Github 账号：bleaner

实验摘要：

- 熟悉Matlab 软件平台和基本操作；
- 掌握利用Matlab 来显示常用信号波形；
- 掌握利用 Matlab 来实现信号的时域变换。

实验题目

1. 利用MATLAB求下列函数的卷积，并绘制出图形

   (1) $f_1(t)=\varepsilon(t)-\varepsilon(t-1)$，$f_2(t)=2t[\varepsilon(t)-\varepsilon(t-1)]$

   (2) $f_1(t)=\cos(30t)g_5(t)$，$f_2(t)=\varepsilon(t)-\varepsilon(t-4)$

   参考函数：conv( )

2. 某系统满足的微分方程为

   $$y''(t)+4y'(t)+3y(t)=2f'(t)+f(t)$$

   (1) 利用MATLAB求系统的单位冲击响应，并绘出图形

   (2) 利用MATLAB求系统的单位阶跃响应，并绘出图形

   (3) 利用MATLAB求系统对信号 $f(t)=4\sin(2\pi t)\varepsilon(t)$ 的响应，并绘出图形

   参考函数：tf( )，impulse( )，step( )，lsim( )，conv( )

3. 利用MATLAB产生高斯白噪声，绘出图形，并求其自相关函数，绘出图形。

   参考函数：randn()，wgn()，xcorr()，autocorr()

4. 预习关于傅里叶级数的内容，用MATLAB或者Python进行以下实验，回答问题并给出实验过程中产生的结果图。

   (1)信号 $f(t)$ 的傅里叶级数为 $\sum_{1}^{\infty}\frac{\sin nt}{n}$，代入数字去逼近或者用解析法分析，估计 $f(t)$ 的形式。

   (2)写出你估计出的 $f(t)$ 的傅里叶级数，与上式对比，说明它的谐波和正余弦分量的情况。

   (3)取 $N=50,100,200,......$ 画出 $f_N(t)=\sum_{n=1}^{N}\frac{\sin nt}{n}$，当 $N\to\infty$ 时，判断这个部分和与 $f(t)$ 的区别。

   (4)同样，取 $N=50,100,200,......$ 画出 $F_N(t)=\frac{f_1(t)+f_2(t)+f_3(t)+...+f_N(t)}{N}$，和上面的图对比，分析他们之间的不同。

实验内容

1.关键函数

conv 卷积和多项式乘法

用法：

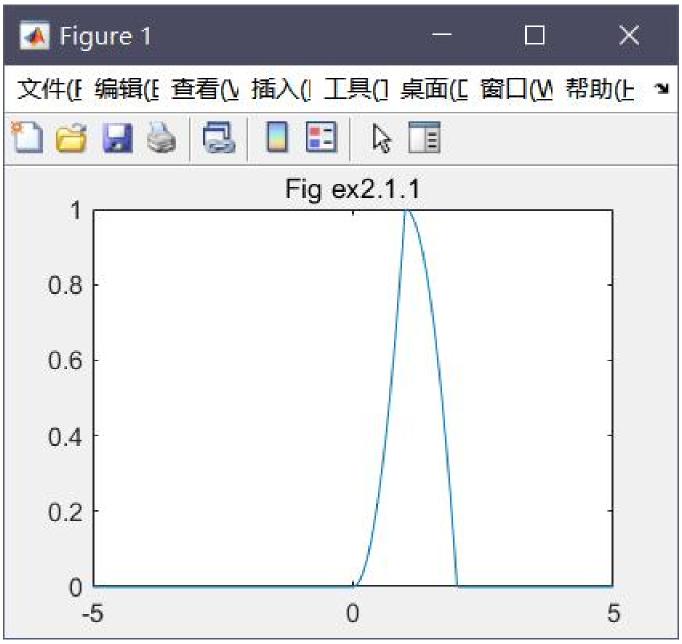
w = conv(u,v) %返回向量 u 和 v 的卷积。如果 u 和 v 是多项式系数的向量，对其卷积与将这两个多项式相乘等效。

w = conv(u,v,shape) %返回如 shape 指定的卷积的分段。例如，conv(u,v,'same') 仅返回与 u 等大小的卷积的中心部分，而 conv(u,v,'valid') 仅返回计算的没有补零边缘的卷积部分。

