



北京理工大学

信号与信息处理课程设计实验报告

信号产生、分析与处理软件系统设计

学 院 : 信息与电子学院

专 业 : 电子信息工程(徐特立英才班)

班 级 : [REDACTED]

姓 名 : [REDACTED]

学 号 : [REDACTED]

目录

实验七 脑电 ERP 信号的提取方法.....	1
一、 实验题目	1
1. 基本知识.....	1
2. 实验数据.....	2
3. ERP 计算步骤.....	2
二、 实验原理与方法.....	2
1. 滤波器设计.....	2
1.1 基本原理.....	2
1.2 关键 MATLAB 函数.....	4
2. ERP 信号的提取与频谱分析.....	4
2.1 基本原理.....	4
2.2 关键 MATLAB 函数.....	4
三、 实验结果.....	5
1. 软件系统界面设计.....	5
1.1. 标签页 1：滤波器设计.....	5
1.1.1. 上侧参数控制区.....	5
1.1.2. 下侧绘图展示区.....	5
1.2. 标签页 2：ERP 波形显示和频谱分析	6
1.2.1. 上侧参数控制区.....	6
1.2.2. 下侧绘图展示区.....	6
2. 滤波器设计.....	7
2.1. 设计结果.....	7
2.2. 分析讨论.....	7
3. ERP 信号的提取.....	8
3.1. 提取结果.....	8
3.2. 分析讨论.....	10
四、 实验总结.....	10
五、 程序代码.....	11

实验七 脑电 ERP 信号的提取方法

一、 实验题目

1. 基本知识

事件相关电位（Event related potential, ERP）是指当外加一种特定的刺激，作用于感觉系统或脑的某一部位，在给予或撤销刺激时，或当某种心理因素出现时，在脑区所产生的电位变化。EEG 的波幅可达 $50\sim100\mu\text{V}$ ，一次刺激诱发的 ERP 的波幅一般约 $2\sim10\mu\text{V}$ ，比自发电位（EEG）小得多，淹没在 EEG 中。二者构成小信号与大噪声的关系。但 ERP 具有两个恒定，一是波形恒定，二是潜伏期恒定，即同一诱发刺激多次出现时，每次刺激引起的 ERP 波形和潜伏期都是相同的。利用这两个恒定，可以通过叠加，从 EEG 中将 ERP 提取出来。

由于作为 ERP 背景的 EEG 与刺激间无固定的关系，而 ERP 波形在每次刺激后是相同的，且 ERP 波形与刺激间的时间间隔（潜伏期）是固定的，所以经过将刺激时间对齐的叠加，ERP 与叠加次数成比例地增大，而 EEG 则按随机噪声方式加和。若刺激次数为 n ，则叠加 n 次后 ERP 增大 n 倍，而 EEG 只增大 \sqrt{n} 倍，信噪比提高 \sqrt{n} 倍。叠加后的 ERP 数值除以叠加次数，其平均值即还原为一次刺激的 ERP 数值。因此，ERP 又称平均诱发电位，即叠加后的平均。

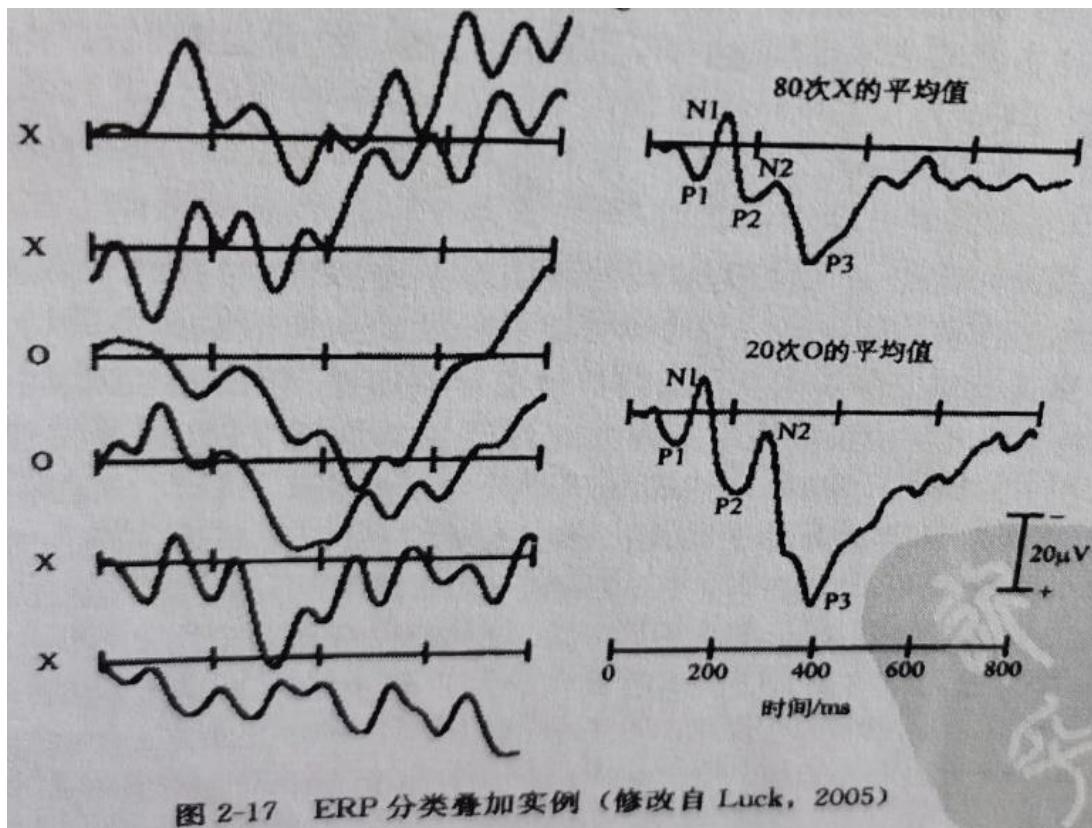


图 1: ERP 分类叠加实例

2. 实验数据

- sig01 文件和 sig02 文件对应不同频率声音刺激下得到的脑电 ERP 信号；数据格式为 mat 文件，在每个文件中，脑电数据存为一个 $2 \times 701 \times 100$ 的矩阵，每个维度分别对应：导联（2 个导联） \times 信号（701 个采样点） \times 试次（100 个试次，即重复刺激 100 次）；第 1 导联为 FCz 电极，第 2 导联为 Cz 电极；数据采样率为 1000Hz；
- 第 1~200 采样点表示刺激前 200ms，用于得到参考基线；第 200 采样点为刺激出现时刻，201~701 采样点为刺激出现后的 700ms 数据。

3. ERP 计算步骤

- (1) 设计高低通滤波器：0.5~30Hz，采样率为 1000 Hz，画出高低通频率响应。对不同试次的脑电信号进行滤波。（butter, freqz, filter 等）
- (2) 去基线：每个试次 701 个点中的前 1-200 个点求均值（mean 函数），然后 1: 701 每个点减去该均值。
- (3) 叠加平均：将所有试次（100 个试次）1: 701 个点逐点对应求均值，即每个试次的第一个点共 100 个值求平均，然后每个试次第二个点求均值，依次类推。（mean (a, 3) 即第 3 维度求均值）
- (4) 分别画出 sig01 文件和 sig02 文件中的第一导联 FCz 和第二导联 Cz 的 ERP 波形。

GUI 界面给出：

1. 滤波器频率响应；
2. FCz 导联 ERP 波形；
3. Cz 导联 ERP 波形。

二、实验原理与方法

事件相关电位（Event Related Potential, ERP）是外加特定刺激时，脑区产生的电位变化，其波幅远小于自发电位（EEG）。ERP 具有波形恒定和潜伏期恒定两大特点，而 EEG 与刺激无固定关联，表现为随机噪声。因此通过叠加平均即可提高信噪比，将 ERP 信号从 EEG 中提取出来。本实验首先对脑电信号进行 0.5~30 Hz 的带通滤波处理，然后通过去基线和叠加平均的操作即可将 ERP 信号提取出来，最后展示提取的 ERP 信号的波形与频谱图。

1. 滤波器设计

1.1 基本原理

实验七的滤波器设计原理与实验五相同，这里再次陈述如下。要实现 0.5~30 Hz 的带通滤波，首先要设计一个对应截止频率的带通滤波器。为了提高滤波器设计的灵

活性，并限制滤波器的阶数不要太高，本实验采取低通滤波器和高通滤波器组合的方式设计带通滤波器。为设计方便起见，并保证在通带范围内滤波器的频率响应尽可能平坦，本实验将采用巴特沃斯滤波器进行 IIR 滤波器设计。

N 阶巴特沃斯低通滤波器的幅度平方响应为：

$$|H_a(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} \quad (1)$$

其中， N 为滤波器阶数， ω_c 是 3dB 截止频率（单位：rad/s）。设定滤波器的截止频率、过渡带宽、通带波纹和阻带衰减等参数，根据公式(1)可计算出模拟巴特沃斯滤波器的最小阶数 N 和 3dB 截止频率 ω_c ，据此可求出滤波器的系统函数 $H_a(s)$ 。

对于 IIR 滤波器，可采用双线性变换法进行设计。根据实验要求可设定低通滤波器的通带截止频率 ω_s 为 30Hz，过渡带宽为 5Hz（即阻带截止频率 ω_p 为 35Hz）；高通滤波器的通带截止频率 ω_s 为 0.5Hz，过渡带宽为 0.1Hz（即阻带截止频率 ω_p 为 0.4Hz），但上述的截止频率不能直接用于巴特沃斯模拟滤波器的设计。首先将实验中设定的模拟截止频率 ω_p 、 ω_s 转换为数字频率 Ω_p 、 Ω_s ，然后按照以下公式进行频率预畸：

$$\omega = \frac{2}{T} \tan\left(\frac{\Omega}{2}\right) \quad (2)$$

由此得到模拟原型滤波器的截止频率 ω'_p 、 ω'_s 。根据预畸后的截止频率 ω'_p 和 ω'_s 、通带波纹和阻带衰减等参数按照上述的方法设计巴特沃斯模拟滤波器 $H_a(s)$ ，再根据双线性变换法的映射关系设计数字滤波器 $H(z)$ ，即为

$$H(z) = H_a\left(\frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}\right) \quad (3)$$

