor it may be expensive and sometimes they could not reach them according their requirement. As we seen in the day today life there will be a fake doctors around us who do not held the proper degree and cheating the people for money. In order to overcome these problems the wireless transmission and reception of the medical image using GSM/GPRS technology can be used. By taking the doctor advice people can be careful with the health issues by asking the second opinion of specialized radiologist in their own place with less cost.

2.2 Existing System

Some of the advanced technologies in the medical field are, a real time health monitoring system of remote patient developed is a wearable device. This device will be wearied by the patient and parameters such as Temperature, Heart Beat and ECG will be continuously transmitted and monitored through wireless technology ZigBee. The radiographic images can be viewed from any workstation within its network. The basic functions carried out by PACS are image retrieval, image transfer, viewing and networking .In medical imaging, the fast delivery of medical reports to referring medical practitioners is a major component of cooperative patient care. Recently, active involvement of smart phones brought effectiveness in telemedicine applications. The phone is a best medium to carry through faster delivery of information to the medical practitioners. An electronic medical report delivery system from a medical imaging department to the mobile phones of the referring doctors. The system which contains both text summary of medical report and a medical image in JPEG format, which are transmitted to 3G GSM mobile phones.

CHAPTER 3

MEDICAL IMAGE

3.1. Image

An image is a picture that has been created or copied and stored in electronic form. An image can be described in terms of vector graphics or raster graphics. An image stored in raster form is sometimes called a bitmap. An image map is a file containing information that associates different locations on a specified image with hypertext links.

Common image file formats include:

- **JPEG** (pronounced JAY-peg) is a graphic image file produced according to a standard from the Joint Photographic Experts Group, an ISO/IEC group of experts that develops and maintains standards for a suite of compression algorithms for computer image files. JPEGs usually have a .jpg file extension.

- **GIF** (pronounced JIF by many, including its designer; pronounced GIF with a hard G by many others) stands for Graphics Interchange Format. The GIF uses the 2D raster data type and is encoded in binary. GIF files ordinarily have the .gif extension.

- **PNG** (pronounced *ping*) is a Portable Network Graphics) is a file format for image compression that was designed to provide a number of improvements over the GIF format. Like a GIF, a PNG file is compressed in lossless fashion (meaning all image information is restored when the file is decompressed during viewing). Files typically have a .png extension.

- **SVG** is Scalable Vector Graphics, the description of an image as an application of XML. Any program such as a browser that recognizes XML can display the image using the information provided in the SVG format. Scalability means that the file can be viewed on a computer display of any size and resolution, whether the small screen of a smartphone or a large widescreen display in a PC. Files usually have .svg extension.

- **TIFF** (Tag Image File Format) is a common format for exchanging raster graphics (bitmap) images between applications programs, including those used for scanner images. A TIFF file can be identified as a file with a .tiff or ".tif" file name suffix.

3.2. Medical Image

Medical imaging is a technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some organs or tissues. Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities. Although imaging of removed organs and tissues can be performed for medical reasons, such procedures are usually considered part of pathology instead of medical imaging.

Medical imaging is often perceived to designate the set of techniques that noninvasively produce images of the internal aspect of the body. In this restricted sense, medical imaging can be seen as the solution of mathematical inverse problems. This means that cause (the properties of living tissue) is inferred from effect (the observed signal).

There are various sorts of medical images. Some of this conclude:

- Plain X-ray
- Computed Tomography (CT)
- Magnetic Resonance Imaging (MRI)
- Ultrasound

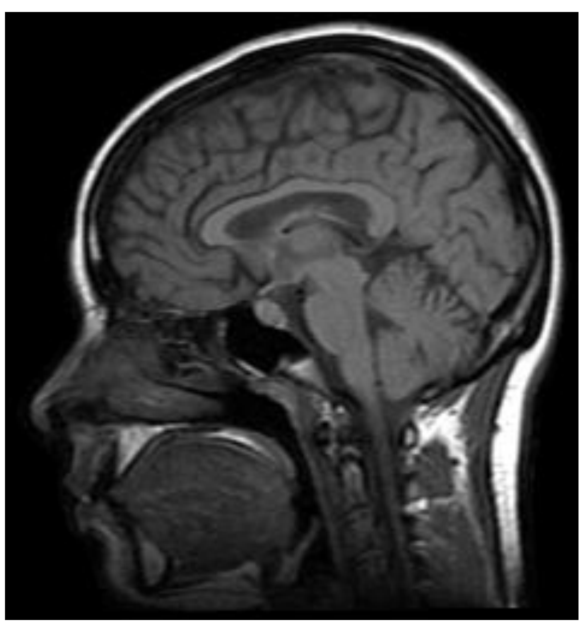

Fig 3.1: Brain - MRI Image

3.3. Utilization of Medical Image

Image transmission has a broad range of applications such as business applications, medical imaging, acoustic imaging, and forensic sciences, As per Medical Imaging is concerned most of the images may be used in the detection of tumors or for screening the patients.

- **Care Facilities**: Institutions use medical image sharing to facilitate transfers between other facilities that may or may not be on the same network. They are also able to instantly send results to referring physicians in the community, as well as directly to patients.
- **Physicians**: Doctors use the technology to have immediate access to images, as opposed to waiting for physical media to arrive. Having access to a patient's medical history improves the point of care service.
- **Patients**: In conjunction with recent US government initiatives, patients are able to receive their imaging exams electronically, without needing to carry and store physical media. It allows for the ability to see physicians in multiple locations and have their imaging at the ready.
- Improved access to patients' medical imaging histories
- Ability to view images instantly
- Real-time collaboration by specialists
- Avoiding duplicate care reduces costs
- Decreased radiation exposure for patients
- Expertise and specialized opinion is remotely accessible to patients

CHAPTER 4

LONG TERM EVOLUTION (LTE) SYSTEM

4.1. Introduction

LTE stands for *Long Term Evolution* and it was started as a project in 2004 by telecommunication body known as the Third Generation Partnership Project (3GPP). SAE (System Architecture Evolution) is the corresponding evolution of the GPRS/3G packet core network evolution. The term LTE is typically used to represent both LTE and SAE.

LTE evolved from an earlier 3GPP system known as the Universal Mobile Telecommunication System (UMTS), which in turn evolved from the Global System for Mobile Communications (GSM). Even related specifications were formally known as the evolved UMTS terrestrial radio access (E-UTRA) and evolved UMTS terrestrial radio access network (E-UTRAN). First version of LTE was documented in Release 8 of the 3GPP specifications.

A rapid increase of mobile data usage and emergence of new applications such as MMOG (Multimedia Online Gaming), mobile TV, Web 2.0, streaming contents have motivated the 3rd Generation Partnership Project (3GPP) to work on the Long-Term Evolution (LTE) on the way towards fourth-generation mobile.

The main goal of LTE is to provide a high data rate, low latency and packet optimized radio access technology supporting flexible bandwidth deployments. Same time its network architecture has been designed with the goal to support packet-switched traffic with seamless mobility and great quality of service.

4.2. Long Term Evolution (LTE)

LTE, the successor technology not only of UMTS but also of CDMA 2000, is important because it brings up to 50 times performance improvement and much better spectral efficiency to cellular networks. It introduced to get higher data rates, 300Mbps peak downlink and 75 Mbps peak uplink. In a 20MHz carrier, data rates beyond 300Mbps can be achieved under very good signal conditions. LTE is an ideal technology to support high date rates for the services such as voice over IP (VOIP), streaming multimedia, videoconferencing or even a high-speed cellular modem.

LTE Evolution:

Year	Event
Mar 2000	Release 99 - UMTS/WCDMA
Mar 2002	Rel 5 – HSDPA
Mar 2005	Rel 6 – HSDPA
Nov 2004	Work started on LTE specification
Year 2007	Rel 7 - DL MIMO, IMS (IP Multimedia Subsystem)
Jan 2008	Spec finalized and approved with Release 8
Year 2010	Targeted first deployment

Table 4.1: LTE Evolution

LTE uses both Time Division Duplex (TDD) and Frequency Division Duplex (FDD) mode. In FDD uplink and downlink transmission used different frequency, while in TDD both uplink and downlink use the same carrier and are separated in Time. It supports flexible carrier bandwidths, from 1.4 MHz up to 20 MHz as well as both FDD and TDD. It is designed with a scalable carrier bandwidth from 1.4 MHz up to 20 MHz which bandwidth is used depends on the frequency band and the amount of spectrum available with a network operator.

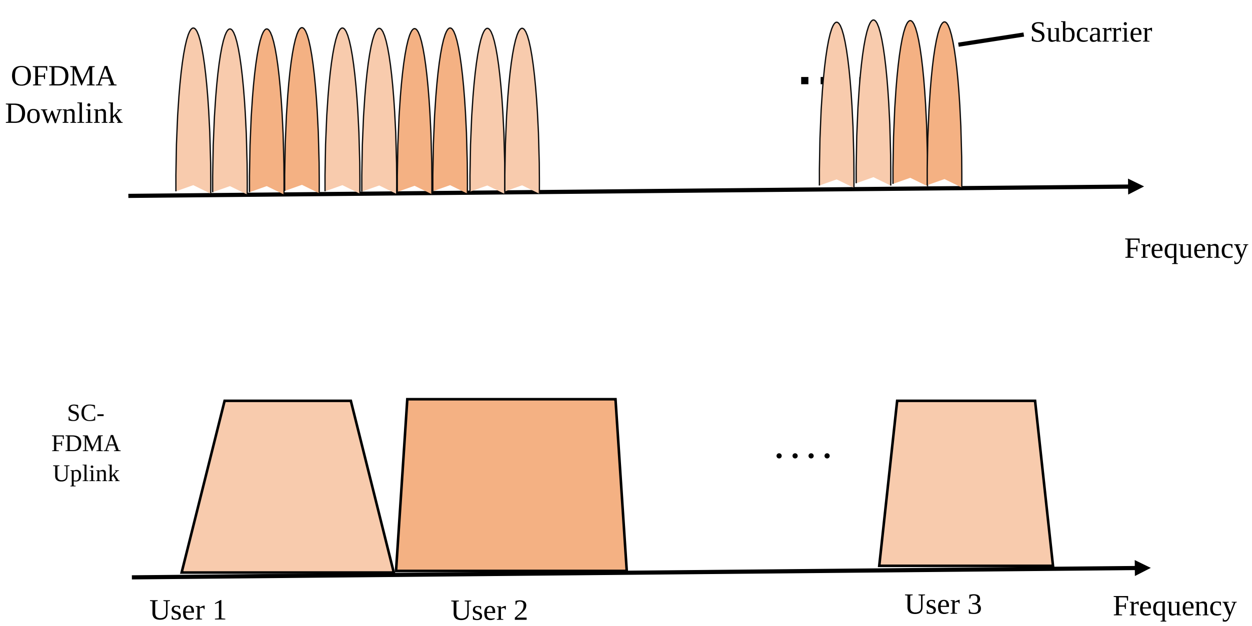

Fig 4.1: LTE uplink and downlink waveform

All LTE devices have to support (MIMO) Multiple Input Multiple Output transmissions, which allow the base station to transmit several data streams over the same carrier simultaneously. All interfaces between network nodes in LTE are now IP based, including the backhaul connection to the radio base stations. This is great simplification compared to earlier technologies that were initially based on E1/T1, ATM and frame relay links, with most of them being narrowband and expensive.

Quality of Service (QoS) mechanism have been standardized on all interfaces to ensure that the requirement of voice calls for a constant delay and bandwidth, can still be met when capacity limits are reached. Works with GSM/EDGE/UMTS systems utilizing existing 2G and 3G spectrum and new spectrum. Supports hand-over and roaming to existing mobile networks.

Advantages of LTE:

- **High throughput:** High data rates can be achieved in both downlink as well as uplink. This causes high throughput.

- **Low latency:** Time required to connect to the network is in range of a few hundred milliseconds and power saving states can now be entered and exited very quickly.

- **FDD and TDD in the same platform:** Frequency Division Duplex (FDD) and Time Division Duplex (FDD), both schemes can be used on same platform.

- **Superior end-user experience:** Optimized signaling for connection establishment and other air interface and mobility management procedures have further improved the user experience. Reduced latency (to 10 ms) for better user experience.

- **Seamless Connection:** LTE will also support seamless connection to existing networks such as GSM, CDMA and WCDMA.

- **Plug and play:** The user does not have to manually install drivers for the device. Instead system automatically recognizes the device, loads new drivers for the hardware if needed, and begins to work with the newly connected device.

- **Simple architecture:** Because of Simple architecture low operating expenditure (OPEX).

