科技部補助專題研究計畫

結案報告書

華航電 104 產學字 001 號

使用虛擬軟式訊息於非二元低密度奇偶檢查碼的解碼演算法之研究

計畫時程：自民國 104 年 8 月 1 日至民國 105 年 7 月 31 日

計畫執行單位：中華學校財團法人中華科技大學 航空電子系

計畫主持人：陳作舟 副教授

科技部補助專題研究計畫成果報告

（□期中進度報告/■期末報告）

使用虛擬軟式訊息於非二元低密度奇偶檢查碼的解碼演算法之研究

計畫類別：■個別型計畫　□整合型計畫
計畫編號：MOST 104-2221-E-157-001 -
執行期間：104 年 8 月 1 日至 105 年 7 月 31 日

執行機構及系所：中華科技大學 航空電子系

計畫主持人：陳作舟
共同主持人：
計畫參與人員：陳柏丞，鄭許驊，李政穎

本計畫除繳交成果報告外，另含下列出國報告，共 1 份：
□執行國際合作與移地研究心得報告
■出席國際學術會議心得報告
□出國參訪及考察心得報告

中　華　民　國　105　年　8　月　1　日

目錄

1. 中文摘要

本計畫乃針對非二元 LDPC 碼在解碼時,精確的事前軟式訊息不易從通道上直接獲得之應用，設計具有多隨機符元錯誤改錯能力之低複雜度解碼演算法。非二元 LDPC 碼在軟式判決解碼時所需的事前軟式訊息將由 RS 碼之偵錯器輸出獲得,稱為虛擬軟式訊息。非二元 LDPC 碼即利用該虛擬軟式訊息當作事前軟式訊息，進行置信度傳遞解碼。因此我們將設計混合乘積碼以獲得該虛擬軟式訊息，該混合乘積碼是由非二元 LDPC 碼與 RS 碼串接組成。

非二元 LDPC 碼在改錯性能上比二元 LDPC 碼更優異，但其解碼複雜度較高。本計畫將針對非二元 LDPC 碼的解碼演算法中，複雜度最高的校驗節點處理部分研究降低複雜度的方法；所設計的新解碼演算法主要是基於低複雜度的 Min-Max 演算法及可提升解碼收斂速度的分層解碼方法。

計畫將達成下列目標：(1)設計串接非二元 LDPC 碼與 RS 碼之混合乘積碼。(2)理論推導非二元 LDPC 碼解碼所需之虛擬軟式訊息。(3)使用虛擬軟式訊息作為事前軟式訊息，設計低複雜度的非二元 LDPC 碼之解碼演算法，使得性能與運算複雜度上獲得最佳權衡。(4)針對所提出的方法進行複雜度與性能模擬分析，並與國內外研究所提出的方法比較，驗證所提出方法的有效性。(5)針對所提出使用虛擬軟式訊息於非二元 LDPC 碼的解碼演算法,設計具有低複雜度的 FPGA 硬體。

2. 英文摘要

In this project, the low-complexity decoding algorithm of nonbinary low-density parity-check code (NB-LDPC) is studied and designed to correct multiple random symbol errors for some applications whose prior soft information is difficult to obtain from channel. The prior soft information, known as pseudo-soft information, required for the soft-decision decoding of NB-LDPC code is provided from the error detector output of RS code. Then this pseudo-soft information is used as the prior soft information for the brief-propagation (BP) decoding of NB-LDPC code. Therefore, we design the hybrid product codes in which NB-LDPC code is concatenated with the Reed-Solomon (RS) code to obtain the pseudo-soft information.

NB-LDPC codes have better error correcting performance than binary LDPC codes, but the decoding complexity is high. Since check node processing as part of BP decoding algorithm has highest complexity, the method to lower the computational complexity is studied in the project. The new designed decoding algorithm is mainly based on the low-complexity Min-Max algorithm and the layered decoding scheme which can improve decoding convergence speed.

The following objects of this project will be achieved. (1) A hybrid product code in which NB-LDPC code is concatenated with the RS code is designed. (2) The pseudo-soft information for the BP decoding of NB-LDPC code is derived theoretically. (3) By using the pseudo-soft information as the prior soft information, a low complexity decoding algorithm of NB-LDPC code is designed to achieve a good performance and complexity trade-off. (4) To examine the true efficiency of the proposed method, we perform simulations and analyze performances and computational complexity of the proposed encoding and decoding method. The comparisons with the existing methods are also provided. (5) A low complexity FPGA hardware of the proposed decoding algorithm with pseudo-soft information for NB-LDPC code is designed.

3. 研究目的

本計畫之目的乃針對 NB-LDPC 碼在解碼時，精確的事前軟式訊息不易從通道上直接獲得之應用(例如：高速光通訊、快閃記憶體，高密度的磁性儲存器等)，設計具有多隨機符元錯誤(multiple random symbol errors)改錯能力之低複雜度解碼演算法。NB-LDPC 碼在軟式判決解碼時所需的事前軟式訊息將由 RS 碼之偵錯器輸出獲得，稱為虛擬軟式訊息(pseudo-soft information)。NB-LDPC 碼隨即利用該虛擬軟式訊息作為事前軟式訊息，進行置信度傳遞(BP)反覆運算解碼。基於此，首先將設計混合乘積碼(hybrid product code) 以獲得該虛擬軟式訊息，而該混合乘積碼由兩類組成碼(component code) 串聯組合而成串接碼(concatenated codes)，以 NB-LDPC 碼為外碼(outer code)，RS 碼為內碼(inner code)。

NB-LDPC 碼在改錯性能上比二元 LDPC 碼更加優異，但其解碼複雜度較高，本計畫接著將對 NB-LDPC 碼的 BP 解碼演算法中，複雜度最高的校驗節點處理部分研究降低複雜度的方法，主要以可提升解碼收斂速度的分層解碼方法和低複雜度的 Min-Max 演算法為基礎，設計一種新的解碼演算法。

計畫研究達成下列創新目標：(1)設計以 NB-LDPC 碼為外碼，RS 碼為內碼之串接式混合乘積碼。(2)理論推導出 NB-LDPC 碼解碼時所需之虛擬軟式訊息。(3)針對高數據傳輸率的應用，使用虛擬軟式訊息作為事前軟式訊息，開發低複雜度且易於硬體實現的 NB-LDPC 碼解碼演算法。(4)基於符元錯誤率(symbol error rate)性能進行模擬分析，以獲得最佳的 NB-LDPC 碼與 RS 碼串接組合。(5)針對所提出的 NB-LDPC 碼之解碼演算法進行複雜度分析，並與國內外研究所提出的方法比較。(6)針對所提出的 NB-LDPC 碼之解碼演算法進行改錯性能模擬分析，並與國內外研究所提出的方法比較，驗證所提出方法的有效性。(7)針對所提出使用虛擬軟式訊息於 NB-LDPC 碼的解碼演算法，設計具有低複雜度的 FPGA 硬體。

4. 文獻探討

低密度奇偶檢查(low-density parity-check, LDPC)碼於 1962 年由 Gallager[1]所提出的一種錯誤控制編碼，具有優良的改錯性能，但自從它被提出之後，直到 1981 年才由 Tanner[2]提出使用圖形理論(graph theory)構建 LDPC 碼之編解碼方法而重新受到研究者的重視。於文獻[3]提出 LDPC 碼反覆運算訊息互傳解碼方法，文獻[4][5]證明 LDPC 碼在加成白色高斯雜訊(additive white Gaussian noise, AWGN)通道下可達到趨近於 Shannon 容量。LDPC 碼除具有接近 Shannon 限界的優異性能外，尚具有下列優點：與 Turbo 碼相比解碼演算法的複雜度較低；誤碼平層(error floor) 較低；具有平行且反覆運算的架構，解碼速度快，解碼器簡單且硬體成本低，適合於硬體的實現；又因為 LDPC 的奇偶檢查矩陣有著稀疏特性，位元與位元之間相隔比較遠，因此具有等同於交錯器的效果，使其在編碼時不需要額外的交錯器。目前在深空通訊、無線區域網路(IEEE 802.11n)[6]、WiMax (IEEE 802.16e)[7]、衛星數位視訊廣播標準(DVB-S2)[8]等系統中皆被採用，因此在未來有關 LDPC 碼的應用將相當的廣泛。

二元LDPC碼在實際中已經得到了廣泛的應用，另研究發現非二元LDPC (nonbinary LDPC, NB-LDPC)碼的性能比二元LDPC碼更加優異[9]，主要表現在以

