以動態平衡樹為基礎的線性區塊碼解碼研究

Based on Dynamic Balance Tree Decoding Scheme for Linear Block Code

洪偉文　林義楠

Wei-Wen Hung　Yi-Nan Lin

摘要

線性區塊碼解碼方式在傳統上有幾種比較著名的演算法，如：Berlekamp-Massey 等演算法。這些演算法主要是在可更正錯誤的能力範圍下，以解聯立方程式來求出碼字。在經過傳輸通道後，某些位元被雜訊干擾下，接收端經硬決策後，被判斷出錯誤的訊息，而這些演算法乃解出其錯誤位元的位置，而依解出的錯誤位置，來加以更正還原成原來傳送的資料。本文是架構在最大相似解碼的基礎上，發展出一套以動態二元樹為基礎的解碼機制，只要在 O(k)次的比較判斷及運算下(k：字符訊息位元數)，即可解出該碼字是合法的還是錯誤的。在與暴力搜尋法 $O(2^k)$及以上述的演算法需反覆地求解方程式的根來解碼是來得比較有效率的，唯需要有 2^k 個標準碼字的儲存空間。

關鍵詞：線性區塊錯誤更正碼、動態平衡二元樹、最大相似度解碼

ABSTRACT

There are several decoding methods using linear block codes have been developed for many years; one well-known method is Berlekamp-Massey algorithm. These algorithms can correct error bits which are corrupted by the channel noises by solving joint equations and then find out the error bit positions. Finally, the mechanism can get original transmitted source message by summation of received message and error pattern in the last stage. A new method based on maximal likelihood decoding is proposed in which the dynamic binary search, i.e., AVL-tree is used to develop a decoding scheme. This decoding mechanism can determine the received codeword is legal or not with time complexity about approximate O(k) (k: message symbol length) comparison time. Comparing with the full search method which needs O(2k) and above algorithms these need iterations to solve the joint equations, and then gets the result answer. The new proposed method is much more efficient, but it needs O(2k) storage device to save the standard codewords.

Keywords： linear block error code, dynamic balance binary tree, maximal likelihood decoding

1 簡介

錯誤更正碼主要是用來保護數位的訊息在傳輸的過程中，受到外來雜訊的污染下，能適時地予以更正其錯誤位元的一種機制[1]。錯誤更正碼在傳送的訊息中，加入了查核位元；即為編碼，而這些查核的位元則可以在接收端解碼時，發揮其錯誤偵測及錯誤更正的效果。由提高可靠性的角度來看，錯誤控制更正碼是衛星通訊、行動通訊、衛星廣播、文字廣播領域不可缺少的技術，而對於數位的資料儲存系統而言，也可採用錯誤保護的機制來對儲存的資料加以保護，可進一步地提高其資料的可靠度。目前，錯誤控制更正碼的技術已成功地應用在藍芽無線通訊系統[2]、個人無線通訊系統之 AMPS、NA-TDMA、CDMA、GSM [3]及 CD 資料儲存系統[4]等方面。

以下將介紹本文相關的章節內容，第二節線性區

洪偉文　明志科技大學電子工程系教授
林義楠　明志科技大學電子工程系講師

塊碼的編碼與其徵狀解碼的方式，第三節動態平衡二元樹基本原理，及以此樹為基礎所提出的解碼演算法。第四節做一結論與未來相關研究領域的探討。

2 線性區塊碼

線性區塊碼主要是將欲傳送的訊息加入一些冗餘的查核位元。而這些查核位元的目的除了可以使得編出來的碼字均勻地分佈在整個碼字的空間外，更可以用來校正收到的碼字是否為一合法的碼字，以達到錯誤偵測或錯誤更正的效果。

