

行政院國家科學委員會專題研究計畫成果報告
採用重覆性解碼的短區塊碼編碼設計
A short Block coding Scheme Using Iterative Decoding

計畫編號：NSC 91-2213-E-002-101

執行期間：91年8月01日至92年7月31日

主持人：林茂昭 國立臺灣大學電信所

一、中文摘要

在以前之研究中，我們使用兩個乘積碼直和 (direct sums of product codes) 的編碼方式，來設計具大自由距離並適合於重覆性解碼的短區塊碼。建構的方法是設計兩組相同編碼長度但彼此互斥之乘積碼 C_1 及 C_2 ，再將兩組乘積碼進行以 2 為基底的加法運算之編碼方式，亦即 $C = C_1 \oplus C_2$ 。藉由設計兩組具備不同錯誤更正能力之乘積碼的組合，可以創造出比傳統乘積碼具有更大自由距離之區塊碼，此編碼方式並可對不同類型信息位元作不均等保護 (Unequal Error Protection)，而達到可調式傳輸效果，使整體資料流通量最大。

在本研究中，針對先前所提出之編碼架構，我們分別應用 Chase Variant 2 “軟式輸入/軟式輸出”解碼演算法 (algorithm) 及 Sum Product 演算法等解碼技術的觀念，試圖研究適合之重覆性解碼的方法來解碼，並跟傳統渦輪乘積碼 (turbo product code) 作比較。我們同時分析在這種編碼架構下，進行重覆性解碼所遭遇之問題，並提出未來繼續研究之方向。

關鍵詞：乘積碼，重覆性解碼，區塊碼，渦輪乘積碼

英文摘要

In an earlier research, we use the concept of the direct sums of product codes to design a short block code by taking direct sums of product codes. The minimum distance of such codes are greater than those for the simple product codes of comparable dimensions, besides, some message bits

have extra error protection.

In this research, we apply the chase Variant 2 decoding algorithm and sum product algorithm techniques to design iterative decoding schemes for the direct sums of product codes.

Keywords: Product Code, Iterative Decoding, Block Code, Turbo Product Code

二、計畫的緣由與目的(Goals)

在無線通訊時，傳輸通道通常不甚理想，因此必須採用適當的錯誤更正碼以維持傳輸的可靠度。而目前大部分的無線通信系統，如語音之傳輸，對延遲參數非常重視，也就是要求低複雜度及較短編碼長度的編碼系統。

應用重覆性解碼所設計的編碼系統，其除錯性能非常優越，目前已有學者研究使用渦輪碼或其相關碼來設計較短的區塊碼，例如渦輪乘積碼 (turbo product code) [5] 及低密度對偶檢查碼 (low-density parity-check codes) [2-3] 等，但是在複雜度、時間延遲及可靠度的考量之下，仍有很大的改善空間。

在以前研究中 [1]，我們提出使用兩個乘積碼直和的編碼方式，成功創造出比傳統乘積碼具有更大自由距離之區塊碼，並可對不同類型信息符元作不均等保護，使整體資料流通量最大。

由於我們以往在區塊碼及重覆性解碼有相當程度研究，本計畫將以此編碼架構為基礎，設計適合於重覆性解碼的短區塊碼，並研究適合之“軟式輸入/軟式輸出”及重覆性解碼的方法，以求出最

佳的除錯性能，較低的複雜度及較短編碼長度的編碼系統。

三、研究方法與成果 (Methods and Results)

我們設計之乘積碼直和之編碼架構如圖一所示，其中對於第一個乘積碼 $C_1=V_1 \otimes W_1$ ，我們選定 V_1 為 (31,21) BCH 碼，其生成多項式 (Generation Polynomial) 之八進位表示為 [3 5 5 1]， W_1 為 (15,1) 重覆碼；對於第二個乘積碼 $C_2=V_2 \otimes W_2$ ，我們選定 V_2 為 (31,21) BCH 碼，其生成多項式之八進位表示為 [2 3 0 3]， W_2 為 (15,10)，其生成多項式之八進位表示為 [6 5]。