代码及实验结果：

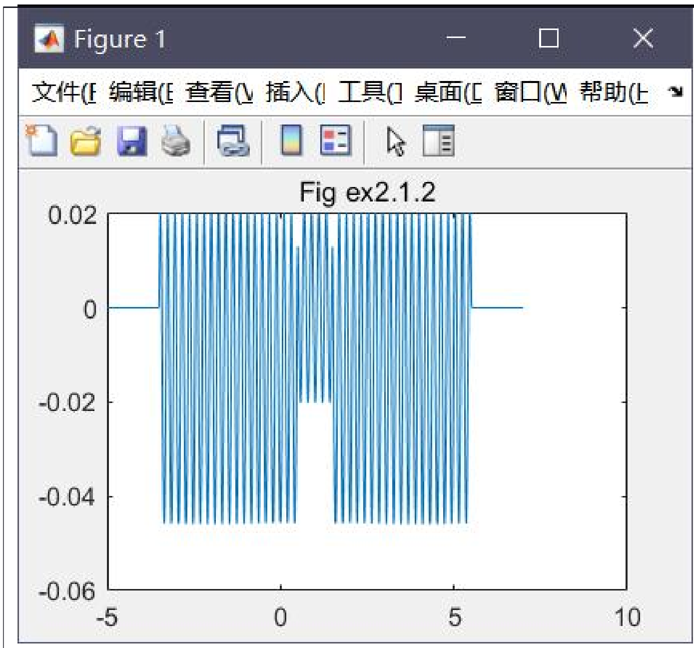
（1）

```
t = -5 : 0.01 : 5;
f1 = heaviside(t) - heaviside(t-1);
f2 = 2 * t .* f1;
rlt = conv(f1, f2, 'same') * 0.01;
plot(t, rlt);
title('Fig ex2.1.1');
```



（2）

```
t = -5 : 0.01 : 7;
g = heaviside(t+2.5) - heaviside(t-2.5);
f1 = cos(30*t) .* g;
f2 = heaviside(t) - heaviside(t-4);
rlt = conv(f1, f2, 'same') * 0.01;
plot(t, rlt);
title('Fig ex2.1.2');
```



2.

关键函数:

(1) impulse 动态系统模型的冲激响应

常用用法:

impulse(sys) %绘制动态系统模型的冲激响应。该模型可以是连续或离散的

impulse(sys, t) % 使用用户提供的时间向量 t 进行仿真

(2) step %动态系统模型的阶跃响应

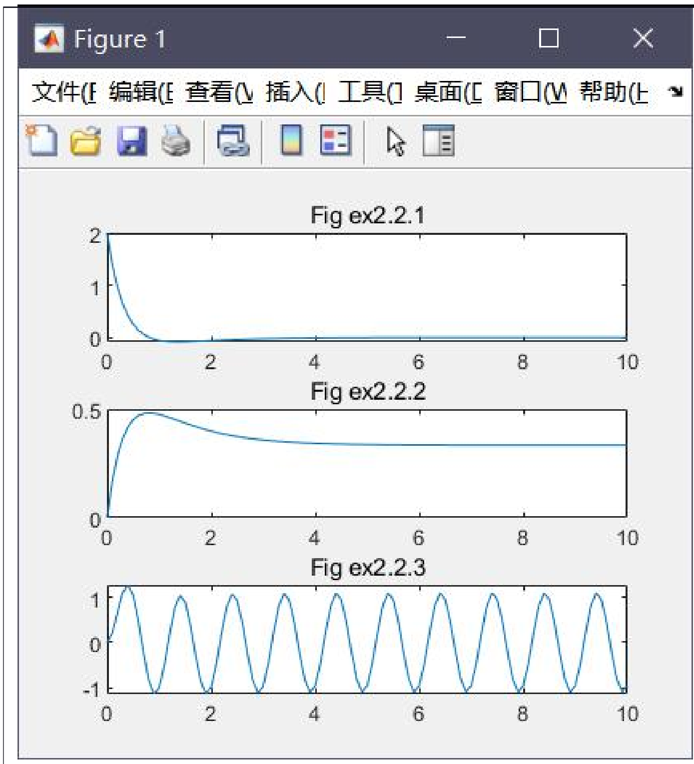
常用用法:

Step(sys) %绘制动态系统模型的阶跃响应。该模型可以是连续或离散的

Step(sys, t) %使用用户提供的时间向量 t 进行仿真

代码及实验结果:

```
clear;
a = [1, 4, 3]; b = [2, 1];
t = 0 : 0.1 : 10;
sys = tf(b, a); %LTI系统初始化
h = impulse(sys, t); %单位冲激响应
g = step(sys, t); %单位阶跃响应
f = 4 * sin(2*pi*t) .* heaviside(t);
y = lsim(sys, f, t); %任意时间信号响应
subplot(3, 1, 1); plot(t, h); title('Fig ex2.2.1');
subplot(3, 1, 2); plot(t, g); title('Fig ex2.2.2');
subplot(3, 1, 3); plot(t, y); title('Fig ex2.2.3');
```



3.

