

國立臺北大學通訊工程學系

1132 無線通訊(U4615)

Wireless Communications

Project 2 - Adaptive Equalizer

學號：411186028 姓名：鐘婉庭

此次作業是透過建立 Rayleigh 衰落加上 AWGN 雜訊的通道模型，設計並比較 LMS 與 RLS 兩種自適應等化器的性能。LMS 演算法以最小均方誤差為目標，結構簡單，收斂速度取決於步長參數；RLS 則藉由遞迴最小平方方法快速收斂，但計算複雜度較高。透過 MSE 曲線、steady-state misadjustment、以及 BER vs. SNR 分析來進行兩種演算法的性能評估。

壹、 背景理論：

通道模型：Rayleigh 衰落與加性白高斯雜訊 (AWGN)

在無線通訊中，訊號常因多徑效應而產生幅度與相位的隨機變化，形成所謂的衰落現象。本作業中，通道以 **Rayleigh 衰落模型** 模擬，並根據 Jakes 頻譜產生通道增益，而最大 Doppler 頻率 f_D 決定通道變化速度。每個 tap $h_k[n]$ 隨時間變動，且符合 Rayleigh 分布。通道輸出同時受到**加性白高斯雜訊** (Additive White Gaussian Noise, AWGN) 影響。雜訊模型我們假設為平均值為 0.001 且變異數為 σ_v^2 的 AWGN，每個時間點的雜訊記作 $v[n] \sim N(0.001, \sigma_v^2)$ 。

因此，最後的接收訊號模型可以表示為「通過多條衰落路徑的傳送訊號」+「隨機的高斯雜訊」：

$$y[n] = \sum_{k=0}^{L-1} h_k[n]x[n-k] + v[n]$$

其中： $x[n]$ 為傳送出來的符號， $h_k[n]$ 為第 k 條路徑在時間 n 的通道增益， L 表示 tap 數（總共的路徑數）。整條式子意義為每一條路徑將延遲 k 個時間單位的傳送訊號 $x[n-k]$ 乘上自己的 Rayleigh 時變衰落係數 $h_k[n]$ ，再把所有路徑的結果加總，最後加上通道中的雜訊 $v[n]$ ，就得到最終的接收訊號 $y[n]$ 。

等化器結構與誤差定義

為了抵抗通道造成的失真與干擾，我們設計一個**線性 FIR 自適應等化器**，並以最小化輸出誤差為目標。

等化器的輸入向量定義為：

$$y_n = [y[n], y[n-1], \dots, y[n-M+1]]^T$$

等化器本身也有一組「權重係數」，這組權重會根據收到的訊號自我調整，學習怎麼組合 $y[n], y[n-1], \dots$ 來最接近傳送端的訊號，權重向量定義為：

$$w(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T$$

為了量化等化器修正的成效，我們又定義一個系統誤差為「正確答案」－「等化器估計值」，公式如下：

$$e(n) = x[n - D] - w^T(n) y_n$$

其中， $x[n - D]$ 表示原本應該要傳到接收端的正確訊號，但因為通道有延遲，需要往前 D 個時間點來對齊。 $w^T(n) y_n$ 則是等化器根據 $y[n]$, $y[n - 1]$, ... 推算出來的估計值。

自適應演算法 (Adaptive Algorithm)

LMS (Least Mean Square) 演算法 → 看現在這點誤差來最小化誤差平方 $e^2(n)$

LMS 是一種基於梯度下降法的自適應濾波器演算法，它透過最小化「期望誤差平方值」來調整濾波器權重，目標是使濾波器輸出盡可能逼近理想訊號。在每一個時間點，LMS 使用當前的接收樣本與對應的目標輸出（訓練資料）計算誤差，然後根據此誤差大小，微幅調整濾波器係數。調整的速度由步進率（Step Size, μ ）所控制。

RLS (Recursive Least Squares) 演算法 → 看過去所有誤差的加權總和做修正

RLS 則是使用歷史所有樣本來估計最小化誤差平方總和，具備比 LMS 更快的收斂速度與更佳的穩定性。RLS 利用遞迴方式更新其濾波器係數與共變異數矩陣，不僅考慮當下的誤差，也會根據先前資料動態調整。其更新過程中有一個稱為遺忘因子（Forgetting Factor, λ ）的參數，用來平衡近期資料與歷史資料的重要性。

貳、模擬方法與步驟：

Step 1: 定義模擬參數

通道脈衝響應：在背景理論中，Rayleigh 多徑通道的第 k 條路徑可表示為：

$$h_k[n] = \alpha_k(t) e^{j\phi_k(t)}$$

其中 $\alpha_k(t) \sim \text{Rayleigh}(1)$ 為第 k 條路徑的時變振幅，服從 Rayleigh 分佈，表示快速衰落； $\phi_k(t) \sim \text{Uniform}[0, 2\pi]$ 為相位同時服從均勻分布； f_D 是最大 Doppler 頻移，表示移動造成的通道變異速率，會影響 $\alpha_k(t)$ 的頻率擴展。

我使用 MATLAB 的 `comm.RayleighChannel` 模擬多徑衰落通道，設定如下：

```
chan = comm.RayleighChannel(...  
    'SampleRate', Fs, ...  
    'MaximumDopplerShift', fD, ...  
    'PathDelays', (0:L-1)/Fs, ...  
    'AveragePathGains', zeros(1,L));  
Fs=1000;  
fD=5;  
L=3;
```

圖一：Rayleigh 通道建立

圖二：通道參數設定

設置取樣率為 F_s ，最大 Doppler 頻移為 f_D ，共 L 條延遲路徑，每條延遲依照取樣時間均勻分佈。各路徑平均增益相同，構成一組典型的均勻分佈 Rayleigh 多徑通道模型。

測試迭代數：

```
N = 500;
N_lms=1500;
N_RLS = 200;
M = 8;
ite500=N+M;
ite1500 = N_lms+M;
ite200=N_RLS+M;
```

圖三：迭代數參數設定

在本次模擬中，考量 LMS 與 RLS 演算法在收斂速度上的差異，設定了不同的更新次數(iterations)，其中 LMS 需要較長時間以觀察其收斂至穩態的行為，因此我設定次數為 $N_{LMS} = 1500$ ；而 RLS 收斂速度較快，故僅需要 $N_{RLS} = 200$ 即可觀察。另外，為了進行初步比較，也設置了一組測試迭代數 $N = 500$ 。

