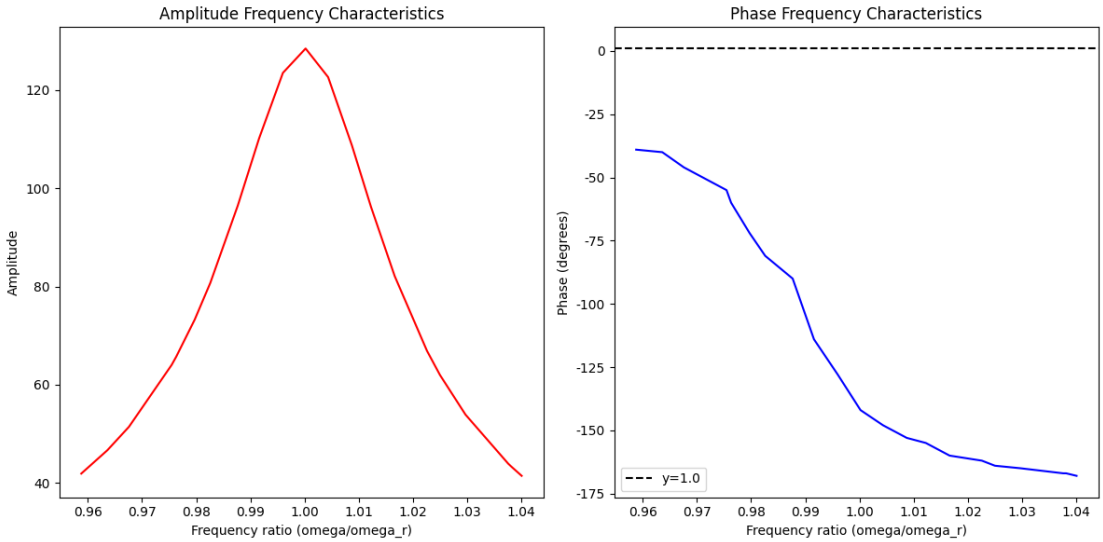


```python
print("================================================")

#实验一 计算
print("实验 1 数据")
# 导入所需库
import pandas as pd
import math

# 定义数据：测量编号、振幅 θ(deg)、周期 T₀(sec)
data = {
    "Ex1_No.": list(range(1, 21)),
    "Ex1_θ(deg)": [120, 125, 134, 133, 129, 137, 140, 139, 134, 132, 130, 129, 129, 128, 126, 122, 119, 117, 113, 112],
    "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,
                    1.570, 1.569, 1.569, 1.569, 1.569]
}

# 将数据转换为 DataFrame 格式
df = pd.DataFrame(data)

# 计算 T₀的平均值和标准偏差
average_T0 = df["Ex1_T₀(sec)"].mean()
std_T0 = df["Ex1_T₀(sec)"].std()

# 计算自然频率 ω₀
omega_0 = 2 * math.pi / average_T0
```

```
26.
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")
31.
32. print("============================================")
33.
34. #实验 2
35. print("实验 2 数据")
36. # 导入所需库
37. import pandas as pd
38.
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. }
44.
45. # 将数据转换为 DataFrame 格式
46. df_experiment = pd.DataFrame(experiment_data)
47.
48. average_T0 = 1.5703
49.
50. # 根据公式计算每组数据的阻尼系数 β
51. df_experiment["Ex2_beta"] = df_experiment["ln_decay"] / (5 * average_
    T0)
52.
53. # 计算阻尼系数 β 的平均值
54. beta_average_updated = df_experiment["Ex2_beta"].mean()
55.
56. # 输出 β 的平均值和完整的实验数据
57. print(f"阻尼系数 β 的平均值: {beta_average_updated:.5f}")
58. print(df_experiment)
59.
60.
61. print("============================================")
62.
63. #实验 3
64. print("实验 3 数据")
65.
66.
67. import pandas as pd
```

```
import numpy as np

# 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
omega_0 = 4.0013  # 从实验 1 的结果中获取
beta_average = 0.05575  # 从实验 2 的结果中获取

# 实验 3 的数据，根据你的实际数据进行了填充
experiment_3_data = {
    "potentiometer_scale": np.arange(0, 10.5, 0.5),  # 电位器刻度
    "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)
    "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)
    "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)
}

# 创建 DataFrame
df_experiment_3 = pd.DataFrame(experiment_3_data)

# 计算每个刻度下的驱动频率 ω
df_experiment_3['omega'] = 2 * np.pi / df_experiment_3['T_sec']


# 计算稳态振动振幅 θs
df_experiment_3['theta_s'] = 1 / np.sqrt((omega_0**2 - df_experiment_3['omega']**2)**2 +
                                         (4 * beta_average**2 * df_experiment_3['omega']**2))

# 将振幅从弧度转换为度
df_experiment_3['theta_s_deg'] = np.degrees(df_experiment_3['theta_s'])


# 打印结果


  T_average = np.mean(df_experiment_3['T_sec']
  )

  # 根据 T 的平均值计算 omega (驱动频率)
```

```
105.  omega = 2 * np.pi / T_average
106.
