

浙江大学 宁波理工学院

信号与系统实验报告二



姓 名： _____ 李明

学 号： _____ 3190432099

专业班级： _____ 电子信息工程 194

学 院： _____ 信 息 科 学 与 工 程 学 院

步骤一

首先是打开 VSCODE，先录入最基础的画图代码，如图

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 plt.rcParams['font.sans-serif']=['SimHei'] #加上这一句就能在图表中显示中文
5 plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
6
```

接着是根据函数图确定坐标轴的取值范围，以及设置坐标轴的名称和中心位置。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 plt.rcParams['font.sans-serif']=['SimHei'] #加上这一句就能在图表中显示中文
5 plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
6
7 #设置坐标轴
8 plt.xlim((-1*np.pi, 4*np.pi))
9 plt.ylim((-100, 100))
10 #设置坐标轴名称
11 plt.xlabel('t/s')
12 plt.ylabel('X(t)')
13 #设置坐标轴中心
14 ax = plt.gca() # get current axis 获得坐标轴对象
15 ax.spines['right'].set_color('none')
16 ax.spines['top'].set_color('none') # 将右边 上边的两条边颜色设置为空 其实就相当于抹掉这两条边
17 ax.spines['bottom'].set_position(('data', 0)) #指定 data 设置的bottom(也就是指定的x轴)绑定到y轴的0点
18 ax.spines['left'].set_position(('data', 0))
19
```

步骤二

为函数图像取好对应的标题，如图（以幅度增长的正弦信号为例）

```
plt.rcParams['font.sans-serif']=['SimHei'] #加上这一句就能在图表中显示中文
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号

#设置坐标轴
plt.xlim((-1*np.pi, 4*np.pi))
plt.ylim((-100, 100))
#设置坐标轴名称
plt.xlabel('t/s')
plt.ylabel('X(t)')
#设置坐标轴中心
ax = plt.gca() # get current axis 获得坐标轴对象
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none') # 将右边 上边的两条边颜色设置为空 其实就相当于抹掉这两条边
ax.spines['bottom'].set_position(('data', 0)) #指定 data 设置的bottom(也就是指定的x轴)绑定到y轴的0点
ax.spines['left'].set_position(('data', 0))

#标题
plt.title(r'x(t)=e^(1/2*t)*cos(4*t+pi)')
```

最后根据要画的函数，对 x 采样，对 y 赋值画图。

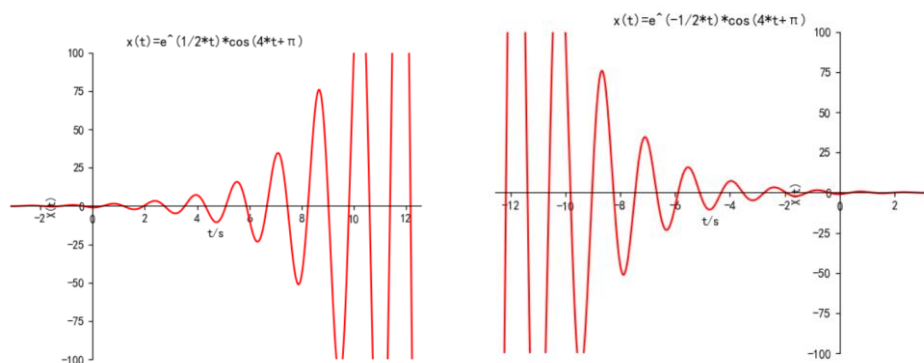
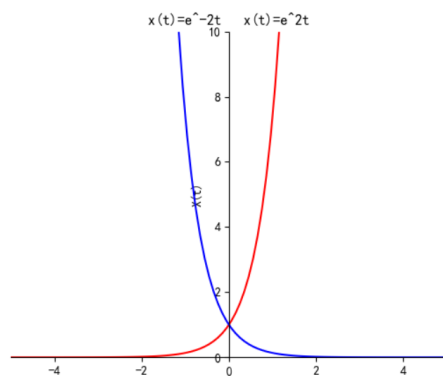
```
plt.ylim((-100, 100))
#设置坐标轴名称
plt.xlabel('t/s')
plt.ylabel('X(t)')
#设置坐标轴中心
ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position(('data', 0))
ax.spines['left'].set_position(('data', 0))

# get current axis 获得坐标轴对象
# 将右边 上边的两条边颜色设置为空 其实就相当于抹掉这两条边
# 指定 data 设置的bottom(也就是指定的x轴)绑定到y轴的0

#标题
plt.title(r'x(t)=e^(1/2*t)*cos(4*t+pi)')

#幅度增长正弦信号
x=np.linspace(-5*np.pi,5*np.pi,5000)
y=np.exp(1/2*x)*np.cos(4*x+np.pi)
plt.plot(x,y,c='r')
plt.show()
```

基本连续信号成果如图



中间遇到的问题

1、最开始是想修改 y 轴的位置，也就是坐标轴中心，后来是百度到的代码如下：

```
ax = plt.gca()

ax.spines['right'].set_color('none')

ax.spines['top'].set_color('none')

ax.spines['bottom'].set_position(('data', 0))

ax.spines['left'].set_position(('data', 0))
```

实现虚化边框，将 x，y 轴移动到 00 对其。

2、之后是坐标轴标签和标题的设置

```
plt.xlabel('t/s')

plt.ylabel('X(t)')

plt.title()
```

3、最后是想对图进行美化，限制显示范围

```
plt.xlim()

plt.ylim()
```

4、画图时对 x 采样数太少的话，信号图会比较扭曲，所以要适当提高。