Basic parameters of the LTE:

Parameters	**Description**
Frequency range	UMTS FDD bands and TDD bands
Duplexing	FDD, TDD, half-duplex FDD
Channel coding	Turbo code
Mobility	350 km/h
Channel Bandwidth (MHz)	• 1.4 • 3 • 5 • 10 • 15 • 20
Transmission Bandwidth Configuration NRB : (1 resource block = 180kHz in 1ms TTI)	• 6 • 15 • 25 • 50 • 75 • 100
Modulation Schemes	UL: QPSK, 16QAM, 64QAM(optional) DL: QPSK, 16QAM, 64QAM

Multiple Access Schemes	UL: SC-FDMA (Single Carrier Frequency Division Multiple Access) supports 50Mbps+ (20MHz spectrum) DL: OFDM (Orthogonal Frequency Division Multiple Access) supports 100Mbps+ (20MHz spectrum)
Multi-Antenna Technology	UL: Multi-user collaborative MIMO DL: TxAA, spatial multiplexing, CDD ,max 4x4 array

Peak data rate in LTE	UL: 75Mbps(20MHz bandwidth) DL: 150Mbps(UE Category 4, 2x2 MIMO, 20MHz bandwidth) DL: 300Mbps(UE category 5, 4x4 MIMO, 20MHz bandwidth)
MIMO (Multiple Input Multiple Output)	UL: 1 x 2, 1 x 4 DL: 2 x 2, 4 x 2, 4 x 4
Coverage	5 - 100km with slight degradation after 30km
QoS	E2E QOS allowing prioritization of different class of service
Latency	End-user latency < 10mS

Table 4.2: LTE Parameters

4.3. OFDMA

The OFDMA system is a multiuser version of the OFDM system, and all that were previously mentioned about the OFDM system also hold for the OFDMA system. Each user in an OFDMA system is usually given certain subcarriers during a certain time to communicate. Usually, subcarriers are allocated in contiguous groups for simplicity and to reduce the overhead of indicating which subcarriers have been allocated to each user. One of the major problems with an OFDMA system is to synchronize the uplink transmissions, because every user has to transmit his frame so that he avoids interfering with the other users. The OFDMA system for mobile communications was first proposed in based on multicarrier FDMA, where each user is assigned a set of randomly selected subcarriers. Figure 4.2 shows the block diagram of the OFDMA system.

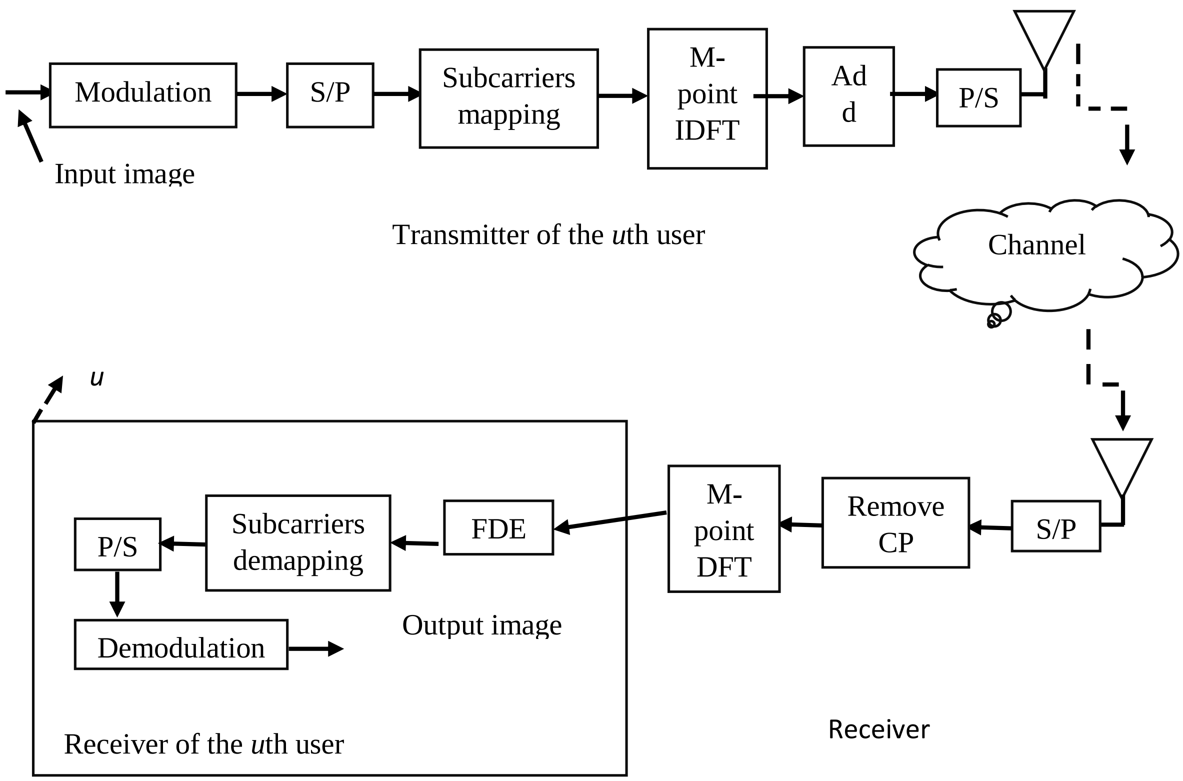

Fig 4.2: Transmitter and receiver structures of the OFDMA system over a frequency-selective channel.

4.4 OFDMA to SCFDMA

The main difference between OFDMA and SC-FDMA transmitter is the DFT mapper. After mapping data bits into modulation symbols, the transmitter groups the modulation symbols into a block of N symbols. An *N-point* DFT transforms these symbols in time domain into frequency domain. The frequency domain samples are then mapped to a subset of M subcarriers where M is typically greater than N. Similar to OFDMA, an M-point *IFFT* is used to generate the time-domain samples of these subcarriers, which is followed by cyclic prefix, parallel to serial converter, DAC and RF subsystems.

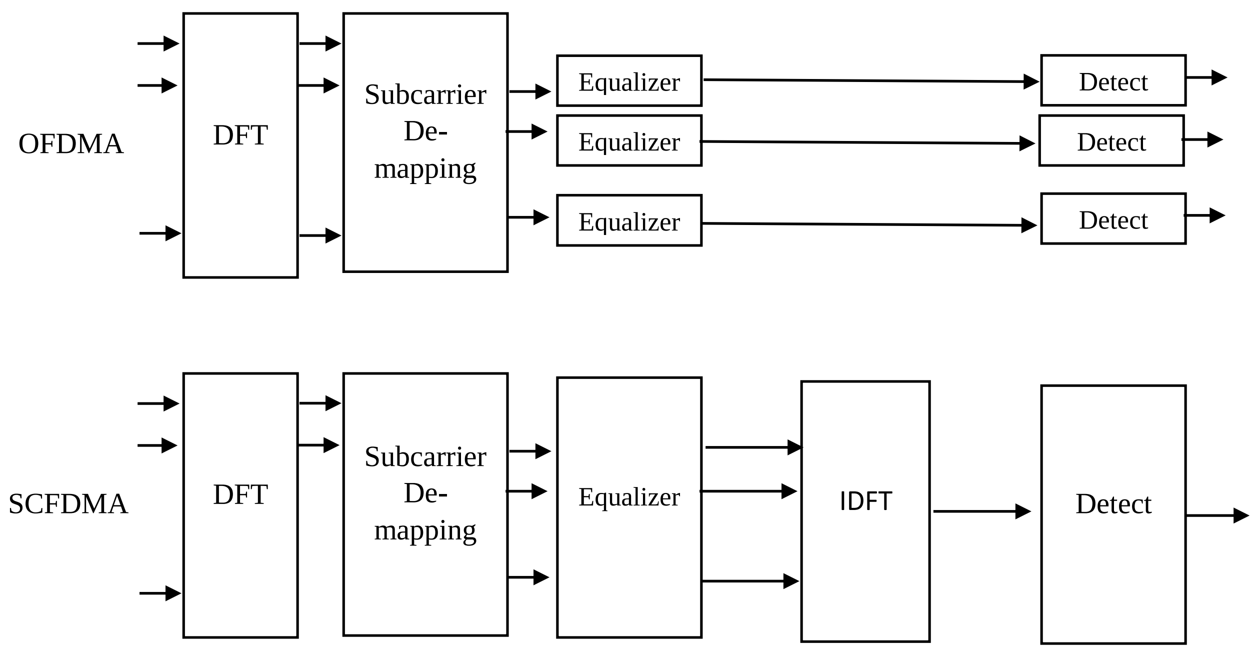

Fig 4.3: OFDMA vs SCFDMA

4.5 SCFDMA

Figure 4.4 shows an SC-FDMA transmitter sending one block of data to a receiver. The input of the transmitter and the output of the receiver are complex modulation symbols. Practical systems dynamically adapt the modulation technique to the channel quality, using binary phase shift keying (BPSK) in weak channels and up to 64-level quadrature amplitude modulation (64-QAM) in strong channels. The data block consists of M complex modulation symbols generated at a rate R source symbols/second. Figure 3.2 provides details of the three central elements of the transmitter in Figure 4.3. The M-point discrete Fourier transform (DFT) produces M frequency domain symbols that modulate M out of N orthogonal subcarriers spread over a bandwidth

$$W_{channel} = N.f_o[Hz]$$

Where f_o Hz is the subcarrier spacing. The channel transmission rate is

$$R_{channel} = \frac{N}{M}.R_{source}\ [symbols/sec]$$

If Q denotes the bandwidth spreading factor, i.e.

$$Q = \frac{N}{M}$$

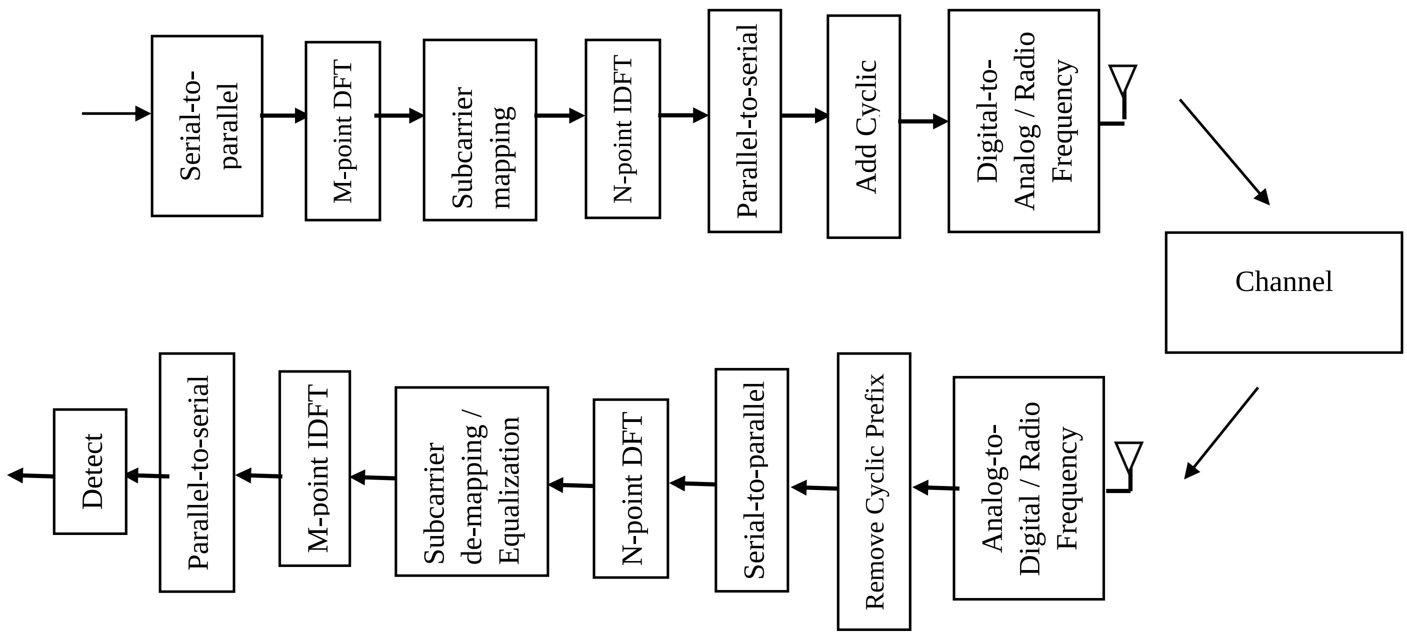

Fig 4.4: Transmitter and receiver structures of the SC-FDMA system over a frequency-selective channel.

then, the SC-FDMA system can handle up to Q orthogonal source signals with each source occupying a different set of M orthogonal subcarriers. In the notation of Figure 4.3, X_m (m=0, 1,...,M−1) represents modulated source symbols and X_k (k=0, 1,..., M−1) represents M samples of the DFT of X_m. Y_l (l=0, 1,..., N−1) represents the frequency domain samples after subcarrier mapping and Y_n (n=0, 1,..., N−1) represents the transmitted time domain channel symbols obtained from the inverse DFT (IDFT) of Y_l.The subcarrier mapping block in Figures 4.4 assigns frequency domain modulation symbols to subcarriers.

The mapping process is sometimes referred to as scheduling. Because spatially dispersed terminals have independently fading channels, SC-FDMA and OFDMA can benefit from channel dependent scheduling. The inverse transform (IDFT) creates a time domain representation, Y_n, of the N subcarrier symbols. The parallel-to-serial converter places Y_0, Y_1,..., Y_{N-1} in a time sequence suitable for modulating a radio frequency carrier and transmission to the receiver.

