

**宁波理工学院**

**信号与系统实验报告**



姓 名 郭章良

学 号  **3190432055**

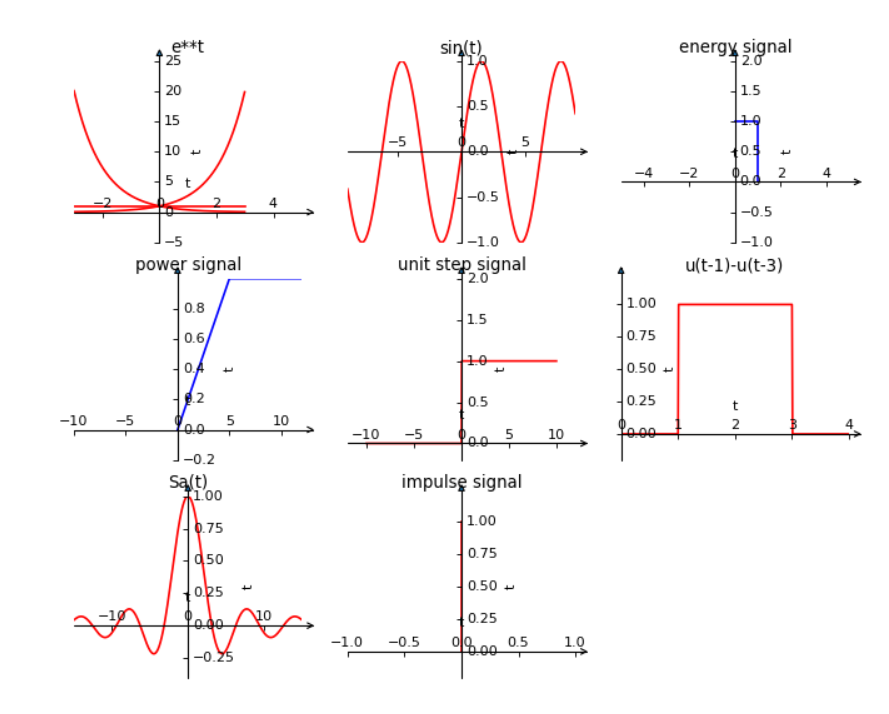
专业班级  **电信192**

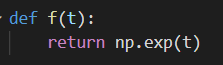
学 院 **信息工程与工程学院**

完成日期  **2020/10/13**

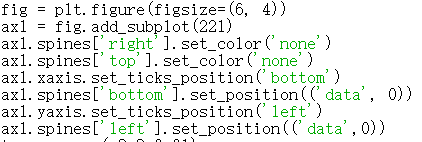
1. 代码：
2. import numpy as np
3. import matplotlib.pyplot as plt
4. import mpl\_toolkits.axisartist as axisartist
5. def f(t):
6. return np.exp(t)
7. def x(n):
8. return np.sin(n)
9. fig = plt.figure(figsize=(10, 8),dpi=80)
10. ax1= axisartist.Subplot(fig, 331)
11. fig.add\_axes(ax1)
12. ax1.axis[:].set\_visible(False)
13. ax1.axis["x"] = ax1.new\_floating\_axis(0,0)
14. ax1.axis["x"].set\_axisline\_style("->", size = 1.0)
15. ax1.axis["y"] = ax1.new\_floating\_axis(1,0)
16. ax1.axis["y"].set\_axisline\_style("-|>", size = 1.0)
17. ax1.axis["x"].set\_axis\_direction("top")
18. ax1.axis["y"].set\_axis\_direction("right")
19. plt.xlim(-3,5)
20. plt.ylim(-5,25)
21. t=np.arange(-3,3,0.01)
22. plt.plot(t,f(t),'r-')
23. plt.plot(t,f(-t),'r-')
24. plt.plot(t,f(0\*t),'r-')
25. plt.xlabel('t')
26. plt.title('e\*\*t')
27. ax2= axisartist.Subplot(fig, 332)
28. fig.add\_axes(ax2)
29. ax2.axis[:].set\_visible(False)
30. ax2.axis["x"] = ax2.new\_floating\_axis(0,0)
31. ax2.axis["x"].set\_axisline\_style("->", size = 1.0)
32. ax2.axis["y"] = ax2.new\_floating\_axis(1,0)
33. ax2.axis["y"].set\_axisline\_style("-|>", size = 1.0)
34. ax2.axis["x"].set\_axis\_direction("top")
35. ax2.axis["y"].set\_axis\_direction("right")
36. plt.xlim(-9,9)
37. plt.ylim(-1, 1)
38. n=np.arange(-9,9,0.01)
39. plt.plot(n,x(n),'r-')
40. plt.xlabel('t')
41. plt.title('sin(t)')
42. ax3= axisartist.Subplot(fig, 333)
43. fig.add\_axes(ax3)
44. ax3.axis[:].set\_visible(False)
45. ax3.axis["x"] = ax3.new\_floating\_axis(0,0)
46. ax3.axis["x"].set\_axisline\_style("->", size = 1.0)
47. ax3.axis["y"] = ax3.new\_floating\_axis(1,0)
48. ax3.axis["y"].set\_axisline\_style("-|>", size = 1.0)
49. ax3.axis["x"].set\_axis\_direction("top")
50. ax3.axis["y"].set\_axis\_direction("right")
51. plt.xlim(-5,5)
52. plt.ylim(-1, 2)
53. t1=[0,1,1]
54. t2=[1,1,0]
55. plt.xlabel('t')
56. plt.title('energy signal')
57. plt.plot(t1,t2,'b-')
58. ax4= axisartist.Subplot(fig, 334)
59. fig.add\_axes(ax4)
60. ax4.axis[:].set\_visible(False)
61. ax4.axis["x"] = ax4.new\_floating\_axis(0,0)
62. ax4.axis["x"].set\_axisline\_style("->", size = 1.0)
63. ax4.axis["y"] = ax4.new\_floating\_axis(1,0)
64. ax4.axis["y"].set\_axisline\_style("-|>", size = 1.0)
65. ax4.axis["x"].set\_axis\_direction("top")
66. ax4.axis["y"].set\_axis\_direction("right")
