# Project 1

16340305 郑先淇

## 题目

*Write programs (using Matlab or other software) to finish the exercises below.*

$$For \quad f(t)=\begin{cases}\frac{1}{2}[1+\cos(t)], & 0\le |t|\le \pi \\ 0, & |t|>\pi\end{cases},$$

*(1) Plot this signal and its frequency spectrum;*

*(2) When the sampling period satisfies $T=1$, $T=\pi/2$, $T=2$, respectively, please plot the sampling signal $f_p(n)$ and its frequency spectrum, respectively. Please give explanation of these results;*

*(3) Using lowpass filter with cutting frequency $\omega_c=2.4$ to reconstruct signal $f_r(t)$ from $f_p(n)$. When the sampling period satisfies $T=1$, $T=2$, respectively, please plot the reconstructed signal $f_r(t)$, and plot the absolute error between the reconstructed signal $f_r(t)$ and the original signal $f(t)$. Please analyze these results.*

## 解答

(1) 首先是画出函数曲线，观察上述函数表达式，f(t)的定义域为 R，但是只在[-π,π]取非 0 值，为画图方便，我们只画出其在[-10,10]的图像，很简单，此处不再赘述，下面我们画一下它的频谱。画出一个时域信号的频谱，首先要对其进行傅里叶变换，因为初始时域信号是连续信号，所以我们要做 CTFT，公式如下：

$$X_a(j\Omega) = \int x_a(t)e^{-j\Omega t}dt$$

但是在计算机上无法直接做定积分运算，所以我们通过用级数模拟积分的方法来近似求解定积分的值。在这里，我们将-10 到 10 划分为时间间隔△t 为 0.001 的时间序列，求所有时刻函数值的级数近似作为-10 到 10 之间的积分，近似公式如下：

$$X_a(j\Omega) = \sum_m x_a(m\Delta t)e^{-j\Omega m\Delta t}\Delta t$$

在 Matlab 中，函数的自变量因变量的集合都是使用矩阵来存储的，从矩阵的角度来看傅里叶变换的公式如下：

$$[X_a(0)\ X_a(1)\ X_a(2)\ ..] = [x_a(0)\ x_a(1)\ x_a(2)\ ..]\begin{bmatrix} e^{-j\omega_0 t_0} & e^{-j\omega_1 t_0} & \cdots & e^{-j\omega_K t_0} \\ e^{-j\omega_0 t_1} & e^{-j\omega_1 t_1} & \cdots & e^{-j\omega_K t_1} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j\omega_0 t_N} & e^{-j\omega_1 t_N} & \cdots & e^{-j\omega_K t_N} \end{bmatrix}$$

