附件 4

华南理工大学

# 实 验 报 告

课程名称：　信号与系统实验

学生姓名：

学生学号：

学生专业：

开课学期：　2019-2020 学年度第二学期

**电子与信息学院**

# 实 验 二　 利用 DFT 分析离散、连续信号频谱

**地　　　　点：**＿＿＿＿＿＿＿＿＿＿＿＿ **实验台号：**＿＿＿＿＿

**实验日期与时间：**＿＿＿＿＿＿＿＿＿＿＿＿ **评　　分：**＿＿＿＿＿

**预 习 检 查 纪 录：**＿＿＿＿＿＿＿＿＿＿＿＿ **实验教师：**＿＿＿＿＿

## 一、实验目的

1.掌握特征函数在系统响应分析中的作用

2.正确理解滤波的概念。

## 二、实验内容

1.函数 Filter、Freqz 和 Freqs 的使用：2.2 节(g)、3.2 节、4.1 节

2.计算离散时间傅里叶级数：3.1 节

3.LTI 系统的特征函数：3.4 节(a),(b),(c)

4. 吉布斯现象：根据英文理论课教材 Example 3.5 验证 Fig3.9 的吉布斯现象(a)~(d)

## 三、实验细节

1. 2.2 节（g）定义一个包含在区间$0 \le n \le 10$上的$x[n]$向量$x2$，并利用

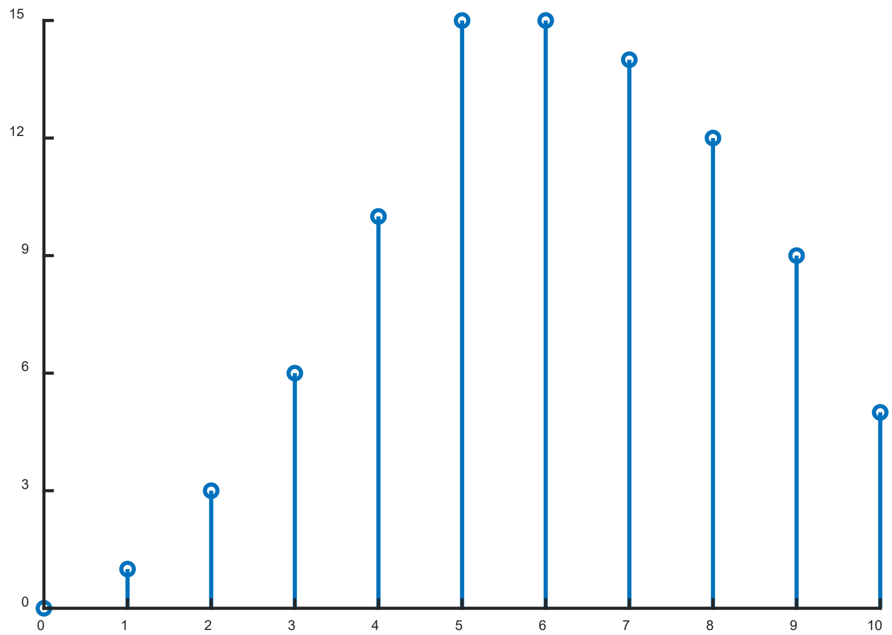
$$\gg y2 = filter(h, 1, x2);$$

计算这个区间内的卷积结果，利用$stem([0{:}10], y2)$画出这一结果，并确认与图 2.2 一致。

主程序：

```
n = 0:10;
x2 = n <= 5; % 定义x[n]
h = 0:5; % 定义h[n]
y2 = filter(h, 1, x2);
stem(n, y2,  'linewidth', 2)
```

实验结果：




利用y2=filter(h,1,x2)绘制的y[n]=x[n]*h[n]的图形

## 3.2 节：利用$freqz$计算 N 等分频率上的频率响应

主程序：

```
clear all; clc;
% 定义系数矩阵
b = [2, 0, -1];
a = [1, -0.8];
% 求在区间[0, π]之间 4 个等分频率上的频率响应
[H1, omega1] = freqz(b, a, 4);
% 求在区间[0, 2π]之间 4 个等分频率上的频率响应
[H2, omega2] = freqz(b, a, 4, 'whole');
```

实验结果：

| | | | | |
|---|---|---|---|---|
| **H1** | 5.0000 + 0.0000i | 2.8200 - 1.3705i | 1.8293 - 1.4634i | 0.9258 - 0.9732i |
| **omega1** | 0.0000 | 0.7854 | 1.5708 | 2.3562 |
| **H2** | 5.0000 + 0.0000i | 2.8200 - 1.3705i | 1.8293 - 1.4634i | 0.9258 - 0.9732i |
| **omega2** | 0.0000 | 0.7854 | 1.5708 | 2.3562 |

