

Final Report of VSP

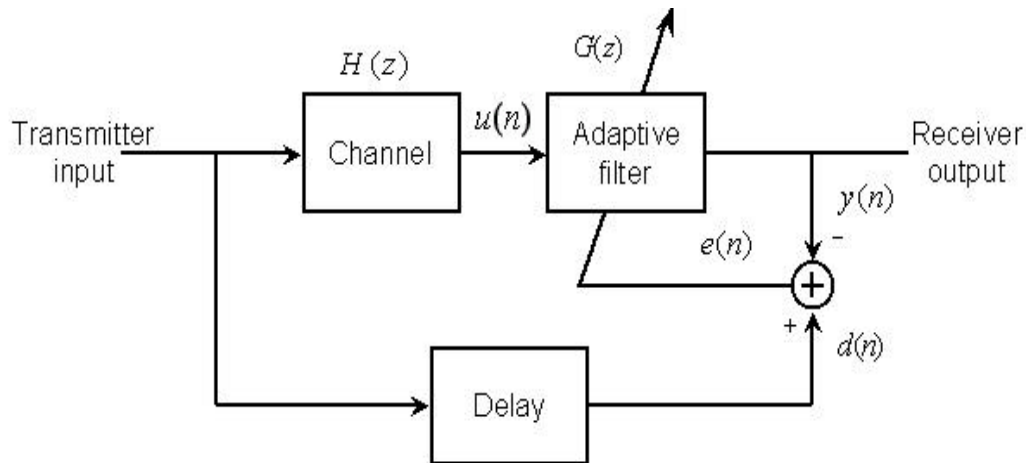
LMS Adaptive filters

學號：R89921153

姓名：陳本欣

Adaptive filter theory

在通訊系統中，由於在傳輸的過程中會有干擾，因此在接收的過程中便需要對收到的信號來做處理，以便使接收者能獲得原先所傳送的資料。濾波器（filter）是一種傳統常用的方法，他可以藉由將轉移函數（transfer function）調成 channel impulse response 的倒數來補償受到破壞的信號，使其回復成原先被傳送到通道上時的狀態。隨著通訊系統的蓬勃發展，在更多變的通道下以高速傳輸使得信號在傳送的過程中受到比以往的破壞要大得多；如現今的無線通訊，由於通道在每一個時間的 impulse response 都會產生變化，在以往的濾波器架構中使用同一組 transfer function 已經無法正確地模擬成通道 impulse response 的倒數。因此在這種通道特性為時變函數的情況下，有人便希望此濾波器也能隨著通道的變化而調整其轉移函數，以便能得到信號回復，而這就是可適性信號處理的基本概念。



圖表 1 可適性信號濾波器方塊圖

上圖即為一個可適性信號濾波器的方塊圖。其中所傳輸的信號（transmitter input）在經過通道以 $H(z)$ 的函數所破壞後得到 $u(n)$ ；然後我們再希望由一個可適性信號濾波器來解回原有的傳輸信號。然而我們要如何去改變這一個濾波器的轉移函數呢？首先我們假設由一個祕密通道可以在經過一個 delay 後收到所要傳輸的信號 $d(n)$ （如傳送 training sequence 時就可以使傳輸 / 接收兩方知道所要傳輸的資料是什麼），另外再將我們由可適性信號濾波器所調回來的信號 $y(n)$ 做比較，可以得到誤差 $e(n)$ ，然後我們再經由 $e(n)$ 和輸入此濾波器的信號 $u(n)$ 來調整其轉移函數。最後我們希望將 $e(n)$ 能調整到 0，這樣便能完全補償信號在傳輸過程中所遭受到的破壞。

由於高速和無線通訊的快速發展，在通訊系統中，傳統固定的濾波器設計不再滿足現今的需求，因此可適性濾波器（Adaptive filter）也越來越被廣泛地使用。

一般來說，在現在的應用中，可適性濾波器主用的實現方法有 LMS 和 RLS 兩種理論。一般來說 LMS 較容易實現，但 RLS 卻有著較好的性能。然而在真正把理論實作成晶片的時候，有許多因素如面積、成本和速度都必須考量在內，也因此我們常常需要對實際的理論作一些簡化的動作，雖然會犧牲一部份的性能，但若換取而來的是實現難度大大地降低，這種方法往往還是會被取用。也因此 LMS 的演算法部份，便陸續有人提出 sign-error、sign-data、sign-sign 的作法來簡化；然而在 1997 年時，有人提出利用 power-of-two 來 quantize 其中 data 和 error 的部份，可以在比原有 LMS 硬體複雜度較低的情況下，獲得差不多的效能。我們將在本文中對這個觀點做一些模擬，並作比較。

LMS & RLS Algorithm

在一般用來實現可適性信號濾波器的理論中，最常見到的方法有兩種，一種是 Least Mean Square(LMS)Algorithm, 另一種則是 Recursive Least Square(RLS) Algorithm。