按照双线性变换法可分别设计低通滤波器 $H_1(z)$ 和 $H_2(z)$ ，则组合后的带通滤波器的系统函数为：

$$H(z) = H_1(z)H_2(z) \quad (4)$$

滤波器设计完成后，需要对待处理的脑电信号进行滤波处理，消除低频噪声和高频噪声对 ERP 信号提取的影响。设输入的单试次脑电信号为 $x[n]$ ，滤波处理后的信号为输入信号与滤波器冲激响应的线性卷积：

$$y[n] = x[n] * h[n] \quad (5)$$

因此有

$$y[n] = -\sum_{k=1}^M a_k y[n-k] + \sum_{k=0}^N b_k x[n-k] \quad (6)$$

其中 a_k 和 b_k 分别为滤波器系统函数 $H(z)$ 分母和分子的系数， $y[n]$ 即为滤波处理后的脑

电信号。

1.2 关键 MATLAB 函数

- `buttord`: 根据滤波器指标, 计算巴特沃斯数字滤波器的最优阶数 N 和截止频率 Wn;
- `butter`: 计算 IIR 巴特沃斯数字滤波器的分子、分母系数;
- `freqz`: 计算滤波器频率响应;
- `filter`: 应用设计的滤波器对信号进行滤波处理。

2. ERP 信号的提取与频谱分析

2.1. 基本原理

去基线处理: 刺激前的脑电信号主要是 EEG 信号, 存在一定幅度的基线漂移, 会影响提取 ERP 波形的准确性。以刺激前 200ms (1~200 个采样点) 的脑电信号的平均值作为基线, 用探测到的全时段脑电信号减去该均值, 即可消除基线漂移对 ERP 信号波形提取的影响。

叠加平均处理: 由于每次刺激引起的 ERP 信号具有恒定的波形和潜伏期, 将每个采样点对应的 100 个试次的信号按采样点逐点对齐求平均, 这样做可将每次刺激产生的 ERP 信号叠加起来, 并抵消随机 EEG 噪声的影响, 从而提取出对应导联的 ERP 信号的特征波形。

ERP 信号的频谱特征仍需按照叠加平均的方法进行求解。首先对每试次的 701 个采样点的脑电信号求 FFT 频谱, 再将 100 个试次的频谱叠加求平均值即可得到对应导联的 ERP 信号的频谱。设一个试次的脑电信号为 $x[n]$, 则有:

$$X[k] = \text{FFT}(x[n]) \quad (7)$$

其中 $X[k]$ 为该试次脑电信号的 L 点 FFT, 为方便计算, 通常取 FFT 点数 L 为大于信号采样点数的最小一个 2 的幂次。由于 FFT 仅是 DFT 的快速计算方法, 根据 DFT 和 FT 之间的关系, 当 k 取值在 $0 \sim \frac{L}{2}$ 时对应的模拟频率为 $0 \sim \frac{f_s}{2}$, 则 FFT 序列第 k 个值对应的模拟频率应当为 $\frac{k f_s}{L}$ 。记 $X(j\omega)$ 为对应的连续频谱, 则存在下述对应关系:

$$X\left(j \frac{k f_s}{L}\right) = \frac{X[k]}{f_s} \quad (8)$$

对于单边谱密度, 还需对除 0 和奈奎斯特频率 $\frac{f_s}{2}$ 的频谱幅度值均乘以 2 进行调整。

2.2. 关键 MATLAB 函数

- `fft`: 快速傅里叶变换, 将时域信号转换为频域;
- `mean`: 计算基线均值和叠加平均;

- `size`: 获取数组维度信息;
- `squeeze`: 压缩矩阵单元素维度;

三、实验结果

1. 软件系统界面设计

本系统基于 MATLAB App Designer 开发，采用标签页式模块化布局，通过标签页的切换可分别选择“滤波器设计”和“ERP 波形显示和频谱分析”这两大功能模块。软件系统界面简洁直观、功能分区明确，操作流程清晰，可以完成脑电 ERP 信号的去噪提取基本功能。下面对软件系统界面进行简要介绍：

1.1. 标签页 1：滤波器设计

标签页 1 实现滤波器的设计并绘制滤波器频率响应的功能，布局主要分为上侧参数控制区和下侧绘图展示区两大区域，如图 2 所示：

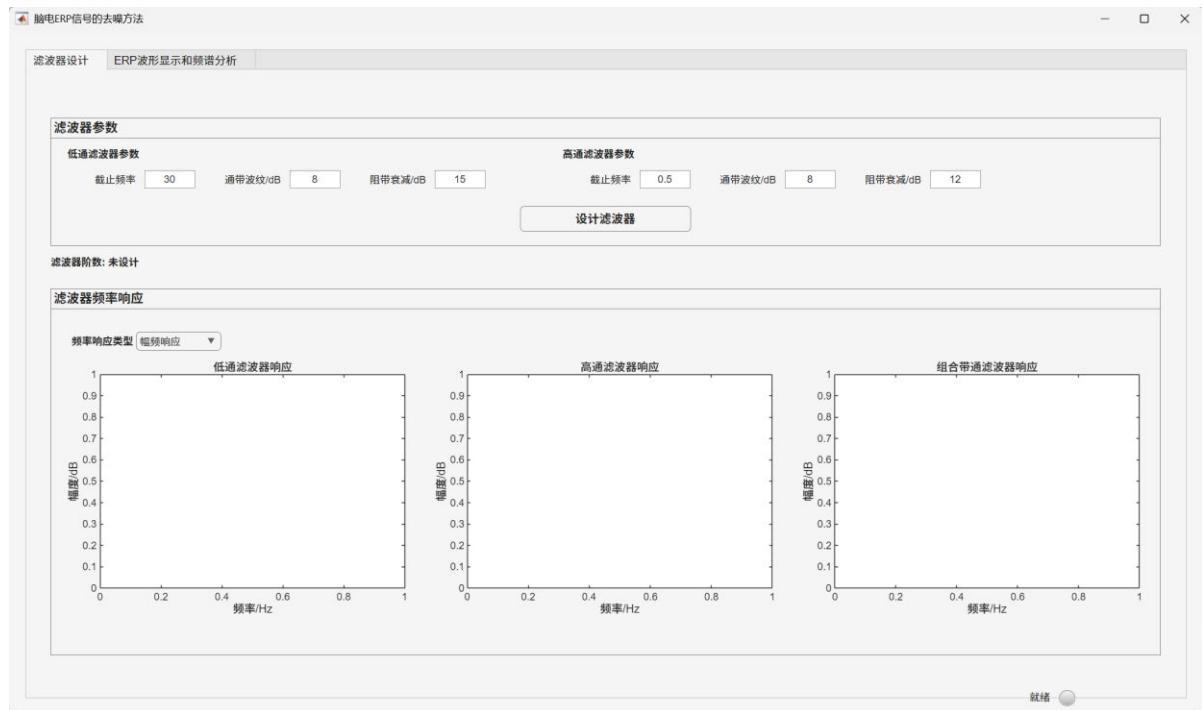


图 2：标签页 1 初始化界面

上图所示界面为直接打开系统，未经任何操作的初始化界面。

1.1.1. 上侧参数控制区

标签页 1 的上侧为滤波器参数面板。滤波器参数面板包含所设计的高通滤波器和低通滤波器的截止频率、通带波纹和阻带衰减等指标，默认输入参数如上图所示，实验中保持默认参数即可。完成滤波器参数的输入后，点击“设计滤波器”按钮即可按照输入的指标进行滤波器的设计，并在下侧绘图展示区绘制滤波器的频率响应。

1.1.2. 下侧绘图展示区

标签页 1 的下侧为滤波器频率响应面板，面板中包含一个“频率响应”下拉框和三个水平排列的坐标区，这三个坐标区分别绘制所设计的低通滤波器、高通滤波器和组合的带通滤波器的频率响应。通过切换“频率响应类型”下拉框，可选择绘制所设计滤波器的幅频响应和相频相应。

1.2. 标签页 2：ERP 波形显示和频谱分析

标签页 2 通过去基线和叠加平均计算出 ERP 时域波形并进行频谱分析，布局主要分为上侧参数控制区和下侧绘图展示区两大区域，如图 3 所示：

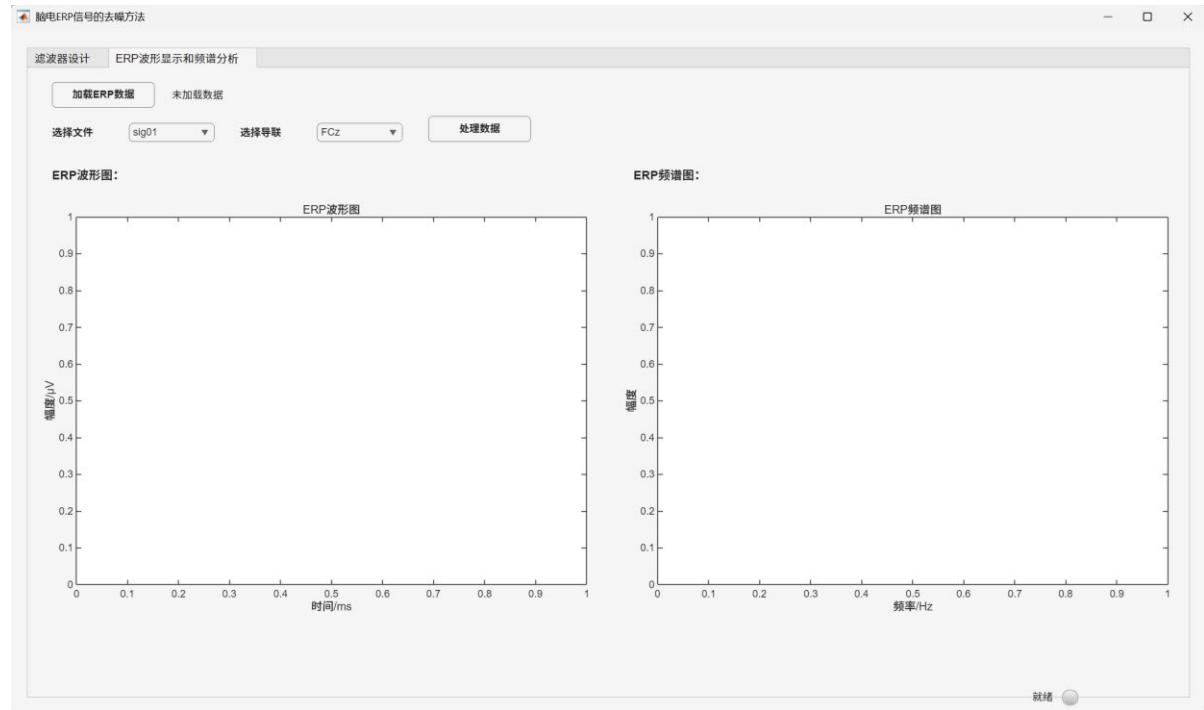


图 3：标签页 2 初始化界面

上图所示界面为直接打开系统，未经任何操作的初始化界面。

1.2.1. 上侧参数控制区

标签页 2 的最上面是“加载 ERP 数据按钮”，点击该按钮即可选择文件夹加载其中的 sig01.mat 和 sig02.mat 脑电 ERP 数据并显示数据维度信息，若加载失败则会弹窗提示错误原因。按钮的下面是“选择文件”和“选择导联”下拉框，用于选择提取脑电 ERP 信号的文件数据及其对应的导联，默认选择 sig01 的 FCz 导联。在完成加载数据和选择文件及其导联的操作后，点击“处理数据”按钮即可进行脑电 ERP 信号的提取并绘制波形和频谱图。

