

基于信号处理方法的一维模拟信号降噪分析

石金泽 朱政颖

2022 年 1 月 3 日

1 关键词

降噪平滑滤波小波变换相似性分析

2 简介

3 模拟电路信号噪音

3.1 定义

3.2 电路噪音分类及特点

3.3 信噪比

在电路系统中信噪比，SNR被用来衡量噪声和信号的关系，信噪比约到，信号质量越好，噪声越小。按照功率定义：

$$SNR = 10 \lg \frac{P_N}{P_S} \quad (1)$$

其中 P_N 为噪音平均功率， P_S 为信号平均功率，按照电压幅值定义：

$$SNR = 20 \lg \frac{V_N}{V_S} \quad (2)$$

其中 V_N 为噪音电压幅值， V_S 为信号电压幅值。

4 低噪声放大器设计

4.1 定义

5 常见一维模拟信号降噪方法

5.1 滤波

模拟电路实际情况下的信噪比受到的影响来源于电路内部和电路外部，电路内部的噪音由器件的物理响应决定，外部的噪音受到环境电磁干扰、温度涨落等等因素的干扰。跟据不同情况下的噪音特性曲线我们可以通过滤波的方式进行降噪。

5.1.1 傅里叶变换(FT)

一个一维信号可以表示为 $u(t)$, 表示测量值关于时间的函数, 该函数数学上如果满足绝对可积条件:

$$\int_0^\infty |u(t)|^2 dt < +\infty \quad or \quad \sum_{n=0}^\infty |u_n(t)|^2 < +\infty \quad (3)$$

则可以对该信号进行频域的傅里叶变换, 以连续信号举例:

$$u(t) = \int_{-\infty}^{+\infty} A(\omega) e^{-j\omega t} d\omega \quad (4)$$

其中 $A(\omega)$ 为频率成份为 ω 的信号的占比强度:

$$F[u(t)] = A(\omega) = \int_0^{+\infty} u(t) e^{j\omega t} dt \quad (5)$$

换言之, 通过傅里叶变换可以计算出一个信号的频谱信息, 在实际计算过程中, 通过快速傅里叶变换FFT算法可以最高效地将一段时域信号转化为频谱信息。

通过傅里叶反变换也可以将频谱信息反推出时域信息:

$$F^{-1}[A(\omega)] = u(t) \quad (6)$$

5.1.2 基于傅里叶变换进行滤波

来自实验环境中的信号按照有无特殊的频谱信息可以进行不同的滤波方式, 如果噪音有特殊的频率信息, 在频谱上进行剔除, 再通过傅里叶反变换即可进行一定程度降噪。例如 $1/f$ 噪声随着频率增加强度逐渐减弱, 可以通过高通滤波把响应的噪声成份滤掉。但通常情况下环境噪音没有明显的频谱信息, 此时可以根据探测信号的频段进行一定频率范围的带通滤波。例如射频电路RF信号的频率范围在300kHz-300GHz, 可以通过带通滤波, 滤掉高频或者低频信息, 获得我们关注的频域信息。

5.1.3 傅里叶变换局限性

在时域或者空域上获得的信号通常有一定的时间或者空间范围, $[0.T]$, 如果需要观察频率信息在该时间或者空间尺度上的变化, 需要采取一定的窗口长度 T/N 将信号进行时域或者空域的分割, 利用FFT获得相应窗口内的频谱, 获得N个频谱, 从而得到频谱信息随时间的变化。这是短时傅里叶变换(STFT)。

$$\Delta t \Delta f > C \quad (7)$$

海森堡不确定原理从根本上决定了频率精度和时间精度的矛盾性, 由于傅里叶变换的基矢函数是时域无限的, 因此具有确定的单色性, 提高FFT的精度需要充足的时间信息, 当对频率信息的时间分辨率提出更高要求时, 意味着需要缩短时间窗口, 必然会FFT信号的频谱解析精度。一方面时间和频率分辨率固定, 不能实现动态范围调整, 对于不同情况选择一个合适的窗宽非常困难。对于低频信号, 为了更好地确定频率, 时域区间需要宽一些, 即时间不确定度 Δt 大一些, 根据海森堡测不准原理, 频率不确定度 Δf 会变小; 对于高频信号, 为了更好地在时域定位, 时域区间需要窄一些, 即时间不确定度 Δt 小一些, 根据海森堡测不准原理, 频率不确定度 Δf 增加。

其次, 在一些突变点, 傅里叶变换仍然需要大量的三角函数进行线性叠加拟合趋近, 对于高阶导数非连续的信号拟合情况较差。

改进的方法之一是通过重新设计基矢函数, 例如小波函数变换。

5.2 小波变换

5.2.1 小波母函数

小波变换中的基矢函数，即小波母函数 $\psi(t)$ 需要满足以下条件：

1. 紧支撑性，存在 $a > 0$, 使得任意 $|t| > a, \psi(t) = 0$
2. 波动性：

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (8)$$

3. 容许性：

$$\int_{-\infty}^{+\infty} \frac{|\psi(f)|^2}{|f|} < +\infty \quad (9)$$

4. 正交性：

$$\int_{-\infty}^{+\infty} \psi_i(t) \psi_j(t)^* = \delta_{ij} \quad (10)$$

常见的满足条件的小波母函数有sinc函数，Meyer's函数，Morlet函数，Mexican hat函数等等。

5.2.2 连续小波变换(CWT)

$$CWT_x^\Psi(\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} u(t) \psi\left(\frac{t}{s} - \tau\right) dt \quad (11)$$

通过改变 s ，小波母函数的时域尺度也跟随变化，紧支撑性决定了小波母函数具有一定的时间窗口，其频域信息也具有一定频率范围，可以看作一个带有滤波效果的窗口， s 增大时，小波的时域尺度增大，对应的频率变小， s 减小时，小波的时域尺度减小，对应的频率增大。