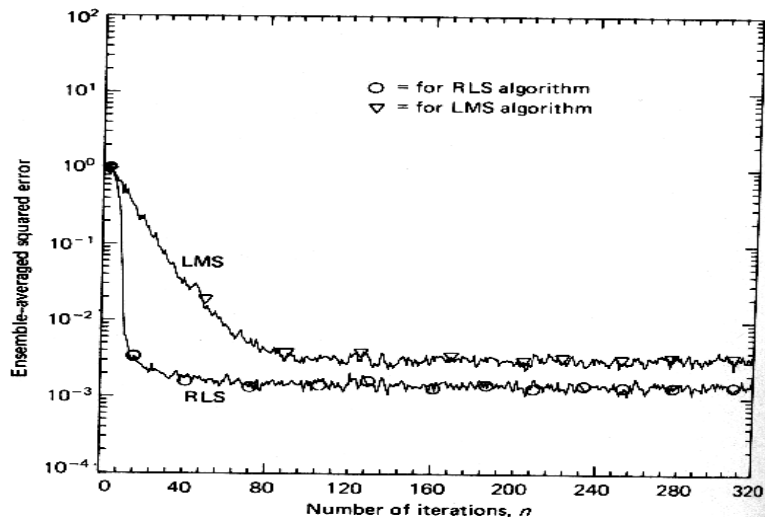
LMS Algorithm 即為直接實行我們在上面所述的方法，在接收的資料一筆筆進來的時候，利用所得到的誤差和接收資料本身來調整濾波器的轉移函數，如下面的方程式所示：

$$w(n+1) = w(n) + \mu f(e(n))g(u(n)) \quad (1)$$

其中 $w(n)$ 是指在時間為 n 的時候，此可適性信號濾波器轉移函數的係數。 μ (step size) 則為一個常數，通常會決定此濾波器所得的誤差是否會收斂 (convergence) 和其收斂的速度快慢。我們通常經由分析一些系統的統計特性，便可以找出 μ 值的範圍。其實訂出 μ 值是設計這一個濾波器最主要的工作之一。

RLS Algorithm 的方法是利用最小平方差，對此時輸入該可適性信號濾波器的 $u(n)$ 數列，找出最適當的一組 $w(n)$ 係數，使其所得到的誤差值 $e(n)$ 為最小。而 recursive 的意思便是在每一筆資料到達接收端的時候，接收器的可適性信號濾波器便再對 $w(n)$ 做修改，得到下一個時間最佳的 $w(n+1)$ ；因此對每一筆資料進入時，我們都需要重新去找到一組最佳的 w 來使得誤差隨著時間增加而越來越小；也因此需要相當多的運算。相對應 LMS 只是乘上一個常數來調整 w 值，RLS 在實現上較為困難。

這兩個方法比較起來，RLS Algorithm 可以得到較 LMS 更好的效果，尤其是在收斂速度方面要快得多。但在實作方面，由於 RLS Algorithm 牽涉到太多的數學式計算，因此在硬體實現上有其相當的難度。也因此到目前為止，很多實際上的製作都是以 LMS 為主。此兩種演算法的運作效能 (以 learning curve 表示) 如下圖所示：



圖表 2 LMS 與 RLS Algorithm 的 Learning Curve

Variety LMS Algorithms

雖然在比較上，LMS Algorithm 較 RLS 來得簡單許多，但由於許多實作上面積大小、功率消耗等的考量，依然有許多人想出方法來簡化 LMS Algorithm 的實現。如現在無線行動通訊裡，硬體的面積和功率消耗都要越小越好，因此合理地犧牲一點效能來換取面積和功率的大幅縮減是必需的。

我們知道在傳統的 LMS 裡，對於複數資料的運算共需要 $4N$ 個實數乘法器和 $4N$ 個實數加法器（ N 為此濾波器的 tap 數）。當 N 很大時， $4N$ 個乘法器將會佔去相當大的面積，這對有些應用來說是很難去實現的。因此有人提出所謂 sign-base 的 LMS Algorithm，其原理是將更新濾波器係數時所需要用到的誤差和輸入數列作減化，取其 sign digit，並對 step size μ 值取 2 的幕次方來表示，我們在下頁做一些簡單的介紹：

$$w(n+1) = w(n) + \mu f(e(n))g(u(n))$$

Sign-error LMS：上式是將傳統 LMS Algorithm 的計算式做改良而成；在傳統的 LMS 中，我們將 $f(e(n))$ 定義為 $e(n)$ ， $g(u(n))$ 則定義為 $u(n)$ 。而在 sign-error LMS 中，我們將 $f(x)$ 函數定義為取其誤差 $e(n)$ 的 sign digit，而 $g(x)$ 函數則保持不變。因此我們可以將實現需求大為降低，由原先所需要的 $4N$ 個實數乘法器和 $4N$ 個實數加法器大幅縮減到 $2N$ 個移位器 (shifter) (乘以 μ 的時候所用) 和 $4N$ 個實數加法器。

Sign-data LMS：和上例相仿，我們將 $g(x)$ 函數定義為取其輸入數列 $u(n)$ 的 sign digit，而 $f(x)$ 則和原來一樣。這樣做我們可以把實現需求降為一個移位器和 $4N$ 個實數加法器。