2.1 編碼機制

在線性區塊碼中，有一些比較著名的碼。如：BCH 碼、RS 碼及 RM 碼[1]。今考慮具單一錯誤訂正能力的(7, 4)BCH 碼，此即(7, 4)漢明碼，其碼字長度為 7 個位元，其中訊息的長度為 4 個位元，因此其查核位元則需要 3 個，且其最小的漢明距離為 $d_{min} = 3$。由錯誤更正能力 $t = [(d_{min} - 1) / 2]$及錯誤偵測能力 $e = d_{min} - 1$。因此，具有偵錯 2 個位元錯誤或是更正 1 個位元錯誤的能力。由(n, k)線性方塊碼的產生原理，若原始的訊息向量為 **M** 其維度為 1 × k，即有 k 個位元的訊息量要進行編碼，其與一個 k × n 的生成矩陣 **G**，透過內積的運算，可得到一個長度為 n 個位元的碼向量 **X**，亦即 **X** = **M**·**G**。又若生成矩陣 **G** 具有[**P** | $\mathbf{I_K}$]的結構時，其中 $\mathbf{I_K}$ 是 k×k 的單位矩陣，而 **P** 是 k × (n - k) 的子矩陣，則可運算出一個具有前 k 個位元為輸入編碼器的訊息位元串，及後半段為 n - k 個查核位元串分開的系統區塊碼(systematic block code)，即 $\mathbf{X} = [m_1, m_2, \ldots, m_k, c_1, c_2, \ldots, c_{n-k}]$。針對生成矩陣 **G** 可產生一系統方塊碼 **X**。

若 $\mathbf{G} = \begin{bmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{bmatrix}_{4\times7} = [\mathbf{P} \mid \mathbf{I_4}]$

表 1：(7, 4)線性區塊系統碼：**X** = **M**·**G**

M	**X**	w(**X**)	$(\mathbf{X})_{10}$
0000	0000000	0	0
0001	0001011	3	11
0010	0010110	3	22
0011	0011101	4	29
0100	0100111	4	39
0101	0101100	3	44
0110	0110001	3	49
0111	0111010	4	58
1000	1000101	3	69
1001	1001110	4	78
1010	1010011	4	83
1011	1011000	3	88
1100	1100010	3	98
1101	1101001	4	105
1110	1110100	4	116
1111	1111111	7	127

其中 w(**X**)：碼字的權重，即碼字的漢明權重；$(\mathbf{X})_{10}$：碼字的十進位值。

2.2 徵狀(syndrome)解碼

當發送端傳送某一個碼字 **X**，在二位元對稱通道傳輸模式下，接收端收到向量 **Y** 時，若 **Y** ≠ **X**，則表示傳輸過程中，因雜訊的干擾而造成了錯誤位元的發生。

對於任一(n, k)線性系統方塊碼，其生成矩陣為 $\mathbf{G} = [\mathbf{P} \mid \mathbf{I_k}]_{k\times n}$ 時，則存在一個(n - k)×n 的正交對應的同位檢驗矩陣 $\mathbf{H} = [\mathbf{I_{(n-k)\times n}} \mid \mathbf{P^T}]$，此同位檢查矩陣的特性為：若 **Y** 為一合法的碼字時，則 $\mathbf{YH^T} = [0\ 0\ \ldots\ 0]_{1\times(n-k)}$；反之，若 **Y** 不為一合法的碼字時，則 $\mathbf{YH^T} \neq [0\ 0\ \ldots\ 0]_{1\times(n-k)}$。故在解碼的過程中，接收端一收到向量 **Y**，則先計算 $\mathbf{S} = \mathbf{YH^T}$，此 **S** 稱為 **Y** 的徵狀。若 **S** ≠ **0** 時，則表示傳輸的過程中，有錯誤的發生。反之 **S** = **0**，則 **Y** = **X**，即表示傳輸的過程中沒有錯誤發生；或者 **Y** 可能解碼為另一個合法的碼字而形成了無法偵測的錯誤。

根據錯誤位元的最大發生機率，則可找出錯誤型態向量(**E**)與徵狀向量(**S**)間之一對一的關係。故在計算出徵狀向量後，即可進一步地計算出錯誤發生的位置，再將此錯誤型態的向量(**E**)與接收的向量(**Y**)相加，即可解出一合法的碼字。

3 以平衡樹為基礎的線性區塊碼之解碼機制

3.1 高度平衡樹(AVL-tree) [5]

是在 1962 年由 G. M. Adelson-Velskiinhe、E.