1.2.2. 下侧绘图展示区

标签页 2 的下侧包含两个水平排列的坐标区，左侧坐标区用于绘制去基线和叠加平均处理后的脑电 ERP 信号的时域波形，右侧坐标区用于绘制去基线和叠加平均处理后

脑电 ERP 信号的频谱。

2. 滤波器设计

2.1. 设计结果

按照默认参数进行滤波器设计，滤波器的幅频响应和相频响应分别如图 5、图 6 所示：

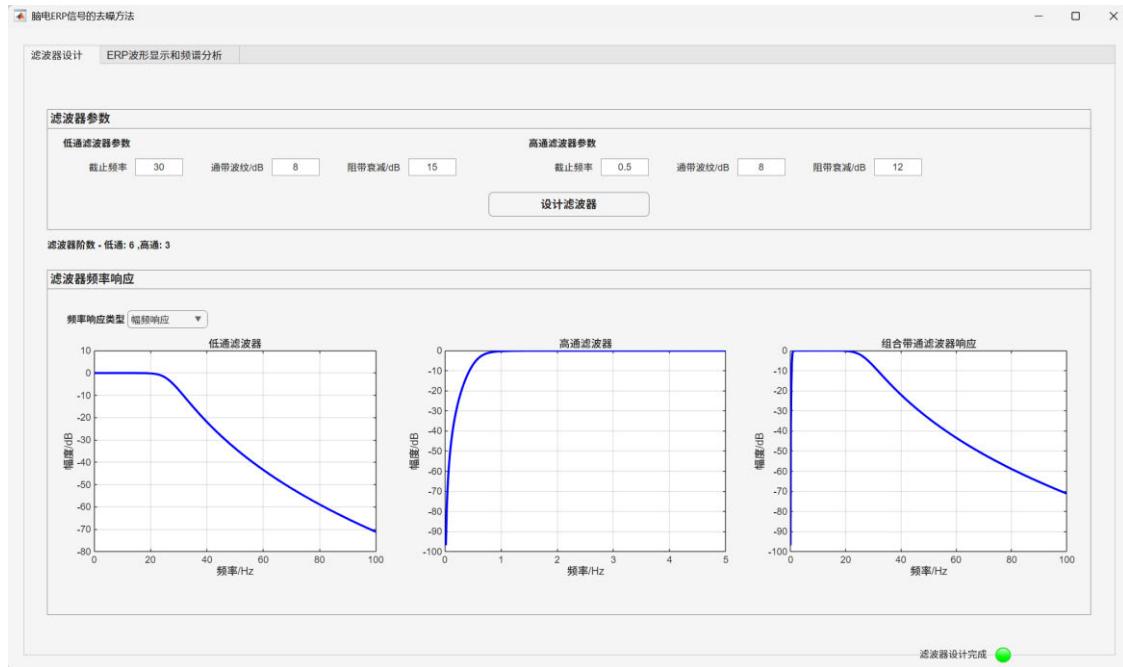


图 4：滤波器幅频响应

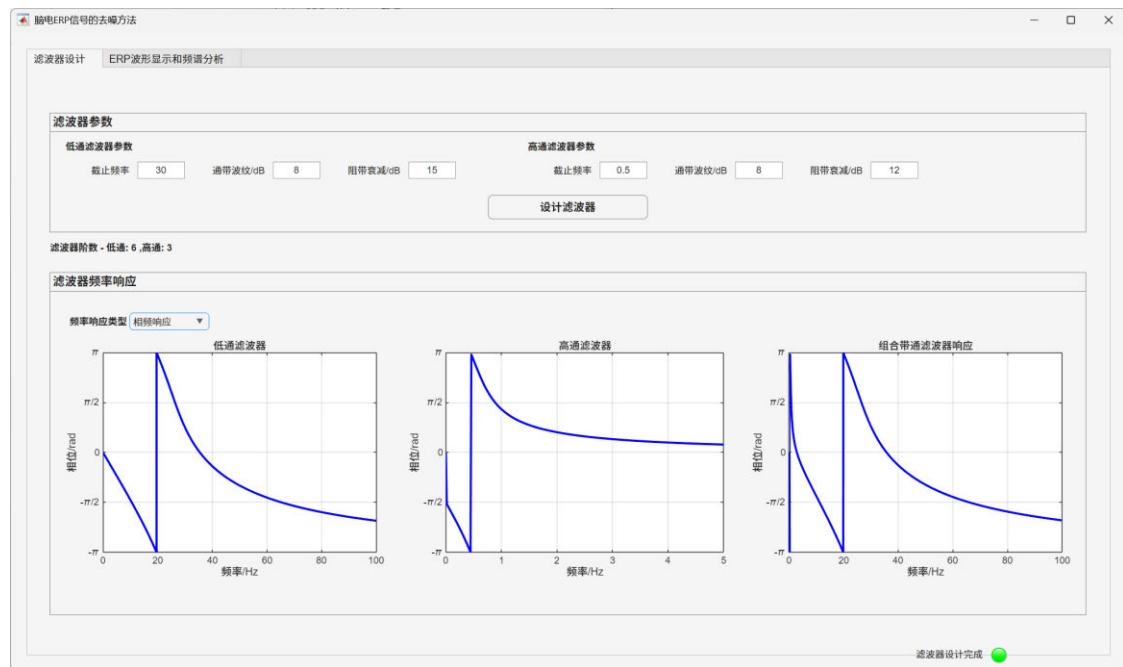


图 5：滤波器相频响应

2.2. 分析讨论

通过观察滤波器的频率响应曲线，可发现低通滤波器和高通滤波器均有较为平坦

的通带，并且截止频率符合设计参数要求。将低通滤波器和高通滤波器组合在一起，即可得到对应的带通滤波器。在本次实验中，由于高通滤波器的截止频率很低，要求过渡带必须充分窄，这在一定程度上会给滤波器的设计带来难度。通过低通和高通滤波器组合构成带通滤波器的设计方法可以很好地满足实验需求，不仅可控制滤波器的阶数不会太大，更可以提高滤波器设计的灵活性，与直接设计带通滤波器相比具有明显的优势。从所设计滤波器的幅频响应曲线可以看出，所设计的巴特沃斯滤波器具有非常平坦的通带，这样使得通带内的频率响应不会出现很大的起伏，并且在阻带范围内仍具有较大的衰减，可以很好地滤除频带外噪声对脑电 ERP 信号提取的影响。

3. ERP 信号的提取

3.1. 提取结果

加载数据文件后，即可进行脑电 ERP 信号的提取。sig01 文件和 sig02 文件中的第一导联 FCz 和第二导联 Cz 的 ERP 波形和频谱分别如图 6、图 7、图 8、图 9 所示：