4.5.1 The DFT Based SC-FDMA Architecture

Figure 4.5 shows the architecture of DFT Based SC-FDMA

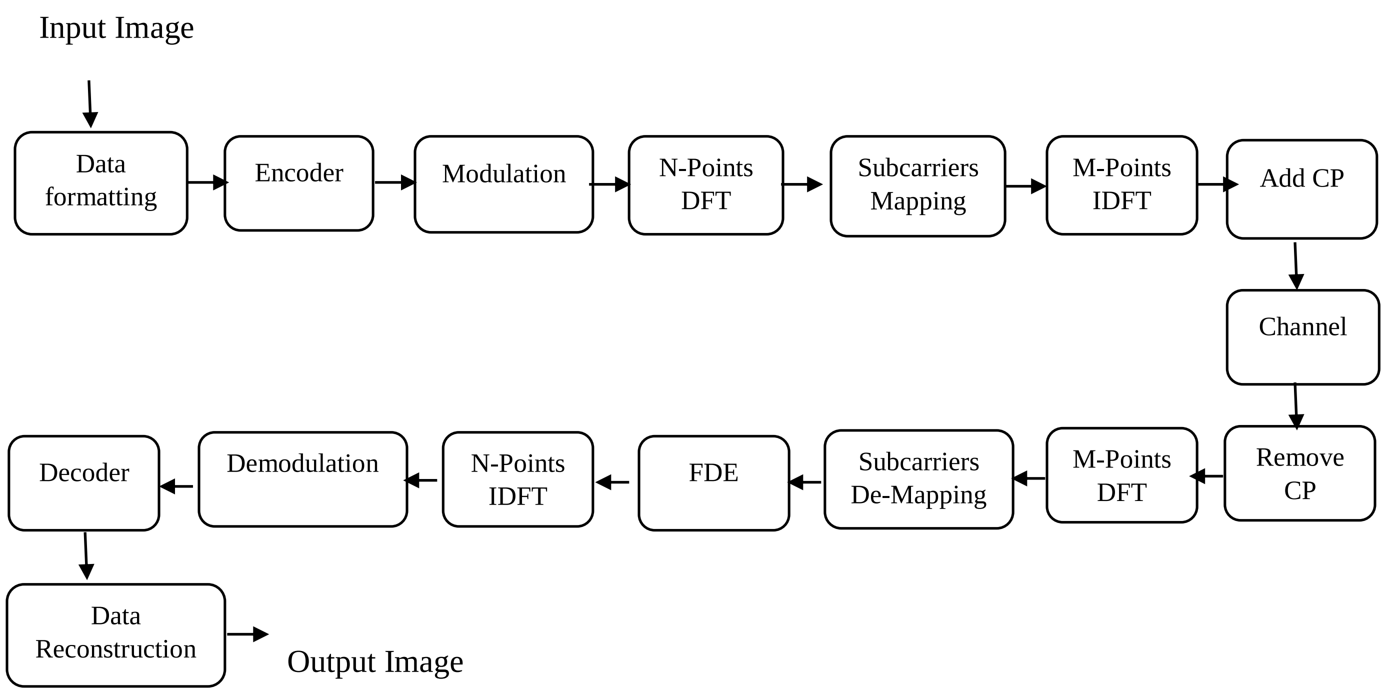

Fig 4.5: DFT based SC-FDMA architecture.

The image formatting is used to transmit the image over the SC-FDMA system by converting it to a binary form suitable to be inserted and processed by the SC-FDMA System. The SC-FDMA transmitter starts with an encoder then a modulation of the input signal using binary Quadrature Phase Shift Keying (QPSK), Let X_n represent the modulated source symbols. Then, the signal is transformed into frequency domain to produce frequency-domain symbols X_k .The signal after the DFT can be expressed as follows:

$$X_k = \sum_{n=0}^{N-1} X_n e^{\frac{-j2\pi}{N}nk}$$

where N is the input block size. {x :n= 0, ...,N-1} represents the modulated data symbols. The outputs are then mapped to M (M>N) orthogonal subcarriers followed by the M-points IDFT to convert to a time-domain complex signal sequence. M= QN is the output block size. Q is the maximum number of users that can transmit. The subcarriers mapping assigns frequency-domain modulation symbols to subcarriers Y_l, which represent the frequency-domain sample after subcarriers mapping. The inverse transform creates a time-domain symbol Y_m.The resulting signal after the IDFT is given by:

$$Y_m = \frac{1}{M}\sum_{l=0}^{M-1} Y_l e^{\frac{j2\pi}{M}nl}$$

Where {Y: l = 0... M-1} represents the frequency-domain samples after the subcarriers mapping scheme. It also inserts a set of symbols referred to as Cyclic Prefix (CP) in order to provide a guard time to prevent Inter Block Interference (IBI) due to multipath propagation. The CP is a copy of the last part of the block. It is inserted at the start of each block. The transmitted data propagates through the channel. At the receiver, the CP is removed, and then the received signal is transformed into the frequency domain in order to recover subcarriers. The de-mapping operation isolates the frequency-domain samples of each source signal. Because SC-FDMA uses single carrier modulation, the samples are passed through the Frequency Domain Equalizer (FDE). After that, the inverse transform at the receiver transforms equalized symbols back to the time domain. The demodulation process recovers the original data, which is passed through the decoder. The image reconstruction is used to convert the binary form to an image to recover the original image.

4.5.2 The DCT Based SC-FDMA Architecture

Fig. 4.6 shows the DCT based SC-FDMA architecture. One disadvantage of the DFT for some applications is that the transform is complex valued, even for real data. On the other hand, the DCT does not have this problem. In particular, a DCT is a Fourier-related transform similar to the DFT, but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry, where in some variants, the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

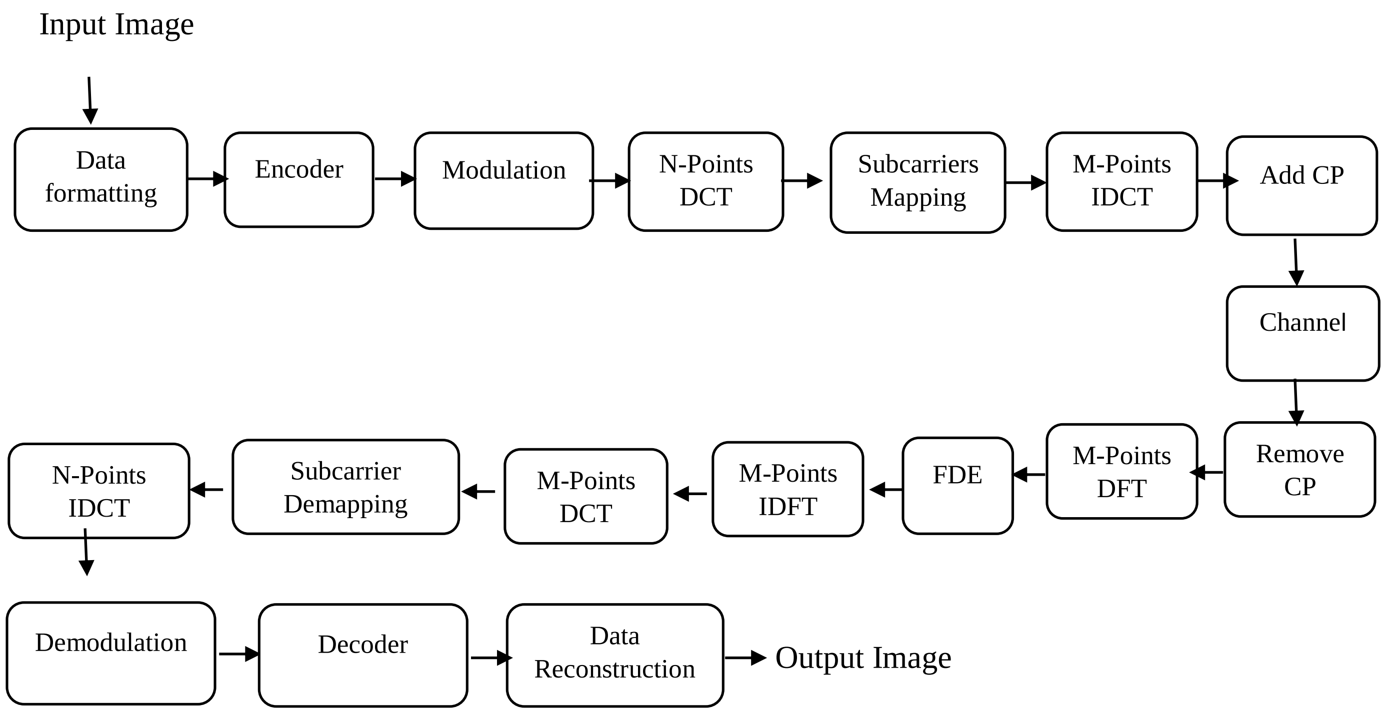

Fig 4.6: DCT based SC-FDMA architecture.

The signal after the DCT can be expressed as follows:

$$X_k = \sqrt{\frac{2}{N}}\beta_k \sum_{n=0}^{N-1} X_n \cos(\frac{\pi K(2n+1)}{2N})$$

Where X_n is the modulated data symbol, and β_k is given by:

$$\beta_k = \begin{Bmatrix} \frac{1}{\sqrt{2}} & K = 0 \\ 1 & K = 1,2,\dots\dots,N-1 \end{Bmatrix}$$

After the Inverse Discrete Cosine Transform (IDCT), the signal can be expressed as follows:

$$Y_m = \sqrt{\frac{2}{M}} \sum_{l=0}^{M-1} Y_l \beta_l \cos(\frac{\pi l(2m+1)}{2M})$$

Where Y_l is the signal after the subcarriers mapping. M is the IDCT length (number of subcarriers) (M=Q.N). Q is the bandwidth expansion factor of the symbol sequence. If all terminals transmit N symbols per block, the system can handle Q simultaneous transmissions without co-channel interference.

4.6 SC-FDMA System with Chaotic Interleaving

The signal samples can be arranged into a 2-D format then randomized using the chaotic Baker map. The chaotic interleaver generates permuted sequences with lower correlation between their samples and adds a degree of encryption to the transmitted signal. The discretized Baker map is an efficient tool to randomize the items in a square matrix. Let B ($n_1 \ldots n_K$), denote the discretized map, where the vector, [$n_1 \ldots n_K$], represents the secret key,S_{Key}. Defining N as the number of data items in one row, the secret key is chosen such that each integer n_i divides N, and $n_1 + \cdots + n_K$= N. Let $N_i = n_i + \cdots + n_{i-1}$. The data item at the indices (q, z), is moved to the indices:

$$B_{(n_1 \ldots n_k)}(q, z) = \left(\frac{N}{n_i}\ (q - N_i) + z \bmod \left(\frac{N}{n_i}\right), \frac{n_i}{N}\left(z - z \bmod \left(\frac{N}{n_i}\right)\right) + N_i\right)$$

Where $n_i \leq q \leq N_i + n_i$, $0 \leq z \leq N$

In steps, the chaotic permutation is performed as follows:

(1) An N × N square matrix is divided into N rectangles of width n_i and number of elements N.
(2) The elements in each rectangle are rearranged to a row in the permuted rectangle. Rectangles are taken from left to right beginning with upper rectangles then lower ones.
(3) Inside each rectangle, the scan begins from the bottom left corner towards upper elements.

$n_1 = 2$ $n_2 = 4$ $n_3 = 2$

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
S_{17}	S_{18}	S_{19}	S_{20}	S_{21}	S_{22}	S_{23}	S_{24}
S_{25}	S_{26}	S_{27}	S_{28}	S_{29}	S_{30}	S_{31}	S_{32}
S_{33}	S_{34}	S_{35}	S_{36}	S_{37}	S_{38}	S_{39}	S_{40}
S_{41}	S_{42}	S_{43}	S_{44}	S_{45}	S_{46}	S_{47}	S_{48}
S_{49}	S_{50}	S_{51}	S_{52}	S_{53}	S_{54}	S_{55}	S_{56}
S_{57}	S_{58}	S_{59}	S_{60}	S_{61}	S_{62}	S_{63}	S_{64}

(a) The 8x8 matrix divided into square rectangles

1-D error burst

S_{31}	S_{23}	S_{15}	S_7	S_{32}	S_{24}	S_{16}	S_8
S_{63}	S_{55}	S_{47}	S_{39}	S_{64}	S_{56}	S_{48}	S_{40}
S_{11}	S_3	S_{12}	S_4	S_{13}	S_5	S_{14}	S_6
S_{27}	S_{19}	S_{28}	S_{28}	S_{29}	S_{21}	S_{30}	S_{22}
S_{43}	S_{35}	S_{44}	S_{36}	S_{45}	S_{37}	S_{46}	S_{38}
S_{39}	S_{51}	S_{60}	S_{52}	S_{61}	S_{53}	S_{62}	S_{54}
S_{25}	S_{17}	S_9	S_1	S_{26}	S_{18}	S_{10}	S_2
S_{57}	S_{49}	S_{41}	S_{33}	S_{58}	S_{50}	S_{42}	S_{34}

2-D error burst

(b) Chaotic interleaving of the 8x8 matrix

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
S_{17}	S_{18}	S_{19}	S_{20}	S_{21}	S_{22}	S_{23}	S_{24}
S_{25}	S_{26}	S_{27}	S_{28}	S_{29}	S_{30}	S_{31}	S_{32}
S_{33}	S_{34}	S_{35}	S_{36}	S_{37}	S_{38}	S_{39}	S_{40}
S_{41}	S_{42}	S_{43}	S_{44}	S_{45}	S_{46}	S_{47}	S_{48}
S_{49}	S_{50}	S_{51}	S_{52}	S_{53}	S_{54}	S_{55}	S_{56}
S_{57}	S_{58}	S_{59}	S_{60}	S_{61}	S_{62}	S_{63}	S_{64}

(c) Effect of error-bursts after the de-interleaving

Fig 4.7 Chaotic Interleaving Process

Fig. 4.7 shows an example for the chaotic interleaving of an (8 × 8) square matrix (i.e. N = 8). The secret key, S_{Key} = [n_1,n_2,n_3] = [2, 4, 2]. Assume that an error burst affects four consecutive samples (1-D error burst) as shown in Fig. 4b with shades. After de-interleaving, the error burst is effectively spread among four different rows as shown in Fig. 4c. When a 2-D (2×2) error burst occurs [17], as shown in Fig. 4b with shades. Figure 4c indicates that this 2×2 error burst is spread, so that there are not adjacent samples in error. As a result, this error burst can be corrected. Fig. 5 and Fig. 6 show the modified SC-FDMA system with chaotic interleaving.