4.1 节：利用*freqs*绘制频率响应曲线。

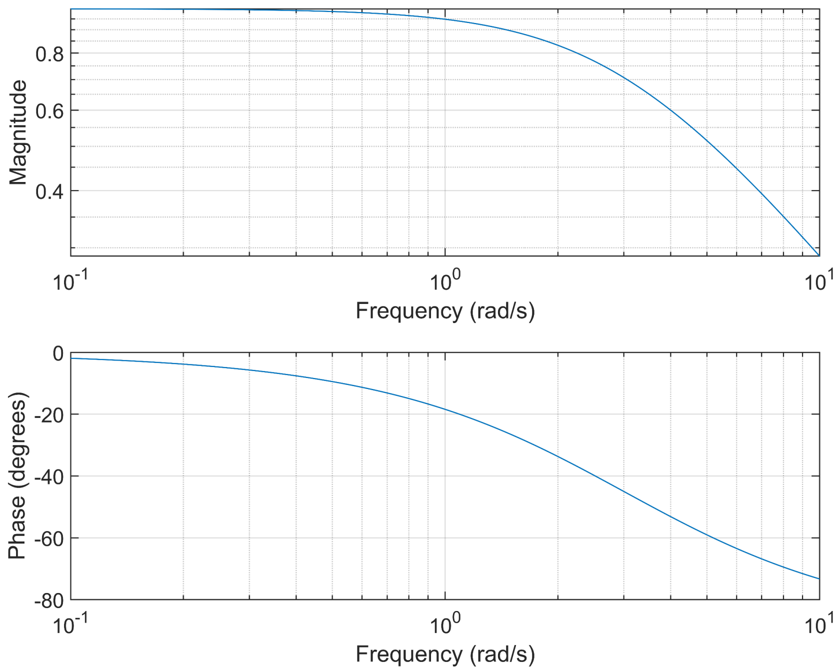
主程序：

```
clear all; clc;
a = [1, 3];
b = 3;
freqs(b, a)
```

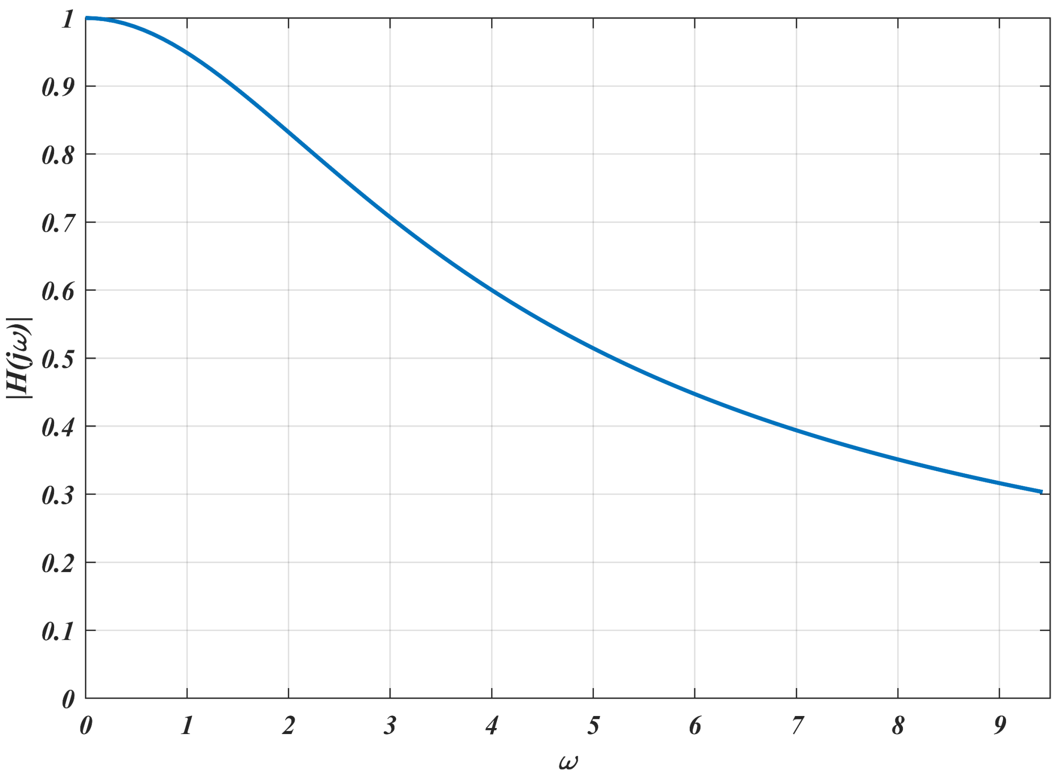
```
clear all; clc;
% 修改坐标轴位置以适应内容
axes('position', [0.1 0.15 0.85 0.80])

a = [1, 3];
b = 3;
w = linspace(0, 3 * pi);
H = freqs(b, a, w);
% 绘制 H 的幅值
pic2 = plot(w, abs(H));
% 添加标签和设置坐标轴范围
ylim([0, 1])
xlim([0, 9.5])
label1 = xlabel('\omega');
label2 = ylabel('|H(j\omega)|');
% 添加标题
title('一阶系统当在 w=linspace(0,3\pi)频率上求值时频率相应的幅值
|H(j\omega)|')

ax = gca;
% 设置文字
[label2.FontName, label1.FontName, ax.Font-
Name] = deal('Times New Roman');
[label2.FontSize, label1.FontSize] = deal(12);
[label2.FontWeight, label1.FontWeight, ax.FontWeight] = deal('Bold');
[label2.FontAngle, label1.FontAngle, ax.FontAngle] = deal('italic');
ax.FontSize = 10;
% 设置图线
pic2.LineWidth = 1.5;
```

```
%% 二阶系统频率响应
a = [3, 4, 1];
b = [1, 0, 5];
freqs(b, a);
```

