**电子科技大学**生命科学与技术**学院**

**标 准 实 验 报 告**

**（实验）课程名称** 生物医学信号处理

**2018-2019-第2学期**

**电子科技大学教务处制表**

|  |  |  |  |
| --- | --- | --- | --- |
| **学生姓名** |  | **学 号** |  |
| **指导教师** | 李凌 | **实验时间** | **2019.4.30** |

**一、实验室名称：** 品学楼B 302

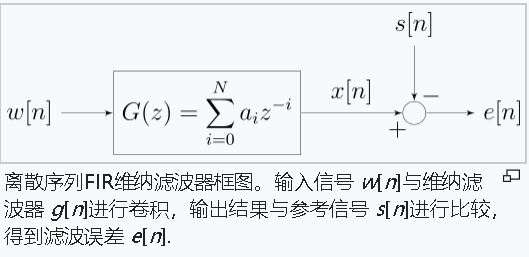
**二、实验名称：**维纳滤波器的实现和应用

**三、实验学时：2**

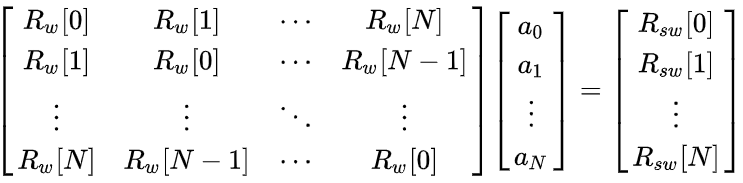
**四、实验原理：**

1.两个随机变量的四阶标准矩可以定义为：其中μ4是四阶中心矩，σ是标准差。在统计学中，峰度（Kurtosis）衡量实数随机变量概率分布的峰态。峰度高就意味着方差增大是由低频度的大于或小于平均值的极端差值引起的。

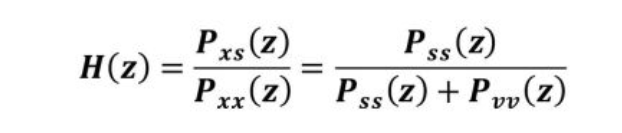
2. 维纳滤波器（Wiener filter）是由数学家维纳（Norbert Wiener）提出的一种以最小平方为最优准则的线性滤波器。在一定的约束条件下，其输出与一给定函数（通常称为期望输出）的差的平方达到最小，通过数学运算最终可变为一个托布利兹方程的求解问题。维纳滤波器又被称为最小二乘滤波器或最小平方滤波器，目前是基本的滤波方法之一。



因果的有限冲激响应（FIR）维纳滤波器通过使用输入和输出信号的统计信息，而不是使用一些给定的数据矩阵X和输出矢量Y，来发现最佳的抽头权重。它将输入信号的自相关的估计（T）代入输入矩阵X，将输出和输入信号的互相关估计（V）代入的输出向量Y。可以通过求解矩阵方程：

的解得到。

而因果解可以利用功率谱计算。



1. **实验目的：**通过上机加深对维纳滤波器的了解
2. **实验内容：（详细填写）**

1. 熟悉心电信号

2. 非因果维纳滤波器的实现

3. 维纳滤波器FIR方法的实现

4. 利用维纳滤波算法对语音信号去噪

**七、实验器材（设备、元器件）： matlab**

**八、实验步骤**

**（一）**熟悉心电信号

1、数据文件的名字是mecg1.dat, fecg1.dat和 noise1.dat，我们可用通过load命令来读取（读取的时候别忘了加.dat）。每个文件包含一个电极记录的数据，采样率是256 Hz。。

2、对这三个数据进行叠加处理，我们就能得到临床上记录到的真实的心电信号

3、计算并画出mECG，fECG和noise的功率谱。（用matlab函数pwelch）。比较fECG和mECG在频率域上的差异和相同之处，用一两句话描述出来

4、 用matlab函数hist来画每个源信号的概率密度分布（PDF），把这些图写在你的报告中。用matlab函数kurtosis来计算信号的四阶统计量（信号的峰度k），记录这些参数。描述k值怎样反映信号的高斯性？把k值和PDF联系起来

（二）非因果维纳滤波器的实现

1、完成wienerFilter函数

2、编写程序调用以上函数，利用所给心电真实信号、噪声、观察信号（假设fecg是我们的期望信号，mecg+noise都是噪声），然后把观察信号通过滤波器得到的输出信号与原来的真实信号进行比较（相关系数作为衡量指标），观察是否达到了去噪目的。