4.7 Cyclic Prefix Extension Channel

The transmitter performs two other signal processing operations prior to transmission. It inserts a set of symbols referred to as cyclic prefix (CP) in order to provide a guard time to prevent inter-block interference (IBI) due to multipath propagation. The transmitter also performs a linear filtering operation referred to as pulse shaping in order to reduce out-of-band signal energy. The cyclic prefix is a copy of the last part of the block. It is inserted at the start of each block for two reasons. First, the CP acts as a guard time between successive blocks. If the length of the CP is longer than the maximum delay spread of the channel, or roughly, the length of the channel impulse response, then, there is no IBI. Second, since the CP is a copy of the last part of the block, it converts a discrete time linear convolution into a discrete time circular convolution. Thus, transmitted data propagating through the channel can be modeled as a circular convolution between the channel impulse response and the transmitted data block, which in the frequency domain is a point-wise multiplication of the DFT frequency samples. Then, to remove the channel distortion, the DFT of the received signal can simply be divided by the DFT of the channel impulse response point-wise. One commonly used pulse-shaping filter is a raised-cosine filter. The frequency domain and time domain representations of the filter are as follows:

4.8 Subcarrier Mapping

DFT output of the data symbols is mapped to a subset of subcarriers, a process called subcarrier mapping. The subcarrier mapping assigns DFT output complex values as the amplitudes of some of the selected subcarriers. Subcarrier mapping can be classified into two types: *localized mapping* and *distributed mapping*. In *localized mapping*, the DFT outputs are mapped to a subset of consecutive sub-carriers thereby confining them to only a fraction of the system bandwidth. In *distributed mapping*, the DFT outputs of the input data are assigned to subcarriers over the entire bandwidth non-continuously, resulting in zero amplitude for the remaining subcarriers. A special case of *distributed SC-FDMA* is called *interleaved SC-FDMA*, where the occupied subcarriers are equally spaced over the entire bandwidth. Figure 4.8 is a general picture of *localized* and *distributed* mapping.

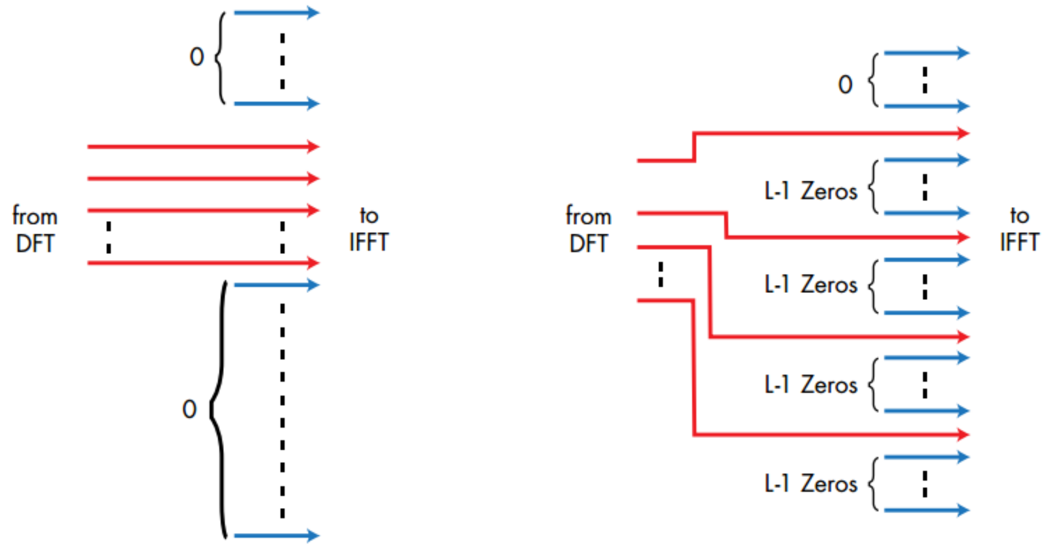

Fig 4.8: Localized mapping vs. Distributed mapping

An example of subcarrier mapping is shown in Figure 4.9. This example assumes three users sharing 12 subcarriers. Each user has a block of four data symbols to transmit at a time. The DFT output of the data block has four complex frequency domain samples, which are mapped over 12 subcarriers using different mapping schemes.

SC-FDMA inherently offers frequency diversity gain over the standard OFDM, as all information data is spread over multiple subcarriers by the DFT mapper. However, the distributed SC-FDMA is more robust with respect to frequency selective fading and offers additional frequency diversity gain, since the information is spread across the entire system bandwidth. Localized SC-FDMA in combination with channel-dependent scheduling can potentially offer multi-user diversity in frequency selective channel conditions.

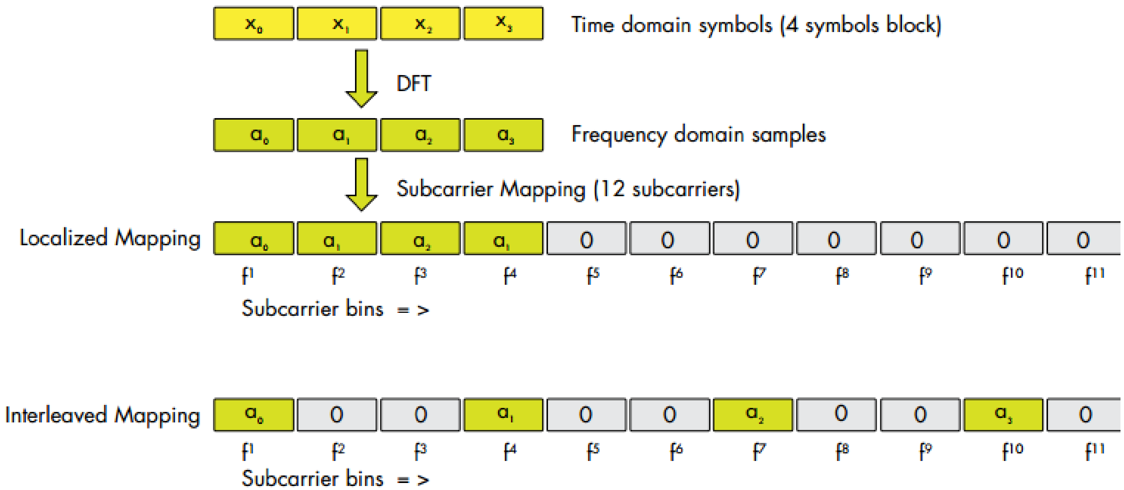

Fig 4.9: Subcarrier mapping example

4.9 Equalizer Design

The FDE cancels the ISI. The equalizer coefficients $W(m)$ are computed according to the type of the FDE as follows:

•The zero forcing (ZF) equalizer:

$$W(m) = \frac{1}{H(m)} \qquad\qquad m = 0,1, \dots\dots\dots\dots\dots.. N_{DFT} - 1$$

•The minimum mean square error (MMSE) equalizer:

$$W(m) = \frac{H^*}{|H(m)|^2 + (\frac{E_b}{N_0})^{-1}}$$

Where, $H(m)$ is the channel transfer function and H*(m) denotes the complex conjugate and $(\frac{E_b}{N_0})$ is the SNR. The ZF FDE perfectly reverses the effect of the channel in the absence of noise. In the presence of noise, it suffers from the noise enhancement phenomenon. On the other hand, the MMSE equalizer takes into account the SNR, making an optimum trade-off between channel inversion and noise enhancement.

4.10 PSNR

The PSNR is used to measure the quality of the reconstructed images at the receiver. It is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of this signal. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. The PSNR is defined as follows:

$$PSNR = 10\ log_{10}\left(\frac{max_f^2}{MSE}\right)$$

Where max_f is the maximum possible pixel value of an image f. For 8 bit pixels, e $max_f = 255$. MSE is the Mean Square Error. For an N ×N monochrome image, it is defined as:

$$MSE = \frac{\sum\left[f(i,j) - \hat{f}(i,j)\right]^2}{N^2}$$

Where $f(i,j)$ the source is image, and $\hat{f}(i,j)$ is the reconstructed image.

CHAPTER 5

SIMULATION & RESULTS

5.1 Introduction

For high-efficiency image transmission, this chapter presents a comparison between two different systems—discrete Fourier transform (DFT)-based SC-FDMA and discrete cosine transform (DCT)-based SC-FDMA—in order to select the proper transmission technique for image transmission

The obtained results show that the DFT-based SC-FDMA system achieves higher peak signal-to-noise ratio (PSNR) values than the DCT-based SC-FDMA system. The DCT has been implemented for both OFDM and SC-FDMA but the image communication problem has not been studied. In this chapter, we develop this approach to work with the SC-FDMA system and investigate its effect on image communication.

We utilized chaotic interleaving to enhance the performance of the continuous phase modulation (CPM)-OFDM system for data transmission. Error bursts are better distributed to samples after de-interleaving in the proposed chaotic interleaving scheme than block interleaving mechanism. As a result, a better PSNR performance can be achieved when applying this scheme to the SC-FDMA system. Moreover, it adds a degree of security to the communication system.

Experimental results show that the DFT-based SC-FDMA structure achieves higher PSNR values than the DCT-based SC-FDMA structure. Furthermore, the results show that the PSNR values are enhanced by applying chaotic interleaving scheme in both structures

5.2 Simulation Results

To evaluate performance and efficiency, QPSK based DFT SCFDMA and DCT SCFDMA system used with chaotic interleaving and without chaotic interleaving. SUI3 channel used for image transmission with AWGN noise environment.

To evaluate the performance and efficiency of the above-mentioned systems, the 256×256 Brain image is used. The simulation parameters are given in Table (1) and the simulation results are shown in Figure Shows a comparison between both schemes.

Simulation Parameter	Value
FFT size	512 symbols
Input block size	128 symbols
Cyclic prefix size	20 samples
Image size	256 × 256 for Brian
Channel Coding	Convolutional code with rate 1/2
Modulation type	QPSK
Subcarriers mapping	Interleaved and localized
Channel model	SUI-3 channel
Noise environment	AWGN
Equalizer type	MMSE equalizer

Table 5.1: Simulation Parameter

Figs. (5.1) Show the simulation results when transmitting the original image through the DFT-SCFDMA and DCT-SC-FDMA systems for SNR values without chaotic interleaving (without randomization). The results show that the DFT-SC-FDMA system achieves higher PSNR values than the DCT SC-FDMA system due to the energy compaction property of the DFT based SC-FDMA system, which makes most of the samples transmitted close to zero leading to a reduction in the effect ISI.

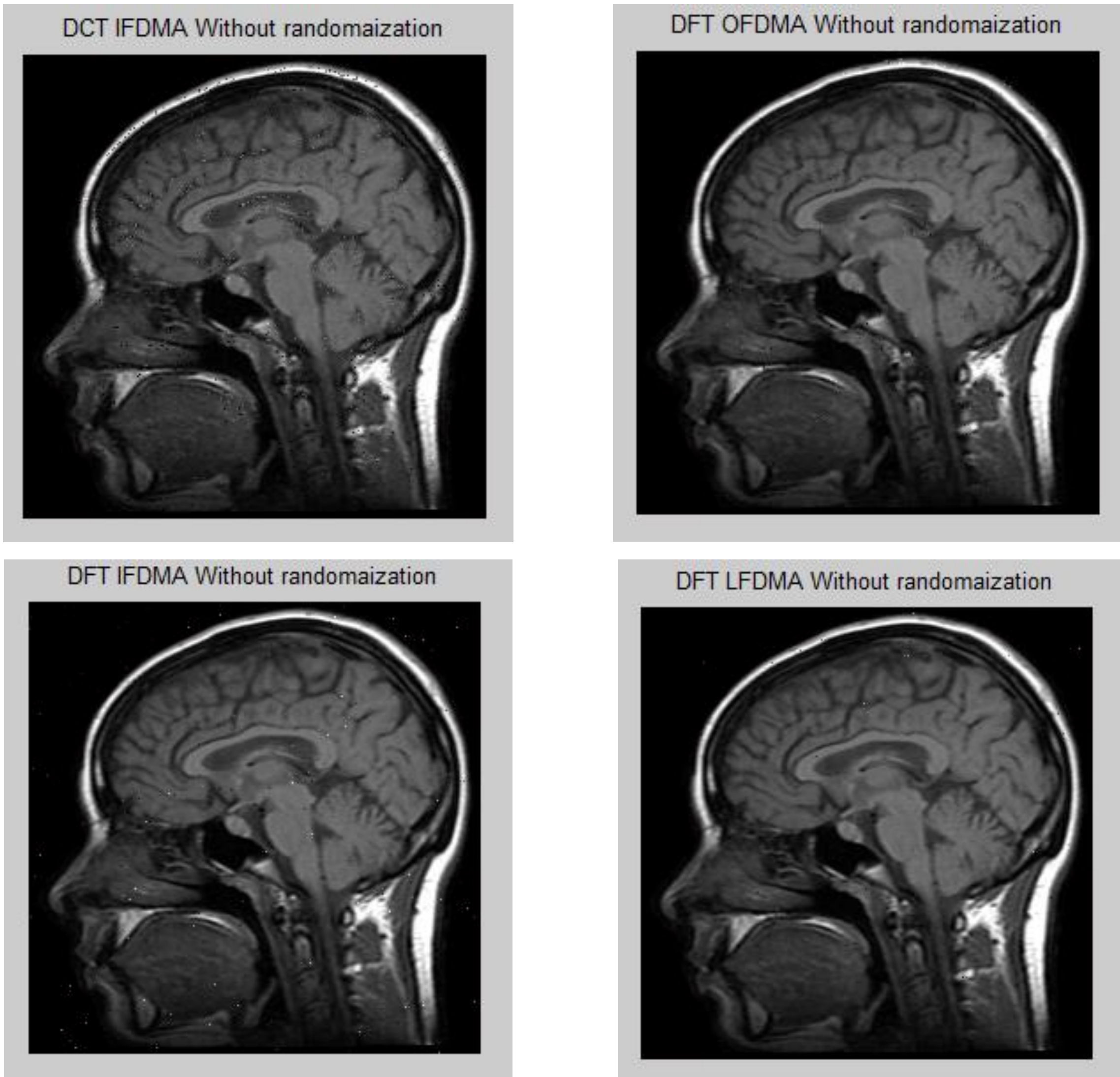

Fig 5.1 Simulation results for SNR =10 dB using without randomization. (a) DCT-SC-FDMA (IFDMA), (b) DCT-SC-FDMA(LFDMA),(c) DFT-SC-FDMA (IFDMA) (d) DFTSC-FDMA(IFDMA)

