低密度奇偶校驗碼之編解碼器實現與改善

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

組別：A291　組員：林峻霆　指導教授：趙啟超 教授

Abstract

低密度奇偶校驗碼(Low-Density Parity-Check Code， LDPC code)為現代通訊系統以及資料儲存系統常用的一種錯誤更正碼。本次專題的目的在於實作LDPC Code的編碼器與解碼器，對不同的code進行比較。此外，也將實作成果搭配QAM，探討在不同的星座圖下解碼效果的變化，並透過修改演算法來使錯誤率進一步降低。

Theory

線性區段碼(Linear Block Code)

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x：

- 生成矩陣(Generator Matrix G)
  x可由u線性轉換得到。將此線性轉換表示為矩陣G，x = uG
- 奇偶校驗矩陣(Parity-Check Matrix H)
  x需要滿足一些聯立方程組，將其用矩陣H表示，得到$Hx^T = 0$

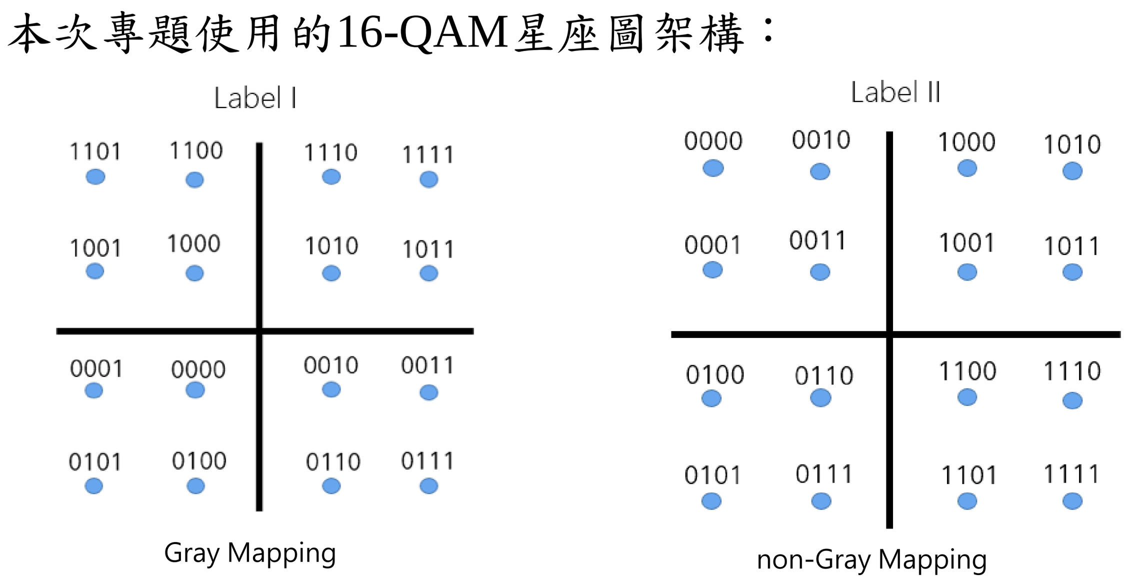
我們稱這個code為(n, k) Linear Block Code。

低密度奇偶校驗碼(LDPC code)

LDPC code是線性區段碼的一種，其特點在於它的奇偶校驗矩陣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。有些特殊結構的LDPC code會讓一個codeword中不同位元有不同的保護能力，稱為不均等保護碼(UEP)，與之相對的是均等保護碼(EEP)。

正交振幅調變(QAM)

以下為本次專題使用的16-QAM星座圖架構：



Algorithms

- 在二分圖上傳遞的資訊為Log Likelihood Ratio(LLR)

