

# 信号与系统课程设计

---

## 信号与系统课程设计

### 实验题目

#### 画出原信号 $f(t)$ 的波形和频谱

##### 理论分析

##### 完善代码

##### 扩展

#### 不同周期的采样信号的波形和频谱

##### 理论分析

##### 完善代码

##### 扩展

##### 解释

#### 重建信号

##### 理论分析

##### 完善代码

##### 解释

### 总结

### 总代码

## 实验题目

---

使用 `Matlab` 或者其它软件编写程序完成以下题目：

给定一个连续时间信号：

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

(1)画出这个信号的波形和它的频谱。

(2)当采样周期分别满足 $T=1$ ,  $T=\pi/2$ ,  $T=2$ 时, 分别画出三个采样信号 $f_p(n)$ 和他们各自的频谱, 并对结果给出解释。

(3)使用截止频率 $\omega_c = 2.4$ 的理想低通滤波器从 $f_p(n)$ 重建信号 $f_r(t)$ 。当采样周期分别是 $T=1$ 和 $T=2$ 时, 画出重建信号 $f_r(t)$ 及其频谱, 并画出 $f_r(t)$ 和原始信号 $f(t)$ 之间的绝对误差, 并对结果给出解释。

可能用到的 `Matlab` 函数：

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

## 画出原信号 $f(t)$ 的波形和频谱

---

### 理论分析

$f(t)$ 是一个连续非周期信号，而计算机中没有“连续”的概念，计算机是通过极高的精度来模拟连续信号的，所以要求 $f(t)$ 的波形，只需在时间轴上按较小的时间间隔依次取得对应的信号值即可。

因为 $f(t)$ 的自变量 $t$ 是分段的，Matlab中有两种方法来得出 $f(t)$ ：第一种方法是通过指令将一个数组的某些元素改写为需要的值；第二种方法是从数学角度出发，将 $f(t)$ 看作几个基本的函数的组合。第一种方法容易想到但可扩性不够，第二种方法可扩性较强且更加便于思考，这里采用第二种方法：

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

$$f(t) = A(t) \cdot B(t) \quad (2)$$

$$A(t) = \frac{1}{2}[1 + \cos(t)], -\infty < t < \infty \quad (3)$$

$$B(t) = \begin{cases} 1, & 0 \leq |t| \leq \pi \\ 0, & |t| > \pi \end{cases} \quad (4)$$

于是我们就可以通过一个连续周期信号 $A(t)$ 和一个矩形方波 $B(t)$ 来描述 $f(t)$ ，并且它们的频谱都是有一般结论的。由第四章相关结论得以下关系式：

$$\therefore X_A(j\omega) = \frac{\pi}{2}[\delta(\omega - 1) + \delta(\omega + 1)] + \pi\delta(\omega) \quad (2)$$

$$X_B(j\omega) = 2\pi \text{Sinc}\left(\frac{\omega\pi}{\pi}\right) \quad (3)$$

$$F(j\omega) = \frac{1}{2\pi} X_A(j\omega) * X_B(j\omega) \quad (4)$$

$$\therefore F(j\omega) = \frac{\pi}{2}[\text{Sinc}(\omega - 1) + \text{Sinc}(\omega + 1)] + \pi\text{Sinc}(\omega) \quad (5)$$

根据(5)式可以画出 $f(t)$ 的理想频谱（注意，是理论上的，实际应该根据(6)式来得到实际频谱），但我们可以比较这两种做法验证我们结果的正确性。

$$F(j\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t} dt \quad (6)$$

## 完善代码

$B(t)$ 函数表达式的描述可以通过一个简单的技巧来得到：

```
B = abs(t) <= pi
```

进一步得到求 $f(t)$ 的函数值序列的代码：

```
t = -2*pi:0.01:2*pi;
f = (abs(t)<=pi) * 0.5 .* (1+cos(t));
%画图
subplot(2,1,1),plot(t,f);
axis([min(t),max(t),-0.5,1.1*max(f)]);
title('原连续信号波形');
xlabel('t');
ylabel('f(t)');
grid on
```

