**实验题目**

**实验一**

实验1：系统的频域和z域分析

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，计算系统的系统函数、零极点分布；改变系统的零极点分布，观察系统频率响应的变化。

**实验二**

实验2：信号的频谱分析

设计计算机程序，产生序列并计算序列的FFT和IFFT，绘制其幅频特性和相频特性曲线；模拟产生离散系统的输入序列和单位脉冲响应，利用FFT和IFFT算法计算系统的输出响应，分析FFT的计算长度对系统输出响应的影响；模拟产生连续时间信号，选取适当的采样频率对其采样，并用FFT算法计算其频谱，分析信号的观测时间长度、FFT的计算长度对信号频谱计算结果的影响。

**实验三**

实验3：IIR数字滤波器设计及结构

设计计算机程序，根据滤波器的主要技术指标设计IIR数字巴特沃斯和切比雪夫低通、高通、带通和带阻滤波器；绘制滤波器的幅频特性和相频特性曲线，验证滤波器的设计结果是否达到设计指标要求；画出数字滤波器的直接型、级联型、并联型结构信号流图。

课外学习：课外利用编程工具，设计和分析IIR数字滤波器；设计实验内容和实验步骤，编写上机实验程序，撰写实验报告。

**实验四**

实验4：FIR数字滤波器设计及结构

设计计算机程序，根据滤波器的主要技术指标设计线性相位FIR数字低通、高通、带通和带阻滤波器；绘制滤波器的幅频特性和相频特性曲线，验证滤波器的设计结果是否达到设计指标要求；画出线性相位FIR数字滤波器的结构信号流图。

数字信号处理

实验报告

|  |  |
| --- | --- |
| 班级： | 2102013 |
| 姓名： | 鹿竞泽 |
| 学号： | 21009101253 |
| 联系方式： | Qq:1098636012 |

西安电子科技大学

电子工程学院

目录

[摘要 4](#_Toc148806958)

[绪论 4](#_Toc148806959)

[1. 发展历史 4](#_Toc148806960)

[2. 现状 5](#_Toc148806961)

[3. 研究意义 5](#_Toc148806962)

[正文 5](#_Toc148806963)

[实验目的 5](#_Toc148806964)

[实验原理与方法 6](#_Toc148806965)

[实验原理 6](#_Toc148806966)

[实验方法 6](#_Toc148806967)

[实验内容及步骤 7](#_Toc148806968)

[实验内容 7](#_Toc148806969)

[实验步骤 8](#_Toc148806970)

[实验结果分析及结论总结 8](#_Toc148806971)

[第一部分 8](#_Toc148806972)

[第二部分 9](#_Toc148806973)

[第三部分 10](#_Toc148806974)

[心得及展望 11](#_Toc148806975)

[心得 11](#_Toc148806976)

[展望 12](#_Toc148806977)

[附件 12](#_Toc148806978)

# **摘要**

# **绪论**

数字信号处理（DSP）是一门涉及数字信号的采集、处理和分析的领域，具有广泛的应用，包括通信、音频处理、图像处理、生物医学工程、雷达、控制系统等。以下是数字信号处理的发展历史、现状和研究意义的概述：

