DSPLAB 專題(二) Wiener filter與LMS的實作及應用

1. 目的

學習wiener filter理論、Least Mean Square (LMS)理論、Room Impulse Response Generator (RIRG) 工具的使用，並在matlab上進行模擬及討論成果。

1. 實驗工具

|  |  |
| --- | --- |
| wienerscalart96.m | Wiener Filter的m檔，丟入音檔以及其他參數即可消除雜訊。 |
| audio\_SNR.m | 輸入乾淨語音與雜訊，產生混雜語音。 |
| Noise.wav\*3 | 三種不同的雜訊。(horn、mouse and white noise) |
| clean speech.wav\*10 | 十段乾淨語音 |
| echo.m | 丟入音檔以及其他參數及可模擬一個房間所產生的echo。 |
| LMS.m | 輸入音檔以及其他參數，來消除回音。 |
| 1.wav | 一段乾淨語音 |

1. 實驗步驟

實驗一

**步驟一、練習完成混雜語音的程式**

請練習依照audio\_SNR.m這支程式旁的註解完成audio\_SNR.m，使其能夠成為一支將乾淨語音與雜訊混合的程式

Hint：訊雜比(SNR):訊號與雜訊的比例

SNR (Signal-to-Noise Ratio)

**步驟二、使乾淨語音(Clean Speech)和雜訊(Noise)能依據指定的訊雜比(5)混合，產生混雜語音(Noisy Speech)**

練習使用audio\_SNR.m，加入以有的雜訊檔案例如火車聲、滑鼠聲…等，並且調整訊雜比以及設定輸出時間長短，單位為秒。

[output,fs]=audio\_SNR('Clean 檔案名稱.wav' , 'Noise 檔案名稱.wav',輸出時間(s),訊雜比);

實驗二

**步驟三、練習音檔讀取及播放**

使用指令audioread、sound將乾淨語音與混雜語音讀取及播放。

[signal,fs]=audioread(‘檔案名稱.wav’); %讀入wav檔 (請切記wav檔案必須在同層目錄下)  
sound(signal,fs); %播放WAV出來聽

Hint：將signal的能量變大，聲音會變大。例如 sound(signal**\*10**,fs);

**步驟四、使用wiener filter濾波器，產生消除雜訊的訊號(Enhancement Speech)**

將帶有雜訊的語音訊號通過wiener filter以消除雜訊，利用wienerscalart96.m輸入信號變數output、取樣頻率 fs、以及採樣噪音秒數s (建議在0~1之間)，將參雜噪音的語音訊號透過wiener filter以消去雜訊。

output1=WienerScalart96(output,fs,s);

sound(output1,fs);

**步驟五、寫出一段可以量化濾波器效能的程式 (比較消除雜訊後語音與原始乾淨語音的差異)**

消除雜訊後之乾淨語音的SSNR值

混雜語音的 SSNR值

評斷的程式分成兩個部分，兩個部分都是在計算各自的SSNR，最後將各自的SSNR按照紅色公式部分計算後，可得SSNRI (Segmental Signal to Noise Ratio Improvement)，即為最後的評斷標準。

Hint1：如果輸出為雙聲道，請先將左聲道取完音框後接著取右聲道，兩句接在一起變成一句。

Hint2：程式在計算SSNR時，請注意兩輸入的維度及長度，若發現當 clean speech (變數 x) 和 processed speech (變數 D)長度不同時，請以經過wienerfilter的音訊長度為準(變數D1)。

Hint3：processed speech表示處理過的語音，可分成是加完雜訊的noisy部分，也可是增強後的Enhancement部分。

Hint4：假設你都按照上面定義及方法寫程式，如果你使用乾淨語音為’01’，噪音為’0129’，雜訊比為1，混合出5秒含雜訊的語音，接著使用wiener filter s=0.8, 然後SSNRI的程式中，frame取180，則會得到SSNRI=17.4296。

|  |  |
| --- | --- |
| 變數名稱 | 意義 |
| X | 乾淨語音 |
| D | 雜訊 |
| D1 | 消除雜訊後的乾淨語音(乾淨語音+噪音，並經過wiener filter後的語音) |
| D2 | 混雜語音(經過audio\_SNR.m後的語音) |
| N | 音框長度(在此專題內請設成180) |
| M | 音框個數 = (音訊長度/N)，若除不盡則無條件去除小數點  例如： = 乾淨語音的第k個音框中的第i個值 |

