西 安 电 子 科 技 大 学

数字信号处理 课程实验报告

实验名称 系统的频域和 z 域分析

人工智能 学院 1920012 班

姓名 杨文韬 学号 18020100245

姓名 刘浩 学号 19069100088

姓名 周泽熙 学号 19069100126

实验日期 2021 年 11 月 05 日

成 绩

指导教师评语：

指导教师：

年 月 日

实验报告内容基本要求及参考格式

一、实验目的

二、实验基本原理及步骤

三、实验仿真结果与分析

四、实验中遇到的问题及解决方法（至少 3 个，每人至少写 1 个，写清楚谁的问题和解决方法）

目录

系统的频域和 z 域分析

1 实验目的

设计计算机程序，产生序列并计算序列的 DTFT，绘制其幅频特性和相频特性曲线；根据系统的单位脉冲响应和差分方程，计算系统的频率响应，绘制系统频率响应的幅频特性和相频特性曲线；根据系统的单位脉冲响应和差分方程，计算系统的系统系统函数、零极点分布；改变系统的零级点分布，观察系统频率响应的变化。

2 实验原理

2.1 序列的离散时间傅里叶变换

2.1.1 DTFT 的定义

一般序列 $x(n)$ 的 DTFT(discrete time Fourier transform) 定义为

$$X(e^{j\omega}) = \mathrm{DTFT}[x(n)] = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

$X(e^{j\omega})$ 是序列 $x(n)$ 的频谱函数。上式的级数不一定总是收敛的，例如，$x(n)$ 是单位阶跃序列时级数就不收敛。序列 $x(n)$ 的 DTFT 存在的充分必要条件是序列 $x(n)$ 绝对可和，即满足

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty$$

2.1.2 DTFT 的周期性

序列的 DTFT 定义式中，n 取整数，因此

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j(\omega+2\pi M)n}, \quad M \in \mathbb{Z}$$

成立。这说明序列的 DTFT 是频率 ω 的连续周期函数，周期为 2π。由于 DTFT 的周期性，只要知道 $X(e^{j\omega})$ 的一个周期，即 $\omega \in [0, 2\pi)$ 或 $\omega \in [-\pi, \pi)$，就可以分析序列的频谱，不需要取整个 $-\infty < \omega < \infty$ 域来分析。在 $\omega = 0, 2\pi M$ 点上，$X(e^{j\omega})$ 表示序列 $x(n)$ 的低频分量，序列 $x(n)$ 的最高频率分量在 $\omega = \pi$ 点上。

一般来说，$X(e^{j\omega})$ 是实变量 ω 的复值函数，可用实部和虚部将其表示为

$$X(e^{j\omega}) = X_R(e^{j\omega}) + jX_I(e^{j\omega})$$

其中，$X_R(e^{j\omega})$、$X_I(e^{j\omega})$ 分别是 $X(e^{j\omega})$ 的实部和虚部。

$X(e^{j\omega})$ 也可以用幅度谱和相位谱表示为

$$X(e^{j\omega}) = |X(e^{j\omega})|e^{j\arg[X(e^{j\omega})]}$$

其中，$|X(e^{j\omega})$、$\arg[X(e^{j\omega})]$ 分别称为序列 $x(n)$ 的幅度谱和相位谱。

2.2 离散时间系统 LTI 系统的频率响应

离散时间 LTI 系统的频域特性可用系统的频率响应和系统函数进行分析。

当系统的输入是频率为 ω 的复指数序列

$$x(n) = e^{j\omega n}$$

时, 系统的零状态响应为

$$y(n) = e^{j\omega n} * h(n) = e^{j\omega n}H(e^{j\omega})$$

式中

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n} = \mathrm{DTFT}[h(n)]$$

$H(e^{j\omega})$ 定义为离散时间 LTI 系统的频率响应。由上式可知，复指数序列 $e^{j\omega n}$ 通过离散时间 LTI 系统后输出序列的频率不变，序列的幅度由系统的频率响应 $H(e^{j\omega})$ 在 ω 点的幅度值确定。所以 $|H(e^{j\omega})|$ 表示系统对不同频率信号的增益。

在一般情况下，离散时间系统 LTI 系统的频率响应 $H(e^{j\omega})$ 是复值函数，可用幅度和相位表示

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\phi(\omega)}$$

式中，$|H(e^{j\omega})|$ 被称为系统的幅频响应，$\phi(\omega)$ 称为系统的相频响应。当 $h(n)$ 为实序列时，由序列的 DTFT 性质可知，$|H(e^{j\omega})|$ 是 ω 的偶函数，$\phi(\omega)$ 为 ω 的奇函数。

3 实验过程

实验用到了 MATLAB 和 Python 两种语言版本，具体的说明在 MATLAB 版本中，Python 版实验环境为 Jupyter Notebook，导入的包如下

```python
import matplotlib.pyplot as plt
import numpy as np
from scipy import signal
```

3.1 序列的 DTFT

3.1.1 MATLAB 版本

我们的序列为

$$x(n)=\begin{cases}\cos(\frac{\pi}{8}n) & n=-16,-15,\cdots,15,16\\ 0 & others\end{cases}$$