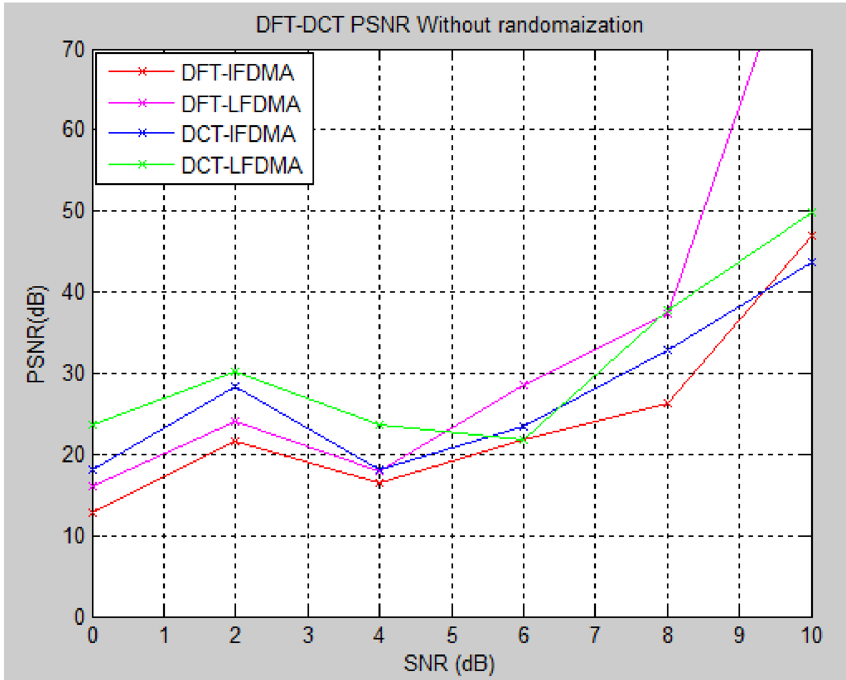

Fig 5.2: shows the PSNR values for DFT-based SC-FDMA and DCT-based SC-FDMA systems without randomization. It is clear that the DFT-based SC-FDMA system outperforms the DCT-based SC-FDMA system, especially, when using LFDMA

Figures (5.3) show the simulation results when transmitting the original image through DFT SC-FDMA and DCT-SC-FDMA systems for SNR = 10 dB values with chaotic interleaving. The results show that the DFT-SC-FDMA system with chaotic interleaving has higher PSNR values than DFT-SC-FDMA system without interleaving, where the chaotic interleaving scheme has a better treatment to error bursts. Errors are better distributed to samples after de-interleaving in the proposed chaotic interleaving scheme. As a result, a better PSNR performance can be achieved when applying this scheme. Moreover, it adds a degree of security to the communication system.

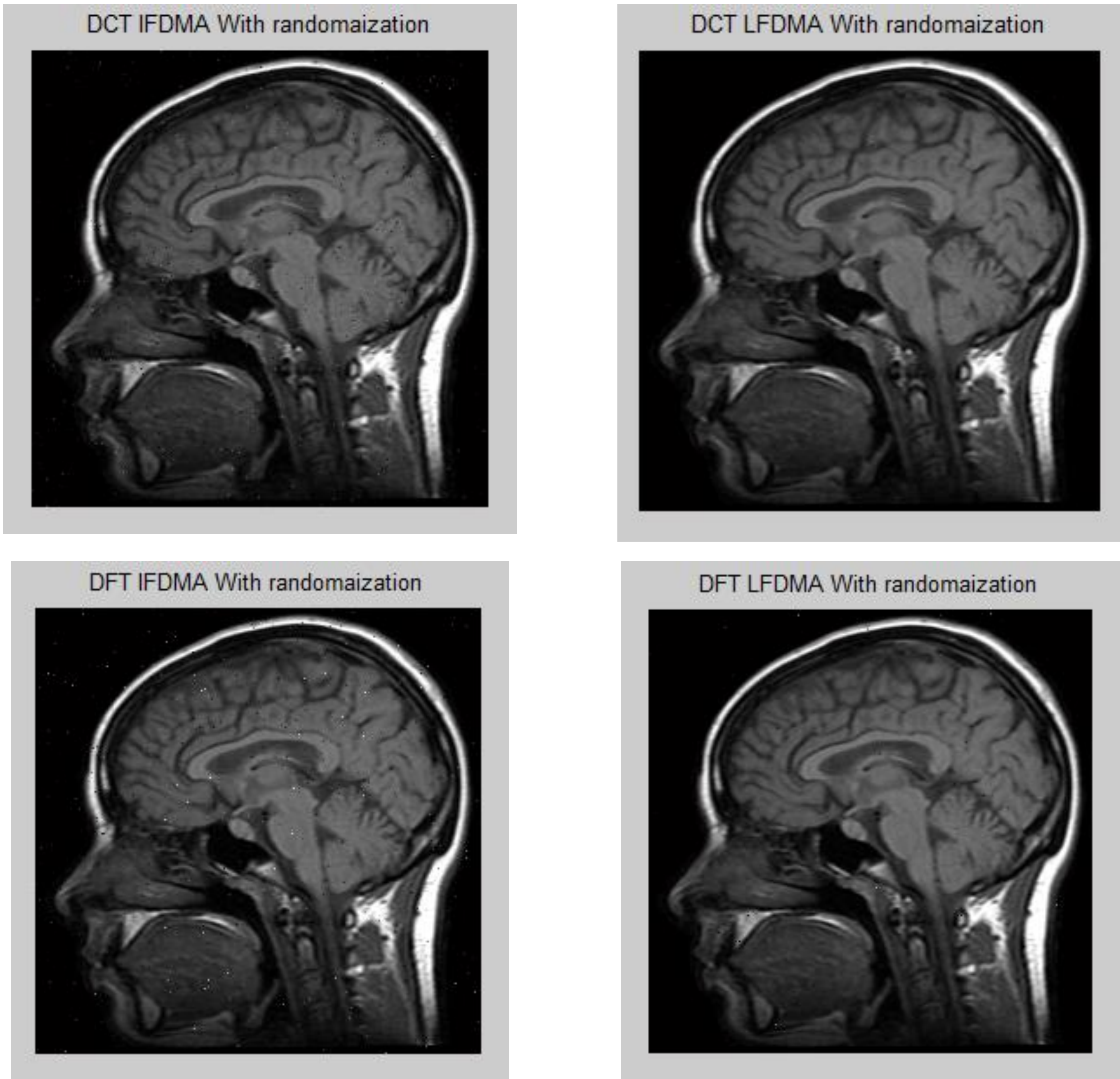

Fig. 5.3 Simulation results for SNR=10 dB with randomization. (a) DCT-SC-FDMA (IFDMA), (b) DCT-SC-FDMA (LFDMA), (c) DFT-SC-FDMA(IFDMA), d) DFT-SC FDMA (LFDMA)

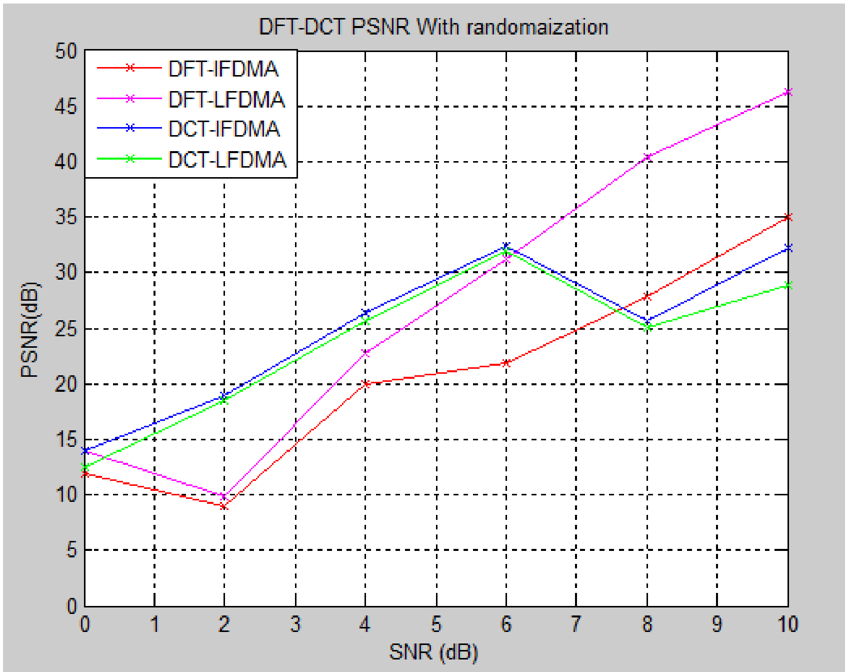

Fig. 5.4 shows the PSNR values for DFT-based SC-FDMA and DCT-based SC-FDMA systems with randomization by chaotic interleaving. It is clear that the DFT-based SC-FDMA system outperforms the DCT-based SC-FDMA system.

5.3 Conclusions

This paper presented an efficient image transmission scheme over the SC-FDMA system. Two different systems have been used; DFT-SC-FDMA and DCT-SC-FDMA. Also the chaotic interleaving scheme was applied to both systems for efficient image transmission. Simulation of both systems using Matlab program was done to evaluate the performance and efficiency of both systems. The experimental results have shown that the DFT based SC-FDMA system achieve higher PSNR values than DCT-based SC-FDMA system, which makes most of the samples transmitted close to zero leading to a reduction in the effect of ISI. In addition, it uses only real arithmetics rather than the complex arithmetic's used in the DFT. This reduces the signal processing complexity. The results also show a noticeable improvement in PSNR values with the chaotic interleaving scheme.

REFERENCE:

1. Jens Berkmann, et al., "On 3G LTE Terminal Implementation – Standard, Algorithms, Complexities and Challenges", IWCMC 2008 Mobile Computing Symposium, 2008

2. Hyung G. Myung, Junsung Lim, and David J. Goodman, "Single Carrier FDMA for Uplink Wireless Transmission." IEEE Vehicular Technology, Sept 2006

3. 3GPP TR 25.912 v 7.1.0, "Feasibility study for evolved Universal Terrestrial Radio Access (UTRA) and Universal Terrestrail Radio Access Network (UTRAN)," Release 7

4. 3GPP TS 36.300 v8.7.0, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2," Release 8

5. 3GPP TS 36.211 v8.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation," Release 8

6. 3GPP TS 36.212 v8.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel coding," Release 8

7. 3GPP TS 36.213 v8.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," Release 8

8. 3GPP, "3rd Generation Partnership Project, Technical specification group radio access network", Feasibility study for further advancements for E-UTRA (LTE-Advanced) (Release 9), 3GPP TR 36.912.

9. 3GPP, "3rd Generation Partnership Project, Technical specification group radio access network", Evolved universal terrestrial radio assess (E-UTRA); Base Station (BS) Radio Transmission and Reception, 3GPP TS 36.104.

10. Wikipedia (http://en.wikipedia.org/wiki/Discrete_cosine_transform).

11. J. Fridrich, (1998). Symmetric ciphers based on two-dimensional chaotic maps. International Journal of Bifurcation and Chaos, 8, 1259–1284.

12. Shi, Y. Q., Zhang, X. M., Ni, Z.-C., & Ansari, N. (2004). Interleaving for combating error bursts. IEEE Circuts and systems magazine, 4, 29–42 (First Quarter 2004).

