上海交通大學

项目报告

学院(系): 生物医学工程

专　　业: 生物医学工程

学生姓名: 陈子龙　学号: 516021910459

学生姓名: 李润桓　学号: 516021910192

2019 年　　11 月　　11 日

目录

项目 1

1. 程序开发逻辑

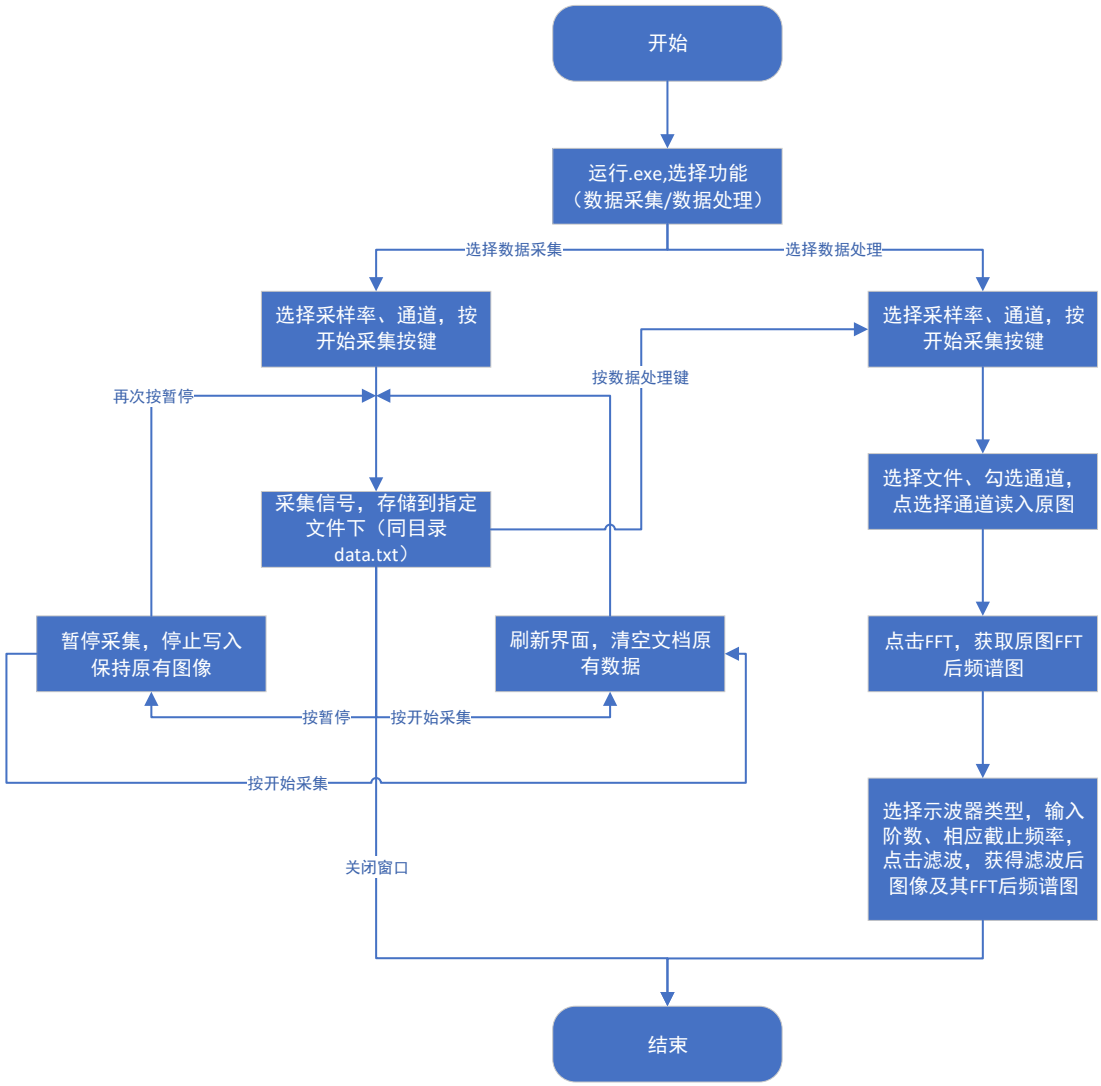

图 1 程序拓扑图

1.1 数据采集部分

1.1.1 画图

在最开始的设计中，我们的逻辑是：当用户使用程序开始采集数据时，程序根据用户设置的采样率设置中断函数的周期。在中断函数中，程序会通过 data.Get 函数获得端口的数据。程序记录当前端口接收到的数据，并和上一个数据连成直线，由于程序将 axes 设置为在一次数据采集的过程中每次新增直线时不清空之前的图像，因此用户看到的是一条连续的折线。