5.2.3 从基函数的角度对比FT,STFT, CWT

FT 的基函数相当于在时域无限长的单色波，不具有紧支撑性，只能筛选频率信息，无法进行时间分辨；

STFT的基函数是具有时间截断的单色波，具有紧支撑性，能筛选出频率信息，也有时间分辨率，频率和时间信息都有一定带宽，但由于窗口长度是固定的，对于不同的频率信号，时间分辨率和频率分辨率相同，所以窗口长度难以确定；

CWT的基函数具有紧支撑性，频率信息和时间窗口通过参量 s 耦合在小波函数的形式里，对于不同的频率有一定的动态调节范围。

$$\frac{1}{\sqrt{s}} \psi\left(\frac{t}{s}\right) \quad (12)$$

5.2.4 离散小波变换(DWT)及Mallet算法

常见的小波函数有Haar小波，Daubechies(dbN)小波，Symles(symN)小波系，Coiflets(coifN)小波等

对于计算机实际处理信号的过程，没有办法进行连续信号的处理，常用Mallet算法进行离散小波变换。设尺度函数为 $\phi(t)$,小波函数为 $\Psi(t)$,在尺度 j 下进行小波变换分解：

高频部分：

$$A_j x = \langle u(t), \phi(t) \rangle \quad (13)$$

低频部分:

$$D_j x = \langle u(t), \Psi(t) \rangle \quad (14)$$

例如在尺度1下对于一个最高频率为 F_s ,采样点为N的信号,通过一次基于小波变换的半子带滤波和2倍下采样,可以得到两个分解信号,频率范围 $[0, F_s/2]$,采样点N/2的低频带信号,频率范围为 $[F_s/2, F_s]$,采样点为N/2的高频带信号;

在尺度j的结果上可以继续进行将低频带信号进行尺度j+1的分解:

$$A_{j+1} x = \sum_k h(k - 2n) A_j x \quad (15)$$

$$D_{j+1} x = \sum_k g(k - 2n) A_j x \quad (16)$$

对于分解尺度为J的算法下信号u(t)可以分解为:

$$x(t) = A_0(x) = A_j x + \sum_{j=1}^J D_j x \quad (17)$$

5.2.5 阈值法小波系数重整

对于信号的高频部分, Mallet算法获得了高分辨率的时域信息,对于信号的低频部分,通过降低时域分辨率获得了频域的高分辨率。不同频率区间的分解信号具有不同的小波系数,通常而言,噪声的小波系数分布分散且幅值较低,而信号的小波系数分布集中,幅值较高,通过阈值法对小波系数进行调整,可以提取信号,降低噪声。

阈值法分为硬阈值法和软阈值法,硬阈值处理函数:

$$\sigma_\lambda^H(\omega) = \omega \quad if \quad |\omega| \geq \lambda; \quad 0, else \quad (18)$$

软阈值处理函数:

$$\sigma_\lambda^S = [sgn(|\omega|) - \lambda], |\omega| \geq \lambda; \quad 0, else \quad (19)$$

阈值的选取方式有多种:

a. VisuShrink阈值:

$$\lambda = \sigma \sqrt{2 \ln N} \quad (20)$$

b. Sqtwolog阈值:

$$\lambda = \sqrt{2 \ln N} \quad (21)$$

c. Rigsure阈值:

是一种基于Stein无偏似然估计的自适应阈值选择,对于给定的初始阈值,利用似然估计将该阈值最小化得到最终的 λ ;

d. Minmax阈值:

$$\lambda = 0, if \quad N \leq 32; \quad 0.3936 + 0.1829 \frac{\ln N}{\ln 2}, N > 32 \quad (22)$$

e. Heursure阈值:

$$\beta = \frac{\sum_{i=1}^N |d_i(i)|^2 - N_j}{N_j}, \gamma = \sqrt{\frac{1}{N_j} \left(\frac{\ln N_j}{\ln 2} \right)^3} \quad (23)$$

5.2.6 信号重构

通过阈值法对小波系数进行调整后，再利用mallet算法进行逆向信和恢复，即可得到降噪后的信号。

6 MATLAB实现

6.1 相似度分析指标

这里采用Matlab对PA信号进行滤波和小波变换分析，通过以下几个指标和Ground信号进行相似度分析。

6.2 SSE

设初始信号为(t),Ground信号为g(t),则SSE定义为:

$$SSE = \sum_{i=1}^{length(t)} |u(t(i)) - g(t(i))|^2 \quad (24)$$

处理前SSE数据汇总:

```
brain1 : 522.2219
brain2 : 624.5647
brain3 : 439.8298
brain4 : 437.2567
brain5 : 547.6440
brain6 : 633.7321
brain7 : 623.6900
brain8 : 596.6787
```

SSE可以衡量两个信号每个位置的值差异的累积。

6.3 相关系数

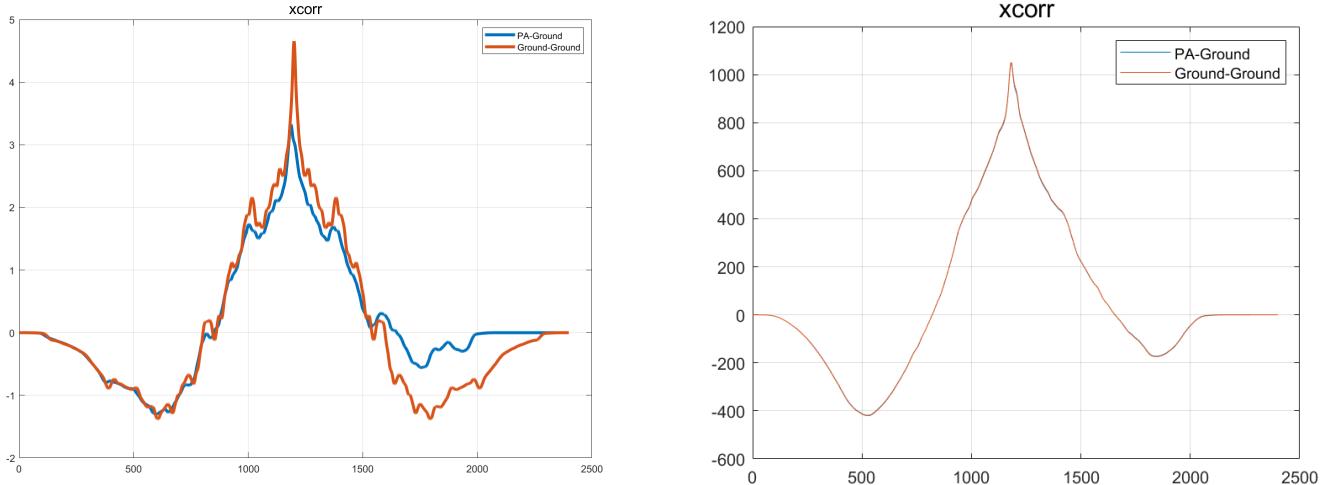
$$r_{ug} = corr(u(t), g(t)) \quad (25)$$