可以视为在区间 $[-2\pi,2\pi]$ 对模拟信号 $y(t)=\cos(t)$ 以采样周期 $T=\frac{\pi}{8}$ 采样并将横坐标转换为 n 获得，如图 1 所示。

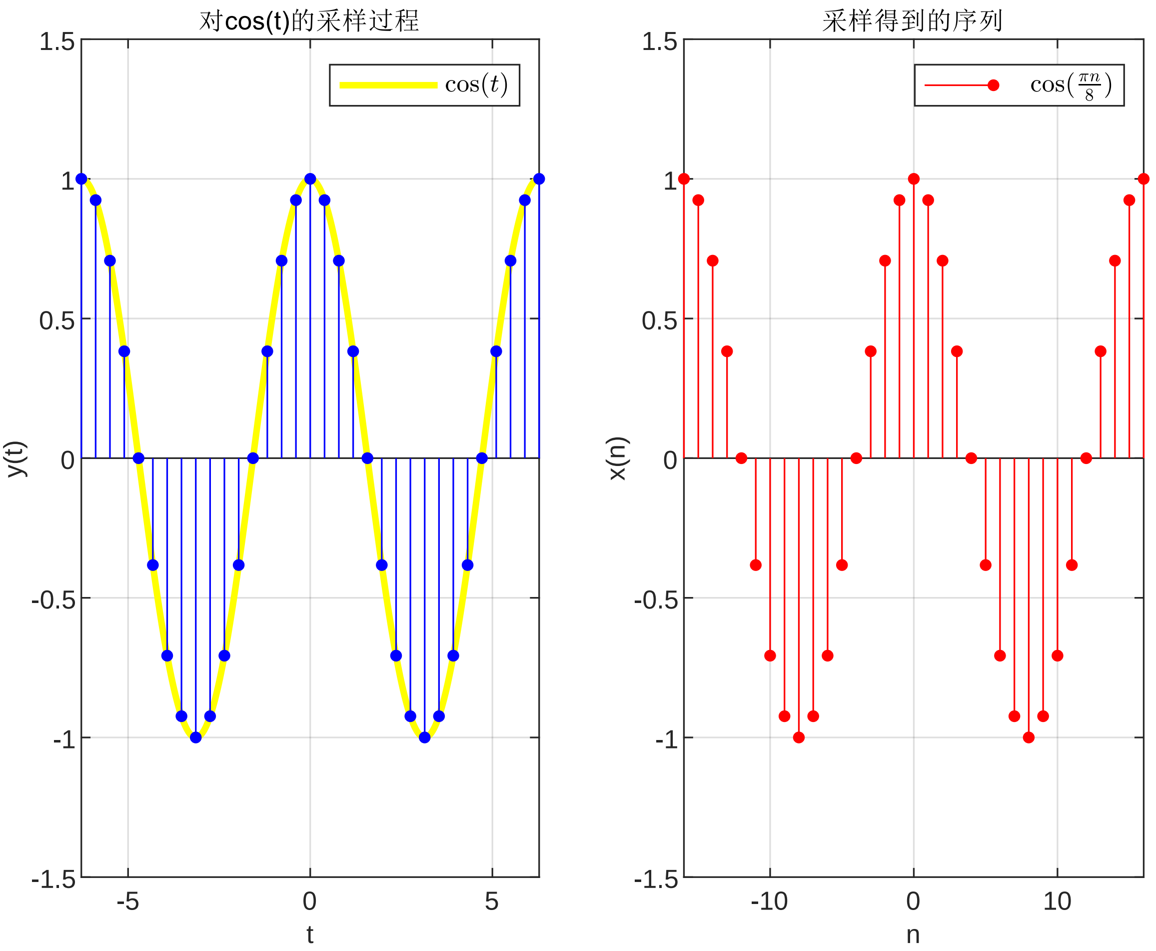

图 1: 序列的产生

```
subplot(1,2,1);
t = -2*pi:0.001:2*pi;
nt = -2*pi:2*pi/16:2*pi;
plot(t,cos(t),'y','LineWidth',2); hold on;
stem(nt,cos(nt),'b','filled','MarkerSize',3)
grid on;
```

```
axis([-2*pi 2*pi -1.5 1.5])
title('对cos(t)的采样过程');
xlabel('t');
ylabel('y(t)');
handle = legend('$\cos(t)$');
set(handle,'Interpreter','latex','FontSize',10)

subplot(1,2,2);
n = -16:16;
xn = cos(pi*n/8);
stem(n,xn,'r','filled','MarkerSize',3);
grid on;
axis([-16 16 -1.5 1.5])
title('采样得到的序列');
xlabel('n');
ylabel('x(n)');
handle = legend('$\cos(\frac{\pi n}{8})$');
set(handle,'Interpreter','latex','FontSize',10)
```