然而使用这个方法画图的话，由于 axes 里的点会越来越多，因此每个周期的运行时间都会变长，难以实现高的采样率，因此我们使用 animatedline 和 addpoint 命令进行画图，使得每个周期的运行时间缩短至 10ms 以下，理论采样率可以达到 100Hz，并且不会出现 timer 函数运行时间随着程序运行时间增长而增加的问题。

1.1.2 图像显示范围与缩放

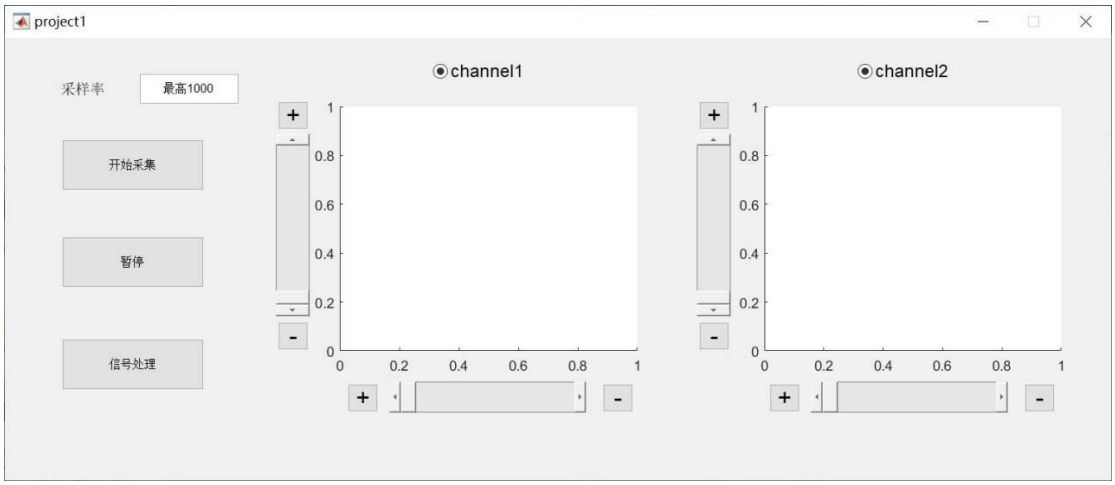

图 1.AI 的 GUI

整个程序的 GUI 如图一所示。Axes 在 X（时间轴）、Y（电压轴）两个坐标轴旁边各有一个 slider 和两个控制缩放的按钮，控制图像显示区域的变量主要有六个：X（当前点的横坐标/时间）、Y（当前点的纵坐标/电压）、XP（当前显示区域中心点的横坐标）、YP（当前显示区域中心点的纵坐标）、regx（X 轴显示范围）、regy（Y 轴显示范围）。X 轴显示的区域是[XP-0.5*regx,XP+0.5*regx]，其中 XP 由 slider 控制，当 slider 在最左边时，XP=0，当 slider 在最右边时，XP=X，当 XP<0.5*regx 时，XP=0.5*regx。Slider 两边的“+”、“-”按钮控制 regx 的大小。Y 轴显示的区域是[YP-0.5*regy,YP+0.5*regy],与 X 轴的控制方式同理，YP 的范围是[-0.5*regy,0.5*regy], Slider 两边的“+”、“-”按钮控制 regy 的大小。