r_{ug} 越接近1，两个信号相似度越高。以256个通道数据的相关系数均值作为该指标:

```
brain1 : 0.7181
brain2 : 0.6648
brain3 : 0.7960
brain4 : 0.7881
brain5 : 0.7593
brain6 : 0.7702
brain7 : 0.7583
brain8 : 0.7773
```

6.4 互相关检验

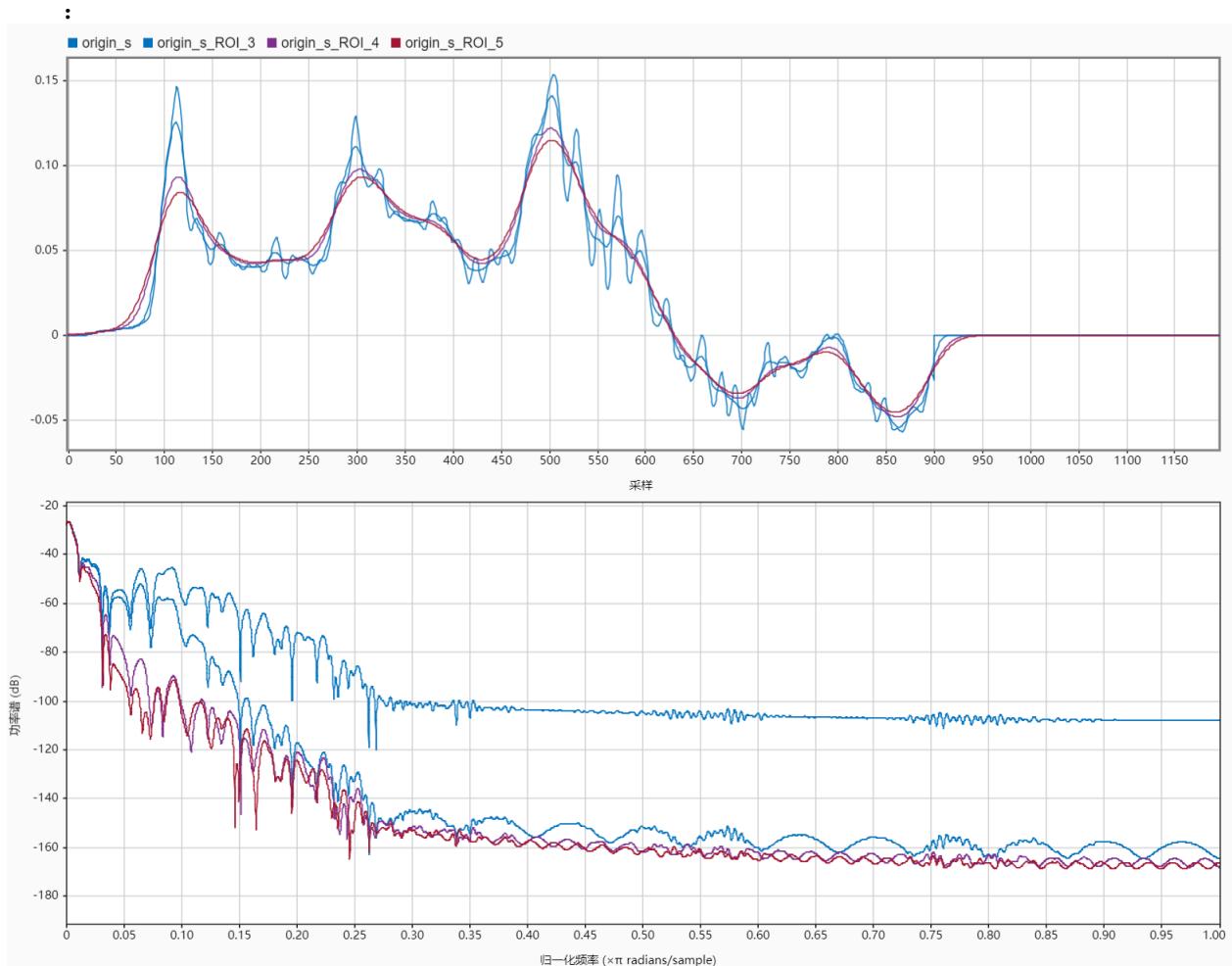
MATLAB 内置了互相关检验函数 `xcorr`, 可以用来检验两个信号的相似度。



虽然单通道的 `xcorr` 在尾部有较大差异, 但对 256 个通道数据进行平均后 `xcorr` 的差异基本不大, 所以互相关系数分析相似度并不是一个特别好的指标。