3、调节信噪比，得到横坐标为信噪比，纵坐标为相关系数的图，回答维纳滤波器适用于多大信噪比的去噪问题？

（三）维纳滤波器FIR方法的实现

1、编写函数解W-H方程，寻找最优滤波器（FIR方法）。

2、编写程序调用以上函数，利用所给心电真实信号、噪声、观察信号（可以是mecg+noise），然后把观察信号通过滤波器得到的输出信号与原来的真实信号进行比较（相关系数作为衡量指标），观察是否达到了去噪目的。

3、调节信噪比，得到横坐标为信噪比，纵坐标为相关系数的图，回答维纳滤波器适用于多大信噪比的去噪问题？

4、第一种方法和第二种方法的比较，哪种方法更适合于心电去噪？（四）利用维纳滤波算法对语音信号去噪

1、自行采用手机录制声音，[y,fs,Nbits]=wavread(‘\*.wav');

导入到matlab程序中，并在figure框中画出该波形；

2、 添加高斯噪声；

3、利用维纳滤波对其去噪；

4、调整信噪比，观察信噪比与提取效果的关系。

1. **实验数据及结果分析：（详细填写）（包括程序、图、结果等）**
2. 熟悉心电信号

clear;

fecg1=load("F:\zjk\生物医学信号处理\上机4\data\fecg1.dat");

mecg1=load("F:\zjk\生物医学信号处理\上机4\data\mecg1.dat");

noise1=load("F:\zjk\生物医学信号处理\上机4\data\noise1.dat");

s1 = fecg1+mecg1+noise1;

fecg2=load("F:\zjk\生物医学信号处理\上机4\data\fecg2.dat");

mecg2=load("F:\zjk\生物医学信号处理\上机4\data\mecg2.dat");

noise2=load("F:\zjk\生物医学信号处理\上机4\data\noise2.dat");

s2 = fecg2+mecg2+noise2;

fs = 256;

N = 2560;

n=1:2560;

fs = 256;

figure(1)

subplot(4,1,1)

plot(n/fs,fecg2)

title("fecg1信号")

ylabel('微伏');xlabel('时间/s')

subplot(4,1,2)

plot(n/fs,mecg2)

title("mecg1信号")

ylabel('微伏');xlabel('时间/s')

subplot(4,1,3)

plot(n/fs,noise2)

title("noise1信号")

ylabel('微伏');xlabel('时间/s')

subplot(4,1,4)

plot(n/fs,s1)

title("真实的心电信号")

ylabel('微伏');xlabel('时间/s')

%加窗并滤波

N1 = 32; % Order

Fc1 = 11; % First Cutoff Frequency

Fc2 = 50; % Second Cutoff Frequency

flag = 'scale';

win = hamming(N1+1);

b = fir1(N1, [Fc1 Fc2]/(fs/2), 'bandpass', win, flag);

fecg1 = filtfilt(b,1,fecg1);

mecg1 = filtfilt(b,1,mecg1);

noise1 = filtfilt(b,1,noise1);

s1 = filtfilt(b,1,s1);

figure(2)

subplot(4,1,1)

plot(n/fs,fecg1)

title("四个信号加窗滤波后")

ylabel('微伏');xlabel('时间/s')

subplot(4,1,2)

plot(n/fs,mecg1)

ylabel('微伏');xlabel('时间/s')

subplot(4,1,3)

plot(n/fs,noise1)

ylabel('微伏');xlabel('时间/s')

subplot(4,1,4)

plot(n/fs,s1)

ylabel('微伏');xlabel('时间/s')

ecg = fecg1;

ecg = diff(ecg).^2;

N2 = 30;

temp2 = [];

r\_wave1 = [];

for i=1:length(ecg)-N2

temp2 = [temp2;trapz(ecg(i:i+N2-1))];

end

ecg = temp2;

Threshold = (max(ecg) - min(ecg))\*0.4 + min(ecg);

[r\_wave1(:,1),r\_wave1(:,2)]=findpeaks(ecg,256,'MinPeakDistance',0.10,'MinPeakHeight',Threshold);

heartrate = length(r\_wave1)\*60\*fs/N

t = (1:length(ecg))./fs;

figure(3)

subplot(2,1,1)

plot(t,ecg);

hold on;

subplot(2,1,1)

plot(r\_wave1(:,2),r\_wave1(:,1),'\*')

xlabel('时间/s');title('胎儿积分后心电信号')

ecg = mecg1;

ecg = diff(ecg).^2;