1.1.3 开始、停止与继续采集

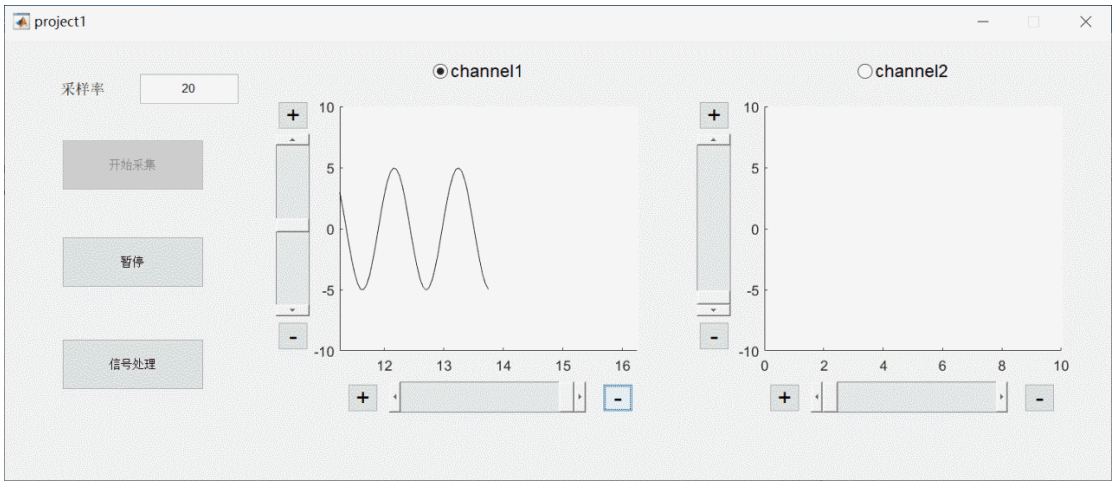

图 2.采集到的正弦信号

当开始按键被摁下，程序会清空当前的图像，并开始画图。当暂停按键被按下，程序会将 flag 变量取反，如果 flag 为 1，则开始计时，中断函数正常运行，如果 flag 为 0，则停止计时，中断函数停止运行，因此当用户第一次摁下暂停时，程序停止，再次摁下暂停，程序会从停止的地方继续运行。

