Problems：

Problem 1.访问 MIT-BIH Arrhythmia Database (mitdb)，选择第 111 号、115 号 两个 record 的 MLII 导联心电数据前 30 秒钟，下载、读入并显示
Problem 2.计算平均心率（30 秒钟），短时心率（5 秒钟内心率，每秒更新 1 次）
Problem 3.短时傅里叶变换，选择合适的窗宽，判断是否存在工频等噪声
Problem 4.计算分析基线漂移等低频噪声
Problem 5.设计滤波器，滤除工频、基线漂移等噪声
Problem 6.设计算法检测 R 波，计算 R-R 间期
第 3 项需对两条数据进行处理、分析，其余只需完成一条数据

数据部分说明：

数据部分存入 data 文件夹中 111.csv 以及 115.csv
分别代表 MIT-BIH Arrhythmia Database (mitdb) 111、115 channel
采样率：360 Hz，持续时间：60 s

代码部分说明：

Process.m 中，定义了如下的函数帮助实现功能，在文件最后。

```
function [second, signal] = get_data(file_path,time_start,time_end)...

function [] = drawSignal(s1,t1,s2,t2)...

function [avg_Hb,points] = get_Hb(s, start_time, end_time)...

function Hb = get_Hb_5s(file_path)...

function s = baseline_filter(s1,fs,fmax)...

function s = bandpass_60(s1,fs)...

function [R,RR_interval] = get_Rwave(s,time_start,time_end)...
```

问题回答与注解：

Problem1.

下图是两个通道的原信号表示，可以看到存在一定的基线漂移。

在读取数据时，调用了 get_data 函数，其中最重要的一部分是将 csv 文件中的时间部分转化为实际的秒数

在 csv 文件中，时间部分的数据类型为：“’M:SS.sss’”，利用 matlab 的 datetime 与 time 进行化解。

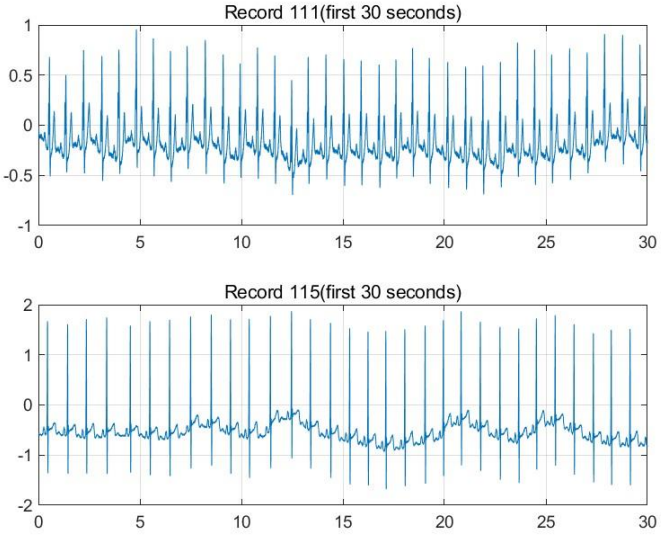


Figure1. Channel 111,115　first 30 seconds signal

Problem2.

Channel111 的 30s 平均心率是 70

Channel115 的 30s 平均心率是 62

在 Process.m 中，对应的变量是 avg_Hb_1 以及 avg_Hb_2

实际操作中，由于原信号存在基线漂移，所以在判断一个心动周期时，是利用整个函数的最大值的 0.4 作为阈值进行判断，效果还不错。

下图是每 5s 更新一次的心率，利用 5 秒内所获得的 RR 间期平均进行心率估计

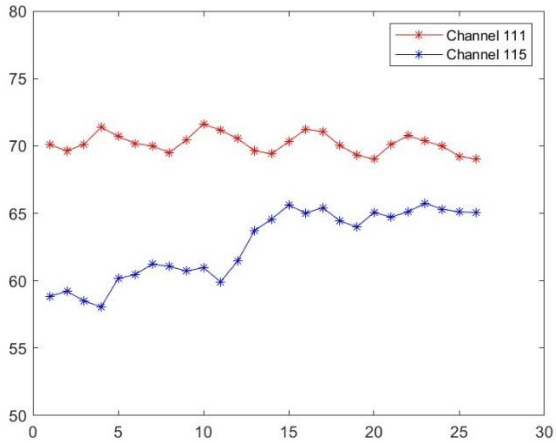


Figure2.两个通道的 5s 心率更新

具体数值如下：Hb1 = 70.0843608046723 69.6324951644101 70.1298701298701
71.3813615333774 70.7037643207856 70.1754385964912
69.9935191186001 69.4980694980695 70.4500978473581
71.6180371352785 71.1696869851730 70.5421293272371
69.6324951644101 69.4087403598972 70.3125000000000
71.2401055408971 71.0526315789474 70.0389105058366
69.3196405648267 69.0095846645368 70.0843608046723
70.7732634338139 70.3583061889251 69.9935191186001

