逢甲大學學生報告 ePaper

遞迴最小平方演算法之實現

Implementation of Recursive Least-Squares Algorithm

作者：林伯彥

系級：通訊四乙

學號：D0550129

開課老師：林育德

課程名稱：高等生醫信號處理

開課系所：自動控制工程學系碩士班

開課學年：108 學年度 第二學期

中文摘要

本篇報告以實現遞迴最小平方演算法為主，並且將其應用在生醫信號中，來濾除心電信號中之電力線干擾。在閱讀此篇報告之前，建議先閱讀[3]中的內容，先了解其原理以及如何應用在生醫信號中，而如果像要更加瞭解演算法的各種不同樣貌，可以在閱讀[1]的內容。在這篇報告中，在驗證部分使用的是 MATLAB (2018b, MathWorks®, Inc., USA)。

關鍵字：遞迴最小平方演算法、非穩態信號處理、統計信號處理、電力線干擾、消雜訊濾波，心電圖信號。

Abstract

This report implements the Recursive Least-Squares algorithm, and is applied to biomedical signal processing to suppress the power-line interference in Electrocardiogram signal. Before you read this report, I suggest to read reference [3] to understand how this algorithm can work and how to use in biomedical signal. If you wanted to know more about this algorithm, you also can read reference [1] to improve your capability. In this report, all of computer experiments were conducted in MATLAB (2018b, MathWorks®, Inc., USA).

目　次

一、 介紹

RLS(Recursive Least-Squares)基本上是從最小平方的方法(Least Squares)衍伸而來。

最小平方法是用來替代維納(wiener filter)濾波的一種方法。基本上，維納濾波器是從綜合平均(ensemble average)衍伸而來，其輸出是既有已知的輸入來獲得，也就是假設其信號是廣義穩態(wide-sense stationary)。在另一方面，最小平方法是確定性的方法，特別是使用了時間平均，也就是濾波結果使用了由之前取樣所計算的數值。最小平方其計算方式為 batch-processing approach， 因此濾波器適用於非穩態信號，藉由重複計算一塊一塊的偏壓這種計算方法雖然會比其他的演算法還要複雜，但是現在機器運算量可以非常大，因此這種方法越來越受歡迎。

而在 RLS 演算法中一個最重要的特徵就是他的收斂率會快於 LMS 演算法，因為 RLS 演算法利用反相關的計算將輸入信號白化，而此平均值為 0，然而卻也增加了其計算複雜度。

圖 1 為其方塊圖，基本上架構與 LMS 演算法極為相似，但是在演算法多了一個叫做正規化的步驟。

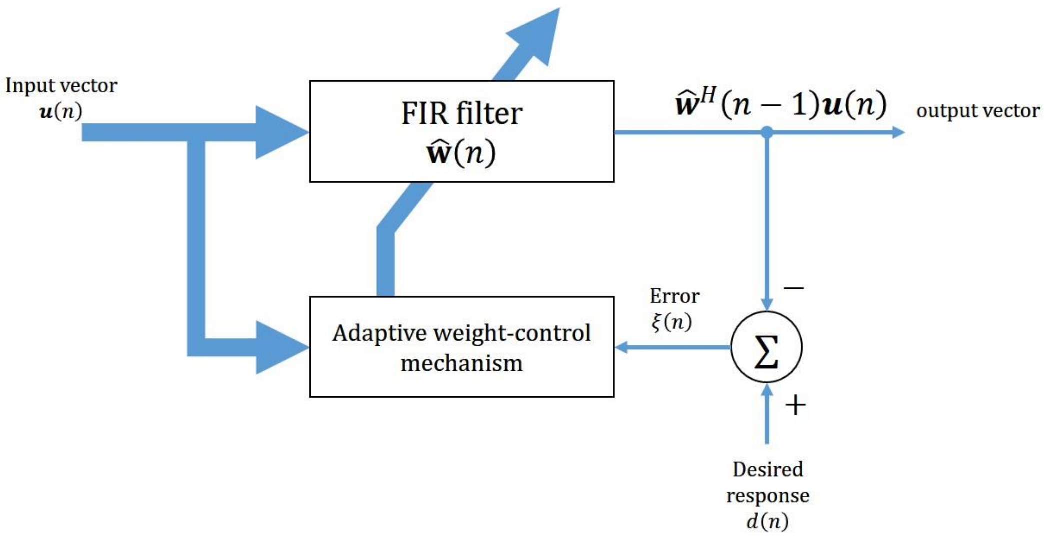

圖 1

最後，RLS 演算法可以由下列步驟來實現

(1) 初始值設定

$$\hat{\boldsymbol{w}}(0) = \mathbf{0}$$
$$\mathbf{P}(0) = \delta^{-1}\mathbf{I}$$

