一、实验目的

1. 掌握信号的表示及其可视化方法。
2. 掌握信号基本时域运算的实现方法。
3. 实现线性时不变LTI系统的全响应求解，并把基于仿真平台内置函数的仿真结果与理论计算结果进行比较。
4. 实现周期信号的傅里叶级数展开。

二、实验报告要求

1. 提交：实验报告一份，PDF格式，其他格式拒收。 实验报告中需要包括：

- 若题目要求理论结果，报告中需要给出理论结果。
- 结果图；图中需要有适当的标识、横坐标、纵坐标等。
- 源代码。源代码中要有合适的注释。
- 实验体会和感悟。

2. 提交实验报告规则：

- 2022年10月28日12am之前将实验报告发到助教邮箱。 （第一课堂交给施锐，邮箱： 296206140@qq.com; 第二课堂曹歌，邮箱：1765578099 **(qq.com?)**）
  文件名命名规则：课堂号-学号-姓名-第几次实验。（比如第2课堂的学生，姓名：李三，学号为2019050，第2次实验，文件名命名为：2-2019050-李三-2）

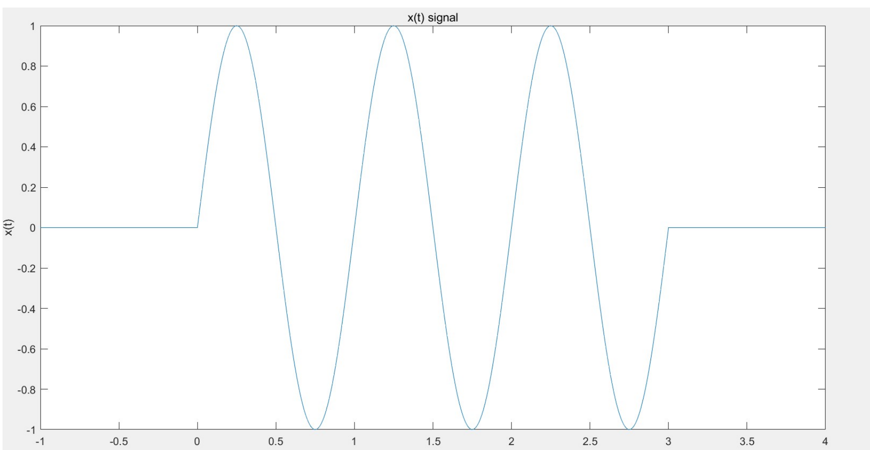
三、实验设备（环境）

操作系统Windows11/10/9/8/7
编程软件：推荐Matlab，版本不低于2016版本。

四、实验内容与实验结果

1. 利用**MATLAB**绘制下列连续时间信号的波形

1）.
结果图：



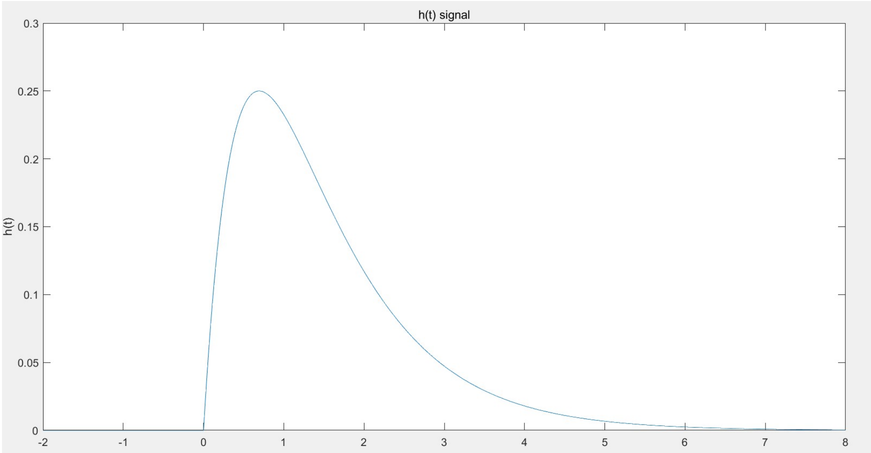
源代码：

```
t=-1:0.01:4; % time vector; unit: second
x_t = sin(2*pi*t).*[heaviside(t)-heaviside(t-3)]; % function vector
```

```
figure; plot(t,x_t); % show the figure
axis([min(t) max(t) min(x_t) max(x_t)]) % limit
the axis
title('x(t) signal');
xlabel('t [s]');
ylabel('x(t)');
```

2）.