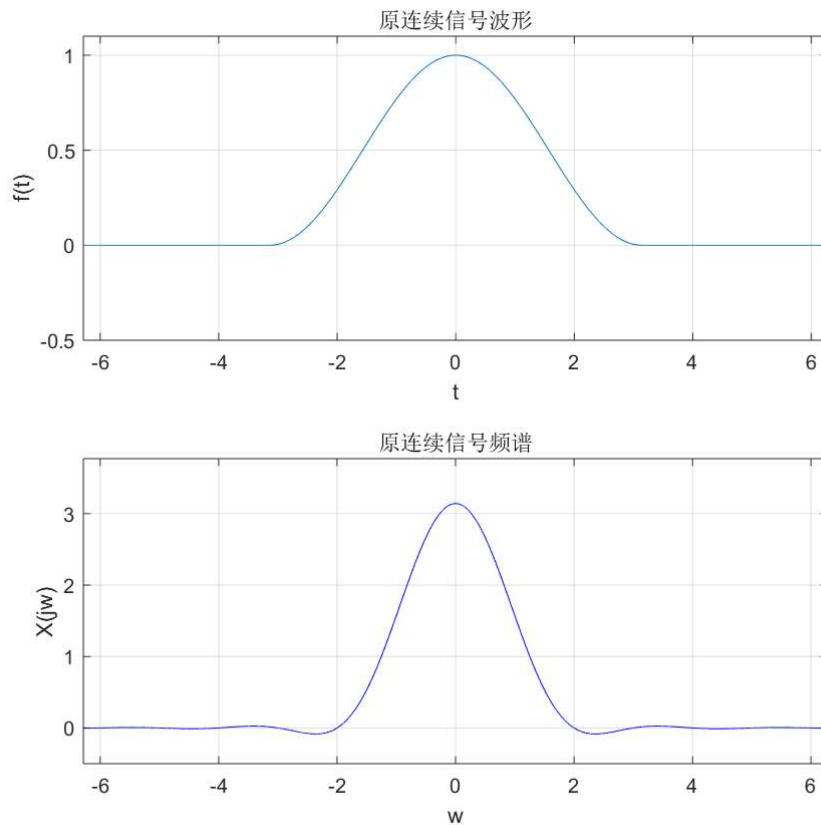
这是求 $f(t)$ 实际频谱的代码：

```

w = -2*pi:0.001:2*pi;
F_D = f * exp(-1j * t' * w)*dt;
subplot(2,1,2),plot(w,F_D, 'b');
axis([min(w),max(w),-0.5,1.2 * max(abs(F_D))]);
grid on
title('原连续信号频谱');
ylabel('X(jw)');
xlabel('w');

```

得到:



图(1) 上面第一幅图是连续信号的波形，而第二幅图是连续信号的频谱

## 扩展

使用(5)式我们得到了上面原连续信号的频谱，在代码中是根据 $f(t)$ 的间隔较小的函数值序列得到的。但实际上也可以根据(5)式得到原连续信号的理想频谱，称其“理论”是因为其用到了 $f(t)$ 的所有函数值，而这在现实生活中是做不到的。