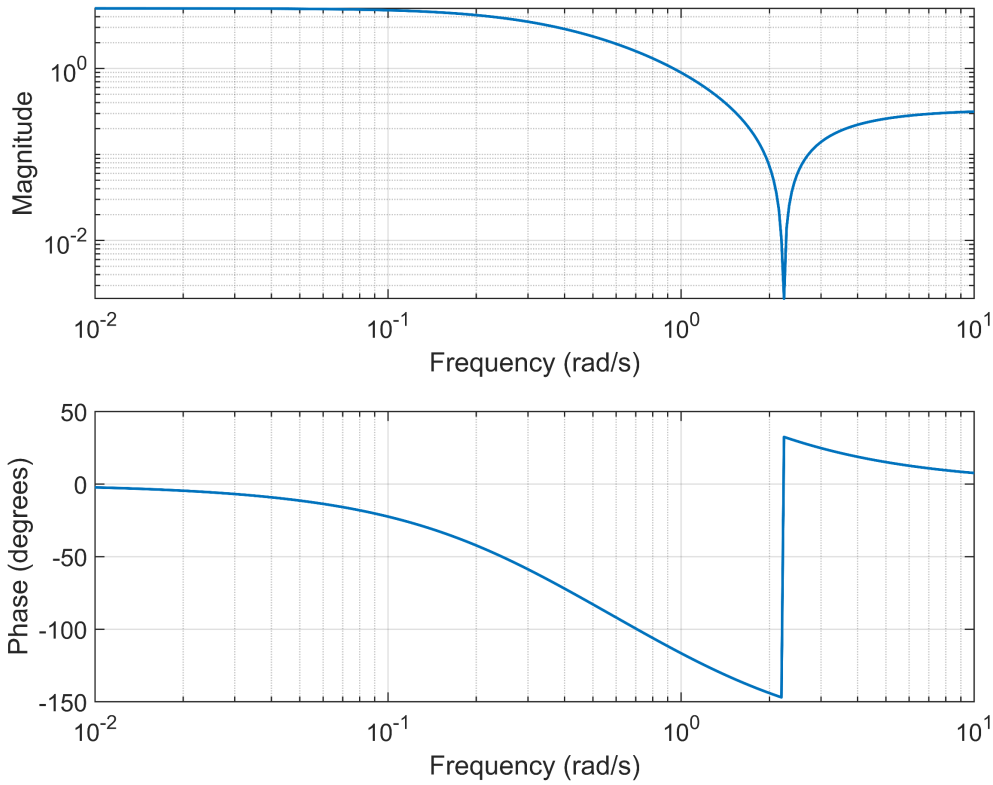
实验结果：




$\frac{dy(t)}{dt} + 3y(t) = 3x(t)$的频率相应曲线




一阶系统当在w=linspace(0,3π)频率上求值时频率相应的幅值|H(jω)|




$\frac{3\,d^2y(t)}{dt^2}+4\frac{dy(t)}{dt}+y(t)=\frac{d^2x}{dt}(t)+5x(t)$的频率相应曲线

实验小结：

函数 Filter 可以用于计算一个由差分方程表征的 LTI 系统的响应，此外也可以用于实现 conv 计算卷积的功能。*Freqz* 和 *Freqs* 都是用于求频率相应。*Freqz* 用于求解在$0\sim\pi$或者$0\sim2\pi$上 $N$ 等分点的频率相应，而 *Freqs* 可以用于描绘由微分方程表征的 LTI 系统的频率相应曲线。只需要输入系数矩阵，就能够很快地画出频率响应曲线，十分的方便。

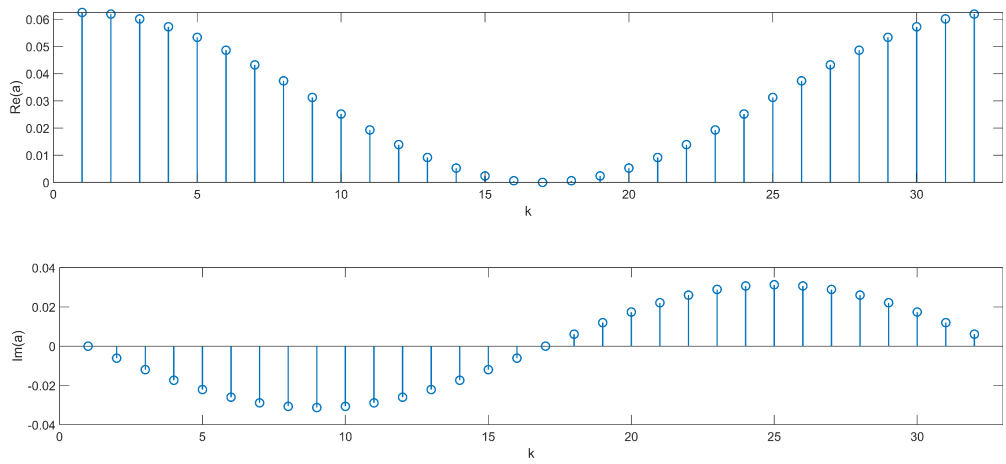
## 2. 3.1 节：利用 fft 算离散时间傅里叶级数

计算基波周期为 32，表达式为$x[n] = \begin{cases} 0, n = 0,1 \\ 1, else \end{cases} \quad 0 \le n \le 31$的信号的傅里叶级数系数。

主程序：

```
x = [1, 1, zeros(1, 30)]; % 定义 x[n]
a = (1 / length(x)) * fft(x); %求 DTFS
% 绘制 a 的实部
subplot(2, 1, 1)
stem(real(a), 'linewidth', 1) % 利用 stem 绘制对应数值，并设置线宽为1
xlabel('k')
ylabel('Re(a)')
xlim([0,33]) % 设置 x 轴的范围
% 绘制 a 的虚部
subplot(2, 1, 2)
stem(imag(a), 'linewidth', 1); % 利用 stem 绘制对应数值，并设置线宽为1
xlabel('k')
ylabel('Im(a)')
xlim([0,33]) % 设置 x 轴的范围
% 保存结果到'3-1-1.png'
print('-dpng', '-r1200', '3-1-1')
```