實驗三

**產生、播放echo(請使用附檔中提供的echo.m)**

此實驗我們將使用工具echo.m來產生一個空間的echo。此程式碼已經內涵RIRG的工具，首先將所需要的各種參數(房間大小、喇叭位置、麥克風位置、反射次數…等等)輸入，RIRG便會產生一個房間的Impulse Response (*h*) ，之後我們再將 *h* 與我們讀入的音訊進行convolution的動作，就可以產生該房間的echo。

[input,fs]=audoiread(‘檔案名稱’); %讀入提供的wav檔以及其取樣頻率

mic=[長 寬 高]; %麥克風位置(公尺)

nn= %反射次數 (請設置1~5)

src=[長 寬 高]; %喇叭位置(公尺)

rm=[長 寬 高]; %房間大小(公尺)

a=1 %反射係數 (請設置-1~1)

h=rir(fs, mic, nn, a, rm, src); %產生房間的*h*

d=filter(h,1,input);

sound(d,fs); %播放echo

Hint：請切記wav檔案必須在同層目錄下

實驗四

**使用所提供的LMS程式來進行回音消除(Matlab)**

使用我們提供的LMS.m程式，觀察濾波器所估測的脈衝響應() 與RIRG所模擬出來的脈衝響應(h )，可使用先前教過的hold on 指令並改變顏色來比較兩者，兩者越相近，代表濾波器所估測的 越正確。此程式中的w即為 ，請選擇最後一列來繪製。

進行此工具時由於考慮到電腦的硬體設備問題、以及方便之後的實驗，所以實驗參數不能像實驗一自己設定，請依照下列的表格來輸入參數。輸入音訊則改為高斯雜訊。

|  |  |
| --- | --- |
| input | randn(1,8e4) |
| fs | 10000 |
| mic (麥克風位置) | [1 2 2] |
| src(聲源位置) | [2 3 1] |
| rm(房間大小) | [5 5 4] |
| a(反射係數) | 1 |
| nn(反射次數) | 1 |
| N(濾波器長度) | 291 |
| mu(步階值) | 0.0006(可自行調整) |

觀察MSE(Mean Square error)

延續實驗四，我們繼續觀察MSE的收斂情形。

此實驗我們透過MSE來觀察、兩者之間的誤差，MSE的值越小，代表兩者差異越小，回音消除的效果越好。此實驗中MSE計算方法為：

此計算方法在本實驗中有兩個用途：

* 將單位轉換成
* 可觀察 error 收斂趨近於0時的詳細數值

而計算MSE的程式已在LMS.m中幫同學寫好，同學們只需將mse\_lms劃出並觀察即可。

實驗五

**更新NLMS(Normalized Least Mean Squares)**

有關回音消除的濾波器，LMS只是其中一種，除了LMS以外還有其他效能更好的濾波器，因此在實驗四中我們將藉由修改full\_lms.m內LMS的演算法，來將LMS更新成NLMS，來比較兩種濾波器估測 以及MSE的效能優劣為何。

Hint：可從底下實驗理論中提供比較兩者濾波器的演算法下手。

四、補充說明

**用Matlab繪出波形圖以及頻譜圖**

[signal,fs]=audioread(‘1.wav');

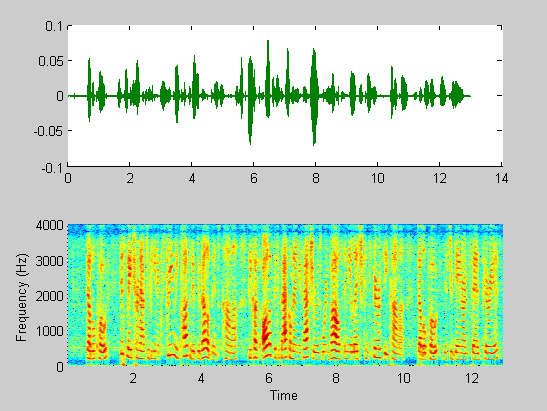
t=(0:length(signal(:,1))-1)/fs;

subplot(2,1,1);

plot(t,signal)

subplot(2,1,2);