(2) 濾波計算

$$\boldsymbol{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\boldsymbol{u}(n)}{1+\lambda^{-1}\boldsymbol{u}^{H}\mathbf{P}(n-1)\boldsymbol{u}(n)}$$
$$\xi(n) = d(n) - \widehat{\boldsymbol{w}}^{H}(n-1)\boldsymbol{u}(n)$$
$$\widehat{\boldsymbol{w}}(n) = \widehat{\boldsymbol{w}}^{H}(n-1) + \boldsymbol{k}(n)\xi^{*}(n)$$
$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{u}^{H}(n)\mathbf{P}(n-1)$$

其中$d(n)$為時間n的觀察到的信號；$\boldsymbol{u}(n)$為$\mathrm{M}\times 1$的參考信號；$\widehat{\boldsymbol{w}}(n)$與$\widehat{\boldsymbol{w}}^{H}(n-1)$為時間在$n$的 $\mathrm{M}\times 1$以及時間在$n-1$ 的 $1\times\mathrm{M}$的權重向量；λ為遺落因子，其值介於 0 到 1 之間，而且非常接近於 1，而當$\lambda = 1$時，則為一般的 LS 法，而$(1-\lambda)^{-1}$嚴格來說，是可以用來測量延遲的演算法，所以說，當$\lambda = 1$則為無限延遲，也就是無限記憶體；而最後δ則為正規化變數(regularization parameter)，當輸入擁有大的訊雜比時(約大於 30dB)，則δ小，反之亦然。而計算中的$\xi(n)$又可稱為前置誤差(a priori error)，是由時間n的觀測信號與時間$n-1$的估測信號相減的結果，因為是估測時間n前就做相減，因此就叫做前置誤差，因此如果誤差事經過估測後再相減，則叫做後置誤差(a posteriori error)，而後置誤差的準確度會大於前置誤差，不過有些演算法兩種都會使用，例如卡爾曼濾波器(kalman filter)。

二、 程式碼

```
clc;clear;close all;
Fs = 500;M=10;
fid = fopen('D:\project\code\ECG_Signal\e3a.txt','r');
ECG= fscanf(fid, '%f')';
T=(0:length(ECG)-1)/Fs;
Ref=.2*cos(2*pi*60*T);
Xn=ECG+Ref;
ww=zeros(1,length(Xn));
W=zeros(M,1);
P=eye(M)/.01;
Lamda=1;
EE=zeros(1,length(Xn));
figure,
subplot(211),
plot(T,Xn);axis([0 10 -inf inf]);grid;
```

```
xlabel('time(sec.)'),ylabel('amplitude(mV.)');
[h,w]=FFT(Xn);
subplot(212),
plot(w/(2*pi)*Fs,(abs(h)));axis([0 250 0 .04]);grid;
xlabel('frequency(Hz)'),ylabel('amplitude(mV.)');
%% RLS
for ii=M:length(Xn)
    R=Ref(ii:-1:ii-M+1)';
    K=(Lamda^(-1)*P*R)/(Lamda^(-1)*R'*P*R+1);
    Alpha=Xn(ii)-W'*R;
    EE(ii)=Alpha;
    W=W+K*Alpha;
    P=Lamda^(-1)*(P-K*R'*P);
    ww(ii)=W'*R-Ref(ii);
end
figure,
subplot(211),
plot(T,EE);axis([0 10 -inf inf]);grid;
xlabel('time(sec.)'),ylabel('amplitude(mV.)');
[h,w]=FFT(EE);
subplot(212),
plot(w/(2*pi)*Fs,(abs(h)));axis([0 250 0 .04]);grid;
xlabel('frequency(Hz)'),ylabel('amplitude(mV.)');
%% FileClose
fclose(fid);
```