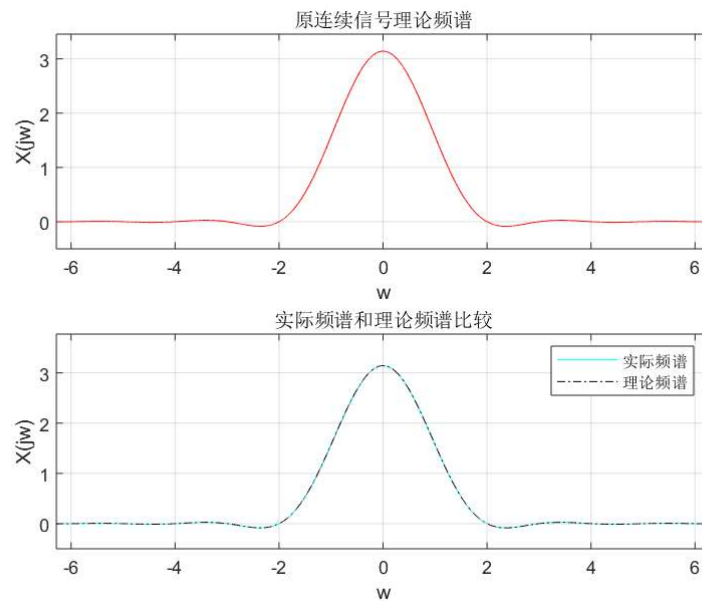
这是求原连续信号的理论频谱的代码：

```

w = -2*pi:0.001:2*pi;
fs = 0.5 * pi .* (sinc(w - 1) + sinc(w + 1)) + pi .* sinc(w);
%画图
subplot(2,1,1),plot(w,fs, 'r-');
axis([min(w),max(w),-0.5,1.1*max(fs)]);
title('频谱');
title('原连续信号理论频谱');

```

我还将理论频谱和实际频谱画在了一个图上，可以发现很大程度上的重合。



图(2) 上面第一幅图是连续信号的理论频谱，第二幅图是理论频谱和实际频谱的比较

## 不同周期的采样信号的波形和频谱

### 理论分析

采样信号的波形很好绘制，和第一问中一样，在时间轴上按规定的時間间隔依次取得 $f(t)$ 对应的信号值即可。从这个角度来看，在计算机中，无论是连续信号还是其采样信号，归根到底都是采样——前者时间间隔很小，人的肉眼便认为其实连续的。

但是采样就遇到问题了：这里的连续信号是时限的，所以其频谱是非带限的，这与我们学的冲激串采样的前提条件不合，所以使用冲激串采样一定会导致频谱的混叠。但这里的特别之处是，频谱在 $|w|>2$ 左右就基本为0了，我们就可以将其看作带限于最高频率 $w_m=2$ 。这就意味着我们将其看作带限信号影响是不大的——除非不满足采样频率 $w_s>2w_m$ 。

因此，接下来就用冲激串采样的思想将原连续信号（即 $f(t)$ ）采样，得到离散信号 $f_p(n)$ ，显然满足 $f_p[n]=f(nT)$

我们可以使用(7)式求得频谱。

$$F_p(e^{jw}) = \sum_{n=-\infty}^{+\infty} f_p[n]e^{-jwn} \quad (7)$$

### 完善代码

我们可以用一个简单的技巧得到时间轴上以0为起点，间隔为T的时间序列

```
t = -2*pi:T(i):2*pi; %先随便得到一个等间隔的时间序列
nt = length(t); %获得序列长度
t0 = t(fix(nt/2)+1); %取得靠近0的时间节点
t = t-t0; %将整个时间序列偏移 to 正确位置
```

采样信号的完整波形代码：

```

T=[1,pi/2,2]; %使用数组存储不同采样周期
for i=1:3
    t = -2*pi:T(i):2*pi; %先随便得到一个等间隔的时间序列
    nt = length(t); %获得序列长度
    t0 = t(fix(nt/2)+1); %取得靠近0的时间节点
    t = t-t0; %将整个时间序列偏移到正确位置
    n = t/T(i); %n便是下标序列
    sample = (abs(t)<=pi) * 0.5 .* (1+cos(t)); %采样
    %绘制
    subplot(3,3,3*i-2),plot(t,sample);
    axis([min(t),max(t),-0.5,1.1*max(sample)]);
    grid on
    title('采样信号波形');
    xlabel('t');
    ylabel('fp(t)');
    %.....
end

```

根据(7)式求频谱：

```

f_D = (abs(t)<=pi) * 0.5 .* (1+cos(t));
F_D = f_D * exp(-1j * t * w);

subplot(3,3,3*i-1),plot(w,abs(F_D));
axis([min(w),max(w),-0.5,1.1*max(abs(F_D))]);
title('采样信号实际频谱');
grid on
xlabel('w');
ylabel('x(jw)');