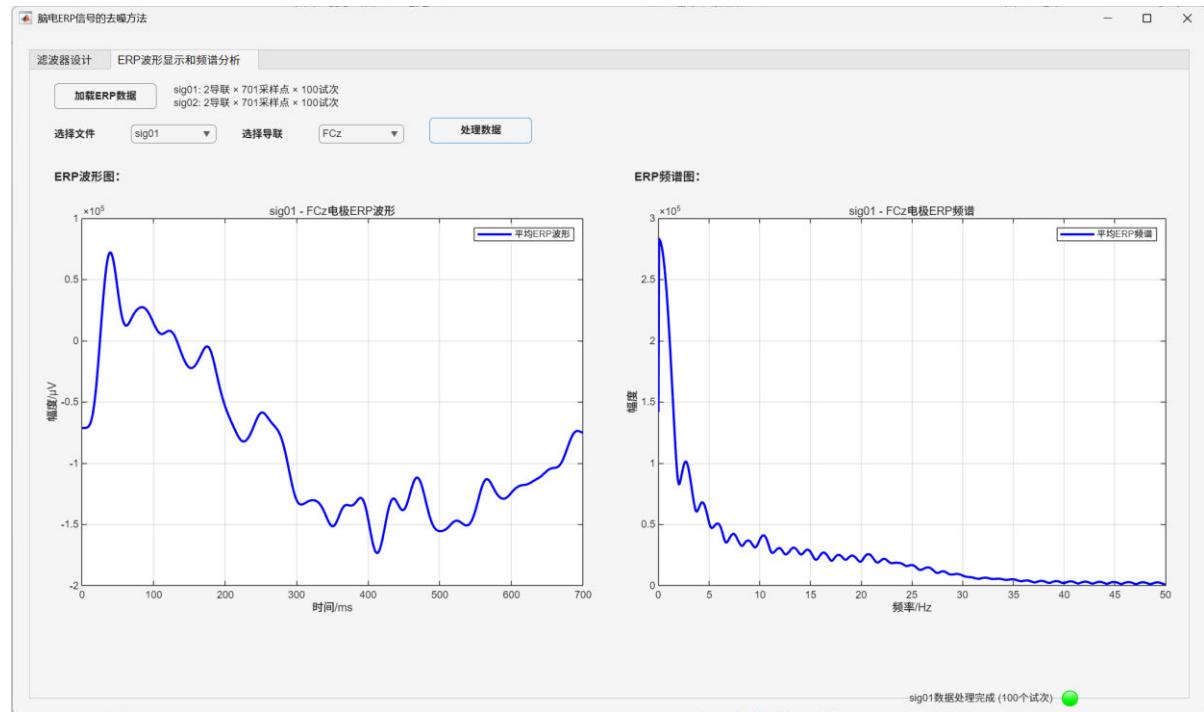


图 6：sig01 文件第一导联 FCz 的 ERP 波形

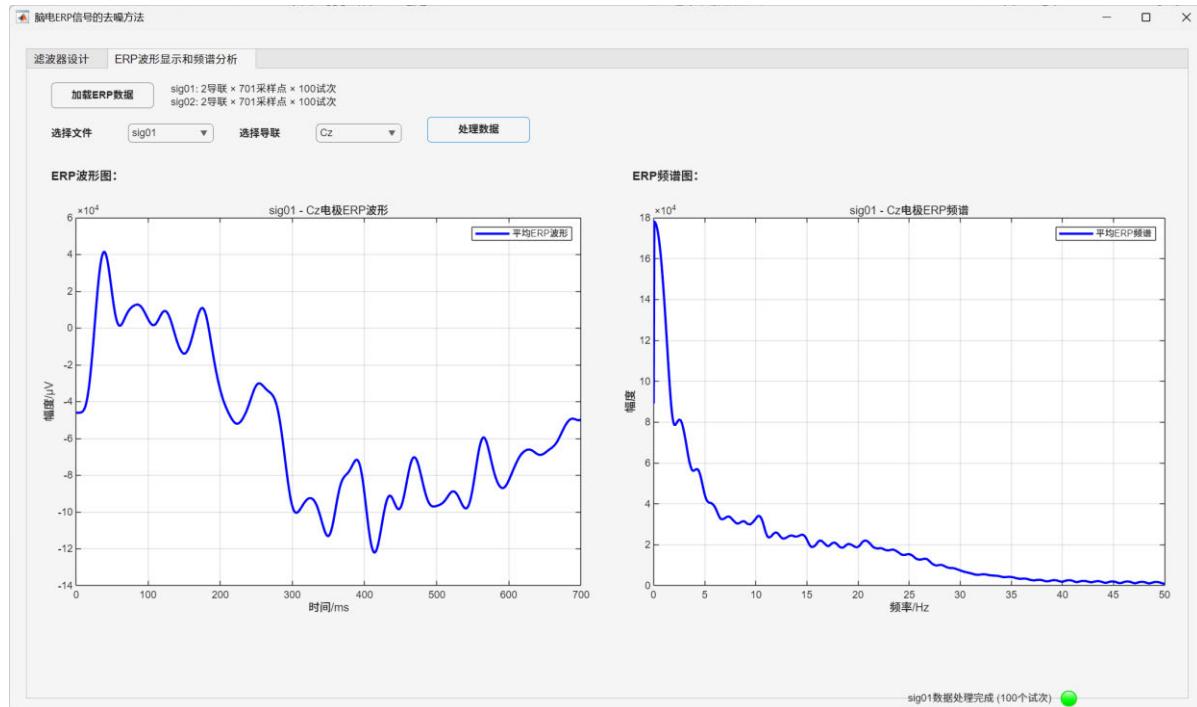


图 7: sig01 文件第二导联 Cz 的 ERP 波形

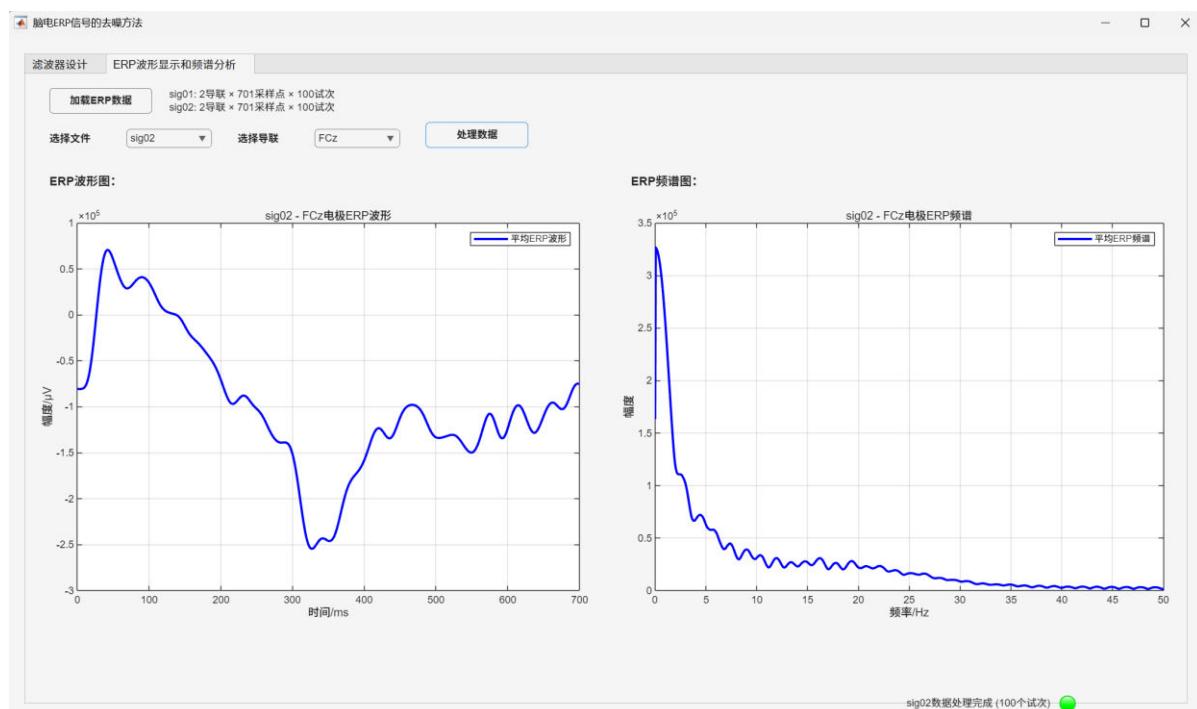


图 8: sig02 文件第一导联 FCz 的 ERP 波形

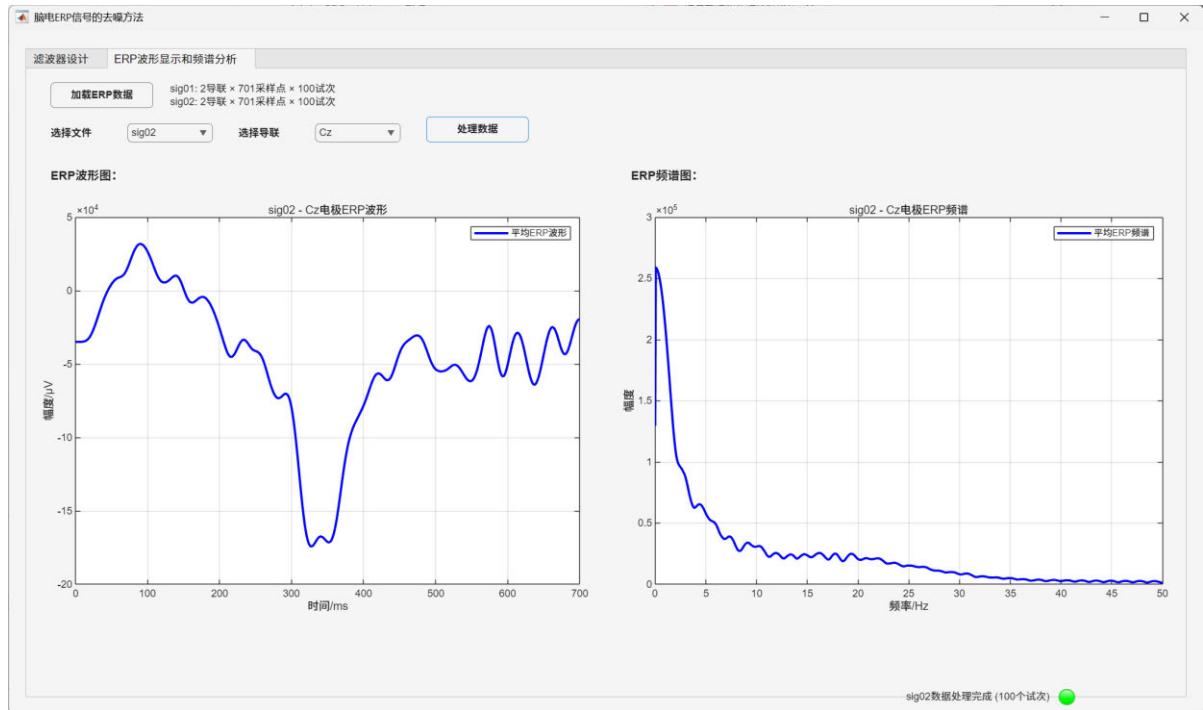


图 9: sig02 文件第二导联 Cz 的 ERP 波形

3.2. 分析讨论

通过观察上述提取的 ERP 信号的波形图可以发现，四张结果图的 ERP 信号时域波形的基线段（1~200）平稳无漂移，并且整个时域波形没有明显的毛刺。频域的能量则集中于 0.5~30Hz，几乎没有高频噪声和低频干扰。因此通过带通滤波、去基线和叠加平均处理可有效滤除频带外噪声，消除 EEG 信号造成的基线漂移和噪声干扰，从而提取出所需要的 ERP 信号。对比观察同一信号文件不同导联的 ERP 信号，可以发现提取出的 ERP 信号波形非常相似。这是因为相同刺激产生的 ERP 信号具有相同的波形，但是不同导联受到的 EEG 信号的干扰不完全相同，所以提取的 ERP 信号仍存在细微差别。然而对于不同刺激产生的 ERP 信号，即使是同一导联，信号波形也存在明显不同，这进一步说明了脑电 ERP 信号取决于外界刺激的类型。

四、实验总结

本次脑电 ERP 信号去噪方法实验，让我系统夯实了数字信号处理的专业知识与 MATLAB APP 设计的实践技能，并对 ERP 信号的提取算法与去噪原理形成了完整且深入的认识。实验过程中，我不仅深入理解了 ERP 信号的产生机制以及 EEG 信号的特性差异，还熟练掌握了 IIR 巴特沃斯滤波器设计、FFT 频谱分析、去基线和叠加平均处理等信号处理技术，成功开发出模块化脑电 ERP 信号提取的 MATLAB App GUI 系统。

在实验进行过程中，我遇到了两大难题。首先是滤波器参数选取不恰当。初期因滤波器的通带波纹和阻带衰减设置不合理，导致滤波后的信号呈现发散的特征。通过

反复优化通带波纹与阻带衰减，并将滤波处理后的信号与 ERP 信号的特征进行比对，最终设计出合适的滤波器。二是对叠加平均的算法理解存在偏差。初期我在对 ERP 信号进行频谱分析时，直接利用叠加平均后的 ERP 信号求解 FFT，这种方法是不恰当的，经过 [] 的提醒我意识到自己的错误所在。因为 ERP 信号具有波形恒定和潜伏期恒定的特点，所以不论是分析 ERP 信号的波形还是频谱都必须严格遵循去基线和叠加平均的步骤。因此求解 ERP 信号频谱的正确方法应当是对去基线后的每个试次求解 FFT，然后将 100 个试次的 FFT 求解结果叠加平均。

此次实验让我体会到理论与实践深度融合的重要价值。之前数字信号处理课程学习的滤波器设计原理、FFT 频谱分析的叠加平均去噪等理论知识，在这次脑电 ERP 信号提取实验中得到了充分运用。在实验中我不仅观察到带通滤波对高低频噪声的抑制效果、去基线和叠加平均对 ERP 信号的提取作用，还通过对 FCz 与 Cz 导联、sig01 与 sig02 信号的结果差异，深化了对 ERP 信号特性的认识和理解，真切感受到该技术在认知神经科学研究中的应用价值。

