Implementation and Improvement of Low-Density Parity-Check Codes with Quadrature Amplitude Modulation

低密度奇偶校驗碼之編解碼器實現與改善

組別：A291　　組員：林峻霆　　指導教授：趙啟超 教授

報告摘要

低密度奇偶校驗碼(Low-Density Parity-Check Code， LDPC code)是一種錯誤更正碼，目的在於更正資料因雜訊而產生的錯誤，進而降低錯誤率。在本次專題中，我實作 LDPC Code 的編碼器與解碼器(encode and decoder)，了解 LDPC code 的原理以及解碼方式(Sum-Product Algorithm，SPA)，並對不同的 code 進行比較。

在現代的通訊技術中，正交振幅調變(Quadrature Amplitude Modulation，QAM)被廣泛使用。我也將 LDPC code 與 QAM 進行搭配，探討不同的星座圖(Constellation)下對解碼效果的影響。

在本次的專題中，我使用 C 語言模擬通道傳輸的情況，對收到的訊號進行解碼，接著將解碼成果與傳輸資訊進行比較計算錯誤率，最後使用 MATLAB 對不同訊噪比(Signal-to-Noise Ratio，SNR)下的位元錯誤率(bit error rate，BER)進行繪圖。

經過實驗結果可以發現，在使用 QAM 進行調變時，若採用 non-Gray Mapping 的星座圖，同一個 symbol 間各個位元的統計獨立特性會被破壞，進而導致 SPA 的解碼效果變差。在此情況下，我們應採用修正過的解碼方法：CMBP，來解決此問題，進一步降低錯誤率。

報告內容

(一) 前言

在現代的通訊系統以及資料儲存系統中，往往會加入錯誤更正來確保資料的正確性。低密度奇偶校驗碼(Low-Density Parity-Check Code， LDPC code)即為一種錯誤更正碼，其錯誤更正能力極佳，接近理論最大值(夏農極限)。

在本次專題中，我實作 LDPC Code 的編碼器與解碼器(encode and decoder)，理解 LDPC code 的原理以及其解碼方式(Sum Product Algorithm，SPA)，並對不同的 code 進行比較。

另外，我也將 LDPC code 搭配正交振幅調變(Quadrature Amplitude Modulation ，QAM)去觀察其錯誤率的變化，並透過論文[3]中提到的方法去修改 Sum-Product Algorithm 來進一步低錯誤率。

(二) 原理分析與系統設計

2.1 原理分析

線性區段碼(Linear Block Code)

線性區段碼是一種錯誤更正碼，他的編碼方式可以表示成一個矩陣，也就是一個線性轉換。所有的 codeword 構成的集合則稱作 Code C，假設我們要將 k 位元的 information u 編碼成 n 位元的 codeword x，我們可以從兩種觀點來描述 C：

- 生成矩陣(Generator Matrix G)
  x 可由 u 經線性轉換得到，若我們將此線性轉換表示成一個矩陣，我們就稱這個矩陣為生成矩陣 Generator Matrix(G)，也就是說$C = \{uG \mid u \in F^k\}$
- 奇偶校驗矩陣(Parity-Check Matrix H)
  x 需要滿足一些聯立方程組，若我們將其用矩陣來表示，我們就稱這個矩陣為奇偶校驗矩陣 Parity-Check Matrix(H)，也就是說$C = \{Hx^T = 0 \mid x \in F^n\}$

根據上述的兩個觀點，我們可以得知 H 和 G 彼此正交(Orthogonal)。因此可以從 H (resp. G)找到相對應的 G (resp. H)。我們將 G 的 row 與 column 數量的比值定義成 code rate $R = \frac{\#\ of\ row}{\#\ of\ column} = \frac{k}{n}$，這個 code 稱為 (n, k) code。

低密度奇偶校驗碼(Low-Density Parity-Check Code， LDPC code)

