傅里叶变换与图示

Dezeming Family

2022 年 4 月 15 日

DezemingFamily 系列书和小册子因为是电子书，所以可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列书，可以从我们的网站 [https://dezeming.top/] 找到最新版。对书的内容建议和出现的错误欢迎在网站留言。

目录

一　基本介绍

学会看傅里叶变换的频谱很重要，这是做傅里叶分析的基础。尤其是离散时间傅里叶变换和离散傅里叶变换，分析其高低频成分是很重要的。

二　连续时间傅里叶变换

对连续时间非周期信号 $x(t)$ 的连续时间傅里叶变换的公式是：

$$X(j\omega) = \int_{-\infty}^{+\infty} x(x)e^{-j\omega x}dx \tag{二.1}$$

很明显这里 ω 越大则表示频率越大。

三　离散时间傅里叶变换

对离散时间非周期信号 $x[n]$ 的离散时间傅里叶变换的公式是：

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n} \tag{三.1}$$

可知傅里叶变换是以 2π 为周期重复的：

$$\begin{aligned} X(e^{j(\omega+2\pi)}) &= \sum_{n=-\infty}^{+\infty} x[n]e^{-j(\omega+2\pi)n} \\ &= \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}e^{-j2\pi n} \\ &= \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n} \end{aligned} \tag{三.2}$$

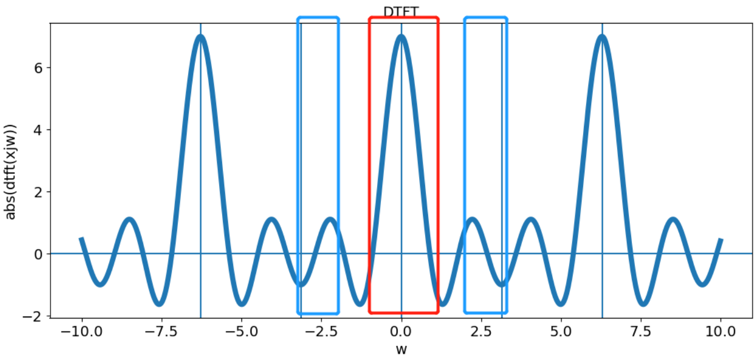
那么，问题在于，既然傅里叶变换是周期函数，那么高频部分是在什么地方呢？

我们用 python 写一个程序来验证一下（代码参考自 [2]，但是索引博客里面有不少错误，这里进行了更正）：

```python
import math
import numpy as np
class dtft():
    def __init__(self,xvalues=[], xindex=[]):
        self.yvalues = []
        self.index = xindex
        self.xvalues = xvalues
    def xjw(self,fre = []):
        for f in fre:
            p = 0;
            i = 0
            for x in self.xvalues:
                p = math.e**(-1j * f * self.index[i]) * x + p
                i = i + 1
            self.yvalues.append(p)
```

```python
import matplotlib.pyplot as plt
# 数据与坐标
xvalues = [1,1,1,1,1,1,1]
xindex = [-3,-2,-1,0,1,2,3]
# 画图
DTFT = dtft(xvalues,xindex)
W = np.arange(-10, 10, 0.01)
DTFT.xjw(W)
xjw = []
for x in DTFT.yvalues:
    xjw.append((x))
plt.plot(W, xjw, linewidth=5)
# 画出 x 为 pi的整数倍的竖线
for i in range(-2,3,1):
    plt.axvline(i * 3.1415926)
# 画出 y=0 水平线
plt.axhline(0)
# 标题与坐标
plt.title('DTFT', fontsize=14)
plt.xlabel("w", fontsize=14)
plt.ylabel("abs(dtft(xjw))", fontsize=14)
plt.tick_params(axis='both', labelsize=14)
# 显示
plt.show()
```

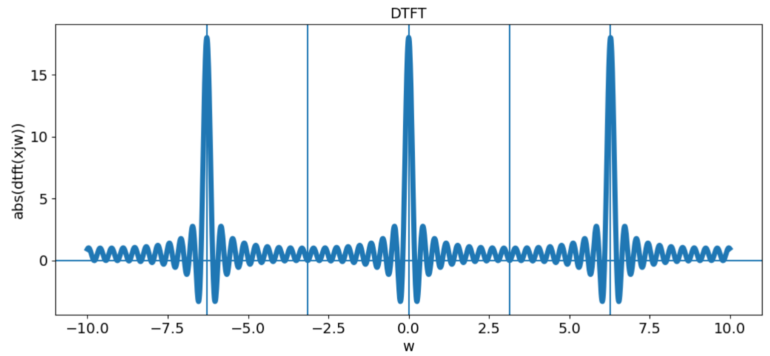
结果为：



当信号设置为：

```python
xvalues = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
xindex = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17]
```