实验结果：




$a$ 的实部和虚部

实验小结：

在 Matlab 中，利用 *fft* 函数可以方便地算出 DTFS。但是在使用的过程中要记得乘上 1/N 才是正确的结果。

3. 离散时间 LTI 系统的特征函数

这个练习要检验离散时间 LTI 系统的特征函数性质。复指数是 LTI 系统的特征函数，也就是说当输入序列是某一复指数时，输出就是同样的复指数乘以一个复常数，这个常数可以由单位脉冲响应$h[n]$计算出来。当一个离散时间 LTI 系统的输入是$x[n] = z^n$时，输出就是$y[n] = H(z)z^n$式中

$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n}$$

现考虑下列各输入信号：

$$x_1[n] = e^{j(x/4)n}$$

$$x_2[n] = sin(\pi n \,/\, 8 + \pi \,/\, 16)$$

$$x_3[n] = (9/10)^n$$

$$x_4[n] = n + 1$$

当每个信号是由下面线性常系数差分方程：

$$y[n] - 0.25y[n-1] = x[n] + 0.9x[n-1]$$

描述的因果 LTI 系统的输入时，要计算输出$y_1[n]$~$y_4[n]$。

主程序：（为了报告代码的简洁，在写实验报告的时候删去了用于调整样式的代码）

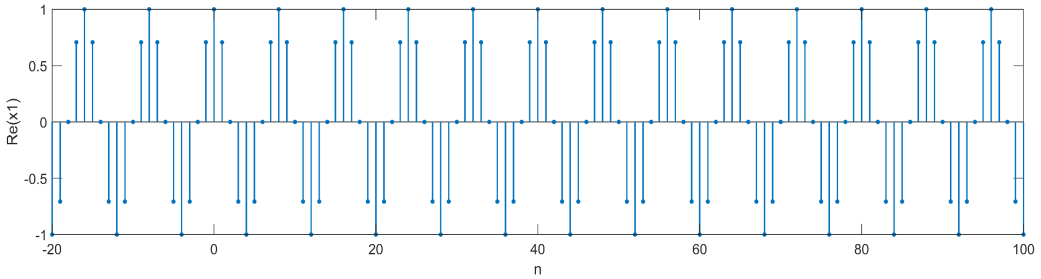
```
clear all; clc;
%% (a)内的基本操作
n = -20:100;  % 创建时间序号向量 n
% 定义4个信号在向量 n 区间内的值
x1 = exp(i * pi .* n / 4);
x2 = sin(pi * n / 8 + pi / 16);
x3 = 0.9.^n;
x4 = n + 1;

%% 对以上四个信号绘图

%绘制 x1的图形
% 绘制 x1的实部
subplot(2, 1, 1)
stem(n, real(x1)); hold on;
xlabel('n')
ylabel('Re(x1)')

% 绘制 x1的虚部
subplot(2, 1, 2)
stem(n, imag(x1)); hold on;
xlabel('n')
ylabel('Im(x1)')

% 利用循环绘制 x2-x4
for k = 2:4
    % 利用 stem 绘制对应数值
    stem(n, eval(['x', num2str(k)])); hold on;
    xlabel('n')
    ylabel(['x', num2str(k)])
end

%% 求在差分方程表征的 LTI 系统下各个信号的输出
% LTI 系统差分方程的系数矩阵
a = [1, -0.25];
b = [1, 0.9];
% LTI 系统的输出
y1 = filter(b, a, x1);
y2 = filter(b, a, x2);
y3 = filter(b, a, x3);
y4 = filter(b, a, x4);
```