LDPC code 是線性區段碼的一種，其特點在於它的 parity check matrix H 是一個稀疏矩陣(sparse matrix)。根據 H 中1的位置，我們可以用一個二分圖(bipartite graph)來代

表這個 LDPC code。假設 H 是一個$k \times n$的矩陣，則二分圖的一側會有 k 個 check node，另一側有 n 個 variable node，第 i 個 check node 與第 j 個 variable node 相連 ⇔ $H_{ij} = 1$。

以 $H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$為例，可以得到以下的二分圖：

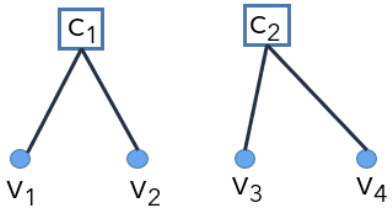

圖2-1 代表 H 的二分圖

接著我們介紹 LDPC 的 decoding algorithm(Sum-Product Algorithm，SPA)：

在二分圖上傳遞的資訊為 Log Likelihood Ratio(LLR)，其定義為：$L = ln\frac{P(x=0 \mid y)}{P(x=1 \mid y)} = ln\frac{p_0}{p_1}$，其中 y 為接收端收到的訊號而 x 則代表傳送的位元。接著，我們可以根據 check node 和 variable node 的關係去計算其所傳遞的 LLR。

- Variable Node Operation：$VAR(L_p, L_q) = L_p + L_q$
- Check Node Operation：$CHK(L_p, L_q) = sgn(L_p)sgn(L_q)\min(|L_p|, |L_q|) + \Delta(L_p, L_q)$
  其中$\Delta(L_p, L_q)$可以透過查表來實現。

接下來介紹 Sum-Product Algorithm 的各個步驟，首先我們先定義一些 notation：

(a) $V(i) = \{j : H_{ij} = 1\}$ = 跟第 i 個 check node 相連的 variable node

(b) $C(j) = \{i : H_{ij} = 1\}$ = 跟第 j 個 variable node 相連的 check node

(c) $q_{i,j}$ = 從第 j 個 check node 傳到第 i 個 variable node 的 LLR

(d) $u_{i,j}$ = 從第 i 個 variable node 傳到第 j 個 check node 的 LLR

Sum-Product Algorithm 的步驟如下：

(1) Initialization：對於所有$H_{ij} = 1$，$u_{i,j} = L_i = \ln\frac{P(x_i=0 \mid y_i)}{P(x_i=1 \mid y_i)} = \ln\frac{P(y_i \mid x_i=0)}{P(y_i \mid x_i=1)}$

(2) Bottom-Up：對於所有$H_{ij} = 1$，$q_{i,j} = CHK_{i' \in V(j)\setminus\{i\}}(u_{i',j})$

(3) Top-Down：對於所有$H_{ij} = 1$，$u_{i,j} = VAR(VAR_{j' \in V(i)\setminus\{j\}}(q_{i,j'}), L_i)$

(4) Calculate LLR：對於所有 $i = 1, 2, \ldots n$，$q_i = VAR(VAR_{j' \in V(i)}(q_{i,j'}), L_i)$

(5) Decision：對於所有 $i = 1, 2, \ldots n$，$\hat{x}_i = \begin{cases} 0 & if\ q_i \geq 0 \\ 1 & if\ q_i < 0 \end{cases}$

(6) Compute Syndrome：若上一步驟得到的$\hat{x}$滿足$H\hat{x}^T = 0$，則$\hat{x}$為 SPA 找到的 codeword；若不滿足則回到(2)繼續傳遞 LLR。

(2)~(5)進行一次稱為一個 iteration，一般來說我們會先預設一個 iteration 次數的上限(threshold)，當超過 threshold 之後 decoder 就會宣告解碼失敗(decoding failure)。

不均等錯誤保護碼(Unequal Error Protection Code， UEP)