## 1. 发展历史

20世纪初，DSP的基础开始奠定，主要用于通信领域。

20世纪60年代，数字计算机的普及使DSP技术更加实际可行。

20世纪70年代和80年代，DSP在音频、图像处理和通信领域取得显著进展。

20世纪90年代，DSP应用领域不断扩展，包括生物医学、雷达、声纳、控制系统等。

21世纪以来，DSP在无线通信、多媒体处理、机器学习等领域得到广泛应用，技术不断演进。

## 2. 现状

- DSP技术已经深入到许多领域，如5G通信、自动驾驶、智能音箱、数字音乐、医学成像等。

- 近年来，DSP在人工智能和机器学习中的应用迅速增长，包括图像识别、语音识别、自然语言处理等。

- 嵌入式DSP处理器的性能不断提高，使得实时信号处理变得更加高效。

- DSP技术也与硬件加速器（如GPU和FPGA）结合，提供更大的计算能力。

## 3. 研究意义

改善信号质量：DSP可以用于降噪、滤波和增强信号，提高信息的质量和可用性。

实时处理：DSP可以在实时应用中处理信号，如通信系统、自动控制和传感器网络。

数据分析：DSP可以用于提取信号中的有用信息，帮助决策制定和模式识别。

通信：DSP在调制、解调、错误检测和纠正等方面对通信领域至关重要。

医疗应用：DSP用于医学成像、心电图分析、生物信号处理等，有助于医疗诊断和疾病监测。

人工智能：DSP与机器学习结合，推动了图像、语音和文本处理等领域的发展。

总的来说，数字信号处理在现代科技和工程领域扮演着至关重要的角色。它不仅帮助改善了现有技术的性能，还推动了新技术的发展，为人类社会的进步和创新提供了强大的工具。

# 正文

## 实验目的

1. **理解数字滤波器**：掌握数字滤波器的基本概念、工作原理以及不同类型的IIR数字滤波器（如巴特沃斯和切比雪夫）的特点。
2. **掌握设计过程**：了解数字滤波器设计的基本步骤，包括选择滤波器类型、确定阶数和截止频率、设计滤波器、分析特性和绘制特性曲线。
3. **应用数字信号处理工具**：熟练使用编程工具（如Python中的SciPy）进行数字滤波器的设计、分析和验证，学习如何编写程序来实现数字滤波器。
4. **验证设计结果**：通过绘制幅频特性和相频特性曲线，验证设计的数字滤波器是否符合预期的设计要求和指标。
5. **掌握滤波器结构**：学习并理解数字滤波器的直接型、级联型和并联型结构信号流图，了解不同结构对信号处理的影响。
6. **撰写实验报告**：学习撰写实验报告的能力，包括实验设计、方法、结果分析和结论，以及对设计过程中遇到的挑战和解决方案的描述。

## 实验原理与方法

### 实验原理

1. **IIR数字滤波器**：IIR（Infinite Impulse Response）滤波器在数字信号处理中使用反馈来实现滤波功能，其具有无限脉冲响应。这些滤波器可以是巴特沃斯、切比雪夫等类型，每种类型有不同的特性。
2. **滤波器设计**：设计数字滤波器的过程包括选择滤波器类型、确定阶数和截止频率、设计滤波器的传递函数以及计算滤波器的系数。
3. **特性分析**：设计完成后，需要对滤波器进行特性分析，包括计算频率响应、绘制幅频特性和相频特性曲线，以验证设计是否符合预期要求。
4. **滤波器结构**：数字滤波器可以采用直接型、级联型和并联型等结构，每种结构对信号处理有不同的影响，需要学会分析和理解这些结构。

### 实验方法

1. **选择滤波器类型和设计要求**：根据实验要求，选择所需的滤波器类型（如巴特沃斯、切比雪夫等）和相应的设计指标（如截止频率、通带波纹、阻带衰减等）。
2. **使用编程工具进行设计**：利用数字信号处理工具（如Python中的SciPy）进行滤波器设计。根据选定的滤波器类型、阶数和截止频率，计算滤波器的系数。
3. **分析滤波器特性**：计算滤波器的频率响应、绘制幅频特性和相频特性曲线，以验证设计结果是否符合设计要求。
4. **绘制滤波器结构信号流图**：画出数字滤波器的直接型、级联型和并联型结构信号流图，展示滤波器的结构，理解不同结构对信号处理的影响。
5. **撰写实验报告**：撰写实验报告，详细描述实验的步骤、方法、设计过程、结果分析、以及针对设计过程中遇到的问题的解决方案和讨论。

## 实验内容及步骤

### 实验内容

1. **选择滤波器类型**：

确定使用的滤波器类型，如巴特沃斯、切比雪夫等，根据设计要求和性能特点进行选择。

1. **确定设计要求**：

确定设计要求，包括所需的滤波器类型（低通、高通、带通或带阻）、截止频率、通带波纹、阻带衰减等参数。

1. **设计滤波器**：

使用所选的数字信号处理工具（如MATLAB、Python中的SciPy等），根据设计要求和所选滤波器类型，设计IIR数字滤波器。这涉及计算滤波器的系数和传递函数。