绘制得到的幅频特性和相频特性曲线如图 2 所示，实部和虚部曲线如图 3 所示。

```
n = -16:16;
xn = cos(pi*n/8);
omega = linspace(0,2*pi,1000);
X = xn * exp(-j*n'*omega); % DTFT

figure(1)

subplot(1,2,1)
plot(omega,abs(X),'g')
grid on;

xlim([0 2*pi])
title('幅频特性');
xlabel('\omega');
ylabel('|X(e^{j\omega})|');

subplot(1,2,2)
plot(omega,angle(X),'b')
grid on;
title('相频特性');
xlabel('\omega');
ylabel('arg[X(e^{j\omega})]');
xlim([0 2*pi])

figure(2)

```

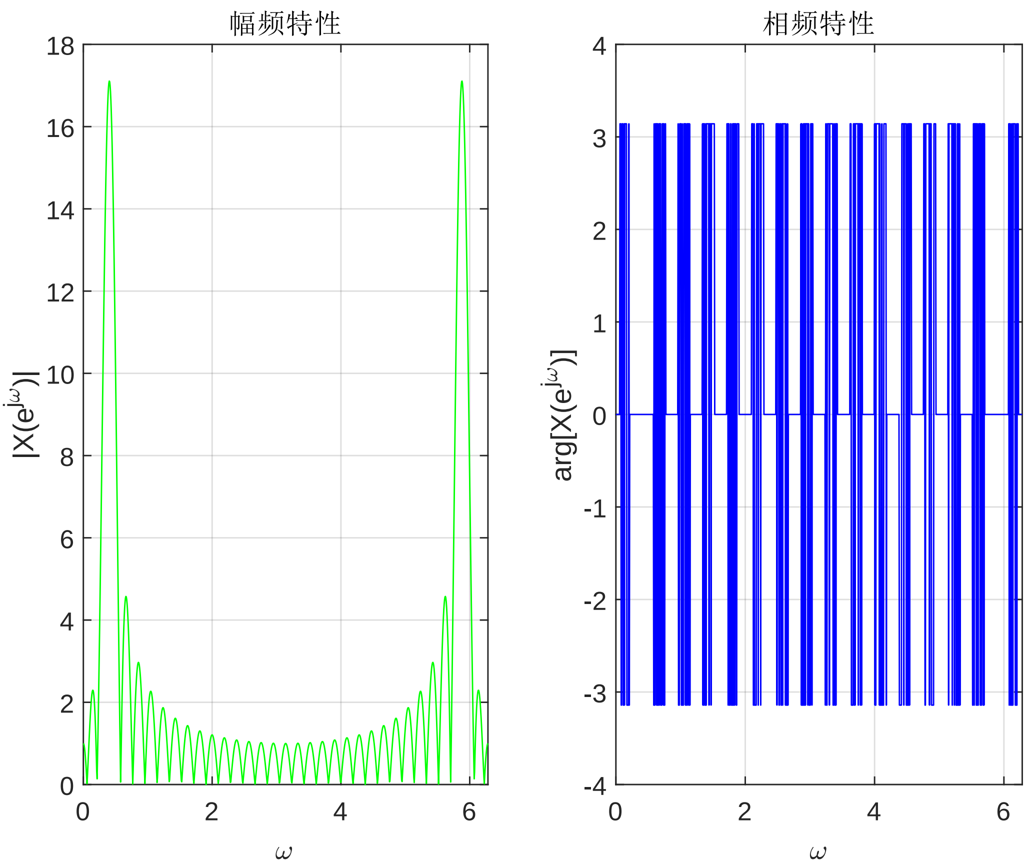

图 2: 幅频特性和相频特性

```
subplot(1,2,1);
plot(omega,real(X),'y')
grid on;
xlim([0 2*pi])
title('实部');
xlabel('\omega');
ylabel('X_R(e^{j\omega})');

subplot(1,2,2);
plot(omega,imag(X),'m')
grid on;
xlim([0 2*pi])
title('虚部');
xlabel('\omega');
ylabel('X_I(e^{j\omega})');
```