因為 $W_1 \cap W_2 = \{0\}$ ，所以其相對應之 $V_1 \otimes W_1$ 及 $V_2 \otimes W_2$ 兩組乘積碼集合亦彼此互斥。如此設計之編碼系統為一個 (465,231) 碼，其編碼長度 (block length) 為 465，訊息位元長度為 231，整體碼率為 0.497。由 [1] 中，我們知道在這種編碼架構下，對於 C_1 及 C_2 乘積碼之個別信息位元而言，其最小自由距離分別為 21 及 20，亦即當收到之軟性通道資訊 (Soft Channel Values) 之錯誤個數 $\leq \left\lfloor \frac{21-1}{2} \right\rfloor = 10$ ，可以正確解出 C_1 乘積碼之信息位元，而錯誤個數 $\leq \left\lfloor \frac{20-1}{2} \right\rfloor = 9$ ，可以正確解出 C_2 乘積碼之信息位元。

由於所收到的軟性通道資訊 r 為「兩個乘積碼的值和」加上「加成性白色高斯雜訊 (AWGN)」，在應用傳統區塊碼之重覆性解碼技術前，我們必須先計算出 C_1 及 C_2 乘積碼個別之軟性通道資訊 r_1 及 r_2 。為瞭解所設計出之具大自由距離之編碼系統是否優於傳統渦輪乘積碼，及探討進行重覆性解碼的可行性。首先，假設我們可以事先獲得 C_1 乘積碼之前置輸入資訊 (A Priori Values) $L(C_1)$ ，由 [2-3] 中，我們可以利用 r 及 $L(C_1)$ 推導出 C_2 之軟性通道資訊 r_2 ，公式如下：

$$r_2 = f(r, L(C_1))$$

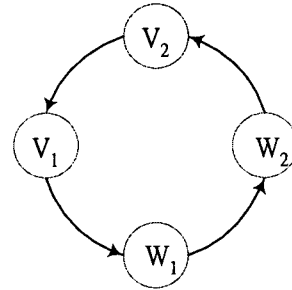
其中

$$f(x, y) = -\text{sgn}(x) \cdot \text{sgn}(y) \cdot \Phi(|\Phi(|x|) + \Phi(|y|)|)$$

$$\Phi(x) = \log \frac{e^x - 1}{e^x + 1}$$

在獲得 C_2 之軟性通道資訊 r_2 後，我們即可以使用傳統 Chase 解碼技術來計算 C_2 之軟性輸出 $L(\hat{C}_2)$ 及非本質資訊 (Extrinsic Values) $L_e(\hat{C}_2)$ 。其中 C_2 之軟性輸出資訊 $L(\hat{C}_2)$ 可用來計算 C_1 乘積碼之軟性通道資訊 $r_1 = f(r, L(\hat{C}_2))$ ，非本質資訊在每一次遞迴解碼後得到，並且在下次遞迴解碼被視為前置輸入資訊。詳細重覆性解碼之系統方塊圖如圖 2 所示。

我們採用之整個遞迴解碼順序如下圖所示：



亦即我們先解 C_2 乘積碼中之 W_2 然後 V_2 ，再解 C_1 乘積碼中之 V_1 然後 W_1 ，之後我們再繼續遞迴解 C_2 及 C_1 。

圖 3 為在執行第一次疊代時，假設 C_1 乘積碼之前置輸入資訊已知的前提下，其除錯性能對疊代次數 (iteration number) 做圖，其中我們稱執行一次 $W_2 \rightarrow V_2 \rightarrow V_1 \rightarrow W_1$ 之解碼過程為一個疊代次數。由模擬結果顯示，我們發現在這種重覆性解碼方法下，除錯性能會隨者疊代次數增加而逐漸收斂，而且執行 5 次疊代就已經足夠。

圖 4 是同樣在假設 C_1 乘積碼之前置輸入資訊已知的前提下，比較所提出之乘積碼直和架構和傳統渦輪乘積碼的除錯性能，圖 4 中所參考比較之傳統乘積碼為 $(32, 21, 5)^2$ ，其編碼長度為 1024，碼率為 0.431，由模擬結果分析，我們發現所設計之編碼系統略優於傳統渦輪乘積碼，而且我們所設計之編碼長度較短且碼率較高。