1. **分析滤波器特性**：

计算滤波器的频率响应，绘制幅频特性和相频特性曲线。验证设计的数字滤波器是否符合设计要求，包括滤波器的通频带、截止频率、通带波纹和阻带衰减等。

1. **绘制滤波器结构信号流图**：

绘制数字滤波器的直接型、级联型和并联型结构信号流图，以展示滤波器的不同结构对信号处理的影响。

1. **上机实验**：

编写程序实现设计的滤波器，并进行模拟或实际数据的处理，观察滤波器的效果。

1. **撰写实验报告**：

撰写详细的实验报告，包括实验目的、所选滤波器类型和设计要求、设计过程、所用工具和方法、实验结果的分析和讨论，以及对设计过程中遇到的问题和解决方案的描述。

### 实验步骤

**准备阶段**：

1. **明确实验目的和要求**：

确定所需设计的滤波器类型（比如巴特沃斯或切比雪夫）、滤波器的种类（低通、高通、带通或带阻）、以及设计要求（截止频率、通带波纹、阻带衰减等）。

1. **选择工具和环境**：

确定使用的编程工具（Python中的SciPy库）进行设计和分析。

**实验步骤**：

1. **设计滤波器**：

根据选定的滤波器类型和设计要求，使用所选的工具进行设计，计算滤波器的系数和传递函数。

1. **分析滤波器特性**：

计算滤波器的频率响应，绘制幅频特性和相频特性曲线。验证设计的数字滤波器是否符合设计要求。

1. **绘制滤波器结构信号流图**：

绘制数字滤波器的直接型、级联型和并联型结构信号流图，以展示滤波器的不同结构对信号处理的影响。

1. **实验验证**：

编写程序实现设计的滤波器，利用模拟信号或实际数据进行滤波处理，观察滤波器的效果，与设计的预期进行比较和分析。

**总结阶段**：

1. **实验报告撰写**：

撰写详细的实验报告，包括实验目的、所选滤波器类型和设计要求、设计过程、所用工具和方法、实验结果的分析和讨论，以及对设计过程中遇到的问题和解决方案的描述。

## 实验结果分析及结论总结

### 第一部分

|  |  |
| --- | --- |
| Butterworth |  |
|  |  |

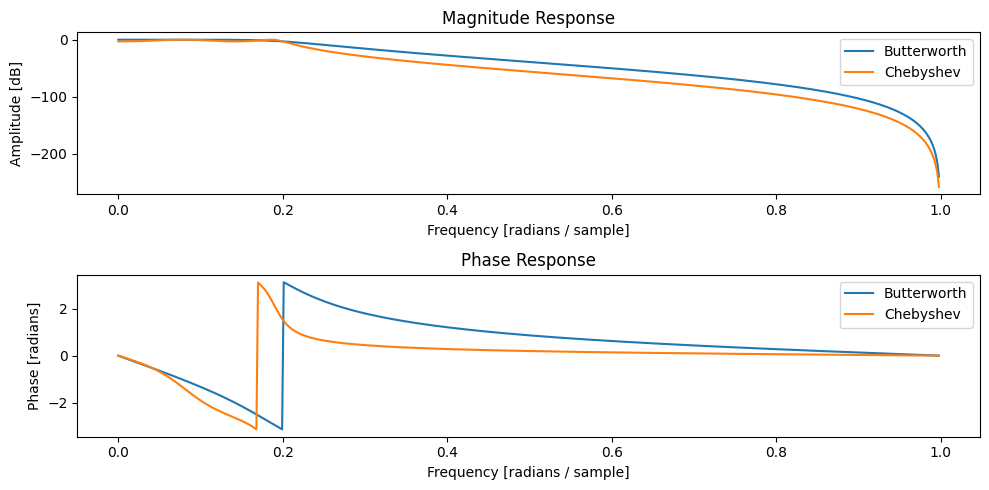
|  |  |
| --- | --- |
| Butterworth |  |
|  |  |