```

## 扩展

同样的，我们可以直接理论分析出各个采样信号的频谱，根据第七章相关知识：

$$F_p(jw) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} F(j(w - kw_s)), w_s = \frac{2\pi}{T} \quad (8)$$

于是我们可以写出这样的代码：

```

w = -6:0.01:6; %w为自变量，间隔取小一点图形会更丝滑
N = length(w);
sample_fs = zeros(1,N); %为求和做准备
ws = 2*pi/T(i);
for k = -20:20 %k不必取到无穷，取到一个较大的数使得w-k*ws落到[-
wm,wm]即可
    sample_fs = F_fs(w-k*ws) + sample_fs;
end %F_fs是傅里叶变化的函数
sample_fs = 1/T(i) .* sample_fs; %乘以1/T
%绘图
subplot(3,3,3*i),plot(w,sample_fs);
axis([min(w),max(w),-0.5,1.1*max(sample_fs)]);
title('采样信号理论频谱');
grid on
xlabel('w');

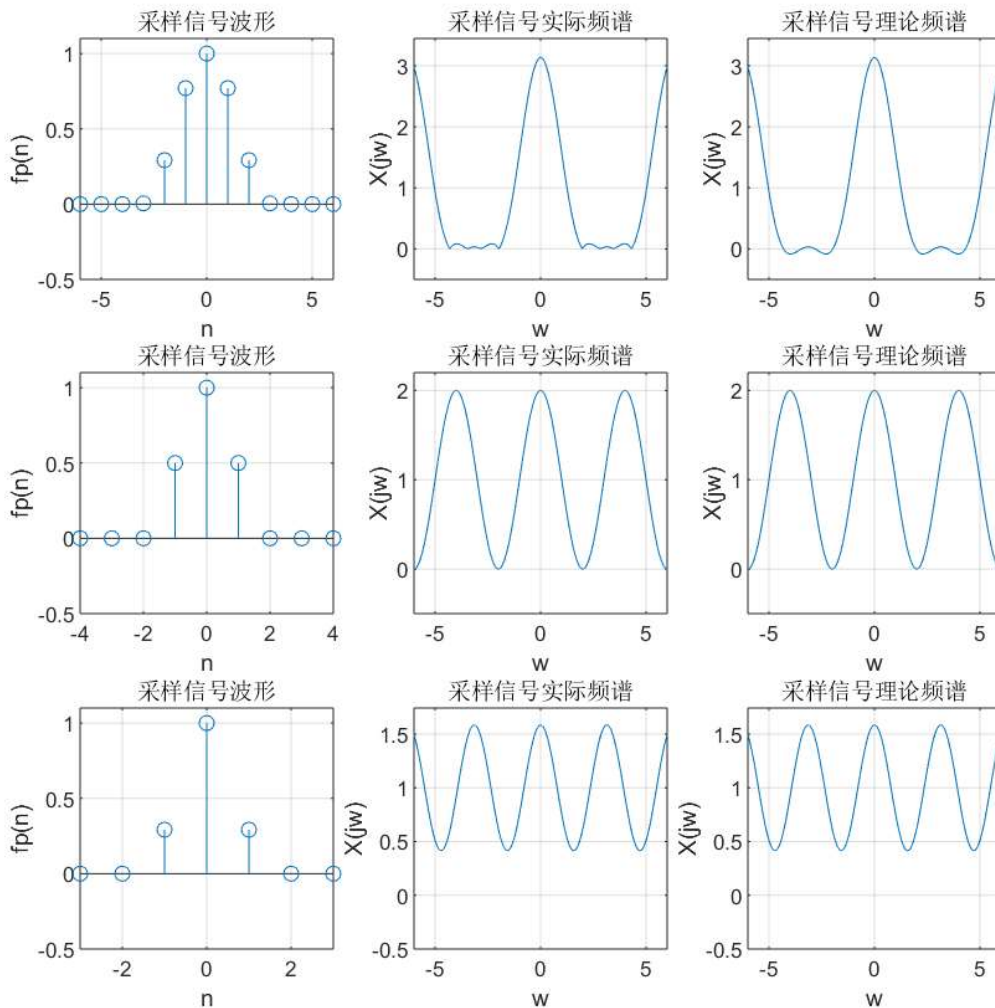
```

```
ylabel('X(jw)');
```

