**各类基本信号图形显示**

# 一．背景与意义

在信号与系统学习过程中，各类基本信号图像是我们学习信号与系统的基础。首先正所谓万丈高楼平地起，再怎么强调基础的重要性都不为过，不仅对于学习知识领域，在很多方面都发挥了举足轻重的作用。没有基础，何来进阶，知识体系环环相扣，没有夯实的基础，我们的知识体系只会是漏洞百出，只懂表面，不懂原理，会做一题，稍稍变动又不会做了，学习任何事物想要学好必须学好基础，懂其原理，万丈高楼拔地而起，还要靠地基打的好，任何事物基础都很重要，更深奥的知识都是有最基础的知识，理论原理组合而成的，没有基础，就不可能去理解更深奥的知识理论，就不可能往更高的层次进阶，基础学好了扎实了才能再进阶更深奥的课程，要想有很大的成那么基础必须要学好。所以我们必须学好各类基本信号图形显示。

# 二．程序解读

import numpy as np是导入NUMPY作为NP，NumPy系统是Python的一种开源的数值计算扩展。这种工具可用来存储和处理大型矩阵，比Python自身的嵌套列表（nested list structure)结构要高效的多。

import matplotlib.pyplot as plt是操作matplotlib画折线图显示数据。Matplotlib是一个python的2D绘图库。

我们先画出xy坐标轴

ax = plt.gca() 是获得坐标轴对象

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ax.spines['right'].set\_color('none')  ax.spines['top'].set\_color('none')是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  因为matplotlib的基本元素包括x轴和y轴，以及水平和垂直的轴线，所以我们为了美观，先去掉两条轴线。  ax.xaxis.set\_ticks\_position('bottom')  ax.yaxis.set\_ticks\_position('left') 是指定下边的边作为 x 轴，指定左边的边为 y 轴  ax.spines['bottom'].set\_position(('data', 0)) 是指定 data 设置的bottom(也就是指定的x轴)绑定到y轴的0这个点上。  画出xy坐标轴后，我们利用相同的方法画出了单位冲激信号，单位阶跃信号，冲激偶信号，斜坡信号，抽样信号，方波脉冲信号，三角形脉冲信号，符号函数。  具体方法如下，以画单位冲激信号为例。   |  | | --- | |  | | def func11(): | |  | x = [0] | |  | y = [1] | |  |  |   plt.xlim((-2, 2)) 是设置X坐标轴的范围  plt.ylim((-1, 2)) 是设置Y坐标轴的范围  plt.xlabel('t') 是设置坐标轴的文字标签  plt.ylabel('deta(t)') 是设置坐标轴的文字标签  plt.title('单位冲激信号')是设置标题  plt.grid()是显示网格线  plt.scatter(x, y, color='r')画出散点图  plt.show()  然后我们在第九个程序中画了单位阶跃序列，单位样值序列，斜波序列，矩形序列四个子图，方法一样，以画单位样值序列为例。  因为在matplotlib中，整个图像为一个figure图像。在figure对象中可以包含一个或者多个axes对象，每个axes对象都是一个拥有自己坐标系统的绘图区域；这个程序中有四个函数，所以用ax1到ax4来表示，plt.subplots函数的作用是创建figure和axes的列表。  ax1 = plt.subplot(141)  ax1.set\_title('单位样值序列')  ax1.set\_xlim((-1.5, 3.5))  ax1.set\_ylim((-0.5, 4))  ax1.set\_xlabel('n')  ax1.set\_ylabel('Sigma(n)')  ax1.spines['right'].set\_color('none')  ax1.spines['top'].set\_color('none') 是将右边 上边的两条边颜色设置为空 其实就相当于抹掉这两条边  ax1.xaxis.set\_ticks\_position('bottom')  ax1.yaxis.set\_ticks\_position('left') 是指定下边的边作为 x 轴 指定左边的边为 y 轴  ax1.spines['bottom'].set\_position(('data', 0))是 指定 data 设置的bottom(也就是指定的x轴)绑定到y轴的0这个点上  ax1.spines['left'].set\_position(('data', 0))  x = [0] 设定x值为1  y = [1] 设定y值为1  ax1.scatter(x, y, color='r')  for i, xx in enumerate(x):把x,y结果一一枚举输出  ax1.plot([xx, xx], [0, y[i]], color='b', linewidth=2.5, linestyle="-")  if \_\_name\_\_ == "\_\_main\_\_":  setting()  # func11()  # func12()  # func13()  # func14()  # func15()  # func16()  # func17()  # func18()  func19()  最后利用if \_\_name\_\_ == '\_\_main\_\_'运行程序。\_\_name\_\_ 是当前模块名，当模块被直接运行时模块名为 \_\_main\_\_ 。这句话的意思就是，当模块被直接运行时，以下代码块将被运行，当模块是被导入时，代码块不被运行。  线条相关属性标记设置  用来该表线条的属性   | 线条风格linestyle或ls | 描述 | 线条风格linestyle或ls | 描述 | | --- | --- | --- | --- | | '-' | 实线 | ':' | 虚线 | | | '--' | 破折线 | 'None',' ','' | 什么都不画 | | | '-.' | 点划线 | |  |  |   线条标记   | 标记maker | 描述 | 标记 | 描述 | | | --- | --- | --- | --- | --- | | 'o' | 圆圈 | '.' | 点 |  | | 'D' | 菱形 | 's' | 正方形 |  | | 'h' | 六边形1 | '\*' | 星号 |  | | 'H' | 六边形2 | 'd' | 小菱形 |  | | '\_' | 水平线 | 'v' | 一角朝下的三角形 |  | | '8' | 八边形 | '<' | 一角朝左的三角形 |  | | 'p' | 五边形 | '>' | 一角朝右的三角形 |  | | ',' | 像素 | '^' | 一角朝上的三角形 |  | | '+' | 加号 | '\ | ' | 竖线 | | 'None','',' ' | 无 | 'x' | X |  |   颜色  可以通过调用matplotlib.pyplot.colors()得到matplotlib支持的所有颜色。   | 别名 | 颜色 | 别名 | 颜色 | | | --- | --- | --- | --- | --- | | b | 蓝色 | g | 绿色 |  | | r | 红色 | y | 黄色 |  | | c | 青色 | k | 黑色 | | | m | 洋红色 | w | 白色 |  | |
|  |  |

# 三．创新过程

在看到我们的课题用python画出各类基本信号图形，我们仔细翻阅了书籍，结合了各类基本信号图像的特点，决定使用Python第三方库matplotlib来绘制。（Matplotlib是一个Python 2D绘图库，它可以在各种平台上以各种硬拷贝格式和交互式环境生成出具有出版品质的图形。 Matplotlib可用于Python脚本，Python和IPython shell，Jupyter笔记本，Web应用程序服务器和四个图形用户界面工具包。Matplotlib试图让简单的事情变得更简单，让无法实现的事情变得可能实现。 只需几行代码即可生成绘图，直方图，功率谱，条形图，错误图，散点图等。）

# 四．实验结果分析

使用Python将基本信号图形显示出来，我们可以更直观地理解各类基本信号的特点与作用，在解释它们的关系时也可以直观比较，只要改动数值就可以观察信号的变化，对于理论和实践都有不小的帮助。

**五．结论**

利用Python将各类基本信号图形显示出来，这是一个非常重要的解决问题的方式。当我们遇到比较复杂的基本信号表达式时，使用Python以图形的方式显示出来，能更加直观的面对问题，对于我们的解题有着非常大的帮助。各类基本信号图形是信号与系统的础，也是重中之重。我们应当将这部分知识用各种方法掌握透彻，更利于以后的学习。