在此，我要衷心感谢 [] 的悉心指导。[] 在课堂上为我们耐心讲解了滤波器设计原理、脑电 ERP 信号的特征与提取方法，耐心指正我实验中出现的问题，帮助我快速理清实验思路并解决了技术难题。通过本次实验，我不仅巩固了数字信号处理与 MATLAB App 开发的基础能力，更培养了科学严谨的实验态度与问题导向的调试思维。这些收获将为我后续学习信号处理进阶算法、开展相关科研实践筑牢基础，对专业能力提升与未来发展具有重要的推动作用。

五、程序代码

```
1. classdef exp7 < matlab.apps.AppBase
2.
3. % Properties that correspond to app components
4. properties (Access = public)
5.     ERPUIFigure           matlab.ui.Figure
6.     ProgressLamp          matlab.ui.control.Lamp
7.     StatusLabel           matlab.ui.control.Label
8.     TabGroup               matlab.ui.container.TabGroup
9.     FilterDesignTab       matlab.ui.container.Tab
10.    FilterResponsePanel   matlab.ui.container.Panel
11.    FreqzDropDown         matlab.ui.control.DropDown
12.    FreqzLabel             matlab.ui.control.Label
13.    CombinedResponseAxes matlab.ui.control.UIAxes
14.    HighpassResponseAxes matlab.ui.control.UIAxes
15.    LowpassResponseAxes  matlab.ui.control.UIAxes
16.    FilterOrderInfoLabel matlab.ui.control.Label
```

```

17.      FilterParametersPanel      matlab.ui.container.Panel
18.      DesignFilterButton       matlab.ui.control.Button
19.      HighpassStopAttenEditField matlab.ui.control.NumericEditField
20.      HighpassStopAttenLabel   matlab.ui.control.Label
21.      HighpassPassRippleEditField matlab.ui.control.NumericEditField
22.      HighpassPassRippleLabel  matlab.ui.control.Label
23.      HighpassCutoffEditField  matlab.ui.control.NumericEditField
24.      HighpassCutoffLabel     matlab.ui.control.Label
25.      HighpassParametersLabel matlab.ui.control.Label
26.      LowpassStopAttenEditField matlab.ui.control.NumericEditField
27.      LowpassStopAttenLabel   matlab.ui.control.Label
28.      LowpassPassRippleEditField matlab.ui.control.NumericEditField
29.      LowpassPassRippleLabel  matlab.ui.control.Label
30.      LowpassCutoffEditField  matlab.ui.control.NumericEditField
31.      LowpassCutoffLabel     matlab.ui.control.Label
32.      LowpassParametersLabel matlab.ui.control.Label
33.      ERPWaveformTab        matlab.ui.container.Tab
34.      ERPLabel               matlab.ui.control.Label
35.      WaveformLabel          matlab.ui.control.Label
36.      ProcessDataButton      matlab.ui.control.Button
37.      ChannelSelectorDropDown matlab.ui.control.DropDown
38.      ChannelSelectorLabel   matlab.ui.control.Label
39.      FileSelectorDropDown  matlab.ui.control.DropDown
40.      FileSelectorLabel     matlab.ui.control.Label
41.      DataInfoLabel         matlab.ui.control.Label
42.      LoadDataButton        matlab.ui.control.Button
43.      SpectrumAxes          matlab.ui.control.UIAxes
44.      WaveformAxes          matlab.ui.control.UIAxes
45.  end
46.
47.
48.  properties (Access = private)
49.      Property % Description
50.      % 数据属性
51.      Sig01Data              % sig01 文件数据
52.      Sig02Data              % sig02 文件数据
53.      CurrentData            % 当前选择的数据
54.      Fs = 1000               % 采样频率 (Hz)
55.
56.      % 滤波器系数
57.      lpCoeff                % 低通滤波器系数
58.      hpCoeff                % 高通滤波器系数
59.      lpH                     % 低通滤波器频率响应
60.      hpH                     % 高通滤波器频率响应

```

```

61.      combinedH          % 组合滤波器频率响应
62.      combinedF          % 滤波器频率范围
63.
64.      % 处理结果
65.      FilteredData        % 滤波后数据
66.      BaselineCorrectedData % 基线校正后数据
67.      ERPAverage           % ERP 平均波形
68.      ERPAverageFFT        % ERP 平均频谱
69.      Freq                 % ERP 频率范围
70.      TimeAxis             % 时间轴
71.
72.      % 滤波器参数
73.      LowpassCutoff = 30    % 低通截止频率 (Hz)
74.      LowpassPassRipple = 8   % 低通通带波纹 (dB)
75.      LowpassStopAtten = 15   % 低通阻带衰减 (dB)
76.
77.      HighpassCutoff = 0.5    % 高通截止频率 (Hz)
78.      HighpassPassRipple = 8   % 高通通带波纹 (dB)
79.      HighpassStopAtten = 12   % 高通阻带衰减 (dB)
80.
81.      % 电极名称
82.      ElectrodeNames = {'FCz', 'Cz'}
83.
84.      % 当前处理的文件标识
85.      CurrentFile = 'sig01'
86.
87.      % 当前选择的导联
88.      CurrentChannel = 1       % 1=FCz, 2=Cz
89.  end
90.
91. methods (Access = private)
92.
93.     function filterSignal = applyCascadeFilter(app, signal)
94.         % 应用级联滤波器 (先低通后高通)
95.         % 第一步: 低通滤波
96.         %
97.         % % 第二步: 高通滤波
98.         lpFiltered = filter(app.hpCoeff.b, app.hpCoeff.a, signal);
99.         filterSignal = filter(app.lpCoeff.b, app.lpCoeff.a, lpFiltered);
100.    end
101.
102.    function displayResults(app)
103.        % 显示结果: 波形图和频谱图
104.        if isempty(app.ERPAverage)
105.            return;

```

```

106.        end
107.
108.        % 获取当前选择的导联
109.        channelIdx = app.CurrentChannel;
110.        channelName = app.ElectrodeNames{channelIdx};
111.
112.        % 获取当前导联的平均 ERP 波形
113.        erpWaveform = app.ERPAverage(channelIdx, :);
114.
115.        % 显示波形图（左侧）
116.        plot(app.WaveformAxes, app.TimeAxis, erpWaveform, 'b', 'LineWidth', 2);
117.        box(app.WaveformAxes, 'on');
118.
119.        app.WaveformAxes.Title.String = sprintf('%s - %s 电极 ERP 波形
', app.CurrentFile, channelName);
120.        app.WaveformAxes.XLabel.String = '时间/ms';
121.        app.WaveformAxes.YLabel.String = '幅度/μV';
122.        app.WaveformAxes.XGrid = 'on';
123.        app.WaveformAxes.YGrid = 'on';
124.        legend(app.WaveformAxes, '平均 ERP 波形', 'Location', 'northeast');
125.        xlim(app.WaveformAxes, [0, 700]);
126.
127.        % 计算并显示频谱图（右侧）
128.        AverageFFT = app.ERPAverageFFT(channelIdx, :);
129.        plot(app.SpectrumAxes, app.Freq, AverageFFT, 'b', 'LineWidth', 2);
130.        box(app.SpectrumAxes, 'on');
131.
132.        app.SpectrumAxes.Title.String = sprintf('%s - %s 电极 ERP 频谱
', app.CurrentFile, channelName);
133.        app.SpectrumAxes.XLabel.String = '频率/Hz';
134.        app.SpectrumAxes.YLabel.String = '幅度';
135.        app.SpectrumAxes.XGrid = 'on';
136.        app.SpectrumAxes.YGrid = 'on';
137.        legend(app.SpectrumAxes, '平均 ERP 频谱', 'Location', 'northeast');
138.        xlim(app.SpectrumAxes, [0, 50]);
139.    end
140.
141.    function PlotAmpFreqz(app)
142.        % 幅频响应
143.        % 显示低通滤波器响应
144.        plot(app.LowpassResponseAxes, app.combinedF, 20*log10(abs(app.lpH)), 'b',
'LineWidth', 2);
145.        app.LowpassResponseAxes.Title.String = sprintf('低通滤波器');
146.        app.LowpassResponseAxes.XLabel.String = '频率/Hz';

```

```

147.         app.LowpassResponseAxes.YLabel.String = '幅度/dB';
148.         app.LowpassResponseAxes.XGrid = 'on';
149.         app.LowpassResponseAxes.YGrid = 'on';
150.         xlim(app.LowpassResponseAxes, [0 100]);
151.         ylim(app.LowpassResponseAxes, 'auto')
152.         yticks(app.LowpassResponseAxes, 'auto')
153.         yticklabels(app.LowpassResponseAxes, 'auto')

154.

155.         % 显示高通滤波器响应
156.         plot(app.HighpassResponseAxes, app.combinedF, 20*log10(abs(app.hpH)), 'b',
157.             'LineWidth', 2);
158.         app.HighpassResponseAxes.Title.String = sprintf('高通滤波器');
159.         app.HighpassResponseAxes.XLabel.String = '频率/Hz';
160.         app.HighpassResponseAxes.YLabel.String = '幅度/dB';
161.         app.HighpassResponseAxes.XGrid = 'on';
162.         app.HighpassResponseAxes.YGrid = 'on';
163.         xlim(app.HighpassResponseAxes, [0 5]);
164.         ylim(app.HighpassResponseAxes, 'auto')
165.         yticks(app.HighpassResponseAxes, 'auto')
166.         yticklabels(app.HighpassResponseAxes, 'auto')

167.         % 显示组合滤波器响应
168.         plot(app.CombinedResponseAxes, app.combinedF, 20*log10(abs(app.combinedH))
169.             , 'b', 'LineWidth', 2);
170.         app.CombinedResponseAxes.XLabel.String = '频率/Hz';
171.         app.CombinedResponseAxes.YLabel.String = '幅度/dB';
172.         app.CombinedResponseAxes.XGrid = 'on';
173.         app.CombinedResponseAxes.YGrid = 'on';
174.         xlim(app.CombinedResponseAxes, [0 100]);
175.         ylim(app.CombinedResponseAxes, 'auto')
176.         yticks(app.CombinedResponseAxes, 'auto')
177.         yticklabels(app.CombinedResponseAxes, 'auto')

178.     end

179.     function PlotPhasedFreqz(app)
180.         % 相频响应
181.         % 显示低通滤波器响应
182.         plot(app.LowpassResponseAxes, app.combinedF, angle(app.lph), 'b', 'LineWidth',
183.             2);
184.         app.LowpassResponseAxes.Title.String = sprintf('低通滤波器');
185.         app.LowpassResponseAxes.XLabel.String = '频率/Hz';
186.         app.LowpassResponseAxes.YLabel.String = '相位/rad';
187.         app.LowpassResponseAxes.XGrid = 'on';
188.         app.LowpassResponseAxes.YGrid = 'on';
189.         xlim(app.LowpassResponseAxes, [0 100]);

