Assignment3

12110644周思呈

1 Applications of FFT

Some of the important applications of the FFT include:

- fast large-integer multiplication algorithms and polynomial multiplication,
- efficient matrix–vector multiplication for Toeplitz, circulant and other structured matrices,
- filtering algorithms (see overlap–add and overlap–save methods),
- fast algorithms for discrete cosine or sine transforms (e.g. fast DCT used for JPEG and MPEG/MP3 encoding and decoding),
- fast Chebyshev approximation,
- solving difference equations,
- computation of isotopic distributions.
- modulation and demodulation of complex data symbols using orthogonal frequency division multiplexing (OFDM) for 5G, LTE, Wi-Fi, DSL, and other modern communication systems. [1]

2 MP3 Encoding in Detail

This section quotes 知乎-傅里叶变换[2] and CSDN-全面解析傅立叶变换（非常详细）[3].

To represent sound signal in digital world, we have to encode it into digital signal. After sampling, analog signals are converted into digital signals, which can be processed by computers. FFT is quite useful in converting and processing period.
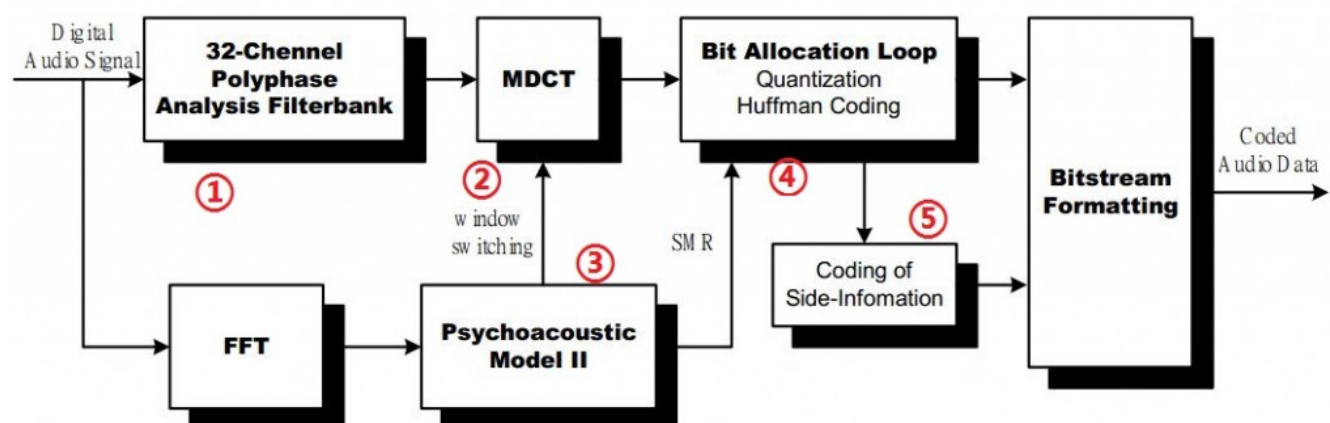

Fig2.1, MP3 encoding flowchart. [4]

2.1 FT and DFT in Sampling

A sound signal consists of multiple sound waveforms of different frequencies which propagate together by causing perturbations to the medium. When recording the sound, we only capture the resultant amplitudes of these

waveforms. FT(Fourier Transform) can decompose a continuous signal into its constituent frequencies and magnitudes, while DFT(Discrete Fourier Transform) can deal with the discrete ones, which can be processed by computer.