67. plt.xlim(-10,12)
68. plt.ylim(-0.2, 1)
69. x1=[0,5,13]
70. y1=[0,1,1]
71. plt.plot(x1,y1,'b-')
72. plt.xlabel('t')
73. plt.title('power signal')
74. def u(t):
75. return np.array(t>0,dtype=np.int)
76. ax5= axisartist.Subplot(fig, 335)
77. fig.add\_axes(ax5)
78. ax5.axis[:].set\_visible(False)
79. ax5.axis["x"] = ax5.new\_floating\_axis(0,0)
80. ax5.axis["x"].set\_axisline\_style("->", size = 1.0)
81. ax5.axis["y"] = ax5.new\_floating\_axis(1,0)
82. ax5.axis["y"].set\_axisline\_style("-|>", size = 1.0)
83. ax5.axis["x"].set\_axis\_direction("top")
84. ax5.axis["y"].set\_axis\_direction("right")
85. plt.xlim(-12,12)
86. plt.ylim(-0.2, 2)
87. t3=np.arange(-10,10,0.01)
88. plt.plot(t3,u(t3),'r-')
89. plt.xlabel('t')
90. plt.title('unit step signal')
91. ax6= axisartist.Subplot(fig, 336)
92. fig.add\_axes(ax6)
93. ax6.axis[:].set\_visible(False)
94. ax6.axis["x"] = ax6.new\_floating\_axis(0,0)
95. ax6.axis["x"].set\_axisline\_style("->", size = 1.0)
96. ax6.axis["y"] = ax6.new\_floating\_axis(1,0)
97. ax6.axis["y"].set\_axisline\_style("-|>", size = 1.0)
98. ax6.axis["x"].set\_axis\_direction("top")
99. ax6.axis["y"].set\_axis\_direction("right")
100. plt.xlim(0,4)
101. plt.ylim(-0.2, 1.2)
102. t4=np.arange(0,4,0.01)
103. plt.plot(t4,u(t4-1)-u(t4-3),'r-')
104. plt.xlabel('t')
105. plt.title('u(t-1)-u(t-3)')
106. def g(t):
107. return np.sin(t)/t
108. ax7= axisartist.Subplot(fig, 337)
109. fig.add\_axes(ax7)
110. ax7.axis[:].set\_visible(False)
111. ax7.axis["x"] = ax7.new\_floating\_axis(0,0)
112. ax7.axis["x"].set\_axisline\_style("->", size = 1.0)
113. ax7.axis["y"] = ax7.new\_floating\_axis(1,0)
114. ax7.axis["y"].set\_axisline\_style("-|>", size = 1.0)
115. ax7.axis["x"].set\_axis\_direction("top")
116. ax7.axis["y"].set\_axis\_direction("right")
117. plt.xlim(-15,15)
118. plt.ylim(-0.4, 1.0)
119. t5=np.arange(-15,15,0.01)
120. plt.plot(t5,g(t5),'r-')
121. plt.xlabel('t')
122. plt.title('Sa(t)')
123. ax8= axisartist.Subplot(fig, 338)
124. fig.add\_axes(ax8)
125. ax8.axis[:].set\_visible(False)
126. ax8.axis["x"] = ax8.new\_floating\_axis(0,0)
127. ax8.axis["x"].set\_axisline\_style("->", size = 1.0)
128. ax8.axis["y"] = ax8.new\_floating\_axis(1,0)
129. ax8.axis["y"].set\_axisline\_style("-|>", size = 1.0)
130. ax8.axis["x"].set\_axis\_direction("top")
131. ax8.axis["y"].set\_axis\_direction("right")
132. plt.xlim(-1.0,1.0)
133. plt.ylim(-0.2, 1.2)
134. t6=[0,0]
135. t7=[0,1]
136. plt.plot(t6,t7,'r-')
137. plt.title("impulse signal")
138. plt.xlabel('t')
139. plt.show()