69.2307692307692 69.0095846645368

Hb2 = 58.8555858310627 59.2186429061001 58.4969532836831

58.0645161290323 60.1671309192201 60.4618614415675

61.2331679659816 61.0600706713781 60.7167955024596

60.9738884968243 59.9167822468793 61.4509246088193

63.7168141592920 64.5739910313901 65.6036446469248

65.0210716435882 65.4149000605694 64.4391408114558

63.9810426540284 65.0602409638554 64.7191011235955

65.1092690278825 65.7534246575342 65.3061224489796

65.1092690278825 65.0602409638554

Problem3.

对两个信号进行短时傅里叶变换，可以看到 record 111 存在 60Hz 的工频干扰，而 record115 已经去除这部分的工频干扰了。所以后续对 record 111 进行滤波处理

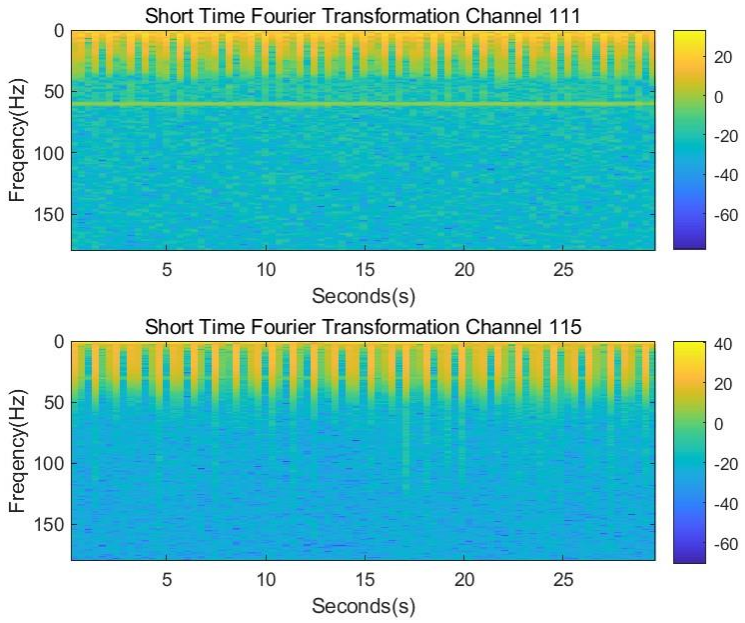


Figure3. STFT of both records

Problem4.

利用傅里叶变换，对于 Record 111 我们可以看到信号在 0Hz 处拥有很大的频谱，低频部分拥有很大的频谱，这就是基线噪声的影响，过于低频的信号并不属于心电信号的有效成分。