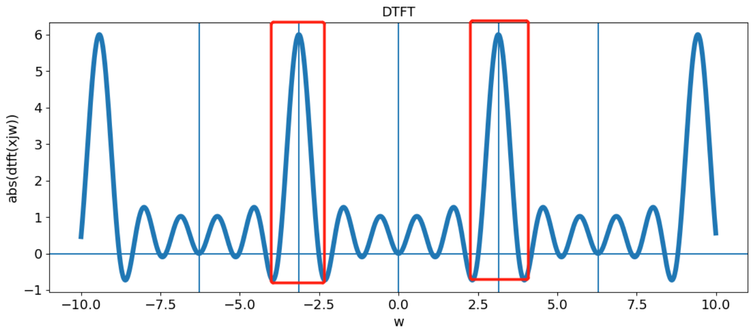
傅里叶变换的结果是：



当信号变为高频时：

```
xvalues = [1,−1,1,−1,1,−1]
xindex = [0,1,2,3,4,5]
```

傅里叶变换的结果是：



上面的两幅图中，有的用红色框标示了出来，表示成分比较高的区域，蓝色框表示成分比较低的区域。可以看到，第一个信号和第二个信号因为是方形信号，所以低频占主要成分；第三个信号是高频信号，高频占主要成分，因此，离散时间傅里叶变换中，靠近 π 的整数倍的地方表示高频，靠近 2π 的整数倍的地方表示低频。

四　离散傅里叶变换

非周期信号的离散傅里叶变换表示为：

$$W_N = e^{-j\frac{2\pi}{N}}$$
$$X[k] = \begin{cases} \sum_{n=0}^{N-1} x[n]W_N^{kn}, & 0 \le k \le N-1 \\ 0, & otherwise \end{cases} \tag{四.1}$$

我们应该如何考虑它的频谱特点呢？

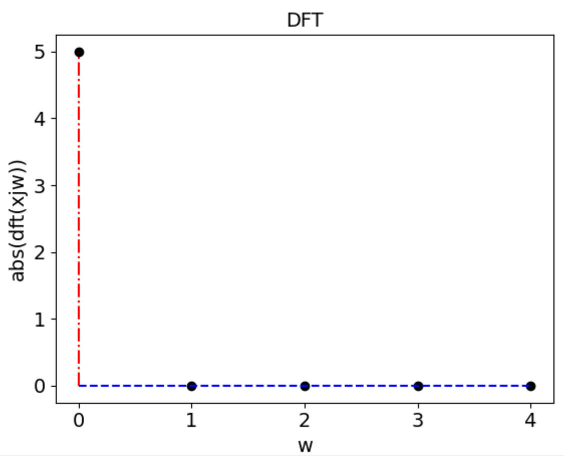
我们写一个程序来验证一下，我们实现的 DFT 只是最基本的思路，并没有进行时间或者频域抽取（不是用 FFT 做的）：