Sign-sign LMS：顧名思義，我們將 $f(x)$ 和 $g(x)$ 都定義成取其輸入的 sign digit；在

這個方法中硬體更大量的被減化，但其效能也大大地降低。
綜合以上所述可整理得下表：

Algorithm	Area
LMS	$N(4L+5L^2)$
Sign-error LMS	$N(4L+8L)$
Sign-data LMS	$4NL+8L$
Sign-sign LMS	$2NL+2L$

圖表 3 sign-base LMS Algorithm 硬體複雜度比較

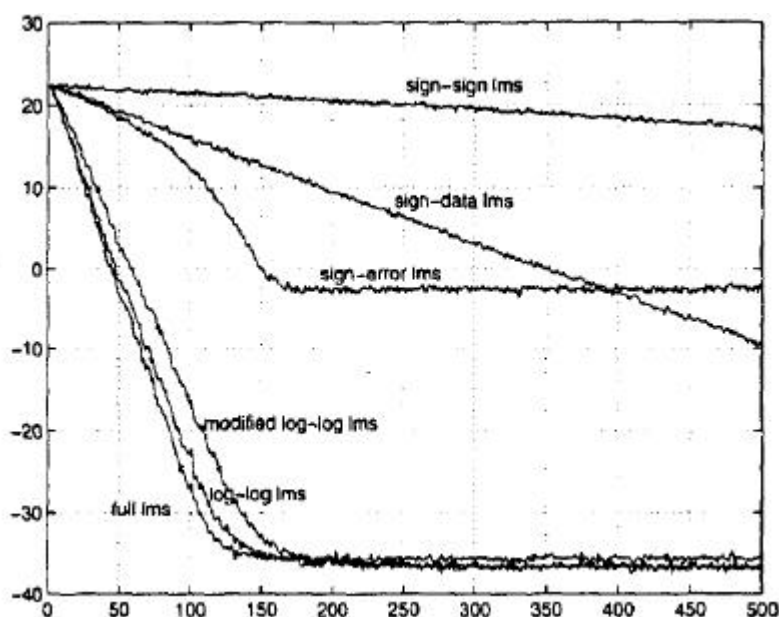
以上所述是我們所見到的一些傳統的做法[1,2]。然而有一篇在 1997 年提出的 paper 宣稱有另一種作法可以使在 LMS Algorithm 的簡化過程中，其效能不致於受到太大的影響，而實現的複雜度卻又可以比原有的 LMS 要降低許多。他們叫這個 Algorithm 為 Log-log LMS Algorithm。

這個演算法的概念跟上述的 sign base LMS 有點像，是將方程式(1)中用來更新下一個一時間的 w 所使用的的誤差 $e(n)$ 和輸入資料 $u(n)$ 都用 2 的冪次方 (power of two) 來表示；也就是當我們決定要用到 n 個 bit 時，我們就採用 2^{-1} 、 2^{-2} 、 2^{-n} 來表示這個數。因此我們可以推出下列的轉換方程式，其中 $\text{sgn}(x)$ 為取 x 的正負值，而 b 則代表此 log-log algorithm 所使用的 bit 數：

$$\begin{aligned}
 Q(x) &= \text{sgn}(x) && \text{if } |x| \geq 1 \\
 Q(x) &= 2^{\lfloor \log_2(x) \rfloor} \text{sgn}(x) && \text{if } 2^{-b+1} \leq |x| < 1 \\
 Q(x) &= 0 && \text{if } |x| < 2^{-b+1}
 \end{aligned}$$

也因為使用這個方法的緣故，因此在乘法方面可以大大地化減，兩個用 2 的冪次方表示的數在乘法方面只需要移位器和加法器就可以達成，省去了乘法器在實現上的難度。我們可以推出這個架構共需要 $4N+kN$ 個加法器（其中 k 趨近於 1），比原有的 LMS 減少了實現的複雜度。作者在文章中採用了一個使用 256 QAM 的纜線數據機（cable modem）來做為例子，其中濾波器的 tap 數 $N = 10$ 、step size $\mu = 2^{-11}$ ，SNR = 40dB，所畫出的 learning curve 如下圖所示。

我們可以在這個例子裡看到，sign-sign LMS 在收斂速度和收斂之後的誤差表現最差（但他的架構最簡單）。而 sign-error LMS 雖然很快就收斂得到結果，但他收斂所得的誤差則比原有的 LMS 多了約 30dB；sign-data LMS 則在收斂結果上有較好的表現，但收斂速度相當地慢，甚至超出了這個畫面外才收斂，這在高速或即時的系統應用中是一大缺點。至於 log-log LMS 的效能則相當接近於原有 LMS 的架構，而我們在上面又看到該架構的硬體複雜度亦比起原有的架構減少很多，因此相當適用於這個例子。



圖表 4 各項 LMS Algorithm 的 learning curve

然而，我們在 paper 的 survey 中，卻沒有發現有關這種將 error 和 data 皆以 2 的幕次方數來表示的例子。最早有相關的例子是在 1986 年所提出，將 sign function 的 LMS 取 4 個 level 取代原本只取 sign bit 的方式，可以獲得較好的結果[3]；另外也有人把這種技巧用來改進原有的 sign-base LMS[4][5]，在這兩篇論文中，作者完整地推導在 sign-error 上所做的應用，得到的結論是其 learning curve 最終所得的誤差值會隨著 noise 的 variance 和 μ 值的降低而變小；並且隨著用來 quantize 誤差值所用的 bit 數 b 而呈現指數性地變小，直到達到飽和的狀態（即 b 再增大亦不會影響該值），這個特性應該也適用於 log log LMS 裡。