1.1.4 通道选择

通道的选通状态记录在 channels 数组中，当 channels 对应的为 0 时，通道未被选择，中断函数中不运行对应通道的画图程序。

1.1.5 使用鼠标取点

通过开启 datacursormode，使得程序允许用户随时使用鼠标获取图像上点的值。

1.1.6 数据存储

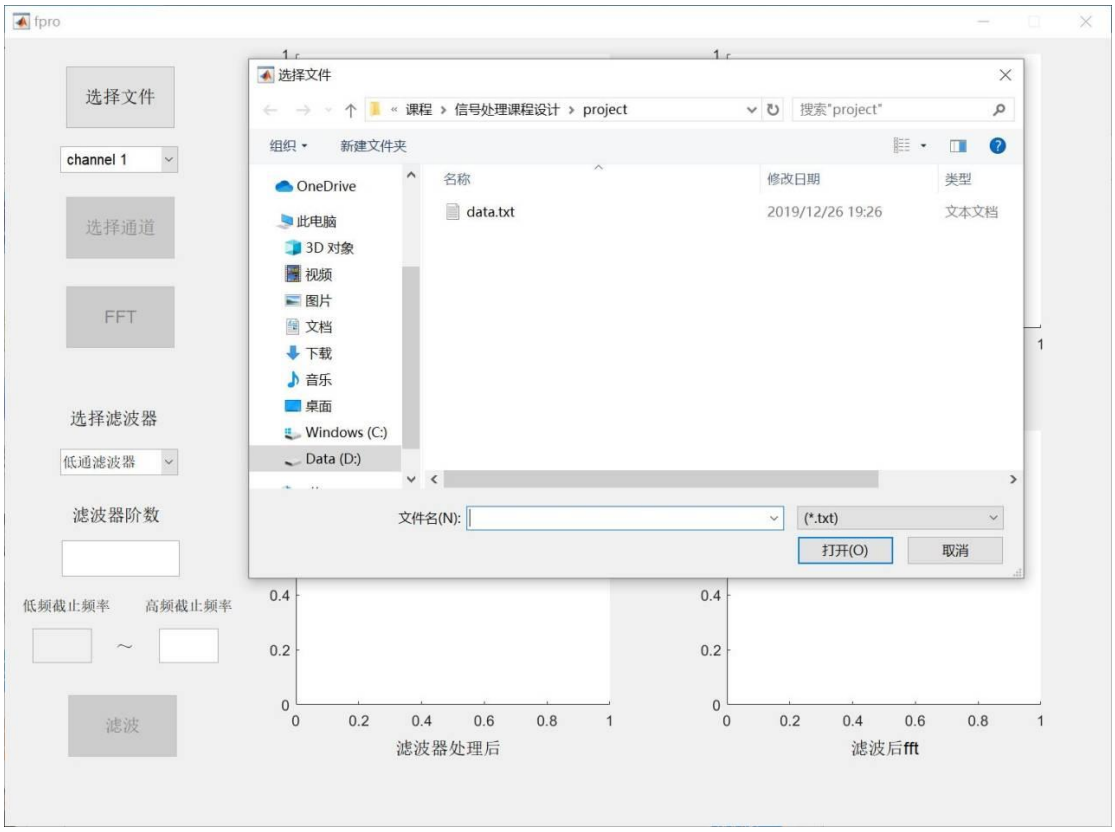

图 3.选择要处理的数据

数据的写入主要依靠 Matlab 的 fopen，fprintf， fclose 等函数。通过 fopen 打开同目录下的 data.txt 文件通过 fprintf 写入，每次读入一个数据存储一个数据。数据的格式为：采样率放在第一行，之后每一列数据（除第一行外）代表一个通道的记录。全部完成后 fclose 关闭文件。

1.2 数据处理部分

本程序实现的是所采集数据被选择的单通道的处理。数据处理可细分为以下三个功能的实现：数据的读入（包括文件及通道的选择）、FFT、滤波。我们编写了相应的函数来实现这些功能。

数据的读入：在之前记录中，我们已经将数据按我们所需的格式存好（采样率放在开头，之后每一列数据代表一个通道），给入地址，通过 fopen, fscanf, fclose 等文件相应操作，读出数据，通过所给通道选择相应数据，将所需数据读出，并显示。

FFT 的实现：为了更好使用 FFT，我们通过补零，将数据长度扩展至最近的 2 的阶次，通过 Matlab 自带的 fft 函数对其进行 FFT 计算，得到其频谱图。频谱图的 X 轴通过所

记录的采样率计算得出。整个函数包括四个输入的变量：Signal（数据）, Fs（采样率）, handles（GUI 所需），type(用于区分数据来源于滤波前还是滤波后，与显示相关)

滤波函数:我们使用了 Matlab 自带的巴特沃斯滤波器函数 Butter()，编写了相应的滤波函数，AI_My_Filter。整个函数所需输入的变量为:Signal(数据)，Fs(采样率)，n(滤波器阶数),type(滤波器类型)，FH(高频截止频率)，FL（低频截至频率）

整个数据处理界面的使用流程如示意图所示，在选择好数据位置及通道后，将显示相应通道数据，按 FFT 键进行原图的 FFT，得到其频谱图。滤波器部分，选择其类型，输入阶数，相应的截至频率，按下滤波按键，调用滤波函数和 FFT，得到原图滤波后图像及其 FFT 后的频谱图，完成整个数据处理及显示过程。数据处理部分如图 4-8 所示。