```

```

189.         ylim(app.LowpassResponseAxes, [-pi pi]);
190.         yticks(app.LowpassResponseAxes, -pi:pi/2:pi);
191.         yticklabels(app.LowpassResponseAxes, {'-\pi','-\pi/2','0','\pi/2','\pi'});
192.
193.         % 显示高通滤波器响应
194.         plot(app.HighpassResponseAxes, app.combinedF, angle(app.hpH), 'b', 'LineWidth', 2);
195.         app.HighpassResponseAxes.Title.String = sprintf('高通滤波器');
196.         app.HighpassResponseAxes.XLabel.String = '频率/Hz';
197.         app.HighpassResponseAxes.YLabel.String = '相位/rad';
198.         app.HighpassResponseAxes.XGrid = 'on';
199.         app.HighpassResponseAxes.YGrid = 'on';
200.         xlim(app.HighpassResponseAxes, [0 5]);
201.         ylim(app.HighpassResponseAxes, [-pi pi]);
202.         yticks(app.HighpassResponseAxes, -pi:pi/2:pi);
203.         yticklabels(app.HighpassResponseAxes, {'-\pi','-\pi/2','0','\pi/2','\pi'});
204. ;
205.
206.         % 显示组合滤波器响应
207.         plot(app.CombinedResponseAxes, app.combinedF, angle(app.combinedH), 'b', 'LineWidth', 2);
208.         app.CombinedResponseAxes.XLabel.String = '频率/Hz';
209.         app.CombinedResponseAxes.YLabel.String = '相位/rad';
210.         app.CombinedResponseAxes.XGrid = 'on';
211.         app.CombinedResponseAxes.YGrid = 'on';
212.         xlim(app.CombinedResponseAxes, [0 100]);
213.         ylim(app.CombinedResponseAxes, [-pi pi]);
214.         yticks(app.CombinedResponseAxes, -pi:pi/2:pi);
215.         yticklabels(app.CombinedResponseAxes, {'-\pi','-\pi/2','0','\pi/2','\pi'});
216.     end
217. end
218.
219.
220. % Callbacks that handle component events
221. methods (Access = private)
222.
223.     % Button pushed function: DesignFilterButton
224.     function DesignFilterButtonPushed(app, event)
225.         % 设计巴特沃斯 IIR 滤波器（先低通后高通）- 严格按照示例代码规范
226.         try
227.             appStatusLabel.Text = '正在设计滤波器...';
228.             app.ProgressLamp.Color = 'yellow';

```

```

229.         drawnow;
230.
231.         % 获取滤波器参数
232.         app.LowpassCutoff = app.LowpassCutoffEditField.Value;
233.         app.LowpassPassRipple = app.LowpassPassRippleEditField.Value;
234.         app.LowpassStopAtten = app.LowpassStopAttenEditField.Value;
235.
236.         app.HighpassCutoff = app.HighpassCutoffEditField.Value;
237.         app.HighpassPassRipple = app.HighpassPassRippleEditField.Value;
238.         app.HighpassStopAtten = app.HighpassStopAttenEditField.Value;
239.
240.         % 归一化频率
241.         nyquist = app.Fs / 2;
242.
243.         % 低通滤波器设计参数
244.         lpWp = app.LowpassCutoff / nyquist;          % 通带截止频率
245.         lpWs = (app.LowpassCutoff + 5) / nyquist; % 阻带截止频率(比通带高 5Hz)
246.
247.         % 高通滤波器设计参数
248.         hpWp = app.HighpassCutoff / nyquist;          % 通带截止频率
249.         hpWs = (app.HighpassCutoff - 0.1) / nyquist; % 阻带截止频率
250.         % 设计低通滤波器
251.         [lpN, lpWn] = buttord(lpWp, lpWs, app.LowpassPassRipple, app.LowpassStopAtten);
252.         [lpB, lpA] = butter(lpN, lpWn, 'low');
253.
254.         % 设计高通滤波器
255.         [hpN, hpWn] = buttord(hpWp, hpWs, app.HighpassPassRipple, app.HighpassStopAtten);
256.         [hpB, hpA] = butter(hpN, hpWn, 'high');
257.
258.         % 保存滤波器系数
259.         app.lpCoeff.b = lpB;
260.         app.lpCoeff.a = lpA;
261.         app.lpCoeff.order = lpN;
262.         app.lpCoeff.passRipple = app.LowpassPassRipple;
263.         app.lpCoeff.stopAtten = app.LowpassStopAtten;
264.
265.         app.hpCoeff.b = hpB;
266.         app.hpCoeff.a = hpA;
267.         app.hpCoeff.order = hpN;
268.         app.hpCoeff.passRipple = app.HighpassPassRipple;
269.         app.hpCoeff.stopAtten = app.HighpassStopAtten;
270.

```

```

271.          % 计算频率响应
272.          N = 32768;
273.          [app.lpH, app.combinedF] = freqz(lpB, lpA, N, app.Fs);
274.          [app.hpH, ~] = freqz(hpB, hpA, N, app.Fs);
275.
276.          % 计算组合响应
277.          app.combinedH = app.lpH .* app.hpH;
278.          % 绘制频率响应
279.          switch app.FreqzDropDown.Value
280.              case '幅频响应'
281.                  PlotAmpFreqz(app)
282.              case '相频响应'
283.                  PlotPhasedFreqz(app)
284.              otherwise
285.                  PlotAmpFreqz(app)
286.          end
287.
288.          % 显示滤波器阶数信息
289.          app.FilterOrderInfoLabel.Text = sprintf('滤波器阶数 - 低通: %d ,高通: %d', lpN, hpN);
290.          appStatusLabel.Text = '滤波器设计完成';
291.          app.ProgressLamp.Color = 'green';
292.
293.      catch ME
294.          uialert(app.ERPUIFigure, ME.message, '滤波器设计错误');
295.          appStatusLabel.Text = '滤波器设计失败';
296.          app.ProgressLamp.Color = 'red';
297.      end
298.  end
299.
300.  % Button pushed function: LoadDataButton
301.  function LoadDataButtonPushed(app, event)
302.      % 加载 ERP 数据 - 只选择路径, 自动识别 sig01 和 sig02 文件
303.      try
304.          appStatusLabel.Text = '正在加载 ERP 数据...';
305.          app.ProgressLamp.Color = 'yellow';
306.          drawnow;
307.
308.          % 选择文件夹
309.          folderPath = uigetdir('', '选择包含 sig01 和 sig02 数据的文件夹');
310.          if folderPath == 0
311.              appStatusLabel.Text = '数据加载已取消';
312.              app.ProgressLamp.Color = [0.8 0.8 0.8];
313.          return;
314.      end

```

```

315.
316.         % 自动查找文件
317.         sig01Files = dir(fullfile(folderPath, '*sig01*.mat'));
318.         sig02Files = dir(fullfile(folderPath, '*sig02*.mat'));
319.
320.         if isempty(sig01Files)
321.             % 如果没有找到 sig01 文件, 提示用户选择 sig01 文件
322.             [file1, path1] = uigetfile({'*.mat', 'MATLAB Data Files'}, '请选择
323.             sig01 数据文件', folderPath);
324.             if isequal(file1, 0)
325.                 appStatusLabel.Text = '数据加载已取消';
326.                 return;
327.             end
328.             sig01Path = fullfile(path1, file1);
329.         else
330.             sig01Path = fullfile(folderPath, sig01Files(1).name);
331.         end
332.
333.         % 加载 sig01
334.         data1 = load(sig01Path);
335.         fieldNames1 = fieldnames(data1);
336.         if ismember('sig01', fieldNames1)
337.             app.Sig01Data = data1.sig01;
338.         elseif length(fieldNames1) == 1
339.             app.Sig01Data = data1.(fieldNames1{1});
340.         else
341.             uialert(app.ERPUIFigure, '无法识别 sig01 数据变量', '数据格式错误');
342.             return;
343.         end
344.
345.         % 检查 sig02 文件
346.         if ~isempty(sig02Files)
347.             sig02Path = fullfile(folderPath, sig02Files(1).name);
348.             data2 = load(sig02Path);
349.             fieldNames2 = fieldnames(data2);
350.             if ismember('sig02', fieldNames2)
351.                 app.Sig02Data = data2.sig02;
352.             elseif length(fieldNames2) == 1
353.                 app.Sig02Data = data2.(fieldNames2{1});
354.             else
355.                 uialert(app.ERPUIFigure, '无法识别 sig02 数据变量', '数据格式错误
356.             ');
357.             app.Sig02Data = [];
end

```

```

358.         else
359.             app.Sig02Data = [];
360.         end
361.
362.         % 检查数据维度
363.         [numElectrodes1, numSamples1, numTrials1] = size(app.Sig01Data);
364.         if numElectrodes1 ~= 2 || numSamples1 ~= 701
365.             uialert(app.ERPUIFigure, 'sig01 数据格式应为 2x701x100', '维度错误
366. ');
367.             return;
368.         end
369.
370.         if ~isempty(app.Sig02Data)
371.             [numElectrodes2, numSamples2, numTrials2] = size(app.Sig02Data);
372.             if numElectrodes2 ~= 2 || numSamples2 ~= 701
373.                 uialert(app.ERPUIFigure, 'sig02 数据格式应为 2x701x100', '维度错
374. 误');
375.             app.Sig02Data = [];
376.         end
377.
378.         % 更新数据显示
379.         infoText = sprintf('sig01: %d 导联 x %d 采样点 x %d 试次', ...
380.                         numElectrodes1, numSamples1, numTrials1);
381.
382.         if ~isempty(app.Sig02Data)
383.             infoText = sprintf('%s\nsig02: %d 导联 x %d 采样点 x %d 试次', ...
384.                             infoText, numElectrodes2, numSamples2, numTrials2);
385.
386.         app.DataInfoLabel.Text = infoText;
387.
388.         % 设置当前数据和下拉列表
389.         app.CurrentData = app.Sig01Data;
390.         app.CurrentFile = 'sig01';
391.         appStatusLabel.Text = 'ERP 数据加载完成';
392.         app.ProgressLamp.Color = 'green';
393.
394.         catch ME
395.             uialert(app.ERPUIFigure, ME.message, '数据加载错误');
396.             appStatusLabel.Text = '数据加载失败';
397.             app.ProgressLamp.Color = 'red';
398.         end
399.     end
400.