UEP 是一種特殊的 LDPC code，透過特殊的設計使 codeword 可以切割成多個 level，不同 level 有不同的保護能力。因此，我們可以將比較重要的資訊透過保護力較強的 level 來 encode，把比較不重要的資訊用能力較差的 level 來 encode，讓整個 code 的實用更加彈性。

正交振幅調變(Quadrature Amplitude Modulation，QAM)

訊號透過兩個正交的載波來傳送資訊，透過改變各載波的振幅來代表不同的資訊。由於本專題中使用的是16-QAM，以下以16-QAM 進行解釋分析。16-QAM 的訊號點代表4個位元，訊號點根據不同的對應(Mapping)方式，會得到不同的星座圖(Constellation)，以下為本專題用到的兩個 Mapping 方式：

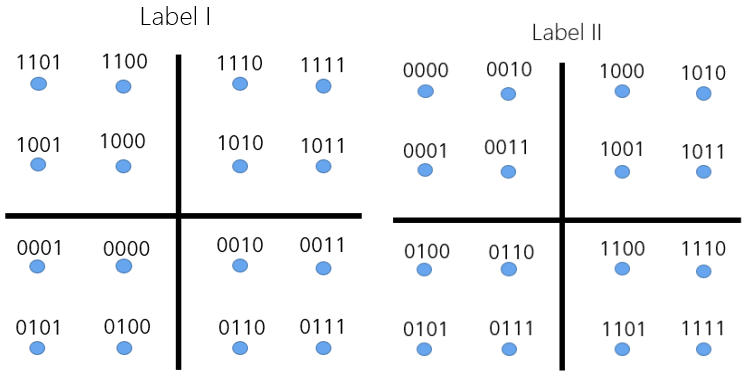

圖2-2 不同 Mapping 的16-QAM

Label I 用 Gray code 的方式來進行 Mapping，而 Label II 則是 non-Gray Mapping。當訊號在可加性高斯白雜訊通道(Additive White Gaussian Channel，簡稱 AWGN channel)傳送，我們可以預期 Label I 的平均位元錯誤率相較於 Label II 應該比較低，但在某些通訊系統中為了達成特殊的目的(例如：降低特定位元的錯誤率)，會採用 Label II 或者是其餘 non-Gray Mapping 的星座圖。

編碼調變置信度傳播(Coded Modulation Belief Propagation，CMBP)

在使用 QAM 時，星座圖中訊號點的 Mapping 方式會影響到位元間的獨立性，進而影響 SPA 的成效。為了解決此問題，我們把同一 symbol 中各位元間的相關性納入考量，得出修正過的 decoding Algorithm：CMBP。

假設 L 個位元為一個 symbol，我們將 codeword 切成$m = \frac{n}{L}$個 symbol，並用x_i

代表第 i 個 symbol 以及$x_{(i,k)}$來代表第 i 個 symbol 的第 k 個位元，$i = 1, 2, \dots m$ 以及 $k = 1, 2, \dots L$。以下用(i, k)來表示各個位元的 index：

(a) $Codeword\ x = (x_1, x_2, \dots x_m) = (x_{(1,1)}, x_{(1,2)}, \dots x_{(1,L)}, \dots x_{(m,1)}, x_{(m,2)}, \dots x_{(m,L)})$

(b) s為一個 symbol，則s^k = s 中的第 k 個 symbol。

(c) $U_{(i,k)}^T = VAR(VAR_{j' \in V((i,k))}(q_{(i,k),j'}), L_i)$

(d) $U_{(i,k)} = ln \frac{\sum_{s:s^k=0} P(x_i=s|Y_i)\, e^{\sum_{r=1, r\neq k}^{L}(1-s^r)U_{(i,r)}^T}}{\sum_{s:s^k=1} P(x_i=s|Y_i) e^{\sum_{r=1, r\neq k}^{L}(1-s^r)U_{(i,r)}^T}}$