圖 5 為在執行第一次疊代時， C_1 乘積碼之前

置輸入資訊未知下，對其除錯性能做圖。由模擬結果顯示，我們發現在這種情況下，其除錯性能並不理想。這是由於在降低解碼複雜度考量下，我們使用 Chase Variant 2 之軟式解碼方法，但在中低訊雜比下，第一次疊代時估計 C_2 之軟性通道資訊 r_2 之錯誤率較高，造成 Chase Variant 2 軟式解碼方法以所接收之軟性通道資訊較小者較容易出錯之假設不適用，因此使用 Chase Variant 2 軟式解碼方法在中低訊雜比下會造成瓶頸。我們相信若採用 Chase Variant 1 軟式解碼方法，將可以提高其除錯性能，但相對其複雜度亦大幅提高。

另外，我們也研究使用 Sum Product Algorithm 的觀念來進行解碼，我們分別事先求出 $V_1 \otimes W_1$ 及 $V_2 \otimes W_2$ 之系統化生成矩陣(Systematic Generator Matrix) G_1 及 G_2 ，也就是使矩陣中對應到資料的那些列的每一列只有一個 1 存在，而其他部份為 0，則整體 $V_1 \otimes W_1 \oplus V_2 \otimes W_2$ 之生成矩陣為

$$G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

再透過列運算 (Row Operation) 來把它的生成矩陣系統化

$$\bar{G} = [I | P].$$

則其相對應之檢查矩陣(parity check matrix)為

$$H = [P^T | I].$$

唯經觀察求得之檢查矩陣 H ，我們發現其每行(列) 1 個數的分佈極不平均，部分行僅包含 1 個 1，部分行包含 1 的個數超過 100 個，因此在疊代效能及複雜度考量下，並不適合使用[2-3]中之 sum product 演算法。另透過適當的列運算來平均 1 的分佈，這種列運算之處理方式應是一種具挑戰性之工作，相信即可使用 sum product 演算法來進行解碼。

四、結論與討論(Concluding Remarks)

在本研究中，我們利用上述之方法，創造出比傳統乘積碼具有更大自由距離之區塊碼，並提出適合之重覆性解碼的方法來解碼。由模擬結果分

析，我們發現在假設執行第一次疊代時， C_1 乘積碼之前置輸入資訊已知的前提下，所提出之乘積碼直和架構優於傳統渦輪乘積碼的除錯性能。但在 C_1 乘積碼之前置輸入資訊未知下，其除錯性能並不理想，所以，剛開始如何找到 C_2 的軟性通道資訊 r_2 ，仍然是一個有待繼續研究的問題。我們相信採用 Chase Variant 1 及 Sum product algorithm 是一個值得繼續研究之方向。

五、參考文獻(References)

- [1] Mao-Chao Lin and Shu Lin, "Codes with Multi-Level Error Correcting Capabilities, " *Discrete Mathematics*, 83, pp.301-314, 1990.
- [2] D. J. C. MacKay and M. C. Davey, "Two small Gallager codes, " Available at <http://wol.ra.phy.cam.ac.uk/mackay/>, July 1999.
- [3] D. J. C. MacKay and M. C. Davey, "Evaluation of Gallager codes for short block length and high rate application, " to appear in proceedings of IMA workshop on Codes, Systems and Graphical Models, Aug 1999.
- [4] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 170-182, Jan. 1972.
- [5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding decoding: Turbo-Code(1)," *IEEE Int. Conf. On Communications*, 1993.

六、圖表 (Figures and Tables)

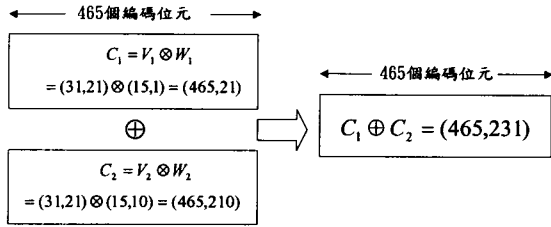
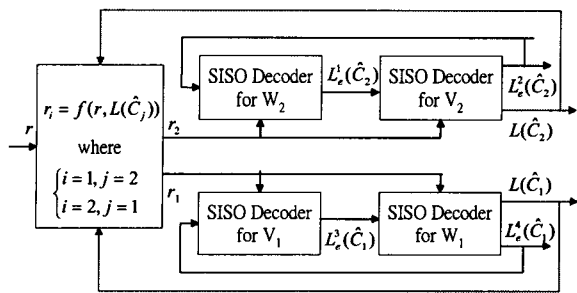