结果图：

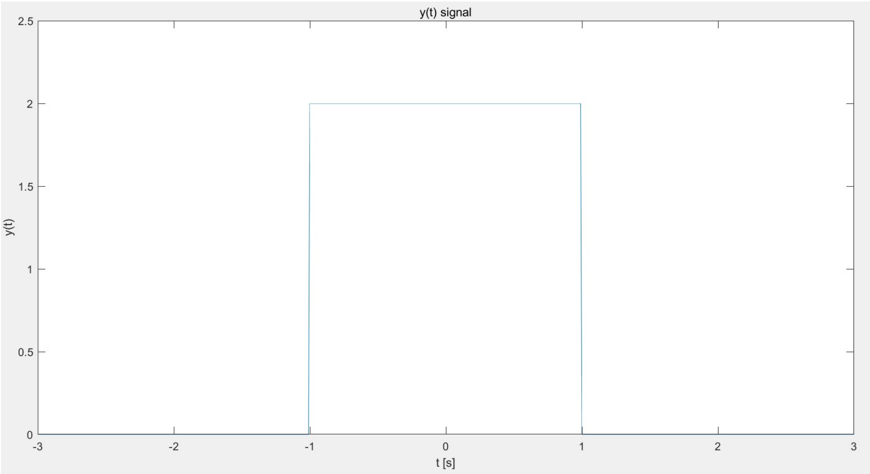


源代码：

```
t=-2:0.01:8; % time vector; unit: second
h_t = [exp(-t)-exp((-2)*t)].*heaviside(t); %
function vector
figure; plot(t,h_t); % show the figure
axis([min(t) max(t) 0 0.3]) % limit the axis
title('h(t) signal');
xlabel('t [s]');
ylabel('h(t)');
```

3）.

结果图：



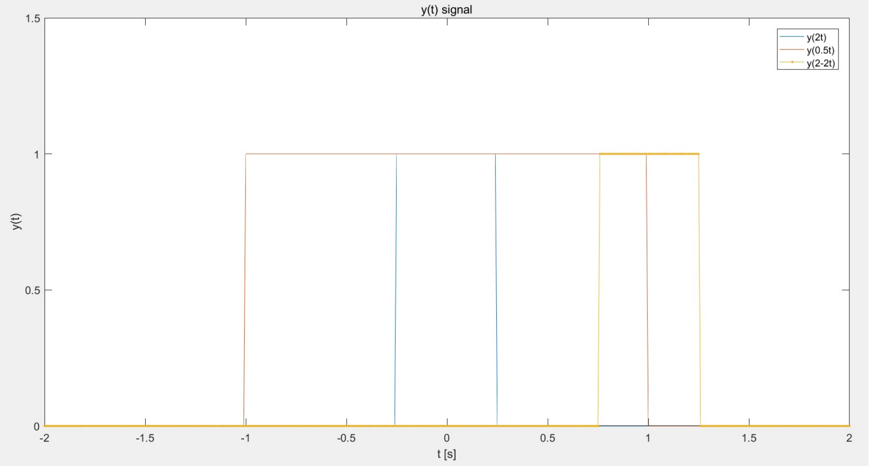
源代码：

```
t=-3:0.01:3; % time vector; unit: second
y_t = 2.*rectpuls(0.5*t); % function vector
figure; plot(t,y_t); % show the figure
axis([min(t) max(t) 0 2.5]) % limit the axis
title('y(t) signal');
```

```
xlabel('t [s]');
ylabel('y(t)');
```