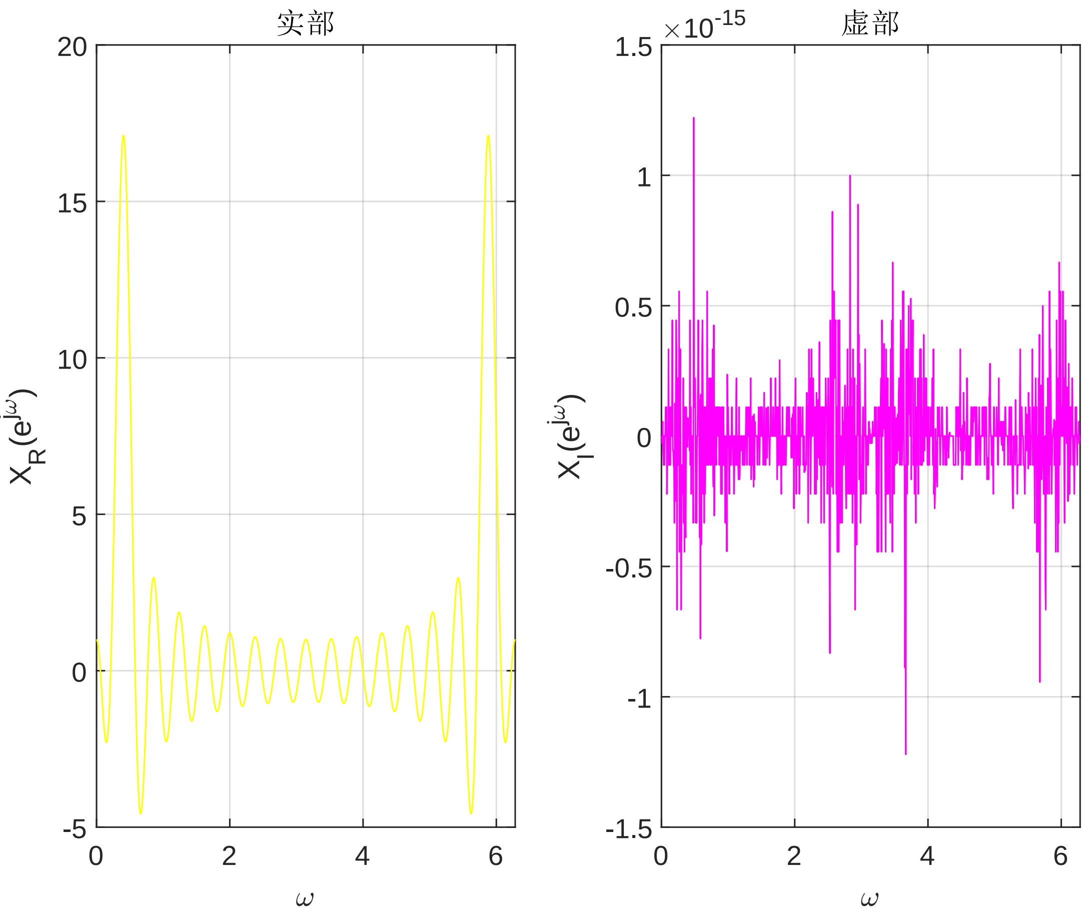

图 3: DTFT 的实部和虚部

3.1.2 Python 版本

用 Python 绘制得到的幅频特性和相频特性曲线如图 4 所示，实部和虚部曲线如图 3 所示。

```
# 计算 DTFT
n = np.arange(-16,17)
n = np.mat(n)
xn = np.cos(np.pi*n/8)
omega = np.arange(0, 2*np.pi, 2*np.pi/1000)
X = xn * np.exp(-1j*n.T*omega)
X = X.T
```

```
# 绘制幅频特性和相频特性曲线
plt.figure(figsize=(8, 6), dpi=80)

plt.subplot(121)
plt.plot(omega, np.abs(X), 'g', linewidth=1.0)
```

