

GSM(或數位式)蜂巢系統中數據傳輸速率加快之設計

The Potential Improvement for Higher Data Rate Transmission in GSM (or The Third Generation) Cellular System

計畫編號: NSC 89-2213-E-034-002

執行期限: 88年8月1日至89年7月31日

計畫主持人: 劉宗慶 中國文化大學電機工程學系副教授

壹、中文摘要

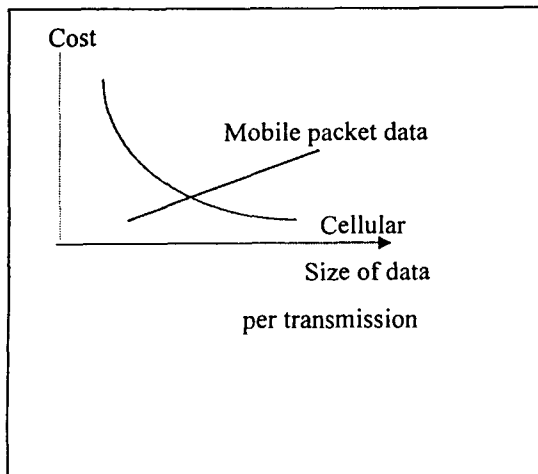
如何提昇 GSM 系統上數據傳送的速率一直為熱門之話題, 本研究針對目前 GSM 架構探討高速(或多重速率)的 $(n-1)/n$ Punctured Convolutional Code 應用的可能性, 亦即在不改變頻寬的條件下, 加快數據傳送的速率。此相同的理論亦可應用於其他數位蜂巢系統或未來第三代的系統。設計的實例輔以多重模擬, 並和理論性能曲線比較, 以找出最佳可應用之 Punctured Convolutional Code。

貳、英文摘要

To increase data transmission rate in GSM or even in the future third generation cellular system is in great demand. The proposed method will emphasize but not limit to the current running GSM system. We use higher n of $(n-1)/n$ convolutional code design with puncturing technique for different rate. Multiple simulations of the designed structures justify the theoretical analysis.

參、計畫緣由與目的

目前行動數據通信的方式有兩種, 包括封包交換式(Packet-switched), 與線路交換式(Circuit-switched)。前者為非同步性質(Asynchronous), 如 CDPD 和 DataTac 等; 後者為等時性(Isochronous), 如 AMPS 及 GSM 的數據服務。圖一顯示封包交換式與線路交換式中價格與數據傳輸量的關係圖 [1]。當每次交換量少時, 封包交換式較為經濟, 但資料量大於某個級數時, 經由適當的設計, 線路交換式則較為經濟。例如 CDPD 及 DataTac 的傳輸速率為(19.2K), 在 E-AMPS 內可達到 33.6K 甚至 56K。GSM 為目前最為流行的數位式蜂巢系統, 著重於語音通話的服務, 兼附簡訊功能, 如何增大其數據傳輸的速率, 將可增高其效益。

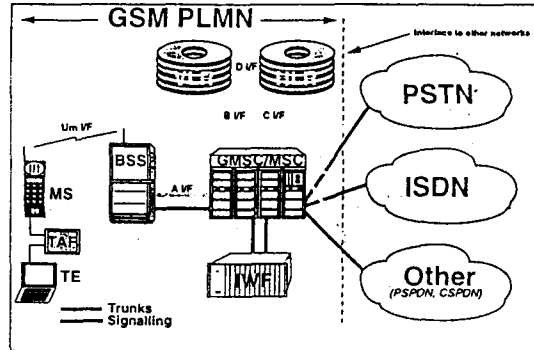


圖一、封包交換式與線路交換式中價格與數據傳輸量的關係圖。X 軸為每次傳送的資料量, Y 軸為價格。

肆、研究方法與步驟

一、簡介

GSM 網路架構, 如下圖二所示:



圖二、GSM 網路架構圖

使用者設備包含:

- 行動台(Mobile Station / MS)
- 終端轉接功能(Terminal Adaption Function / TAF)
- 終端設備(Terminal Equipment / TE)

網路節點包含:

- 基地台系統(Base Station System / BSS)
- 訪問者位置記錄器(Visitor Location Register / VLR)
- 原始位置記錄器(Home Location Register / HLR)
- 行動服務交換中心(Gateway / Mobile Switch Center / GMSC/MS)
- 介接功能(InterWorking Function / IWF)