13. Safey. A. S. Abdelwahab, I. I. Mahmoud and H. M. Frahat, “Image Compression Using a Fast 2-D DCT Algorithm”, International Conference on Industrial Electronics, Technology & Automation, Cairo, Egypt, 2001

APPENDIX

****

RunSimS_DFT_and_DCT_SC_FDMA.m

****

```
function runSimS_DFT_and_DCT_SC_FDMA()
clear all
tic;
%======= Choose simulation Parameters
SP.inputBlockSize = 128;
SP.FFTsize = 512;% Set the values of the FFT and IFFT
SP.CPsize = 20;%Set the Cyclic prefix length
SP.subband = 0;
SP.SNR =0:2:10;%Choose the range of the SNR in dB
%%%% Choose the coding rate and the modulation Type%%%
SP.cod_rate = '1/2';
SP.modtype = 'QPSK';
%%%%%%%%%% Choose the Equalization Type%%
SP.equalizerType = 'MMSE';

image = imread('mri.gif');
figure(1)
imshow(image);
title('Input Image');

% %%%%%%%%%% Run the simulation for With Randomization PSNR_DFT_SCFDMA %%
[PSNR_DFT_ifdma PSNR_DFT_lfdma y1_ifdma_DFT y1_lfdma_DFT] = DFT_SCFDMA(SP);
%%%%%%%%% Run the simulation for DCT_SCFDMA %%
[PSNR_DCT_ifdma PSNR_DCT_lfdma y1_ifdma_DCT y1_lfdma_DCT] =
DCT_SCFDMA(SP);
save PSNR_DCT_ifdma;
save PSNR_DCT_lfdma;
save PSNR_DFT_ifdma;
save PSNR_DFT_lfdma;
save y1_ifdma_DCT;
save y1_lfdma_DCT;
save y1_ifdma_DFT;
save y1_lfdma_DFT;
%%%%%%%%%% Run the simulation for Without Randomization PSNR_DFT_SCFDMA %%
[PSNR_DFT_ifdma1 PSNR_DFT_lfdma1 y1_ifdma_DFT1 y1_lfdma_DFT1] =
DFT_SCFDMA_wo(SP);
%%%%%%%%%% Run the simulation for DCT_SCFDMA %%
```

```
[PSNR_DCT_ifdma1 PSNR_DCT_lfdma1 y1_ifdma_DCT1 y1_lfdma_DCT1] =
DCT_SCFDMA_wo(SP);

save PSNR_DCT_ifdma1;
save PSNR_DCT_lfdma1;
save PSNR_DFT_ifdma1;
save PSNR_DFT_lfdma1;
save y1_ifdma_DCT1;
save y1_lfdma_DCT1;
save y1_ifdma_DFT1;
save y1_lfdma_DFT1;


%%%%%%%%% Plot the Results %%
figure(11)
plot(SP.SNR,PSNR_DFT_ifdma,'rx-',SP.SNR,PSNR_DFT_lfdma,'mx-');
hold on
plot(SP.SNR,PSNR_DCT_ifdma,'bx-',SP.SNR,PSNR_DCT_lfdma,'gx-');
% hold on
% plot(SP.SNR,PSNR_ofdma,'yx-');
legend('DFT-IFDMA','DFT-LFDMA','DCT-IFDMA','DCT-LFDMA')
xlabel('SNR (dB)'); ylabel('PSNR(dB)');
axis([0 20 0 70])
title('DFT-DCT PSNR With randomaization');
grid on

%without randomization
figure(12)
plot(SP.SNR,PSNR_DFT_ifdma1,'rx-',SP.SNR,PSNR_DFT_lfdma1,'mx-');
hold on
plot(SP.SNR,PSNR_DCT_ifdma1,'bx-',SP.SNR,PSNR_DCT_lfdma1,'gx-');
% hold on
% plot(SP.SNR,PSNR_ofdma1,'yx-');
legend('DFT-IFDMA','DFT-LFDMA','DCT-IFDMA','DCT-LFDMA')
xlabel('SNR (dB)'); ylabel('PSNR(dB)');
axis([0 20 0 70])
title('DFT-DCT PSNR With randomaization');
grid on
toc
```

****
DCT_SCFDMA.m
****

```
function [PSNR_DCT_ifdma PSNR_DCT_lfdma y1_ifdma_DCT y1_lfdma_DCT] =
DCT_SCFDMA(SP)

for sss=1: length(SP.SNR)
%======= Choose channel type

SP.paths= fadchan(SP);
SP.equalizerType ='MMSE';
numSymbols = SP.FFTsize;
Q = numSymbols/SP.inputBlockSize;
%%%%%%%%========== Channel Generation============
%%%%%%%%============ SUI3 channel============
SUI3_1=[SP.paths(1,1) 0 0 SP.paths(2,1) 0 SP.paths(3,1)];
SP.channel = SUI3_1/norm(SUI3_1);
H_channel = fft(SP.channel,SP.FFTsize);

im1=imread('mri.gif');%image reading
im=imresize(im1,[256 256]);
xx=randomization(im);%image randomization
%*****************Data Generation*******************
f = zeros(256,256);
f = double(xx);
[M,N]=size(f);
g=im2col(f, [M,N],'distinct');%image to column converter
h=dec2bin(double(g));%pixel value to binary conversion...every value replaced by 8 bits string
[M1,N1] = size(h);
z=zeros (M1,N1);
clear i j
for i=1:M1
   for j=1:N1
       z(i,j)= str2num(h(i,j)); %string to number conversion
   end;
end;
[M2,N2] = size(z) ;
zz = reshape(z,M2*N2, 1);%parallel data reshaping date to vector
% ********** Dividing the image into blocks***********
nloops = ceil((M2*N2)/SP.inputBlockSize );%number of image %blocks by approximation
new_data = nloops*SP.inputBlockSize ;%new vector proportional to %block size
nzeros = new_data  - (M2*N2);%number of zeros to be added to %old data vector
input_data = [zz;zeros(nzeros,1)];%construction of new data %vector
input_data2 = reshape(input_data ,SP.inputBlockSize ,nloops); %reshape the new data to matrix
of block size rows to number of %blocks columns
```

```
save input_data2
%************** transmission ON SC-FDMA ***************
demodata1_ifdma = zeros(SP.inputBlockSize ,nloops);% this matrix to %store received data vector
demodata1_lfdma = zeros(SP.inputBlockSize ,nloops);
clear jj
for jj =1: nloops % loop for columns
   b1= input_data2(:,jj)';%every block size
   %%%%%%%%%%%%%%%% QPSK Modulation %%%%%%%%%%%%%%%%
   tmp = b1;
   tmp = tmp*2 - 1;
   inputSymbols = (tmp(1,:) + i*tmp(1,:))/sqrt(2);
   %%%%%%%%%%%%% SC-FDMA  Modulation %%%%%%%%%%%%%
   inputSymbols_freq = dct(inputSymbols);
   inputSamples_ifdma = zeros(1,numSymbols);
   inputSamples_lfdma = zeros(1,numSymbols);
   %%%%%%%%%%%%% Subcarriers Mapping %%%%%%%%%%%%%
   inputSamples_ifdma(1+SP.subband:Q:numSymbols) = inputSymbols_freq;
   inputSamples_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband) =
inputSymbols_freq;
   inputSamples_ifdma = idct(inputSamples_ifdma);
   inputSamples_lfdma = idct(inputSamples_lfdma);
   %%%%%%%%%%%%%% Add Cyclic Prefix %%%%%%%%%%%%%
   TxSamples_ifdma = [inputSamples_ifdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_ifdma];
   TxSamples_lfdma = [inputSamples_lfdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_lfdma];
   %%%%%%%%%%%%%% Wireless channel %%%%%%%%%%%%%%%
   RxSamples_ifdma = filter(SP.channel, 1,TxSamples_ifdma); % Multipath Channel
   RxSamples_lfdma = filter(SP.channel, 1,TxSamples_lfdma); % Multipath Channel
   %%%%%%%%%%%%%% Noise Generation %%%%%%%%%%%%%
   tmp = randn(2, numSymbols+SP.CPsize);
   complexNoise = (tmp(1,:) + i*tmp(2,:))/sqrt(2);
   noisePower = 10^(-SP.SNR(sss)/10);
   %%%%%%%%%%%%%%%% Received signal%%%%%%%%%%%%%%
   RxSamples_ifdma = RxSamples_ifdma + sqrt(noisePower/Q)*complexNoise;
   RxSamples_lfdma = RxSamples_lfdma + sqrt(noisePower/Q)*complexNoise;
   %%%%%%%%%%%%% Remove Cyclic Prefix%%%%%%%%%%%%
   RxSamples_ifdma = RxSamples_ifdma(SP.CPsize+1:numSymbols+SP.CPsize);
   RxSamples_lfdma = RxSamples_lfdma(SP.CPsize+1:numSymbols+SP.CPsize);
   %%%%%%%%%%%%%% SC-FDMA demodulation%%%%%%%%%%%
   Y_ifdma = dct(RxSamples_ifdma, SP.FFTsize);
   Y_lfdma = dct(RxSamples_lfdma, SP.FFTsize);
   %%%%%%%%%%%%% subcarriers demapping%%%%%%%%%%%
   Y_ifdma = Y_ifdma(1+SP.subband:Q:numSymbols);
   Y_lfdma = Y_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
```

```
  %%%%%%%%%%%%%%%%% Equalization %%%%%%%%%%%%%%%%%
  H_eff = H_channel(1+SP.subband:Q:numSymbols);
  if SP.equalizerType == 'ZERO'
     Y_ifdma = Y_ifdma./H_eff;
  elseif SP.equalizerType == 'MMSE'
     C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
     Y_ifdma = Y_ifdma.*C;
  end
  H_eff = H_channel([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
  if SP.equalizerType == 'ZERO'
     Y_lfdma = Y_lfdma./H_eff;
  elseif SP.equalizerType == 'MMSE'
     C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
     Y_lfdma = Y_lfdma.*C;
  end
  EstSymbols_ifdma = idct(Y_ifdma);
  EstSymbols_lfdma = idct(Y_lfdma);
  %%%%%%%%%%%%%%%%% demodulation%%%%%%%%%%%%%%%%
  EstSymbols_ifdma = sign(real(EstSymbols_ifdma)) ;
  EstSymbols_ifdma =(EstSymbols_ifdma+1)/2;
  EstSymbols_lfdma = sign(real(EstSymbols_lfdma)) ;
  EstSymbols_lfdma = (EstSymbols_lfdma+1)/2;
  demodata1_ifdma(:,jj) =  EstSymbols_ifdma(:);   % the output of scfdma columns%storing of
received image data
  demodata1_lfdma(:,jj)  = EstSymbols_lfdma(:);
  % the output of scfdma columns%storing of received image data
end
%*****************  Received image  ***************
[M3,N3] = size(demodata1_ifdma);
yy1_ifdma = reshape (demodata1_ifdma,M3,N3);
%reshaping the matrix to vector
yy1_lfdma = reshape (demodata1_lfdma,M3,N3);
%reshaping the matrix to vector
received_image_ifdma = yy1_ifdma(1:M2*N2);%taking the original data
received_image_lfdma = yy1_lfdma(1:M2*N2);%taking the original data
%**************  Regeneration of image  **************
zz1_ifdma=reshape(received_image_ifdma,M2* N2,1);
%reshaping to M2*N2 vector
zz1_lfdma=reshape(received_image_lfdma,M2* N2,1);
%reshaping to M2*N2 vector
yy_ifdma = reshape(zz1_ifdma,M2, N2);
yy_lfdma = reshape( zz1_lfdma,M2, N2);
clear i j
for i=1:M1
  for j=1:N1
     zn_ifdma(i,j)=num2str(yy_ifdma(i,j));
```

```
        zn_lfdma(i,j)=num2str(yy_lfdma(i,j));
    end;
end;
hn_ifdma=bin2dec(zn_ifdma);
hn_lfdma=bin2dec(zn_lfdma);
gn_ifdma=col2im(hn_ifdma, [M,N], [M,N], 'distinct');
gn_lfdma=col2im(hn_lfdma, [M,N], [M,N], 'distinct');
y1_ifdma=derandomization(gn_ifdma);
y1_lfdma=derandomization(gn_lfdma);
y1_ifdma=y1_ifdma/255;
y1_lfdma=y1_lfdma/255;
% **************** The output results***************
figure (4)
imshow(y1_ifdma)
title('DCT IFDMA With randomaization');
MSE1_ifdma=sum(sum((double(im)/255-y1_ifdma).^2))/prod(size(im));
PSNR_DCT_ifdma(sss)=10*log(1/MSE1_ifdma)/log(10);
figure (5)
imshow(y1_lfdma)
title('DCT LFDMA With randomaization');
MSE1_lfdma=sum(sum((double(im)/255-y1_lfdma).^2))/prod(size(im));
PSNR_DCT_lfdma(sss)=10*log(1/MSE1_lfdma)/log(10);
y1_ifdma_DCT(:,:,sss)=y1_ifdma;
y1_lfdma_DCT(:,:,sss)=y1_lfdma;
toc
end
```