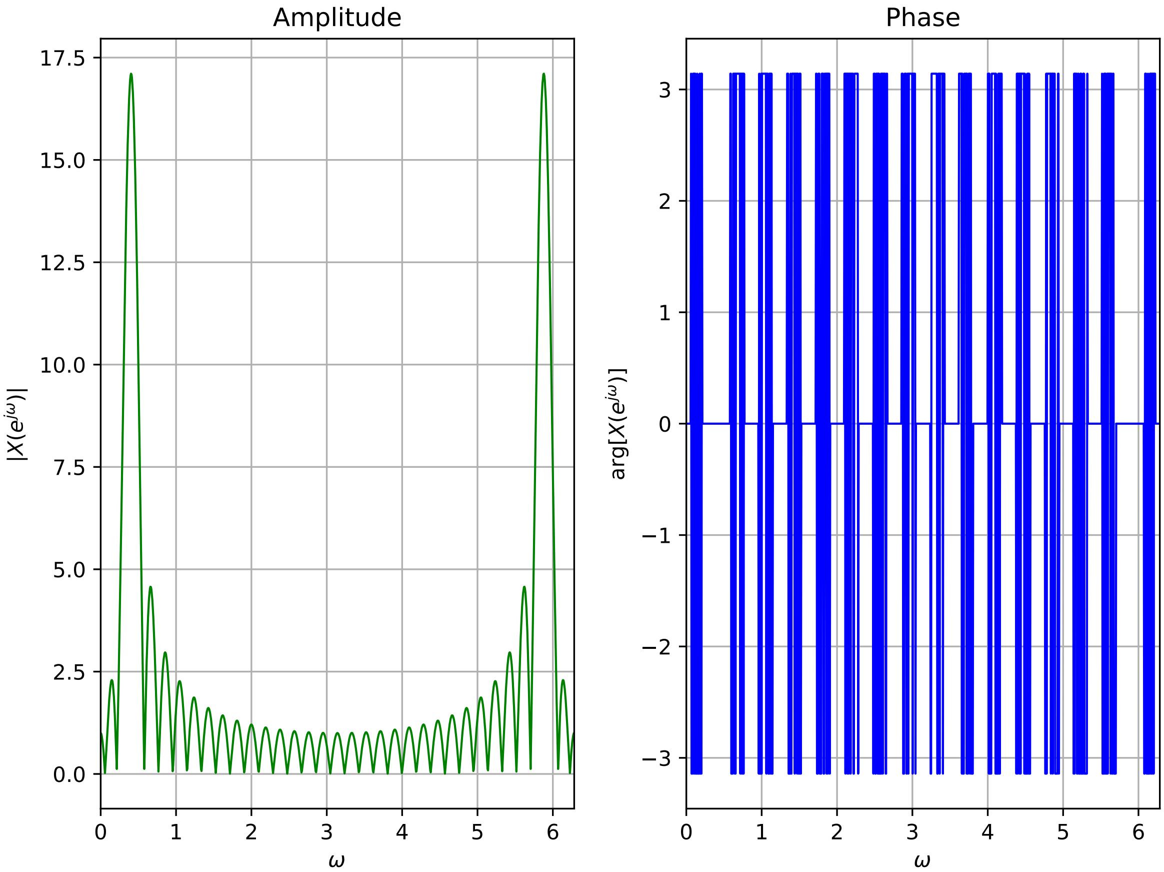

图 4: 幅频特性和相频特性 (Python)

```
plt.xlim(0, 2*np.pi)
plt.grid()
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')
plt.title('Amplitude')

plt.subplot(122)
plt.plot(omega, np.angle(X), 'b', linewidth=1.0)
plt.xlim(0, 2*np.pi)
plt.grid()
plt.xlabel('$\omega$')
plt.ylabel('$\mathrm{arg}[X(e^{j\omega})]$')
plt.title('Phase')

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

```
# 绘制实部和虚部曲线
plt.figure(figsize=(8, 6), dpi=80)

plt.subplot(121)
plt.plot(omega, np.real(X), 'y', linewidth=1.0)
```

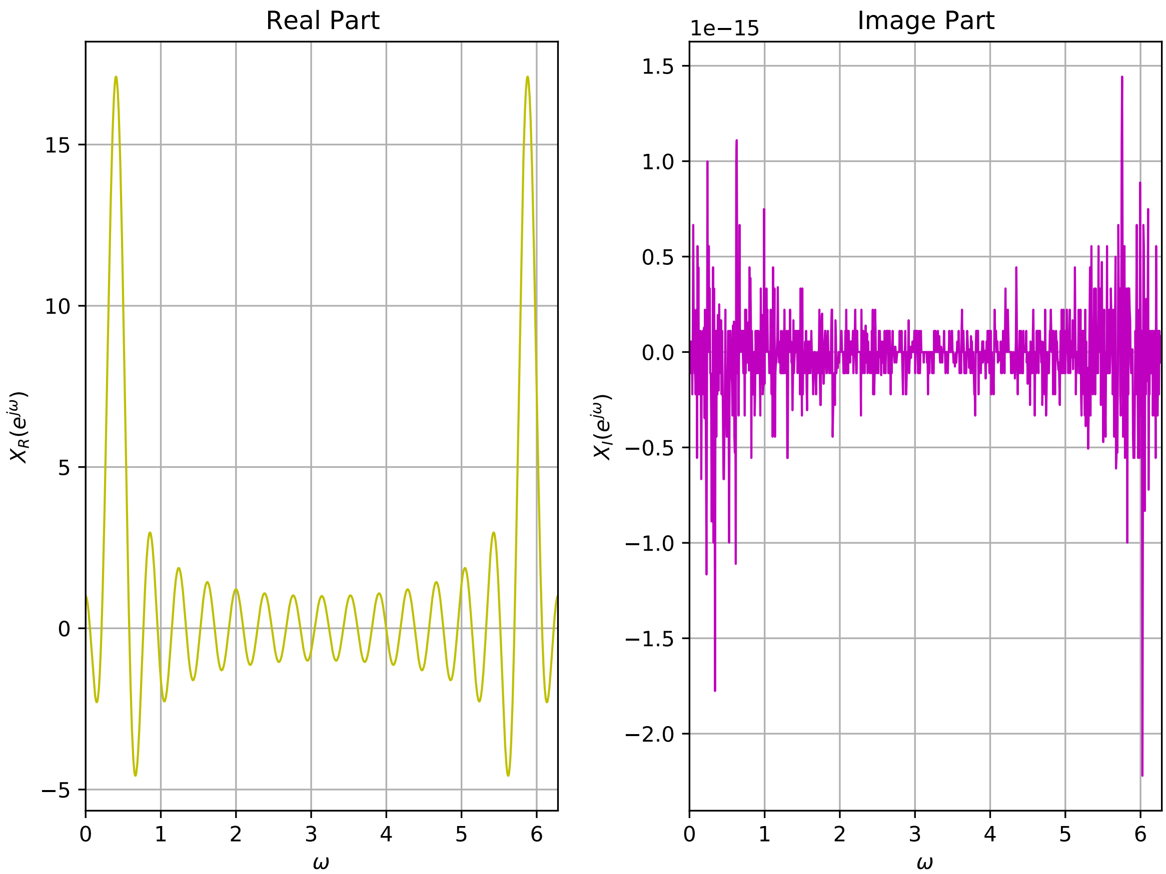

图 5: DTFT 的实部和虚部 (Python)

