

多輸入多輸出無線通訊系統偵測器之 FPGA 設計與實作

FPGA Design and Implementation of MIMO Detectors for Wireless Communications

陳柏任, 周沛毅

指導老師:黃元豪教授

ABSTRACT

因應現代人對於通訊流量的需求，MIMO 技術因此被發展出來並且被廣泛地用於現代通訊設備。因為相較於過去在傳送端和接收端各只有一個天線的通訊系統，MIMO 系統藉由兩端更多的天線數，提高通訊系統頻譜使用效率，以及提升資料吞吐量，提升通訊系統流量。這個專題主要就在設計 MIMO 無線通訊系統，大致上流程為先在 C/C++ 上模擬 MIMO 的功能與表現，再將 C/C++ 上的演算法轉換成 Verilog code，並且燒入至 FPGA 驗證模擬結果。

Introduction

所有通訊系統皆由三者組成，傳送端、通道以及接收端。MIMO 不同於其他通訊系統的地方在於其傳送端和接收端皆由多根天線組成，以下為 MIMO 數學表示法：

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \dots (1)$$

向量 $\mathbf{x}$ 代表從傳送端傳送的訊號，維度為 M ；向量 $\mathbf{y}$ 代表接收端接收到的訊號，維度為 N 。 $\mathbf{H}$ 為 $N \times M$ 矩陣， $\mathbf{H}$ 矩陣中的某一個元素表示出某個傳送端的天線至某個接收端的天線之間通道的增益(gain)和相位變化效應，向量 $\mathbf{n}$ 則表示在 MIMO 系統中的雜訊。在本次專題中，我們採用 $M=N=4$ 的規格實作。

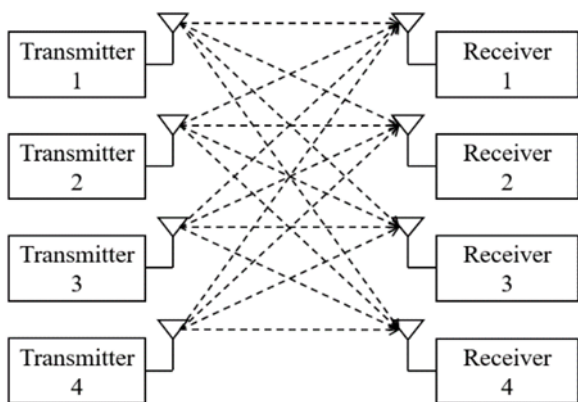


Fig 1.MIMO 示意圖

1. MIMO preprocessor

MIMO detector 演算法中，有些演算法需要 SQRD(sorted QR decomposition)，而我們專題中的兩個演算法皆需要使用 SQRD。通常 MIMO 接收端會先利用 data-aided 的方式將 $\mathbf{H}$ 矩陣值偵測出來，再將 $\mathbf{H}$ 矩陣經由繁雜的矩陣運算求出 $\mathbf{H}=\mathbf{Q}\mathbf{R}$ ， $\mathbf{Q}$ 為 orthonormal matrix，也可以說是旋轉矩陣，而 $\mathbf{R}$ 則為上三角矩陣，且 $\mathbf{H}$ 的 column 有另外排序，排序順序紀錄在一個向量上方便最後將排序順序調整回來。

2. MIMO detector

MIMO detector 主要目的為利用 MIMO Preprocessor 輸出的 $\mathbf{Q}$ 、 $\mathbf{R}$ 兩矩陣資訊，與天線所收到的 complex envelope，進行一系列的計算，進而推估傳送端最有可能傳送的 symbol 值。Euclidean distance，以下簡稱 ED，即是在 ML decision rule 中，signal 與 symbol 的距離。距離相對越小，解出的 symbol 為 signal 的原始答案相對機率越高。在有使用 SQRD 的 MIMO 通訊系統中，ED 的公式如下：

$$d(\bar{s}) = \left\| \begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{bmatrix} - \begin{bmatrix} R_{00} & R_{01} & R_{02} & R_{03} \\ 0 & R_{11} & R_{12} & R_{13} \\ 0 & 0 & R_{22} & R_{23} \\ 0 & 0 & 0 & R_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_0 \\ \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} \right\|^2 \dots (2)$$
$$= \sum_{i=0}^3 \left| \hat{y}_i - \sum_{j=i}^3 R_{ij} \hat{x}_j \right|^2 \dots (3)$$