由於等化器結構採用長度為 $M = 8 \text{ taps}$ 的 FIR 線性濾波器，每次權重更新需要取用包含當前樣本在內的 M 個連續接收樣本，因此必須在原設定的更新次數基礎上額外增加 M 筆資料，以確保在每次更新時皆有足夠資料形成完整的輸入向量。因此總資料長度設定為：

$$Total\ Samples = N + M$$

對應到不同設定，資料長度分別為：

$$\text{短測試： } ite500 = 500 + 8 = 508$$

$$LMS\text{測試： } ite1500 = 1500 + 8 = 1508$$

$$RLS\text{測試： } ite200 = 200 + 8 = 208$$

決策延遲 Decision Delay：

設定 $delay$ 為 5，此數值經由延遲參數 sweep 分析後選定，相關結果與 LMS 性能的關聯將於後續說明（見「延遲參數調整對 LMS 性能影響」部分）。

通道雜訊變異數：為模擬低噪聲通訊情境，設定為 $\sigma_v^2 = 0.001$ 。

蒙地卡羅模擬次數 (Monte Carlo Runs)：選擇 500 次做為模擬次數。

LMS 初始參數設定：為了在收斂速度與穩態誤差間取得合理平衡，步進尺寸 μ

設定為 0.005，符合 LMS 收斂條件 $0 < \mu < \frac{2}{\lambda_{max}}$ 。

RLS 初始參數設定：RLS 初始誤差協方差矩陣設定為

$$P(0) = \delta^{-1} I$$

其中， $\delta = 250$ ， I 為單位矩陣，依據書上的建議作為初始化設定，確保初期收斂穩定並避免數值問題，並搭配遺忘因子 $\lambda = 0.98$ 達到良好的收斂表現。

```
delay = 5;
noise_var = 0.001;
runs = 500;           %Monte Carlo runs
mu = 0.005;           %LMS initial parameter
delta = 250;          %RLS initial parameter
lambda = 0.98;
```

圖四：其他參數設定

Step 2: 產生訓練資料與通道模型

在模擬自適應等化器（如 LMS 或 RLS）前，首先需建立傳送端的輸入訊號，而我選用 BPSK（二位元相位鍵移調變，Binary Phase Shift Keying）一種最基本且常用的數位調變方式。

- (1) **隨機位元產生**：首先，系統會模擬一串隨機的**二元訊號**，代表資料來源。
- (2) **調變映射**：這些 0 和 1 會被轉換成對應的調變符號。在 BPSK 調變中，位元 0 被對應到實數 -1 （相位 180° ）位元 1 被對應到實數 $+1$ （相位 0° ）。這種設計利用兩種相反極性（正負號）的實數符號，對應二元位元的不同狀態，使其在通道傳輸中能透過極性或相位加以辨識。
- (3) **成為訊號源**：經過上述轉換後，這串 -1 與 $+1$ 組成的 BPSK 調變訊號就代表了**傳送端真正要送出的訊號**，在模擬中我們通常假設每個符號的傳送間隔是等長的，並通過一個具有通道響應與加性雜訊的模型通道，進入接收端。

$y = \text{real}(y) + \text{sqrt}(\text{noise_var}) * \text{randn}(N,1);$

- (4) **訓練資料長度設定**：為了確保 LMS 與 RLS 等化器能進行有效的濾波器係數學習與穩態分析，需提供足夠長度的訓練資料。以 FIR 等化器為例，設定階數為 $M=8$ ，即每次更新需使用長度為 8 的輸入向量。
由於輸入向量為：

$$u[n] = [y[n], y[n-1], \dots, y[n-M+1]]^T$$

需確保在最後一筆更新時仍有完整的 M 個樣本可供存取，因此實際的訓練資料長度需比預定的更新次數 N 多出 M 筆，即：

$$\text{Total Samples} = N + M$$

在經過通道失真與雜訊干擾後，接收訊號將與原始 BPSK 序列產生誤差，此時自適應等化器的任務就是根據已知的訓練序列與接收訊號，動態調整其濾波器係數，嘗試還原出最接近原始資料的輸出。

Step 3: 實作等化器學習演算法

當訓練資料與通道模型建立完成後，接下來要實作自適應等化器的兩種學習演算法。等化器的目的是根據接收到的失真訊號，逐步調整濾波器係數，以便逼近原始未失真的 BPSK 訊號。本次模擬主要在完成兩種學習演算法為 **LMS** 及 **RLS**，以下為各自的實作方式。

LMS (Least Mean Square)

每次根據「現在的誤差」，小小修正等化器權重，讓估計訊號越來越接近真實訊號。更新規則是：

$$w(n+1) = w(n) + \mu e(n)y_n$$

每一次更新新的權重 $w(n+1)$ ，是由原本的權重 $w(n)$ 加上一小步修正量(步長(step size) $\mu \times$ 誤差 $e(n) \times$ 輸入訊號 y_n)，因此選擇一個適合的 μ 很重要，可以在「收斂速度」和「穩定性」之間找到平衡，此次作業我們選擇 $\mu = 0.005$ 。

RLS (Recursive Least Squares)

每次根據「過去所有資料」調整權重，最小化加權誤差平方和，不像 LMS 只管現在這筆誤差，所以 RLS 收斂快。RLS 演算法流程如下：

(1) 計算增益向量 $k(n)$ ：計算要怎麼用現在的輸入，把權重調到更好

$$k(n) = \frac{P(n-1)y_n}{\lambda + y_n^T P(n-1)y_n}$$

λ 是 Forgetting Factor，控制要不要忘掉很久以前的訊息。

(2) 計算誤差 $e(n)$ ：算這次的估計錯多少

$$e(n) = x[n] - w^T(n-1)y_n$$

(3) 更新權重 $w(n)$ ：根據剛剛的 $k(n)$ 和 $e(n)$ 更新權重

$$w(n) = w(n-1) + k(n)e(n)$$

上一個權重 $w(n-1)$ 加上這次的修正量 (增益 \times 誤差)