下幾個方面：第一：NB-LDPC具有更好的多隨機符元錯誤(multiple random symbol errors)之偵錯與改錯能力。有限場$GF(q)$上的NB-LDPC碼的具有更低的錯誤平層(error floor)和快速的收斂性；第二：NB-LDPC碼有較強的抗叢發錯誤(burst error)能力。實際通道中產生的錯誤經常是叢發錯誤或叢發錯誤與隨機錯誤並存。而二元LDPC碼的抗叢發錯誤能力不是很強，因此在實際系統中通常需要與RS碼串聯。由於NB-LDPC碼可以將位元錯誤合併成較少的符元(symbol)錯誤，因而NB-LDPC碼改正叢發錯誤的性能優於二元LDPC碼；第三：NB-LDPC碼更適合高速率傳輸系統，例如光通訊。NB-LDPC碼是基於高階有限場設計的，因此非常適合與高階調變或多天線系統結合，從而提供更高的資料傳輸速率和頻譜利用效率。因此，高性能NB-LDPC碼的編解碼是很重要的研究領域[10-13]。NB-LDPC碼的可構建較長的Tanner圖周長(girth)，這種特性使得對其解碼的最佳化成為可能，周長更長隱含著短環路對解碼收斂性的影響更小，因此使得基於置信度傳遞(brief propagation, BP)解碼或和積演算法(sum-product algorithm, SPA)能更有效地趨近最大似然(maximum likelihood)解碼演算法。NB-LDPC碼的優勢還表現在高階調變方面，假設NB-LDPC碼所處的有限場的階數為a，調變星座的階數為b，那麼很多情況下a都是大於b的，這樣就使得解碼器得到的事前訊息(prior information)是不相關的向量，有利於實現最佳解碼，從而獲得更好的改錯性能，因此，NB-LDPC碼非常適合應用在高階調變系統，實現高速率的資料傳輸。這些成果同時也說明NB-LDPC碼擁有廣泛的應用前景，足以在通信與存儲系統中替代RS碼。

關於NB-LDPC之解碼研究，1998年Davey和MacKay第一次提出NB-LDPC碼[14]，在文獻中提出了一種NB-LDPC碼之q-ary和積解碼演算法(QSPA)，但是在編碼增益增加的同時，解碼複雜度隨著編碼場元素的增加迅速變大，變得非常不利於硬體實現。因此，在一段時期內，NB-LDPC碼的研究與應用並不像二元LDPC碼那樣廣泛，發展速度也受到了一定程度的限制。但後來的研究說明，NB-LDPC碼的複雜度是可以被降低的。Fossorier提出的擴展最小和(extended min-sum, EMS)解碼演算法[15]和Lin Shu等學者提出的非二元QC-LDPC碼[16][17]，顯著降低了NB-LDPC碼的實現複雜度[18]。文獻[19]提出Min-Max演算法，其含有兩種LLR (Log-Likelihood Ratio)實現方式，一種是標準實現方式，它的複雜度與有限場元素的個數平方成正比；另一種是選擇性實現方式，相較於標準實現方式，這種實現方式的複雜度更低，這就使得Min-Max演算法在實際應用方面表現出了很強的吸引力。

在一系列LDPC碼中，結構化的QC-LDPC碼得到了廣泛的應用，這主要是因為QC-LDPC碼在硬體實現層面效率很高，並且它的解碼性能很好，而分層解碼(layered decoding)方法非常適合於QC-LDPC碼的解碼方法。分層解碼方法不僅可以應用到二元LDPC碼[20][21]，而且同樣適用於NB-LDPC碼[22][23]。分層解碼方法不僅可有效地增加解碼的收斂速度，從而使得解碼所需的最大反覆運算次數大大降低，使得數據輸送量得到了極大的提高，而且在一定程度上改善了解碼性能，因此該解碼方法在學術上和工業應用上都受到了極大的關注。

一般而言，NB-LDPC碼進行解碼時需要有精確的事前軟式訊息(prior soft information)，以進行軟式BP解碼演算法。但對於某些應用，其精確的軟式訊息將較難以從通道中直接獲得[24-33]，例如：高速光通訊、快閃記憶體，高密度的磁性儲存器等。雖然上述應用的軟式訊息可經由量化獲得，但量化的軟式訊息將使得性能降低，且在硬體實現的複雜度將隨量化的離散取值的數量而增加

[25][32]。在文獻[33]說明軟式訊息使用3個位元量化時，在位元錯誤率為10^{-6}時，編碼增益(coding gain)損失3dB。因此本計畫將研究在某些應用缺乏事前軟式訊息的情況下，針對NB-LDPC碼，開發可使用軟式判決解碼(soft-decision decoding)之BP演算法，其所需的事前軟式訊息將由Reed-Solomon (RS)碼的偵錯器(error detector)輸出獲得，稱為虛擬軟式訊息(pseudo-soft information)；基於此首先需設計以NB-LDPC碼為外碼，以RS碼為內碼之串接式混合乘積碼，以獲得虛擬軟式訊息。為方便說明本計畫的研究方法，先就乘積碼的背景描述如下。

乘積碼最早由 Elias 於 1954 年提出[34]，但由於過高的解碼複雜度和當時的硬體實現能力限制了其應用。1993 年由 C. Berrou 等人提出的 Turbo 碼[35]採用了軟式訊息反覆運算解碼的構想，經研究證明其改錯性能可接近 Shannon 限界，因而成為編碼領域的研究重點，並被廣泛應用於深空通信和行動通信領域。但另一方面由於其解碼演算法的複雜性，導致其解碼器結構較複雜、成本過高而且解碼器的處理速度受到限制。1994 年，R. Pyndiah 將 Turbo 軟式訊息反覆運算解碼的構想應用於乘積碼之中，提出了方塊 Turbo 碼(block Turbo code，BTC)，又稱 Turbo 乘積碼(Turbo product codes，TPC)[36]，從這個時期開始更多的學者和研究機構開始重新關注和研究乘積碼。相對於 Turbo 碼而言，乘積碼有效地實現了解碼性能與硬體實現複雜度的折衷，很容易由硬體實現。大量研究證明，Turbo 乘積碼可以做到高碼率，近通道容量時仍可保持較好的性能，而且具有很強的抗衰落、抗干擾能力，正是基於這一點，在通道條件較差的無線通訊系統中 Turbo 乘積碼具有很大的應用潛力。乘積碼的概念是一種能用兩個或多個短碼構建為長碼的有效方法。

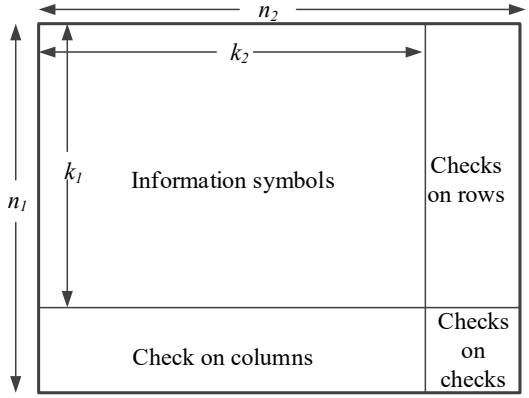

圖 1 乘積碼 $P = C_1 \otimes C_2$ 之構成

考慮兩個系統線性方塊碼：參數為 (n_1, k_1, δ_1) 的 C_1 碼和參數為 (n_2, k_2, δ_2) 的 C_2 碼，其中 n_1, n_2 為碼長，k_1, k_2 為資訊數，δ_1, δ_2 表示碼的最小距離(minimum distance)。如圖 1 所示。乘積碼 $C_1 \otimes C_2$ 構成如下：

(1) 將 $(k_1 \times k_2)$ 資訊排成 $k_1 \times k_2$ 矩陣。

(2) 用 C_2 碼對 k_1 列進行編碼，得到 $(k_1 \times n_2)$ 矩陣。

(3) 對上一步驟所得到的 $(k_1 \times n_2)$ 矩陣，用 C_1 對 n_2 行進行編碼，得到 $(n_1 \times n_2)$ 碼字矩陣 **P**。

所獲得的乘積碼 $P = C_1 \otimes C_2$ 的參數為 $(n = n_1 \times n_2,\ k = k_1 \times k_2,\ \delta = \delta_1 \times \delta_2)$。碼率 $R_c = R_{c_1} \times R_{c_2}$，其中 R_{c_i} 為 C_i 碼的碼率。由於 C_1, C_2 為系統碼，顯然碼字矩陣 **P** 的全部 n_1 列都是 C_2 的碼字而所有 n_2 行都是 C_1 碼的碼字。

當 BTC 解碼器接收到一組碼字時，首先按照編碼器的結構把序列恢復成一個碼塊 **R**。如圖 2 所示 BTC 採用反覆運算解碼的方式，每輪反覆運算解碼都先把碼塊的各列逐一列列解碼，列的解碼結束的碼塊記為 $\mathbf{R}'$，並進行如下更新：$\mathbf{R}' = \mathbf{R} + \alpha \mathbf{W}'$，其中 $\mathbf{W}'$ 為上次解碼所獲得的外部資訊(extrinsic information)，係數 α 的作用是控制外部資訊對 soft-in soft-out (SISO)解碼器的影響，在前幾輪反覆運算中位元錯誤率相對較高，外部資訊尚未很可靠，α 可以取較小的值，隨著反覆運算次數的增多，位元錯誤率逐步降低，此時可將 α 的值逐步增大，使外部資訊可被充分利用。然後進行的解碼，相當於將碼塊轉置，然後逐列解碼，最後再將碼塊轉置，再進行更新：$\mathbf{R}' = \mathbf{R} + \alpha \mathbf{W}'$，這樣就完成了一輪反覆運算解碼。

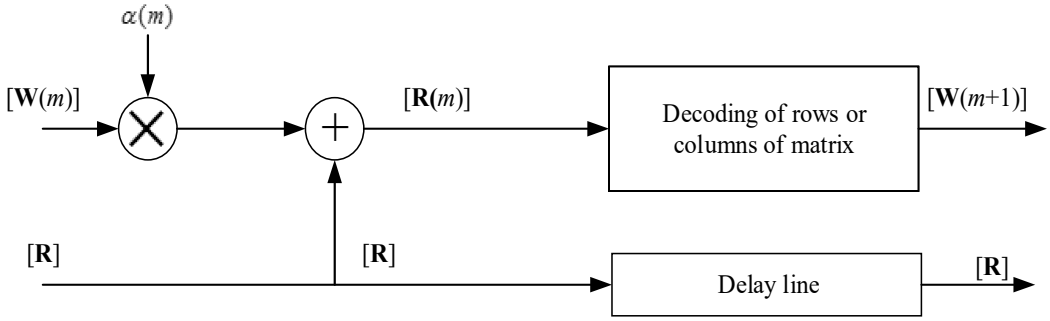

圖 2 Block turbo decoder 方塊圖[3]

乘積碼可由兩類組成碼(component code)串聯組合而成串接碼(concatenated codes)。由於LDPC碼具有接近Shannon限界的優異性能外，並可視為由多個單一奇偶檢查碼(single parity-check (SPC) codes)交錯連接而成，因此LDPC碼可選擇為構成乘積碼其中之一組成碼。

參考文獻

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.