$F_{fs}$ 是傅里叶变化函数，定义如下：

```
function F_fs = F_fs(w)
    F_fs = 0.5 * pi .* (sinc(w - 1) + sinc(w + 1)) + pi .* sinc(w);
end
```

于是可得：



图(3)从上到下每行分别是 $T = 1, \pi/2, 2$ 的离散采样信号波形，实际频谱以及理想频谱

## 解释

采样信号的频谱就是原信号的频谱以 $\omega_s$ 为周期进行延拓。根据不同的采样间隔 $T$ 得到不同的 $\omega_s$ ，只有当采样频率 $\omega_s > 2\omega_m$ 才不会有混叠。但正如理论分析中提到的，在这里使用冲激串采样一定会导致频谱的混叠，但是程度有差异。

$T_1 = 1, T_2 = \pi/2, T_3 = 2$ 对应的 $\omega_s$ 分别是 $\omega_{s1} = 2\pi, \omega_{s2} = 4, \omega_{s3} = \pi$ 。可见 $\omega_{s1} > 2\omega_m, \omega_{s2} = 2\omega_m, \omega_{s3} < 2\omega_m$ ，分别是过采样，临界采样，欠采样。图(3)中也可看出第一二三行的频谱均是原频谱的周期延拓，但只有第三行发生了明显混叠。

此外，也可从图(3)中观察到采样信号的频谱的最高处从上至下从3减到1.5，这是由于采样时频谱幅度减小了，且减小程度是原来连续性好频谱的 $1/T$ ，因此，在接下来的内插恢复信号中滤波器要有 $T$ 倍的通带增益。

## 重建信号

### 理论分析

由Nyquist定理知 $T=1$ 可以基本从采样信号确定原信号，但 $T=2$ 不满足前提条件，可能不行。这里用理想内插的方法重建信号。根据第七章相关内容：

$$f_r(t) = f_p(t) * h(t) = \sum_{n=-\infty}^{+\infty} f_p[n] \delta(t - nT) * h(t) = \sum_{n=-\infty}^{+\infty} f_p[n] h(t - nT) \quad (9)$$

$$f_p[n] = f(nT) \quad (a)$$

$$h(t) = \frac{T\omega_c}{\pi} \text{Sinc} \frac{\omega_c t}{\pi}, \omega_c = 2.4 \quad (b)$$

这样便可以得到重建信号的波形，再使用(6)式即可类似第一问中的得到重建信号的频谱。我们还可以在这基础上画出滤波器的频谱。

### 完善代码

据此可写出截止频率为2.4的理想滤波器的代码：

```
function x = imfilter(t)
    x = (2.4/pi)*sinc(2.4*t/pi);%这里方便起见未有乘以增益T(i)，将在以下代码中
    补上
end
```

求重建信号的波形的代码：

```
for i=1:2
    nT = -2*pi:T(i):2*pi;    %先随便得到一个等间隔的时间序列
    n = length(nT);          %获得序列长度
    t0 = nT(fix(n/2)+1);      %取得靠近0的时间节点
    nT = nT-t0;               %将整个时间序列偏移到正确位置
    f_nT = (abs(nT)<=pi) * 0.5 .* (1+cos(nT));    %抽样
    T_N = ones(length(nT),1)*t - nT' * ones(1,length(t));    %t_nT

    y = T(i)*f_nT*imfilter(T_N);    %得到fr(t)
    %滤波器要有T倍的通带增益

    %画图
    subplot(4,2,2+i),plot(t,y);
    axis([min(t),max(t),-0.5,1.1*max(y)]);
    grid on
    title('重建信号波形');
    xlabel('t');
    ylabel('fr(t)');
```