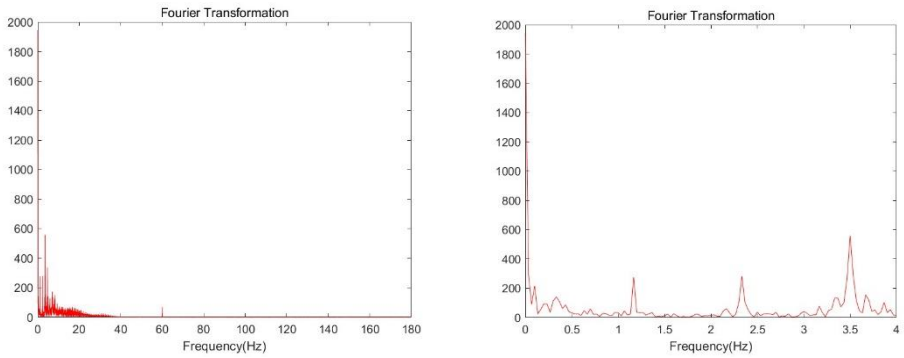


Figure4.1 信号的傅里叶变换后

而心电信号的频率范围在 0.5Hz~40Hz，所以可以认为这部分噪声就是基线漂移噪声

通过一个简单的低通滤波器，我们可以得到如下

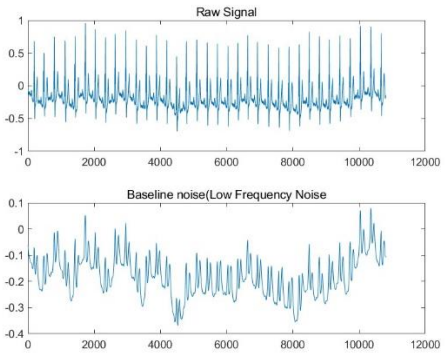
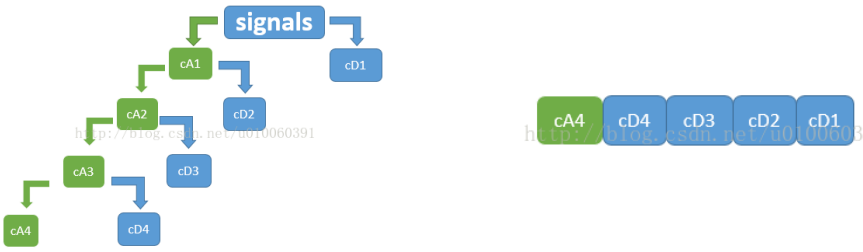


Figure4.2 原信号和基线漂移噪声

Method2：也可以采用 wavelet 的方法进行提取，具体代码见 Process.m -> Problem4 -> Method2。

以 level 为 4 的 wavelet 为例，其实际做的事情就是将信号分为高频（细节系数）以及低频（近似系数）(摘自 https://blog.csdn.net/weixin_29369363/article/details/116062047)



由于心电信号比较复杂，在提取基线漂移部分，我们用了 level 为 8 的 wavelet 滤波器

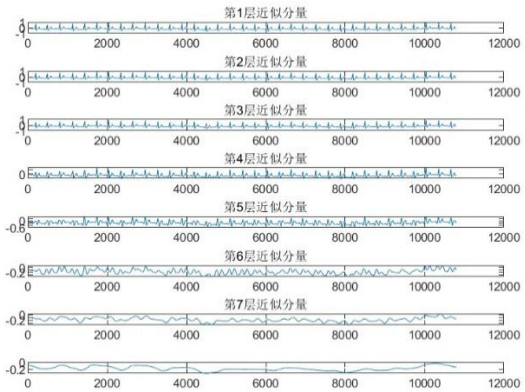


Figure4_2_1 小波变换依次得到的近似分量

具体获得的基线分量与低频信号进行比较

可以提取到更好的基线漂移分量。

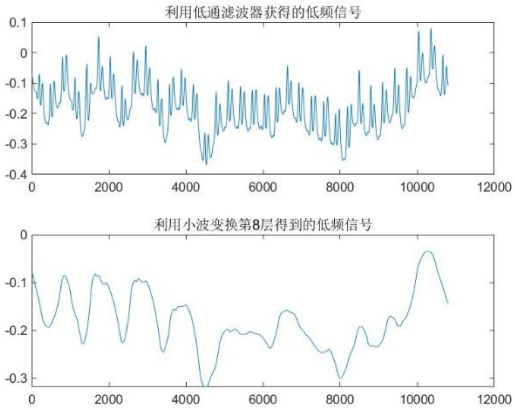


Figure4_2_2. 对比

心电信号的基线漂移分量是由于多种原因造成的，比较主要的原因是基线漂移是由于人的呼吸运动和电极在皮肤上的微小移动,频率通常小于 5Hz，也有可能是器件本身的原因。

Problem5.

Record 111：通过基线漂移噪声滤波器以及 60Hz 带组滤波器，可以得到最后处理到的信号。

基线漂移噪声可以通过简单的高通滤波器，或者原信号减去低通滤波器，也可以利用小波变换找到的噪声趋势，这部分的两种实现在本代码中都有 baseline_filter()函数中，可自行查看。