[2] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 5, pp. 533-547, 1981.

[3] M. P. C. Fossorier, "Iterative reliability-based decoding of low-density parity-check codes," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 908-917, 2005.

[4] D. J. C. Mackay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, pp. 533-547, 1981.

[5] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58-60, 2001.

[6] *IEEE P802.11 Wireless LANs TGn Sync Proposal: TGn Sync Proposal Technical Specification,* IEEE 11-04-0889-06-000n, May 2005.

[7] IEEE Std. 802.16e and IEEE Std. 802.16-2004/Cor. 1-2005, "Part 16: Air interface for fixed and mobile broadband wireless access systems," Feb. 2006.

[8] European Telecommunications Standards Institute (ETSI), "Digital Video Broadcasting (DVB): Second generation framing structure, channel coding and modulation system for broadcasting, interactive services, news gathering and other broadband satellite applications," EN 302 307, V1.1.2, www.dvb.org.
[9] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*, Cambridge University Press, Cambridge, UK, 2009, pp.592-633.
[10] C. Poulliat, M. Fossorier, and D. Declercq, "Optimization of nonbinary LDPC codes using binary images," in Proc. Int. Trubo Code Symp. Munich, April 2006, pp. 93-97.
[11] C. Poulliat, M. Fossorier, and D. Declercq, "Design of nonbinary LDPC codes using their binary images: algebraic properties," in Proc. Int. Trubo Code Symp., Munich, April 2006, pp. 963-965.
[12] S. Song, L. Zen, S. Lin, and K. Abdel-Ghaffar, "Algebraic constructions of nonbinary quasi-cyclic LDPC codes," in Proc. IEEE Int. Symp. Information Theory, Seattle, WA, July 2006, pp. 83-87.
[13] A. Voicila, D. declercq, F. Verdier, M. Fossorier, and P. Urard, "Split nonbinary LDPC codes," in Proc. IEEE Int. Symp. Information Theory, Toronto, Ontario, July 2008, pp.955-959.
[14] M. C. Davey and D. J. MacKay, "Low-density parity check codes over GF(q)," *IEEE Commun. Lett.,* vol. 2, no. 6, pp. 165-167, 1998.
[15] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633-643, 2007.
[16] L. Lan, L. Zeng , Y. Y. Tai , L. Chen , S. Lin and Î. Abdel-Ghaffar "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: a finite field approach," *IEEE Trans. Inf. Theory,* vol. 53, no. 7, pp.2429 -2458, 2007.
[17] S. Lin, S. Song, B. Zhou, J. Kang, Y.Y. Tai, and Q. Huang, "Algebraic constructions of nonbinary quasi-cyclic LDPC codes: Array masking and dispersion," in Proc. 2nd Workshop for the Center of Information Theory and its Applications, San Diego, 2007.
[18] J. Kang, Q. Huang, L. Zhang, et al., "Quasi-cyclic LDPC codes: An algebraic construction," *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1383-1396, 2010.
[19] V. Savin, "Min-Max decoding for nonbinary LDPC codes," in Proc. IEEE Int. Symp. Information Theory, Canada, 2008, pp. 960-964.
[20] Y. Sun and J. R. Cavallaro, "VLSI architecture for layered decoding of QC-LDPC codes with high circulant weight," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no.10, pp.1960-1964, 2012.
[21] X. M. Zhang and F. Cai, "Reduced-complexity decoder architecture for non-binary LDPC codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 7, pp. 1229-1238, 2011.
[22] Z. Shuai, S. Jin, L. Li, et al., "Layered decoding for non-binary LDPC codes," in Proc. IEEE Int. Symp. Circuits and Systems, Paris, 2010, pp. 481-484.
[23] L. U. Yeong, Y. L. Chen, J. Y. Chung, et al., "An efficient layered decoding architecture for nonbinary QC-LDPC codes," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 59, no. 2, pp.385-398, 2012.
[24] H. Kidorf, N. Ramanujam, and I. Hayee, et al., "Performance improvement in high capacity ultra-long distance WDM systems using forward error correction codes," in Proc. Optical Fiber Commun. Conf., vol. 3, 2000, pp. 274-276.
[25] O. A. Sab, "FEC techniques in submarine transmission systems," in Proc. Optical Fiber Commun. Conf., vol. 2, 2001, pp. TuF1-1–TuF1-3.
[26] J. D. Andersen, "Product codes for optical communication," in Proc. ECOC,

Copenhagen, Sept. 2002, vol. 3, pp. 1-2.
[27] J. Yuan and W. Ye, "A novel block turbo code for high speed long-haul DWDM optical communication systems," Optik-International Journal for Light and Electron Optics, vol. 120, pp. 758-764, Oct. 2009.
[28] Y. Li, S. Lee, et al., "A 16Gb 3b/cell NAND flash memory in 56nm with 8MB/s write rate," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 195-207, Jan. 2009.
[29] C. E. A. Trinh, "A 5.6mb/s 64gb 4b/cell NAND flash memory in 43nm CMOS," in Solid-State Circuits Conference - Digest of Technical Papers, 2009, pp. 246-247.
[30] J. Wang, T. Courtade, H. Shankar, and R. Wesel, "Soft information for LDPC decoding in flash: Mutual-information optimized quantization," in Proc. IEEE Global Commun. Conf. (GLOBECOM), 2011, pp.5-9.
[31] C. Yang, Y. Emre and C. Chakrabarti, "Product code schemes for error correction in MLC NAND flash memories," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 12, pp.2302 -2314, 2012.
[32] G. Bosco, G. Montorsi, and S. Benedetto, "Soft decoding in optical systems," *IEEE Trans. Commun.*, vol. 51, no. 8, pp. 1258-1265, Aug. 2003.
[33] T. Mizuochi, "Recent progress in forward error correction and its interplay with transmission impairments," *IEEE J. Select. Topics Quantum Electronics*, vol. 12, no. 4, Aug. 2006.
[34] P. Elias, "Error-free coding," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 29-37, Sept. 1954.
[35] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error correcting coding and decoding: Turbo codes," ICC '93, Conference Record, Geneva, pp. 1064-1070, 1993.
[36] R. M. Pyndiah, "Near optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun*., vol. 46, pp. 1003-1010, Aug. 1998.
[37] S. Haykin, *Communication systems,* 4th ed., John Wiley & Son, 2001.
[38] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed., Upper Saddle 292River, NJ, USA: Pearson Education, 2004.

5. 研究方法

本計畫研究乃針對NB-LDPC碼在解碼時，精確的事前軟式訊息不易從通道上直接獲得之應用(例如：高速光通訊、快閃記憶體，高密度的磁性儲存器等)，開發具有多隨機符元錯誤改錯能力之低複雜度解碼演算法。NB-LDPC碼在軟式判決解碼時所需的事前軟式訊息將由RS碼之偵錯器輸出獲得，稱為虛擬軟式訊息。NB-LDPC碼隨即利用該虛擬軟式訊息進行置信度傳遞(BP)反覆運算解碼。為獲得該虛擬軟式訊息，我們首先需設計混合乘積碼，該混合乘積碼由兩類組成碼串聯組合而成串接碼，以NB-LDPC碼為外碼，RS碼為內碼。研究方法說明如下：(1)設計以NB-LDPC碼為外碼，RS碼為內碼之串接式混合乘積碼。(2)理論推導NB-LDPC碼解碼時所需的虛擬軟式訊息。(3)使用上述所獲得的虛擬軟式訊息作為事前軟式訊息，開發NB-LDPC碼解碼演算法，以有效對抗叢發錯誤及多隨機符元錯誤，並使得在解碼性能與運算複雜度上取得最佳的權衡(trade-off)。(4)進行符元錯誤率性能之模擬分析，獲得最佳的RS碼與NB-LDPC碼之串接組合。(5)針對所設計的NB-LDPC碼的解碼演算法進行運算複雜度分析與改良，並與國內外研究所提出的方法比較。(6)模擬分析與改良所設計的NB-LDPC碼的編碼增益性能，並與國內外研究所提出的方法比較，驗證所提出方法的有效性。(7)本計畫研究所設計之編解碼方法將具有低複雜度與易於硬體實現的特性，因此將針對所提出的混合乘積碼及NB-LDPC解碼演算法進行低複雜度的FPGA硬體電路設計。將研究方法分別說明如下。