Periodically transformed signals can be represented by Fourier series. Fourier series treat $\{1, \sin x, \cos x, \sin 2x, \cos 2x, \cdots, \}$ as bases in the space and express the original function as a linear combination of these bases.

$$f(x) = a_0/2 + \sum_{n=1}^{\infty} a_n \cos nx + b_n \sin nx$$

where

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)dx$$
$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx dx$$
$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx dx$$

In this way, every frequency function can be represented by combinations of certain frequencies.
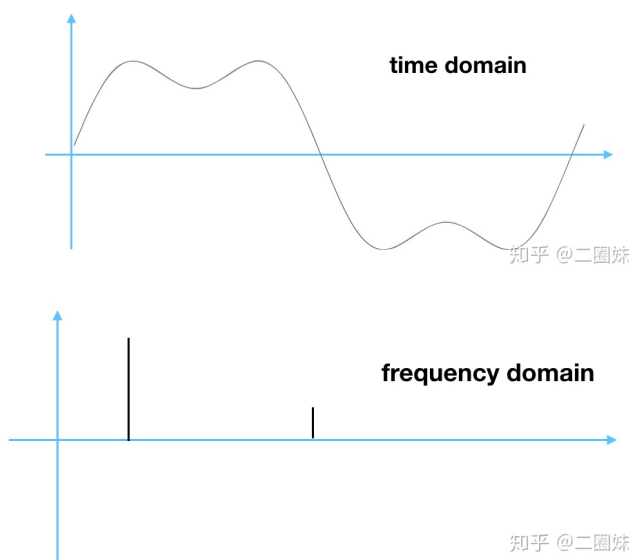

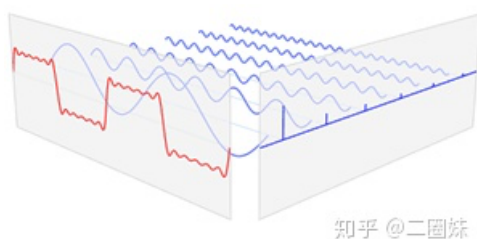*Fig 2.2, the same function in time domain and frequency domain.*

Fig 2.3, every periodic function can be expressed as combinations of frequencies. Red, time domain. Blue, frequency domain.

Aperiodic function can be expressed by using FT(Fourier Transform). By using Euler's Formula, we have:

$$e^{i\varphi} = \cos \varphi + i \sin \varphi$$
$$\cos \varphi = \frac{e^{i\varphi} + e^{-i\varphi}}{2}$$
$$\sin \varphi = \frac{e^{i\varphi} - e^{-i\varphi}}{2}$$

$$Thus, Fourier Series can be rewritten in this way:$$
$$\begin{cases} f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega_0 nx} \\ c_n = \frac{1}{T} \int_{x_0}^{x_0+T} f(x) e^{-i\omega_0 nx} dx \end{cases}$$

In another word, $e^{i\omega_0 nx}$ are used as bases to represent $f(x)$ as its linear combination.

$$X(f) = \mathcal{F}[x(t)] = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt$$
$$x(t) = \mathcal{F}^{-1}[X(f)] = \int_{-\infty}^{\infty} X(f) e^{i2\pi ft} df$$

DFT(Discrete Fourier Transform) is used to transform a signal sampled in the time domain into a signal sampled in the discrete time Fourier transform frequency domain.

$$x_n = \sum_{k=0}^{N-1} X_k e^{i\frac{2\pi}{N} kn} \quad n = 0, \ldots, N-1$$

We have already known that every signal can be expressed in the form of linear combinations of sine wave and cosine wave, therefore we can use Signal Correlation method to compute the coefficients of linear combination.

Before computation, we need to know that a signal of length N in the Fourier transform be decomposed into $N/2 + 1$ cosine and sine signals. For a signal of length N, it can be decomposed into N complex exponential functions (including sine and cosine functions) with frequency range from 0 to N-1. However, in practical applications, we usually only need to consider frequency components ranging from 0 to $N/2$, because the frequencies of the signal are symmetric, and frequency components exceeding $N/2$ can be obtained by taking the conjugate of the first half frequency components.

If two signals A and B are conjugate symmetric, i.e., $A[n] = B[N - n]^*$ and $B[n] = A[N - n]^*$ (where * denotes complex conjugate), then their DFT frequency domain representations are also conjugate symmetric. The DFT result is computed by multiplying the input signal with the sine and cosine of each frequency (the correlation operation) to get the correlation degree of the original signal with each frequency.