CMBP 的步驟如下：

(1) Initialization：對於所有$H_{(i,k),j} = 1$，$u_{(i,k),j} = L_i = \ln\frac{P(x_i=0 \mid y_i)}{P(x_i=1 \mid y_i)} = \ln\frac{P(y_i \mid x_i=0)}{P(y_i \mid x_i=1)}$

(2) Bottom-Up：對於所有$H_{(i,k),j} = 1$，$q_{(i,k),j} = CHK_{(i,k)' \in V(j)\setminus\{(i,k)\}}(u_{(i,k)',j})$

(3) Top-Down：對於所有$H_{(i,k),j} = 1$，$u_{(i,k),j} = U_{(i,k)}^T - q_{(i,k),j} + U_{(i,k)}$

(4) Calculate LLR：對於所有$i = 1, 2, \dots m$以及 $k = 1, 2, \dots L$，$q_{(i,k)} = U_{(i,k)}^T + U_{(i,k)}$

(5) Decision：對於所有 $i = 1, 2, \dots n$，$\hat{x}_{(i,k)} = \begin{cases} 0 & if\ q_{(i,k)} \geq 0 \\ 1 & if\ q_{(i,k)} < 0 \end{cases}$

(6) Compute Syndrome：若上一步驟得到的$\hat{x}$滿足$H\hat{x}^T = 0$，則$\hat{x}$為 CMBP 找到的 codeword；若不滿足則回到(2)繼續傳遞 LLR。

CMBP 與 SPA 差別在於 CMBP 只是在步驟(3)和(4)多了$U_{(i,k)}$，其目的就是用來解決前面提到的同一個 symbol 間各位元的相關性問題。

2.2 系統設計

模擬方式與流程

本次專題的模擬都在 AWGN channel 下進行，模擬流程如下：

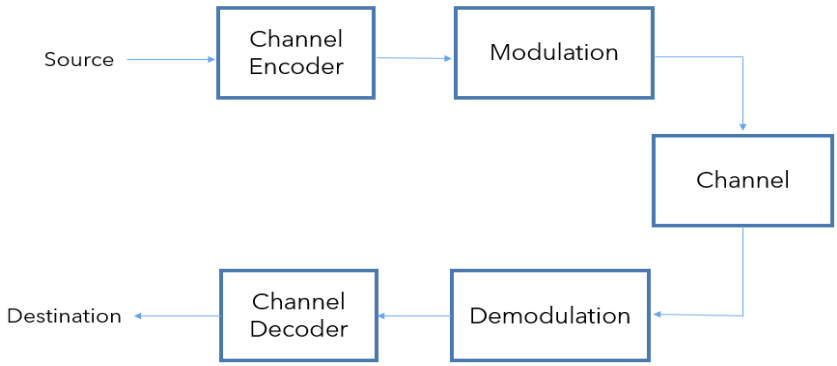

圖2-3 模擬流程圖

首先我們將 information 經過生成矩陣 G 進行 encode，接著根據調變的形式(BPSK 或16-QAM)找到對應的訊號點傳入 channel。進入 channel 後在訊號點上加入雜訊，接著在接收端進行解調，計算各個 bit 的 LLR 傳入 decoder 進行解碼得到，最後將解碼的結果與原先傳送的進行比對計算錯誤率。通道雜訊的部分，我用 C 語言產生 AWGN 來進行模擬， 透過設定不同的 SNR，觀察錯誤率的變化，進而進行比較。

奇偶校驗矩陣(parity-check matrix)架構

以下是我在這個專題裡所使用的 LDPC code 的 parity-check matrix：

code	(n, k)	Type	R	Level I : level II
QC-LDPC	(7376, 3688)	regular EEP	0.5	x
C_2	(7376, 3688)	UEP	0.5	1 : 1
Mackay	(9972, 4986	Irregular EEP	0.5	x
C_3	(9872, 4936)	UEP	0.5	9 : 7