5.1 串接NB-LDPC碼與RS碼的混合乘積碼之設計方法

串接NB-LDPC碼與RS碼之混合乘積碼研究方法說明如下。讓(N_R,K_R)RS碼C_R當作內碼，(N_L,K_L) NB-LDPC碼C_L當作外碼，其中N_R,N_L分別表示RS碼與NB-LDPC碼的碼長，K_R,K_L分別表示RS碼與NB-LDPC碼的資訊長度，碼向量的各元素均取值於有限場$GF(q=2^p)$，其中p為訊息符元的位元數；所構成的混合乘積碼$P=C_L\otimes C_R$如圖3所示。混合乘積碼的訊息結構安排如圖4所示。在編碼時，先使用外碼C_L(亦即NB-LPDC碼)，將K_L符元之資訊序列編碼成碼長為N_L的碼字(codeword)。接著將N_L符元長度分為$K_s=\lfloor N_L/K_R \rfloor$段序列，每一段序列由$K_R$資訊符元組成，另將最後一段序列之最後$N_p=K_sK_R-N_L$符元填補為0。$K_s$序列分別使用內碼$C_R$(亦即RS碼)編碼為$K_s$碼字，其碼長均為$N_R$，最後將$K_s$碼字依序傳送。

在接收端解碼時，首先將每一接收到的N_R符元使用內碼C_R(亦即RS碼)解碼產生K_R資訊符元並將其儲存於NB-LDPC解碼暫存器，依序進行K_s序列解碼(或僅由癥狀值(syndrome)所構成的關係式計算獲得每一序列之錯誤符元的個數，作為計算虛擬軟式訊息之參數)，完成後共計N_L符元儲存於NB-LDPC解碼暫存器。接著，將所形成碼長N_L的訊息使用外碼C_L(亦即NB-LPDC碼)解碼，形成K_L資訊符元，最後將解碼完成後之K_L資訊符元傳送給使用者。

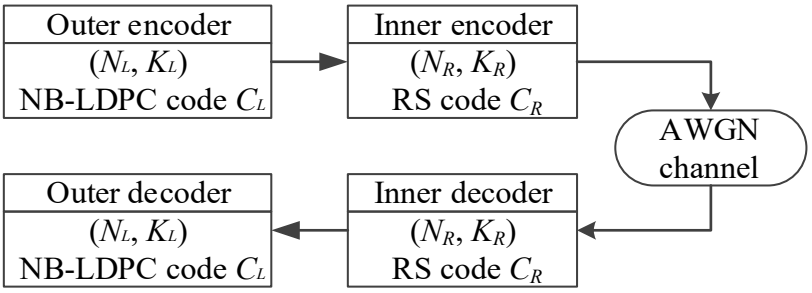

圖 3 串接 NB-LDPC 碼與 RS 碼之混合乘積碼

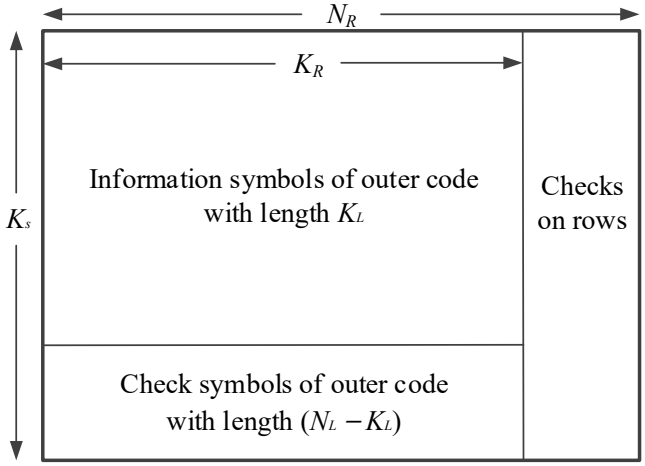

圖 4 混合乘積碼的訊息結構

在每一次解碼時，外碼(NB-LPDC)碼所需的事前軟式訊息(prior soft information)，將由內碼(RS 碼)解碼器(或偵錯器)之輸出(亦即是輸出每一序列之錯誤符元數目)獲得，因該軟式訊息(或稱可靠度訊息)非實際由通道所接收的訊息，故稱為虛擬軟式訊息，其獲得的方法將於下節說明。為利用該虛擬軟式訊息來獲得好的改錯性能，混合乘積碼設計時將選用較短碼長的內碼(RS 碼)，另選用較長碼長的外碼(NB-LDPC 碼)，並採用複雜度較低的軟式判決解碼。在 NB-LDPC 碼解碼的初始步驟，將採用由 RS 碼偵錯器所輸出的虛擬軟式訊息當作每一符元的事前機率(prior probability)。NB-LDPC 碼之 BP 演算法可使用 Tanner 圖表示(如圖 5)，其中軟式訊息 $v_{m,n}$ 及 $u_{m,n}$ 分別表示「檢查節點(check node) m 對變數節點(variable node) n」及「變數節點 n 對檢查節點 m」訊息，該訊息在變數節點與檢查節點之間相互傳遞。圖 5 中 NB-LDPC 碼解碼器之變數節點訊息 v_n，$n = 1+(k-1)K_R,\ 2+(k-1)K_R,\ldots,\ K_R+(k-1)K_R$ 的初始值將設定為第 k 個 RS 碼偵錯器所輸出的虛擬軟式資訊 γ_k，$k = 1,2,\ldots,K_s$。如此，變數節點訊息 V_n，$n = 1,2,\ldots,N_L$ 將被 NB-LDPC 碼解碼器之檢查節點所交錯(如圖 5)，接著即可採用 BP 演算法進行 NB-LDPC 碼的反覆運算解碼。

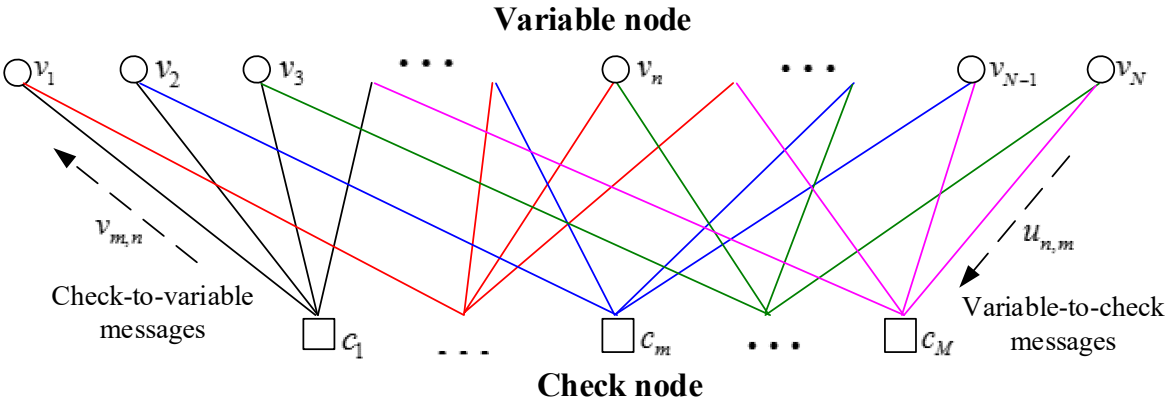

圖 5 NB-LDPC 碼的 Tanner 圖表示法

5.2 虛擬軟式訊息之萃取方法

本節將描述由RS碼偵錯器的輸出，求得虛擬軟式訊息γ的方法，該虛擬軟式訊息將作為NB-LDPC碼每一符元的事前軟式訊息。

考慮$(N_R, K_R, d_{\min})$RS 碼，其中$d_{\min}$表示最小距離(minimum distance)。針對AWGN 通道，訊息符元$\mathbf{s}_n = [s_{1n}, s_{2n}, ..., s_{pn}]$($s_{in} \in GF(2),\ i = 1,2,..., p$)的對數似然函數(log-likelihood function)可表示為[37]：

$$\begin{aligned} l(\mathbf{s}_n) &= \frac{1}{2\sigma^2}\sum_{j=1}^{p}(r_{jn} - s_{jn})^2 \\ &= \frac{1}{2\sigma^2}\left\|\mathbf{r}_n - \mathbf{s}_n\right\|, \quad n = 1,2,..., N_R \end{aligned} \tag{1}$$

其中$\mathbf{s}_n = [s_{1n}, s_{2n}, ..., s_{pn}] \in GF(2^p)$及$\mathbf{r}_n = [r_{1n}, r_{2n}, ..., r_{pn}] \in GF(2^p)$分別表示訊息符元與接收符元，$\left\|\mathbf{r}_n - \mathbf{s}_n\right\|$ 表示訊息符元$\mathbf{s}_n$與接收符元$\mathbf{r}_n$間之 Euclidean 距離。

讓 RS 碼C_R的第 i 個碼字向量表示為$C_i = [\mathbf{s}_{1i}, \mathbf{s}_{2i}, ..., \mathbf{s}_{N_R i}],\ i = 1,2,..., 2^{pK_R}$，而向量$C_i$的元素$\mathbf{s}_{ni} \in GF(2^p),\ n = 1,2,.., N_R$；另接收向量表示為$R = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{N_R}]$。因此碼字$C_i$的對數似然函數可表示為：

$$l(C_i) = \frac{1}{2\sigma^2}\sum_{n=1}^{N_R}\left\|\mathbf{r}_n - \mathbf{s}_{ni}\right\|, \quad i = 1,2,..., 2^{pK_R} \tag{2}$$