2.2 FFT in Acceleration

Up to now, we only use real number to do computation. However, in FFT, we make full use of complex number to accelerate the computation. We do this in the order of following steps.

1. Write the sine and cosine functions as complex numbers. The complex Fourier transform treats the original signal $x[n]$ as a signal represented by a complex number, where the real part represents the original signal value and the imaginary part is 0. The transform result $x[k]$ is also in the form of a complex number, but here the imaginary part is valued.
2. Carry out the correlation algorithm for complex numbers. As in the real range, we can multiply the original signal by a signal in the form of an orthogonal function, and then sum it up to obtain the components of the orthogonal signal contained in the original signal.

3 Wi-Fi in Detail

Fast Fourier Transform (FFT) has wide applications in WiFi technology. WiFi technology uses Frequency Division Multiplexing (FDM, specifically OFDM, Orthogonal Frequency-Division Multiplexing) to divide wireless signals into different sub-channels and send different data streams on each sub-channel. To achieve FDM, WiFi technology needs to use FFT to transform the time-domain signal into the frequency-domain signal and separate the signals on different sub-channels.

In addition, channel estimation also requires the use of FFT. Since the signal is subject to various interference and attenuation during transmission, it is necessary to estimate and correct the channel to ensure the accuracy and reliability of the data. By performing FFT on the received signal, the frequency response of the channel can be obtained, and the channel can be estimated and corrected.

Furthermore, FFT can also be used for multipath fading estimation, an important issue in wireless communication. Multipath fading refers to the signal encountering multiple paths and fading on different paths during propagation. By performing FFT on the received signal, the signal propagation delay and phase offset on different paths can be obtained, and the multipath fading can be estimated and corrected, improving the reliability and robustness of the signal.

Notice that the applications of FFT in Wi-Fi technology requires professional math knowledge, a lot of which is quite beyond me. Hopefully I will be able to explain them in detail.

3.1 OFDM

The bandwidth provided by the channel of a communication system is usually much wider than the bandwidth required to transmit a single signal. FDM modulates multiple signals in one bandwidth to make full use of it. OFDM divides the channel into several orthogonal subchannels, converts the high-speed data signals into parallel low-speed subdata streams, and uses IFFT and FFT to realize modulation and demodulation.
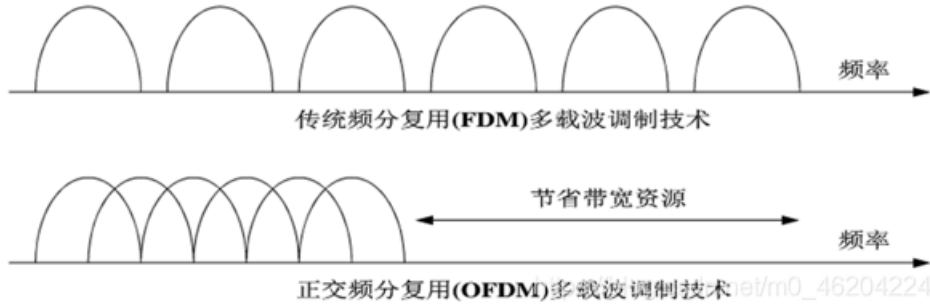

Fig 3.1, FDM and OFDM.

In a general wireless channel, the base frequency model can be regarded as follows. [5]
Input: $x[n]$
Output: $y[n] = \sum_{j=0}^{L-1} h_l x[n-l] + z[n] = h_0 x[n] + \sum_{j=1}^{L-1} h_l x[n-l] + z[n]$

In the output formula, $h_0 x[n]$ is the signal we want, $\sum_{j=1}^{L-1} h_l x[n-l]$ is inter-symbol interference(ISI), which means the interference from past signal to present signal, and $z[n]$ is noise.