****

DFT_SCFDMA.m

****

```
function [PSNR_DFT_ifdma PSNR_DFT_lfdma y1_ifdma_DFT y1_lfdma_DFT] =
DFT_SCFDMA(SP)

for sss=1: length(SP.SNR)
%======= Choose channel type
%======= Uniform Channel
%SP.channel=(1/sqrt(10))*(randn(10,SP.numRun)+sqrt(-1)*randn(10,SP.numRun))/sqrt(2);
%======= SUI3 Channel
SP.paths= fadchan(SP);

SP.equalizerType ='MMSE';

numSymbols = SP.FFTsize;
Q = numSymbols/SP.inputBlockSize;
%%%%%%%%%=========== SUI3 channel===========
```

```
SUI3_1=[SP.paths(1,1) 0 0 SP.paths(2,1) 0 SP.paths(3,1)];
SP.channel = SUI3_1/norm(SUI3_1);
H_channel = fft(SP.channel,SP.FFTsize);

im1=imread('mri.gif');%image reading
im=imresize(im1,[256 256]);
xx=randomization(im);%image randomization
%*****************Data Generation*******************
f = zeros(256,256);
f = double(xx);
[M,N]=size(f);
g=im2col(f, [M,N],'distinct');%image to column converter
h=dec2bin(double(g));%pixel value to binary conversion...every value replaced by 8 bits string
[M1,N1] = size(h);
z=zeros (M1,N1);
clear i j
for i=1:M1
   for j=1:N1
       z(i,j)= str2num(h(i,j)); %string to number conversion
   end;
end;
[M2,N2] = size(z) ;
zz = reshape(z,M2*N2, 1);%parallel data reshaping date to vector
% ********** Dividing the image into blocks***********
nloops = ceil((M2*N2)/SP.inputBlockSize );%number of image %blocks by approximation
new_data = nloops*SP.inputBlockSize ;%new vector proportional to %block size
nzeros = new_data  - (M2*N2);%number of zeros to be added to %old data vector
input_data = [zz;zeros(nzeros,1)];%construction of new data %vector
input_data2 = reshape(input_data ,SP.inputBlockSize ,nloops); %reshape the new data to matrix
of block size rows to number of %blocks columns
save input_data2
%************** transmission ON SC-FDMA ***************
demodata1_ifdma = zeros(SP.inputBlockSize ,nloops);% this matrix to %store received data
vector
demodata1_lfdma = zeros(SP.inputBlockSize ,nloops);
clear jj
for jj =1: nloops % loop for columns
   b1= input_data2(:,jj)';%every block size
   %%%%%%%%%%%%%%%%% QPSK Modulation %%%%%%%%%%%%%%%%
   tmp = b1;
   tmp = tmp*2 - 1;
   inputSymbols = (tmp(1,:) + i*tmp(1,:))/sqrt(2);
   %%%%%%%%%%%%% SC-FDMA  Modulation %%%%%%%%%%%%%
   inputSymbols_freq = fft(inputSymbols);
   inputSamples_ifdma = zeros(1,numSymbols);
   inputSamples_lfdma = zeros(1,numSymbols);
```

```
    %%%%%%%%%%%% Subcarriers Mapping %%%%%%%%%%%%
    inputSamples_ifdma(1+SP.subband:Q:numSymbols) = inputSymbols_freq;
    inputSamples_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband) =
inputSymbols_freq;
    inputSamples_ifdma = ifft(inputSamples_ifdma);
    inputSamples_lfdma = ifft(inputSamples_lfdma);
    %%%%%%%%%%%%% Add Cyclic Prefix %%%%%%%%%%%%%
    TxSamples_ifdma = [inputSamples_ifdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_ifdma];
    TxSamples_lfdma = [inputSamples_lfdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_lfdma];
    %%%%%%%%%%%%% Wireless channel %%%%%%%%%%%%%
    RxSamples_ifdma = filter(SP.channel, 1,TxSamples_ifdma); % Multipath Channel
    RxSamples_lfdma = filter(SP.channel, 1,TxSamples_lfdma); % Multipath Channel
    %%%%%%%%%%%%% Noise Generation %%%%%%%%%%%%
    tmp = randn(2, numSymbols+SP.CPsize);
    complexNoise = (tmp(1,:) + i*tmp(2,:))/sqrt(2);
    noisePower = 10^(-SP.SNR(sss)/10);
    %%%%%%%%%%%%%%% Received signal%%%%%%%%%%%%%
    RxSamples_ifdma = RxSamples_ifdma + sqrt(noisePower/Q)*complexNoise;
    RxSamples_lfdma = RxSamples_lfdma + sqrt(noisePower/Q)*complexNoise;
    %%%%%%%%%%%% Remove Cyclic Prefix%%%%%%%%%%%
    RxSamples_ifdma = RxSamples_ifdma(SP.CPsize+1:numSymbols+SP.CPsize);
    RxSamples_lfdma = RxSamples_lfdma(SP.CPsize+1:numSymbols+SP.CPsize);
    %%%%%%%%%%%%% SC-FDMA demodulation%%%%%%%%%%
    Y_ifdma = fft(RxSamples_ifdma, SP.FFTsize);
    Y_lfdma = fft(RxSamples_lfdma, SP.FFTsize);
    %%%%%%%%%%%% subcarriers demapping%%%%%%%%%%
    Y_ifdma = Y_ifdma(1+SP.subband:Q:numSymbols);
    Y_lfdma = Y_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
    %%%%%%%%%%%%%%%%% Equalization %%%%%%%%%%%%%%%%
    H_eff = H_channel(1+SP.subband:Q:numSymbols);
    if SP.equalizerType == 'ZERO'
        Y_ifdma = Y_ifdma./H_eff;
    elseif SP.equalizerType == 'MMSE'
        C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
        Y_ifdma = Y_ifdma.*C;
    end
    H_eff = H_channel([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
    if SP.equalizerType == 'ZERO'
        Y_lfdma = Y_lfdma./H_eff;
    elseif SP.equalizerType == 'MMSE'
        C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
        Y_lfdma = Y_lfdma.*C;
    end
    EstSymbols_ifdma = ifft(Y_ifdma);
```

```
    EstSymbols_lfdma = ifft(Y_lfdma);
    %%%%%%%%%%%%%%%% demodulation%%%%%%%%%%%%%%%
    EstSymbols_ifdma = sign(real(EstSymbols_ifdma)) ;
    EstSymbols_ifdma =(EstSymbols_ifdma+1)/2;
    EstSymbols_lfdma = sign(real(EstSymbols_lfdma)) ;
    EstSymbols_lfdma = (EstSymbols_lfdma+1)/2;
    demodata1_ifdma(:,jj) =  EstSymbols_ifdma(:);   % the output of scfdma columns%storing of
received image data
    demodata1_lfdma(:,jj)  = EstSymbols_lfdma(:);
    % the output of scfdma columns%storing of received image data
end
%*****************  Received image  ***************
[M3,N3] = size(demodata1_ifdma);
%  demodata2 = demodata1(:);
yy1_ifdma = reshape (demodata1_ifdma,M3,N3);
%reshaping the matrix to vector
yy1_lfdma = reshape (demodata1_lfdma,M3,N3);
%reshaping the matrix to vector
received_image_ifdma = yy1_ifdma(1:M2*N2);%taking the original data
received_image_lfdma = yy1_lfdma(1:M2*N2);%taking the original data
%**************  Regeneration of image  **************
zz1_ifdma=reshape(received_image_ifdma,M2* N2,1);
%reshaping to M2*N2 vector
zz1_lfdma=reshape(received_image_lfdma,M2* N2,1);
%reshaping to M2*N2 vector
yy_ifdma = reshape(zz1_ifdma,M2, N2);
yy_lfdma = reshape( zz1_lfdma,M2, N2);
clear i j
for i=1:M1
    for j=1:N1
        zn_ifdma(i,j)=num2str(yy_ifdma(i,j));
        zn_lfdma(i,j)=num2str(yy_lfdma(i,j));
    end;
end;
hn_ifdma=bin2dec(zn_ifdma);
hn_lfdma=bin2dec(zn_lfdma);
gn_ifdma=col2im(hn_ifdma, [M,N], [M,N], 'distinct');
gn_lfdma=col2im(hn_lfdma, [M,N], [M,N], 'distinct');
y1_ifdma=derandomization(gn_ifdma);
y1_lfdma=derandomization(gn_lfdma);
y1_ifdma=y1_ifdma/255;
y1_lfdma=y1_lfdma/255;
% **************** The output results***************
% figure (11)
% imshow(im)
% title('Input Image');
```

```
figure (2)
imshow(y1_ifdma)
title('DFT IFDMA With randomaization');
MSE1_ifdma=sum(sum((double(im)/255-y1_ifdma).^2))/prod(size(im));
PSNR_DFT_ifdma(sss)=10*log(1/MSE1_ifdma)/log(10);
figure (3)
imshow(y1_lfdma)
title('DFT LFDMA With randomaization');
MSE1_lfdma=sum(sum((double(im)/255-y1_lfdma).^2))/prod(size(im));
PSNR_DFT_lfdma(sss)=10*log(1/MSE1_lfdma)/log(10);
y1_ifdma_DFT(:,:,sss)=y1_ifdma;
y1_lfdma_DFT(:,:,sss)=y1_lfdma;
%PSNR(e)=10*log(1/MSE1)/log(10);% Peak signal-to-noise ratio 10*log10((max possible pixel value of the image)^2/MSE)
% end
toc
end
```

****

DCT_SCFDMA_wo.m

****

```
function [PSNR_DCT_ifdma1 PSNR_DCT_lfdma1 y1_ifdma_DCT1 y1_lfdma_DCT1] = DCT_SCFDMA_wo(SP)

for sss=1: length(SP.SNR)
SP.paths= fadchan(SP);
SP.equalizerType ='MMSE';
%%%%%%%==================================
numSymbols = SP.FFTsize;
Q = numSymbols/SP.inputBlockSize;
%%%%%%%%%============ SUI3 channel============
SUI3_1=[SP.paths(1,1) 0 0 SP.paths(2,1) 0 SP.paths(3,1)];
SP.channel = SUI3_1/norm(SUI3_1);
H_channel = fft(SP.channel,SP.FFTsize);

im1=imread('mri.gif');%image reading
im=imresize(im1,[256 256]);
%xx=randomization(im);%image randomization
%*****************Data Generation*******************
f = zeros(256,256);
f = double(im);
[M,N]=size(f);
g=im2col(f, [M,N],'distinct');%image to column converter
h=dec2bin(double(g));%pixel value to binary conversion...every value replaced by 8 bits string
```

```
[M1,N1] = size(h);
z=zeros (M1,N1);
clear i j
for i=1:M1
   for j=1:N1
      z(i,j)= str2num(h(i,j)); %string to number conversion
   end;
end;
[M2,N2] = size(z) ;
zz = reshape(z,M2*N2, 1);%parallel data reshaping date to vector
% ********** Dividing the image into blocks***********
nloops = ceil((M2*N2)/SP.inputBlockSize );%number of image %blocks by approximation
new_data = nloops*SP.inputBlockSize ;%new vector proportional to %block size
nzeros = new_data  - (M2*N2);%number of zeros to be added to %old data vector
input_data = [zz;zeros(nzeros,1)];%construction of new data %vector
input_data2 = reshape(input_data ,SP.inputBlockSize ,nloops); %reshape the new data to matrix
of block size rows to number of %blocks columns
save input_data2
%************** transmission ON SC-FDMA ***************
demodata1_ifdma = zeros(SP.inputBlockSize ,nloops);% this matrix to %store received data
vector
demodata1_lfdma = zeros(SP.inputBlockSize ,nloops);
clear jj
for jj =1: nloops % loop for columns
   b1= input_data2(:,jj)';%every block size
   %%%%%%%%%%%%%%%% QPSK Modulation %%%%%%%%%%%%%%%%
   tmp = b1;
   tmp = tmp*2 - 1;
   inputSymbols = (tmp(1,:) + i*tmp(1,:))/sqrt(2);
   %%%%%%%%%%%% SC-FDMA  Modulation %%%%%%%%%%%%
   inputSymbols_freq = dct(inputSymbols);
   inputSamples_ifdma = zeros(1,numSymbols);
   inputSamples_lfdma = zeros(1,numSymbols);
   %%%%%%%%%%%% Subcarriers Mapping %%%%%%%%%%%%
   inputSamples_ifdma(1+SP.subband:Q:numSymbols) = inputSymbols_freq;
   inputSamples_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband) =
inputSymbols_freq;
   inputSamples_ifdma = idct(inputSamples_ifdma);
   inputSamples_lfdma = idct(inputSamples_lfdma);
   %%%%%%%%%%%%% Add Cyclic Prefix %%%%%%%%%%%%%
   TxSamples_ifdma = [inputSamples_ifdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_ifdma];
   TxSamples_lfdma = [inputSamples_lfdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_lfdma];
   %%%%%%%%%%%%% Wireless channel %%%%%%%%%%%%%
   RxSamples_ifdma = filter(SP.channel, 1,TxSamples_ifdma); % Multipath Channel
```

```
    RxSamples_lfdma = filter(SP.channel, 1,TxSamples_lfdma); % Multipath Channel
    %%%%%%%%%%%%% Noise Generation %%%%%%%%%%%%%
    tmp = randn(2, numSymbols+SP.CPsize);
    complexNoise = (tmp(1,:) + i*tmp(2,:))/sqrt(2);
    noisePower = 10^(-SP.SNR(sss)/10);
    %%%%%%%%%%%%%%% Received signal%%%%%%%%%%%%%
    RxSamples_ifdma = RxSamples_ifdma + sqrt(noisePower/Q)*complexNoise;
    RxSamples_lfdma = RxSamples_lfdma + sqrt(noisePower/Q)*complexNoise;
    %%%%%%%%%%%% Remove Cyclic Prefix%%%%%%%%%%%%
    RxSamples_ifdma = RxSamples_ifdma(SP.CPsize+1:numSymbols+SP.CPsize);
    RxSamples_lfdma = RxSamples_lfdma(SP.CPsize+1:numSymbols+SP.CPsize);
    %%%%%%%%%%%%% SC-FDMA demodulation%%%%%%%%%%
    Y_ifdma = dct(RxSamples_ifdma, SP.FFTsize);
    Y_lfdma = dct(RxSamples_lfdma, SP.FFTsize);
    %%%%%%%%%%%% subcarriers demapping%%%%%%%%%%
    Y_ifdma = Y_ifdma(1+SP.subband:Q:numSymbols);
    Y_lfdma = Y_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
    %%%%%%%%%%%%%%%%%% Equalization %%%%%%%%%%%%%%%%%%
    H_eff = H_channel(1+SP.subband:Q:numSymbols);
    if SP.equalizerType == 'ZERO'
       Y_ifdma = Y_ifdma./H_eff;
    elseif SP.equalizerType == 'MMSE'
       C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
       Y_ifdma = Y_ifdma.*C;
    end
    H_eff = H_channel([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
    if SP.equalizerType == 'ZERO'
       Y_lfdma = Y_lfdma./H_eff;
    elseif SP.equalizerType == 'MMSE'
       C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
       Y_lfdma = Y_lfdma.*C;
    end
    EstSymbols_ifdma = idct(Y_ifdma);
    EstSymbols_lfdma = idct(Y_lfdma);
    %%%%%%%%%%%%%%%% demodulation%%%%%%%%%%%%%%%
    EstSymbols_ifdma = sign(real(EstSymbols_ifdma)) ;
    EstSymbols_ifdma =(EstSymbols_ifdma+1)/2;
    EstSymbols_lfdma = sign(real(EstSymbols_lfdma)) ;
    EstSymbols_lfdma = (EstSymbols_lfdma+1)/2;
    demodata1_ifdma(:,jj) =  EstSymbols_ifdma(:);   % the output of scfdma columns%storing of
received image data
    demodata1_lfdma(:,jj)  = EstSymbols_lfdma(:);
    % the output of scfdma columns%storing of received image data
end
%*****************  Received image  ***************
[M3,N3] = size(demodata1_ifdma);
```

```
%  demodata2 = demodata1(:);
yy1_ifdma = reshape (demodata1_ifdma,M3,N3);
%reshaping the matrix to vector
yy1_lfdma = reshape (demodata1_lfdma,M3,N3);
%reshaping the matrix to vector
received_image_ifdma = yy1_ifdma(1:M2*N2);%taking the original data
received_image_lfdma = yy1_lfdma(1:M2*N2);%taking the original data
%**************  Regeneration of image  **************
zz1_ifdma=reshape(received_image_ifdma,M2* N2,1);
%reshaping to M2*N2 vector
zz1_lfdma=reshape(received_image_lfdma,M2* N2,1);
%reshaping to M2*N2 vector
yy_ifdma = reshape(zz1_ifdma,M2, N2);
yy_lfdma = reshape( zz1_lfdma,M2, N2);
clear i j
for i=1:M1
   for j=1:N1
       zn_ifdma(i,j)=num2str(yy_ifdma(i,j));
       zn_lfdma(i,j)=num2str(yy_lfdma(i,j));
   end;
end;
hn_ifdma=bin2dec(zn_ifdma);
hn_lfdma=bin2dec(zn_lfdma);
gn_ifdma=col2im(hn_ifdma, [M,N], [M,N], 'distinct');
gn_lfdma=col2im(hn_lfdma, [M,N], [M,N], 'distinct');
%y1_ifdma=derandomization(gn_ifdma);
%y1_lfdma=derandomization(gn_lfdma);
y1_ifdma=gn_ifdma/255;
y1_lfdma=gn_lfdma/255;
% **************** The output results***************
% figure (211)
% imshow(im)
figure (8)
imshow(y1_ifdma)
title('DCT IFDMA Without randomaization');
MSE1_ifdma=sum(sum((double(im)/255-y1_ifdma).^2))/prod(size(im));
PSNR_DCT_ifdma1(sss)=10*log(1/MSE1_ifdma)/log(10);
figure (9)
imshow(y1_lfdma)
title('DCT LFDMA Without randomaization');
MSE1_lfdma=sum(sum((double(im)/255-y1_lfdma).^2))/prod(size(im));
PSNR_DCT_lfdma1(sss)=10*log(1/MSE1_lfdma)/log(10);
y1_ifdma_DCT1(:,:,sss)=y1_ifdma;
y1_lfdma_DCT1(:,:,sss)=y1_lfdma;
toc
end
```