使用最大似然(maximum-likelihood)解碼時，最佳的碼字C_i之判決需滿足下列條件：

$$\sum_{n=1}^{N_R}\left\|\mathbf{r}_n - \mathbf{s}_{ni}\right\| < \sum_{n=1}^{N_R}\left\|\mathbf{r}_n - \mathbf{s}_{nj}\right\|, \quad \text{for } j \in \left\{1,2,..., 2^{pK_R}\right\} \text{ and } j \neq i \tag{3}$$

因此碼字C_i的可靠度，可定義為對數似然比(log-likelihood ratio, LLR)：

$$\Lambda \equiv \ln\left(\frac{P(R \mid C_i)}{P(R \mid \overline{C}_i)}\right) \tag{4}$$

其中$\overline{C}_i$表示最接近於C_i的碼字，亦即是$\overline{C}_i$與C_i之間的 Euclidean 距離$d(C_i, \overline{C}_i)$在所有碼字中為最短。

應用 Bayes 法則並假設於C_R中之所有碼字為均勻分配 (uniformly distribution)，則(4)式可推導得到：

$$\Lambda \equiv \ln\left(\frac{P(C_i \mid R)}{P(\overline{C}_i \mid R)}\right) \quad (5)$$

讓$D(A,B)$表示碼字 A 與碼字 B 之間不同符元的數目。假設接收向量 R 的錯誤符元數目為 e，$e \in \{0,1,...,\lfloor (d_{\min}-1)/2 \rfloor\}$。根據三角不等式，$D(R,\overline{C}_i) \geq d_{\min} - e$，在較高的訊號雜訊比條件下，等號成立的機率相當高，因此我們使用下限 $D(R,\overline{C}_i) = d_{\min} - e$，以求得碼字的可靠度 Λ。讓 P_e 表示發送符元在接收端產生錯誤的機率，則(5)式之碼字可靠度 Λ 可表示為：

$$\begin{aligned}\Lambda &= \ln\left(\frac{(P_e)^e(1-P_e)^{N_R-e}}{(P_e)^{d_{\min}-e}(1-P_e)^{N_R-(d_{\min}-e)}}\right) \\ &= (d_{\min}-2e)\ln\left(\frac{1-P_e}{P_e}\right)\end{aligned} \quad (6)$$

假設使用 q-ary QAM (quadrature amplitude modulation)調變傳送訊息，符元在接收端產生錯誤的機率 P_e 可表示為[37]：

$$P_e \approx 4\left(1-\frac{1}{\sqrt{q}}\right)Q\left(\sqrt{\frac{3E_{av}}{(q-1)N_0}}\right) \quad (7)$$

其中 E_{av} 表示符元的平均能量，(以下推導合理假設 $E_{av}=1$)，$q=2^p$，雜訊功率頻譜密度 $\sigma^2 = N_0/2$。Q-函數可定義為 $Q(z) \equiv \int_z^\infty (1/\sqrt{2\pi})e^{-y^2/2}dy$，其上限可表示為[38]，$Q(z) \leq (1/2)e^{-z^2/2}$。我們將 Q-函數上限應用於(7)式，可獲得

$$P_e \approx 2\left(1-\frac{1}{\sqrt{q}}\right)e^{\frac{-3E_{av}}{2(q-1)N_0}} \quad (8)$$

將(8)式代入(6)式，可獲得

$$\begin{aligned}\Lambda &= (d_{\min}-2e)\ln\left(\frac{1-2\left(1-\frac{1}{\sqrt{q}}\right)e^{\frac{-3}{2(q-1)N_0}}}{2\left(1-\frac{1}{\sqrt{q}}\right)e^{\frac{-3}{2(q-1)N_0}}}\right) \\ &\approx \frac{3(d_{\min}-2e)}{4(q-1)\sigma^2}\end{aligned} \quad (9)$$

在(9)式的推導簡化中，我們假設 $\sigma^2 = N_0/2 \to 0$。

假設考慮穩態通道(stationary channel)，碼字可靠度 Λ 可利用常數 $\sigma^2/2$ 將其正規化。因此，依據(9)式，碼字中每一符元的平均可靠度可表示為：

$$\begin{aligned}\gamma &= \frac{1}{N_R}\frac{\sigma^2}{2}\Lambda \\ &= \frac{1}{N_R}\frac{3(d_{\min}-2e)}{8(q-1)}\end{aligned} \quad (10)$$

其中 e 可由 RS 碼偵錯器所接收到符元的癥狀值所構成的關係式計算而得。然後

依據(10)式計算所獲得的γ值當作虛擬軟式訊息，該虛擬軟式訊息即可提供作為NB-LDPC 碼每一符元的事前軟式訊息。

以上推導之(10)式為通式，當將虛擬軟式訊息γ應用於二元場時，(10)式中的$q=2$，而$\gamma=(3/8N_R)(d_{\min}-2e)$。我們可串接二元 LDPC 碼與 BCH 碼之混合乘積碼，而進行 LDPC 碼之 BP 演算法(例如：SPA, MSA(min-sum algorithm)等)解碼。

5.3 NB-LDPC碼的解碼演算法之設計方法

近年來，解碼演算法的改進主要針對三個方面，第一：解碼性能的提昇，經過改進的解碼演算法可以在同樣的信號雜訊比條件下大幅降低誤碼率；第二：運算複雜度的降低，經過改進後的演算法更加易於實現，演算法的研究大部分最後都是要得到實際的應用，高複雜度的演算法在實際應用中沒有競爭力，因此低複雜度的演算法是實際應用所迫切需要的；第三：解碼速度的提高，高解碼速度隱含著較高的數據輸送量，這在現今寬頻數位化時代顯得更為重要。NB-LDPC 碼的解碼演算法針對這三方面均受到研究者的重視並有很好的研究成果，例如從 BP 演算法到 EMS 演算法[15]，計算複雜度被大幅降低，在降低複雜度的同時需保證解碼性能沒有損失；再比如從 EMS 演算法到分層的 EMS 演算法，解碼速度又被大幅提昇，而運算複雜度僅少許的增加，運算複雜度的少許增加與解碼速度的提昇相比是可以忽略的。另文獻[19]提出減少複雜度的 Min-Max 演算法，其具有兩種 LLR 實現方式，一種是標準實現方式，它的複雜度與有限場元素的個數平方成正比；另一種是選擇性實現方式，該實現方式的複雜度比於標準實現方式更低，因此使得 Min-Max 演算法在實際應用上具有很強的競爭力。

本計畫研究主要在充分利用分層解碼演算法與 Min-Max 演算法的優點，設計一種高效率的 NB-LDPC 解碼演算法，在保證解碼性能的同時可有效降低解碼複雜度，更重要的是該演算法將可大幅提昇解碼速度。本研究的解碼演算法將針對 Min-Max 演算法進一步改良，Min-Max 演算法中複雜度最高的部分是對校驗節點的處理，在文獻[21]中提出的 forward-backward 演算法簡化了這一過程，使解碼得到了簡化因而更易於實現，forward-backward 演算法雖然簡化了校驗節點訊息的計算，但是為了得到最終的校驗節點訊息，必須將 forward-backward 演算法的計算結果存儲下來，如此將需要大量的存儲空間，所佔用的存儲空間隨著有限場元素以及奇偶檢查矩陣 **H** 的行重(column weight)之增加而增加，因此對於高階以及高行重之奇偶檢查矩陣，存儲空間將成為限制應用的主要因素。改進方法為，針對 forward-backward 校驗節點處理方法，可僅選取了一部分最可靠的訊息來參與 forward-backward 處理過程，這就使得需要的計算量可大幅降低，且可大量減少存儲空間。為便於描述本計畫提出的 NB-LDPC 解碼演算法之研究方法，先就 Min-Max 演算法與分層解碼方法做簡要介紹，接著說明本計畫研究之改良 Min-Max 解碼演算法。

(1) Min-Max 演算法

在有限場上的 NB-LDPC 碼的奇偶檢查矩陣中絕大多數元素都是 0，非零元素所占的密度很小，這些非零元素可由有限場的本原元素(Primitive element)得到。在解碼過程中，可以將反覆運算過程看成在 Tanner 圖的邊上不斷傳遞可靠度訊息，傳遞的訊息的多寡是由場的階數大小決定，階數越大，傳遞的訊息越多，所以對資料的處理更加複雜。對於階數為 q 的有限場，需要用 q 個可靠度訊息來表示各個元素的可靠度大小。在對數域 Min-Max 演算法中，符元的可靠度訊息可用 LLR 表示：$L(\alpha)=\ln(P(\beta)/P(\alpha))$，其中 β 是可能性最大的符元，α 可以是有限場中 q 個元素之一，LLR 值可以用來衡量一個符元與可靠度最高的符元間的距離，值越小則距離越小。

當計算出 q 個 LLR 後，為便於計算與表示，可將這 q 個 LLR 用向量表示，為此定義了下述的幾種向量。如圖 5，以 m 表示校驗節點，n 表示變數節點，n 傳遞到 m 的 LLR 向量用 $u_{m,n}$ 表示，相應的 m 傳遞到 n 的 LLR 向量用 $v_{m,n}$ 表示；與 n 關聯的 m 的集合以 $S_c(n)$ 表示，與 m 關聯的 n 集合以 $S_v(m)$ 表示；以 $\mathcal{L}(m|a_n=\alpha)$ 表示滿足限制條件 $\sum_{j\in S_v(m)\backslash n} h_{mj}a_j = h_{mn}\alpha$ 的場元素 $a_j(j\in S_v(m)\backslash n)$ 的序列集合，其中 h_{ij} 是奇偶檢查矩陣 **H** 中第 i 列第 j 行的元素。以 γ_n 表示接收端得到的事前軟式訊息，其代表了變數節點 n 取某個值的原始可能性大小，在實際情況是傳輸後的訊息 γ_n 已經被通道雜訊干擾，所以需要經過計算來解碼。Min-Max 解碼演算法的步驟如下[19]：