(4) 更新誤差共變異矩陣 $P(n)$

$$P(n) = \frac{1}{\lambda} [P(n-1) - k(n)y_n^T P(n-1)]$$

λ 越小，表示越容易忘記過去 (反應越快但更容易不穩)， λ 接近 1，表示很重視過去 (學得穩，但適應慢)。

Step 4: 效能評估與誤差分析

完成 LMS 或 RLS 等化器的訓練後，接下來的重點在於評估其學習過程的效果與準確度。常用的評估指標為均方誤差 (Mean Squared Error, MSE)，可以反映出等化器輸出與理想訓練訊號之間的差距程度，MSE 是對每個時間點誤差平方值取平均，計算方式如下：

$$MSE = \frac{1}{N} \sum_{n=1}^N |d(n) - y(n)|^2$$

若 MSE 趨近於 0，表示等化器已成功逼近理想輸出，誤差極小；反之，MSE 越大代表等化器輸出偏離原始訊號較遠。

參、 主要程式與註解：

通道模型介紹：comm.RayleighChannel

comm.RayleighChannel 是用於建立 Rayleigh 衰落通道的物件化函數。Rayleigh 通道常用於模擬 Non-Line-of-Sight 情境下的無線信號傳輸，主要可以模擬的特性如時變 Rayleigh 衰落（包含 Doppler 效應）、多條路徑的延遲、每條路徑可設定平均增益（dB）、支援複數輸入與實數輸出處理、可重複使用與重設。雖然它是內建的黑盒工具，但根據 MATLAB 官方文檔以及我設定的程式碼，我整理成如下：

MATLAB 使用 **Jakes 模型的改進版本**（又稱 Sum-of-Sinusoids model）來產生每個路徑的時變複數係數，數學上表示為：

$$h_k[n] = \sum_{m=1}^{N_0} a_m \cos(2\pi f_D \cos(\theta_m) nT_s + \phi_m) + j \sum_{m=1}^{N_0} \beta_m \sin(2\pi f_D \cos(\theta_m) nT_s + \psi_m)$$

在我們呼叫 chan(x) 時，為每個 path ($l = 0 \sim L - 1$) 生成一條獨立的 Rayleigh 衰落通道 $h_k[n]$ ，每條 $h_k[n]$ 根據 MaximumDopplerShift 產生時間變化(根據 sum-of-sinusoids)，這些 tap 會套用在輸入訊號的延遲版本 $x[n - k]$ ，組合出 $y[n]$ 。

簡單來說就是利用多個不同角度與相位的正弦波加總，產生具時間相關性的複數高斯隨機過程。路徑變化速度受 Doppler 頻移 f_D 控制，當 $f_D = 0$ 時表示靜態通道， $f_D > 0$ 則為時變通道。每個路徑係數再與輸入訊號不同延遲版本相乘並加總，即構成接收端訊號 $y[n]$ 。

LMS (Least Mean Squares) 自適應濾波器:

```
%% %LMS
function mse = lms(runs, mu, N, M, chan, noise_var, delay)
mse = zeros(N,1); %產生一個長度 N 的「全是零的直向向量」。
for i=1:runs
    w=zeros(M,1);
    x = 2 *randi([0 1], N, 1)-1;
    reset(chan);
    y = chan(x);
    y = real(y) + sqrt(noise_var) * randn(N, 1);
    for n = max(M, delay+1):N
        u_n=y(n:-1:n-M+1);
        e=x(n-delay)-w'*u_n;
        w=w+mu*u_n*e;
        mse(n)=mse(n)+abs(e)^2;
    end
end
mse=mse/runs;
end
```

圖五：LMS 計算 mse 函式

一開始會先建立一個長度為 N 的零向量 mse ，用來累積各次模擬的誤差平方。每次模擬中都會先將權重向量 w 初始化為零，接著使用 BPSK 調變產生輸入資料 $x[n]$ ，其中 0 與 1 映射為 ± 1 的符號。並呼叫 $chan(x)$ 通過時變 Rayleigh 通道產生通道響應輸出，再加上加性高斯白雜訊模擬實際接收端訊號 $y[n]$ 。

從第 $n = \max(M, \text{delay} + 1)$ 起，開始進行 LMS 權重更新迴圈，輸入向量提取、目標值抓取、預測誤差計算、權重更新、MSE 累積，每次迭代中，誤差平方會累加到對應時間點的 mse 上。所有模擬執行完畢後，將累積的誤差平方除以總模擬次數，得到最終的平均 MSE 曲線。

RLS (Recursive Least Squares)自適應濾波器:

```
%% %RLS
function mse = rls(runs, delta, lambda, N, M, chan, noise_var, delay)
mse = zeros(N,1);
for i = 1:runs
    w = zeros(M,1);
    P = delta * eye(M);
    x = 2 * randi([0 1], N, 1) - 1;

    reset(chan);
    y = chan(x);
    y = real(y) + sqrt(noise_var) * randn(N, 1);
    for n = max(M, delay+1):N
        u_n = y(n:-1:n-M+1);
        e = x(n - delay) - w' * u_n;
        k = (P * u_n) / (lambda + u_n' * P * u_n);
        w = w + k * e;
        P = (P - k * u_n' * P) / lambda;
        mse(n) = mse(n) + abs(e)^2;
    end
end
mse = mse / runs;
end
```

圖六：RLS 計算 mse 函式

與 LMS 最大的差異在於權重更新的方式與誤差最小化的規則。雖然兩者同樣以最小化輸出誤差平方為目標，LMS 是以當下的誤差來逐步下降(stochastic gradient descent)，而 RLS 則是直接最小化所有歷史資料加權下的總誤差平方和，因此額外需要一個 inverse covariance matrix P 也就是 **RLS (Recursive Least Squares) 演算法**中的第四步驟，根據公式使得系統在每個時間點都能快速且準確地收斂到最佳解。而 LMS 則僅依賴單一的固定學習率 μ 進行簡單的權重微調，總結來說，RLS 比 LMS 多了增益計算、矩陣更新與遺忘因子(λ)的使用。

評估不同 delay 對 LMS 等化器穩態誤差 (MSE) 的影響

```
%% Delay Sweep 測試
delay_range = 3:9;
runs_per_delay = 10;
final_mse = zeros(size(delay_range));
final_std = zeros(size(delay_range));

for i = 1:length(delay_range)
    d = delay_range(i);
    mse_set = zeros(runs_per_delay, 1);

    for r = 1:runs_per_delay
        mse_tmp = lms(runs, mu, N+M, M, chan, noise_var, d);
        mse_set(r) = mean(mse_tmp(end-100:end));
    end

    final_mse(i) = mean(mse_set); % 對多次的結果再取平均
    final_std(i) = std(mse_set);
end
```