公式(2)、(3)分別為 ED 的向量表式法和化簡成一般算式的表示法。由公式(2)可知，ED 的定義為接收端收到的向量 $\mathbf{y}$ 與 $\mathbf{Q}^H$ 旋轉矩陣相乘後的向量 $\hat{\mathbf{y}}$ 與某一個解出的 symbol 的 $\hat{\mathbf{x}}$ 向量和 $\mathbf{R}$ 矩陣相乘之後的向量之距離。當 $\hat{\mathbf{y}}$ 完全等於某一個 symbol 的 $\mathbf{R}\mathbf{x}$ 的時候，因為兩項兩邊同乘 $\mathbf{Q}$ 矩陣的話，左邊那項因為 $\mathbf{Q}$ 矩陣為 orthonormal 矩陣，符合矩陣等式 $\mathbf{Q}\mathbf{Q}^H = \mathbf{I}$ ，所以最後會變成接收端接收到的 $\mathbf{y}$ 向量，即 $\mathbf{Q}(\mathbf{Q}^H\mathbf{y}) = \mathbf{y}$ 。至於右邊項，因為 $\mathbf{Q}\mathbf{R} = \mathbf{H}$ (如果 QR 分解有 sort 就是交換 column 順序的 $\mathbf{H}$)，所以最後右邊項會變成 $\mathbf{Q}(\mathbf{R}\mathbf{x}) = \mathbf{H}\mathbf{x}$ ，也就是沒有 noise 情況下的 $\mathbf{y}$ 值。基於 ML decision rule，當 $\|\hat{\mathbf{y}} - \mathbf{R}\hat{\mathbf{x}}\|^2 = \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|^2$ 越小時， $\mathbf{y}$ 與 $\mathbf{H}\hat{\mathbf{x}}$ 的距離就越小，noise 在向量空間的長度就越小，解出的 $\hat{\mathbf{x}}$ 所代表的 symbol 值越有可能是一開始傳送端傳送的 symbol 值。

在公式(3)中，每一個平方項為 partial Euclidean distance，簡稱 PED。PED 為 ED 的一部分。有了上述的 SQRD 將 $\mathbf{H}$ 矩陣分解成 $\mathbf{Q}\mathbf{R}$ 矩陣之後，我們才可以在每次算完一個 PED 之後，依據當前累積 ED 大小排序，馬上決定出一個 $\mathbf{x}$ 向量的 element 而不受其他尚未決定的 element 值影響。

FSD 為 Fixed complexity sphere decoder 的縮寫，它的作法如同 K-best detector，會將一個 $\hat{\mathbf{x}}$ 向量拆成它的分量 $(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)^T$ ，每個分量個別處理。它在解分量的時候，會先列舉幾個候選的 constellation point 當作前幾個分量的值，然後用 SIC 的方法，找出 $\hat{\mathbf{x}}$ 向量中剩餘的分量。最後我們得到許多可能為正確解的 $\hat{\mathbf{x}}$ 向量，算出這些 $\hat{\mathbf{x}}$ 的 PED 後，取 ED 為最小的 $\hat{\mathbf{x}}$ 當作 detection 的最終結果。舉例來說，一個 4×4 的 MIMO 系統，調變方式是 QPSK，則 $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4)^T$ ，且 constellation point 有四個點。FSD 會從 $\hat{x}_4$ 開始解，接著解 $\hat{x}_3, \hat{x}_2, \hat{x}_1$ ，如下圖。