```

```

401.      % Button pushed function: ProcessDataButton
402.      function ProcessDataButtonPushed(app, event)
403.          % 处理 ERP 数据: 滤波、去基线、叠加平均
404.          try
405.              if isempty(app.CurrentData)
406.                  uialert(app.ERPUIFigure, '请先加载数据', '数据未加载');
407.                  return;
408.              end
409.
410.              if isempty(app.lpCoeff) || isempty(app.hpCoeff)
411.                  uialert(app.ERPUIFigure, '请先设计滤波器', '滤波器未设计');
412.                  return;
413.              end
414.
415.              appStatusLabel.Text = '正在处理 ERP 数据...';
416.              appProgressLamp.Color = 'yellow';
417.              drawnow;
418.
419.              % 获取数据维度
420.              [numElectrodes, numSamples, numTrials] = size(app.CurrentData);
421.              NFFT = 2^(nextpow2(numSamples)+4); % FFT 点数, 下一个 2 的幂
422.
423.              % 初始化处理结果
424.              app.FilteredData = zeros(size(app.CurrentData));
425.              app.BaselineCorrectedData = zeros(size(app.CurrentData));
426.              app.ERPAverage = zeros(numElectrodes, numSamples);
427.              app.ERPAverageFFT = zeros(numElectrodes, NFFT/2+1);
428.              app.Freq = (0:NFFT/2) * app.Fs / NFFT;
429.
430.              % 创建时间轴 (刺激在 200ms 处, 即第 200 个采样点)
431.              app.TimeAxis = (0:numSamples-1) / app.Fs * 1000; % 转换为毫秒
432.
433.              % 处理每个试次
434.              for trial = 1:numTrials
435.                  for electrode = 1:numElectrodes
436.                      % 提取原始信号
437.                      rawSignal = squeeze(app.CurrentData(electrode, :, trial));
438.
439.                      % 步骤 1: 滤波
440.                      filteredSignal = app.applyCascadeFilter(rawSignal);
441.                      app.FilteredData(electrode, :, trial) = filteredSignal;
442.
443.                      % 步骤 2: 去基线 (前 200 个点求平均)
444.                      baselineMean = mean(filteredSignal(1:200));
445.                      baselineCorrected = filteredSignal - baselineMean;

```

```

446.                     app.BaselineCorrectedData(electrode, :, trial) = baselineCorre
cted;
447.
448.             % 对去基线的数据进行频谱分析
449.             fftResult = fft(baselineCorrected, NFFT);
450.             spectrum = abs(fftResult(1:NFFT/2+1)/app.Fs);
451.             % 计算单边频谱
452.             spectrum(2:end-1) = spectrum(2:end-1) * 2;
453.
454.             % 累加用于平均
455.             app.ERPAverage(electrode, :) = app.ERPAverage(electrode, :) +
baselineCorrected;
456.             app.ERPAverageFFT(electrode, :) = app.ERPAverageFFT(electrode,
:) + spectrum;
457.         end
458.     end
459.
460.     % 步骤 3: 叠加平均 (除以试次数)
461.     app.ERPAverage = app.ERPAverage / numTrials;
462.     app.ERPAverageFFT = app.ERPAverageFFT / numTrials;
463.
464.     % 显示结果
465.     app.displayResults();
466.     appStatusLabel.Text = sprintf('%s 数据处理完成 (%d 个试
次)', app.CurrentFile, numTrials);
467.     app.ProgressLamp.Color = 'green';
468.
469.     catch ME
470.         uialert(app.ERPUIFigure, ME.message, '数据处理错误');
471.         appStatusLabel.Text = '数据处理失败';
472.         app.ProgressLamp.Color = 'red';
473.     end
474. end
475.
476. % Value changed function: FileSelectorDropDown
477. function FileSelectorDropDownValueChanged(app, event)
478.     % 切换当前处理的文件
479.     selectedFile = app.FileSelectorDropDown.Value;
480.
481.     if strcmp(selectedFile, 'sig01') && ~isempty(app.Sig01Data)
482.         app.CurrentData = app.Sig01Data;
483.         app.CurrentFile = 'sig01';
484.         appStatusLabel.Text = '已切换到 sig01 数据';
485.
486.     elseif strcmp(selectedFile, 'sig02') && ~isempty(app.Sig02Data)

```

```

487.         app.CurrentData = app.Sig02Data;
488.         app.CurrentFile = 'sig02';
489.         appStatusLabel.Text = '已切换到 sig02 数据';
490.
491.     else
492.         appStatusLabel.Text = '所选文件数据未加载';
493.         appProgressLamp.Color = 'yellow';
494.     end
495. end
496.
497. % Value changed function: ChannelSelectorDropDown
498. function ChannelSelectorDropDownValueChanged(app, event)
499.     % 切换当前选择的导联
500.     selectedChannel = app.ChannelSelectorDropDown.Value;
501.     if strcmp(selectedChannel, 'FCz')
502.         app.CurrentChannel = 1;
503.     elseif strcmp(selectedChannel, 'Cz')
504.         app.CurrentChannel = 2;
505.     end
506. end
507.
508. % Value changed function: FreqzDropDown
509. function FreqzValueChanged(app, event)
510.     % 绘制频率响应
511.     switch app.FreqzDropDown.Value
512.         case '幅频响应'
513.             PlotAmpFreqz(app)
514.         case '相频响应'
515.             PlotPhasedFreqz(app)
516.         otherwise
517.             PlotAmpFreqz(app)
518.     end
519. end
520. end
521.
522. % Component initialization
523. methods (Access = private)
524.
525.     % Create UIFigure and components
526.     function createComponents(app)
527.
528.         % Create ERPUIFigure and hide until all components are created
529.         app.ERPUIFigure = uifigure('Visible', 'off');
530.         app.ERPUIFigure.Position = [100 100 1400 800];

```

```

531.         app.ERPUIFigure.Name = '脑电 ERP 信号的去噪方法';
532.
533.         % Create TabGroup
534.         app.TabGroup = uitabgroup(app.ERPUIFigure);
535.         app.TabGroup.Position = [20 20 1360 760];
536.
537.         % Create FilterDesignTab
538.         app.FilterDesignTab = uitab(app.TabGroup);
539.         app.FilterDesignTab.Title = '滤波器设计';
540.
541.         % Create FilterParametersPanel
542.         app.FilterParametersPanel = uipanel(app.FilterDesignTab);
543.         app.FilterParametersPanel.Title = '滤波器参数';
544.         app.FilterParametersPanel.FontWeight = 'bold';
545.         app.FilterParametersPanel.FontSize = 15;
546.         app.FilterParametersPanel.Position = [30 530 1300 150];
547.
548.         % Create LowpassParametersLabel
549.         app.LowpassParametersLabel = uilabel(app.FilterParametersPanel);
550.         app.LowpassParametersLabel.FontWeight = 'bold';
551.         app.LowpassParametersLabel.Position = [20 97 120 22];
552.         app.LowpassParametersLabel.Text = '低通滤波器参数';
553.
554.         % Create LowpassCutoffLabel
555.         app.LowpassCutoffLabel = uilabel(app.FilterParametersPanel);
556.         app.LowpassCutoffLabel.HorizontalAlignment = 'right';
557.         app.LowpassCutoffLabel.Position = [20 67 80 22];
558.         app.LowpassCutoffLabel.Text = '截止频率';
559.
560.         % Create LowpassCutoffEditField
561.         app.LowpassCutoffEditField = uieditfield(app.FilterParametersPanel, 'numer
      ic');
562.         app.LowpassCutoffEditField.Limits = [10 200];
563.         app.LowpassCutoffEditField.HorizontalAlignment = 'center';
564.         app.LowpassCutoffEditField.Position = [110 67 60 22];
565.         app.LowpassCutoffEditField.Value = 30;
566.
567.         % Create LowpassPassRippleLabel
568.         app.LowpassPassRippleLabel = uilabel(app.FilterParametersPanel);
569.         app.LowpassPassRippleLabel.HorizontalAlignment = 'right';
570.         app.LowpassPassRippleLabel.Position = [190 67 80 22];
571.         app.LowpassPassRippleLabel.Text = '通带波纹/dB';
572.
573.         % Create LowpassPassRippleEditField

```

```

574.         app.LowpassPassRippleEditField = uieditfield(app.FilterParametersPanel, 'n
      umeric');
575.         app.LowpassPassRippleEditField.HorizontalAlignment = 'center';
576.         app.LowpassPassRippleEditField.Position = [280 67 60 22];
577.         app.LowpassPassRippleEditField.Value = 8;
578.
579.         % Create LowpassStopAttenLabel
580.         app.LowpassStopAttenLabel = uilabel(app.FilterParametersPanel);
581.         app.LowpassStopAttenLabel.HorizontalAlignment = 'right';
582.         app.LowpassStopAttenLabel.Position = [360 67 80 22];
583.         app.LowpassStopAttenLabel.Text = '阻带衰减/dB';
584.
585.         % Create LowpassStopAttenEditField
586.         app.LowpassStopAttenEditField = uieditfield(app.FilterParametersPanel, 'nu
      meric');
587.         app.LowpassStopAttenEditField.HorizontalAlignment = 'center';
588.         app.LowpassStopAttenEditField.Position = [450 67 60 22];
589.         app.LowpassStopAttenEditField.Value = 15;
590.
591.         % Create HighpassParametersLabel
592.         app.HighpassParametersLabel = uilabel(app.FilterParametersPanel);
593.         app.HighpassParametersLabel.FontWeight = 'bold';
594.         app.HighpassParametersLabel.Position = [600 97 120 22];
595.         app.HighpassParametersLabel.Text = '高通滤波器参数';
596.
597.         % Create HighpassCutoffLabel
598.         app.HighpassCutoffLabel = uilabel(app.FilterParametersPanel);
599.         app.HighpassCutoffLabel.HorizontalAlignment = 'right';
600.         app.HighpassCutoffLabel.Position = [600 67 80 22];
601.         app.HighpassCutoffLabel.Text = '截止频率';
602.
603.         % Create HighpassCutoffEditField
604.         app.HighpassCutoffEditField = uieditfield(app.FilterParametersPanel, 'nume
      ric');
605.         app.HighpassCutoffEditField.Limits = [0.1 20];
606.         app.HighpassCutoffEditField.HorizontalAlignment = 'center';
607.         app.HighpassCutoffEditField.Position = [690 67 60 22];
608.         app.HighpassCutoffEditField.Value = 0.5;
609.
610.         % Create HighpassPassRippleLabel
611.         app.HighpassPassRippleLabel = uilabel(app.FilterParametersPanel);
612.         app.HighpassPassRippleLabel.HorizontalAlignment = 'right';
613.         app.HighpassPassRippleLabel.Position = [770 67 80 22];
614.         app.HighpassPassRippleLabel.Text = '通带波纹/dB';
615.