三、 結果

在此部分將會展示使用使演算法所濾波的結果，在這裡使用的例子為心電信號之電力線干擾消除，在展示結果之前，先給定以下條件:

$$d(n) = m(n) + v(n) \quad (1)$$
$$= m(n) + A_v cos(2\pi f_v n) \quad (2)$$
$$u(n) = A_u cos(2\pi f_u n)$$

其中$m(n)$為未受干擾的心電信號，$v(n)$代表的是電力線干擾信號，也就是(2)中的第二項。而在(2)中，A_v為電力線干擾振幅；f_v為電力線干擾頻率，在此為

60Hz。再來是參考信號$u(n)$，其中數值與(1)中$v(n)$相似，在此假設相等。再來是誤差式

$$\xi(n) = d(n) - \hat{\boldsymbol{w}}^H(n-1)\boldsymbol{u}(n) \qquad (3)$$

其中$\hat{\boldsymbol{w}}^H$為估測權重的共軛轉至，其中(3)中$\hat{\boldsymbol{w}}^H(n-1)\boldsymbol{u}(n)$若與$v(n)$相等，則

$$\xi(n) = m(n) + v(n) - v(n) \qquad (4)$$

$$\xi(n) = m(n)$$

因此，如果經由參考信號運算出的$\hat{\boldsymbol{w}}^H(n-1)\boldsymbol{u}(n)$越接近真實干擾信號，則相減出來的誤差會越接近真實的心電信號，還有在此假設λ為 1。

在結果中，除了直接從時域以及頻率域觀察演算法結果外，在此也引用了[2]中的 MSE 演算法來觀察濾波器階數、δ大小，以及干擾振幅大小對濾波器所造成的性能會有何變化，而[2]中 MSE 演算法定義如下:

$$MSE = \frac{\sum_{n=0}^{L-1}\left(d(n) - \hat{d}(n)\right)^2}{L} \qquad (5)$$

(5)代表在時間長度 L 中，將原始信號與經過濾波器的信號相減、平方、累加 L-1 後除上時間長度 L。如果經過(5)運算後的數值越小，代表濾波器的性能越好，反之，則越差。除了(5)，也使用了學習曲線(learning curve)來觀察濾波器的收斂速度，在此使用的不是$\xi(n)$，是$u(n) - \hat{\boldsymbol{w}}^H(n-1)\boldsymbol{u}(n)$的結果，因此跟[1]中不太一樣，不過也具有相同的趨勢。