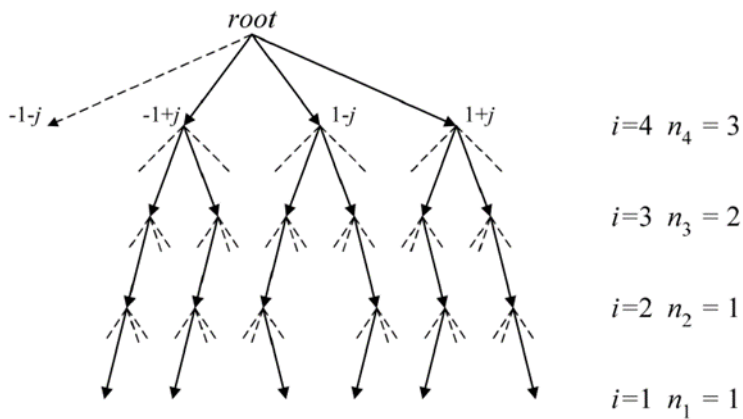


Fig 2. FSD 流程示意圖

圖中的 (n_1, n_2, n_3, n_4) 為每一層展開的分支數。每一層展開分支的方法為，用 SIC 找出 $\hat{x}_i$ 一個解，再用 zig-zag enumeration 列舉出這個解附近的 constellation point，視 n_i 的值列舉出相對應數量的 $\hat{x}_i$ 。此圖為 $(n_1, n_2, n_3, n_4) = (1, 1, 2, 3)$ 的一個例子，圖中的 root 分支出三條路徑，所以第一層列舉了3個 $\hat{x}_4$ ，每個 $\hat{x}_4$ 再分支2條路徑，所以每個 $\hat{x}_4$ 會列舉2個 $\hat{x}_3$ ，一共6個 $\hat{x}_3$ 在第二層。依此類推，解完4個 $\hat{x}$ 的分量以後，得到1·1·2·3=6組 $\hat{x}$ 可能的解，再選出 ED 最小者當作 $\hat{x}$ 的解。而在本專題中，調變技術採用16-QAM，一層分支數最多16條，而每層分支數為 $(n_1, n_2, n_3, n_4) = (1, 1, 1, 16)$ 。

K-Best 演算法的核心精神為在計算每一層 PED 與累積 ED 之後，會經由大量的比較運算，將當下具最小 PED 的 K 條路徑比較出來，並且下一層所有可能的路徑皆由這 K 條路徑延伸。每一層反覆做相同的運作，直到最後一層，將最佳路徑解出來，即可得出最可能的 symbol 組合。以下先採用在向量空間中只有4個 signal point 的 QPSK 調變方式，以及 K = 2 的情況下 K-Best 演算法的運作模式。

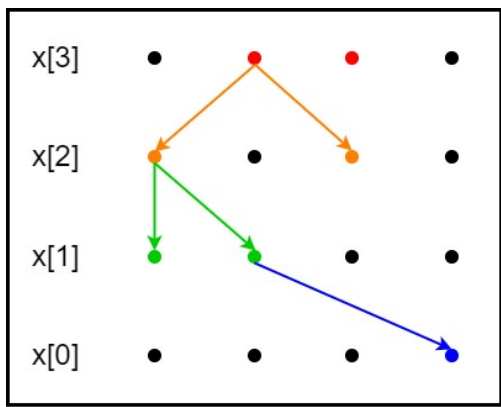


Fig 3. K-Best 流程示意圖

K-Best 的流程為由上至下，左邊的 element 代表在那一層所決定的 element，每一層的每一個點代表每一種 element 的選擇。不同於其他層，因為第一層沒有前面 K 條最佳路徑，所以在第一層只要從4個 element 的選擇選出最佳的 K 個 element(即上圖中兩個紅點)，再由這 K 個點延伸出 $K \times M$ 條可能路徑(M 為調變方式的訊號點個數，QP SK 的 $M = 4$)。之後從 $K \times M$ 條路徑中選出累積 PED 最小的 K 條路徑(即圖中兩條橘色線段)，剩下的路徑在 K-Best 後面就被捨棄掉。在第二層到倒數第二層都是從已知的 K 條路徑延伸出 $K \times M$ 種可能路徑，並且從可能路徑選出累積 PED 最小的 K 條路徑。最後一層也是先列出 $K \times M$ 條可能的路徑，並且計算所有路徑的累積 PED，但是最後一層不同的點在於只需要找出具有最小累積 PED 的路徑就好，不需要找 K 個最佳解，因為在最後一層最佳路徑即為最後輸出的決定結果(即圖中藍色路徑)。