M. Landis 二位數學學者所提出的。它限定了左、右子樹的生長高度差必須維持在 1 範圍之內，所動態建立出的一種高度平衡二元樹。其建立的動態表格，在執行各種新增、刪除及搜尋的動作，均可維持在 $O(\log_2 k)$的時間內完成。因 AVL-tree 的高度 h 為 $\log_2(k+1) \le h \le 1.44\log_2(k+2) - 0.328$；其中 k 為資料節點的個數。故可保證在最壞的情形下，搜尋只需花費 $O(\log_2 k)$的比較次數。所以，用它來製作搜尋的索引，在 k 個記錄下的檔案，需有 k 個索引。此時，AVL-tree 的最大高度約為 $1.4\log_2 k$。今若有 10^6 個約有 2^{20} 次方大小的記錄時，則最多只需要 $1.4\log_2(2^{20}) \approx 28$ 次的比較，即可完成所需鍵值的查詢。

3.2 建立以 AVL-tree 為基礎之解碼二元樹

以(7, 4)漢明碼為例，其真正合法的碼字只有 16 個，即在可能的 2^7 為 128 個 7 位元的傳輸向量中，其中只有 16 個才是真正合法的碼字。在 k = 4 個位元的訊息，經由生成矩陣(**G**)的運算下，即可得到如表 1 n = 7 個位元的碼字向量。以本文所提出之以平衡樹為基礎的最大相似解碼方式為例，其所建立出之 AVL-tree 為一棵(7, 4)線性區塊碼的動態平衡二元樹，如圖 1 所示。

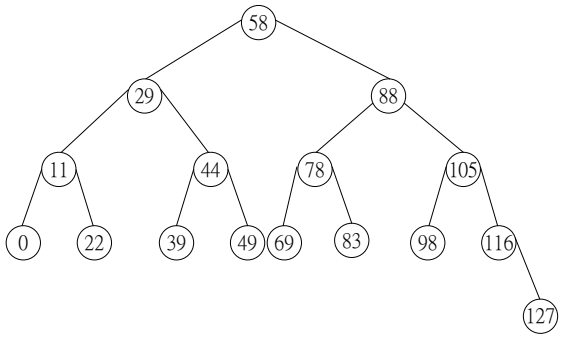

圖 1. (7, 4)線性區塊碼的 AVL-tree

3.3 以 AVL-tree 為基礎之最大相似解碼演算法

本文主要提出之解碼機制如下：

Step1.當接收端收到的碼字訊息 **R**，經由一個硬決策，即令 Ri ≥ 0 時，則 Yi = 1；否則 Yi = 0，即可得到一個二元的碼字向量 **Y**。

Step2.將此二元的碼字向量 **Y**，轉成其對應的十進制的鍵值 V。

Step3.搜尋此鍵值 V，是否存在於此動態二元樹中。由樹根的節點值開始進行比較，當鍵值 V 不等於拜訪到的節點值 Ci 時，則計算二個節點值的漢明距離 d = (**Y**, **Xi**)，當 d ≤ 1 時，即可立刻解碼出碼字為 **Xi**，解碼完成。否則當 d > 1 時，則需比較 V 與 Ci 的大小，當 V 大，則往右分支走；若 V 小，則往左分支拜訪。一直遞迴拜訪下去，即重複步驟 3，直到拜訪到樹葉節點時，仍未解出最佳的 **Xi** 時，即表示可偵測出錯誤。此時，可通知對方重新再發送一次相同的訊息。

