

1. **print**("==============================================")
3. #实验一 计算
4. **print**("实验1数据")
5. # 导入所需库
6. **import** pandas as pd
7. **import** math
9. # 定义数据：测量编号、振幅θ(deg)、周期T₀(sec)
10. data = {
11. "Ex1\_No.": list(range(1, 21)),
12. "Ex1\_θ(deg)": [120, 125, 134, 133, 129, 137, 140, 139, 134, 132, 130, 129, 129, 128, 126, 122, 119, 117, 113, 112],
13. "Ex1\_T₀(sec)": [1.572, 1.572, 1.571, 1.571, 1.571, 1.571, 1.571, 1.571, 1.570, 1.570, 1.570, 1.570, 1.570, 1.570, 1.570,
14. 1.570, 1.569, 1.569, 1.569, 1.569]
15. }
17. # 将数据转换为DataFrame格式
18. df = pd.DataFrame(data)
20. # 计算T₀的平均值和标准偏差
21. average\_T0 = df["Ex1\_T₀(sec)"].mean()
22. std\_T0 = df["Ex1\_T₀(sec)"].std()
24. # 计算自然频率ω₀
25. omega\_0 = 2 \* math.pi / average\_T0
27. # 输出计算结果
28. **print**(f"平均自然周期 T₀: {average\_T0:.4f} sec")
29. **print**(f"T₀的标准偏差: {std\_T0:.5f} sec")
30. **print**(f"自然频率 ω₀: {omega\_0:.4f} rad/sec")
32. **print**("==============================================")
34. #实验2
35. **print**("实验2数据")
36. # 导入所需库
37. **import** pandas as pd
39. # 定义实验二的振幅数据和对数衰减值
40. experiment\_data = {
41. "theta\_i": [139, 127, 116, 107, 98],  # 前五个振幅值
42. "ln\_decay": [0.434664263, 0.437468, 0.436102, 0.438722, 0.441833]
43. }
45. # 将数据转换为DataFrame格式
46. df\_experiment = pd.DataFrame(experiment\_data)
48. # 假设 average\_T0 是之前实验计算出的平均周期T₀
49. average\_T0 = 1.5703  # 示例值，应使用实际计算结果
51. # 根据更新的公式计算每组数据的阻尼系数β
52. df\_experiment["Ex2\_beta"] = df\_experiment["ln\_decay"] / (5 \* average\_T0)
54. # 计算阻尼系数β的平均值
55. beta\_average\_updated = df\_experiment["Ex2\_beta"].mean()
57. # 输出β的平均值和完整的实验数据
58. **print**(f"阻尼系数 β 的平均值: {beta\_average\_updated:.5f}")
59. **print**(df\_experiment)

62. **print**("==============================================")
64. #实验3
65. **print**("实验3数据")

68. **import** pandas as pd
69. **import** numpy as np
71. # 使用实验1中的自然频率 omega\_0 和实验2中的阻尼系数 beta\_average
72. omega\_0 = 4.0013  # 这个值需要你从实验1的结果中获取
73. beta\_average = 0.05575  # 这个值需要你从实验2的结果中获取
75. # 实验3的数据，根据你的实际数据进行了填充
76. experiment\_3\_data = {
77. "potentiometer\_scale": np.arange(0, 10.5, 0.5),  # 电位器刻度
78. "T\_sec": [1.5101, 1.5129, 1.5137, 1.5253, 1.5323, 1.5359, 1.5449, 1.5516, 1.5570, 1.5638, 1.5703, 1.5769, 1.5839, 1.5902, 1.5984, 1.6031, 1.6087, 1.6101, 1.6232, 1.6299, 1.6381],  # 电机振动周期T (sec)
79. "theta\_deg": [34, 34, 36, 40, 44, 49, 56, 62, 71, 83, 101, 117, 128, 131, 126, 119, 107, 96, 85, 77, 77],  # 振幅θ (deg)
80. "phi\_deg": [168, 167, 167, 165, 164, 162, 160, 155, 153, 148, 142, 128, 114, 90, 81, 72, 60, 55, 46, 40, 39],  # 相位φ (deg)
81. }
83. # 创建DataFrame
84. df\_experiment\_3 = pd.DataFrame(experiment\_3\_data)
86. # 计算每个刻度下的驱动频率ω
87. df\_experiment\_3['omega'] = 2 \* np.pi / df\_experiment\_3['T\_sec']

90. # 计算稳态振动振幅θs
91. df\_experiment\_3['theta\_s'] = 1 / np.sqrt((omega\_0\*\*2 - df\_experiment\_3['omega']\*\*2)\*\*2 +
92. (4 \* beta\_average\*\*2 \* df\_experiment\_3['omega']\*\*2))
94. # 将振幅从弧度转换为度
95. df\_experiment\_3['theta\_s\_deg'] = np.degrees(df\_experiment\_3['theta\_s'])


99. # 打印结果
101. #print(df\_experiment\_3[['potentiometer\_scale', 'T\_sec', 'theta\_deg', 'phi\_deg', 'omega', 'theta\_s\_deg']])
103. T\_average = np.mean(df\_experiment\_3['T\_sec']
104. )
106. # 根据T的平均值计算omega (驱动频率)
107. omega = 2 \* np.pi / T\_average
109. # 打印结果
110. **print**(f"电机振动周期T的平均值: {T\_average} sec")
111. **print**(f"根据平均值计算出的omega (驱动频率): {omega} rad/sec")

114. **print**("========================")
116. **import** math
118. # 已知的自然频率和阻尼系数
119. omega\_0 = 4.0013  # 自然频率 (rad/sec)
120. beta = 0.05575  # 阻尼系数
122. # 计算 omega(r)
123. omega\_r = math.sqrt(omega\_0\*\*2 - 2\*beta\*\*2)
125. # 输出结果
126. **print**("omega(r):", omega\_r)

129. df\_experiment\_3['omega\_R'] = omega\_r
130. df\_experiment\_3['omega / omega\_r'] = df\_experiment\_3['omega'] / omega\_r
131. **print**(df\_experiment\_3[['potentiometer\_scale', 'T\_sec', 'theta\_deg', 'phi\_deg', 'omega', 'omega / omega\_r']])


135. # 生成幅频特性曲线和相频特性曲线
136. plt.figure(figsize=(12, 6))
138. plt.subplot(1, 2, 1)
139. plt.plot(df\_experiment\_3['omega']/df\_experiment\_3['omega\_R'], df\_experiment\_3['theta\_s\_deg'], 'r-')
140. plt.title('Amplitude Frequency Characteristics')
141. plt.xlabel('Frequency ratio (omega/omega\_r)')
142. plt.ylabel('Amplitude')
144. plt.subplot(1, 2, 2)
145. plt.plot(df\_experiment\_3['omega']/df\_experiment\_3['omega\_R'], -(df\_experiment\_3['phi\_deg']), 'b-')
146. plt.title('Phase Frequency Characteristics')
147. plt.xlabel('Frequency ratio (omega/omega\_r)')
148. plt.ylabel('Phase (degrees)')

151. # 在图表中添加纵坐标为1.0的水平线
152. plt.axhline(y=1.0, color='k', linestyle='--', label='y=1.0')
154. # 显示图例
155. plt.legend()
157. plt.tight\_layout()
158. plt.show()