2. 利用MATLAB验证信号的基本运算

1).
结果图：

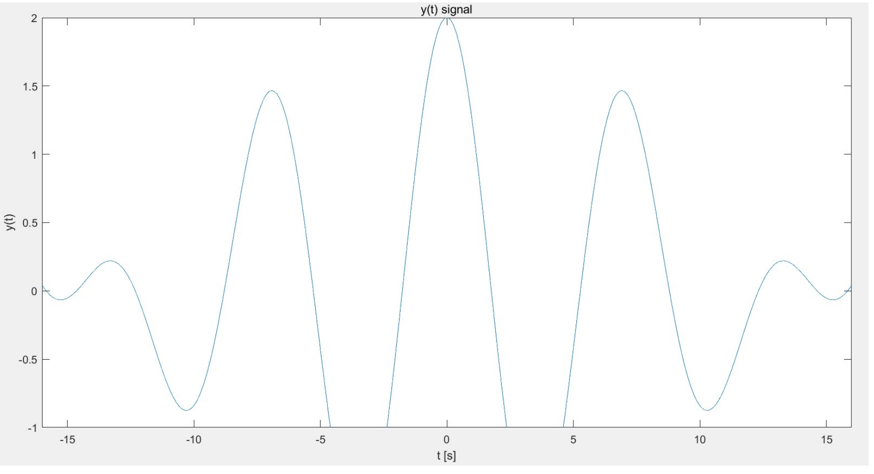


源代码：

```
t=-2:0.01:2; % time vector; unit: second
figure;
plot(t,rectpuls(2*t),t,rectpuls(0.5*t),t,rectpuls(2-2*t),'.-
'),legend('y(2t)','y(0.5t)','y(2-2t)'); % show the figure
axis([min(t) max(t) 0 1.5]) % limit the axis
title('y(t) signal');
xlabel('t [s]');
ylabel('y(t)');
```

MATLAB画出的结果和理论分析得出的结果基本一致。

2).
结果图：



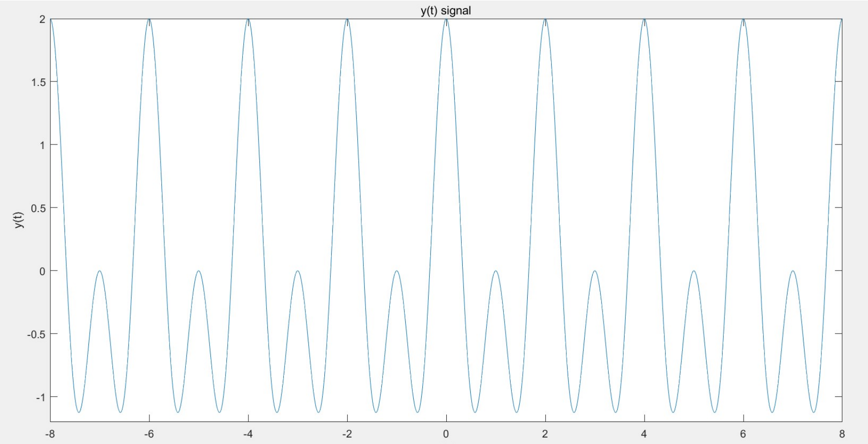
源代码：

```
t=-16:0.01:16; % time vector; unit: second
y_t = cos(t)+cos(0.25*pi*t); % function vector
figure; plot(t,y_t); % show the figure
axis([min(t) max(t) -1 2]) % limit the axis
title('y(t) signal');
xlabel('t [s]');
ylabel('y(t)');
```

不是周期函数.

3).

结果图：



源代码：

```
t=-8:0.01:8; % time vector; unit: second
y_t = cos(pi*t)+cos(2*pi*t); % function vector
figure; plot(t,y_t); % show the figure
axis([min(t) max(t) -1.2 2]) % limit the axis
title('y(t) signal');
xlabel('t [s]');
ylabel('y(t)');
```

是周期函数，周期$T = 2\pi/\pi = 2.$

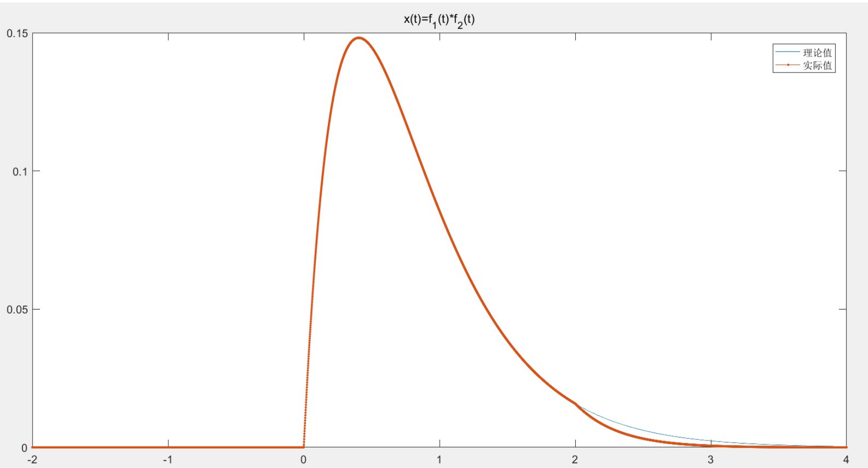
3. 卷积运算

1).