初始化：

$$u_{m,n}(\alpha)=\gamma_n(\alpha) \quad (11)$$

反覆運算過程：

1) 校驗節點更新：

$$v_{m,n}(\alpha)=\min_{a_j\in\mathcal{L}(m|a_n=\alpha)}\left(\max_{j\in S_v(m)\backslash n} u_{m,j}(a_j)\right) \quad (12)$$

2) 變數節點更新：

$$u'_{m,n}(\alpha)=\gamma_n(\alpha)+\sum_{i\in S_c(n)\backslash m} v_{i,n}(\alpha) \quad (13)$$

$$u_{m,n}(\alpha)=u'_{m,n}(\alpha)-\min_{w\in GF(q)} u'_{m,n}(w) \quad (14)$$

3) 事後訊息(posteriori information) 更新：

$$\tilde{\gamma}_n(\alpha)=\gamma_n(\alpha)+\sum_{i\in S_c(n)} v_{i,n}(\alpha) \quad (15)$$

在進行變數節點更新時，(14)式的計算主要在保證訊息的穩定性，並且使得一個 LLR 向量中的最小值永遠是 0。在每次反覆運算之後進行判決，直到正確解碼或達到了規定的最大反覆運算次數為止。因為在(12)式中低估了 LLR 值，所以 Min-Max 演算法是次佳的，因此可以利用調整因子或偏移量來對該低估進行補償。在校驗節點更新時，要想得到滿足限制條件 $\sum_{j\in S_v(m)\backslash n} h_{mj}a_j = h_{mn}\alpha$ 的場元素的序列集合是很複雜的，為了降低複雜度，[21]文獻研究提出 forward-backward 演算法，該演算法避開了直接計算 $\mathcal{L}(m|a_n=\alpha)$，因此簡化了解碼過程。

(2) 分層解碼方法

分層解碼方法被分為行分層解碼方法和列分層解碼方法，在行分層解碼方法中，LDPC 碼的奇偶檢查矩陣 **H** 以「行」為基準被分為若干層，稱之為行層，訊息的更新以行層為基礎一列接一列進行；列分層解碼方法與行分層解碼方法類似，奇偶檢查矩陣 **H** 以「列」為基準被分為若干層，訊息的更新以列層為基礎一列接一列進行。研究結果說明，行分層解碼方法的收斂速度與列分層解碼方法接近，但是行分層解碼方法的運算複雜度要高於列分層解碼方法，因此在這裡主要介紹列分層解碼方法。

當 NB-LDPC 碼具有如圖 7 所示的結構化特徵時，依據行數將 **H** 矩陣分成若干層，例如對於矩陣 $\mathbf{H}_{15\times25}$ 共有 15 列，那麼每 5 列為一層，此矩陣可被分為 3 層，其劃分的方式需經由設計而得。對於規則性 LDPC 碼，Tanner 圖上每一層的校驗節點均與所有的變數節點有連接，基於此，校驗節點及變數節點的訊息更新可以一層接一層的進行，具體而言就是首先更新這一層的校驗節點訊息，然後再更新這一層的所有變數節點訊息，這樣的一次運算作為一次完整反覆運算的子運算，一層接一層地進行下去，當最後一層被處理過後，一次完整的反覆運算才算完成。從上一層得到的校驗節點到變數節點的訊息馬上被應用到下一層的變數節點到校驗節點的資訊計算中，在每一次校驗節點到變數節點訊息更新過後計算符元的事後機率，因此可導致事後機率更加頻繁地更新，而得到更高的收斂速度。

Layer 1	$\mathbf{I}^2$	$\mathbf{I}^4$	$\mathbf{I}^2$	$\mathbf{I}^3$	$\mathbf{I}^2$
Layer 2	$\mathbf{I}^1$	$\mathbf{I}^3$	$\mathbf{I}^4$	$\mathbf{I}^2$	$\mathbf{I}^1$
Layer 3	$\mathbf{I}^2$	$\mathbf{I}^3$	$\mathbf{I}^1$	$\mathbf{I}^4$	$\mathbf{I}^3$

$$\mathbf{I}^1 = \begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$$

圖 7 分層解碼的矩陣特徵

一般而言，對於那些 **H** 中的某一層有空的子矩陣(元素全為 0)的情況，分層解碼同樣可行。但是過多的空子矩陣可能會使解碼性能降低，這是因為空矩陣所對應的變數節點沒有從每次的訊息更新中受益。現以 EMS 演算法[15]為例，說明分層解碼方法的解碼過程如下。NB-LDPC 碼的非分層的 EMS 演算法校驗節點更新演算法如下：

$$v_{m,n}^{(k)}(\alpha) = \min_{\substack{(a_j)_{j\in S_v(m)} \\ \in \hat{\mathcal{L}}(m|a_n=\alpha)}} \sum_{j\in S_v(m)\setminus n} u_{m,j}^{(k-1)}(a_j) \quad (16)$$

其中 $u_{m,n}^{(k)}(x_n)$ 表示第 k 次反覆運算變數節點 n 傳遞到校驗節點 m 的訊息，$v_{m,n}^{(k)}(\alpha)$ 表示的是第 k 次反覆運算從校驗節點 m 傳遞到變數節點 n 的訊息，以 $\gamma_n(\alpha)$ 表示從通道得到的第 n 個節點的訊息，以 $\tilde{\gamma}_n(\alpha)$ 表示第 k 次反覆運算後第 n 個變數節點的事後訊息，則 $u_{m,n}^{(k)}(\alpha)$ 和 $\tilde{\gamma}_n(\alpha)$ 的計算方法如下：

$$u_{m,n}^{(k)}(\alpha)=\gamma_n(\alpha)+\sum_{i\in S_c(n)\backslash m} v_{i,n}^{(k)}(\alpha) \quad (17)$$

$$\tilde{\gamma}_n^{(k)}(\alpha)=\gamma_n(\alpha)+\sum_{m\in S_c(n)} v_{m,n}^{(k)}(\alpha) \quad (18)$$

EMS 演算法應用分層解碼方法時，用變數 l 代表當前解碼演算法所在的層數，例如用 $\tilde{\gamma}_n^{(k,l)}(\alpha)$ 表示第 k 次反覆運算在第 l 層得到的節點事後訊息，如此校驗節點的訊息更新被修改為如下：

$$v_{m,n}^{(k,l)}(\alpha)=\min_{\substack{(a_j)_{j\in S_v(m)}\\ \in\mathcal{L}(m|a_n=\alpha)}}\sum_{j\in S_v(m)\backslash n}\left(\tilde{r}_j^{(k,l-1)}(a_j)-v_{m,j}^{(k-1,l)}(a_j)\right) \quad (19)$$

分層 EMS 演算法的變數節點訊息更新方法與非分層 EMS 演算法相同，但是分層 EMS 演算法的節點事後訊息計算可被簡化為：

$$\tilde{\gamma}_n^{(k,l)}(\alpha)=\tilde{\gamma}_n^{(k,l-1)}(\alpha)-R_n^{(k-1,l)}(\alpha)+R_n^{(k,l)}(\alpha) \quad (20)$$

其中

$$R_n^{(k,l)}(\alpha)=\sum_{m\in S_c^l(n)} v_{m,n}^{(k,l)}(\alpha) \quad (21)$$

其中 $S_c^l(n)$ 表示 Tanner 圖上連接到變數節點 n 的校驗節點中屬於第 l 層的部分。在反覆運算開始前，$v_{m,n}^{(0,l)}(\alpha)$ 及 $R_n^{(0,l)}(\alpha)$ 被初始化為 0。

分層解碼方法除可應用到上述的 EMS 演算法外，也可應用於 BP 演算法及 Min-Max 演算法。

(3) 本計畫研究之改良 Min-Max 解碼演算法

如上述分層解碼不僅改善了解碼性能，而且使解碼速度得到了很大的提升。因此本研究的 NB-LDPC 碼的解碼演算法將充分應用 Min-Max 演算法及分層解碼方法的優點，並加以改良，具體的解碼過程描述如后。如圖 8 所示，對於行重為定值 d_c 的 NB-LDPC 碼，在 Tanner 圖上，每一個校驗節點將會有 d_c 個變數節點與之相連接，每一個變數節點傳遞給校驗節點的資訊都是一個 q 維向量，將每一個變數節點稱為一階(stage)，每一階都由 q 個訊息點組成，因此總共有 $d_c\times q$ 個訊息點，從這 $d_c\times q$ 個訊息點中找出前 s 個最小 LLR 值的訊息點，並從小到大排列(如圖 8 之紅點部分)。在 Min-Max 演算法中，(14)式的運算可以得出每一階會有 LLR 值為 0 的訊息點，如此在找出 s 個最小 LLR 值的訊息點後，在某一階上可能只有一個 LLR 值為 0 的訊息點，這意味著這一階上的其它訊息點對應的 LLR 值都很高，因此對這一符元的硬式判決具有很高可靠性，在此我們將這樣的階稱為 zero stage(如圖 8 之*號部分)，其它的訊息點的階稱為 nonzero stage。獲得 zero stage 及 nonzero stage 以後，forward-backward 演算法可以被進一步簡化，假設第 i 階是 zero stage，那麼當 forward 過程前進到第 i 步時，由於計算時僅涉及 LLR