求重建信号的频谱和滤波器的频谱的代码：

```
Y_D = y * exp(-1j * t' * w)*dt; %按照连续信号求频谱

I_D = T(i)& imfilter(t) * exp(-1j * t' * w)*dt; %按照连续信号求频谱
                                         %滤波器要有T倍的通带增益

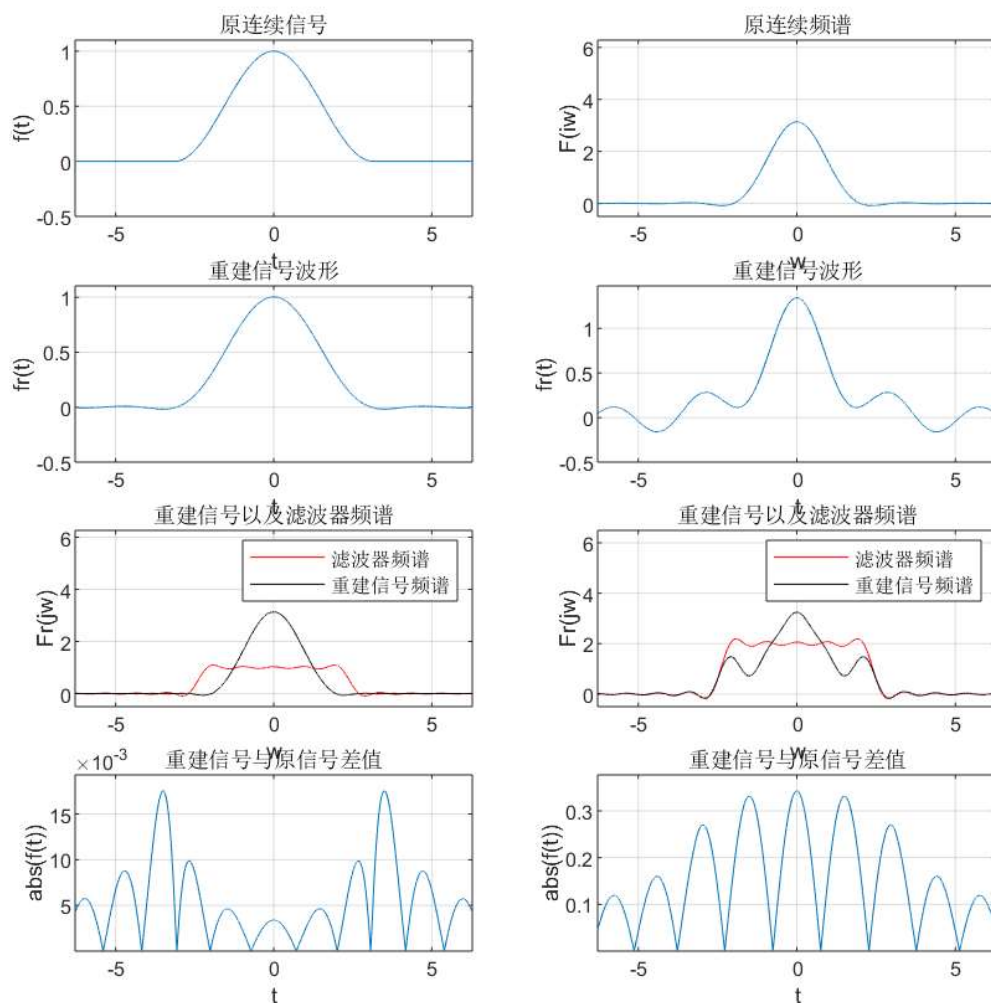
%画图
subplot(4,2,i+4),plot(w,I_D, '-r',w,Y_D, '-k');
h = legend('滤波器频谱','重建信号频谱');
axis([min(w),max(w),-0.5,2*max(abs(Y_D))]);
grid on
title('重建信号以及滤波器频谱');
xlabel('w');
ylabel('Fr(jw)');
```

绝对误差：

```
f_y = f-y; %差值
subplot(4,2,i+6),plot(t,abs(f_y));
axis([min(t),max(t),0.8*min(abs(f_y)),1.1*max(abs(f_y))]);
grid on
title('重建信号与原信号差值');
xlabel('t');
ylabel('abs(t)');
end
```