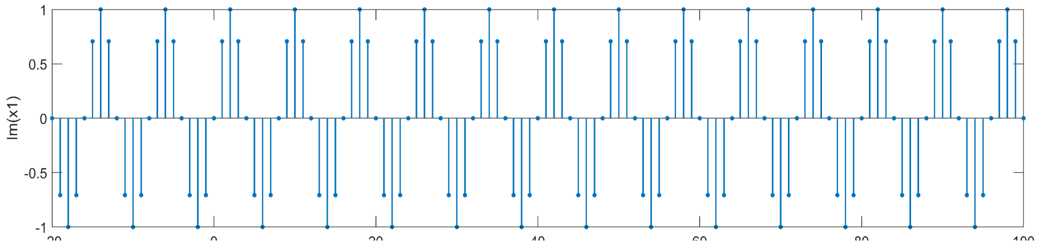
接上页代码

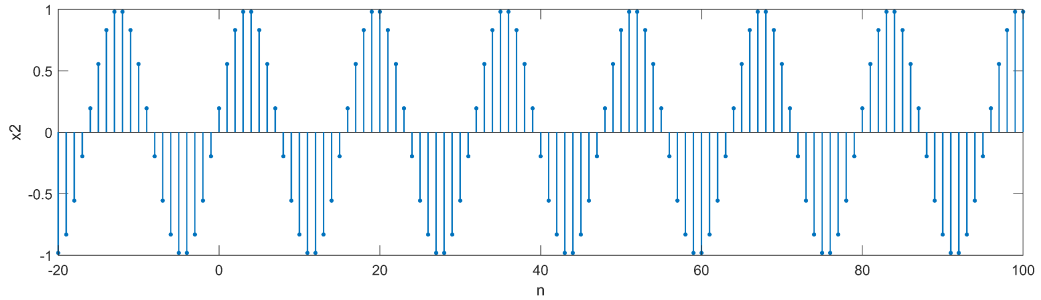
```
% 绘制 y1的图形
subplot(2, 1, 1)
stem(0:100, real(y1(21:121)));
xlabel('n')
ylabel('Re(y1)')
subplot(2, 1, 2)
stem(0:100, imag(y1(21:121)));
xlabel('n')
ylabel('Im(y1)')

% 利用循环绘制 y2-y4
for k = 2:4
    Y = eval(['y', num2str(k)]);
    stem(0:100, Y(21:121));
    xlabel('n');
    ylabel(['y', num2str(k)]);
end

%% 求每一个的信号的 H(z)
h1 = y1 ./ x1;
h2 = y2 ./ x2;
h3 = y3 ./ x3;
h4 = y4 ./ x4;
% 在同一个图上绘制 h1~h4的图形
h1img = plot(0:100, real(h1(21:121))); hold on;
h2img = plot(0:100, imag(h1(21:121))); hold on;
h3img = plot(0:100, h2(21:121)); hold on;
h4img = plot(0:100, h3(21:121)); hold on;
h5img = plot(0:100, h4(21:121)); hold on;
% 添加图例
legd = leg-
end([h1img, h2img, h3img, h4img, h5img], {'Re(H_1(z))', 'Im(H_1(z))', '
H_2(z)', 'H_3(z)', 'H_4(z)'});
% 添加坐标轴标签
label_1 = xlabel('n');
lebel_2 = ylabel('H(z)');
```

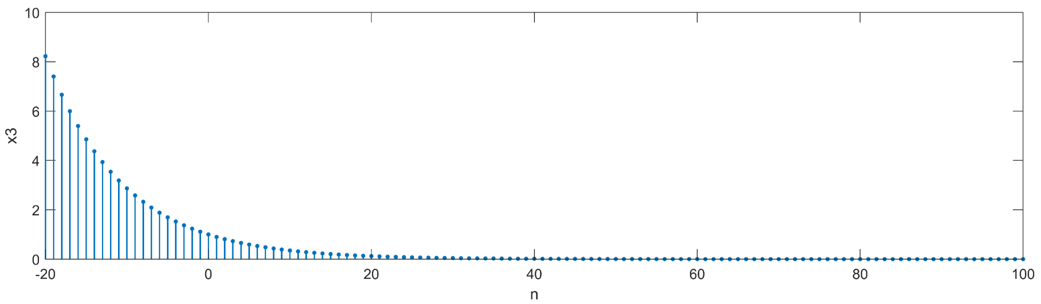
实验结果：

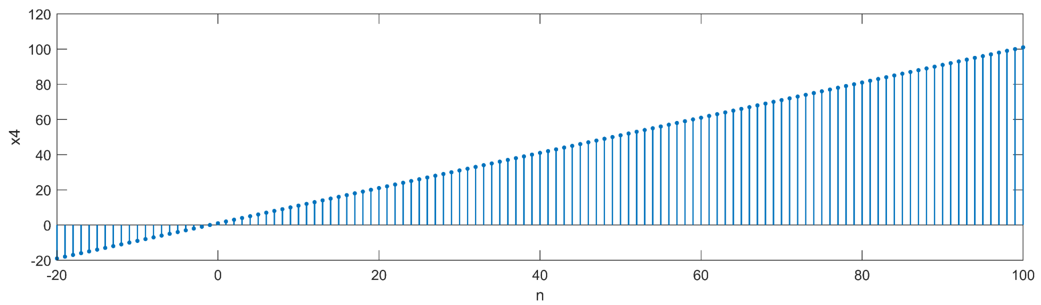
a) $x_1[n] \sim x_4[n]$在区间$[-20,100]$上的值