例 1：若接收到為經過硬決策後的碼字向量 **R** = [0.1 -0.2 -0.3 0.5 0.6 0.4 -0.8]時，則利用本文提出的解碼方式，其解碼步驟如下：

(1). **Y** = [1 0 0 1 1 1 0]

(2). V = 78

(3). 因 V ≠ 58，則計算 d = (**Y**, **Xi**) = 4 > 1 且 V > 58，則往右分支拜訪。

(4). 因 V ≠ 88，則 d = (**Y**, **Xi**) = 3 > 1 且 V < 88，則往左分支拜訪。

(5). V = 78，則解碼完成，其碼字為[1 0 0 1 1 1 1 0]且訊息為 1 0 0 1。

此碼字之解碼只需花費 3 次的比較，即完成解碼。

例 2：若接收到為經過硬決策後的碼字向量 **R** = [0.1 0.2 0.3 0.5 0.6 0.4 0.8]時，則利用本文提出的解碼方式，其解碼步驟如下：

(1) **Y** = [1 1 1 1 1 1 1]

(2) V = 127

(3). 因 V ≠ 58，則 d = (**Y**, **Xi**) = 3 > 1 且 V > 58，則往右分支拜訪。

(4). 因 V ≠ 88，則 d = (**Y**, **Xi**) = 4 > 1 且 V > 78，則往右分支拜訪。

(5). 因 V ≠ 105，則 d = (**Y**, **Xi**) = 3 > 1 且 V > 98，則往右分支拜訪。

(6). 因 V ≠ 116，則 d = (**Y**, **Xi**) = 3 > 1 且 V > 116，則往右分支拜訪。

(7). 因 V = 127，則結束解碼。

故此碼字之解碼共花費了 5 次的比較，即此樹的最大的高度。

4 結論

傳統使用徵狀解碼時，必須先有計算徵狀的電路；然後，利用解出的徵狀，來解出其對應的錯誤位置，這是一個聯立方程式的解。因此，在電路的設計與實現上有其複雜度，且其求解的過程必須經過反覆地疊代來求解其根。因此，也需耗費相當的時間。

本文提出的以動態平衡二元樹為基礎的線性區塊碼解碼方式，也是利用最小的漢明距離來求解其對應碼字。因此，也是一種最大相似度的解碼方式。只不過，此解碼的方式比較適合在具有計算能力的裝置上來實現。其優點為其最大的解碼時間為 O(k)加上接收向量碼字與合法碼字間漢明距離的計算；k 為原始訊息位元數，即可完成一個碼字的解碼。不過，此演算的機制需要有 2^k 個合法碼字的儲存空間，但隨著 VLSI 技術的進步，儲存空間較不是成本的考量重點；反而，時間才是較主要的考量因素。

當碼字的長度隨之增長時，為了減少碼字的搜尋，則可將以 2-way 為基礎的動態平衡二元樹擴展為以平衡的 m-way 搜尋樹為基礎的 B-tree。如此，即可更進一步地降低其平均搜尋次數達到 $O(\log_m N)$；其中 $N = 2^k$，此部份將做為後續研究的方向。

5 誌謝

非常感謝所有編審者，由於你們的修改及寶貴意見的提供，使得本文可以更臻完善。

6 參考文獻

[1] S. Lin and D. J. Costello Jr., “Error Control Coding: Fundamentals and Applications,” NJ: Prentice Hall, 1983.

[2] “Specification of the Bluetooth System,” v1.0 A, July 1999.

[3] D. J. Goodman, “Wireless Personal Communications Systems,” Addison-Wesley Longman, 1997.

[4] K. A. S. Immink, “Coding Techniques for Digital Recoeders,” UK: Prentice Hall, 1991.

[5] R. F. Gilberg, B. A. Forouzan., “Data structures: a pseudo code approach with C++,” Brooks/Cole, 2001.