时间向量如下：

$$t = [t_0 : \Delta t : t_N]$$

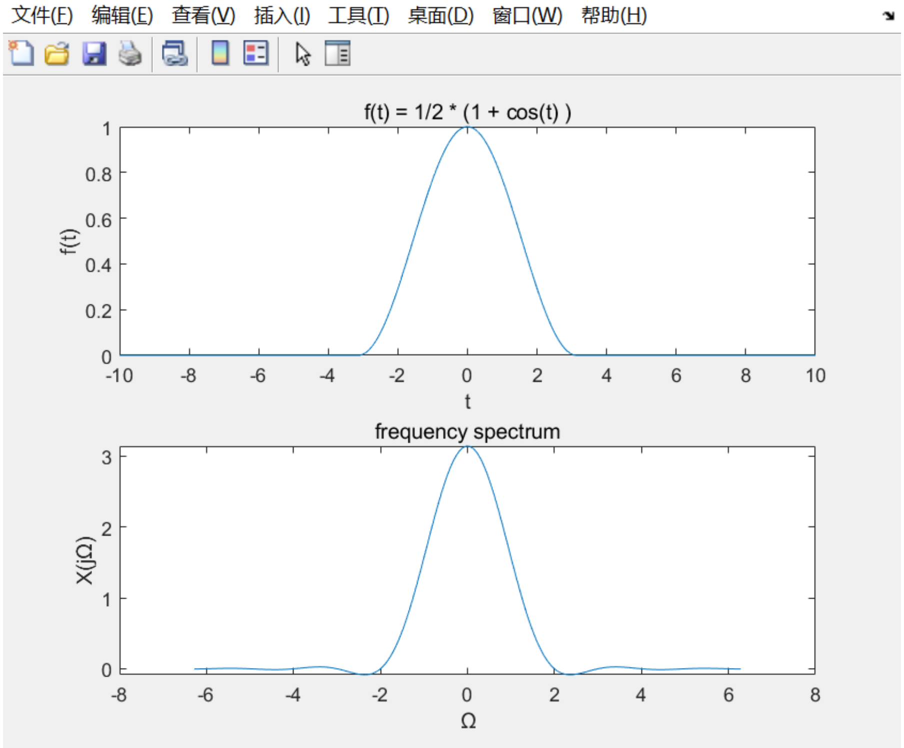
在这里，我们将[-2π,2π]的频率分为 1000 份，频率向量为：

$$\omega = [\omega_0\ \omega_1\ \ldots\ \omega_K]$$

矩阵指数为：

$$-j * t' * \omega$$

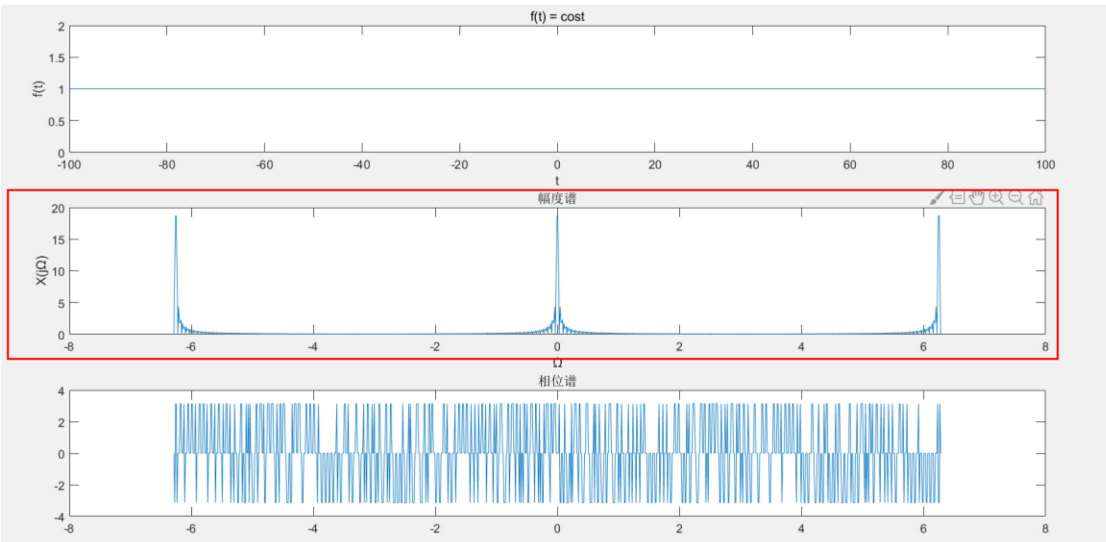
结果如下：



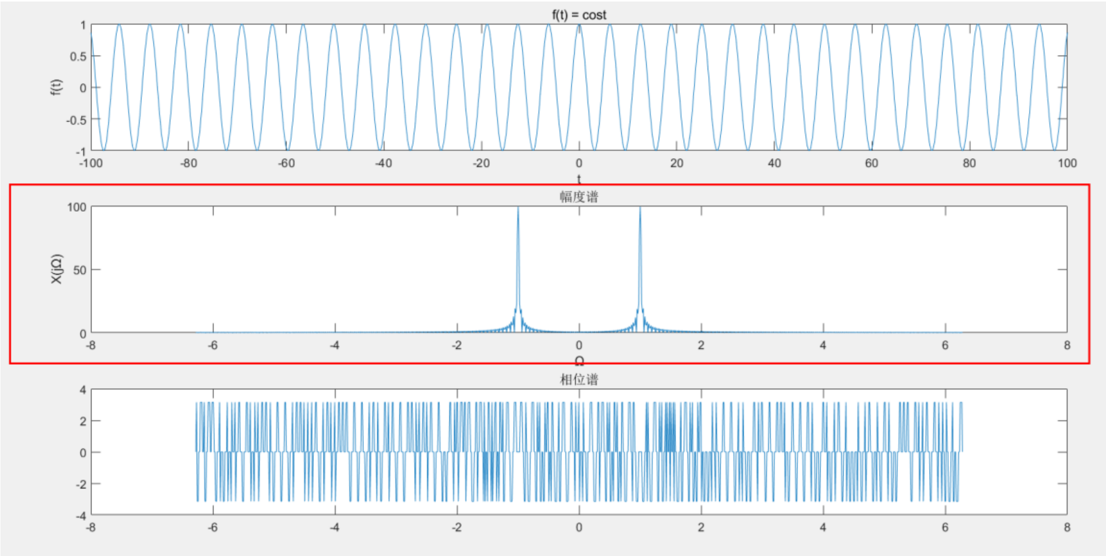

为了验证傅里叶变换过程的正确性，我使用了两个常用的时域函数作为测试函数来验证结果的正确性：

验证：

1> 计算 y = 1 的傅里叶变换是正确的（频谱的正确结果应表现为单位脉冲的周期扩展）




2> 计算 y = cost 的傅里叶变换的结果也是正确的（正确结果应为



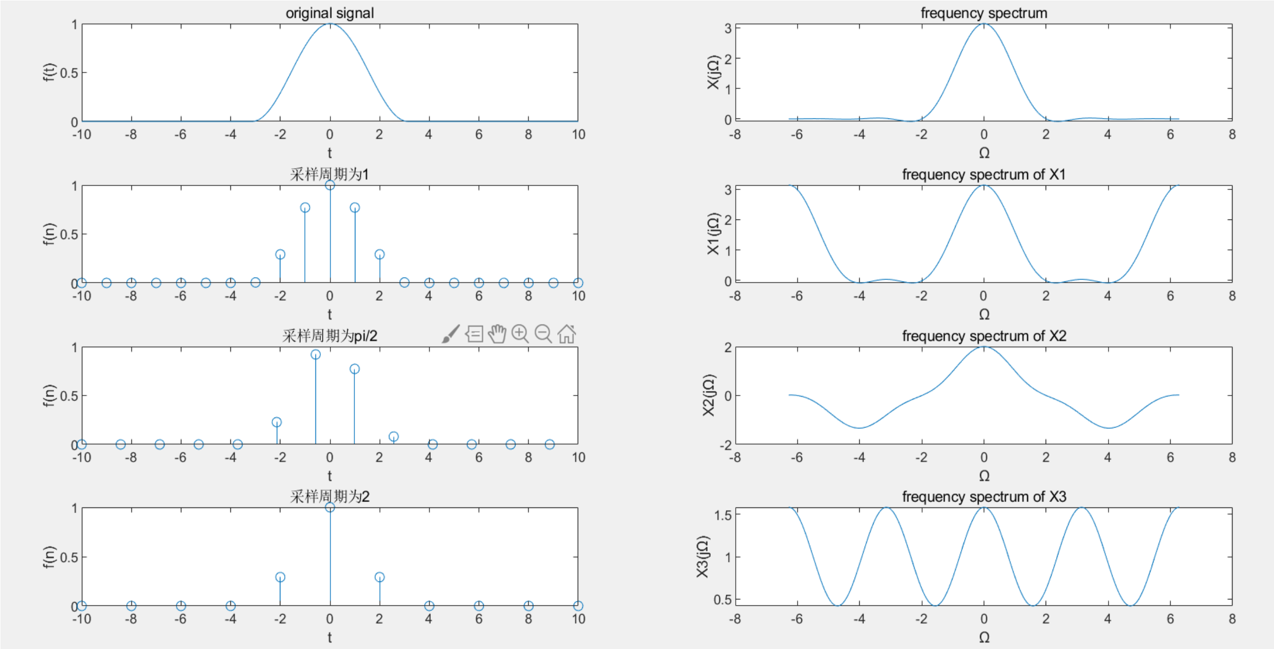

（注意上面的脉冲的最高值不是 1 的原因是因为在计算机上无法实现 t 的取值范围是负无穷到正无穷，增大 t 的范围可以进一步减少误差）

Matlab 源码如下：

```
Dt = 0.1;
t = -10:Dt:10;
f = 0.*(t < -pi) + 0.5*(1 + cos(t)).*(t >= -pi & t <= pi) + 0.*(t > pi); % 分段函数，这里无法表示负无穷到正无穷所以用-10 到 10 近似
subplot(2,1,1);
plot(t,f);
xlabel('t');
ylabel('f(t)');
title('f(t) = 1/2 * (1 + cos(t) )');
w = linspace(-2*pi,2*pi,1000); % [-2pi,2pi]之间的频率区间分割为 1000 份