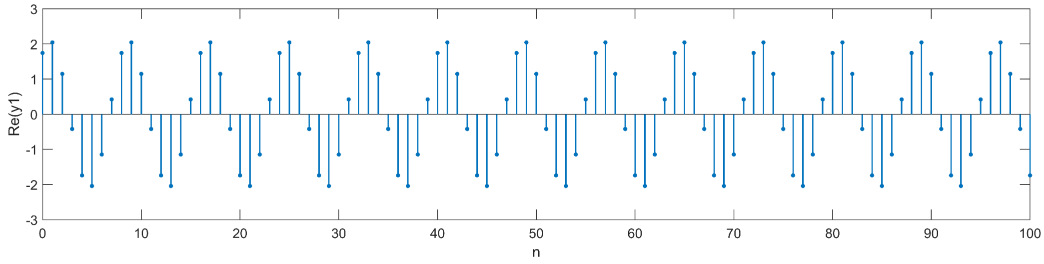



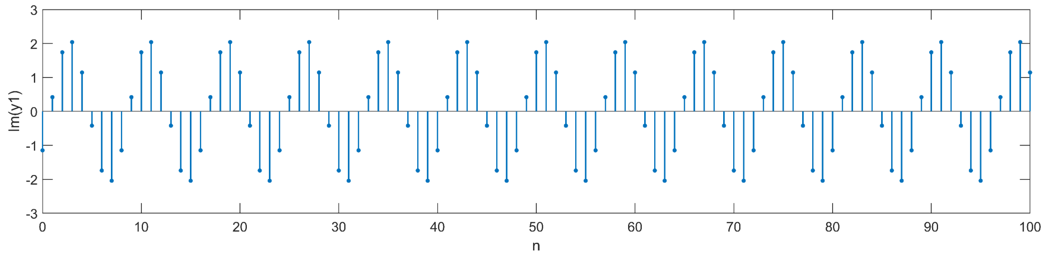



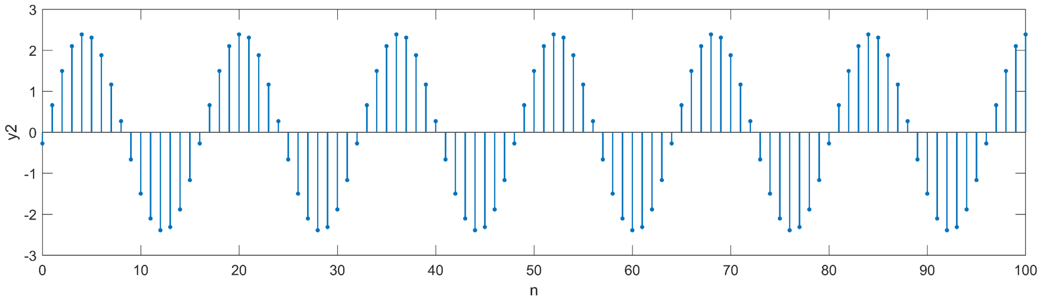



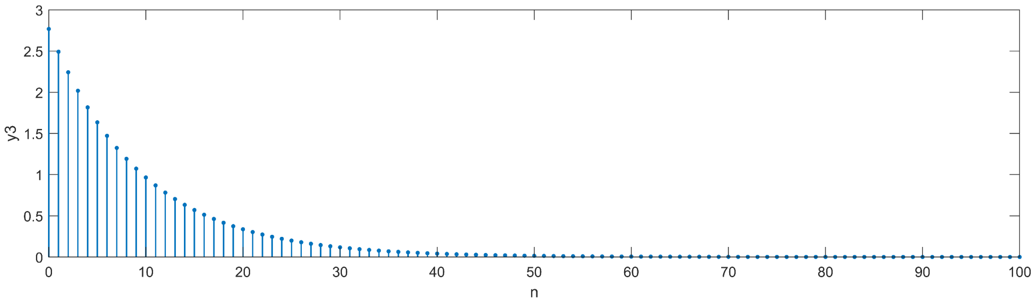



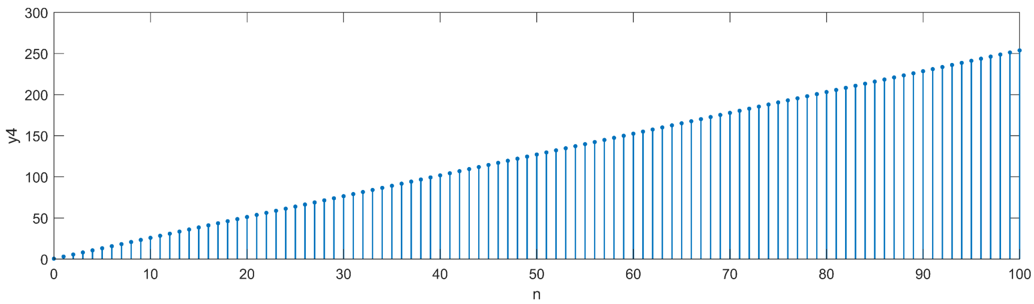
b) $y_1[n]$~$y_4[n]$在 n 的区间为[0,100]上的值



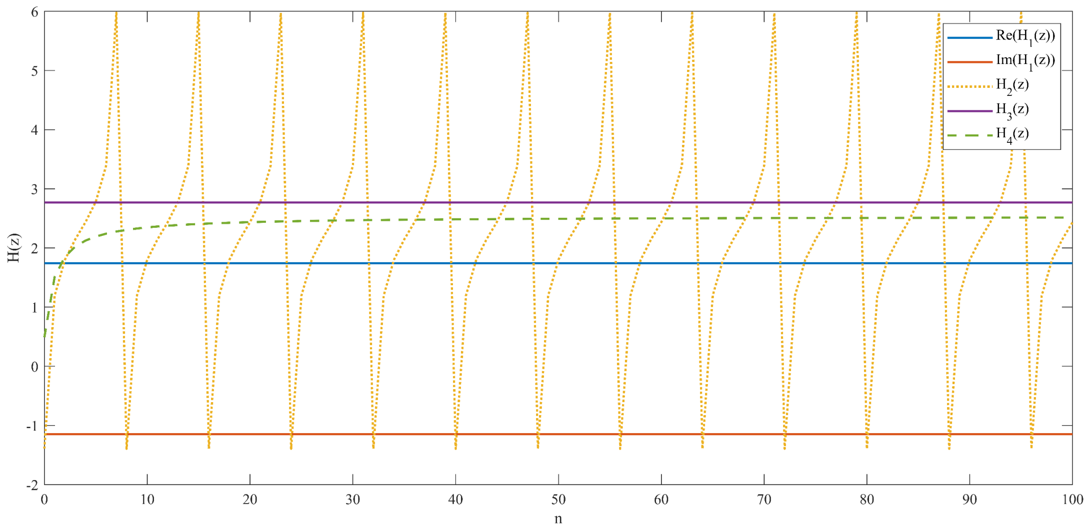



c) 特征值$H(z)$的图形



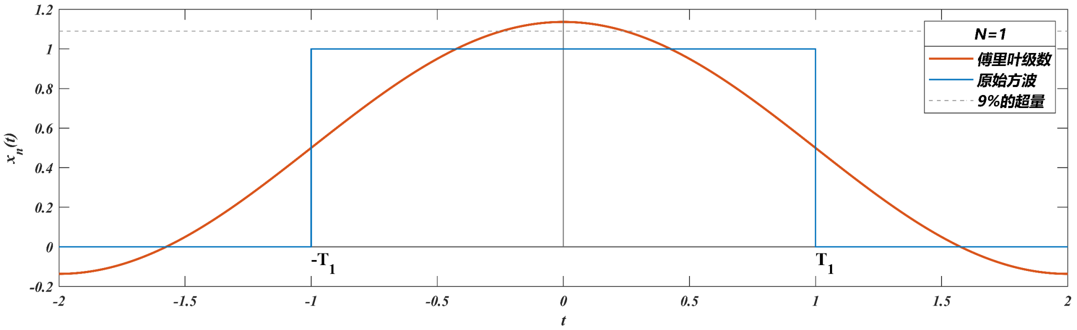

实验结论：从上图可以看出，$H_1(z)$和$H_3(z)$的值为常数，所以$x_1[n]$和$x_3[n]$就是改系统的特征函数。这与我们所学的理论知识也是完全吻合的。$x_1[n]$和$x_3[n]$都是简单基本信号，他们经过一个 LTI 系统的相应是输入的倍数。

4. 验证吉布斯现象

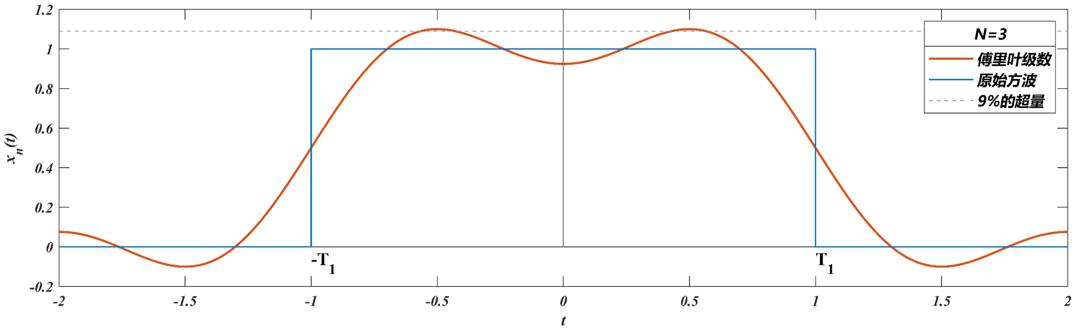
实验代码：

```
N = 3; % 定义 N 的大小
T = 4; % 矩形波脉冲周期
w = 2 * pi / T;
T1 = 1; % 矩形波高电平时间
% 定义句柄函数 ak (k≠0)
ak = @(k) sin(k * w * T1) / (k * pi)
t = -2:0.00001:2;  % 在[-2,2]上绘图