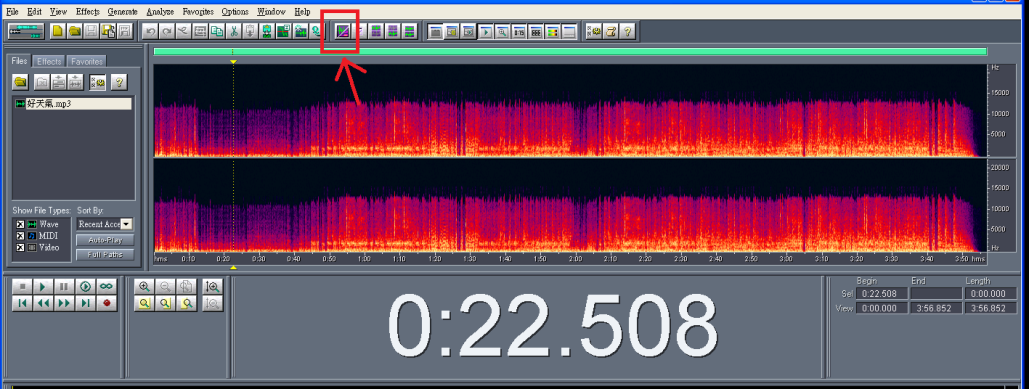
spectrogram(signal(:,1), 1024, 1000, [], fs, 'yaxis')



**用cooledit繪出頻譜圖**

可下載cooledit音訊編輯軟體顯示音軌頻譜圖，可將輸入以及輸出的頻譜圖畫出做比較。

將音訊檔案丟入cooledit中，按照圖示點選顯示頻譜圖。



**書面報告部分**

1. 問題討論

**問題一：將經過wienerscalart96.m消除完雜訊的音檔存檔後並上傳(指定使用乾淨語音2.wav和雜訊horn混合)**

書面報告(圖)：請使用乾淨語音1.wav及三種雜訊，分別畫出其乾淨語音與混雜語音的 波型圖與頻譜圖並比較差異。

**問題二：利用實驗步驟五寫一段小程式使其能夠自動化一次量化10個乾淨語音檔對於3種不同雜訊 (horn、mouse and white noise)、2種訊雜比 (10、50 )的組合，並製作表格來說明數據，請觀察並比較其數據，說明之。**

例如：使用乾淨語音’3’ 、噪音’horn’ 、訊雜比為10時，量化出來的值為12.2072

N = 音框長度 = 180，音框的概念請至下方第九頁SSNR理論部分參閱

m = 音框個數 = ()，若除不盡則無條件去除小數點

程式在計算SSNR時，請注意兩輸入的維度及長度，若發現當乾淨語音(變數 x) 和processed speech (變數 D)長度不同時，請以比較短的長度為準。

表格範例：

表1、乾淨語音：’03’

|  |  |  |
| --- | --- | --- |
| 訊雜比  噪音 | 10 | 50 |
| horn | 12.2072 |  |
| mouse |  |  |

**問題三：調整程式wienerscalart96.m裡的參數，使得Wiener Filter消除雜訊的效能變得更好，並製作與問題二相同的表格，並觀察比較修改wienerscalart96.m參數前後的量化值**

**問題四：將實驗三中各個參數nn、a、rm輸入不同的值之後把畫出來並說明其變化以及echo聽起來的感覺，請至少做三種以上的組合。(ex：nn調大會如何? a如果設為 -1 ，*h*會怎樣? )**

**問題五：請輸入不同的值(步階值)來觀察MSE的收斂情形以及。**

**至少比較三種值(大、中、小)下的MSE。(並說明值調大或調小是否影響MSE收斂速度 ? 過大會如何 ? 過小會如何?)**

**問題六：請比較LMS以及NLMS的差異(MSE收斂情形?input若改為語音檔何者效能較佳?)。並將修改為NLMS後的程式碼上傳。**

**C. 評分標準**

書面報告(80%)

請詳細回答問題與討論六個問題寫成一份報告，在12/10前將報告與m檔上傳至portal並且在12/10上課時將報告以紙本型式印出來繳交

M檔&音檔(20%)