```python
plt.xlim(0, 2*np.pi)
plt.grid()
plt.xlabel('$\omega$')
plt.ylabel('$X_R(e^{j\omega})$')
plt.title('Real Part')

plt.subplot(122)
plt.plot(omega, np.imag(X), 'm', linewidth=1.0)
plt.xlim(0, 2*np.pi)
plt.grid()
plt.xlabel('$\omega$')
plt.ylabel('$X_I(e^{j\omega})$')
plt.title('Image Part')

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

3.2 离散 LTI 系统的频率响应

3.2.1 MATLAB 版本

我们建立如下一阶系统差分方程

$$y(n) - 0.6y(n-1) = x(n)$$

其系统函数为

$$H(z) = \frac{z}{z - 0.6}$$

系统的单位脉冲响应如下，如图 6 所示。

$$h(n) = (0.6)^n u(n)$$

系统函数的零点 $z = 0$，极点 $p = 0.6$，如图 7 所示。幅频响应特性和相频响应特性如图 8 所示。

通过改变系统的零极点分布我们发现，当 ω 从 0 变化到 2π 时，对应单位圆上的点 B，当 B 点转到极点附近时，该极点矢量长度短，因而幅频响应出现峰值，且极点越靠近单位圆，极点矢量长度越短，峰值越高越尖锐。如果极点在单位圆上，该极点对应的幅频响应无穷大，系统是不稳定的，这与稳定系统的收敛域要包含单位圆的条件是一致的。对于零点，结果相反，当 B 点转到零点附近时，该零点矢量长度最短，幅频响应将出现谷值，零点愈靠近单位圆，谷值愈接近零。当零点处在单位圆上时，谷值为 0。

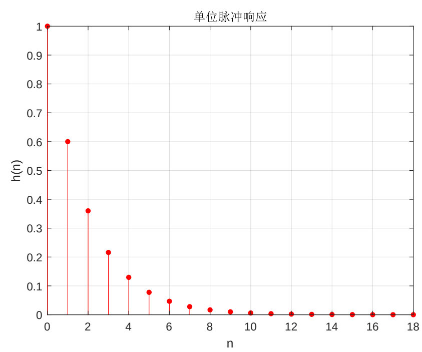

图 6: 单位脉冲响应

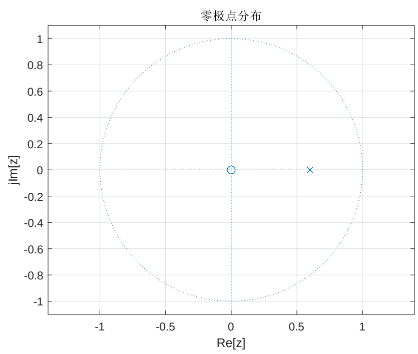

图 7: 零极点分布图

```
b=[1];
a=[1 -0.6];
[h,n]=impz(b,a); % 单位冲激响应
[H,omega]=freqz(b,a,'whole');
H1=abs(H); % 幅频特性
H2=angle(H); % 相频特性

figure;
stem(n,h,'r','filled','MarkerSize',4)
```