結合以上設備即稱為 GSM 行動網路(GSM PLMN)。而在下列表一中為 GSM 使用者設備於不同連接型態的數據傳輸速率 [2]:

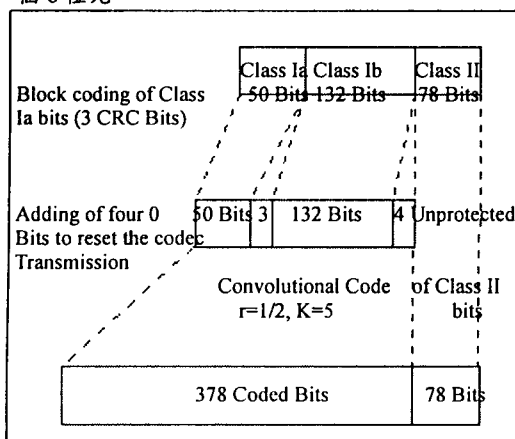
連接型態 (Connection Type)	服務品質 (Quality of Service)	延遲時間 (Delay(two-way))
TCH/F9.6 T	Low	330 ms
TCH/9.6 NT	High	>330 ms
TCH/F4.8 T	Medium	330 ms
TCH/F2.4 T	Medium	200 ms
TCH/H4.8 T	Low	600 ms
TCH/H4.8 NT	High	>600 ms
TCH/H2.4 T	Medium	600 ms

表一、GSM 的數據傳輸連接型式

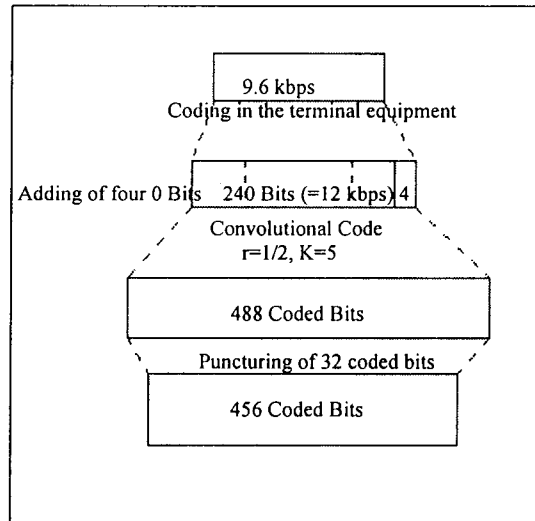
註:TCH表 Traffic Channel, F表 Full Rate, H表 Half Rate, TCH/F 9.6/4.8/2.4 表數據傳輸的速率於 9.6/4.8/2.4 kbps (Fax 資料和電腦資料)。應用的例子如筆記型電腦的 PCMCIA 埠上網。TCH/H4.8/2.4 為半速率傳輸數據, T-mode (Transparent)表誤碼改正完全是以 FEC(Forward Error Correction) 方式處

理，同時 TAF 和 GSM 網間傳輸是同步的。NT-Mode(Non-transparent)表框錯誤(Frame Error)的改正使用再傳送方式(Retransmission)，且採用 RLP(Radio link Protocol)的操作以改正 FEC 所無法改正的錯誤。就 GSM 的 Channel Coding 而言，GSM 用戶數據通道(data channel)的編碼相當複雜，數據資料需較代表語音資料的 Traffic channel data 更為保護。原因是如一位小數點的誤差即將造成交易上的巨大損失；而在最差的情況下，語音資料的一個位元的錯誤，僅造成一個語音框的遺失，Speech Transcoder 將重覆前一語音框，語音的傳輸品質並不會改變太多。因保護的需要，編碼的架構決定了數據資料傳輸的快慢。於表一中每一種 data rate 使用不同的 interleaving 的架構與參數的 Convolution 碼。其中 TCH/F9.6 T 是最高且應用最多的，因此我們將針對此連接型式做一詳述。雖然 9.6kbps 是使用者的傳輸速率，但因加碼後在 TE(Terminal Equipment)端實際的速率是 12kbps，即傳送到行動台的速率是 12kbps。因它和 GSM 無關，是用來在有線環境中錯誤偵測之用，TE 中的編碼不在此討論。GSM 使用修正的 V.110 protocol (ISDN) V110'，此決定碼框架構。