請至portal上傳程式碼(12/10前)與問題二的音檔(11/26當天繳交)

**D. 實驗理論**

Wiener Filter

維納濾波是由作者Norbert.Wiener 在1949發表的理論[1]，此為一種線性濾波，也稱為最小均方誤差濾波器(Minimum mean square error filter)，或是最少平方誤差濾波器(least square error filter )，因為考量到讓誤差最小為其原理。維納濾波器(Wiener filter)除了可以濾除雜訊外，還可以應用在許多地方，單一音源訊號的分離、影像的還原(Image Restoration)、影像雜訊的消除(image noise reduction)。在此，我們使用維納濾波器(Wiener filter)，對麥克風收到的語音訊號做濾波動作。  
我們可以先假設 y[n]為一個含噪音的訊號，x[n]為一個純淨的原始語音信號，d[n]為噪音或雜訊。可由下面式子表示。

期望透過濾波器濾波，使得y[n]通過濾波器後，可以濾除掉雜訊d[n]，得到一個近似 x[n]，純淨的原始語音信號 。假設此為一種線性濾波器，設計的方法為基於統計結果，適當的選擇濾波器係數，讓濾除雜音的訊號 與原始語音的訊號x[n]差異最小。假設此濾波器的脈衝響應(impulse response)為h[n]，輸入為含雜音的訊號y[n]，輸出為 為一個近似x[n]的訊號，

可由下式表示。

(2)

則計算後的誤差能量為(3)式。

將上式展開之後可得(4)式。

E(d)是一個計算後的能量誤差，估測出x[n]和的平方誤差(Squared error)。為選擇一個脈衝響應h[n]讓誤差最小，在上式中對h[n]微分，可以得到(5)。可將(5)再展開之後，得到第(6)式。

其中由(6)可推得以下二式(7)、(8)。

以及

假設d[n]與x[n]不相關，則(9)式成立。

所以，可以推得(10)式和(11)式。此兩式式為了將(6)式簡化成的形式，因此將兩部分分別表示為和，簡化之後對於轉換到頻域比較簡潔，也可以清楚的看出濾波器表現的形式。

將結果(11)式帶入(6)式，可以得到(12)式。

將(12)式轉至頻域可得(13)。

將h[n]從時域轉換到頻域，並移項之後可得。

此即為維納濾波器(Wiener filter)的形式。而 為訊號i功率頻譜密度(Power Spectral Density, PSD ) 。將含雜訊訊號的傅立葉轉換，當作濾波器輸入，可以得到的傅立葉轉換。

必須注意的地方，在以上的推導中，維納濾波器(Wiener filter)是建立在雜訊d[n]與純淨的原始語音信號x[n]不相關的假設下。在實際的應用上，必須在短時間中演算，也就是在一個音框中做計算。加窗之後的音框訊號假設為(15)式。

m表示音框所在位置。將一個音框中寫成維納濾波器(Wiener filter)，其式子如(17)。

其中訊號與噪音只能用估測的方式得到，並不是真的知道確切的數值，所以將其功率頻譜密度(Power Spectral Density)表示成估計值。在頻域中將一個音框訊號通過維納濾波器(Wiener filter)，其輸出就是估計出的語音訊號。

在此我們的目的為濾除雜訊，即用納濾波器(Wiener filter)濾除會干擾麥克風語音辨識的噪音，(14)可用來表示此濾波器。利用(14)式雜訊分離的概念，令為欲保留之語音訊號；為噪音訊號。表示如下。式子中的功率頻譜的密度(Power Spectral Density) 則是由來表示。

上述(18)是將訊號看成由線性疊加而成，而表示組成訊號的成分，也可以看成是訊號的特徵，分別代表語音訊號和噪音訊號的特徵集合，則是各個特徵的權重。最後再將訊號成分分別乘上的權重，可得到各自線性疊加的結果。由於利用維納濾波器(Wiener filter)，因此使用功率頻譜(Power Spectral)來表示特徵，由(18) 式轉至頻域可得(19)式。