N2 = 30;

temp2 = [];

r\_wave1 = [];

for i=1:length(ecg)-N2

temp2 = [temp2;trapz(ecg(i:i+N2-1))];

end

ecg = temp2;

Threshold = (max(ecg) - min(ecg))\*0.4 + min(ecg);

[r\_wave1(:,1),r\_wave1(:,2)]=findpeaks(ecg,256,'MinPeakDistance',0.10,'MinPeakHeight',Threshold);

heartrate = length(r\_wave1)\*60\*fs/N

t = (1:length(ecg))./fs;

figure(3)

subplot(2,1,2)

plot(t,ecg);

hold on;

subplot(2,1,2)

plot(r\_wave1(:,2),r\_wave1(:,1),'\*')

xlabel('时间/s');title('母亲积分后心电信号')



图1.l 母亲、胎儿心电及噪声信号



图1.2 母亲、胎儿心电QRS波群识别

clear;

fs = 256;

load("F:\zjk\生物医学信号处理\上机4\data\fecg1.dat");

load("F:\zjk\生物医学信号处理\上机4\data\mecg1.dat");

load("F:\zjk\生物医学信号处理\上机4\data\noise1.dat");

load("F:\zjk\生物医学信号处理\上机4\data\fecg2.dat");

load("F:\zjk\生物医学信号处理\上机4\data\noise2.dat");

load("F:\zjk\生物医学信号处理\上机4\data\mecg2.dat");

s1 = fecg1 + noise1;

figure(1)

subplot(3,1,1)

pwelch(fecg1,hamming(512),256,[],fs)

title('胎儿功率谱');

subplot(3,1,2)

pwelch(mecg1,hamming(512),256,[],fs)

title('母亲功率谱')

subplot(3,1,3)

pwelch(noise1,hamming(512),256,[],fs)

title('噪声功率谱')

m\_fecg1 = mean(fecg1)

m\_mecg1 = mean(mecg1)

m\_noise1 = mean(noise1)

v\_fecg1 = var(fecg1)

v\_mecg1 = var(mecg1)

v\_noise1 = var(noise1)

k\_fecg1 = kurtosis(fecg1)

k\_mecg1 = kurtosis(mecg1)

k\_noise1 = kurtosis(noise1)

figure(2)

subplot(2,1,1)

hist(fecg1,100)

title('fecg1')

subplot(2,1,2)

hist(mecg1,100)

title('mecg1')

x = fecg1+mecg1+noise1;

[yhat, H] = wienerFilter(fecg1,x,1,fs);



图1.3 功率谱

图1.3 不同信噪比的信号（**x=-s+weight\*n**）



图1.4 概率密度分布

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mean | Var | Kurtosis |
| Fecg1 | -4.2500e-10 | 1.0000 | 8.9901 |
| Mecg1 | -2.4662e-10 | 1.0000 | 14.0421 |
| Noise1 | -4.7691e-10 | 1.0000 | 2.7662 |

采用阈值法得出的心率为：胎儿156（次/分钟）母亲60（次/分钟）。从图中可以看胎儿的功率谱在20-80hz处要大于母亲的功率，母亲心电功率在30hz已经衰减到-40dB以下，而胎儿功率在70hz衰减到-40dB，胎儿的较高频段信号能量高于母亲，从时域图中也可以看出胎儿心率高于母亲。而噪声信号功率谱较为均匀，20hz以下的低频段稍高，其余个频段分布较均匀。三个信号的均值都接近零，方差为1（可能之前已经进行了归一化处理，只有fecg2的方差为0.64，其他信号方差均为1）。三个信号的概率分布如图1.4，噪声信号noise1近似于均值为零的正态分布。Noise1信号的峰度为2.7，接近于3（正态分布的峰度值），胎儿信号fecg1的峰度小于母亲信号，从概率分布也能看出，mecg1信号相比于fecg1信号，分布更加陡峭（偏离正态分布程度大）。

1. 非因果维纳滤波器的实现

winerFilter函数如下：

function [yhat, H] = wienerFilter(desired,observation,graphicsFlagOn,Fs);

%WIENERFILTER Apply Wiener filter to input data (template)

% [YHAT, H] = WIENERFILTER(DESIRED\_DATA,OBSERVED\_DATA) creates a non-causal

% Wiener filter from input signals DESIRED\_DATA (the signal)and OBSERVED\_DATA

% (signal plus noise), and applies this filter to the OBSERVED\_DATA input.