$x(t) = [e^{-2t}\epsilon(t)] * [e^{-3t}\epsilon(t)] = \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau = \int_0^{\infty} e^{-2\tau}\epsilon(\tau)\cdot e^{-3t+3\tau}\epsilon(3t-3\tau)d\tau = (e^{-2t} - e^{-3t}$

2).

结果图：



源代码：

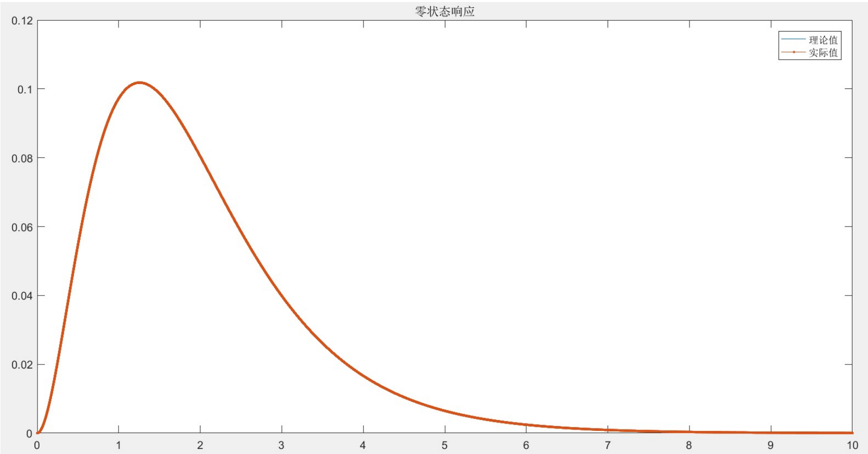
```
t=-1:0.001:2;
f1_t = exp(-2*t).*heaviside(t);
f2_t = exp(-3*t).*heaviside(t);
t3=-2 : 0.001 : 4;
f_t = (exp(-2*t3)-exp(-3*t3)).*heaviside(t3);
f3_t=conv(f1_t, f2_t);
f3_t=f3_t*0.001;
```

```
plot(t3,f_t,t3, f3_t,'.-'),legend('理论值','实际值');
title('x(t)=f_1(t)*f_2(t)');
```

3).
与理论值有小幅差异，猜测是Matlab计算卷积使用的离散化数值算法导致的系统误差。

4. 求解系统的零状态响应

1).
特征方程为：$\lambda^2 + 3\lambda + 2 = 0$,解得$\lambda_1 = -1, \lambda_2 = -2$;
$h(t) = H(p)\delta(t), H(p) = \frac{1}{(p+1)(p+2)} = \frac{1}{p+1} - \frac{1}{p+2}$,所以$h(t) = (e^{-t} - e^{-2t})\epsilon(t)$;
$r_{zs} = e(t) * h(t) = [e^{-2t}\epsilon(t)] * [(e^{-t} - e^{-2t})\epsilon(t)] = [e^{-2t}\epsilon(t)] * [e^{-t}\epsilon(t)] - [e^{-2t}\epsilon(t)] * [e^{-2t}\epsilon(t)] = (e^{-t} -$
2).
结果图：



源代码：

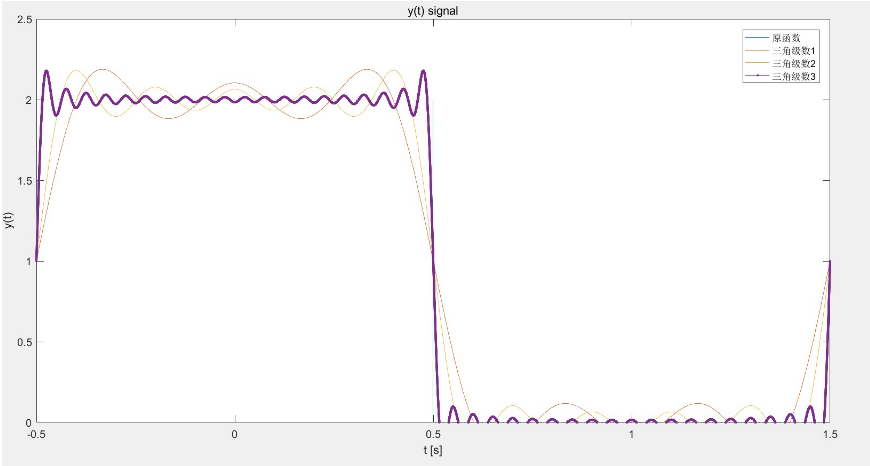
```
sys = tf(1,[1,3,2]);
t=0:0.0001:10;
e = exp(-2*t);
y = lsim(sys,e,t);
r_z_s = (exp(-t)-exp(-2*t)-t.*exp(-2*t)).*heaviside(t);

plot(t,r_z_s,t, y,'.-'),legend('理论值','实际值');
title('零状态响应');
```