圖七：測試 delay 最佳值

為評估 LMS 等化器對 delay 參數的敏感程度，對 delay = 3~9 分別進行 10 次 Monte Carlo 重複模擬，並取其 steady-state MSE 平均與標準差。

在不同 SNR 條件下的位元錯誤率 (BER)

LMS:根據接收到的訊號逐步調整權重，以還原原始輸入訊號，並計算其 BER

```
w_lms = zeros(M,1); % LMS
for n = max(M, delay+1):N
    u_n = u(n:-1:n-M+1);
    y_hat = w_lms' * u_n;
    x_hat = sign(y_hat); % 判決為 ±1
    d = x(n-delay);
    if x_hat ~= d
        bit_errors_LMS = bit_errors_LMS + 1;
    end
    e = d - y_hat;
    w_lms = w_lms + mu * u_n * e;
end
```

圖八：LMS 計算位元錯誤率

在接收端使用 LMS 等化器後，我們根據等化輸出進行 Hard Decision 並與原始資料進行比對，以估算系統的位元錯誤率。首先初始化等化器係數 $W_{LMS} = 0_M$ 表示 M 階的 FIR 線性等化器，也就是初始狀態下未學習任何通道特性。接著從樣本點 $n = \max(M, \text{delay} + 1)$ 起對每筆資料執行以下動作：

1. 接收向量構成：取出當前與前 $M - 1$ 個接收樣本組成向量

$$u[n] = [y[n], y[n-1], \dots, y[n-M+1]]^T$$

2. 估測輸出計算：以內積方式計算估測值

$$\hat{y}[n] = w_{LMS}^T \cdot u[n]$$

3. 判決符號產生：根據符號極性進行 ± 1 判決，可直接透過 `sign()` 函數進行解調判決。

$$\hat{x}[n] = \text{sign}(\hat{y}[n])$$

將 `y_hat` 轉成二元判決 (± 1)，規則如下：

y_hat 數值區間	sign(y_hat) 回傳值	判決為 bit	BPSK 對應符號
y_hat > 0	+1	1	+1
y_hat = 0	0	隨機決定(幾乎不發生)	隨機決定(幾乎不發生)
y_hat < 0	-1	0	-1

4. 錯誤判斷與計數：將判決結果與真實資料 $d[n] = x[n - \text{delay}]$ 比較，若不同則錯誤次數加一。

5. LMS 權重更新：計算誤差並依 LMS 演算法更新權重

$$e[n] = d[n] - \hat{y}[n]$$

$$w_{LMS} \leftarrow w_{LMS} + \mu \cdot u[n] \cdot e[n]$$

完成後即可取得 `bit_errors_LMS`，除以總位元數 $N-M+1$ 得到 LMS 系統的 BER，用以評估其偵測準確性與通道等化能力。

RLS:透過綜合過去所有樣本的誤差資訊，即時更新權重，以更快速且精確地還原原始訊號並計算 BER。

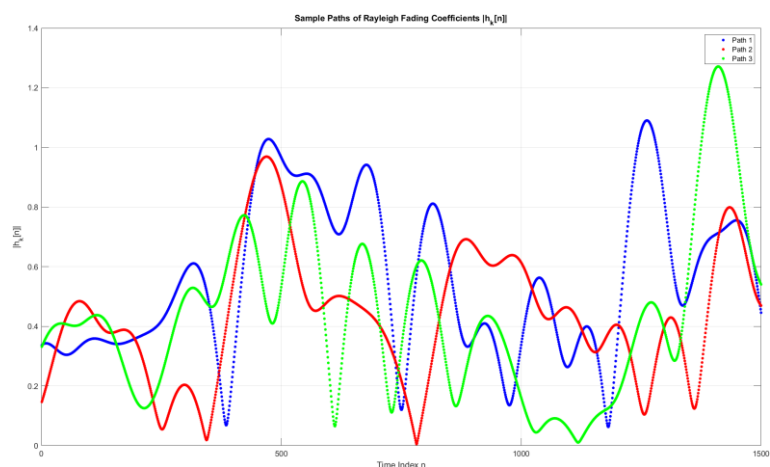
```
w_rls = zeros(M,1); % RLS
P = delta * eye(M);
for n = max(M, delay+1):N
    u_n = u(n:-1:n-M+1);
    y_hat = w_rls' * u_n;
    x_hat = sign(y_hat);
    d = x(n-delay);
    if x_hat ~= d
        bit_errors_RLS = bit_errors_RLS + 1;
    end
    e = d - y_hat;
    k = P*u_n / (lambda + u_n'*P*u_n);
    w_rls = w_rls + k * e;
    P = (P - k*u_n'*P) / lambda;
end
```

圖九：RLS 計算位元錯誤率

在每個時間點，RLS 根據輸入信號 u 與目標訊號 x ，計算預測誤差並即時更新等化器權重 w_{rls} ，使整體預測誤差最小。不同於 LMS 僅考慮目前這筆資料的誤差，RLS 會綜合過去所有樣本進行加權調整。為了控制更新強度，RLS 使用誤差協方差矩陣 P 來動態調整每次的權重增益 k ：當模型不確定性高時，有較大調整；而當系統逐漸收斂，則自動降低更新幅度。預測結果 y_{hat} 經由 `sign()` 函數轉換為 ± 1 ，用以判斷接收位元，並與原始訊號進行比較，以計算最終的 BER。