圖三和圖四所示為語音資料傳輸於 TCH/F9.6 kbps 的 Block 和 Convolution 碼的架構[3]。於圖三中，使用者的資料被分成 4 個區段，每一個區段 60 個位元，共 240 個位元，一起被 Convolutional code 編碼。其中因 CRC 誤碼偵測已在 TE 內完成，並沒有被加於 Convolutional code 之前，4 個零位元用來重置(reset)解碼器，因此總位元數為 244 位元。Convolutional 碼的參數為：碼率(Coding rate) $r=1/2$ ，Constraint length $K=5$ ，因此編碼後變為 488 位元。但由圖二可知 488 位元將放不下 456 位元的語音資料區段，將有 $488-456=32$ 位元的多餘位元。因此，488 位元被依某一法則丟掉(Punctured)其中 32 位元，即其中的 32 位元並不傳送，但 convolutional 碼能將丟掉的 32 位元於接收端再還原。因數據的資料比語音的資料更重要，為了抵抗 fading，數據資料的重置構造比語音資料的重置更為複雜。區段展沿 22 個 bursts 因 456 個位元不能均分裝於 22 個 bursts。複雜的架構排定其中 16 個是 24 位元、2 個 18 位元、2 個 12 位元和 2 個 6 位元。



圖三、Full-rate 語音的編碼架構(convolution code)



圖四、9.6 kbps 數據傳輸的編碼架構

我們將 puncture 的數目增加，以增快 data 的速率，其照造成 coding gain 的下降，可由增長 constrain length K 來補償

二、Punctured Convolutional Code

1. Punctured Convolutional Code 之產生方式：

Punctured Convolutional Code 之產生方式為週期性地將 Convolutional code 打掉(puncture)某些輸出位元(output bit)，以 $r=(n-1)/n$ 的碼為例，可用 $r=1/2$ 的碼來產生。例如若要產生 $r=2/3$ 的碼，可利用 $r=1/2$ 的碼，每輸入 2 個資料位元(data bits)，輸出 4 個輸出位元，將第 4 個輸出位元打掉，只剩下 3 個輸出位元，此即成為一個 $r=2/3$ 的碼。在見附錄的表二中列出了每種 $r=(n-1)/n$ 的碼之 puncture 方式[5]，括弧內之八進位值表示 $r=1/2$ 的碼的產生器(generator)架構(例(5)表 $1+D^2$)。

2. Punctured Convolutional Code 之實用價值：

在 Viterbi 解碼方式中，一個 $r=(n-1)/n$ 的碼在進入每個狀態(state)時有 2^{n-1} 條路徑，而一個 $r=1/n$ 的碼則只有 2 條路徑。所以在解碼過程中， $r=(n-1)/n$ 的碼在每個狀態時將做 2^{n-1} 進位的比較，而一個 $r=1/n$ 的碼則只需 2 進位的比較。故為了降低 $r=(n-1)/n$ 的 Punctured Convolutional Code 在解碼時的複雜度，可以將其表示成 $r=1/2$ ($R=1/n$)的碼，利用的方法為在編碼時打掉的位置上填入特定的碼元。如此不但解碼的效益沒有減少且降低了解碼的複雜度。以 Punctured Convolutional Code 的技術而加快傳輸速率的情形如表三：

Coding Rate	傳輸速率(kbps)
1/2	9.6
2/3	12
3/4	12.8
4/5	13.2
5/6	13.44
6/7	13.6
7/8	13.71
8/9	13.8
9/10	13.87

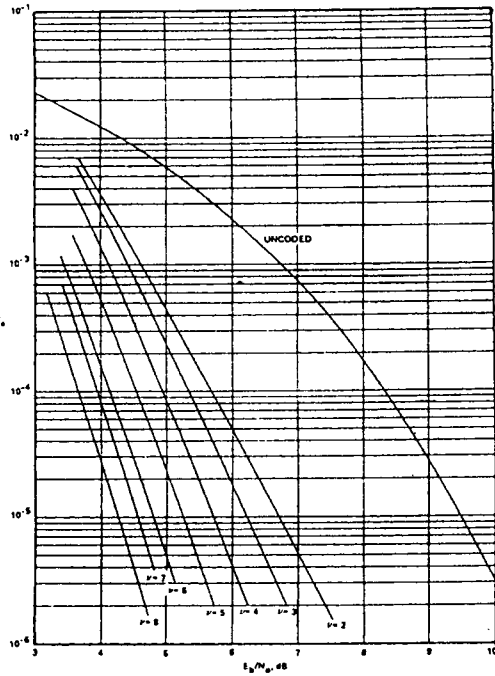
表三數據傳輸速率與 $n-1/n$ Punctured Convolutional Code