  LLR：$L = ln\frac{P(x=0\mid y)}{P(x=1\mid y)} = ln\frac{p_0}{p_1}$

  $VAR(L_p, L_q) = L_p + L_q$

  $CHK(L_p, L_q) = sgn(L_p)sgn(L_q)\min(|L_p|, |L_q|) + \Delta(L_p, L_q)$

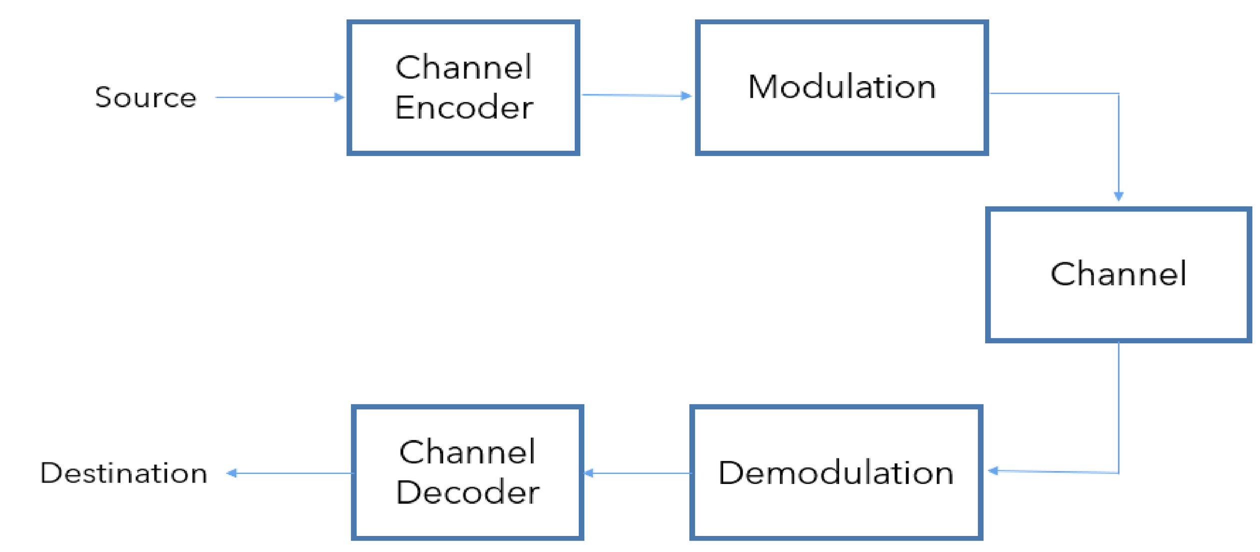
- 根據二分圖的結構，定義
  (a) $V(i) = \{j : H_{ij} = 1\}$ = 跟第i個check node相連的variable node
  (b) $C(j) = \{i : H_{ij} = 1\}$ = 跟第j個variable node相連的check node
  (c) $q_{i,j}$ = 從第j個check node傳到第i個variable node的LLR
  (d) $u_{i,j}$ = 從第i個variable node傳到第j個check node的LLR

- **Sum-Product Algorithm**
  (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$i = 1, 2, \dots n$，$q_i = VAR(VAR_{j'\in V(i)}(q_{i,j'}), L_i)$
  (5)Decision：對於所有$i = 1, 2, \dots n$，$\hat{x}_i = \begin{cases} 0 & if\ q_i \ge 0 \\ 1 & if\ q_i < 0 \end{cases}$
  (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。

- 在進行QAM調變時，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x_i的第k個位元，其中$i = 1, 2, \dots m$ 以及 $k = 1, 2, \dots L$。以下用(i, k)來表示各個位元的index：
  (a) $Codeword\ x = (x_1, x_2, \dots x_m) = (x_{(1,1)}, x_{(1,2)}, \dots x_{(1,L)}, \dots x_{(m,1)}, \dots x_{(m,L)})$
  (b) 若s為一個symbol，則s^k為s中的第k個symbol。
  (c) $U^T_{(i,k)} = VAR(VAR_{j'\in V((i,k))}(q_{(i,k),j'}), L_i)$
  (d) $U_{(i,k)} = ln\frac{\sum_{s:s^k=0} P(x_i=s|Y_i)\, e^{\sum_{r=1,r\neq k}^{L}(1-s^r)U^T_{(i,r)}}}{\sum_{s:s^k=1} P(x_i=s|Y_i) e^{\sum_{r=1,r\neq k}^{L}(1-s^r)U^T_{(i,r)}}}$

- **Coded Modulation Belief Propagation(CMBP)**
  步驟上與SPA相同，只需修改步驟(3)與(4)，如下：
  (3*) Top-Down：對於所有$H_{(i,k),j} = 1$，$u_{(i,k),j} = U^T_{(i,k)} - q_{(i,k),j} + U_{(i,k)}$
  (4*) Calculate LLR：對於所有i, k，$q_{(i,k)} = U^T_{(i,k)} + U_{(i,k)}$
  CMBP與SPA差別在於CMBP只是在步驟(3)和(4)多了$U_{(i,k)}$，其目的在於解決同一個symbol間各位元非統計獨立的問題。