%采样的最高频率不能高于 4*pi（奈奎斯特率）
%采样周期不能小于 1/(4*pi)，上面的采样周期是 0.1，满足条件；
X = f*exp(-1j*t'*w)*Dt; %傅里叶变换
subplot(2,1,2);
plot(w,X);
xlabel('Ω');
ylabel('X(jΩ)');
title('frequency spectrum');
```

(2) 第二小题要求对于采样周期分别为 1，π/2，2 时，画出其采样信号以及对应的频谱。这里依旧设定时域的范围是[-10,10]，且只画出 1000 个频率值，因为采样在上次作业做过，这里不再赘述(代码注释也写得很清楚)，频谱的画法同

(1)，结果如下：




结果分析：

首先明确一点，采样信号在频域上的表现是初始信号频域上图像的左右周期扩展，观察采样周期为 1 的采样信号，可以看到得到了很好的体现。但是对于采样周期为π/2 和 2 的采样信号来说，频域上某个周期内的图像好像和初始信号在频域上的差别很大，这是因为其采样频率过低，无法很好地恢复出初始信号。根据采样定理，采样频率必须至少为被采样信号的最高频率的 2 倍时，采样之后的信号才能不失真。

Matlab 源码如下：

```
%在整个程序中,w、Dt 和 t 的值都是不变的
          % w 表示采样的频率
          % t 是为了模拟连续时间信号取的时间序列
          % Dt 代表时间序列 t 任意两个相邻时间内的时间间隔
Dt = 0.1;
t = -10:Dt:10;
```

```
f = 0.*(t < -pi) + 0.5*(1 + cos(t)).*(t >= -pi & t <= pi) + 0.*(t > pi); % 分段函数，这里无法表示负无穷到正无穷所以用-10 到 10 近似
%以上过程为建立原始信号
subplot(4,2,1);