最後，在一般可適性濾波器的設計中，裡面的 μ 值也常用 2 的幕次方來表示，因為這樣一來，原本需要一個乘法器來執行乘以 μ 的動作也變成只需一個移位器便可以完成；這樣一來我們可以把 log log LMS 變成 log-data、log-error LMS，這三者所用來更新該濾波器係數的運算皆只需要移位器而不需要任何的乘法器。在這樣的情況下，log log LMS 在實作上節省的成本不會比其它二者要來得少很多；因此這理論是否實用，便有待討論了。

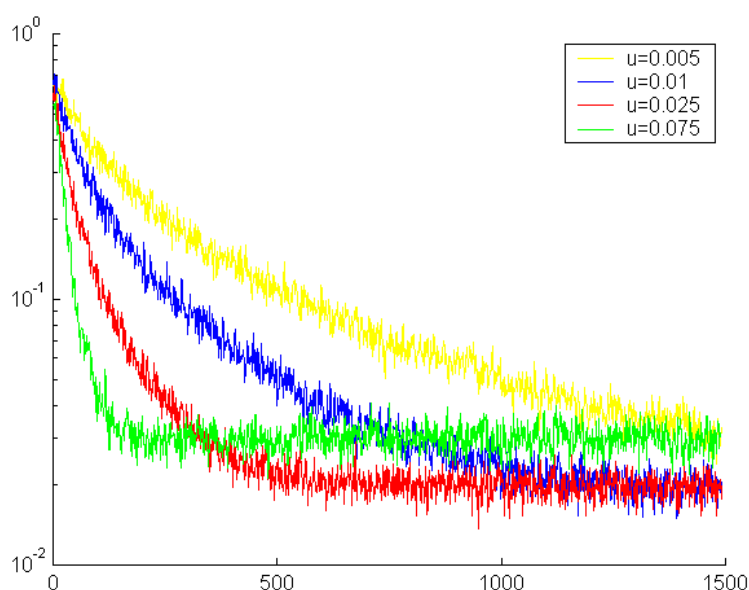
Simulation and discussion

在接下來的內容中，我們將撰寫一 matlab 程式來驗證上述所介紹的各項 LMS Algorithm 的實際表現為何。

我們根據書上所給的範例來做為模擬的基礎，所採用的通道 impulse response $h(n)$ 是一個 raise cosine 函數：

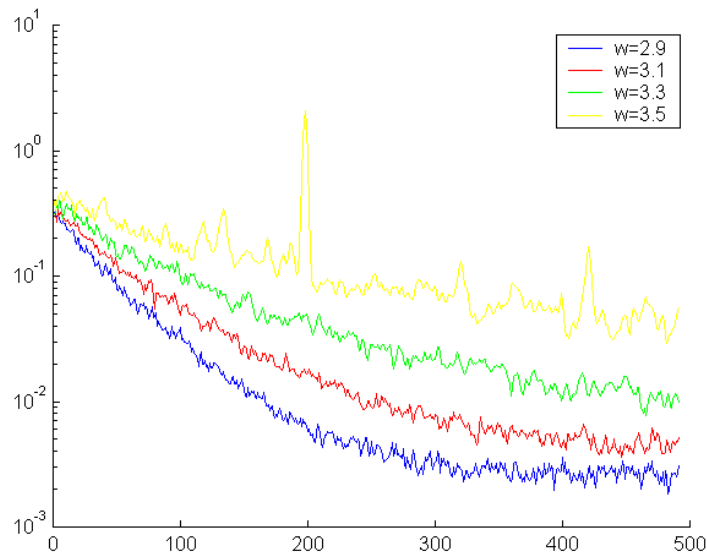
```
h=[0 0.5*(1+cos(2*pi./distortion*(-1))) 0.5*(1+cos(0))  
   0.5*(1+cos(2*pi./distortion)) 0]
```

其中 w 則為通道遭到破壞程度的指標，濾波器的 tap 數為 11；首先我們先觀察 step size μ 值對收斂速度和結果的影響，採用的是最原始的 LMS 架構，所取的 w 值為 3.1，noise 的 variance 為 0.01， μ 值分別為 0.075、0.025、0.01 和 0.005。所得結果如下圖，我們可以很明顯地看到 μ 值對於該 LMS adaptive filter 收斂性質的影響。一般來說， μ 值越大收斂的速度會越快，但該收斂所得的誤差則較大。