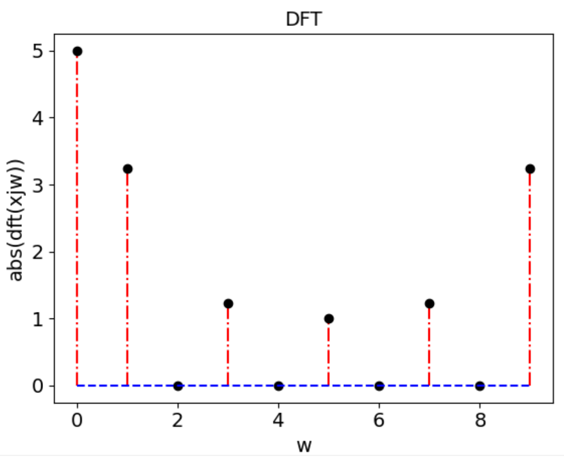
```
import math
import numpy as np
class dft:
    def __init__(self,Num,xvalues):
        self.xvalues = xvalues
        self.yvalues = []
        self.Num = Num
        self.xk = []
        self.index = list(range(0,self.Num,1))
        self.Xk()
```

```python
    def xjw(self,fre = []):
        for f in fre:
            p = 0
            i = 0
            for x in self.xvalues:
                p = math.e**(-1j * f * self.index[i]) * x + p
                i = i + 1
            self.yvalues.append(p)
    def Xk(self):
        W = []
        for x in self.index:
            W.append(x*2*math.pi/self.Num)
        self.xjw(W)
        for y in self.yvalues:
            self.xk.append(abs(y))
```

```python
import matplotlib.pyplot as plt
# 定义数据
xvalues = [1,1,1,1,1]
N1 = 5
DFT1 = dft(N1,xvalues)
# 绘图
X,Y,Z = plt.stem(DFT1.index,DFT1.xk,markerfmt = "o",linefmt = "-.",basefmt =
    "--")
plt.setp(X,color = 'k')
plt.setp(Y,color = 'r')
plt.setp(Z,color = 'b')
plt.title('DFT',fontsize = 14)
plt.xlabel("w",fontsize = 14)
plt.ylabel("abs(dft(xjw))",fontsize = 14)
plt.tick_params(axis = 'both',labelsize = 14,which = 'major')
# 显示
plt.show()
```

程序里的 N1 表示我们设定的有限长序列的长度，设置为 5 和 10 时分别表示如下：



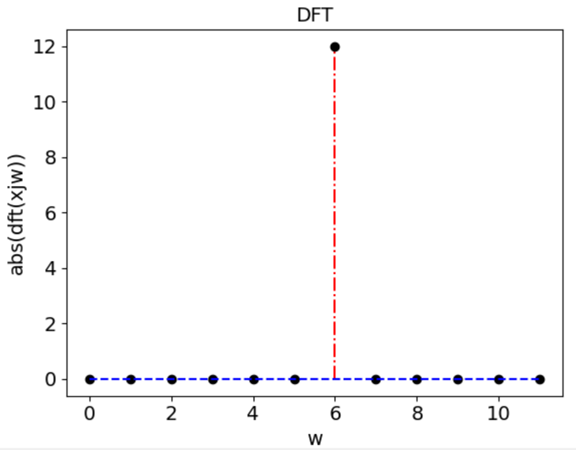


$N = 5$ 时很好理解，周期扩展以后就相当于是一个直流信号；$N = 10$ 则周期拓展就是一个周期方波了。

当我们这么取数据：

```
xvalues = [1,-1,1,-1,1,-1,1,-1,1,-1,1,-1]
N1 = 12
```

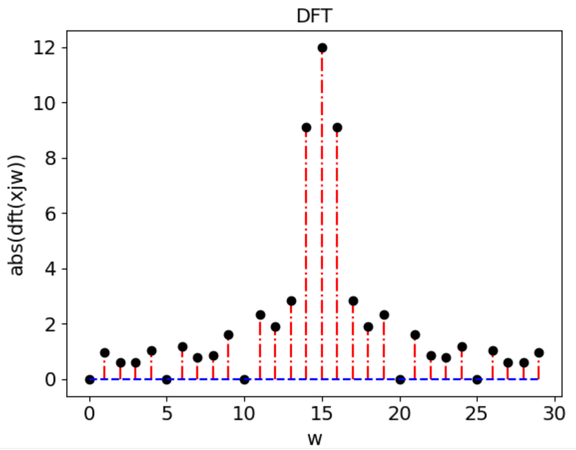
得到的离散傅里叶变换结果为：



由此可知，离散傅里叶变换的结果中，靠近中间的属于高频成分，靠近两边的属于低频成分。如果是一个方波信号：

```
xvalues = [1,-1,1,-1,1,-1,1,-1,1,-1,1,-1]
N1 = 30
```

DFT 为：



我们对照着频谱，自己分析一下，就能明白这里面的关系了。（这里提一句，对图像做傅里叶变换时，四周都是低频，中间是高频，但是由于能量一般都集中在低频区域，所以为了好看会进行“中心化”，使得中间更亮）。

参考文献

[1] Signals and Systems Alan V. Oppenheim, Alan S. Willsky, with S. Hamid Nawab. Prentice Hall, 2013.

[2] [https://blog.csdn.net/weixin_42193451/article/details/122384485]