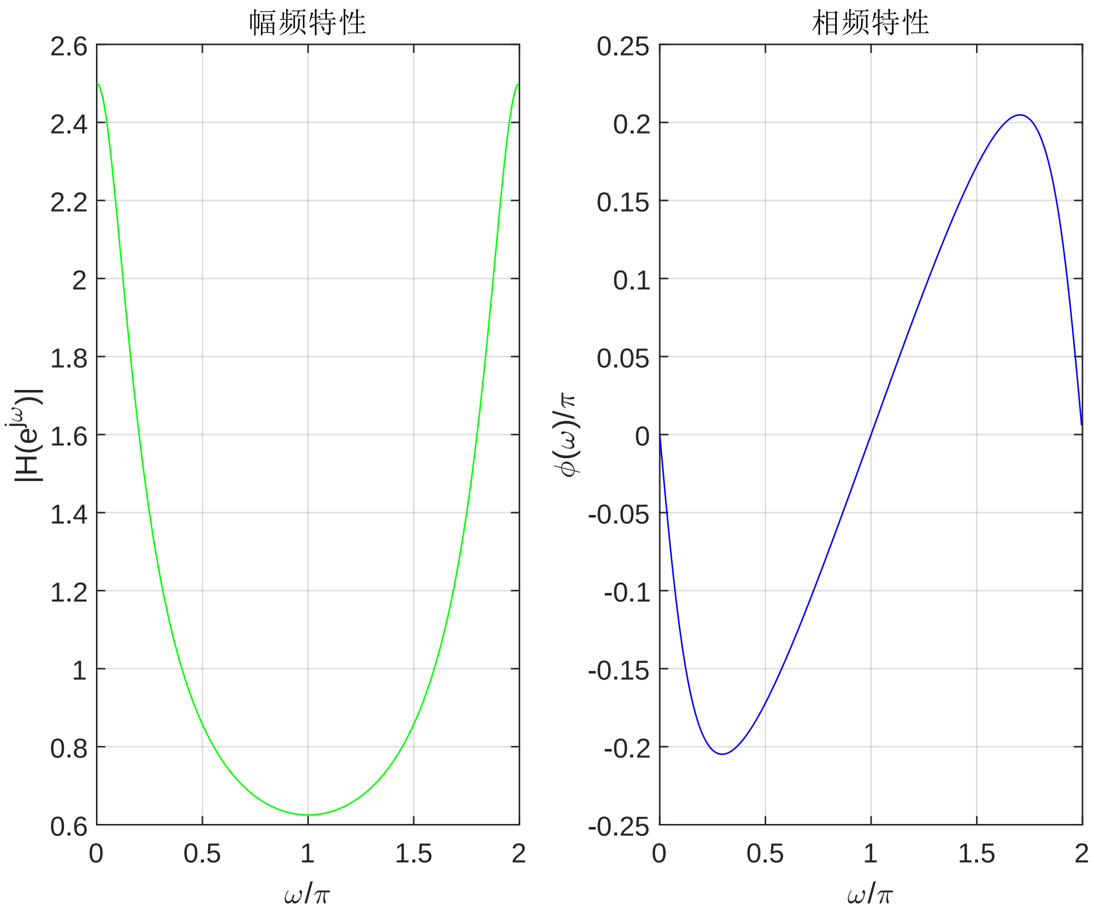

图 8: 幅频特性和相频特性

```
grid on;
title('单位脉冲响应');
xlabel('n');ylabel('h(n)');

figure;
zplane(b,a) % 零极点分布
grid on;
title('零极点分布')
xlabel('Re[z]');ylabel('jIm[z]');

figure;
subplot(1,2,1);
plot(omega/pi,H1,'g')
grid on;
ylabel('|H(e^{j\omega})|')
xlabel('\omega/\pi')
title('幅频特性')

subplot(1,2,2);
plot(omega/pi,H2/pi,'b');
```

```
grid on;
ylabel('\phi(\omega)/\pi')
xlabel('\omega/\pi')
title('相频特性');
```

3.2.2 Python 版本

单位脉冲响应如图 10 所示，零极点分布图如图 11 所示，幅频特性和相频特性曲线如图 9 所示。

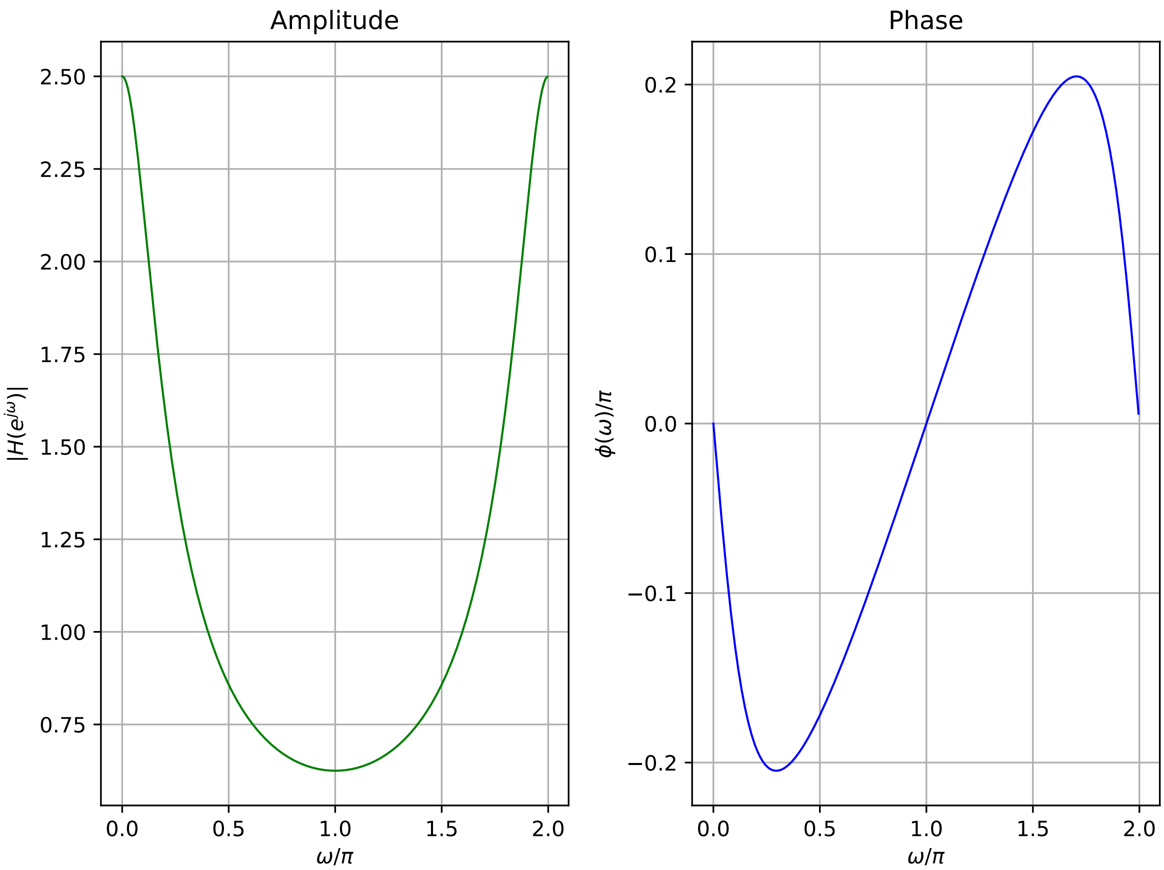

图 9: 幅频特性和相频特性 (Python)

```
# 幅频特性和相频特性
a = np.array([1,-0.6])
b = np.array([1, 0])

omega, H = signal.freqz(b, a, whole=True)

plt.figure(figsize=(8, 6), dpi=80)

plt.subplot(121)
plt.plot(omega/np.pi, np.abs(H), 'g', linewidth=1.0)
plt.grid()
```