(index)音框位置標示為t，如此利用維納濾波器(Wiener filter)來分離語音訊號中含雜訊的部分，可寫成下列式子。

其中可表示為(21)、(22)式。

其中，摻有噪音之語音訊號可以表示成(23)式。

由以上敘述和推導可知，若使用維納濾波器(Wiener filter)來濾除語音訊號中的噪音，需要的有為欲保留之語音訊號；為噪音訊號，各自的特徵，以及各特徵權重 ，由(23)式我們可以看出，為我們欲求得之重要變數。在濾除語音訊號中噪音的過程可以分為兩個部分，一個是訓練過程，一個是分離過程。

在訓練的過程中，目的是將語音訊號和噪音訊號的特徵取出，並加以儲存在資料庫(database)之中。由於維納濾波器(Wiener filter)以功率頻譜密度(Power Spectral Density) 加以用來分離音訊，所以此處功率頻譜密度(Power Spectral Density)為欲尋找之特徵。在分離的過程中，則利用訓練過程得到欲求得之特徵，以及維納濾波概念用來分離含有噪音的語音資訊。

在訓練過程中，將含有噪音的語音訊號輸入，經過前處理，再計算功率頻譜密度(Power Spectral Density)，再將訓練過程中取出的特徵，用來分離語音訊號和噪音訊號，兩種不同特徵的訊號。至於前處理的項目，大致上含有傅立葉轉換(Fourier Transform)、重新取樣、振幅正規劃以及取音框。

SNR (Signal-to-noise ratio)

There are several objective speech quality measures. The most simple one is the Signal to Noise Ratio (SNR) that compares the original and processed speech signals sample by sample. There are also more complex ones that are built based on Human Auditory System model involving complex mathematical calculations. We present the most famous measures in this section. All of them with the exception of the ITU E-model operates on both the original and the processed speech sample. This limitation makes it impossible to work in real time and to include these metrics in designing new mechanisms (rate control or speech codecs design to take into account the user's perception and the network factors). A second disadvantage is that the obtained results do not correlate always with subjective data (thus they cannot measure correctly user's perception). A third drawback is that some of them are computationally extensive. This point limits their usage in lightweight applications including mobile phones. Some of these metrics are designed and optimized basically to consider encoding impairments and restricted conditions, but they do not work efficiently when they used in other conditions (ex. distortion due to the transmission over the network). Some of these methods require a perfect synchronization between the original and processed signals otherwise the performce degrades considerably. In this case several factors including the delay variation's effect cannot be taken into account by these methods. There are three types of objective speech quality measures: time domain, spectral domain, and perceptual domain measures. The time domain measures are usually applicable to analog or waveform coding systems in which the goal is to reproduce the waveform itself. SNR and segmental SNR (SNRseg) are the most known methods. Since the waveform are directly compared in time domain, synchronization of the original and distorted signals is a must. However, synchronization is difficult; if not performed well, the performance is poor. The most simple possible measure is the Signal-to-Noise (SNR) ratio. Its goal is to measure the distortion of the waveform coders that reproduce the input waveform. It is calculated as follows:

where   and  are the original and processed speech samples indexed by   and is the total number of samples point.

Segmental Signal-to-Noise Ratio (segSNR), instead of working on the whole signal, calculates the average of the SNR values of short segments (15 to 20 ms). It is given by:

where  and m are the segment length and the number of segments respectively. segSNR gives better results than SNR for waveform encoders, but it gives very bad results for vocoders. The second type of measures are the spectral domain ones [[1](http://www.irisa.fr/armor/lesmembres/Mohamed/Thesis/node243.html#art194)]. They are generally computed using speech segments typically between 15 and 30 ms long. They are much more reliable than time domain measures and less sensitive to the misalignments between the original and distorted signals. However, these measures are closely related to speech codec design and use the parameters of speech production modules. Hence their ability to adequately describe the listener's auditory response is limited by the constraints of the speech production modules. They include the log likelihood ratio, the Linear Predictive Coding (LPC) parameter distance measures, the cepstral distance, and the weighted slope spectral distance measures (for more details and descriptions see [[1](http://www.irisa.fr/armor/lesmembres/Mohamed/Thesis/node243.html#art194)]). In general, all these methods gives good results for some encoding distortion, but they are not valid for the case when the original speech is passed through a communication system that significantly changes the statistics of the original speech. The third type of objective measures is constituted by the perceptual domain measures [[1](http://www.irisa.fr/armor/lesmembres/Mohamed/Thesis/node243.html#art194)]. In contrast to the spectral domain measures, perceptual domain measures are based on models of human auditory perception. They transform speech signal into a perceptually relevant domain such as bark spectrum or loudness domain, and incorporate human auditory models. They give better prediction of the quality under the condition that the used auditory model used truly describes the human auditorial system. It is clear that this task is very complex and it is not possible to implement exact model of such system. However, by using approximations of the human auditorial system, the obtained results correlate better than that of the other two types of speech measures. Another important point to underline is the fact that these models are optimized for a specific type of speech data; the performance is not good for different speech data. In addition, they have the risk of not describing perceptually important effects relevant to speech quality but simply a curve fitting by parameter optimization. These measures are the most known and used in the literature. We provide a brief description of these metrics as given in [[1](http://www.irisa.fr/armor/lesmembres/Mohamed/Thesis/node243.html#art194)].