值為 0 的訊息點，如果用 α_i^0 表示第 i 階 LLR 值為 0 的訊息點所對應的場元素值，那麼 max 操作可以被忽略，因為 $\max(f_{i-1}(\alpha'),0)$ 一定是 $f_{i-1}(\alpha')$，同時 min 操作也不再需要，因為 α_i^0 與每一 $\alpha' \in GF(q)$ 的和均不同，所以 $f_i(\alpha)$ 可以簡單地等於 $f_{i-1}(\alpha'+\alpha_i^0)$，因此 f_i 可以當成是 f_{i-1} 的一個置換(permutation)，該置換是由 α_i^0 決定。

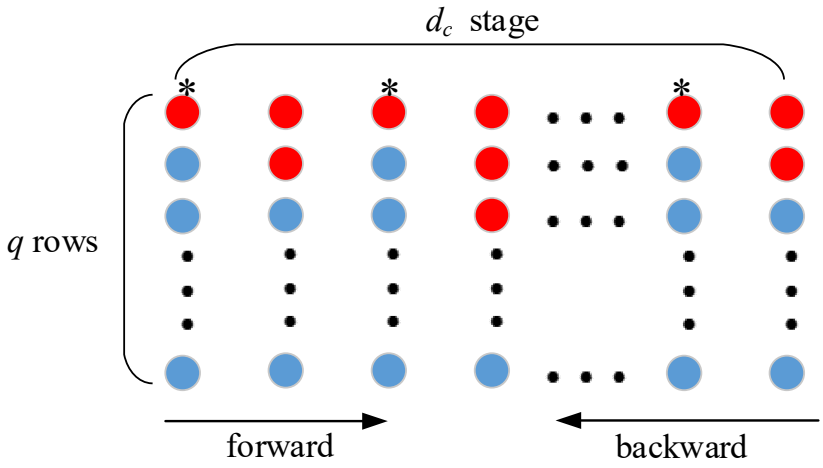

圖 8 NB-LDPC 碼校驗節點與變數節點的圖形表示

假定 forward 過程再向前處理，如此當計算得到 f_{d_c} 後，f_{d_c} 即包含所有圖 8 中所有紅點的 Min-Max 的結果，在計算(12)式 $v_{m,n_i}(\alpha)$ 時，即不再需要第 i 階的訊息，對於 $\alpha=\alpha'+\alpha_i^0$ 的 Min-Max 結果，直接就可以利用 $v_{m,n_i}(\alpha)=f_{d_c}(\alpha'+\alpha_i^0)$ 計算得出，因此對於那些 zero stage 的校驗節點到變數節點訊息可以非常容易地通過 f_{d_c} 計算得出，對應的中間結果 f_{i-1} 和 b_{i+1} 也不再需要存儲。由以上分析可知，中間結果存儲的減少有賴於 zero stage 的多少，為了降低複雜度，減少需要存儲的中間結果，我們可以設置一個最大值 t，用 t 來代表 nonzero stage 的個數，假定 LLR 值不為 0 的訊息點被從小到大排列，它們屬於第 $x_0,x_1,\ldots$ 階，如此前 t 個不同的 x_i 可以被找到，這 t 個 nonzero stage 就是最終被留下的 nonzero stage，因此就可進一步減少需要存儲的中間結果。forward 和 backward 過程進行時所使用的仍然是那些被選出的 s 個訊息點，然而只有那 t 個 nonzero stage 所對應的中間訊息被存儲下來，而且在產生校驗節點訊息時，只有 nonzero stage 對應的變數節點依照原方法計算，其它的 zero stage 則依據 f_{d_c} 就可以計算出校驗節點訊息。因此，需要存儲的訊息量由原來方法的 $2(d_c-1)$ 個訊息向量減少為 $2t+1$ 個訊息向量。

為便於描述改良 Min-Max 解碼演算法，我們將 **H** 矩陣分為 L 層，每一層有 r_l 列，第 k 次反覆運算第 l 層校驗節點到變數節點的訊息用 $v_{m,n}^{(k,l)}$ 來表示；用 $u_{m,n}^{(k,l)}$ 代表第 k 次反覆運算第 l 層變數節點到校驗節點的訊息；用 $\tilde{\gamma}_{m,n}^{(k,l)}$ 表示第 k 次反覆運算第 l 層後所得到的事後訊息；用 Ω_{n_i} 表示從變數節點選取的 s 個最小值中，在第 n_i 階所佔有的部分。對第 0 層、第 1 層、…、第 L-1 層的變數節點和校驗節點都連續處理一遍後，即完成了一次完整的反覆運算過程。

改良 Min-Max 解碼演算法步驟如下：

步驟 1：初始化：

計算 $\gamma_n(\alpha)$，$u_{m,n}(\alpha)=\gamma_n(\alpha)$。對所有的 m 及所有 $n\in S_v(m)$，令 $v_{m,n}(\alpha)=0$。

步驟 2：預先處理：

對於每個 m，找出 $u_{m,n}(\alpha)$ 的前 s 個最小值，找出 t 個 nonzero sage，標記每一階的 LLR 值為 0 處的 α 值。

步驟 3：反覆運算過程：

1) 對於第 k 次反覆運算的第 l 層，計算 $v_{m,n}^{(k,l)}(\alpha)$，其中 forward-backward 計算方法如下：

$$\begin{cases} f_1(\alpha)=u_{m,n_1}^{(k,l)}(\alpha) \\ f_i(\alpha)=\min\limits_{\{\alpha'+\alpha''=\alpha,\alpha''\in\Omega_{n_i}\}}\left(\max(f_{i-1}(\alpha'),u_{m,n_i}^{(k,l)}(\alpha''))\right),\ 1\le i\le d_c \end{cases} \quad (22)$$

$$\begin{cases} b_{d_c}(\alpha)=u_{m,n_{d_c}}^{(k,l)}(\alpha) \\ b_i(\alpha)=\min\limits_{\{\alpha'+\alpha''=\alpha,\alpha''\in\Omega_{n_i}\}}\left(\max(b_{i+1}(\alpha'),u_{m,n_i}^{(k,l)}(\alpha''))\right),\ 1\le i\le d_c \end{cases} \quad (23)$$

由此可得 $v_{m,n}^{(k,l)}(\alpha)$ 為：

$$\begin{cases} v_{m,n_1}^{(k,l)}(\alpha)=b_2(\alpha) \\ v_{m,n_{d_c}}^{(k,l)}(\alpha)=f_{d_c}(\alpha) \end{cases} \quad (24)$$

對於 $1\le i\le d_c$，如果第 i 階是 zero stage，則

$$v_{m,n_i}^{(k,l)}(\alpha)=f_{d_c}(\alpha+\alpha_i^0) \quad (25)$$

其中 α_i^0 代表的是 zero stage 的 α 值，如果第 i 階是 nonzero stage，則

$$v_{m,n_i}^{(k,l)}(\alpha)=\min_{\alpha'+\alpha''=\alpha}\left(\max(f_{i-1}(\alpha'),b_{i+1}(\alpha''))\right) \quad (26)$$

2) 對於第 k 次反覆運算第 l 層，計算事後訊息：

$$\tilde{\gamma}_n^{(k,l)}(\alpha)=\tilde{\gamma}_n^{(k,l-1)}(\alpha)+v_{m,n}^{(k,l)}(\alpha)-v_{m,n}^{(k-1,l)}(\alpha),\ (l-1)\cdot r_l\le m\le l\cdot r_l \quad (27)$$

其中 r_l 為第 l 層的列數。

3) 若解碼成功或達到最大反覆運算次數則終止反覆運算，否則計算變數節點的訊息，並對 $u_{m,n}^{(k,l+1)}$ 繼續進行預先處理，處理方法同步驟 2。

$$\begin{aligned} u'^{(k,l+1)}_{m,n}(\alpha)&=\tilde{\gamma}_n^{(k,l)}-v_{m,n}^{(k,l)}(\alpha) \\ u_{m,n}^{(k,l+1)}(\alpha)&=u'^{(k,l+1)}_{m,n}(\alpha)-\min_{w\in GF(q)}\left(u'^{(k,l+1)}_{m,n}(w)\right) \end{aligned} \quad (28)$$

6. 結果與討論

為了驗證所提出方法的有效性，我們使用由 (N_L, K_L) LDPC 碼與 (N_B+1, K_B) eBCH 碼所構成的串接碼進行模擬，所提出的 LDPC-PSI 性能將以位元錯誤率(bit error rate (BER))及每一次解碼之平均迭代次數(average number of iterations (ANI))進行評估。

模擬時採用(1023, 781) EG-LDPC 碼串接(16, 7)eBCH。LDPC 碼之解碼 NMSA [4]，IMWBF [7] 及 PWBF [8] 演算法 圖 3 和圖 4 分別說明使用接受到的軟式資訊的 LDPC 解碼演算法與所提出的 LDPC-PSI 演算法的 BER 性能及 ANI 比較。圖中顯示 LDPC-PSI 演算法的性能與使用接受到的軟式資訊的 LDPC 解碼演算法的性能大致相同，例如在，LDPC-PSI 演算法與使用接受到的軟式資訊的 LDPC 解碼演算法的性能相比，僅相差 0.1-0.2dB。然而 LDPC-PSI 結合 IMWBF 演算法(LDPC(IMWBF)-PSI)可快速的收斂，與使用接受到的軟式資訊 IMWBF 演算法相比，每一次解碼的 ANI 可減少 40% 至 50%。