plot(t,f);
xlabel('t');
ylabel('f(t)');
title('original signal');
%绘制原始信号的图像
w = linspace(-2*pi,2*pi,1000); % [-2pi,2pi]之间的频率区间分割为 1000 份
X = f*exp(-1j*t'*w)*Dt;
subplot(4,2,2);
plot(w,X);
xlabel('Ω');
ylabel('X(jΩ)');
title('frequency spectrum');

T1 = 1; %采样周期为 1
dt1 = 1/T1; %采样间隔，在上面我们初始化原始信号的范围是-10 到 10;
tn1 = -10:dt1:10;
f1 = 0.*(tn1 < -pi) + 0.5*(1 + cos(tn1)).*(tn1 >= -pi & tn1 <= pi) + 0.*(tn1 > pi);
subplot(4,2,3);
stem(tn1,f1); %绘制采样信号图像
xlabel('t');
ylabel('f(n)');
title('采样周期为 1');
X1 = f1*exp(-1j*tn1'*w);
subplot(4,2,4);
plot(w,X1);
xlabel('Ω');
ylabel('X1(jΩ)');
title('frequency spectrum of X1');

T2 = pi/2; %采样周期为 pi/2
tn2 = -10:T2:10;
f2 = 0.*(tn2 < -pi) + 0.5*(1 + cos(tn2)).*(tn2 >= -pi & tn2 <= pi) + 0.*(tn2 > pi);
subplot(4,2,5);
stem(tn2,f2); %绘制采样信号图像
xlabel('t');
ylabel('f(n)');
title('采样周期为 pi/2');
```