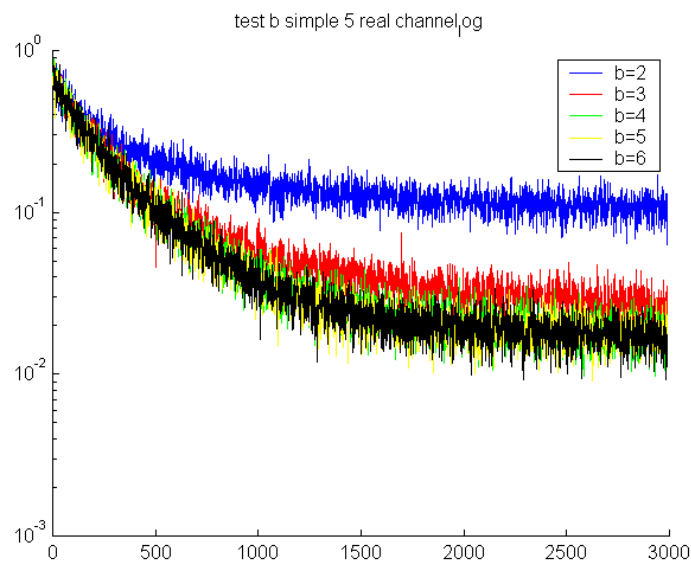
圖表 5 μ 值對收斂性質的影響

接下來我們便以不同的 w 來做比較，說明 w 值的不同對於濾波器作用所造成的影響；如下頁圖 6，我們可以發現當 w 越大的時候，濾波器所能回復原先信號的能力越差，即所造成的誤差為最大。如此我們便可由該 w 值來控制所希望通道的好壞。



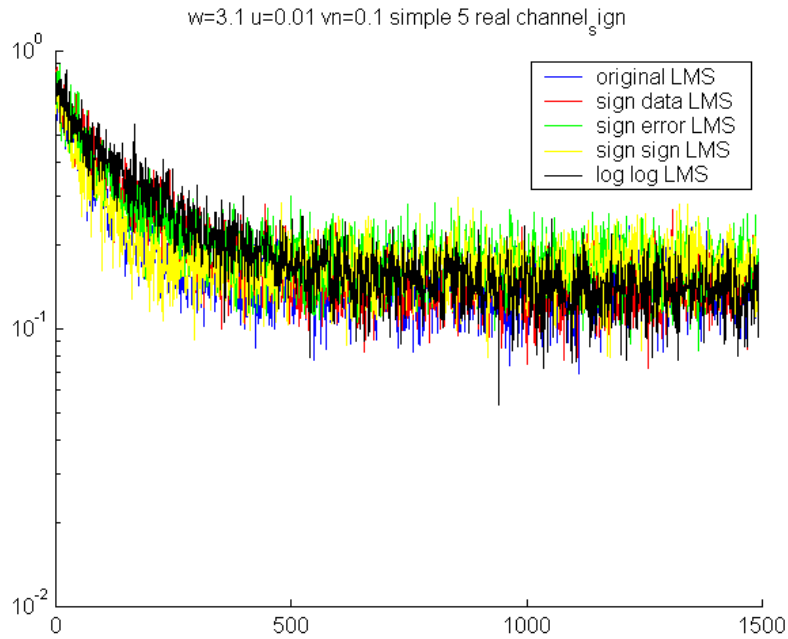
圖表 6 w 值對收斂性質的影響

接下來我們在比較各個 LMS 演算法之前，我們所先要決定的是 $\log \log$ LMS 所用來 quantize 的 bit 數 b 。在之前的討論我們說過，在使用 power of two 的方法時，所取用的 bit 數 b 值會在一定值時達到飽和，即不再對最終誤差的降低有所貢獻；因此我們以 $\log \log$ LMS 來做模擬，我們發現在各種不同的狀況下，誤差先會隨著 b 值的提高而大幅減少，最後達到飽和，與我們之前所說的符合。以下便秀出我們模擬的其中 1 個圖，我們可以看到當 $b=4$ 的時候，誤差便降到最低，之後無論 b 再提高，誤差也不會隨之降低。我們在模擬各種情況之後，將 b 值定為 6，來進行以下的模擬。

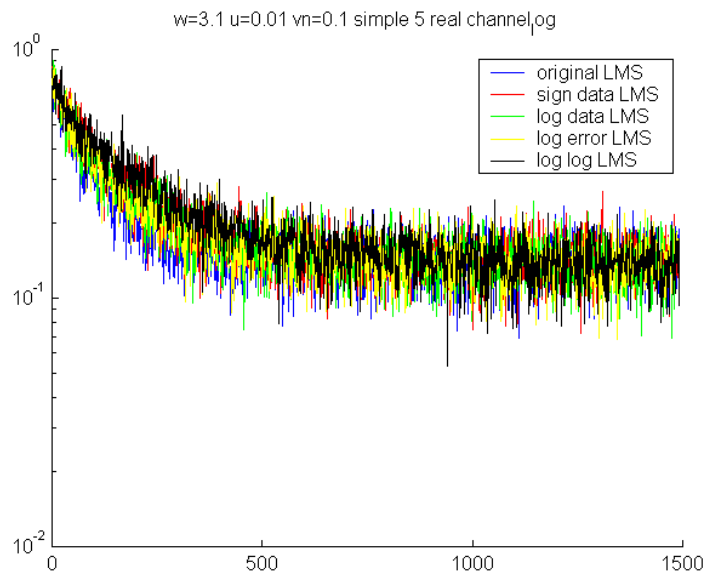


圖表 7 b 值的飽和現象

接下來我們便對不同的 LMS 架構來做模擬和比較。採用的架構有原先的 LMS、sign-error、sign-data、sign-sign 和 log-log LMS。採用的通道 impulse response 跟上面一樣， μ 值則採用 0.01， $w=3.1$ ，noise variance = 0.1，所得結果如下圖 8 所示，我們可以發現不同的地方並不多，除了 sign-error 和 sign-sign LMS 所得的收斂誤差較大外，其它的 LMS 所得的性質均差不多。