3. Simulation Result

本次專題流程為先在 C/C++ 上模擬 MIMO 的功能與表現，再將 C/C++ 上的 MIMO detector 設計成硬體架構，轉換成 Verilog code，並且燒入至 FPGA 驗證模擬結果，以下為 FPGA 上模擬出的 BER curve。

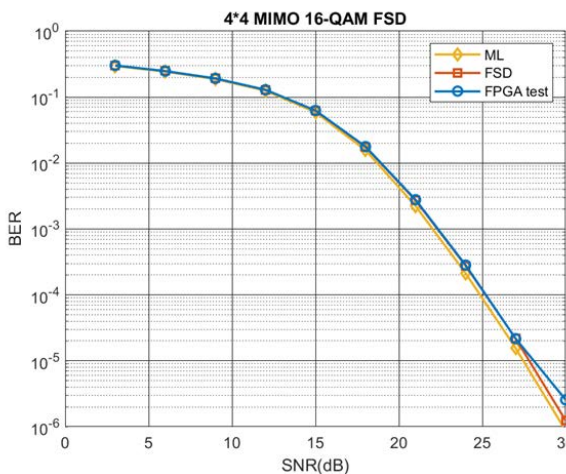


Fig 4(a). FSD 的硬體模擬結果

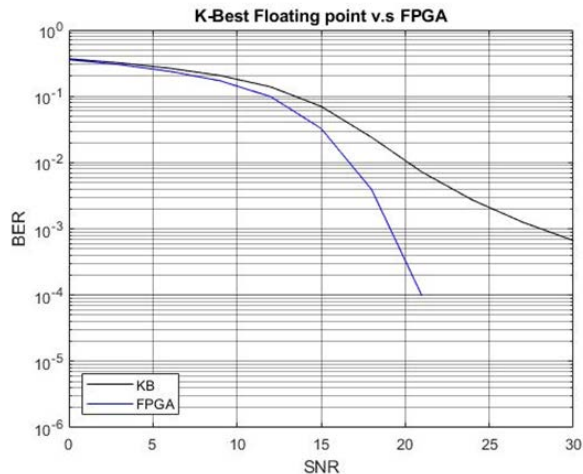


Fig 4(b). K-Best 的硬體模擬結果



Fig 5. FPGA Demo 照片

心得感想

陳柏任：

去年暑假剛開始讀 MIMO detection 演算法的課本和講義的時候，對於公式還沒有深刻的理解，只知道它符合數學的運算和定理，所以求出來的答案就會對。去年8月底到12月底之間，我把講義裡所有 MIMO detection 的演算法都打成 C 語言的程式來模擬通訊系統。此時我才發現公式中隱藏很多細節，若沒有清楚理解公式背後的數學原理或是適用條件，很容易就會誤解公式而打出一個錯誤的程式。打完程式以後我對 MIMO detection 的運作情形了解得更透徹。而且打 C 語言的程式還有許多好處，一是可以練習基本的資料結構，像是 tree 之類的，尤其適用於 K-Best detector 和 Sphere decoder。二是可以練習用數值方法解數學問題，例如：矩陣運算、求反矩陣、解線性方程式、用迭代法逼近正確答案等等。三是它的語法和構造很基本，演算法的細節都必須自己親手寫成程式，這樣有利於把 C 語言改寫成 Verilog code。四是 C 語言的運算速度很快，在跑大量資料模擬通訊系統的時候，比較不耗費太多時間。打完 MIMO detection 演算法並模擬成功一刻是我最開心的時候，代表我又多學會一個演算法並親手實踐。去年12月到今年1月是從軟體模擬過渡到硬體實作的階段，在段時期我學會一個重要的觀念，就是 fixed-point simulation。這是我們為了用最少的硬體資源去達到一個可接受的運算精確度而做的模擬。做完 fixed-point simulation 以後，硬體上各個訊號的位元數就定下來了。今年2月底到3月初，我們畫 block diagram，設計系統架構，此時我學到一些 block diagram 要注意到的小細節，並整理思路，釐清演算法中的各種運算要如何用電路實現。3月到5月我們打 Verilog code，並燒錄到 FPGA 做測試。過程中遇到一些電路的問題但也一一解決了，我們也學會操控 FPGA，像是燒錄程式、輸入波形到 FPGA、從 FPGA 讀取輸出波形等等。