表2-1 本次專題使用的 LDPC code

在進行 EEP 和 UEP 的比較時，我們會將長度相近的 codeword 進行比較。在這個專題中，就是將 QC-LDPC 與 C2做比較、Mackay 與 C3做比較。

（三）實驗結果

1. UEP 與 EEP 在 BPSK 下的錯誤率比較

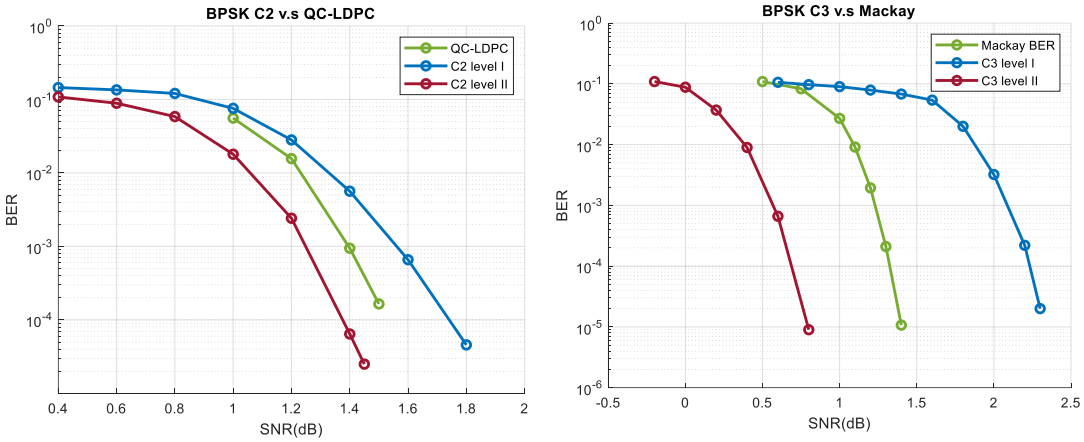

圖3-1 BPSK 下 UEP 與 EEP 比較

首先，我們可以觀察到隨著 SNR 上升，BER 都有下降的趨勢，且在到達某個 SNR 後有急遽下降的趨勢，我們稱其為 waterfall region。接著，可以觀察到 C2的 level I 和 level II 有保護能力上的差異，兩者在錯誤率為10^{-5}時所需的 SNR 有約 0.5dB 的差距。而 C3則更加明顯，兩個 level 在錯誤率為10^{-5}時所需的 SNR 有 1.5dB 的差距。模擬結果顯示 UEP code 確實能在不同 level 展現不同的保護能力，實際運用上我們能將 level II 來保護較容易產生錯誤的資料，level I 保護不容易產生錯誤的資料。