可以看到巴特沃斯滤波器确实很平滑。

### 第二部分

|  |
| --- |
| 切比雪夫 |
|  |

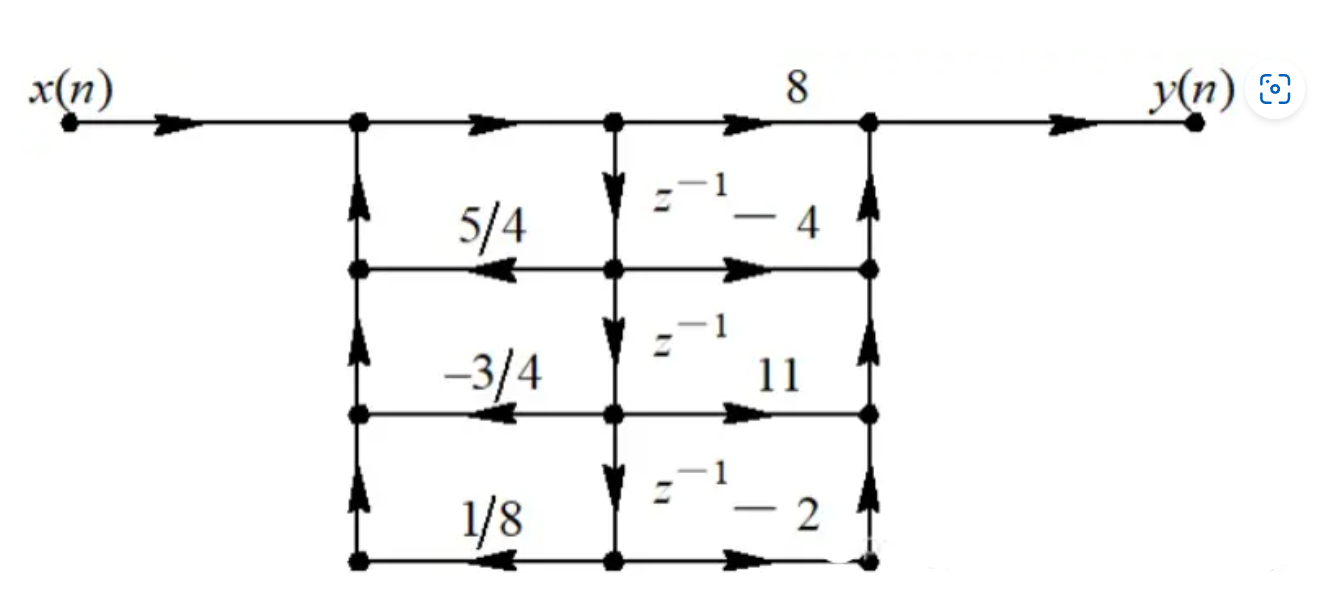
可以看到切比雪夫滤波器确实下降很快但是有波纹。



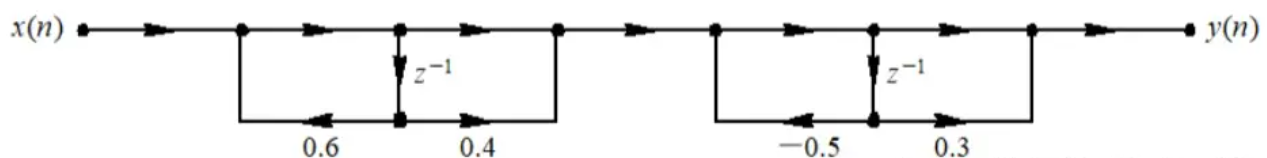
对比看出巴特沃斯下降慢但是没有起伏

### 第三部分

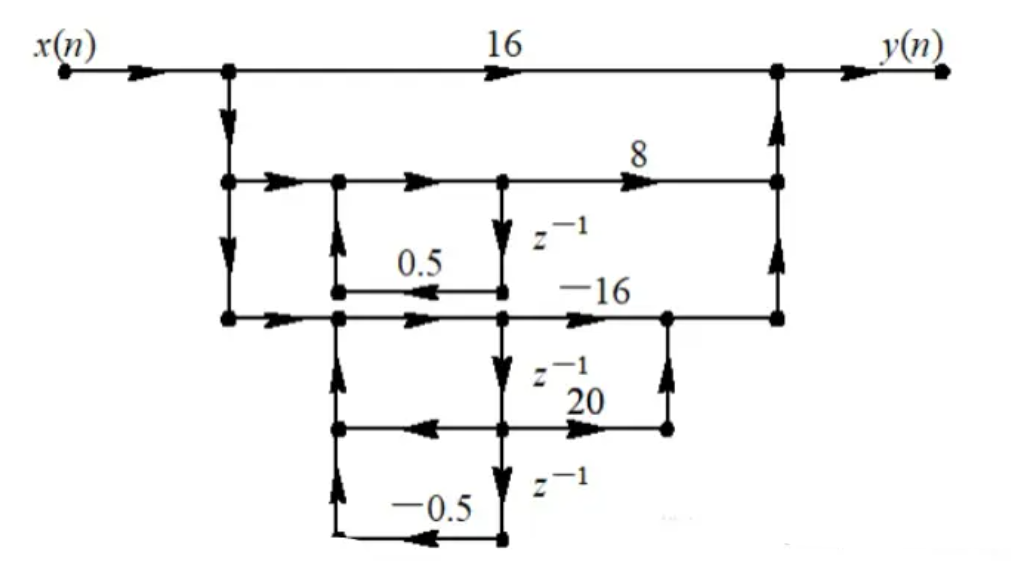
直接型



级联型



并联型



## 心得及展望

### 心得

**理论知识的巩固**：

实验使我更深入地理解了数字滤波器的设计原理和工作方式，对IIR滤波器的无限脉冲响应有了更清晰的认识。

**工程实践能力**：

实验锻炼了我的工程实践能力，通过设计、分析和验证滤波器的过程，掌握了数字信号处理工具的使用和数字滤波器设计的方法。

**问题解决与调试能力**：

在实验中可能会遇到各种问题，如设计偏差、频率特性不符合预期等。逐步解决这些问题，提高了我的问题解决和调试能力。

### 展望

**进一步学习**：

从本次实验中发现了一些设计和分析的局限性，因此希望能进一步学习更高级的滤波器设计方法，了解更多复杂滤波器的特性和应用。

**实际应用与研究**：

将所学到的知识应用到实际工程项目中，或者深入研究数字信号处理领域的相关课题，探索更多滤波器在实际场景中的应用。

**加强编程技能**：

通过本次实验，发现编程能力对于数字信号处理的重要性。希望提升编程技能，尤其是与数字信号处理相关的编程能力。

## 附件

|  |
| --- |
| 代码1幅频 |
| import numpy as np  from scipy import signal  import matplotlib.pyplot as plt  # 设计巴特沃斯低通滤波器  *def* design\_butterworth\_lowpass(*cutoff\_freq*, *order*):      b, a = signal.butter(*order*, *cutoff\_freq*, *btype*='low')      return b, a  # 设计巴特沃斯高通滤波器  *def* design\_butterworth\_highpass(*cutoff\_freq*, *order*):      b, a = signal.butter(*order*, *cutoff\_freq*, *btype*='high')      return b, a  # 设计巴特沃斯带通滤波器  *def* design\_butterworth\_bandpass(*low\_cutoff*, *high\_cutoff*, *order*):      b, a = signal.butter(*order*, [*low\_cutoff*, *high\_cutoff*], *btype*='band')      