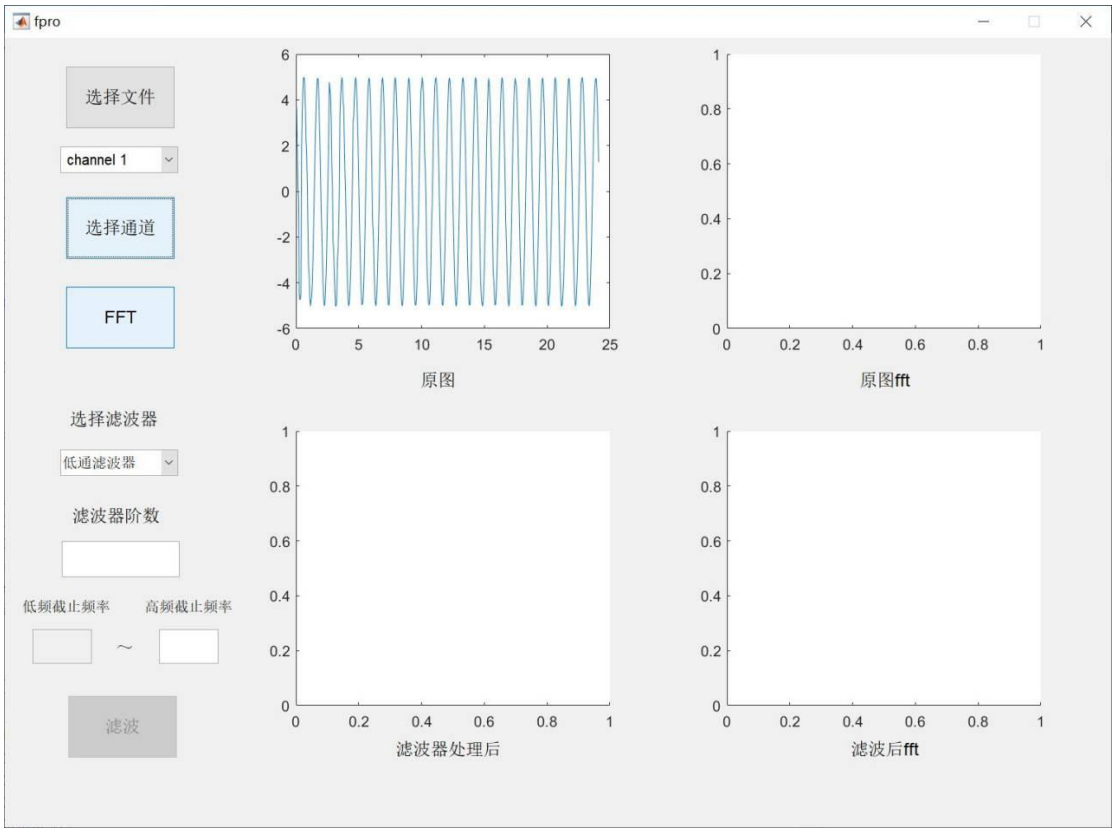

图 4.导入正弦函数数据

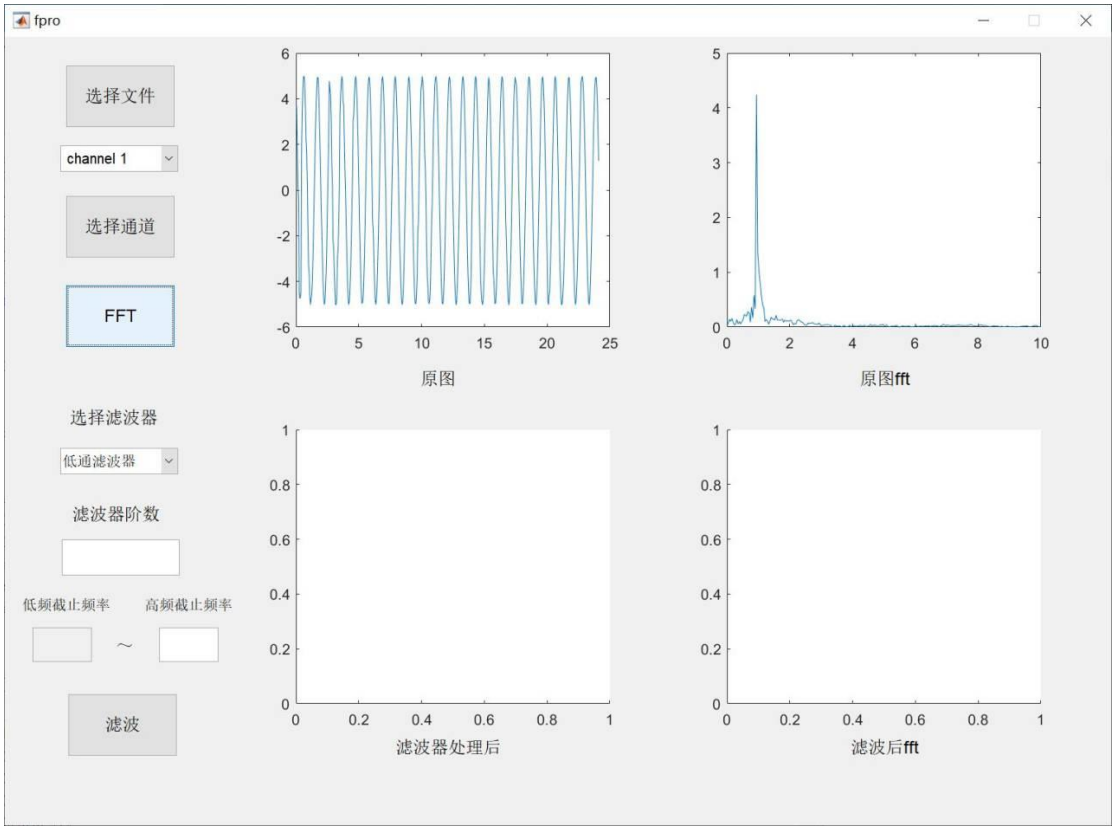

图 5.对正弦函数进行 FFT

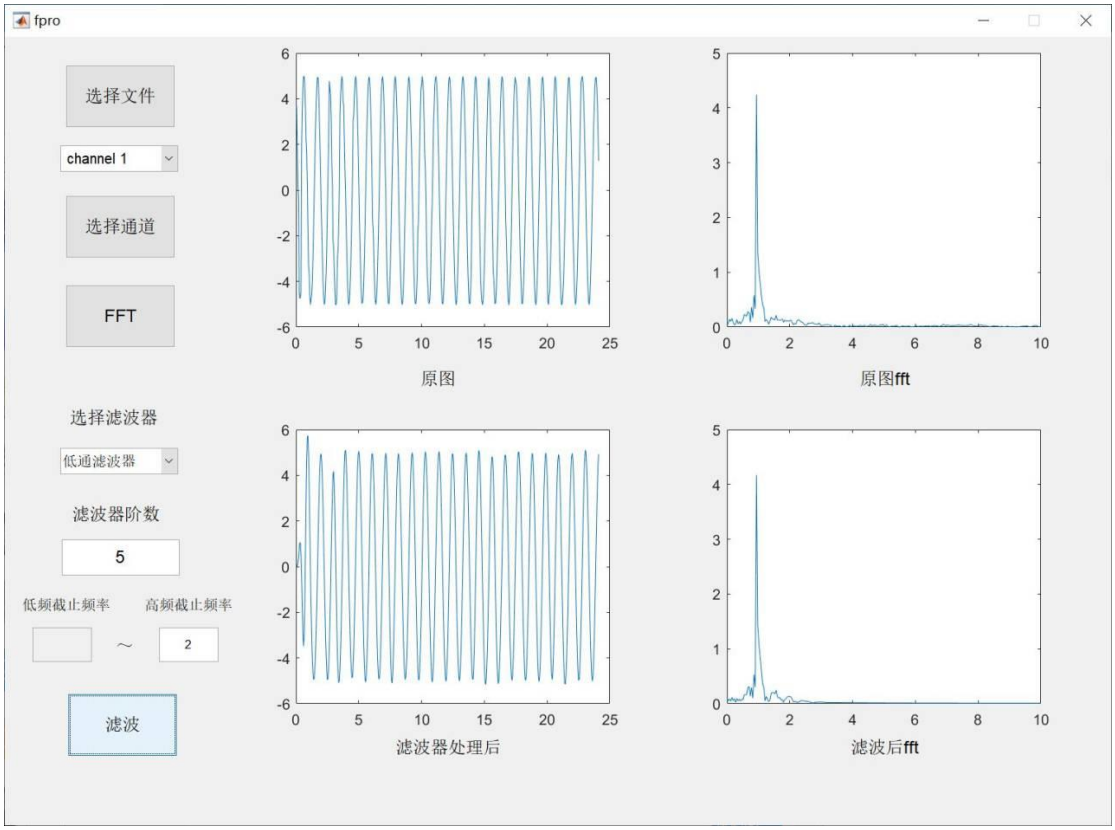

图 6.对正弦函数进行滤波

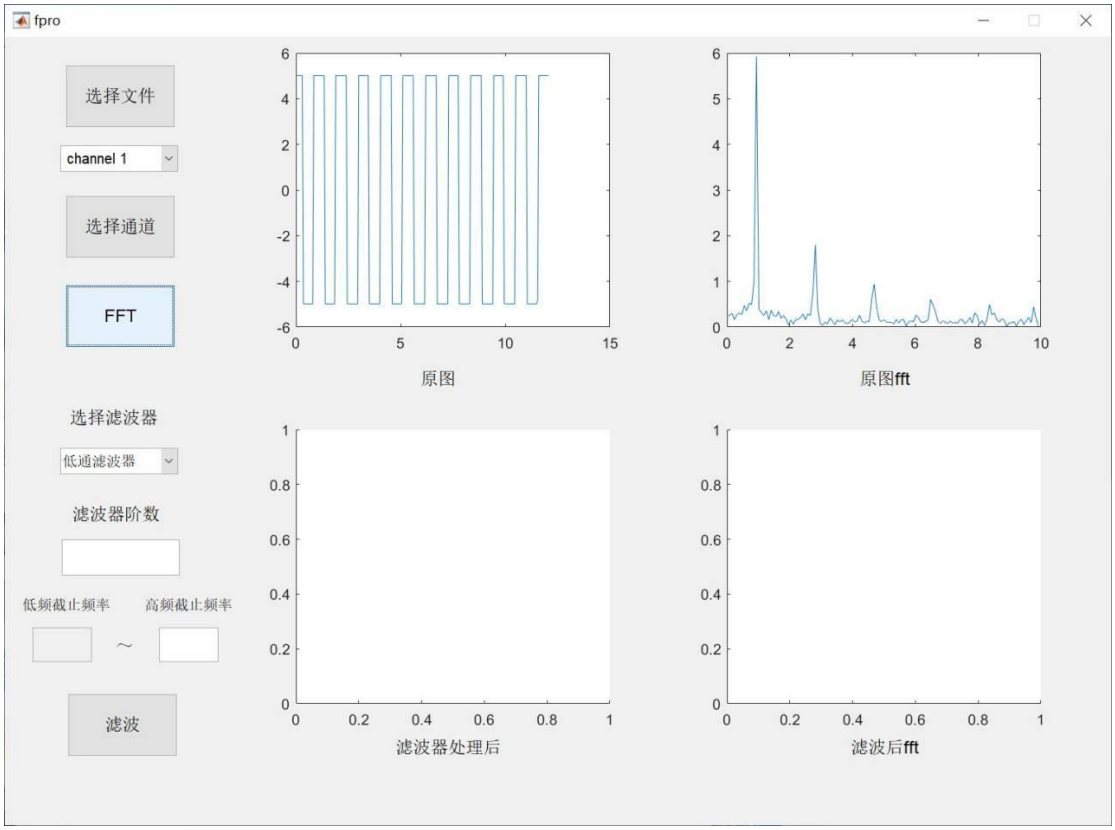

图 7.对方波进行 FFT

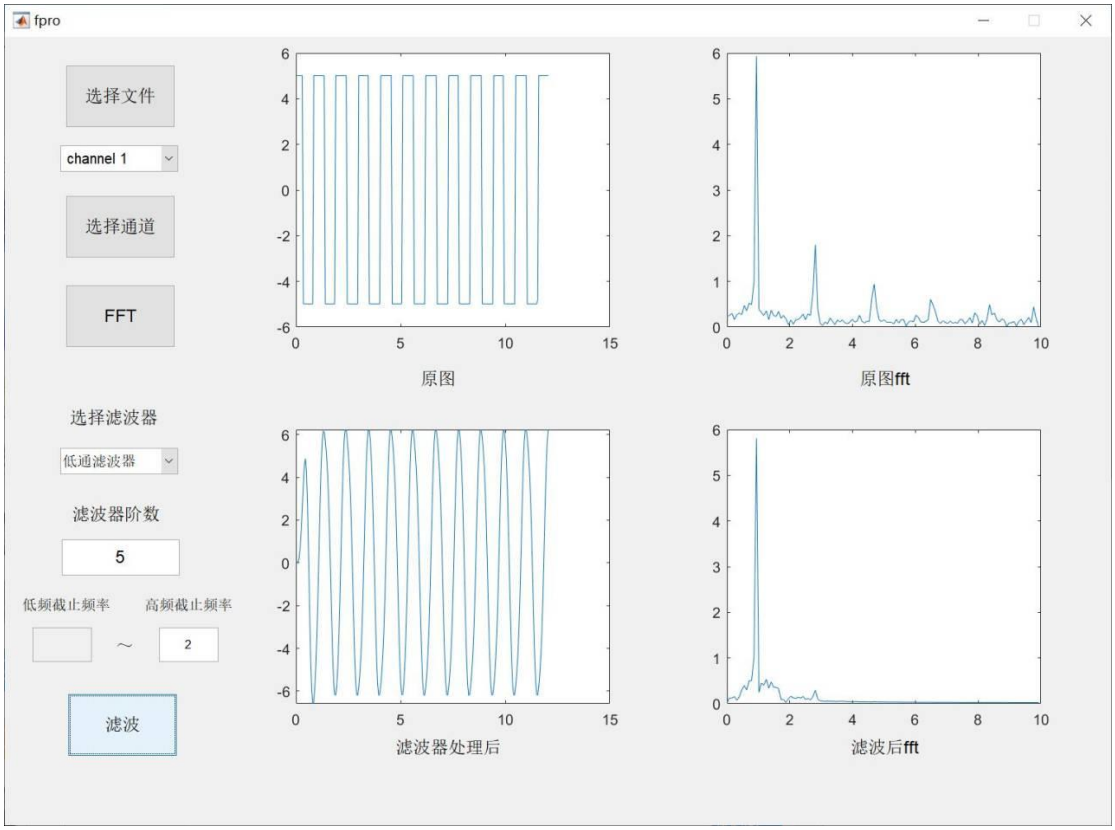

图 8.对方波进行滤波

2. 出现的问题及解决方案

2.1 数据异常

具体描述：在第二次按采集数据，刷新界面并清空数据，重新进行记录时，会发现显示的数据出现乱跳的现象，而不是像第一次采集时得到索要采集的波形。