% Output YHAT is the Wiener filtered output signal, and H is the Wiener

% filter's frequency response.

%

% This function automatically generates a plot of the results by calling

% function plotWiener.m. To prevent graphical output, pass a third input

% argument equal to zero. For plotting, a sample frequency of 256 Hz is

% assumed. You may pass a fourth input argument FS to change this:

% WIENERFILTER(DESIRED\_DATA,OBSERVED\_DATA,0,FS)

%

% You must update this function to make it fully operational.

%

% IMPORTANT CONCEPTUAL NOTE: Designing a Wiener filter requires knowledge of the

% POWER SPECTRUM of the desired signal and the POWER SPECTRUM of the noise signal.

% It does not require the time-domain signals themselves; if we had those signals,

% there would be no need for the Wiener filter. This m-file estimates the power

% spectra, creates and applies the Wiener filter, and plots the result.

% G.D. Clifford 2004 gari AT mit DOT edu

% Licensing: GNU GPL applies

% Documentation updated 5/06/06, Eric Weiss

% Documentation updated 5/14/15, Rachelle Horwitz-Martin

% Documentation updated 4/11/16, Christopher Dean

% Input argument checking

if nargin < 4

Fs = 256;

end

Fs = round(Fs);

if nargin < 3

graphicsFlagOn = 1;

end

% Estimate noise from desired signal

%-----------------------------------

% The noise model is the difference between the observation and the desired

% signal

% \*\*\*\* complete this

noise = observation - desired;

% Wiener filter

%--------------

Sf2 = pwelch(desired,[],[],[],[],'twosided');

Nf2 = pwelch(noise,[],[],[],[],'twosided');

% Create frequency response of Wiener filter

%-------------------------------------------

% \*\*\*\* complete this

H = Sf2 ./(Sf2 + Nf2);

% Compute impulse response (filter coefficients) from

% frequency response of Wiener filter

%----------------------------------------------------

% \*\*\* complete this

h = ifft(H);

% Apply impulse response of Wiener filter to observation using fftfilt

%---------------------------------------------------------------------

% \*\*\* complete this

yhat = fftfilt(h,observation);

% Plot the data?

%---------------

if graphicsFlagOn==1

plotWiener(desired, noise, observation, yhat, Fs);

end;

plot函数：

function plotWiener(model, noise, observation, utilda, Fs, nwindow)

if nargin < 5

error('Improper function call to plotWiener.m');

end;

if nargin < 6

nwindow = 512;

end;

[Psig, Fsig] = pwelch(observation,nwindow,[],[],Fs);

[Pm, Fm] = pwelch(model,nwindow,[],[],Fs);

[Pn, Fn] = pwelch(noise,nwindow,[],[],Fs);

[Pw, Fw] = pwelch(utilda,nwindow,[],[],Fs);

% Plot time-domain data

%----------------------

figure;

t=[1/Fs:1/Fs:length(observation)/Fs];

subplot(3,1,1); plot(t,observation); ylabel('观察信号');

title('Wiener Filter');

subplot(3,1,2);plot(t,model);

ylabel('真实信号');

subplot(3,1,3);plot(t,utilda);

ylabel('输出信号');

xlabel('time (s)');

if 0

figure; plot(t,observation);hold on;plot(t,utilda,'r');

ylabel('Comparison: Observation vs Wiener Filtered Signal');

xlabel('time (s)');

end;

测试脚本：

clear;

fs = 256;

load("F:\zjk\生物医学信号处理\上机4\data\fecg1.dat");

load("F:\zjk\生物医学信号处理\上机4\data\mecg1.dat");

load("F:\zjk\生物医学信号处理\上机4\data\noise1.dat");

n1 = mecg1 + noise1;

snr = [20,10,5,-5,-10];

for i = 1:length(snr)

w = sqrt(var(fecg1)./(var(n1).\*(10.^(snr(i)./10))));

x = fecg1+w\*n1;

[yhat, H] = wienerFilter(fecg1,x,1,fs);

corr\_sx(i) = corr(fecg1,yhat);

end

figure(20)

plot(snr,corr\_sx)

title('相关系数与信噪比的关系图')

ylabel('相关系数');xlabel('信噪比')



图2.1 维纳滤波



图2.2 信噪比-10



图2.3 信噪比10



图2.4

改变信噪比，可以看到相似程度随信噪比降低而减小，信噪比为-10时已经难以看出真实信号。从图中可已看出，当信噪比大于10，能较好地观测到输出信号中的有用信号。

1. 维纳滤波器FIR方法的实现

函数如下

function [hopt,error]=wein(x,s)