关键函数:

（1）wgn 生成高斯白噪声

常用用法:

noise = wgn(m,n,power) % 生成 m*n 的高斯白噪声样本矩阵，power 为功率，单位 dbw

（2）autocorr 样本自相关（不是特别明白）

常用用法:

[acf,lags,bounds] = autocorr(y,NumLags)

% y:时间序列 NumLags:延迟，当 NumLags=[]或缺省时，计算 ACF 时在延迟点 0、1、2、。。。、T 处，T=min([20 length(Series-1)])

Acf:样本自相关函数 lags:与 ACF（0,1,2，…，NumLags）相对应的延迟 Bounds--置信区间的近似上下限

(3) xcorr 自相关/互相关函数

常用用法:

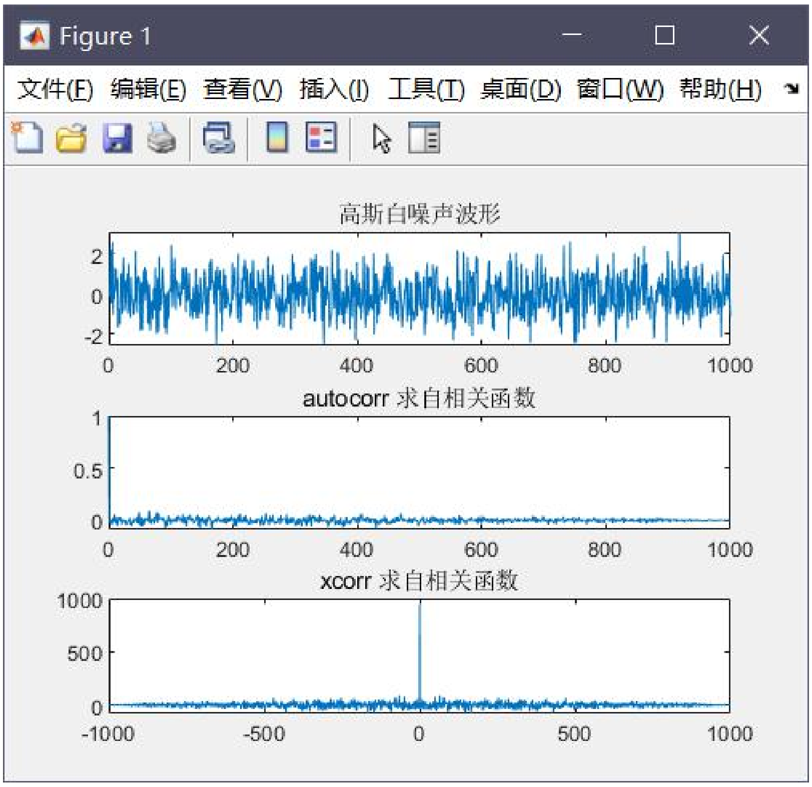
[r,lags] = xcorr(x) %返回 x 的自相关函数

[r,lags] = xcorr（x,y）%返回 x,y 的互相关

%关于 xcorr 和 autocorr 的区别，在官方文档中我也没有看明白，查网上的大概就
%是 autocorr 是对序列减去均值后做的自相关，最后又进行了归一化。而且由于
%自相关本身是偶函数，autocorr 只是取了以中点 n 为起始的后面 n 个序列
%实验里我没有发现这个区别，可能是因为 wgn 产生的高斯白噪声均值等于 0
%目前对自相关/互相关的理解还不深刻

代码及实验结果:

```
clc; clear;
y = wgn(1000, 1, 1000);%产生1000*1, 功率 0dbw 的高斯白噪声样本
subplot(3, 1, 1); plot(y); title('高斯白噪声波形');
n = length(y);
[ACF, lags] = autocorr(y,'NumLags',n-1);
subplot(3, 1, 2); plot(lags, ACF); title('autocorr 求自相关函数');
[r1, lags]=xcorr(y);
subplot(3, 1, 3); plot(lags, r1); title('xcorr 求自相关函数');
```