107.  # 打印结果
108.  print(f"电机振动周期 T 的平均值: {T_average} sec")
109.  print(f"根据平均值计算出的 omega (驱动频率): {omega} rad/sec")
110.
111.
112.  print("========================")
113.
114.  import math
115.
116.  # 已知的自然频率和阻尼系数
117.  omega_0 = 4.0013  # 自然频率 (rad/sec)
118.  beta = 0.05575  # 阻尼系数
119.
120.  # 计算 omega(r)
121.  omega_r = math.sqrt(omega_0**2 - 2*beta**2)
122.
123.  # 输出结果
124.  print("omega(r):", omega_r)
125.
126.
127.  df_experiment_3['omega_R'] = omega_r
128.  df_experiment_3['omega / omega_r'] = df_experiment_3['omega'] / om
     ega_r
129.  print(df_experiment_3[['potentiometer_scale', 'T_sec', 'theta_deg'
     , 'phi_deg', 'omega', 'omega / omega_r']])
130.
131.
132.
133.  # 生成幅频特性曲线和相频特性曲线
134.  plt.figure(figsize=(12, 6))
135.
136.  plt.subplot(1, 2, 1)
137.  plt.plot(df_experiment_3['omega']/df_experiment_3['omega_R'], df_e
     xperiment_3['theta_s_deg'], 'r-')
138.  plt.title('Amplitude Frequency Characteristics')
139.  plt.xlabel('Frequency ratio (omega/omega_r)')
140.  plt.ylabel('Amplitude')
141.
142.  plt.subplot(1, 2, 2)
143.  plt.plot(df_experiment_3['omega']/df_experiment_3['omega_R'], -
     (df_experiment_3['phi_deg']), 'b-')
144.  plt.title('Phase Frequency Characteristics')
```

```
plt.xlabel('Frequency ratio (omega/omega_r)')
plt.ylabel('Phase (degrees)')


# 在图表中添加纵坐标为 1.0 的水平线
plt.axhline(y=1.0, color='k', linestyle='--', label='y=1.0')

# 显示图例
plt.legend()

plt.tight_layout()
plt.show()
```

```
==============================================
实验 1 数据
平均自然周期 T₀: 1.5703 sec
T₀的标准偏差: 0.00092 sec
自然频率 ω₀: 4.0013 rad/sec
==============================================
实验 2 数据
阻尼系数 β 的平均值: 0.05575
   theta_i  ln_decay  Ex2_beta
0      139  0.434664  0.055361
1      127  0.437468  0.055718
2      116  0.436102  0.055544
3      107  0.438722  0.055877
4       98  0.441833  0.056274
==============================================
实验 3 数据
电机振动周期 T 的平均值: 1.570490476190476 sec
根据平均值计算出的 omega (驱动频率): 4.000778993846973 rad/sec
========================
omega(r): 4.000523161412767
    potentiometer_scale   T_sec  theta_deg  phi_deg  omega      omega / omega_r
0                   0.0  1.5101         34      168  4.160774         1.040058
1                   0.5  1.5129         34      167  4.153074         1.038133
2                   1.0  1.5137         36      167  4.150879         1.037584
3                   1.5  1.5253         40      165  4.119311         1.029693
4                   2.0  1.5323         44      164  4.100493         1.024989
5                   2.5  1.5359         49      162  4.090882         1.022587
6                   3.0  1.5449         56      160  4.067050         1.016629
7                   3.5  1.5516         62      155  4.049488         1.012240
8                   4.0  1.5570         71      153  4.035443         1.008729
```

9	4.5	1.5638	83	148	4.017896	1.004343
10	5.0	1.5703	101	142	4.001264	1.000185
11	5.5	1.5769	117	128	3.984517	0.995999
12	6.0	1.5839	128	114	3.966908	0.991597
13	6.5	1.5902	131	90	3.951192	0.987669
14	7.0	1.5984	126	81	3.930922	0.982602
15	7.5	1.6031	119	72	3.919397	0.979721
16	8.0	1.6087	107	60	3.905753	0.976311
17	8.5	1.6101	96	55	3.902357	0.975462
18	9.0	1.6232	85	46	3.870863	0.967589
19	9.5	1.6299	77	40	3.854951	0.963612
20	10.0	1.6381	77	39	3.835654	0.958788