%%%FIR 方法实现weinner滤波

M = 2;

error(M)=1;error(M-1)=0;

rx = xcorr(x,x,'biased');

if abs(error(M)-error(M-1))>0.0001

M=M+1;

end

rxx = rx(length(x):end);

Rxx = toeplitz(rxx);

rxs = xcorr(x,s,'biased');

Rxs = rxs(length(x):length(x)+N-1);

hopt = inv(Rxx)\*Rxs;

error(M) = Rxs(1)-Rxs'\*hopt;

测试脚本如下：

clear;

fs = 256;

load("F:\zjk\生物医学信号处理\上机4\data\fecg1.dat");

load("F:\zjk\生物医学信号处理\上机4\data\mecg1.dat");

load("F:\zjk\生物医学信号处理\上机4\data\noise1.dat");

N = length(fecg1);

n=1:N;

n1 = mecg1 + noise1;

w = 1;

x = fecg1+w\*n1;

[h, err] = wein(x,fecg1,N);

xx=filter(h,1,x);

subplot(3,1,1)

plot(n/fs,x)

title('观察信号');ylabel('微伏');xlabel('时间/s')

subplot(3,1,2)

plot(n/fs,fecg1)

title('真实信号');ylabel('微伏');xlabel('时间/s')

subplot(3,1,3)

plot(n/fs,xx)

title('滤波后信号');ylabel('微伏');xlabel('时间/s')

figure(3)

x1 = [];s= fecg1;xn=n1;

snr = [10 5 0 -5 -10];

w = sqrt(var(s)./(var(n1).\*(10.^(snr./10))));

[h, err] = wein(x,fecg1,N);

xx=filter(h,1,x);

subplot(3,2,1)

plot(fecg1)

title('未加噪声的原始信号')

for i = 1:length(snr)

x1(:,i) = -s + w(i)\*xn;

corr\_sx(i) = corr(s,x1(:,i));

end

for i = 1:length(snr)

str = ['信噪比',num2str(snr(i)),'dB'];

subplot(3,2,i+1)

plot(x1(:,i))

title(str)

end

figure(4)

plot(snr,corr\_sx)

title('相关系数与信噪比的关系图')

ylabel('相关系数');xlabel('信噪比')



图3.1FIR实现的维纳滤波器滤波效果



图3.2



图3.3 信噪比

利用相关系数，我们可以看到，信噪比相同时，FIR方法实现维纳滤波的相关程度更低。因此，非因果维纳滤波器的效果相对更好。

1. 利用维纳滤波算法对语音信号去噪

figure(1)

n=50000:52000;

plot(n/fs,data(n,1))

title('音频文件')

xlabel('时间/s');ylabel('幅度');

xn = randn(1,2001)';

snr = [20 10 5 -5 -10 -20];

w = sqrt(var(data(n,1))./(var(xn).\*(10.^(snr./10))));

for i = 1:length(snr)

x = w(i)\*xn +data(n,1);

[yhat, H] = wienerFilter(data(n,1),x,1,fs);

end



图4.1 音频信号



图4.2 截取的一小段音频信号



图4.3 信噪比20



图4.4 信噪比5



图4.4 信噪比-10



图4.4 信噪比-20

从图中可以看到，当信噪比大于5时滤波后波形和真实信号波形类似，信噪比低于-10时有较大的失真。但是信噪比大于5时实际听到的滤波后声音有较明显的噪音和失真，这可能是由于维纳滤波器的相位线性度差，而人耳对相位敏感，相位的改变导致了信号听起来有一定的变化。

1. **总结及心得体会：**

本次实验，借助matlab的函数工具箱，读取了母亲和胎儿的心电信号并用统计手段和维纳滤波进行了分析，编写了维纳滤波函数的两种实现并比较了二者的效果。 最后用维纳滤波函数对音频信号进行了处理。只使用维纳滤波的效果并不是非常好，而预先对心电进行频域的带通滤波，可以有效抑制高频噪声和基线漂移。维纳滤波需要已知目标信号的特征，一般在滤除和目标信号频域重叠的白噪声干扰时有一定的效果，可以配合其他经典信号处理手段使用。

**十一、对本实验过程及方法、手段的改进建议：**

希望能够增加一些开放性和探究性的问题。上机时内容很多都在上机课上讲解了或在课件中列出了详细的实现方法。希望能够减少讲解，多一些自己完成的部分。

**报告评分：**

**指导教师签字：**