Framework

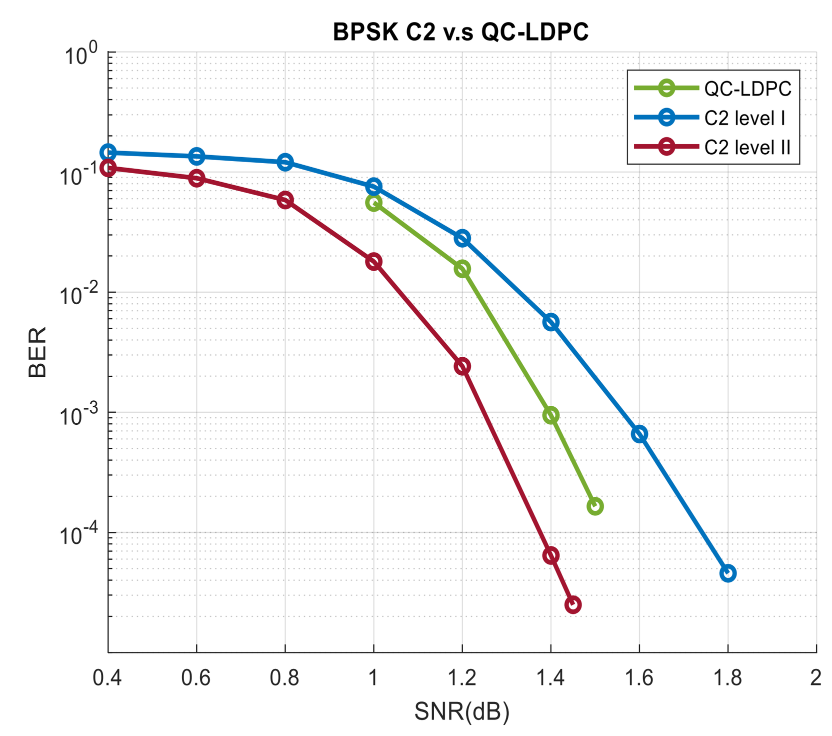


Stage	Function
Encoder	codeword x = uG
Modulation	BPSK or 16-QAM
Channel	AWGN channel
Demodulation	Calculate initial LLR
Decoder	SPA or CMBP