错误原因：排查后发现，出现该问题的原因是在第二次按下采集数据调用相应函数时，原来的时钟中断没有关掉，导致系统出现了两个时钟中断，使得采集的数据发生错乱。

解决方式：在第 N 次按下采集数据(N>1)时，清空原来的时钟中断。

2.2 程序运行越来越慢

具体描述：随着采集数据的增多，程序的运行速度会越来越慢，导致不能很好的满足采样率的要求。

错误原因：通过断点调试，我们发现问题出在绘图函数中，我们通过 plot 函数这进行绘图（保留原图，将新的数据点与原图最后的数据点相连），该过程会随着数据量的增多运行时间变长，推测可能是保留原图后继续绘制时其函数可能会将之前的数据读入，随着数据的增加导致运行时间的变长。

解决方案：将动态显示由 plot 函数改为 animatedline 和 addpoint 函数，程序的运行时间显著减小，且不会随着运行而增长。

2.3 静态变量在 GUI 中无法使用

具体描述：静态变量在同一.m 文件下可以自由传递，但是在 GUI 使用中却无法自由传递。

错误原因：可能与 GUI 内部的原理有关，原因未找到。

解决方式：我们将所需的变量加入 handles 结构体中，通过传递 handles 达到相同的作用。

3. 思考与讨论

3.1 如何针对不同频率的信号设置合适的采样率，并分析设置采样率时考虑的因素；

根据奈奎斯特采样定律，采样率的 1/2 需要大于所采集的信号的频率，在此基础上，采样率越大越好，但需要注意其上限 fLimited（后面会具体解释）。

在设置采样率时，需要考虑的因素主要是程序本身的运行时间，由于程序本身采集一个数据并进行相应的操作需要一定的时间，其会限制一个采样率的最大值（fLimited），当采样率(Fs)大于这一限制值 fLimited 时，实际的采样率为 fLimited,导致错误。

3.2 分析 USB-4704 的模拟输入功能可采集信号的频率范围，若输入信号在该范围外，会出现哪些问题，并探讨可能的解决方案。

利用 tic、toc 指令得出每次中断函数运行的时间在 10ms 左右（不考虑随着程序的运行逐渐变慢），因此可采集的信号最高频率为 50Hz。如果信号超出这个范围，那么会造成：1.由于程序中点对应的时间并非真实运行时间，而是用当前点的个数乘中断函数的周期计算出来的，因此在时间轴不变的情况下，整个图像会被压缩，而且图像显示的时间会慢于真实时间。2.FFT 后会失真，高频成分会和低频成分产生混叠。

解决方案：1.减少中断函数运行所需的时间，提高可采集信号的频率范围。2.由于中断函数中 plot 用时最长，因此可以采集多个数据保存在程序中最后一次画上。3.避免输入超出可采集范围的信号。