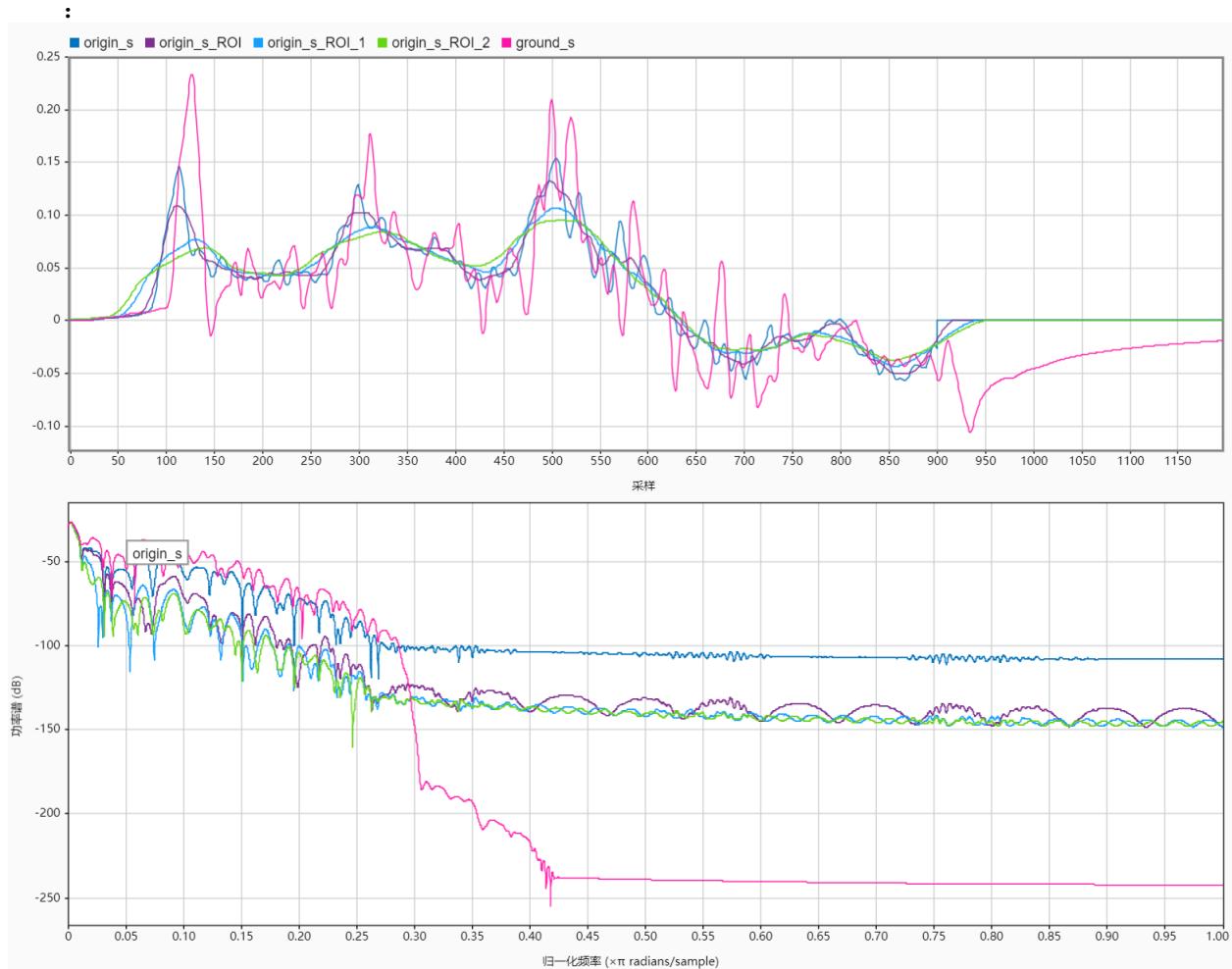
6.5 滤波降噪分析

6.5.1 高斯滤波



通过高斯滤波，信号曲线变得更加平滑，在频谱信息上反映出的是所有频率信息强度都有一定的降低，高频信息衰弱得格外强烈，所以高斯滤波实际上相当于一个全带加权得滤波器。

6.5.2 移动平均

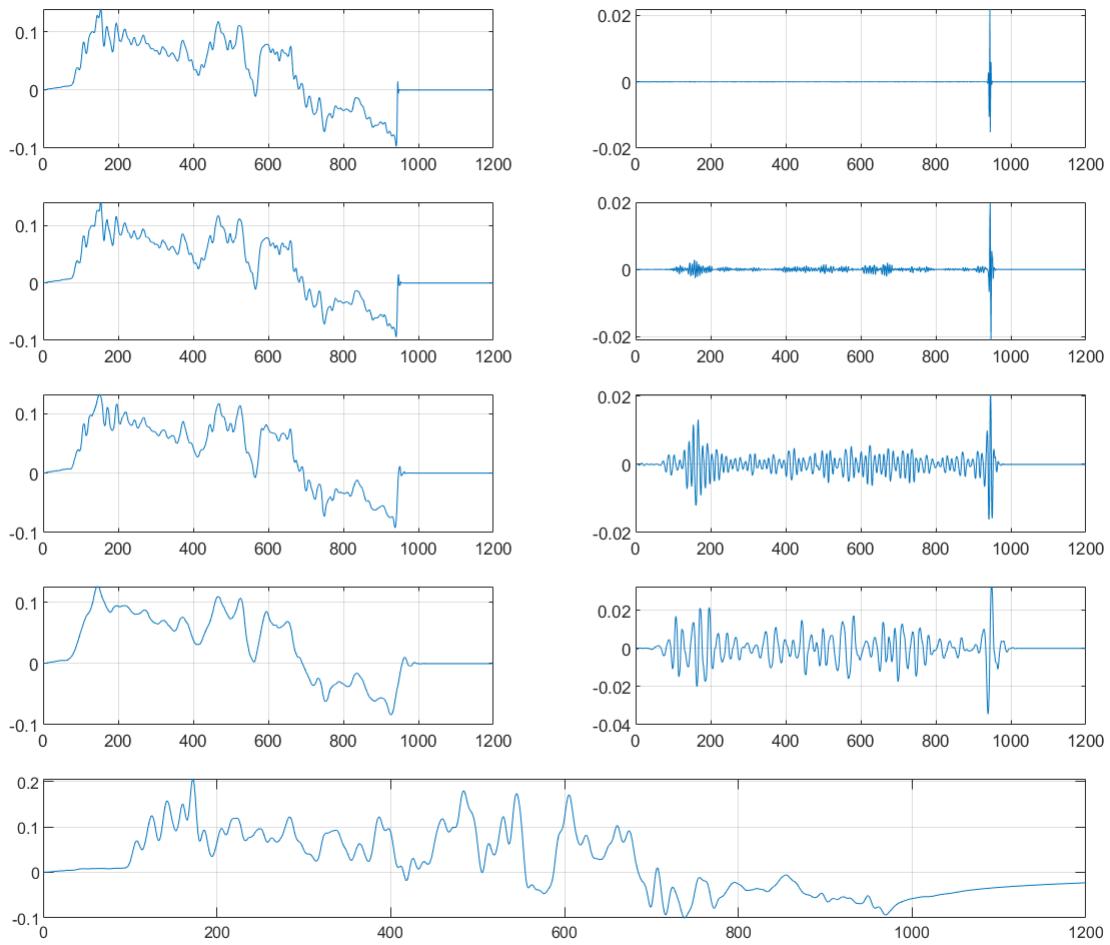


移动平均平滑方法对也相当于一个滤波器，其对低频信息的权重比高斯平滑更大，通过和ground信号的频谱对比可以发现归一化频率0.3以上有明显的降低，意味着ground参考信号的高频信号成份很少。