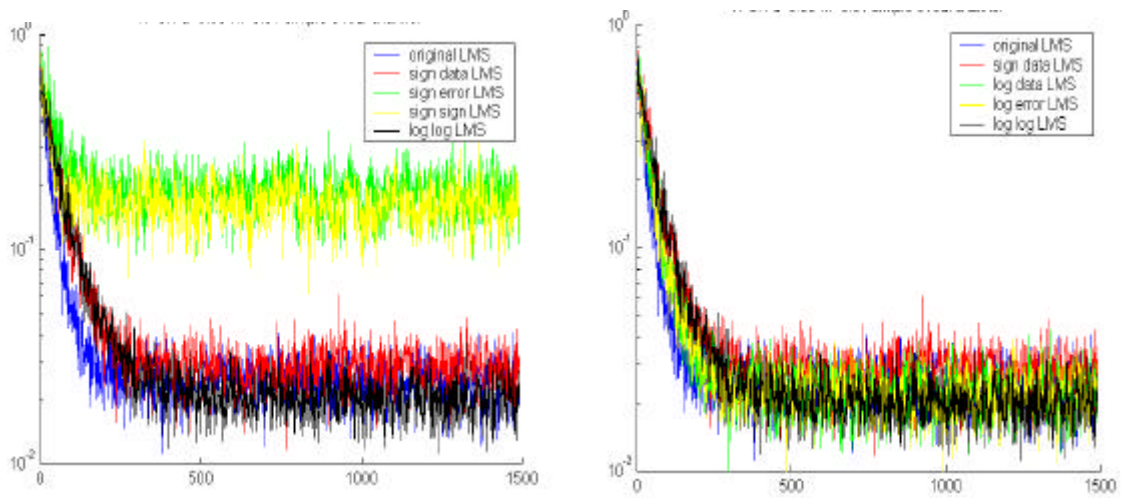
(a) sign base LMS 與 log log LMS



(b) log data、log error LMS 與 log log LMS

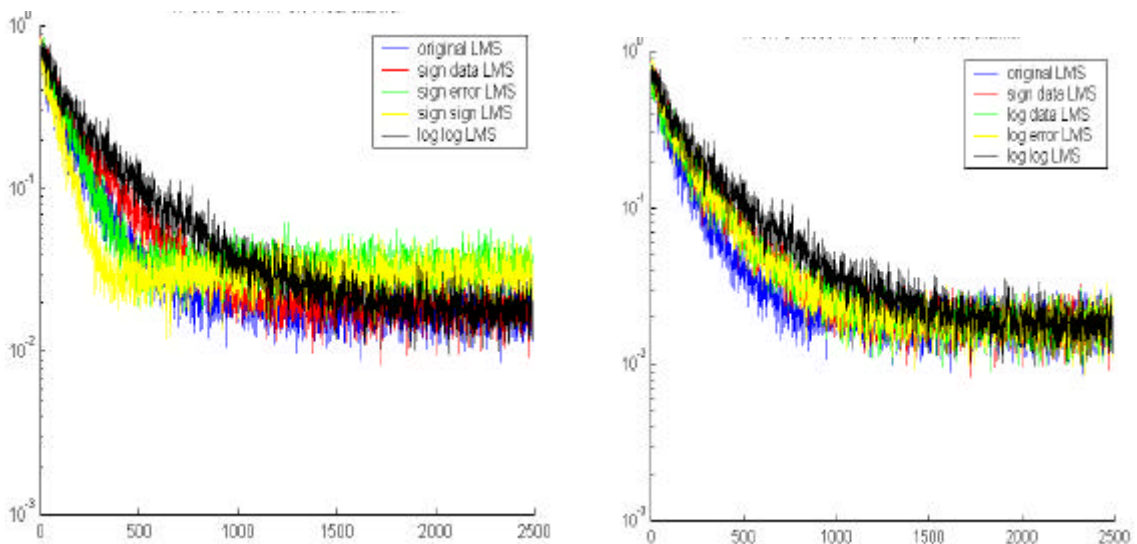
圖表 8 $w=3.1$ $\mu=0.01$ $vn=0.1$ 時各種 LMS 的 learning curve