肆、實作執行與結果討論：

1. 基本訊號觀察：Rayleigh 衰落通道與 AWGN 雜訊樣本

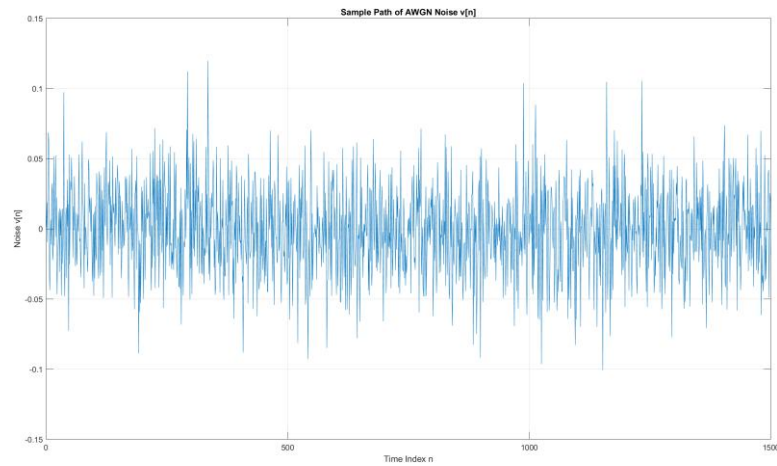


圖十：Rayleigh 衰落係數 $|h_k[n]|$ 的瞬時振幅變化

圖十顯示三條獨立路徑下，Rayleigh 通道中複數衰落係數 $h_k[n]$ 的瞬時幅度變化，其絕對值由下式計算：

$$|h_k[n]| = \sqrt{\text{Re}(|h_k[n]|)^2 + \text{Im}(|h_k[n]|)^2}$$

代表的是第 k 條多徑通道在第 n 個時間點的瞬時衰落增益 (gain)，圖中的 3 條多徑中每條路徑分別由獨立的 Rayleigh 隨機過程產生。利用 `stem` 函數畫出它離散時間的通道係數幅度。可看出這些通道具有隨時間緩慢變動的隨機性，反映出多徑傳輸中因 Doppler 效應造成的時變衰落現象。

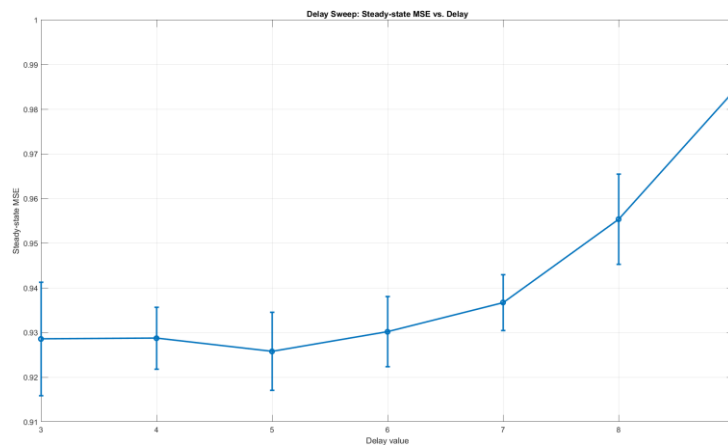


圖十一：AWGN $v[n]$ 的樣本路徑

圖十一為加性高斯白雜訊 $v[n]$ 的時間序列圖，可觀察出其平均值約為 0，呈現對稱分佈，符合 Gaussian 分布的統計特性。振幅在 ± 0.15 以內隨機分布（對應設定的雜訊變異數 $\text{noise_var} = 0.001$ ），整體雜訊分布隨時間指數 n 呈現隨機震盪、無明顯趨勢，具備「白雜訊」的頻率特性（各頻率分量能量平均）。

2. 參數影響分析:分析 LMS/RLS 演算法對初始化參數的反應

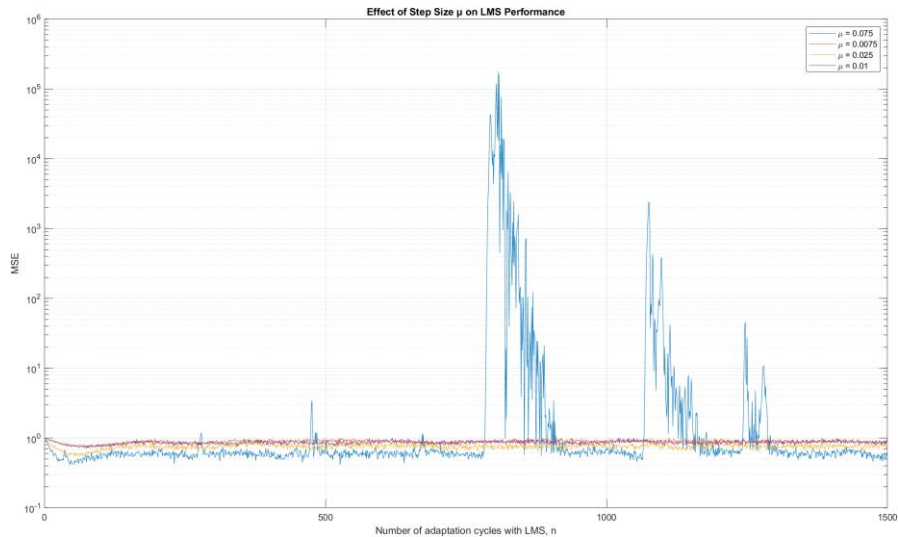
延遲參數調整對 LMS 性能影響



圖十二：在不同 Delay 設定下的穩態 MSE 表現

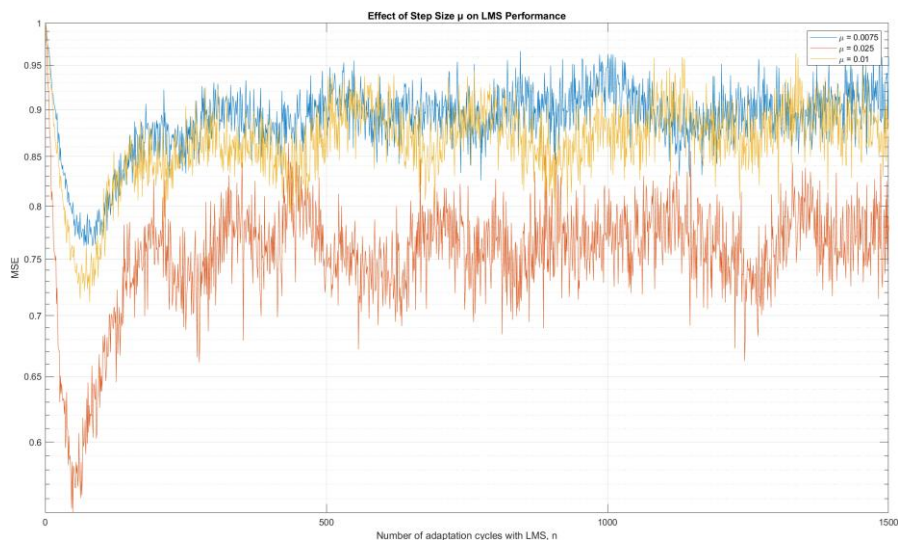
圖十七中誤差棒反映各 delay 結果的穩定性，可以觀察到當 delay 為 5 時，MSE 最低且誤差棒較短，顯示此為較佳選擇。整體結果說明延遲參數會顯著影響 LMS 等化器的效能，因此透過參數 sweep 可得知 delay = 5 為最適值，後續模擬亦以此作為固定設定。