通过两种滤波方式可以看出，信号虽然变得更加光滑但是很多原有信息也因为平均被消除了，信号的细节消失，反而降低了信号的分辨程度，所以通常情况下平滑滤波仅对频率不依赖的噪声信号叠加有比较好的消除效果。对于特定的，频率特征未知的噪声，滤波反而会消除部分信号信息，降低信噪比。并且平滑因子越大，信号损失得越严重。

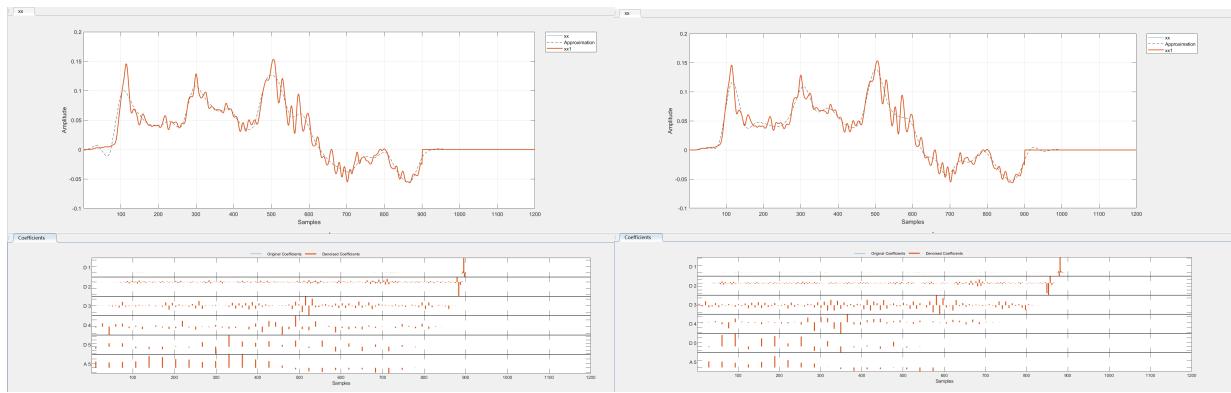
6.6 小波变换降噪分析

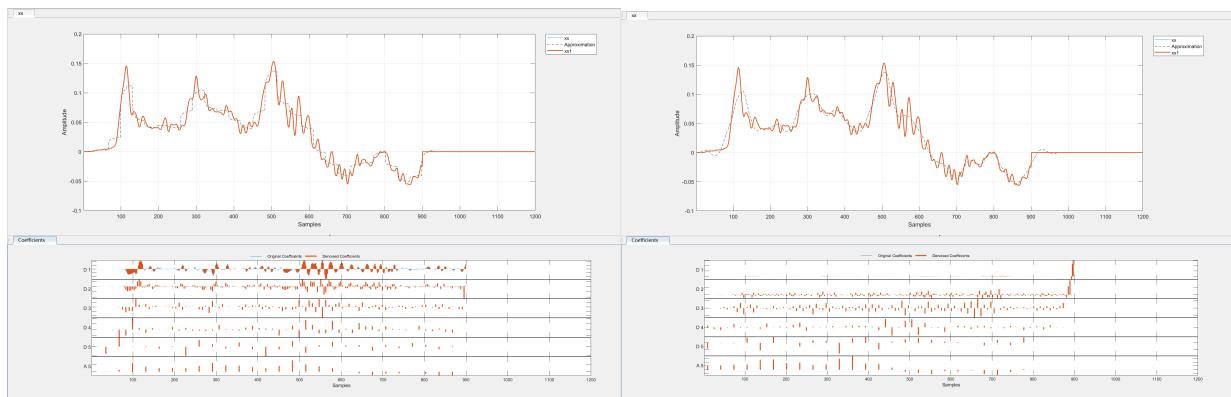
利用MATLAB的wavedec函数和wrcoef函数，对brain1X通道1的信号进行db5小波变换：



上图左四是小波函数db5下的尺度1到4阶的近似信号，右四是1到4阶的细节信号，最下方是brain1X的ground truth信号，阶数越高近似信号越平滑，和滤波存在同样的问题，一些信号细节被损失。