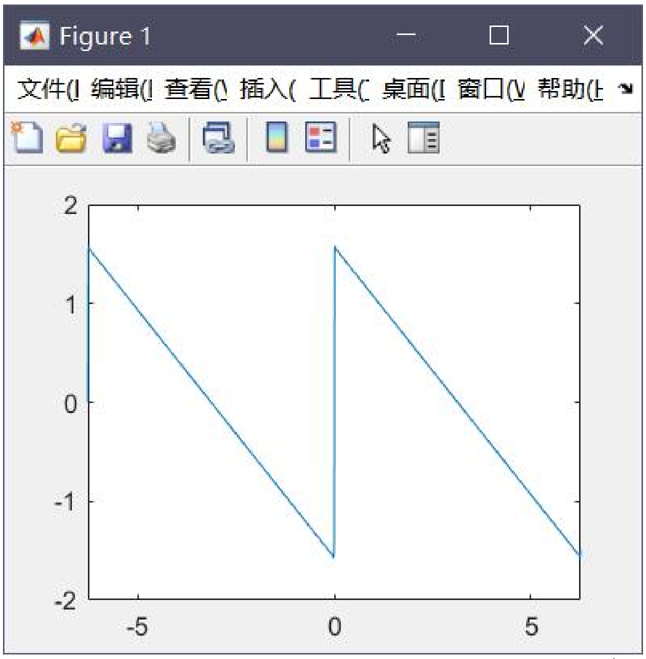
4.

该实验用到的 matlab 知识很基础，主要的难点是对傅里叶级数以及吉布斯效应的理解。

（1）对给出的级数进行 matlab 仿真，代码如下

```
clc; clear;
t = -2 * pi : 0.01 : 2 * pi;
f = zeros(size(t));
for n = 1 : 1 : 2000
    f = f + sin(n*t) ./ n;
end
plot(t, f);
```

结果如下



f(t)为周期为 2pi，猜测 t 在(0,2pi)上 $f(t)=\frac{\pi-t}{2}$

（2）

f(t)的傅里叶级数为 $\sum_{n=1}^{\infty}\frac{\sin(nt)}{n}$

n 次谐波为 $\frac{1}{n}cos\left(nt-\frac{\pi}{2}\right)$

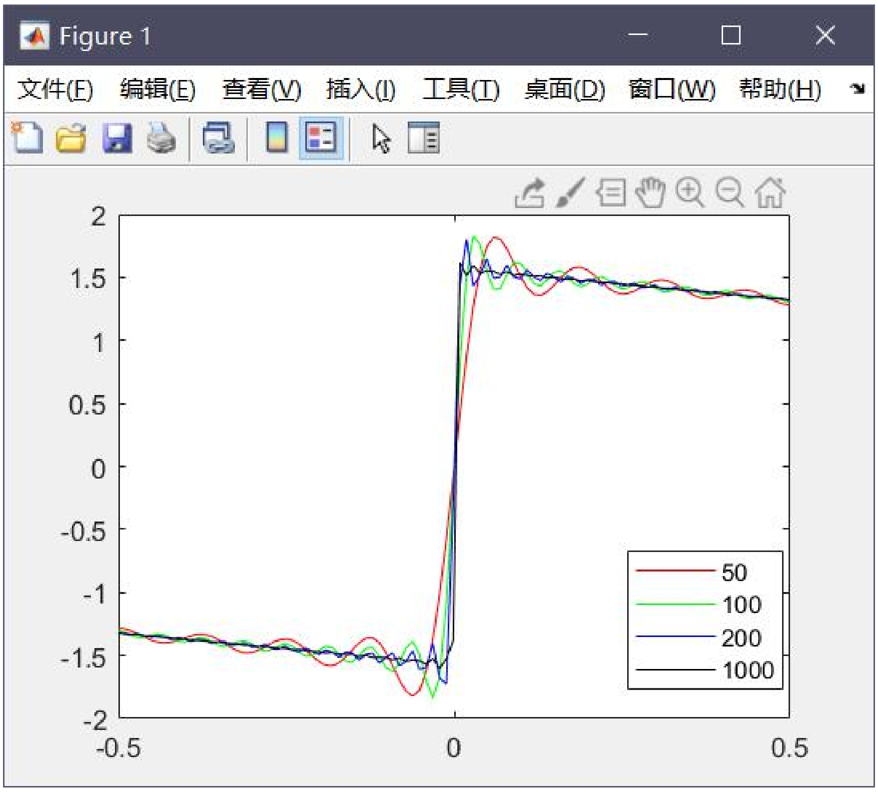
正弦分量为 $\sum_{n=1}^{\infty}\frac{\sin(nt)}{n}$，余弦分量为 0