3). 一致。

5. 周期信号的傅里叶级数展开

1). $f(t) = 1 + \sum_{n=1}^{\infty} \frac{4}{n\pi}\sin\frac{n\pi}{2}\cos(n\pi t)$
2).
结果图：



源代码：

```
t=-0.5:0.00001:1.5; % time vector; unit: second
y_t = 2.*rectpuls(t); % function vector
f_1 = 1;
f_2 = 1;
f_3 = 1;
for n=1:2:5
    f_1 = f_1 + 4/(n*pi)*sin(0.5*n*pi)*cos(n*pi*t);
end;
for n=1:2:9
    f_2 = f_2 + 4/(n*pi)*sin(0.5*n*pi)*cos(n*pi*t);
end;
for n=1:2:39
    f_3 = f_3 + 4/(n*pi)*sin(0.5*n*pi)*cos(n*pi*t);
end;

figure; plot(t,y_t,t,f_1,t,f_2,t,f_3,'.-'),legend('原函数','三角级数1','三角级数2','三角级数3'); % show the figure
axis([min(t) max(t) 0 2.5]) % limit the axis
title('y(t) signal');
xlabel('t [s]');
ylabel('y(t)');
```

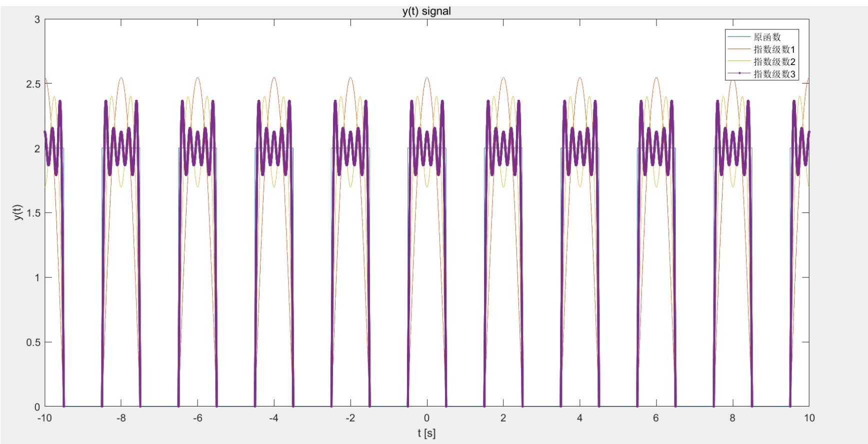
根据观察，加的项数越多，得到的傅里叶展开与原函数近似程度越大。

3).

$$F(t) = \sum_{-\infty}^{\infty} \frac{2j}{n\pi}\left(e^{-\frac{jn\pi}{2}} - e^{\frac{jn\pi}{2}}\right)e^{jn\pi t}$$

4).

结果图：



源代码：

```
t=-10:0.00001:10; % time vector; unit: second
y_t = square(pi*(t+0.5),50)+1; % function vector
f_1 = 0;
f_2 = 0;
f_3 = 0;
for n=1
    f_1 = f_1 + 8/(n*pi)*sin(0.5*n*pi)*exp(1i*n*pi*t);
end;
for n=1:2:3
    f_2 = f_2 + 8/(n*pi)*sin(0.5*n*pi)*exp(1i*n*pi*t);
end;
for n=1:2:10
    f_3 = f_3 + 8/(n*pi)*sin(0.5*n*pi)*exp(1i*n*pi*t);
end;

figure; plot(t,y_t,t,f_1,t,f_2,t,f_3,'.-'),legend('原函数','指数级数
1','指数级数2','指数级数3'); % show the figure
axis([min(t) max(t) 0 3]) % limit the axis
title('y(t) signal');
xlabel('t [s]');
ylabel('y(t)');
```

根据观察，加的项数越多，得到的傅里叶展开与原函数近似程度越大。同时，指数形式的展开的误差似乎整体上比三角形式的大。

五、实验体会和感悟

体会到了Matlab作为专业数学编程绘图软件的功能强大、开发简单、绘图清晰。
对各信号的图像有了更直观的认识。