因此我們將 v_n 調低為 0.01，並以不同的兩種 μ (0.01, 0.05) 值來作討論。圖表 9 為 μ 值 0.05 時的情況；在我們把 v_n 調低之後，在圖(a)很明顯地見到 sign-sign 和 sign-error 兩種演算法所得的誤差都相當大；而本文所要探討的 log-log LMS 則有相當不錯的結果，收斂速度只比原本的 LMS 慢一點點，但收斂所得的誤差值是一樣的。另外在圖(b)我們則可以看到各種 log LMS 演算法的比較圖，我們發現 log-data 和 log-error 在收斂速度上皆比 log log LMS 來得好，而最終所得的誤差也一樣。



(a) sign base LMS 與 log log LMS (b) log data log error LMS 與 log log LMS

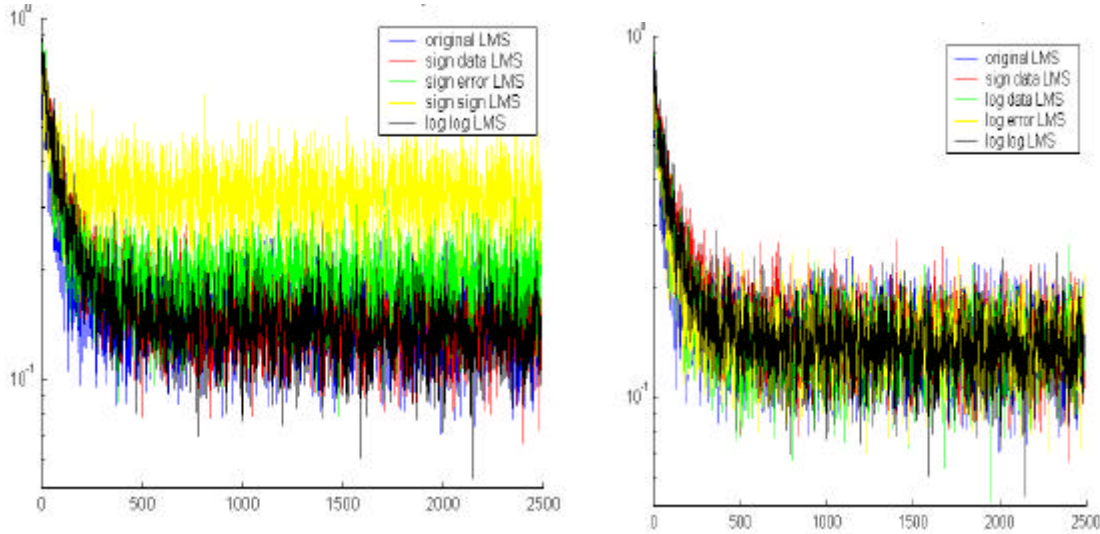
圖表 9 $w=3.1$ $\mu=0.05$ $v_n=0.01$ 時各種 LMS 的 learning curve



(a) sign base LMS 與 log log LMS (b) log data log error LMS 與 log log LMS

圖表 10 $w=3.1$ $\mu=0.01$ $v_n=0.01$ 時各種 LMS 的 learning curve

在圖表 10 裡，我們很驚訝地發現 μ 值縮小後，sign-sign 和 sign-error LMS 在收斂速度上都超越了原先的 LMS（可以以較少的時間獲得相同的誤差值）！不過在大家都收斂之後，原本的 LMS、sign-data 和 log-log 演算法收斂的誤差還是較好。我們猜測可能是因為課本使用的通道 impulse response 是對稱的關係；因此我們把原先的通道 impulse response 打亂，另外亦把其對稱性破壞掉，經由 matlab 模擬可得結果如下；如我們所預測的，sign sign 和 sign error LMS 演算法在較亂的 channel 裡更正能力較差，不但速度退回來比 LMS 慢，而且最終所得誤差也高了許多。



(a) sign base LMS 與 log log LMS (b) log data LMS 與 log log LMS
圖表 11 channel 打亂時各種 LMS 的 learning curve

由於這個現象在自己的觀念裡是不曾出現過的，因此找了些之前的 paper，希望能再更深入了解關於收斂速度的比較。在很多篇的 paper[8][9]裡其實都有提到過，若要在 steady state 狀況下（即濾波器已收斂）獲得相同的誤差值，則 sign base 的收斂速度會比原先 LMS 慢上許多；不過若在不同的收斂結果時，sign base 的收斂速度便有可能較快（如[3]的 fig 便是一個例子）。其中很值得一提的是在 [9]該篇 paper 中，作者推導出原來的和 sign base LMS 收斂速度的表示式如下：

LMS :
$$T_{av}^{LMS} = \ln(\alpha/\delta)(\mu N)^{-1} \sum_{p=1}^N \frac{1}{\lambda_p}.$$

Sign-data LMS :
$$T_{av}^{SRA} = \sqrt{\pi/2}\sigma_x \ln(\alpha/\delta)(\mu N)^{-1} \sum_{p=1}^N \frac{1}{\lambda_p}.$$

Sign-error LMS :
$$T_{av}^{SA} = [\mu N \sqrt{2/\pi}]^{-1} (\alpha - \delta) \sum_{n=1}^N \lambda_p^{-1/2}, \quad \mu \sqrt{\frac{2}{\pi}} \lambda_N \leq \delta, \quad \lambda_1 \delta^2 > \sigma_b^2$$

$$T_{av}^{SA} = [\mu N \sqrt{2/\pi}]^{-1} \sigma_b \ln(\alpha/\delta) \sum_{p=1}^N \lambda_p^{-1}, \quad \mu \sqrt{\frac{2}{\pi}} \lambda_N \ll \alpha, \quad \lambda_N \alpha^2 < \sigma_b^2$$