输出截图



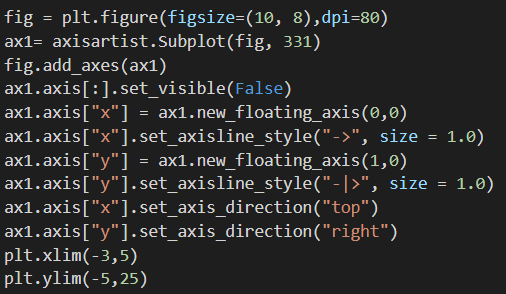
2、刚开始写代码时对于e\*\*t这个函数图像的画法没有思路，于是就找到了上学期学习python时的PPT，在其中一个PPT中，发现了老师是先用函数，把想要画的图对应的函数写成独立的模块，我就模仿老师的做法，因为python中无法输入自然常数e，自己也没有好的办法，就在百度上搜了一下，发现是通过调用已有的函数完成的，于是就写出了

第一次写画图函数代码时没有考虑到坐标轴，后来与同学交流时注意到了坐标轴，在百度中搜到了下图的代码

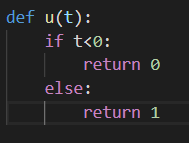


在电脑上运行成功，接着又查了一下add\_subplot函数的使用方法以及以上代码的大致含义

后来发现坐标轴没有箭头，就在百度上搜了一下坐标轴加上箭头的方法，就搜到了下图所示的代码，配合代码旁边的注释，大致了解了下图代码的含义



接着，我尝试着画多个子图，就把subplot这个函数里的参数进行了修改，在写单位阶跃函数时开始写了下面的代码：



电脑一直报错，在多次尝试没有成功后，在百度上搜了一下阶跃函数的python代码，改为了下图：



先计算t>0的布尔值，再转换为int类型

最后，发现在给坐标轴进行标注时图像显示并不清楚，就把plt.ylabel()改为了plt.title()

保留了plt.xlabel()，运行并上传了代码