2. UEP 與 EEP 在16-QAM Label I 下的錯誤率比較

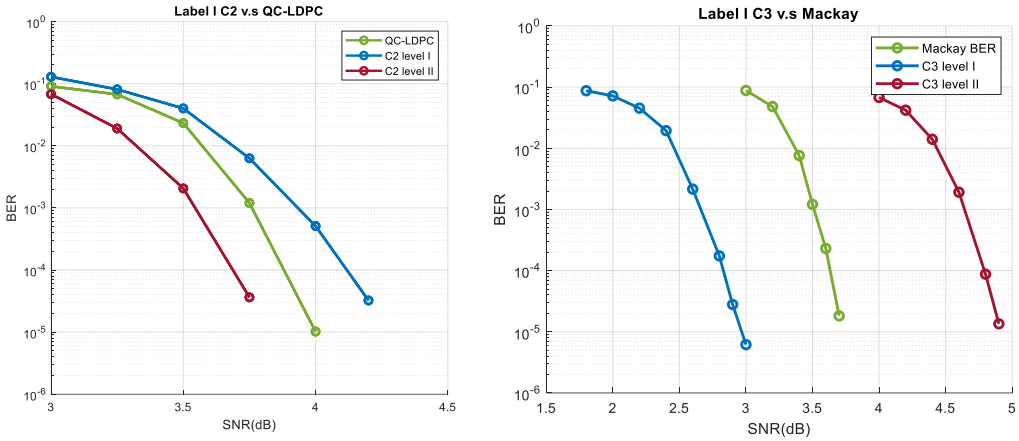

圖3-2 Label I 下 UEP 與 EEP 比較

可以發現圖3-1與圖3-3差異不大、圖3-2與圖3-4差異不大，只是在達成相同錯誤率的情況下，16-QAM所需的SNR較高，BPSK與16-QAM相比兩者所需的SNR約有2.2、2.3dB 的差距。16-QAM 在提升頻寬效率時所付出的代價就是錯誤率上升。

3. CMBP 在16-QAM 下對錯誤率的改善

◆ Label I　SPA v.s CMBP

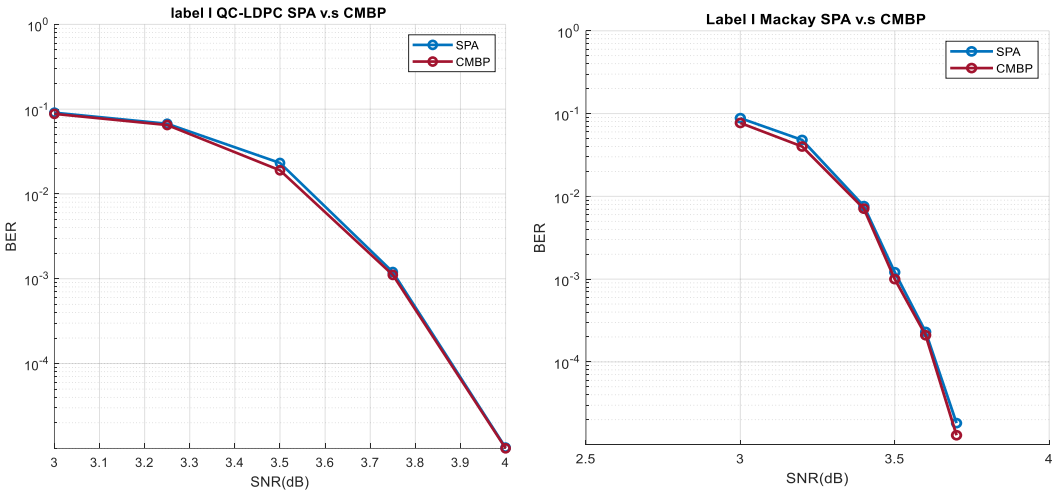

圖3-3 EEP 在 Label I 下 SPA 與 CMBP 的比較

在使用 Label I 的情況下，採用 CMBP 作為 decoding algorithm 所得到的 BER 與 SPA 的效果相比只有些微的改善差異不大。這與我們的預期相符，因為 Label I 為 Gray Mapping，同一個 symbol 間各位元的相關性不大，接近統計獨立，與 SPA 的預先假設相符。