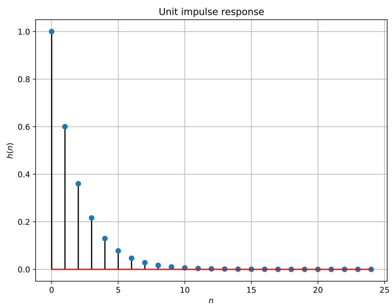

图 10: 单位脉冲响应 (Python)

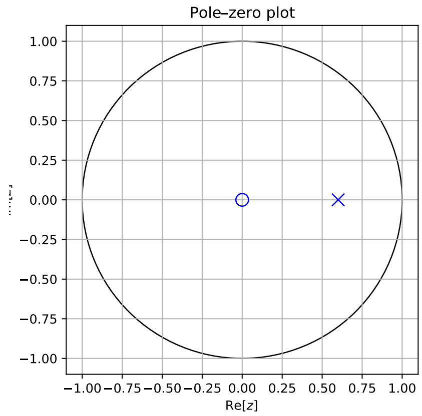

图 11: 零极点分布图 (Python)

```
plt.xlabel('$\omega/\pi$')
plt.ylabel('$|H(e^{j\omega})|$')
plt.title('Amplitude')

plt.subplot(122)
plt.plot(omega/np.pi, np.angle(H)/np.pi, 'b', linewidth=1.0)
plt.grid()
plt.xlabel('$\omega/\pi$')
plt.ylabel('$\phi(\omega)/\pi$')
plt.title('Phase')

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

```
# 单位脉冲响应
n, h = signal.dimpulse((b,a,1),n=25)
plt.figure(figsize=(8, 6), dpi=80)
plt.stem(n, np.squeeze(h), linefmt='black',  basefmt='r-', markerfmt="C0o")
plt.grid()
plt.xlabel('$n$')
plt.ylabel('$h(n)$')
plt.title('Unit impulse response')
plt.show()
```

```
# 零极点分布图
from matplotlib.patches import Circle

z, p, k = signal.tf2zpk(b, a) # 求极点零点

fig, ax = plt.subplots(figsize=(5,5))
circle = Circle(xy=(0.0, 0.0), radius=1, fill=False, color='black')
ax.add_patch(circle)

```

```python
plt.plot(p.real, p.imag, 'bx', markersize=10)
plt.plot(z.real, z.imag, 'o', markersize=10, color='none', markeredgecolor=
    'b')
plt.grid()

r = 1.1 * np.amax(np.concatenate((abs(z), abs(p), [1]))) # z, p 模值和 1 的
    最大值乘以 1.1
plt.xlabel('$\mathrm{Re}[z]$')
plt.ylabel('$\mathrm{Im}[z]$')
plt.title('Pole-zero plot')
plt.axis([-r, r, -r, r])
plt.show()
```

4 总结

4.1 杨文韬

主要负责 LaTeX 排版和所有代码的 Python 版本重写。

- 问题 1：在 Python 中如何求 DTFT？

  思路：一种方法是直接模拟，生成序列根据公式直接计算；另一种方法是其形式是有理多项式时可以通过库函数 `scipy.signal.freqz` 求解。

- 问题 2：如何绘制零极点图？

  思路：在 MATLAB 中可以通过 `zplane()` 函数直接获得。在 Python 中可以通过库函数 `scipy.signal.tf2zpk` 获得零点和极点，调用 `matplotlib.patches` 库里的函数生成圆再绘制零极点。

4.2 刘浩

主要负责 3.1.1 部分 MATLAB 代码编写。

- 问题 1：matlab 如何将多个效果图放到一起显示？

  解决方法：经查询资料后学会用 `subplot` 函数确定图像放置位置

- 问题 2：如何使用 matlab 得到序列的幅频和相频特性？

  解决方法：使用 `abs` 函数获得幅度的绝对值，使用 `angle` 函数获得序列的相位特性

- 问题 3：获得的图像显示不全怎么办？

  解决方法：使用 `axis` 函数调整横纵坐标轴可以获得更好的图像显示效果

4.3 周泽熙

主要负责 3.2.1 部分 MATLAB 代码编写。

- 问题：如何通过差分方程求解频率响应？

  在查阅相关库函数后，发现信号处理包中自带 `freqz()` 函数可用于求解。在使用时仅需传入差分方程两侧系数，便能得到频率响应。再通过 `abs()` 和 `angle()` 求模和角度，便可得到幅频特性和相频特性。