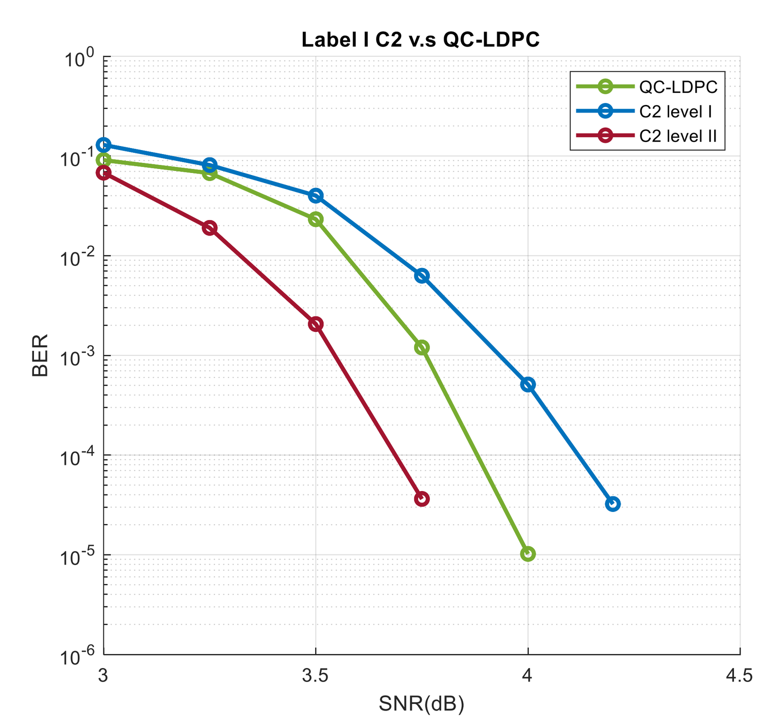
Results

UEP與EEP在BPSK與16-QAM下的比較

BPSK



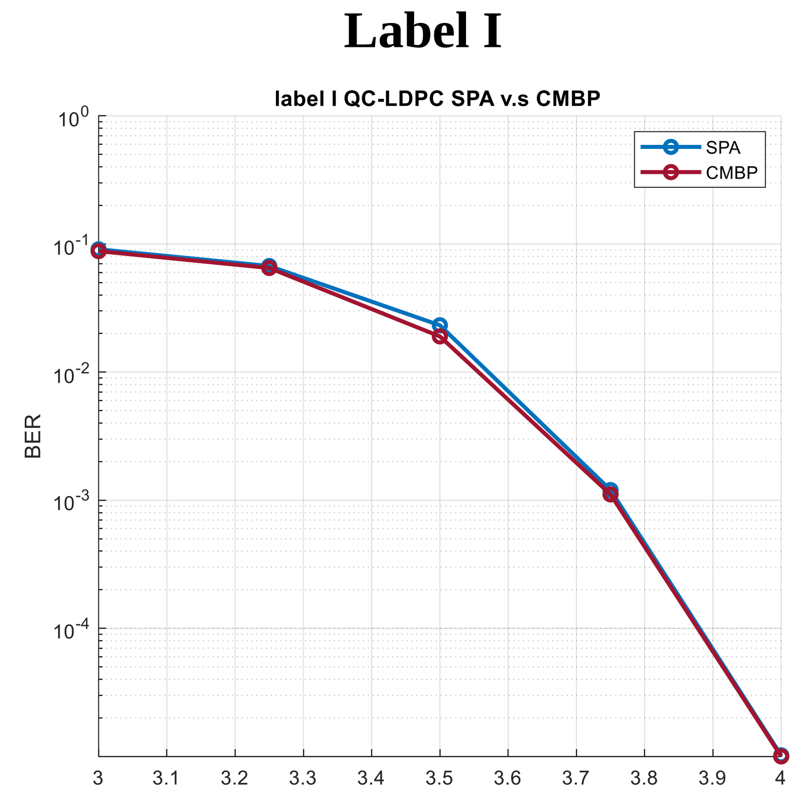
16-QAM Label I



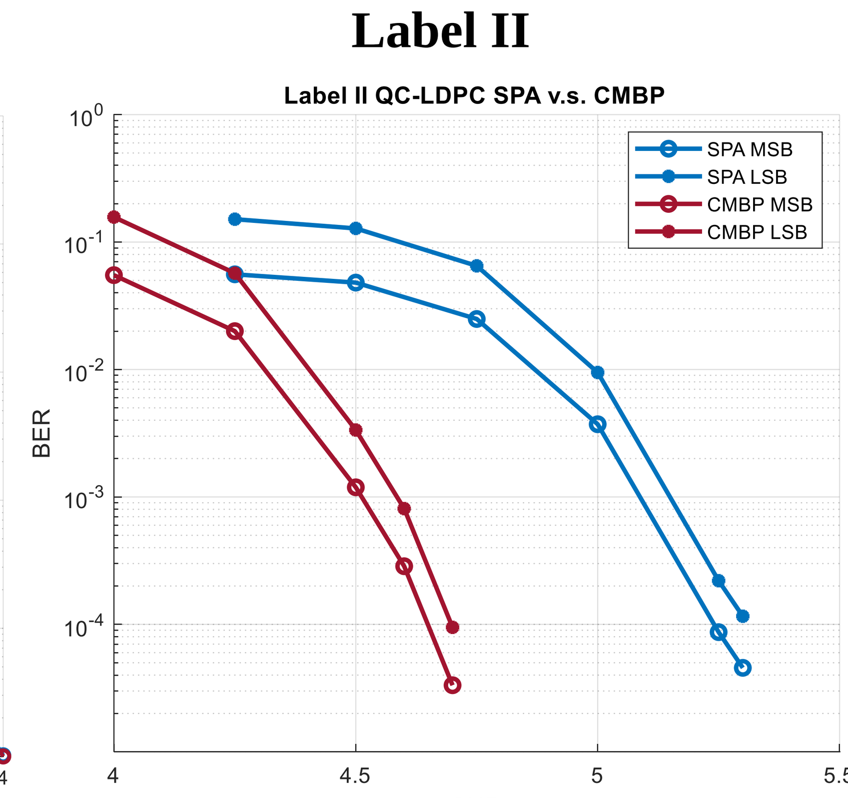
UEP code確實能夠在同一個codeword中提供不同程度的保護效果。我們能夠將保護力較強的部分使用在較為重要或容易出錯的訊息上，而不必設計一個整體保護力強的EEP code，降低設計上的難度並提升系統的彈性。

CMBP在16-QAM下對錯誤率的改善

Label I



Label II



LDPC code與QAM做搭配時，QAM星座圖的Mapping方式會影響解碼的成效。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。