return b, a  # 设计巴特沃斯带阻滤波器  *def* design\_butterworth\_bandstop(*low\_cutoff*, *high\_cutoff*, *order*):      b, a = signal.butter(*order*, [*low\_cutoff*, *high\_cutoff*], *btype*='bandstop')      return b, a  # 示例：设计一个4阶巴特沃斯低通滤波器，截止频率为0.2（范围 0 到 1）  order = 2  cutoff\_frequency = 0.2  b, a = design\_butterworth\_lowpass(cutoff\_frequency, order)  # 绘制频率响应  w, h = signal.freqz(b, a)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), *label*='Lowpass')  # 设计一个4阶巴特沃斯高通滤波器，截止频率为0.2（范围 0 到 1）  b, a = design\_butterworth\_highpass(cutoff\_frequency, order)  w, h = signal.freqz(b, a)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), *label*='Highpass')  # 设计一个4阶巴特沃斯带通滤波器，截止频率为0.3到0.5（范围 0 到 1）  low\_cutoff = 0.3  high\_cutoff = 0.5  b, a = design\_butterworth\_bandpass(low\_cutoff, high\_cutoff, order)  w, h = signal.freqz(b, a)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), *label*='Bandpass')  # 设计一个4阶巴特沃斯带阻滤波器，截止频率为0.3到0.5（范围 0 到 1）  b, a = design\_butterworth\_bandstop(low\_cutoff, high\_cutoff, order)  w, h = signal.freqz(b, a)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), *label*='Bandstop')  plt.title('Butterworth Filter Frequency Response')  plt.xlabel('Frequency [radians / sample]')  plt.ylabel('Amplitude [dB]')  plt.legend()  plt.show() |