The flow of OFDM is as follows:

$$\{\tilde{x_k}\} \rightarrow \text{ pre-processing } \rightarrow \{x[n]\} \rightarrow h_l \rightarrow \{y[n]\} \rightarrow \text{ post-processing } \rightarrow \{\tilde{y_k}\}$$

If we have $\tilde{y_k} = \tilde{h_k}\tilde{x_k}$, this system is ISI-free.

Notice that we have to do linear convolution to the original signal. One possible way to reduce time complexity is to convert it from time domain to frequency domain, because then we can transfer convolution into direct multiplications[6]. For the reasons why convolution in the time domain equals multiplication in the frequency domain, the proof is omitted(because it's beyond me).

So the big picture now becomes:

$$\{\tilde{x_k}\} \rightarrow \text{ IDFT } \rightarrow \{x[n]\} \rightarrow h_l \rightarrow \{y[n]\} \rightarrow \text{ DFT } \rightarrow \{\tilde{y_k}\}$$

3.2 Channel Estimation

The signal will be subject to a variety of interference in the propagation process, when it reaches the receiver, the amplitude, phase and frequency of the signal will change greatly. The role of channel estimation and channel equalization is to recover the signal as much as possible.
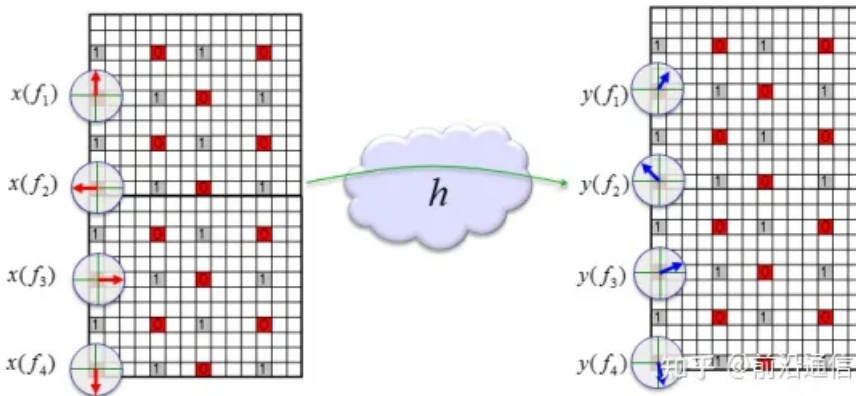

Fig 3.2, Channel estimation is the process of finding correlations between the complex array on the left and the complex array on the right. [7]

The process is as follows[8]:

1. Set up a mathematical model that relates the "transmitted signal" to the "received signal" using a "channel" matrix.
2. Sends a known signal ("reference signal" or "pilot signal") and detects the received signal.
3. The elements of the channel matrix can be calculated by comparing the transmitted and received signals.

The basic channel estimation algorithm is Least Square Estimation. It use least square to compute CFR(Channel Frequency Response) and recover the data symbols in the whole channel by interpolation.

Based on LS, IDFT channel estimation greatly improve the efficiency. In order to improve the performance of the LS algorithm, the CFR obtained by the LS algorithm was first transformed to the time domain by IDFT. Then according to the characteristics of the Channel Impulse Response (CIR), the remaining positions in the CIR except the time delay point were set to zero to achieve the purpose of denoising.

---

1. https://en.wikipedia.org/wiki/Fast_Fourier_transform#Applications ↵
2. https://zhuanlan.zhihu.com/p/104079068 ↵
3. https://blog.csdn.net/liusandian/article/details/51788953 ↵
4. https://blog.csdn.net/qingkongyeyue/article/details/70984891 ↵
5. https://zh.wikipedia.org/wiki/正交頻分复用#使用FFT演算法實現 ↵
6. https://www.zhihu.com/question/25525824 ↵
7. https://zhuanlan.zhihu.com/p/368716371 ↵
8. https://blog.csdn.net/qq_36554582/article/details/108918657 ↵