以下為實驗結果。

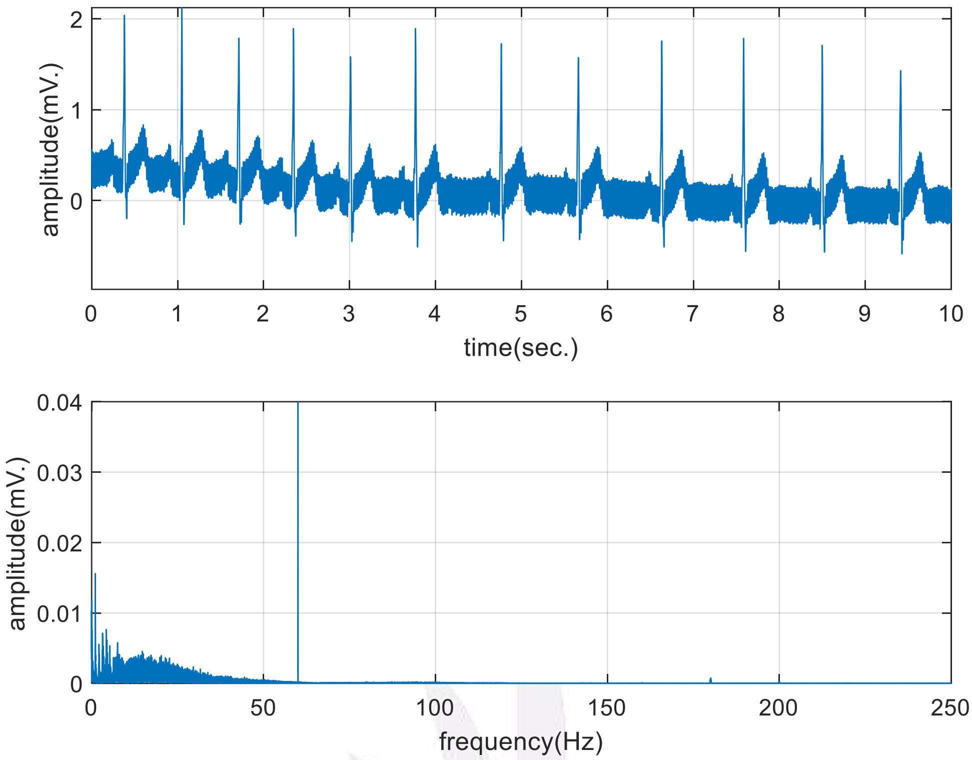

圖 2

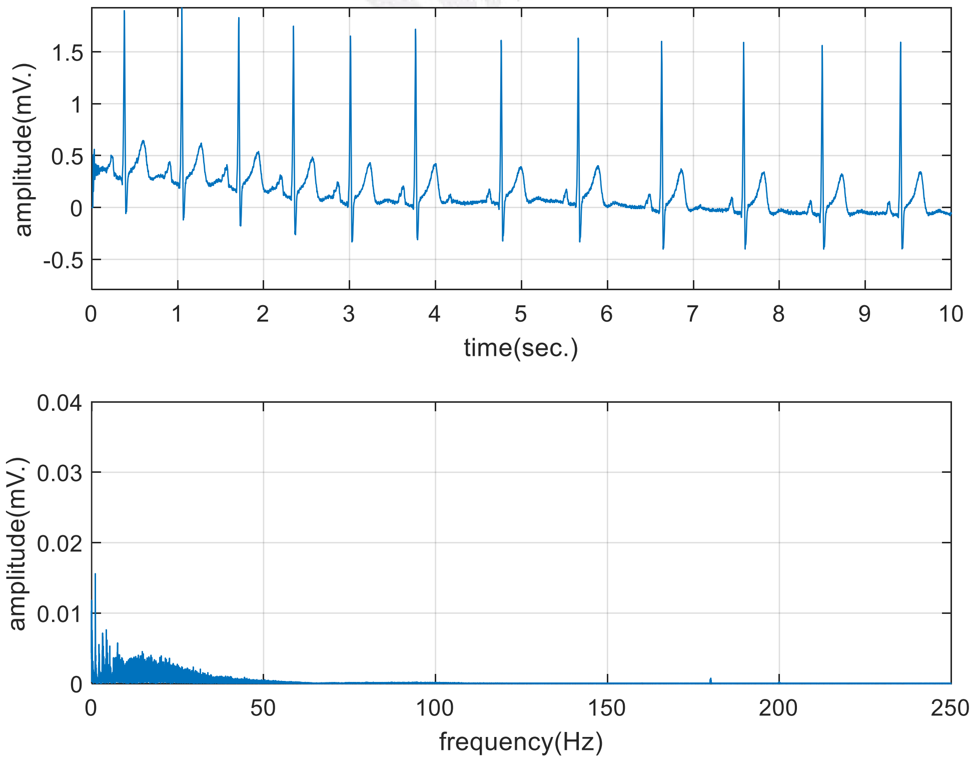

圖 3($\delta = 0.1$)

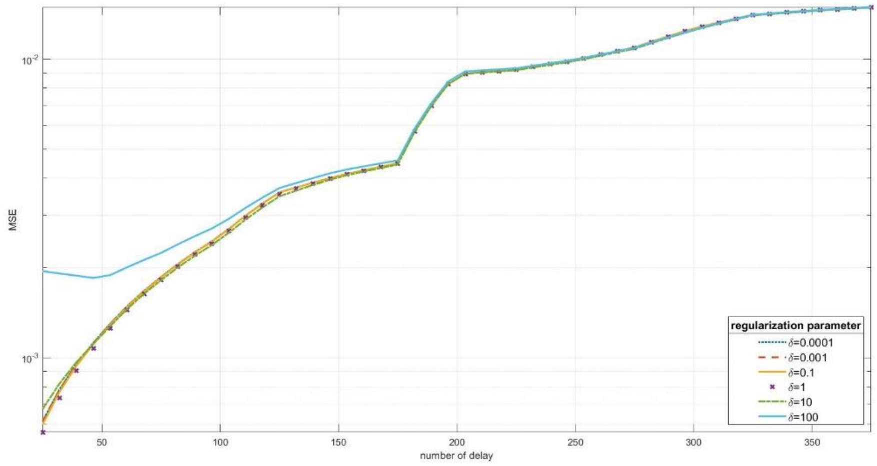

圖 4($A_v = A_u = 0.2$)