μ 大小影響收斂速度與震盪



圖十三：不同 μ 對 LMS 演算法學習的 MSE 曲線比較圖

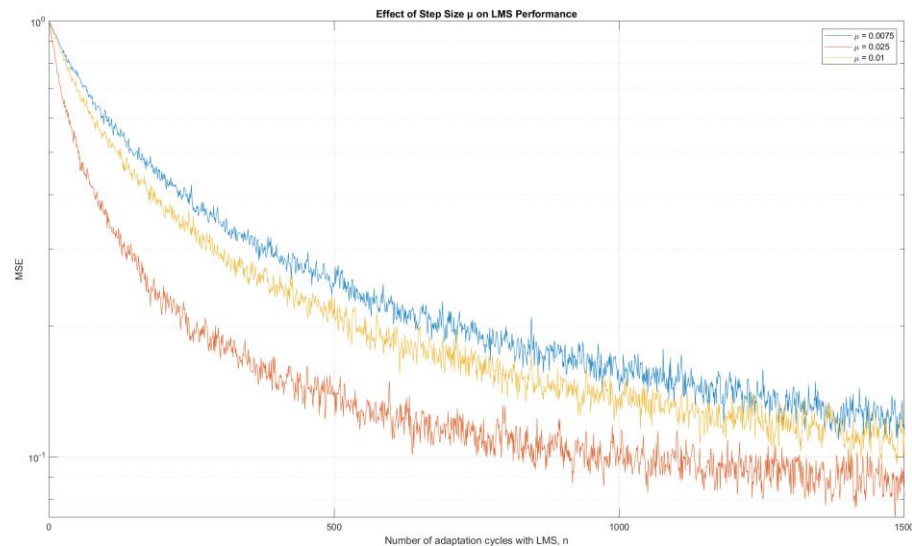
圖十三可以觀察到當 Step Size μ 為 0.075 時在 Rayleigh 通道有發散現象，雖然前期收斂速度最快最低，但在學習後期出現嚴重震盪甚至發散（MSE 突然升高至 10^5 以上）顯示步長過大造成系統不穩。為了更仔細看其他三者收斂情況，我接著把藍線(0.075)作註解，其餘三線輸出如下圖。



圖十四：不同 μ 對 LMS 演算法學習效果之 MSE 曲線比較圖

在三條不同步長 μ 的 LMS 曲線中， $\mu = 0.025$ （橘色）展現出最佳的整體效能，其 MSE 在初期下降速度快，並一度達到最低值。這是因為 LMS 在初始

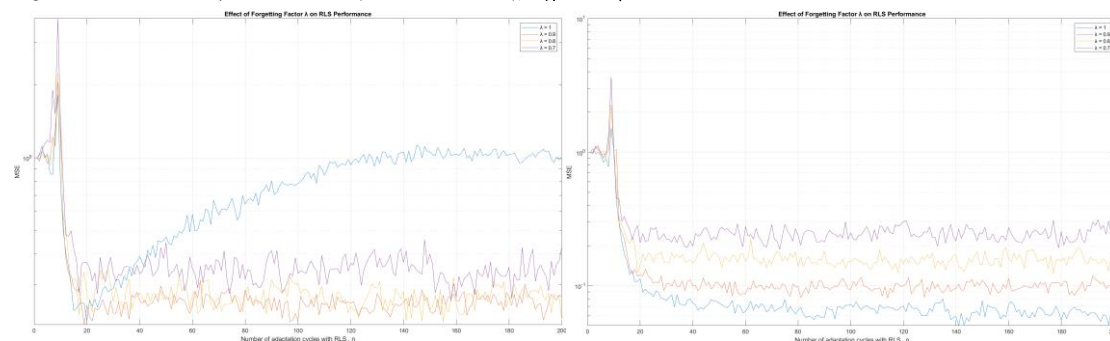
階段能迅速學習並對應當下的通道狀態。然而，由於 Rayleigh 通道具有時間變動性，通道係數會持續受到 Doppler 效應影響而改變，導致早期學習到的權重逐漸失效。若 LMS 的步長 μ 設定過大，等化器將難以即時跟上通道變化，最終造成誤差重新上升。因此，在 Rayleigh 通道環境中，LMS 誤差曲線常出現「先下降再回升」的特性，反映出其對動態通道的追蹤能力有限。



圖十五：不同 μ 在靜態通道對 LMS 演算法學習效果之 MSE 曲線比較圖

為了觀察若通道為靜態時 LMS 的收斂情況，我將 f_D 改為 0。圖十五可發現權重能穩定地收斂到固定最小 MSE 的解，因為環境不再變化，權重也不需要持續調整，因此誤差只會單調下降並趨於穩定，且 $\mu = 0.025$ 在靜態中也是表現最為平衡，穩態誤差達到最小值。

遺忘因子 λ 對 RLS 演算法效能的影響分析



圖十六：Rayleigh 通道下 RLS 不同 λ 值之 MSE 表現比較圖(左)

圖十七：靜態通道下 RLS 不同 λ 值之 MSE 表現比較圖(右)

圖十六為 RLS 演算法在 Rayleigh 動態通道環境下，不同遺忘因子 λ 的 MSE 表現曲線。由於 $\lambda = 1$ 時無法適當「捨棄過時資訊」，導致等化器無法及時追蹤當前通道的變化，因此系統誤差會逐漸累積，最終導致 MSE 回升甚至高於初期的收斂值。而較小的 λ 如 0.9(橘)、0.8(黃)、0.7(紫)則代表系統有能