Sign-sign LMS :

$$T^{\text{SSA}}(\alpha, \mathbf{A}_p) = \left[\frac{2\mu}{\pi\sigma_x} \sqrt{\lambda_p} \right]^{-1} (\alpha - \delta), \quad \lambda_p \delta^2 > \sigma_b^2, \quad \frac{2\mu}{\pi\sigma_x} \sqrt{\lambda_p} \leq \delta$$

$$T^{\text{SSA}}(\alpha, \mathbf{A}_p) = \left[\frac{2\mu}{\pi\sigma_x} \sqrt{\lambda_p} \right]^{-1} (\alpha - \sigma_b \lambda_p^{-1/2}) + \left[\frac{2\mu}{\pi\sigma_x} \lambda_p \right]^{-1} \sigma_b \ln(\sigma_b \delta^{-1} \lambda_p^{-1/2}),$$

$$\lambda_p \delta^2 < \sigma_b^2 < \lambda_p \alpha^2, \quad \frac{2\mu}{\pi\sigma_x} \sqrt{\lambda_p} \leq \delta$$

$$T^{\text{SSA}}(\alpha, \mathbf{A}_p) = \left[\frac{2\mu}{\pi\sigma_x} \lambda_p \right]^{-1} \sigma_b \ln(\alpha/\delta), \quad \lambda_p \alpha^2 < \sigma_b^2, \quad \frac{2\mu}{\pi\sigma_x} \sqrt{\lambda_p} \ll \alpha.$$

其中 α 是收斂時所得的濾波器係數和正確的係數差， δ 則是一開始係數的初始值， σ_b 則是 noise 的 variance， λ_p 為輸入數據的 covariance matrix 的 eigenvalue。由上式我們可以看到，只有 sign-sign 和 sign-error 和 σ_b 有關；而收斂所得的誤差不同更會造成 α 值的不同，也因此會使 sign-sign 和 sign-error LMS 產生較快的收斂速度看起來是合理的。

Conclusion

由上面各項假設的模擬結果，我們可以知道在我們所設定的條件之下，log-log LMS 的效能表現還算不錯，但似乎仍無法和 log-data 或是 log-error 這兩種一樣經過 quantize 的 LMS 演算法做出明顯應用上的區隔。事實上，已經有一些實作是應用 log-error 的方法（如 gigabit 的 equalizer），而且也能獲得不錯的效果。在我們上面的模擬之中，我們可以知道不同的通道對於不同演算法的執行效果會有影響；其中有幾個在 log-log 和其它 log base 的表現上幾乎差不多，此時 log log LMS 便可以較低的實現成本獲得優勢。

What I do

1. 一些與 LMS 相關的論文調查與閱讀
2. 檢驗 log log LMS 的執行效能
3. 以程式模擬來驗證一些論文中所提到的演算法分析與特性

Reference

- [1] A. Gersho, " **Adaptive Filtering with Binary Reinforcement,**" *IEEE Trans. On Info. Thy., IT-30*, pp.191-199, March 1984.
- [2] J. R. Trierchler, et. Al., **Theroy and Design of Adaptive Filters,** John Wiley and Sons, 1987.
- [3] C.P.Kwong, " **Dual Sign Algorithm for Adaptive Filtering** " *IEEE Trans. Commun., vol. Com-34*, pp.1272-1275, Dec. 1986
- [4] E. Eweda, " **Convergence Analysis and Design of an Adaptive Filter with Finite-Bit Power-of-Two Quantized Error,**" *IEEE Trans. Circuits and systems* , vol. 39, No.2, Feb 1992
- [5] E. Eweda, " **Comment on Convergence Analysis and Design of an Adaptive Filter with Finite-Bit Power-of-Two Quantized Error,**" *IEEE Trans. Circuits and systems* , vol. 42, No.7, July 1995
- [6] S. H. Cho and V. J. Mathews, " **Tracking analysis of the sign algorithm in nonstationary environments,**" *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 2046–2057, Dec. 1990.
- [7] E. Eweda, " **Convergence of the sign algorithm for adaptive filtering with correlated data,**" *IEEE Trans. Inform. Theory*, vol. 37, pp. 1450–1457, Sept. 1991.
- [8] Theo A.C.M. Claasen " **Comparison of convergence of two algorithms for adaptive FIR digital filters,**" *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 670-678, June. 1981.
- [8] **The log-log LMS algorithm**
Mahant-Shetti, S.S.; Hosur, S.; Gatherer, A. Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on , Volume: 3 , 1997 Page(s): 2357 -2360 vol.3
- [9] E. Eweda, " **Transient Performance Degradation of the LMS, RLS, Sign, Signed Regressor, and Sign-Sign Algorithms with Data Correlation,**" *IEEE Trans. Circuits and systems* , vol. 46, No.8, AUG 1999