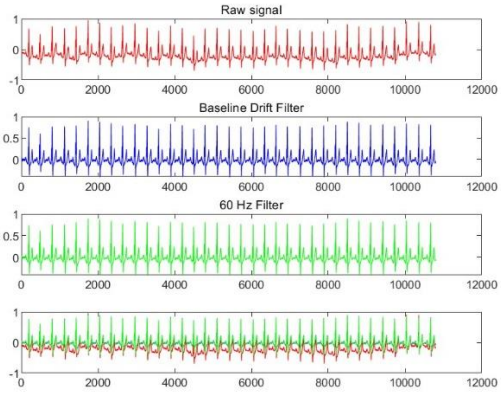


Figure5.1. 滤波器滤后的信号

其频谱图如下所示

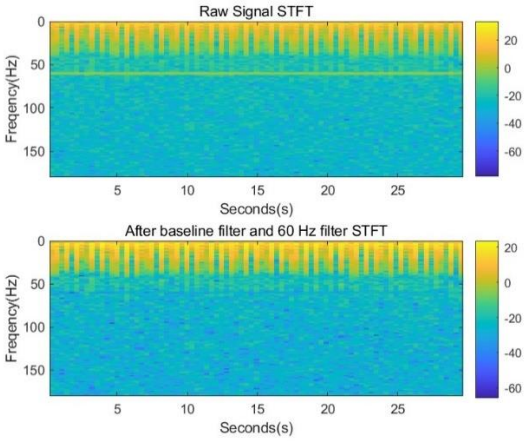


Figure5.2 滤波器滤后的频谱

同样在这道题目中，也可以利用 level 为 8 的 wavelet 进行滤波，得到以下信号

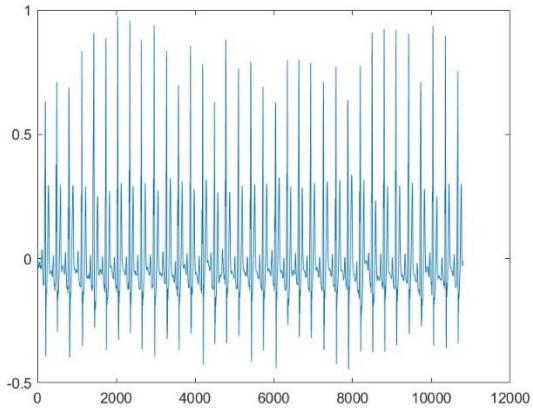
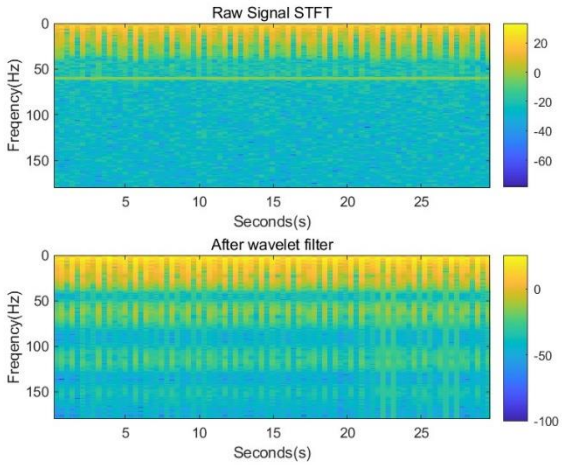


Figure5.3 利用 Wavelet 进行滤波

具体参数中，将第一层，第二层和第三层的高频分量滤除，同时将低频分量滤除，得到了如上的图示，其频谱范围具体如下所示：



可以看出，利用 wavelet 组进行滤波时，频率并不是连续的，会导致产生很多新的信号成分，所以并不推荐用单纯的 wavelet 滤除工频信号。

Problem6.

Record 111: 通过算法定位到每个 R peak，并计算出 RR_interval 存入 rr_interval 中

得到 RR_interval 平均值为：0.8555

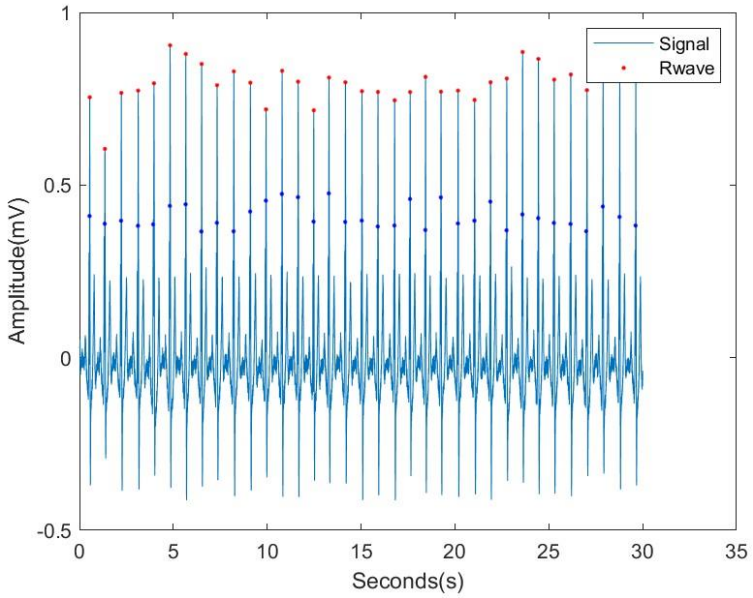


Figure6. R peak detection