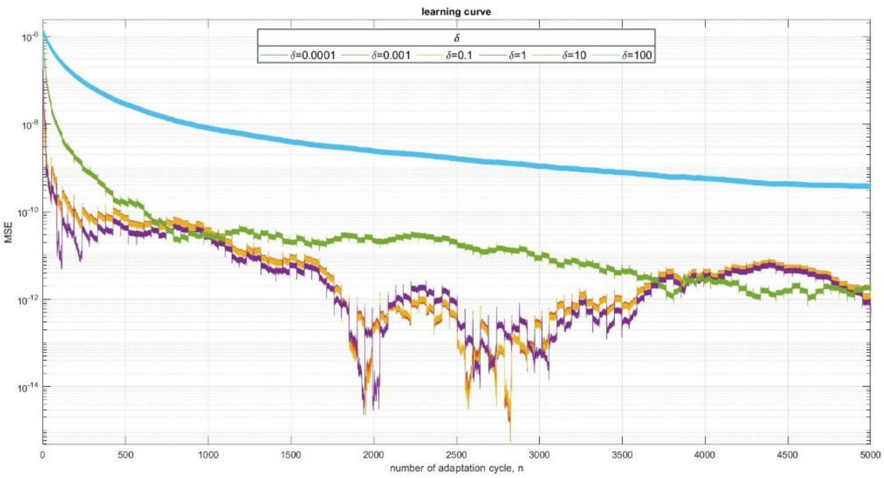

圖 5

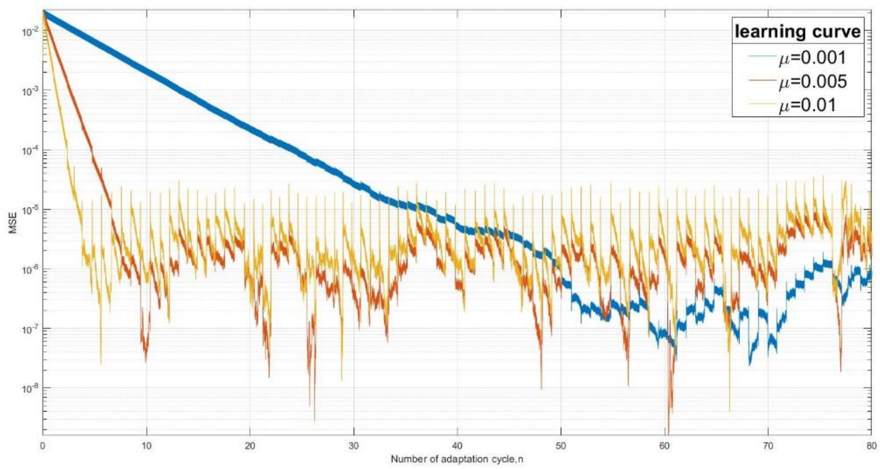

圖 6

從圖 2 及圖 3 可以看到，受干擾的心電信號經過 RLS 演算法後，成功的將干擾濾除，而且比起 LMS 快很多。而在圖 4 可以看到，在相同的延遲下，當δ越大，則其 MSE 值也越大；另外，當δ固定，當使用的延遲越多，MSE 也越大，反之亦然。圖 5 可以很明顯看到，δ越大，收斂速度越慢，跟圖 6 的 LMS 收斂速度比起來，整體速度是快於 LMS。

四、　參考資料

[1] S. Haykin, *Adaptive Filter Theory*, Pearson, 2014.

[2] K. M. Chang, “Arrhythmia ECG Noise Reduction by Ensemble Empirical Mode Decomposition”, *Sensors in Biomechanics and Biomedicine*, vol. 10, issue 6, pp. 6063-6080, 2010.

[3] R. M. Rangayyan, *Biomedical Signal Analysis,* Jon Wiley & Sons, 2015.