◆ **Label II　SPA v.s. CMBP**

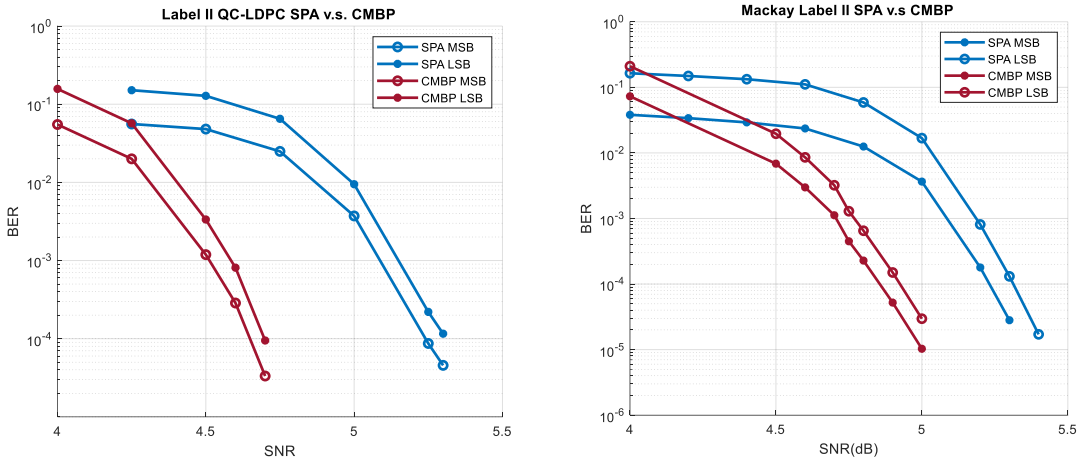

圖3-4 EEP 在 Label II 下 SPA 與 CMBP 的比較

在使用 Label II 的情況下，採用 CMBP 作為 decoding algorithm 所得到的 BER 有所降低，效果相較於 SPA 有顯著的提升。差異的原因如我們先前所述，non-Gray Mapping 會提升位元間的相關性，此時 SPA 的預先假設就不成立了。因此，在這種情況下我們應該採用 CMBP，以達成錯誤率進一步降低。

（四）結論

從模擬結果可以看出，UEP code確實能夠在同一個codeword中提供不同程度的保護效果。我們能夠將保護力較強的部分使用在較為重要或容易出錯的訊息上，而不必設計一個整體保護力強的EEP code，降低設計上的難度並提升系統的彈性。

在將LDPC code與QAM做搭配時，QAM星座圖的Mapping方式會影響解碼的成效，我們需要對SPA進行修改，考慮同一symbol間各位元的統計特性(CMBP)。當星座圖為Gray Mapping時，各位元之間接近統計獨立，SPA與CMBP的效果相近，使用SPA即可達到不錯的效果；當星座圖為non-Gray Mapping時，各位元間的相關性提升，此時我們應該採用CMBP，改善SPA效果衰減的問題。

心得感想

首先，我要感謝趙啟超教授在這兩學期中的指導，在每次 meeting 中給予我專題上的幫助、觀念的釐清以及專題報告撰寫上的建議。另外，我也想感謝實驗室的楊博鈞學長，在研究之餘能抽空解答我一些實作上的細節。

當初會選擇錯誤更正碼作為專題主題是因為本身對數學和程式設計滿有興趣，也是在修的課中成績較好的，因此通訊領域中的錯誤更正碼就成為一個選擇，再加上大二時因為數理必選在數學系修了代數(一)和代數(二)，其中就有簡略地介紹到 BCH code(一種錯誤更正碼)以及一些 Finite Field 的觀念，這些知識對於讓我在閱讀 LDPC code 相關的論文時，更容易了解其背後的理論。電機系的必修課當中有一半的數學課，在過去的修課中課程內容往往著重與理論而非實際應用的部分，學習的過程中時常會有迷惘的感覺。透過這次專題的過程，我實際體會過去學習的理論於實務方面的用處，有學以致用的喜悅。

另外，理解理論後寫程式進行實作的過程也讓我獲益良多。在 SPA 以及 CMBP 實作上，理論所推出的公式不一定能直接用程式實作(ex. check node operation 或是 CMBP 中的$U_{(i,k)}$)，往往需要將公式進行整理，不僅能降低計算複雜度，甚至能避免一些潛在的風險(ex. double overflow)。尤其是 overflow 的問題相當常見，在 SNR 很高、錯誤率極低時，LLR 很容易出現 overflow 的情況，這時要如何修改 check node operation 以及 variable node operation 就很重要。這些都是我在過去撰寫程式時沒注意過的細節。

最後再次感謝趙啟超教授以及楊博鈞學長在這次專題中的指導以及幫助，讓我對通訊領域更加認識，度過充實且有趣的兩個學期。