xn = zeros(1, length(t)); % 设置 xn 初始值

%利用循环结构计算在指定的 N 下的求和结果
for k = -N:N
    if k ~= 0    % 处理 k 不等于 0 的时候的结果
        xn = xn + ak(k) * exp(1i * k * w .* t);
    else     % 处理 k 等于 0 的时候的结果
        xn = xn + 2 * T1 / T;
    end
end

% 绘制求和结果
fs=plot(t, xn); hold on;
% 绘制 x 轴和 y 轴
line([-2, 2], [0, 0]); hold on;
line([0, 0], [1.2, 0]); hold on;
% 绘制 9%超量对应的直线
maxVal = line([-2, 2], [1.09, 1.09]); hold on;
% 绘制原来的方波信号
squareWave = plot([-2, -1, -1, 1, 1, 2], [0, 0, 1, 1, 0, 0]);
hold on;
% 添加文本标注
text(-1, -0.08, '-T_1'); hold on;
text(1, -0.08, 'T_1'); hold on;
xlabel('t'); ylabel('x_n(t)');

% 添加图例
legd = legend([fs, squareWave, maxVal], {'傅里叶级数', '原始方波
', '9%的超量'}, 'FontName', '微软雅黑', 'FontSize', 10);
title(legd, ['N=', num2str(N)])
```