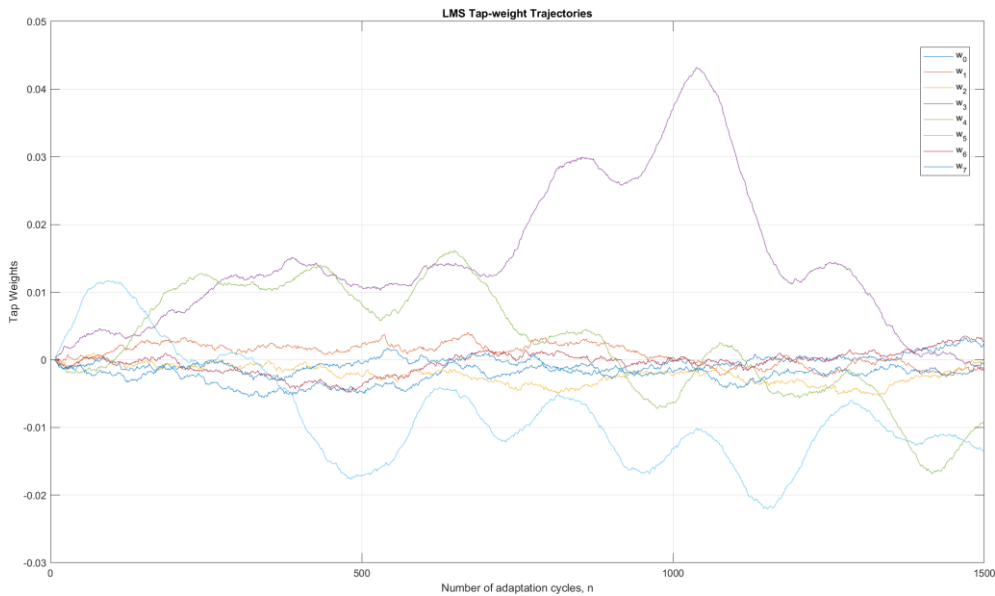
力漸進捨棄舊資訊，保留較近期的資料進行學習，因此能夠更靈活地追蹤通道變化，維持穩定且較低的 MSE。

而圖十七為 RLS 在靜態通道($f_D = 0$)下的學習表現，此時通道固定不變， λ 越大代表越能記住過去的資料，因此 $\lambda = 1$ 時收斂最穩定、誤差最低；而 $\lambda = 0.9、0.8、0.7$ 依序誤差上升，因為快速遺忘造成收斂速度快但誤差大、波動明顯，顯示在靜態環境下較不穩定。

3. 演算法整體性能比較

LMS 權重的逐步收斂行為

透過 runs 次蒙地卡羅模擬平均 MSE，並在每個時間點 n 累加 LMS 演算法在每次模擬中的 tap-weight 參數 $w = [w_0, w_1, \dots, w_{M-1}]^T$ 最終取平均後存進 W_record 矩陣中，表示每階 tap-weight 隨時間的平均演化軌跡。為了評估 LMS 演算法收斂後 tap weights 是否落在合理範圍內，我使用 `disp(max(abs(W_record(:))))` 將紀錄的權重矩陣展平成一維向量，計算每個 tap weights 的絕對值，並輸出整體權重軌跡中絕對值最大的那一項，藉此可快速判斷是否出現數值異常（如發散）。繪製出如下圖：

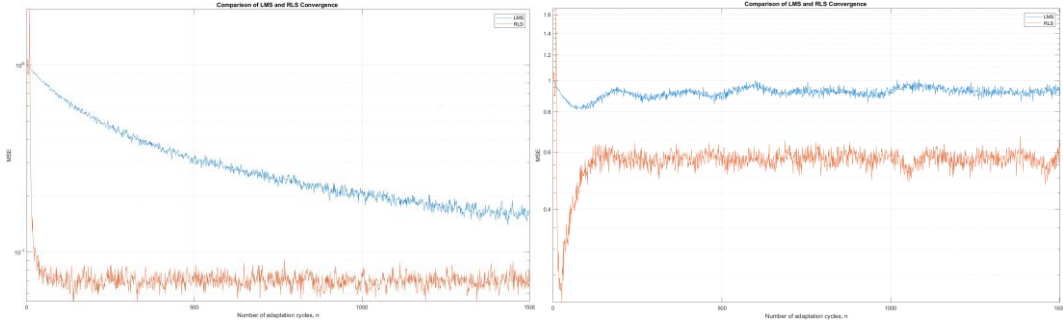


圖十八：在 Rayleigh 通道中 LMS 中各 tap-weight 的收斂軌跡

圖十八顯示 LMS 演算法在訓練過程中，每一階濾波器權重 $w_0 \sim w_7$ 隨時間的演化軌跡。因為初始設定 $M = 8$ ，表示 LMS 等化器為一個長度為 8 的 FIR 濾波器，具有 8 個 tap weight，分別對應於濾波器中的每一階延遲單元的權重。在多徑 Rayleigh 通道與雜訊條件下，各 tap weight 並未收斂至穩定值，而是持續擺動，反映出 LMS 對時變通道的響應與敏感性。整體權重維持在接近 0 的小範圍內變動，顯示 LMS 嘗試追蹤通道變化但穩定度有限。

整體 MSE 收斂速度與震盪對比

接下來直接比較 LMS (藍) 與 RLS (橘) 兩種自適應演算法在學習過程中的收斂行為與誤差表現。主要分成兩層面比較，分別是收斂速度與穩態誤差，並分別呈現等化器在 Rayleigh 時變通道與靜態通道環境下的 MSE 收斂行為。



圖十九：靜態通道下 LMS (藍) 與 RLS (紅) MSE 比較圖(左)

圖二十：Rayleigh 通道下 LMS (藍) 與 RLS (紅) MSE 比較圖(右)

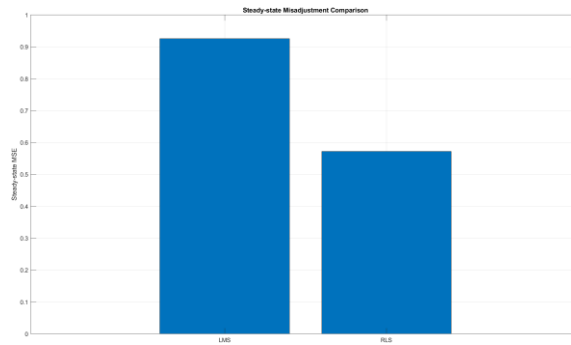
由兩圖可觀察到 RLS 在兩種通道下皆展現極佳的學習性能與穩定性。在靜態通道中 (圖十九)，RLS 的 MSE 極速下降並收斂至約 0.05 以下，曲線波動幅度小；而 LMS 雖具備單調下降趨勢，但收斂速度顯著較慢，最終穩定於約 0.2 附近。