****

DFT_SCFDMA_wo.m

****

```
function [PSNR_DFT_ifdma1 PSNR_DFT_lfdma1 y1_ifdma_DFT1 y1_lfdma_DFT1] =
DFT_SCFDMA_wo(SP)

for sss=1: length(SP.SNR)

SP.paths= fadchan(SP);
SP.equalizerType ='MMSE';
%%%%%%%==================================
numSymbols = SP.FFTsize;
Q = numSymbols/SP.inputBlockSize;
%%%%%%%%%============ SUI3 channel============
SUI3_1=[SP.paths(1,1) 0 0 SP.paths(2,1) 0 SP.paths(3,1)];
SP.channel = SUI3_1/norm(SUI3_1);
H_channel = fft(SP.channel,SP.FFTsize);

im1=imread('mri.gif');%image reading
im=imresize(im1,[256 256]);
%xx=randomization(im);%image randomization
%*****************Data Generation*******************
f = zeros(256,256);
f = double(im);
[M,N]=size(f);
g=im2col(f, [M,N],'distinct');%image to column converter
h=dec2bin(double(g));%pixel value to binary conversion...every value replaced by 8 bits string
[M1,N1] = size(h);
z=zeros (M1,N1);
clear i j
for i=1:M1
   for j=1:N1
      z(i,j)= str2num(h(i,j)); %string to number conversion
   end;
end;
[M2,N2] = size(z) ;
zz = reshape(z,M2*N2, 1);%parallel data reshaping date to vector
% ********** Dividing the image into blocks***********
nloops = ceil((M2*N2)/SP.inputBlockSize );%number of image %blocks by approximation
new_data = nloops*SP.inputBlockSize ;%new vector proportional to %block size
nzeros = new_data  - (M2*N2);%number of zeros to be added to %old data vector
```

```
input_data = [zz;zeros(nzeros,1)];%construction of new data %vector
input_data2 = reshape(input_data ,SP.inputBlockSize ,nloops); %reshape the new data to matrix
of block size rows to number of %blocks columns
save input_data2
%************** transmission ON SC-FDMA ***************
demodata1_ifdma = zeros(SP.inputBlockSize ,nloops);% this matrix to %store received data
vector
demodata1_lfdma = zeros(SP.inputBlockSize ,nloops);
clear jj
for jj =1: nloops % loop for columns
   b1= input_data2(:,jj)';%every block size
   %%%%%%%%%%%%%%% QPSK Modulation %%%%%%%%%%%%%%%
   tmp = b1;
   tmp = tmp*2 - 1;
   inputSymbols = (tmp(1,:) + i*tmp(1,:))/sqrt(2);
   %%%%%%%%%%%% SC-FDMA  Modulation %%%%%%%%%%%%
   inputSymbols_freq = fft(inputSymbols);
   inputSamples_ifdma = zeros(1,numSymbols);
   inputSamples_lfdma = zeros(1,numSymbols);
   %%%%%%%%%%%% Subcarriers Mapping %%%%%%%%%%%%
   inputSamples_ifdma(1+SP.subband:Q:numSymbols) = inputSymbols_freq;
   inputSamples_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband) =
inputSymbols_freq;
   inputSamples_ifdma = ifft(inputSamples_ifdma);
   inputSamples_lfdma = ifft(inputSamples_lfdma);
   %%%%%%%%%%%%% Add Cyclic Prefix %%%%%%%%%%%%%
   TxSamples_ifdma = [inputSamples_ifdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_ifdma];
   TxSamples_lfdma = [inputSamples_lfdma(numSymbols-SP.CPsize+1:numSymbols)
inputSamples_lfdma];
   %%%%%%%%%%%%% Wireless channel %%%%%%%%%%%%%%
   RxSamples_ifdma = filter(SP.channel, 1,TxSamples_ifdma); % Multipath Channel
   RxSamples_lfdma = filter(SP.channel, 1,TxSamples_lfdma); % Multipath Channel
   %%%%%%%%%%%%% Noise Generation %%%%%%%%%%%%%
   tmp = randn(2, numSymbols+SP.CPsize);
   complexNoise = (tmp(1,:) + i*tmp(2,:))/sqrt(2);
   noisePower = 10^(-SP.SNR(sss)/10);
   %%%%%%%%%%%%%%% Received signal%%%%%%%%%%%%%%
   RxSamples_ifdma = RxSamples_ifdma + sqrt(noisePower/Q)*complexNoise;
   RxSamples_lfdma = RxSamples_lfdma + sqrt(noisePower/Q)*complexNoise;
   %%%%%%%%%%%% Remove Cyclic Prefix%%%%%%%%%%%%
   RxSamples_ifdma = RxSamples_ifdma(SP.CPsize+1:numSymbols+SP.CPsize);
   RxSamples_lfdma = RxSamples_lfdma(SP.CPsize+1:numSymbols+SP.CPsize);
   %%%%%%%%%%%%% SC-FDMA demodulation%%%%%%%%%%%
   Y_ifdma = fft(RxSamples_ifdma, SP.FFTsize);
   Y_lfdma = fft(RxSamples_lfdma, SP.FFTsize);
```

```
    %%%%%%%%%%%%% subcarriers demapping%%%%%%%%%%%
    Y_ifdma = Y_ifdma(1+SP.subband:Q:numSymbols);
    Y_lfdma = Y_lfdma([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
    %%%%%%%%%%%%%%%%%%% Equalization %%%%%%%%%%%%%%%%%%%
    H_eff = H_channel(1+SP.subband:Q:numSymbols);
    if SP.equalizerType == 'ZERO'
       Y_ifdma = Y_ifdma./H_eff;
    elseif SP.equalizerType == 'MMSE'
       C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
       Y_ifdma = Y_ifdma.*C;
    end
    H_eff = H_channel([1:SP.inputBlockSize]+SP.inputBlockSize*SP.subband);
    if SP.equalizerType == 'ZERO'
       Y_lfdma = Y_lfdma./H_eff;
    elseif SP.equalizerType == 'MMSE'
       C = conj(H_eff)./(conj(H_eff).*H_eff +10^(-SP.SNR(sss)/10));
       Y_lfdma = Y_lfdma.*C;
    end
    EstSymbols_ifdma = ifft(Y_ifdma);
    EstSymbols_lfdma = ifft(Y_lfdma);
    %%%%%%%%%%%%%%%%%% demodulation%%%%%%%%%%%%%%%%%%
    EstSymbols_ifdma = sign(real(EstSymbols_ifdma)) ;
    EstSymbols_ifdma =(EstSymbols_ifdma+1)/2;
    EstSymbols_lfdma = sign(real(EstSymbols_lfdma)) ;
    EstSymbols_lfdma = (EstSymbols_lfdma+1)/2;
    demodata1_ifdma(:,jj) =  EstSymbols_ifdma(:);   % the output of scfdma columns%storing of
received image data
    demodata1_lfdma(:,jj)  = EstSymbols_lfdma(:);
    % the output of scfdma columns%storing of received image data
end
%*****************  Received image  ***************
[M3,N3] = size(demodata1_ifdma);
%  demodata2 = demodata1(:);
yy1_ifdma = reshape (demodata1_ifdma,M3,N3);
%reshaping the matrix to vector
yy1_lfdma = reshape (demodata1_lfdma,M3,N3);
%reshaping the matrix to vector
received_image_ifdma = yy1_ifdma(1:M2*N2);%taking the original data
received_image_lfdma = yy1_lfdma(1:M2*N2);%taking the original data
%**************  Regeneration of image  **************
zz1_ifdma=reshape(received_image_ifdma,M2* N2,1);
%reshaping to M2*N2 vector
zz1_lfdma=reshape(received_image_lfdma,M2* N2,1);
%reshaping to M2*N2 vector
yy_ifdma = reshape(zz1_ifdma,M2, N2);
yy_lfdma = reshape( zz1_lfdma,M2, N2);
```

```
clear i j
for i=1:M1
   for j=1:N1
      zn_ifdma(i,j)=num2str(yy_ifdma(i,j));
      zn_lfdma(i,j)=num2str(yy_lfdma(i,j));
   end;
end;
hn_ifdma=bin2dec(zn_ifdma);
hn_lfdma=bin2dec(zn_lfdma);
gn_ifdma=col2im(hn_ifdma, [M,N], [M,N], 'distinct');
gn_lfdma=col2im(hn_lfdma, [M,N], [M,N], 'distinct');
%y1_ifdma=derandomization(gn_ifdma);
%y1_lfdma=derandomization(gn_lfdma);
y1_ifdma=gn_ifdma/255;
y1_lfdma=gn_lfdma/255;
% ***************** The output results****************
% figure (111)
% imshow(im)
figure (6)
imshow(y1_ifdma)
title('DFT IFDMA Without randomaization');
MSE1_ifdma=sum(sum((double(im)/255-y1_ifdma).^2))/prod(size(im));
PSNR_DFT_ifdma1(sss)=10*log(1/MSE1_ifdma)/log(10);
figure (7)
imshow(y1_lfdma)
title('DFT LFDMA Without randomaization');
MSE1_lfdma=sum(sum((double(im)/255-y1_lfdma).^2))/prod(size(im));
PSNR_DFT_lfdma1(sss)=10*log(1/MSE1_lfdma)/log(10);
y1_ifdma_DFT1(:,:,sss)=y1_ifdma;
y1_lfdma_DFT1(:,:,sss)=y1_lfdma;
%PSNR(e)=10*log(1/MSE1)/log(10);% Peak signal-to-noise ratio 10*log10((max possible pixel
value of the image)^2/MSE)
% end
toc
end
```