為更進一步說明 LDPC-PSI 演算法與使用 HIHO 解碼的 TPC 演算法之性能比較。我們使用(4095, 3367)EG-LDPC, eBCH $(32, 26, 4)^2$ Closed-chains TPC (CCTPC) [20] 和 eBCH $(32, 26, 4)^2$ 標準 HIHO (TPC) 解碼器比較。BER 性能比較顯示於圖 5。圖中說明所提出的方法和 CCTPC 及 TPC 比較，在 BER=10^{-6} 所提出的方法分別可獲得 0.6dB 和 1.7dB 的解碼增益。

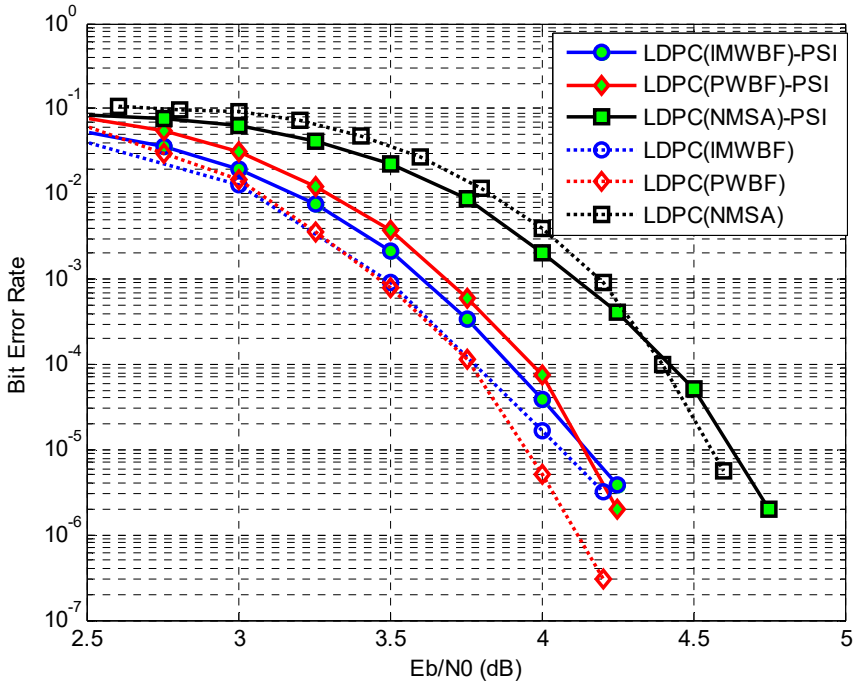

圖 3 使用接受到的軟式資訊的 LDPC 解碼演算法與所提出的 LDPC-PSI 演算法的 BER 性能比較

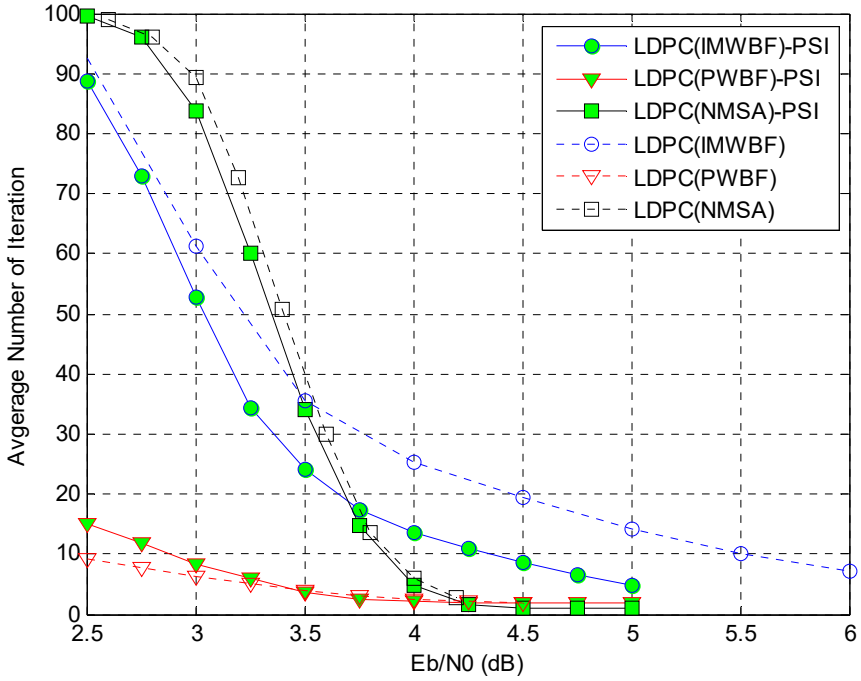

圖 4 使用接受到的軟式資訊的 LDPC 解碼演算法與所提出的 LDPC-PSI 演算法的 ANI 性能比較

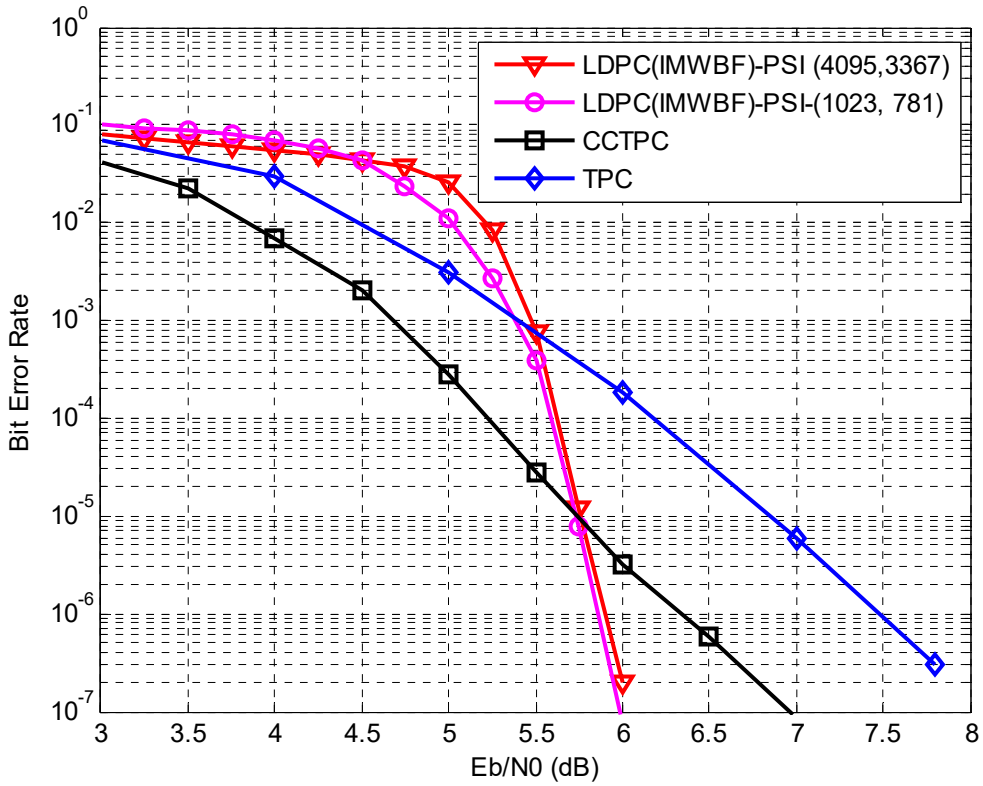

圖 5 LDPC-PSI 演算法與使用 HIHO 解碼的 TPC 演算法之性能比較

由於LDPC碼的錯誤控制效能非常接近Shannon限界，同時具有更快的解碼速度、較低的誤碼平層、具有平行且疊代的架構、解碼器簡單適合硬體的實現等優點。目前在深空通訊、無線區域網路(IEEE 802.11n)、Wimax (IEEE 802.16e)、衛星數位視訊廣播標準(DVB-S2)等系統中皆被採用，因此在未來有關LDPC碼的應用將更為廣泛。本計畫研究主要針對NB-LDPC碼在解碼時，事前軟式訊息不易從通道獲得的應用，提出兼顧低複雜度與高編碼增益性能之解碼演算法的理論設計與硬體實現，其研究成果在學術研究與實際應用等方面將有實質貢獻，主要包括：

(1) 本計畫設計串接 NB-LDPC 碼與 RS 碼之混合乘積碼及使用虛擬軟式訊息於 NB-LDPC 碼之解碼演算法，設計完成的混合乘積碼之編解碼器，將具有低複雜度與高編碼增益性能，易於實現且可達到即時性的傳輸需求，因此將可推展至目前及未來的超高速無線通訊系統使用。

(2) 本計畫設計針對事前軟式訊息不易從通道獲得的應用，開發具有抗叢發錯誤和多隨機符元錯誤能力之 NB-LDPC 碼的解碼演算法，將應用至高速光通訊、快閃記憶體，高密度的磁性儲存器等系統，是目前錯誤控制碼研究之重要議題。

(3) 藉由本計畫對 NB-LDPC 解碼演算法之研究，開發符合低複雜度與即時性需求的 FPGA 硬體電路，技轉給民間產業，開發無線通訊、光通訊、快閃記憶體，磁性儲存器所需的相關產品，提昇國家產業競爭力。