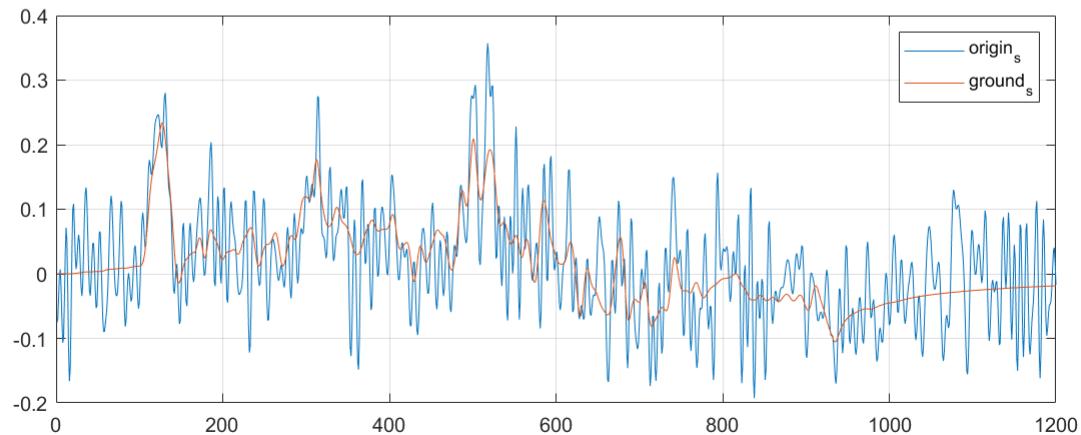
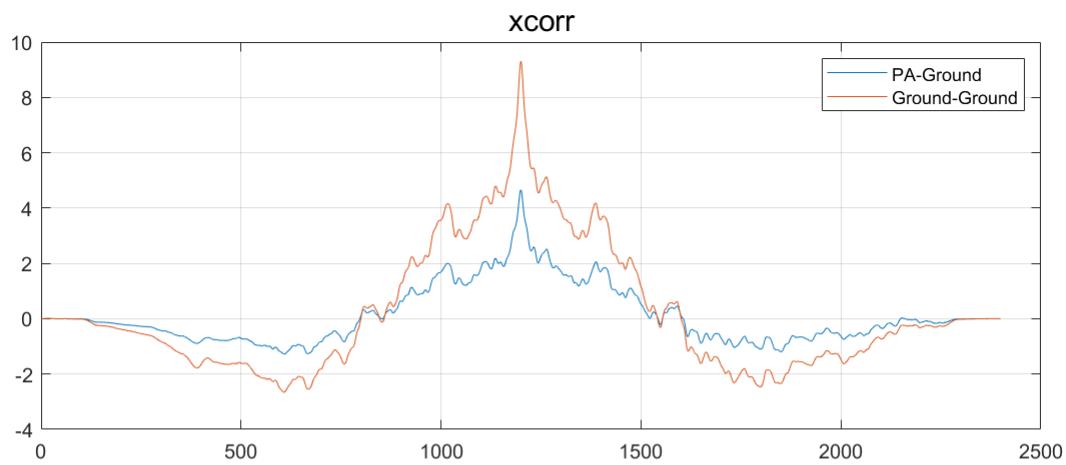
通过MATLAB的Wavelet Signal Denoiser用不同的小波函数进行降噪，都无法同时满足降噪和保真：





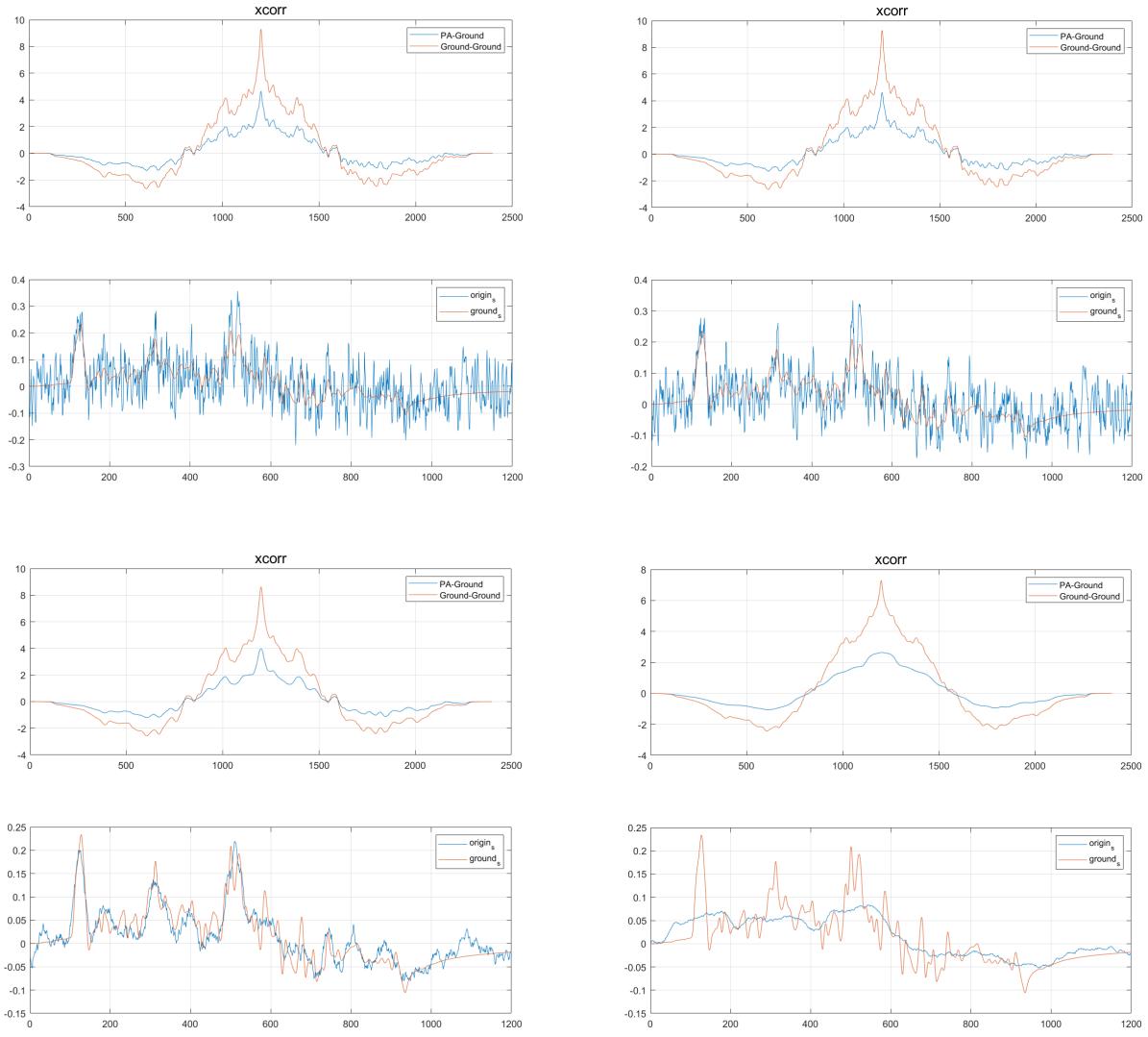
滤波法和小波变换法对于PA信号都无法获得一个较好的恢复效果，为了进一步分析两种方式降噪后与ground truth信号的相似度，可以利用rand函数对ground truth信号引入幅值为 ± 0.2 的随机噪声进行分析。此处选用brain1Y进行分析：