|  |
| --- |
| 代码2 相频 |
| # 示例：设计一个4阶巴特沃斯低通滤波器，截止频率为0.2（范围 0 到 1）  order = 2  cutoff\_frequency = 0.2  b, a = design\_butterworth\_lowpass(cutoff\_frequency, order)  # 打印滤波器系数  # 绘制幅频响应  w, h = signal.freqz(b, a)  # 绘制相频响应  plt.plot(w/np.pi, np.angle(h), *label*='Lowpass')  plt.title('Butterworth Filter Phase Response')  plt.xlabel('Frequency [radians / sample]')  plt.ylabel('Phase [radians]')  # 设计一个4阶巴特沃斯高通滤波器，截止频率为0.2（范围 0 到 1）  b, a = design\_butterworth\_highpass(cutoff\_frequency, order)  w, h = signal.freqz(b, a)    plt.plot(w/np.pi, np.angle(h), *label*='Highpass')  # 设计一个4阶巴特沃斯带通滤波器，截止频率为0.3到0.5（范围 0 到 1）  low\_cutoff = 0.3  high\_cutoff = 0.5  b, a = design\_butterworth\_bandpass(low\_cutoff, high\_cutoff, order)  w, h = signal.freqz(b, a)  plt.plot(w/np.pi, np.angle(h), *label*='Bandpass')  # 设计一个4阶巴特沃斯带阻滤波器，截止频率为0.3到0.5（范围 0 到 1）  b, a = design\_butterworth\_bandstop(low\_cutoff, high\_cutoff, order)  w, h = signal.freqz(b, a)  plt.plot(w/np.pi, np.angle(h), *label*='Bandstop')  plt.tight\_layout()  plt.legend()  plt.show() |