```

```

616.         % Create HighpassPassRippleEditField
617.         app.HighpassPassRippleEditField = uieditfield(app.FilterParametersPanel, 'n
       umeric');
618.         app.HighpassPassRippleEditField.HorizontalAlignment = 'center';
619.         app.HighpassPassRippleEditField.Position = [860 67 60 22];
620.         app.HighpassPassRippleEditField.Value = 8;
621.
622.         % Create HighpassStopAttenLabel
623.         app.HighpassStopAttenLabel = uilabel(app.FilterParametersPanel);
624.         app.HighpassStopAttenLabel.HorizontalAlignment = 'right';
625.         app.HighpassStopAttenLabel.Position = [940 67 80 22];
626.         app.HighpassStopAttenLabel.Text = '阻带衰减/dB';
627.
628.         % Create HighpassStopAttenEditField
629.         app.HighpassStopAttenEditField = uieditfield(app.FilterParametersPanel, 'n
       umeric');
630.         app.HighpassStopAttenEditField.HorizontalAlignment = 'center';
631.         app.HighpassStopAttenEditField.Position = [1030 67 60 22];
632.         app.HighpassStopAttenEditField.Value = 12;
633.
634.         % Create DesignFilterButton
635.         app.DesignFilterButton = uibutton(app.FilterParametersPanel, 'push');
636.         app.DesignFilterButton.ButtonPushedFcn = createCallbackFcn(app, @DesignFil
       terButtonPushed, true);
637.         app.DesignFilterButton.FontSize = 14;
638.         app.DesignFilterButton.FontWeight = 'bold';
639.         app.DesignFilterButton.Position = [550 17 200 30];
640.         app.DesignFilterButton.Text = '设计滤波器';
641.
642.         % Create FilterOrderInfoLabel
643.         app.FilterOrderInfoLabel = uilabel(app.FilterDesignTab);
644.         app.FilterOrderInfoLabel.FontWeight = 'bold';
645.         app.FilterOrderInfoLabel.Position = [30 500 1000 22];
646.         app.FilterOrderInfoLabel.Text = '滤波器阶数：未设计';
647.
648.         % Create FilterResponsePanel
649.         app.FilterResponsePanel = uipanel(app.FilterDesignTab);
650.         app.FilterResponsePanel.Title = '滤波器频率响应';
651.         app.FilterResponsePanel.FontWeight = 'bold';
652.         app.FilterResponsePanel.FontSize = 15;
653.         app.FilterResponsePanel.Position = [30 50 1300 430];
654.
655.         % Create LowpassResponseAxes
656.         app.LowpassResponseAxes = uiaxes(app.FilterResponsePanel);

```

```

657.         title(app.LowpassResponseAxes, '低通滤波器响应')
658.         xlabel(app.LowpassResponseAxes, '频率/Hz')
659.         ylabel(app.LowpassResponseAxes, '幅度/dB')
660.         zlabel(app.LowpassResponseAxes, 'Z')
661.         app.LowpassResponseAxes.Box = 'on';
662.         app.LowpassResponseAxes.Position = [20 47 400 300];
663.
664.         % Create HighpassResponseAxes
665.         app.HighpassResponseAxes = uiaxes(app.FilterResponsePanel);
666.         title(app.HighpassResponseAxes, '高通滤波器响应')
667.         xlabel(app.HighpassResponseAxes, '频率/Hz')
668.         ylabel(app.HighpassResponseAxes, '幅度/dB')
669.         zlabel(app.HighpassResponseAxes, 'Z')
670.         app.HighpassResponseAxes.Box = 'on';
671.         app.HighpassResponseAxes.Position = [450 47 400 300];
672.
673.         % Create CombinedResponseAxes
674.         app.CombinedResponseAxes = uiaxes(app.FilterResponsePanel);
675.         title(app.CombinedResponseAxes, '组合带通滤波器响应')
676.         xlabel(app.CombinedResponseAxes, '频率/Hz')
677.         ylabel(app.CombinedResponseAxes, '幅度/dB')
678.         zlabel(app.CombinedResponseAxes, 'Z')
679.         app.CombinedResponseAxes.Box = 'on';
680.         app.CombinedResponseAxes.Position = [880 47 400 300];
681.
682.         % Create FreqzLabel
683.         app.FreqzLabel = uilabel(app.FilterResponsePanel);
684.         app.FreqzLabel.HorizontalAlignment = 'right';
685.         app.FreqzLabel.FontWeight = 'bold';
686.         app.FreqzLabel.Position = [20 358 77 22];
687.         app.FreqzLabel.Text = '频率响应类型';
688.
689.         % Create FreqzDropDown
690.         app.FreqzDropDown = uidropdown(app.FilterResponsePanel);
691.         app.FreqzDropDown.Items = {'幅频响应', '相频响应'};
692.         app.FreqzDropDown.ValueChangedFcn = createCallbackFcn(app, @FreqzValueChang
ged, true);
693.         app.FreqzDropDown.Position = [100 358 100 22];
694.         app.FreqzDropDown.Value = '幅频响应';
695.
696.         % Create ERPWaveformTab
697.         app.ERPWaveformTab = uitab(app.TabGroup);
698.         app.ERPWaveformTab.Title = 'ERP 波形显示和频谱分析';
699.
700.         % Create WaveformAxes

```

```

701.         app.WaveformAxes = uiaxes(app.ERPWaveformTab);
702.         title(app.WaveformAxes, 'ERP 波形图')
703.         xlabel(app.WaveformAxes, '时间/ms')
704.         ylabel(app.WaveformAxes, '幅度/μV')
705.         zlabel(app.WaveformAxes, 'Z')
706.         app.WaveformAxes.Box = 'on';
707.         app.WaveformAxes.Position = [20 100 640 480];
708.
709.         % Create SpectrumAxes
710.         app.SpectrumAxes = uiaxes(app.ERPWaveformTab);
711.         title(app.SpectrumAxes, 'ERP 频谱图')
712.         xlabel(app.SpectrumAxes, '频率/Hz')
713.         ylabel(app.SpectrumAxes, '幅度')
714.         zlabel(app.SpectrumAxes, 'Z')
715.         app.SpectrumAxes.Box = 'on';
716.         app.SpectrumAxes.Position = [700 100 640 480];
717.
718.         % Create LoadDataButton
719.         app.LoadDataButton = uibutton(app.ERPWaveformTab, 'push');
720.         app.LoadDataButton.ButtonPushedFcn = createCallbackFcn(app, @LoadDataButtonPushed, true);
721.         app.LoadDataButton.FontWeight = 'bold';
722.         app.LoadDataButton.Position = [30 690 120 30];
723.         app.LoadDataButton.Text = '加载 ERP 数据';
724.
725.         % Create DataInfoLabel
726.         app.DataInfoLabel = uilabel(app.ERPWaveformTab);
727.         app.DataInfoLabel.Position = [170 690 600 30];
728.         app.DataInfoLabel.Text = '未加载数据';
729.
730.         % Create FileSelectorLabel
731.         app.FileSelectorLabel = uilabel(app.ERPWaveformTab);
732.         app.FileSelectorLabel.FontWeight = 'bold';
733.         app.FileSelectorLabel.Position = [30 650 80 22];
734.         app.FileSelectorLabel.Text = '选择文件';
735.
736.         % Create FileSelectorDropDown
737.         app.FileSelectorDropDown = uidropdown(app.ERPWaveformTab);
738.         app.FileSelectorDropDown.Items = {'sig01', 'sig02'};
739.         app.FileSelectorDropDown.ValueChangedFcn = createCallbackFcn(app, @FileSelectorDropDownValueChanged, true);
740.         app.FileSelectorDropDown.Position = [120 650 100 22];
741.         app.FileSelectorDropDown.Value = 'sig01';
742.
743.         % Create ChannelSelectorLabel

```

```

744.         app.ChannelSelectorLabel = uilabel(app.ERPWaveformTab);
745.         app.ChannelSelectorLabel.FontWeight = 'bold';
746.         app.ChannelSelectorLabel.Position = [250 650 80 22];
747.         app.ChannelSelectorLabel.Text = '选择导联';
748.
749.         % Create ChannelSelectorDropDown
750.         app.ChannelSelectorDropDown = uidropdown(app.ERPWaveformTab);
751.         app.ChannelSelectorDropDown.Items = {'FCz', 'Cz'};
752.         app.ChannelSelectorDropDown.ValueChangedFcn = createCallbackFcn(app, @Chan
nelSelectorDropDownValueChanged, true);
753.         app.ChannelSelectorDropDown.Position = [340 650 100 22];
754.         app.ChannelSelectorDropDown.Value = 'FCz';
755.
756.         % Create ProcessDataButton
757.         app.ProcessDataButton = uibutton(app.ERPWaveformTab, 'push');
758.         app.ProcessDataButton.ButtonPushedFcn = createCallbackFcn(app, @ProcessDat
aButtonPushed, true);
759.         app.ProcessDataButton.FontWeight = 'bold';
760.         app.ProcessDataButton.Position = [470 650 120 30];
761.         app.ProcessDataButton.Text = '处理数据';
762.
763.         % Create WaveformLabel
764.         app.WaveformLabel = uilabel(app.ERPWaveformTab);
765.         app.WaveformLabel.FontSize = 14;
766.         app.WaveformLabel.FontWeight = 'bold';
767.         app.WaveformLabel.Position = [30 600 300 22];
768.         app.WaveformLabel.Text = 'ERP 波形图：';
769.
770.         % Create ERPLabel
771.         app.ERPLabel = uilabel(app.ERPWaveformTab);
772.         app.ERPLabel.FontSize = 14;
773.         app.ERPLabel.FontWeight = 'bold';
774.         app.ERPLabel.Position = [710 600 300 22];
775.         app.ERPLabel.Text = 'ERP 频谱图：';
776.
777.         % Create StatusLabel
778.         appStatusLabel = uilabel(app.ERPUIFigure);
779.         appStatusLabel.HorizontalAlignment = 'right';
780.         appStatusLabel.Position = [20 10 1200 22];
781.         appStatusLabel.Text = '就绪';
782.
783.         % Create ProgressLamp
784.         app.ProgressLamp = uilamp(app.ERPUIFigure);
785.         app.ProgressLamp.Position = [1230 10 20 20];
786.         app.ProgressLamp.Color = [0.8 0.8 0.8];

```

```
787.  
788.         % Show the figure after all components are created  
789.         app.ERPUIFigure.Visible = 'on';  
790.     end  
791. end  
792.  
793. % App creation and deletion  
794. methods (Access = public)  
795.  
796.     % Construct app  
797.     function app = exp7  
798.  
799.         % Create UIFigure and components  
800.         createComponents(app)  
801.  
802.         % Register the app with App Designer  
803.         registerApp(app, app.ERPUIFigure)  
804.  
805.         if nargout == 0  
806.             clear app  
807.         end  
808.     end  
809.  
810.     % Code that executes before app deletion  
811.     function delete(app)  
812.  
813.         % Delete UIFigure when app is deleted  
814.         delete(app.ERPUIFigure)  
815.     end  
816. end  
817. end
```