低通(截止频率：归一化0.3，坡度：0.9999，阻带衰减：60dB)：



单通道SSE: 4.6245, 相关系数: 0.6984

移动平均(平滑因子: 0.2, 0.4, 0.6, 0.8)：

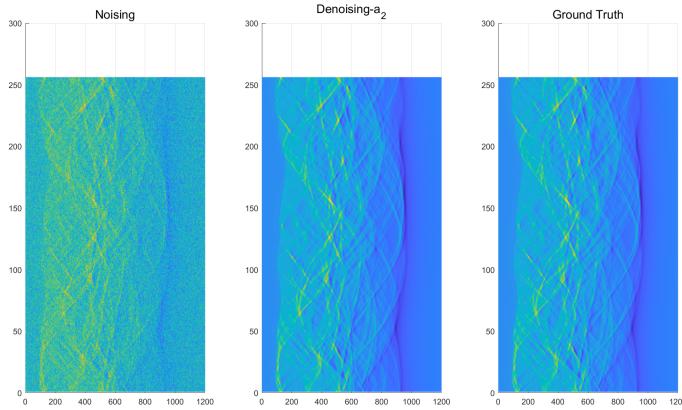


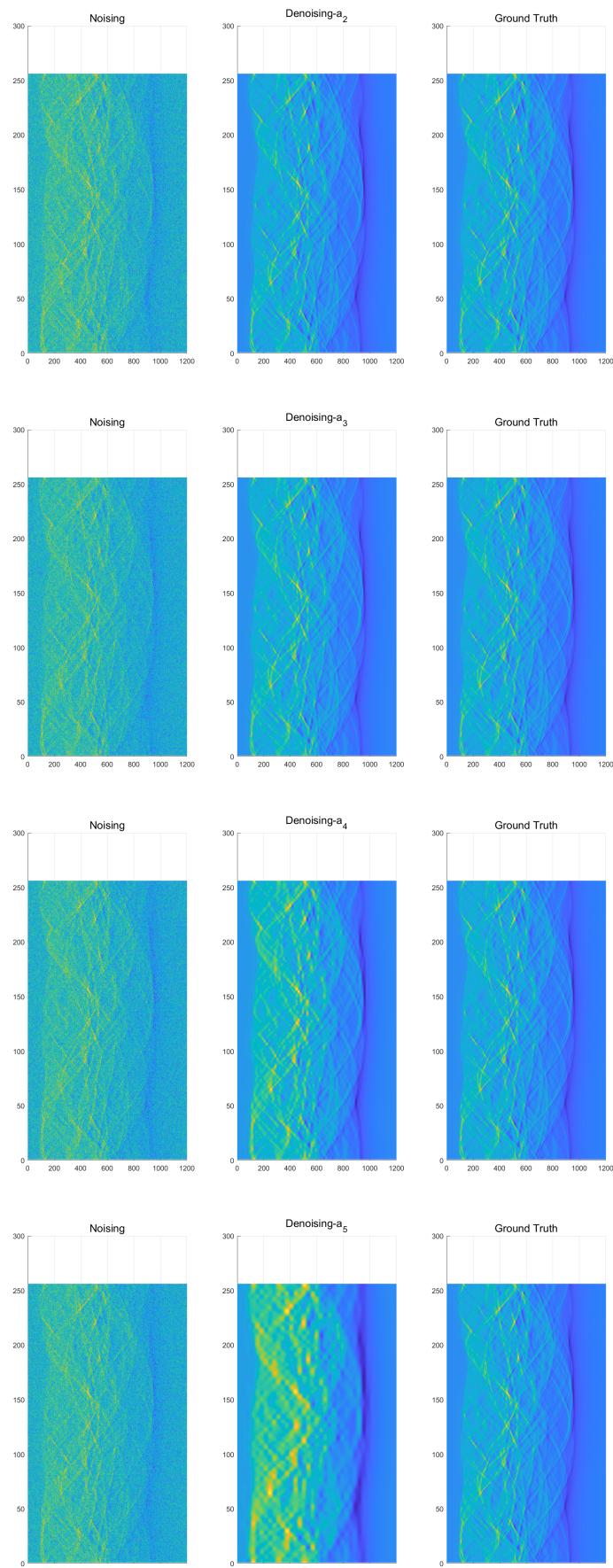
单通道SSE分别为: 5.0558, 3.7605, 0.8514, 1.6602