最後我們測出用 FPGA 跑 MIMO detection 的 BER 曲線，與軟體模擬的結果吻合，兩學期的實作專題大功告成。感謝老師每個禮拜和我們 meeting，指導專題內容和報告方法，關心我們的專題進展並給予我們幫助。我也謝謝同組的成員，每個禮拜 meeting 完都和我交流意見，給我很多靈感。做 FPGA 實驗和撰寫專題報告時，有成員共同努力真的是很大的助力。我學完兩個學期的 MIMO detection 專題獲得很多收穫，不論是課本上的知識、報告和研究的方法、寫出演算法的 C 和 Verilog code、操作 FPGA，這些都是非常珍貴的經驗，很難在其他課有一個完整的流程把一個學問學到透徹。

周沛毅：

印象還很深刻，在專題的一開始，我其實連為什麼通訊數學模型會用複數形式表現都不太清楚。不過因為上學期有修通一，在通一的第二章最後部分教 I/Q 收發機原理之後，自己更清楚通訊數學模型上複數背後的物理意義，也更知道為什麼電機系要安排我們學複變函數，讓我們更熟悉複數和 sin 與 cos 之間的關係。除此之外，上學期還有一個令我印象深刻的觀念，就是通訊上能量的定義。記得那時候看到教授提供的原文書上面 SNR 的公式內有變異數，那時候只知道原文書這樣寫，所以先硬吞這個定義。直到有一次通一上課有講到「變異數其實就是能量」這個觀念，還有下學期通二的第一章直接開宗明義清楚定義通訊上能量的定義式，我才對上述的 SNR 公式更有感覺。通訊是一個非常注重觀念的科目，而在這一次專題中，我在觀念上有不少收穫。

再來是軟體方面，二下才剛修完資料結構，結果馬上在專題上學期發現 K-Best 和 Sphere decoder 兩個演算法可以運用 C++ 的 tree 來寫，因為資料結構那時候只有練習一次，所以自己其實沒有很有把握可以駕馭 tree，因此在上學期打 C++ 的時候，剛好可以練習 tree。果然如我所料，在打 code 的途中確實有遇到問題，不過後來也知道問題成因，並且順利解決。這是在打 C++ 階段最令人印象的部分。

至於硬體方面，距離上一次打 Verilog 已經一年半了，所以一開始有點忘記語法，不過看一下大一下邏設實驗的作業之後就想起來怎麼打。不同於邏設實驗，這次專題 code 的 loading 大非常多，像是最後沒有放在報告裡的 SQRD，其實花費兩位組員相當大的心力，不論是設計複雜度還是 code 行數都相當有挑戰性，不過後來也順利完成。

最後，首先要感謝教授在上學期每個禮拜五和下學期每個禮拜四都撥出一小時的時間殷勤地指導我們專題，我們都從中學到不少額外補充的知識。再來要感謝隊友陳柏任，在我遇到瓶頸的時候和我討論，以及和我分享一些數學推導式子，能和你同組專題真的很幸運。還有在實驗室幫助過我們的學長們，這段時間真的麻煩你們了！