相較之下，Rayleigh 通道 (圖二十) 因具有隨機時間變動的特性，導致 LMS 的學習結果出現「先下降再上升」的現象。RLS 穩態 MSE 約在 0.55 ~ 0.6 之間，LMS 穩態 MSE 約在 0.9 左右。兩者在 Rayleigh 通道變動條件下，在初期因為通道係數尚未明顯變化，LMS 與 RLS 能夠有效利用訓練資料來逼近理想權重，因此 MSE 快速下降。但 Rayleigh 通道的每一條路徑係數是隨時間變化的，會根據 Doppler 效應產生衰落與變化。當通道改變後，原本學到的濾波器不再適用。此時新資料與舊通道特性不匹配，導致誤差重新上升。

整體而言，RLS 無論在靜態或時變通道下，皆優於 LMS。RLS 擁有更快的收斂速度、更低的穩態誤差與更強的通道適應能力，證實其在動態通訊環境中為更具彈性的等化選擇。

穩態誤差定量比較

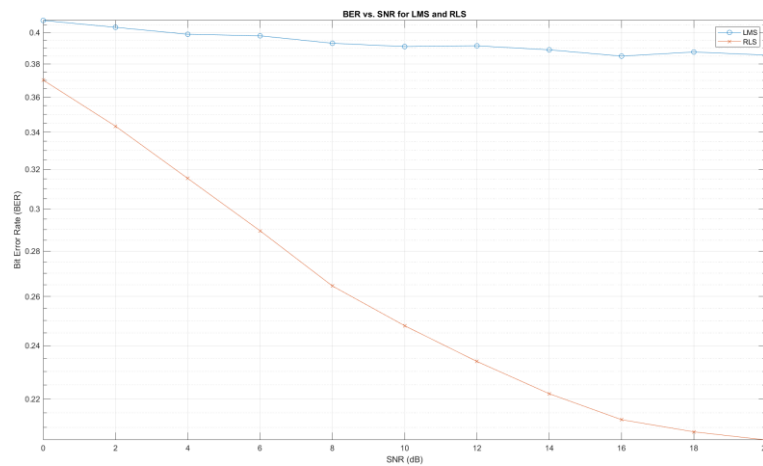
在觀察完 LMS 與 RLS 的整體收斂趨勢後，我們進一步比對兩者在 Rayleigh 通道中長時間運作後的穩態誤差 (steady-state misadjustment) 進行定量比較。我取學習週期第 300 次至第 500 次期間的 MSE 平均值作為穩態誤差指標。而 RLS 演算法的數據我僅採用 $\lambda = 0.98$ ，以作為標準比較依據。



圖二十一：LMS 與 RLS 穩態誤差柱狀圖

LMS 的穩態誤差約為 0.926，RLS 的穩態誤差則僅 0.572，RLS 在穩態區間內的 MSE 顯著低於 LMS，驗證其在誤差控制與準確性方面具備優勢，與前述收斂趨勢圖的觀察結果一致。

4. 通訊效能指標—最終 BER 表現



圖二十二：兩等化器在不同 SNR 條件下的位元錯誤率比較

在 BER 對 SNR 的比較圖中，RLS 表現出明顯優勢。隨著 SNR 提升，RLS 的 BER 持續下降，從約 0.37 降至 0.21，顯示其對於高 SNR 環境具備更佳的抗雜訊能力與等化效能。反觀 LMS，BER 幾乎維持在 0.39 左右，變化幅度極小，顯示其在 Rayleigh 時變通道中的適應性不足，導致無法有效降低錯誤率。

伍、 挑戰、限制、潛在改進方向：

(1) 參數設定敏感度高

無論是 LMS 的步進尺寸 μ ，或是 RLS 的遺忘因子 λ ，都對演算法的穩定性與效能有高度影響。不當的參數設定容易導致收斂不良或誤差震盪，這次模擬的參數是藉由慢慢調整，需要透過多次模擬與經驗調整才能獲得最佳效能。

(2) 初始條件影響明顯

特別是在 LMS 中，初始權重若設定不合理，容易導致學習初期 MSE 出現明顯震盪，甚至短暫發散，造成學習時間延長與收斂品質下降。

(3) RLS 計算成本高

雖然 RLS 的效能很好，但它需要比較多的記憶體和運算資源，如果要把它實作在硬體或即時系統上，會比較麻煩，可能會遇到速度慢或佔用太多資源的問題。

陸、 重點發現與結論摘要：

關於 LMS 等化器

(1) 在 Rayleigh 通道中：

- 初期誤差下降快速，但隨時間通道變動，誤差會再次上升。
- 穩態誤差較大，收斂不穩定，對 Doppler 敏感。

(2) Delay sweep 結果顯示，當 $\text{delay} = 5$ 時，平均 MSE 最低且誤差穩定，為推薦設置。

(3) 步進參數 μ 對性能影響顯著，需根據通道特性調整。

(4) 優點：演算法結構簡單、計算量低，實用性高，適合實作於運算資源有限或通道較穩定的應用情境。

關於 RLS 等化器

(1) RLS 能即時追蹤通道變化，適用於動態環境。

(2) 穩態誤差明顯低於 LMS，且 BER 下降更迅速。

(3) 遺忘因子 λ 決定系統記憶長度：

- 在靜態通道中： λ 越接近 1 則越重視歷史資訊，系統記憶越久，能完整保留通道特性，誤差最小且最穩定。 λ 越小雖然提升反應速度，但也會因為系統過度遺忘歷史資料導致穩態誤差上升。
- 在 Rayleigh 動態通道中：通道特性隨 Doppler 效應變動，歷史資料迅速失效。 $\lambda = 1$ 時無法捨棄過時資訊，導致誤差逐漸升高。適當降低 λ (如 0.9) 可幫助系統及時反應通道變化，穩態誤差下降，適應性提高。

(4) 誤差共變異矩陣 $P(n)$ 的影響：雖然 RLS 穩定性較高，但若初始權重或協方差矩陣設定不當，仍會影響收斂初期的學習行為與 MSE 表現。

總結

LMS 適合靜態或變化緩慢之通道，實作簡便但穩定性有限。RLS 則在穩定性、收斂速度與 BER 效能方面均優於 LMS。綜合考量，RLS 為高效能需求系統之首選，而 LMS 可做為低資源系統之替代方案。