得到：





图(4)第一行一二列是原信号的波形和频谱，第二行的一二列分别是 $T=1, T=2$ 的重建信号波形，第三行的一二列分别是 $T=1, T=2$ 的重建信号频谱和滤波器频谱，第四行的一二列分别是 $T=1, T=2$ 的重建信号与原信号的绝对差值。

## 解释

由图四第四行可看出 $T=1$ 的重建信号的波形图对原连续信号的拟合远远好于 $T=2$ 的重建信号(注意纵坐标单位)。这是因为 $T=1$ 时基本满足Nyquist前提条件，于是可以基本从采样信号恢复原信号，但 $T=2$ 不满足前提条件 ( $\omega_s < 2\omega_m$ )，所以可能无法从采样信号恢复原信号。其本质原因是 $T=1$ 时频谱混叠程度基本没有，而 $T=2$ 时频谱混叠程度较大。这样 $T=2$ 的对应频谱被截取 $|\omega| > \omega_c = 2.4$ 时留下的部分和原连续信号的差距过大，这样恢复出来的信号波形与原连续信号波形就有较大区别。

由图四第三行可看出 $T=1, T=2$ 的频谱的幅度最大值都接近于3，而采样信号中 $T=2$ 时频谱幅度最大处越为1.5。这是因为滤波器带了 $T$ 倍的通带增益。

由图四第二行和之前图三采样信号的频谱可看出 $T=1, T=2$ 的重建信号频谱就是图三采样信号的频谱的一个截取。范围理论上是 $[-2.4, 2.4]$ ，但实际上工程上无法实现理想滤波器的频谱，所以在 $T=2$ 的频谱中，边界处的频谱是与理想滤波器的频谱重合，这意味着我们得到的频谱不完全是图三采样信号的频谱在 $[-2.4, 2.4]$ 的一个截取，实际上滤波器的频谱是影响了 $T=2$ 的频谱的。但 $T=1$ 的情况则不同，其采样信号频谱在 $[-2.4, -2]$ 和 $[2, 2.4]$ 的范围上就已经是接近0了，所以我们滤波器的频谱在由幅度 $T(i)$ 变为0的变化边界处并未影响到原频谱，这意味着这个情况下我们得到的波形图会非常的符合原信号。因此得出的结论是，在实际生活中，应该采取**过采样**的方法，这不仅保证了Nyquist定理的前提，也消除了实际滤波器带来的误差。同样的，**临界采样**也会因实际滤波器而具有更大的误差，所以虽然理论上临界采样是可行的，但实际生活中是不行可行的。

## 总结

看上去较为简单的内容(低于作业题难度)时实践起来才发现有很多的坑——比如离散信号的自变量为  $n$ ，这就要求从连续信号的自变量  $t$  的采样点要包括 0 以保证  $t=nT$ ，否则就不符合公式。但实际上，任意的符合间隔为  $T$  的  $t$  序列得出的采样信号频谱和最终的重建信号都差不多，具体原因我还不知道，但这说明离散傅里叶变化和连续傅里叶变化的在实际应用中的关系远比我想象的要密切。

## 总代码

chg\_1(第1问)

```
clear
dt = 0.01;
t = -2*pi:0.01:2*pi;
f = (abs(t)<=pi) * 0.5 .* (1+cos(t));
% 画图
% subplot(2,1,1),plot(t,f); •
% axis([min(t),max(t),-0.5,1.1*max(f)]);
% title('原连续信号波形');
% xlabel('t');
% ylabel('f(t)');
% grid on
w = -2*pi:0.001:2*pi;
fs = 0.5 * pi .* (sinc(w - 1) + sinc(w + 1)) + pi .* sinc(w);
%画图
subplot(2,1,1),plot(w,fs,'r-'); •
axis([min(w),max(w),-0.5,1.1*max(fs)]);
title('频谱');
title('原连续信号理论频谱');
grid on
ylabel('X(jw)');
xlabel('w');

F_D = f * exp(-1j * t' * w)*dt;
subplot(2,1,2),plot(w,F_D,'-c',w,fs,'-.k');
h = legend('实际频谱','理论频谱');
axis([min(w),max(w),-0.5,1.2 * max(abs(F_D))]);
grid on
title('实际频谱和理论频谱比较');
ylabel('X(jw)');
xlabel('w');
```