（3）

取 N=50,100,200,1000,做出图像，代码如下

```
clc; clear;
t = -pi : 0.01 : pi;
f1 = f(50); f2 = f(100); f3 = f(200); f4 = f(1000);
plot(t, f1, 'r'); hold on;
plot(t, f2, 'g'); hold on;
plot(t, f3, 'b'); hold on;
plot(t, f4, 'y');
xlim([-0.5 0.5]); ylim([-2 2]);
%function fN(t)
function rlt = f(N)
  t = -pi : 0.01 : pi;
  rlt = zeros(size(t));
  for n = 1 : N
     rlt = rlt + sin(n*t) ./ n;
  end
end
```

图像如下



可以观察到随着 N 的增大，选取的项数增多，在所合成的波形中出现的峰起越靠近原信号的不连续点(t=0)，吉布斯效应体现的很清晰。

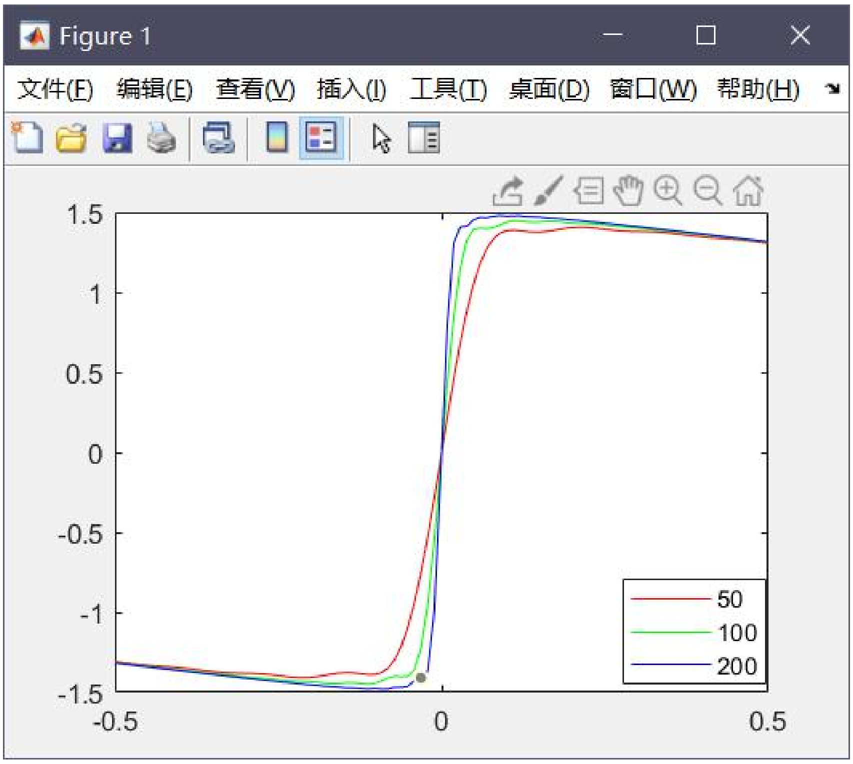
(4)

依然用 matlab 画图，分别取 N=50,100,200，代码如下:

```
%F.m
%function Fn(t)
function F = F(N)
  t = -pi : 0.01 : pi;
  temp = zeros(size(t));
  for k = 1 : N
     temp = temp + f(k);
  end
  F = temp / N;
end
%function fn(t)
function f = f(N)
  t = -pi : 0.01 : pi;
  f = zeros(size(t));
  for n = 1 : N
     f = f + sin(n*t) ./ n;
  end
end
```

```
%ex2_4_4.m
clc; clear;
t = -pi : 0.01 : pi;
F50 = F(50); F100 = F(100); F200 = F(200);
plot(t, F50, 'r'); hold on;
plot(t, F100, 'g'); hold on;
plot(t, F200, 'b'); hold on;
legend('50', '100', '200'); xlim([-0.5 0.5]);
```

图像如下：



很明显，和(3)相比吉布斯效应减弱了许多，而且数字越大越逼近真实的 f(t)。

实验总结

matlab 理解应用；

对傅里叶级数和吉布斯效应的理解；

英文文档有点难理解，要增加这方面的能力。

参考文献

Matlab 官方文档

Matlab 的 autocorr 自相关函数(https://blog.csdn.net/lfdanding/article/details/50726678)