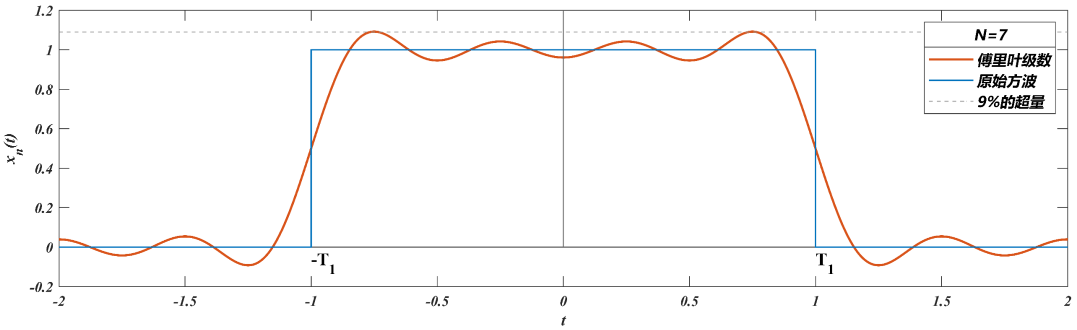
实验结果：



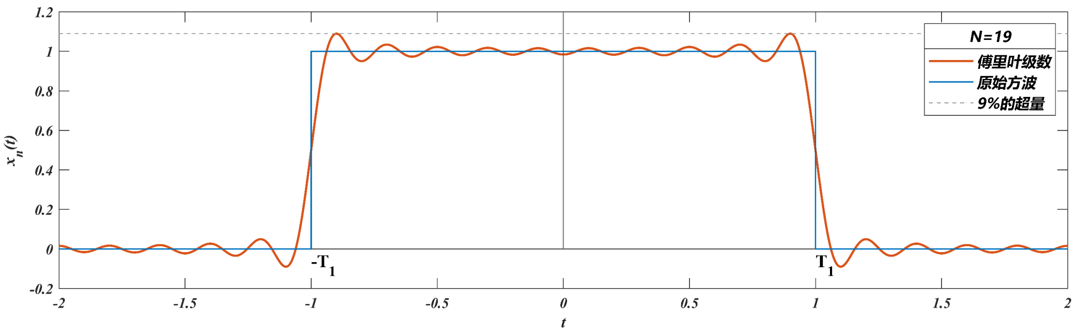

N=1 时候的结果



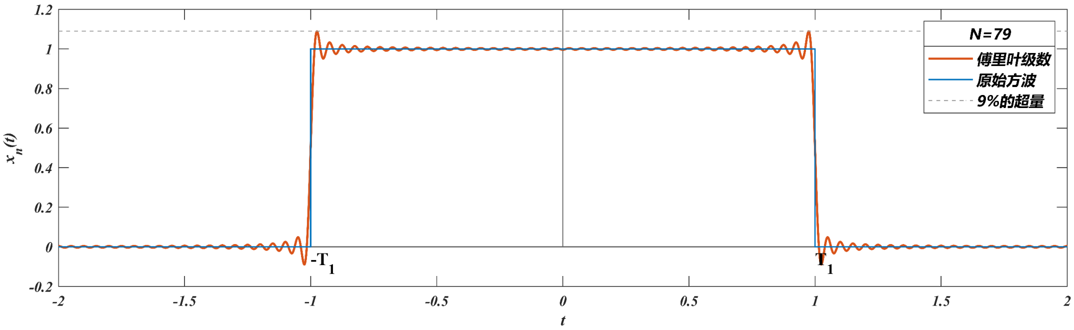

N=3 时候的结果



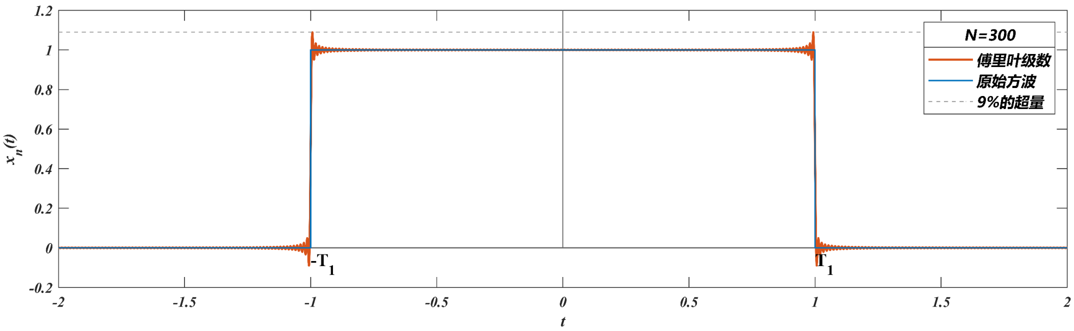

N=7 时候的结果



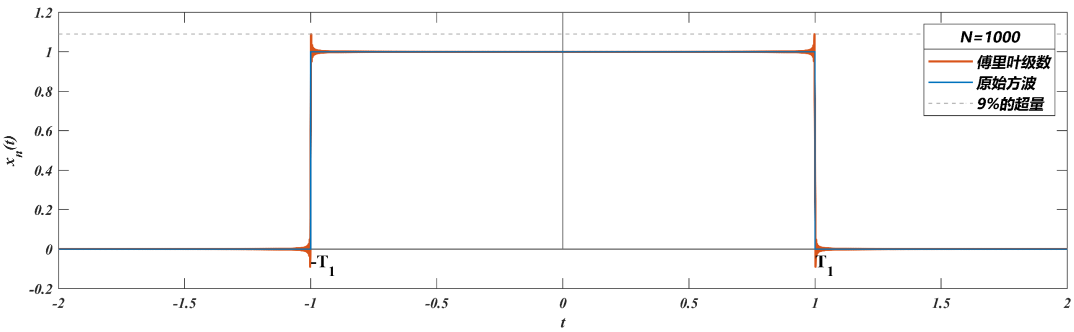

N=19 时候的结果




N=79 时候的结果




N=300 时候的结果




N=1000 时候的结果

实验小结：从上面的一系列图形可以看出，随着 N 的增大，xn 逐渐趋近于原来的方波信号，但是在间断点处有一个 9%的超量，也就是吉布斯现象。

## 四、实验心得

通过这次的实验，让我学会了在 *Matlab* 中进行简单的信号处理的方法。通过自己的操作，对于信号与系统理论课程中的内容加深了理解。在学习的过程中感受到了 *Matlab* 功能的强大，使得处理问题变得这么简单，只需要简单调用 *fft*、*freqz*、*freqs*、*filter* 等函数就能完成自己的要求，能够大幅提高效率。最后通过自己的编写，看出吉布斯现象，印象也更深了。