chg\_2(第2问)

```
clear
T=[1,pi/2,2];
for i=1:3
t = -2*pi:T(i):2*pi;
nt = length(t);
```

```

t0 = t(fix(nt/2)+1);
t = t-t0;
n = t/T(i);
sample = (abs(t)<=pi) * 0.5 .* (1+cos(t));
subplot(3,3,3*i-2),stem(n,sample);
axis([min(n),max(n),-0.5,1.1*max(sample)]);
grid on
title('采样信号波形');
xlabel('n');
ylabel('fp(n)');

w = -6:0.01:6;
N = length(w);
sample_fs = zeros(1,N);
ws = 2*pi/T(i);
    for k = -20:20
        sample_fs = F_fs(w-k*ws) + sample_fs;
    end
sample_fs = 1/T(i).* sample_fs;
subplot(3,3,3*i),plot(w,sample_fs);
axis([min(w),max(w),-0.5,1.1*max(sample_fs)]);
title('采样信号理论频谱');
grid on
xlabel('w');
ylabel('X(jw)');

F_D = sample * exp(-1j * t' * w);

subplot(3,3,3*i-1),plot(w,abs(F_D));
axis([min(w),max(w),-0.5,1.1*max(abs(F_D))]);
title('采样信号实际频谱');
grid on
xlabel('w');
ylabel('X(jw)');
end

```

chg\_3(第3问)

```

clear
T=[1,2];

dt = 0.001;
t = -2*pi:0.001:2*pi;
f = (abs(t)<=pi) * 0.5 .* (1+cos(t));
%画图
subplot(4,2,1),plot(t,f);•
axis([min(t),max(t),-0.5,1.1*max(f)]);
title('原连续信号');
grid on
xlabel('t');
ylabel('f(t)');

```

```

w = -2*pi:0.001:2*pi;
F_D = f * exp(-1j * t' * w)*dt;
subplot(4,2,2),plot(w,F_D);
axis([min(w),max(w),-0.5,2*max(abs(F_D))]);
title('原连续频谱');
grid on
xlabel('w');
ylabel('F(iw)');

for i=1:2
nT = -2*pi:T(i):2*pi;
n = length(nT);
t0 = nT(fix(n/2)+1);
nT = nT-t0;
f_nT = (abs(nT)<=pi) * 0.5 .* (1+cos(nT));
T_N = ones(length(nT),1)*t - nT' * ones(1,length(t));
y = T(i)*f_nT*imfilter(T_N);
%画图
subplot(4,2,2+i),plot(t,y);•
axis([min(t),max(t),-0.5,1.1*max(y)]);

grid on
title('重建信号波形');
xlabel('t');
ylabel('fr(t)');

Y_D = y * exp(-1j * t' * w)*dt;

I_D = imfilter(t) * exp(-1j * t' * w)*dt;
subplot(4,2,i+4),plot(w,I_D,'-r',w,Y_D,'-k');
h = legend('滤波器频谱','重建信号频谱');
axis([min(w),max(w),-0.5,2*max(abs(Y_D))]);
grid on
title('重建信号以及滤波器频谱');
xlabel('w');
ylabel('Fr(jw)');

f_y = f-y;
subplot(4,2,i+6),plot(t,abs(f_y));•
axis([min(t),max(t),0.8*min(abs(f_y)),1.1*max(abs(f_y))]);
grid on
title('重建信号与原信号差值');
xlabel('t');
ylabel('abs(t)');

end

```

函数

```

function F_fs = F_fs(w)
F_fs = 0.5 * pi .* (sinc(w - 1) + sinc(w + 1)) + pi .* sinc(w);
end

```

```
function x = imfilter(t)
    x = (2.4/pi)*sinc(2.4*t/pi);
end
```