Figure 1: Encoder structure for direct sums of product codes Turbo



$L_e(\hat{C}_i)$: extrinsic values for C_i
 $L(\hat{C}_i)$: a posteriori values for C_i

Figure 2: Decoding scheme for direct sums of product codes

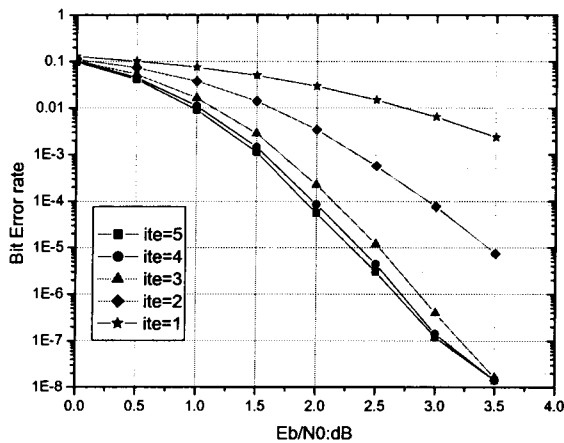


Figure 3: The BER versus the number of decoding iterations for the proposed codes, decoding is based on Chase-2 Algorithm. A Priori Values $L(C_1)$ is used at the first iteration. The number of test patterns is 16. (Code Rate=0.497, block length=465, Ite: iteration number.).

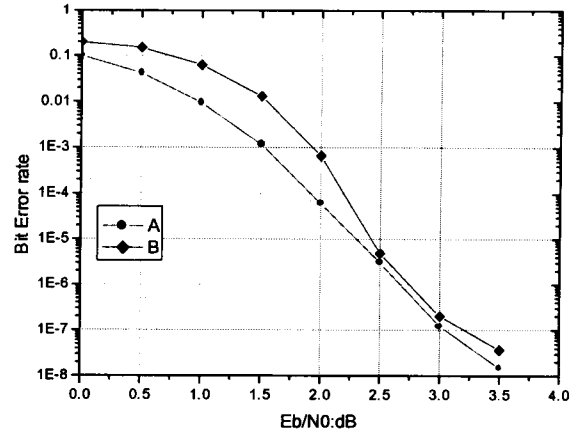


Figure 4: Comparisons between the proposed codes and conventional Turbo product codes (TPC), decoding is based on Chase-2 Algorithm. A Priori Values $L(C_1)$ is used at the first iteration. The number of test patterns is 16, and the number of iteration is 5.

(A) Proposed code, block = 465, R=0.497.
 (B) TPC: $(32, 21, 5)^2$, block = 1024, R=0.431

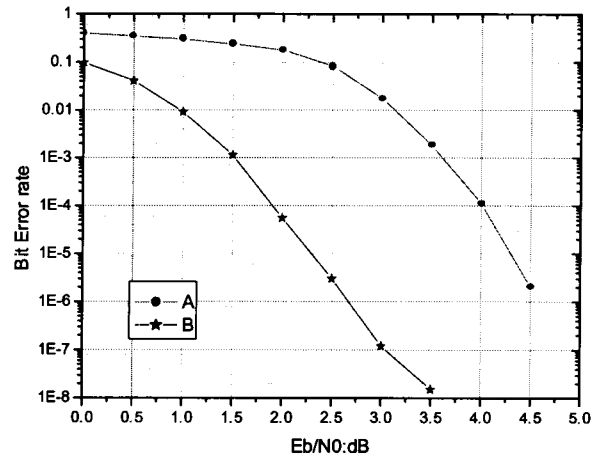


Figure 5: The BER of the proposed codes, decoding with and without the a Priori Values $L(C_1)$. Decoding is based on Chase-2 Algorithm. The number of test patterns is 16, and the number of iteration is 5. (Code Rate=0.497, block length=465).
 (A) A Priori Values $L(C_1)$ is used at the first iteration.
 (B) A Priori Values $L(C_1)$ is not used at the first iteration.