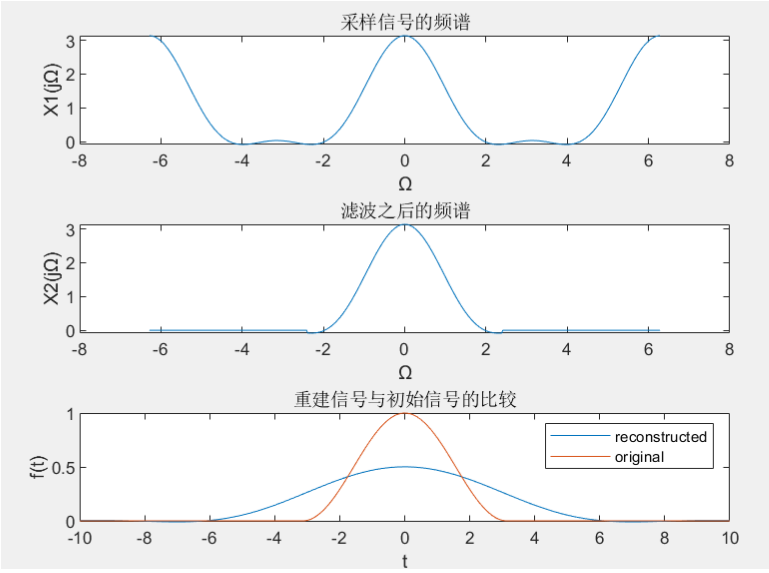
```
X2 = f2*exp(-1j*tn2'*w);
subplot(4,2,6);
plot(w,X2);
xlabel('Ω');
ylabel('X2(jΩ)');
title('frequency spectrum of X2');

T3 = 2; %采样周期为 2
tn3 = -10:T3:10;
f3 = 0.*(tn3 < -pi) + 0.5*(1 + cos(tn3)).*(tn3 >= -pi & tn3 <= pi) + 0.*(tn3 > pi);
subplot(4,2,7);
stem(tn3,f3); %绘制采样信号图像
xlabel('t');
ylabel('f(n)');
title('采样周期为 2');
X3 = f3*exp(-1j*tn3'*w);
subplot(4,2,8);
plot(w,X3);
xlabel('Ω');
ylabel('X3(jΩ)');
title('frequency spectrum of X3');
```

(4) 对于第三小题，要求将采样周期为 1、2 的采样信号分别通过一个截止频率为 w = 2.4 的低通滤波器，再将滤波之后的频域信号转化为时域信号，我的思路是对(2)得到的频域信号进行处理，将其大于 2.4 的频率部分设置为 0，再通过傅里叶逆变换的方式得到对应的时域信号。傅里叶逆变换的公式为：