相关系数分别为: 0.6812, 0.7328, 0.8999, 0.8019

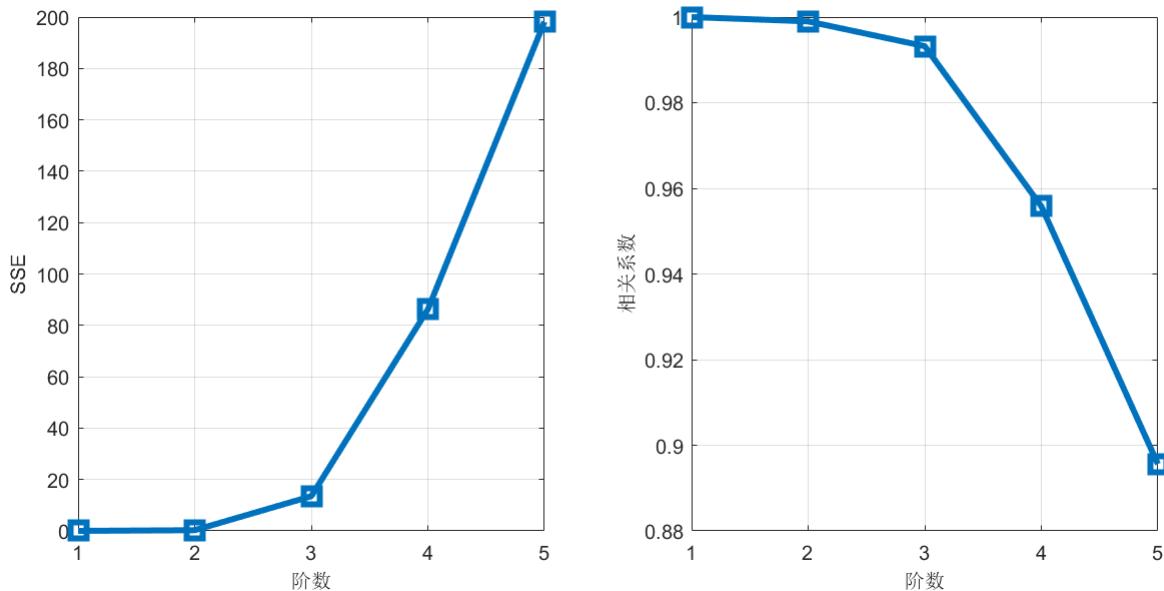
随着平滑因子增加, SSE和相关系数都表现出先降低后增加的趋势, 和之前利用brainX数据进行分析相似, 当平滑程度过高会极大影响信号的保真度。

利用db5小波进行信号降噪恢复:





随着近似信号的阶数增加，SSE和相关系数的变化关系如下图：



在阶数为1的时候降噪效果最好，随着阶数增加，SSE呈现出指数上升的趋势，同时相关系数呈现出快速降低的趋势。

综上可以发现对于信号降噪，db5小波变换整体要好于移动平均滤波以及低通滤波，相关系数均在0.85以上，而低通滤波和平滑滤波的效果则具体取决于参数选取，相关系数在0.6-0.8范围，在平滑因子选择适当的情况下可以达到0.9左右的相关性。

7 参考文献

[1]

[2]

8 附件

8.1 MATLAB code

8.1.1 相似性分析

```

1 SSE = 0;
2 CORR = 0;
3 XCOR = zeros(1,2399);
4 COR = zeros(1,2399);
5 for i =1:256
6 origin_s = X(i,:);
7 ground_s = Y(i,:);
8 SSE = SSE + sse(origin_s ,ground_s );
9 CORR = CORR + corr(origin_s ,ground_s );
10 XCOR = XCOR + xcor(origin_s ,ground_s );
11 COR = XCOR + xcor(ground_s ,ground_s );
12 end
13
14 plot(XCOR)
15 hold on
16 plot(COR)
17 title('xcorr','FontSize',15)
18 legend('PA-Ground','Ground-Ground')

```

```

19 grid on

21 function SSE = sse(u,g)
22 if length(u) == length(g)
23 SSE = (u-g)*(u-g)';
24 end
25 end

27 function CORR = corr(u,g)
28 if length(u) == length(g)
29 m = corrcoef(u,g);
30 CORR = m(1,2);
31 end
32 end

33 function XCOR = xcor(u,g)
34 if length(u) == length(g)
35 XCOR = xcorr(u,g);
36 end
37 end

```

8.1.2 小波变换

```

new_x = zeros(256,1200);
2 construct = Y+(rand(256,1200)-1)*2/5;
for i=1:256
y = Y(i,:);
[c,l] = wavedec(y,5,'db5');
a2 = wrcoef('a',c,l,'db5',3);
new_x(i,:) = a2;
end
subplot(1,3,1)
mesh(construct)
title('Noising','FontSize',15)
12 subplot(1,3,2)
mesh(new_x)
title('Denoising-a_3','FontSize',15)
14 subplot(1,3,3)
mesh(Y)
title('Ground Truth','FontSize',15)
16
18 a = [8.12*10^(-4) 0.2712 13.4950 86.2393 198.1685];
20 b = [1.000 0.999 0.9932 0.9558 0.8956];
22 subplot(1,2,1)
plot(a,'s-','MarkerSize',10,'linewidth',3)
xlabel('jieshu','FontSize',10)
24 ylabel('SSE','FontSize',10)
grid on
26 subplot(1,2,2)
plot(b,'s-','MarkerSize',10,'linewidth',3)
xlabel('jieshu','FontSize',10)
ylabel('corrcoef','FontSize',10)
28 grid on
30

```

```

y = Y(1,:)+rand(1,1200)/5;
2 y = X(1,:);

```

```

[ c , l ] = wavedec(y,5,'db5') ;
4
d5 = wrcoef('d',c,l,'db5',5) ;
d4 = wrcoef('d',c,l,'db5',4) ;
d3 = wrcoef('d',c,l,'db5',3) ;
8 d2 = wrcoef('d',c,l,'db5',2) ;
d1 = wrcoef('d',c,l,'db5',1) ;
10
a5 = wrcoef('a',c,l,'db5',5) ;
12 a4 = wrcoef('a',c,l,'db5',4) ;
a3 = wrcoef('a',c,l,'db5',3) ;
14 a2 = wrcoef('a',c,l,'db5',2) ;
a1 = wrcoef('a',c,l,'db5',1) ;
16 figure
    subplot(5,2,1);
18 plot(a1)
    grid on
20 subplot(5,2,2);
    plot(d1)
22 grid on
    subplot(5,2,3);
    plot(a2)
24 grid on
26 subplot(5,2,4);
    plot(d2)
28 grid on
    subplot(5,2,5);
30 plot(a3)
    grid on
32 subplot(5,2,6);
    plot(d3)
34 grid on
    subplot(5,2,7);
36 plot(a4)
    grid on
38 subplot(5,2,8);
    plot(d4)
40 grid on
    subplot(5,2,[9,10]);
42 plot(Y(1,:))
    grid on

```