[1] W. Yang. “Enhanced Modified Bark Spectral Distortion (EMBSD): an Objective Speech Quality Measrure Based on Audible Distortion and Cognition Model. “ PhD thesis, Temple University Graduate Board, May 1999.

Least mean square algorithm (LMS)

LMS是最被廣泛使用的濾波器演算法，最大的特點就是簡單。運算過程只有加法、乘法，而且它也不需要對相關函數(correlation function)做計算，更不需要複雜的反矩陣運算(matrix inversion)，所以經常被拿來用作比較的基準。

LMS 演算法是使輸出誤差信號的平均平方值最小化（mineralized），而其定義為:

目標信號的估測值與真正目標信號的差就是所謂的估測錯誤（estimation error），而 e(n) 的定義為：

*-*

而d(n)為期望值（desired value）。w(n)為時間點 n 時權重係數向量（weight coefficient vector），其展開式為：

u(n) 是輸入向量（input vector），其展開式為：

其中 L 為濾波器長度（filter length）。所以LMS 演算法，其著重於調控系統期望值d(n)與濾波器輸出 y(n) 之差 e(n)，其演算法不斷更新修改演算中之權重係數向量 *w*(*n*) 值，並使得 e(n) 平方最小，使誤差趨近於零。

各參數的描述如下：

n：目前時間索引值 （The current time index）

u(n)：輸入向量（The vector of buffered input samples at step *n*）

w(n)：係數向量（The vector of filter weight estimates at step *n*）

y(n)：輸出資料（The filtered output at step *n*）

e(n)：誤差值 （The estimation error at step *n*）

d(n)：期望值（The desired response at step *n*）

μ：步階值（The adaptation step size）

以下為LMS演算法之計算式： (以下標註個參數的維度，假設濾波器長度L值為400)

* Filter Output：

1x1 1x400 400x1

* Estimation Error：

1x1 1x1 1x1

* Tap-weight adaptation：

400x1 400x1

LMS演算法其步階參數μ值的取捨問題卻相當的重要， μ 值用來調整加權參數的修正速度，若 μ 值選取過小的話，收斂速度會過於緩慢，若是選取過大時，則是會造成收斂不穩定，導致發散，所以找尋最佳的 μ 值是 LMS演算法很大的課題。μ值的選取上有一定的限制，收斂條件為：

Normalized LMS algorithm (NLMS)：

NLMS演算法是將 LMS 中的 μ 值重新定義，讓 μ 值會隨輸入訊號之 normalization(正規化)作改變，有提升收斂的穩定性。

下列為NLMS演算法所須之計算式：

α：很小的(正)常數（The small positive constant α = 1e-10）

相較於LMS：

各參數的定義和LMS定義相同，新增各參數之意義為

LMS 演算法之穩定度和收歛速度受到 *μ* 值和參考訊號的影響，由於 *μ* 值為一固定值，因此 LMS 的整體收歛速度就受它的影響，所以其收歛速度對變化較快之訊號其反應並不理想。而NLMS 演算法以改善輸入訊號對收歛因子的影響， μ 值隨著時間 n 變化而成為 μ(n)，使之隨時變化而可調控至最佳值。另外為了避免當輸入訊號過小時造成收斂因子（μ）的發散，還加入 α 值。