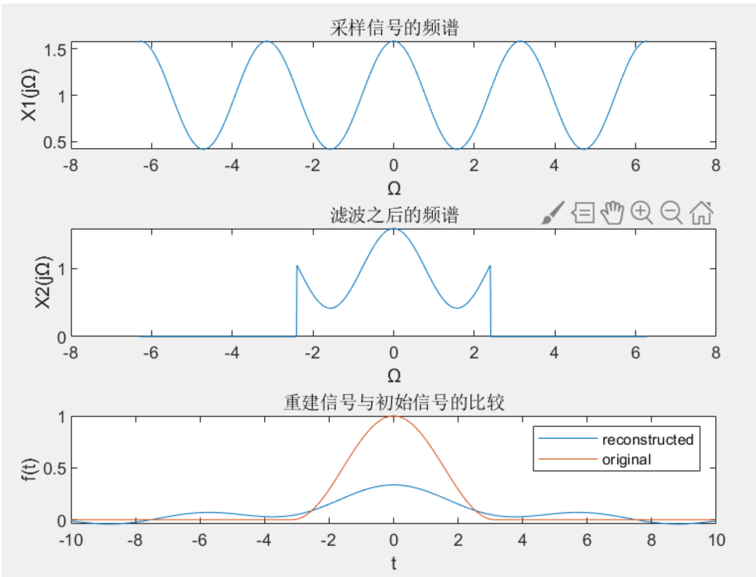
$$x(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(\mathrm{e}^{\mathrm{j}\omega})\mathrm{e}^{\mathrm{j}\omega n}d\omega \quad (-\infty<n<+\infty)$$

有关时间向量和矩阵系数的内容和傅里叶变换差不多，注意逆变换中是 j 不是-j，傅里叶逆变换结果如下：

采样周期为 1：




采样周期为 2：




：

结果分析：

由（2），采样周期为 1 的采样结果比采样周期为 2 的采样结果的误差更小。将采样之后的时域信号通过一个低通滤波器，使其频域信号查过 2.4 的频率成分被滤掉，则出现上面两幅图像中第二个曲线图的形状，可以看到采样周期为 1 的信号的频谱和初始信号的频谱还很接近，因此在恢复成时域信号之后，除了因为高频成分被滤掉而导致的时域信号看起来像是整体被拉宽的原因，时域信号的轮

廓大体上还是很相近的。但是对于采样周期为 2 的信号，通过低通滤波器之后的频谱已经和原始信号的频谱产生了很大的差别，所以在恢复成时域信号之后可以发现两条曲线的轮廓和走向已经有了比较大的不同。

Matlab functions potentially used：

plot；subplot；axis；exp；cos；sinc；ones；length；stem；abs

参考材料：

**连续时间信号傅立叶变换的数值计算**

为了更好地体会 MATLAB 的数值计算功能，这里给出连续信号傅立叶变换的数值计算方法。方法的理论依据为：

$$F(j\omega)=\int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt=\lim_{\tau\to 0}\sum_{n=-\infty}^{\infty} f(n\tau)e^{-j\omega n\tau}\tau \tag{1}$$

对于一大类信号，当取 $\tau$ 足够小时，上式的近似情况可以满足实际需要。若信号 $f(t)$ 是时限的，或当 $|t|$ 大于某个给定值时，$f(t)$ 的值已经衰减得很厉害，可以近似地看成时限信号，则式（1）中的 $n$ 取值就是有限的，设为 $N$，有：

$$F(k)=\tau\sum_{n=-\infty}^{N-1} f(n\tau)e^{-j\omega_k n\tau},0\le k\le N-1 \tag{2}$$

上式是对式（1）中的频率 $\omega$ 进行取样，通常：

$$\omega_k=\frac{2\pi}{N\tau}k \tag{3}$$

采用 MATLAB 实现式（2）时，其要点是要正确生成 $f(t)$ 的 $N$ 个样本 $f(n\tau)$ 的向量 $f$ 及向量 $e^{-j\omega_k n\tau}$，两向量的内积（即两矩阵相乘）的结果即完成式（2）的计算。

此外，还要注意取样间隔 $\tau$ 的确定。其依据是 $\tau$ 需小于奈奎斯特取样间隔。如果对于某个信号 $f(t)$，它不是严格的带限信号，则可根据实际计算的精度要求来确定一个适当的频率 $\omega_0$ 为信号的带宽。