|  |
| --- |
| 代码3 切比雪夫幅频和相频 |
| import numpy as np  from scipy import signal  import matplotlib.pyplot as plt  # 设计切比雪夫低通滤波器  def design\_chebyshev\_lowpass(cutoff\_freq, order, rp):  b, a = signal.cheby1(order, rp, cutoff\_freq, btype='low')  return b, a  # 设计切比雪夫高通滤波器  def design\_chebyshev\_highpass(cutoff\_freq, order, rp):  b, a = signal.cheby1(order, rp, cutoff\_freq, btype='high')  return b, a  # 设计切比雪夫带通滤波器  def design\_chebyshev\_bandpass(low\_cutoff, high\_cutoff, order, rp):  b, a = signal.cheby1(order, rp, [low\_cutoff, high\_cutoff], btype='band')  return b, a  # 设计切比雪夫带阻滤波器  def design\_chebyshev\_bandstop(low\_cutoff, high\_cutoff, order, rp):  b, a = signal.cheby1(order, rp, [low\_cutoff, high\_cutoff], btype='bandstop')  return b, a  # 示例：设计一个4阶切比雪夫低通滤波器，截止频率为0.2，波纹参数rp为3  order = 2  cutoff\_frequency = 0.2  rp = 3  b, a = design\_chebyshev\_lowpass(cutoff\_frequency, order, rp)  # 打印滤波器系数  print("B系数:", b)  print("A系数:", a)  # 绘制幅频响应  plt.figure(figsize=(10, 5))  w, h = signal.freqz(b, a)  plt.subplot(121)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), label='Lowpass')  plt.title('Chebyshev Filter Amplitude Response')  plt.xlabel('Frequency [radians / sample]')  plt.ylabel('Amplitude [dB]')  plt.legend()  # 绘制相频响应  plt.subplot(122)  plt.plot(w/np.pi, np.angle(h), label='Lowpass')  plt.title('Chebyshev Filter Phase Response')  plt.xlabel('Frequency [radians / sample]')  plt.ylabel('Phase [radians]')  # 设计一个4阶切比雪夫高通滤波器，截止频率为0.2，波纹参数rp为3  b, a = design\_chebyshev\_highpass(cutoff\_frequency, order, rp)  w, h = signal.freqz(b, a)  plt.subplot(121)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), label='Highpass')  plt.subplot(122)  plt.plot(w/np.pi, np.angle(h), label='Highpass')  # 设计一个4阶切比雪夫带通滤波器，截止频率为0.3到0.5，波纹参数rp为3  low\_cutoff = 0.3  high\_cutoff = 0.5  b, a = design\_chebyshev\_bandpass(low\_cutoff, high\_cutoff, order, rp)  w, h = signal.freqz(b, a)  plt.subplot(121)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), label='Bandpass')  plt.subplot(122)  plt.plot(w/np.pi, np.angle(h), label='Bandpass')  # 设计一个4阶切比雪夫带阻滤波器，截止频率为0.3到0.5，波纹参数rp为3  b, a = design\_chebyshev\_bandstop(low\_cutoff, high\_cutoff, order, rp)  w, h = signal.freqz(b, a)  plt.subplot(121)  plt.plot(w/np.pi, 20 \* np.log10(abs(h)), label='Bandstop')  plt.legend()  plt.subplot(122)  plt.plot(w/np.pi, np.angle(h), label='Bandstop')  plt.tight\_layout()  plt.legend()  plt.show() |

|  |
| --- |
| 巴特沃斯对比切比雪夫 |
| import numpy as np  from scipy import signal  import matplotlib.pyplot as plt  # 设计巴特沃斯低通滤波器  *def* design\_butterworth\_lowpass(*cutoff\_freq*, *order*):      b, a = signal.butter(*order*, *cutoff\_freq*, *btype*='low')      return b, a  # 设计切比雪夫低通滤波器  *def* design\_chebyshev\_lowpass(*cutoff\_freq*, *order*, *rp*):      b, a = signal.cheby1(*order*, *rp*, *cutoff\_freq*, *btype*='low')      return b, a  # 设计滤波器  order = 4  cutoff\_frequency = 0.2  # 截止频率  rp = 3  # 只在切比雪夫滤波器中有意义  # 设计巴特沃斯低通滤波器  b\_butter, a\_butter = design\_butterworth\_lowpass(cutoff\_frequency, order)  # 设计切比雪夫低通滤波器  b\_cheby, a\_cheby = design\_chebyshev\_lowpass(cutoff\_frequency, order, rp)  # 绘制频率响应  w, h\_butter = signal.freqz(b\_butter, a\_butter)  w, h\_cheby = signal.freqz(b\_cheby, a\_cheby)  plt.figure(*figsize*=(10, 5))  # 绘制幅频特性曲线  plt.subplot(2, 1, 1)  plt.plot(w/np.pi, 20 \* np.log10(abs(h\_butter)), *label*='Butterworth')  plt.plot(w/np.pi, 20 \* np.log10(abs(h\_cheby)), *label*='Chebyshev')  plt.title('Magnitude Response')  plt.xlabel('Frequency [radians / sample]')  plt.ylabel('Amplitude [dB]')  plt.legend()  # 绘制相频特性曲线  plt.subplot(2, 1, 2)  plt.plot(w/np.pi, np.angle(h\_butter), *label*='Butterworth')  plt.plot(w/np.pi, np.angle(h\_cheby), *label*='Chebyshev')  plt.title('Phase Response')  plt.xlabel('Frequency [radians / sample]')  plt.ylabel('Phase [radians]')  plt.legend()  plt.tight\_layout()  plt.show() |