3. Punctured Code 之性能

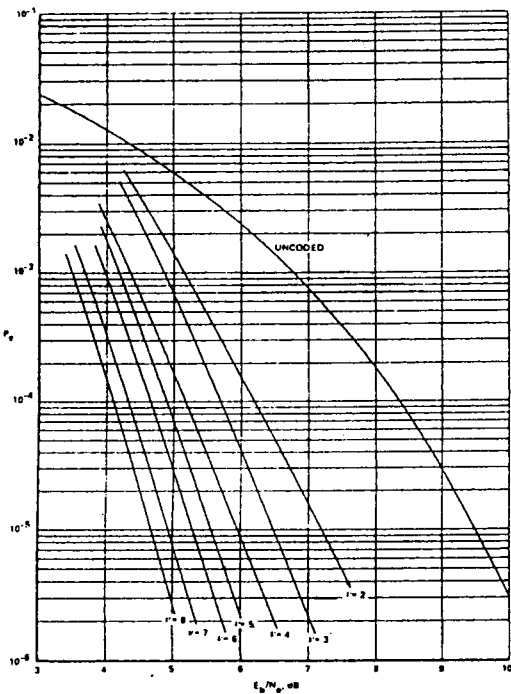
Punctured Code 的性能曲線可以經由 Union Bounding 技術計算而得。對一個 $r=m/n$ code 而言其錯誤率(Bit Error Rate, P_e)在參考資料[4]中所限定的範圍為

$$P_e < \frac{1}{m} \sum_{j=0}^{\infty} w_j(j) P_j$$

其中 P_j 為正確路徑被一個 Hamming distance 為 j 的合併路徑所忽略的機率, $w_j(j)$: 所有 weight- j 路徑的總資訊重。圖五及圖六為經由此 Union bound 所得之結果。以為模擬時的比對參考。



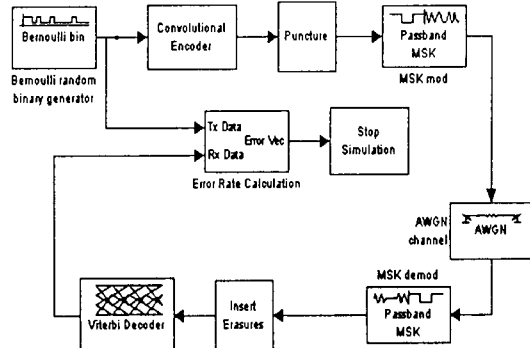
圖五、 $r=2/3$ punctured convolutional code 性能圖



圖六、 $r=3/4$ punctured convolutional code 性能圖

三、電腦模擬

1. 系統架構方塊圖如圖七所示。



圖七、Simulink 系統模擬方塊圖

2. 在 simulink 中每個方塊中之參數設定:

信號產生方面, 0 和 1 所產生的機率各半, 在種子的部份設一可變參數 seed 以用於 matlab 之 command window 上執行, 在取樣時間上, 設定為一單位時間。Convolutional code 編碼器以八進位的型式設定(如表二)。表二中 puncturing: 1 表傳送, 0 表刪除。調變 block 中 symbol interval 設定同 rate 之值。AWGN channel 之種子設定為一固定數值 123456, 信號雜音比(E_s/N_0)設定為 7 作比較, symbol period 和調變中之 symbol interval 一樣。解調使用同步調變。Viterbi 解碼判別型式設為 unquantized, 解碼所需時間設為 96 單位時間。在錯誤率的計算上, 取樣時間設為 1 單位時間。停止模擬設定方面以處理位元之數量(50000), 及錯誤位元之數量(1000), 此二條件其中之一成立, 模擬即停止。

3. 模擬結果

將原本為 $r=1/2$ 之結構, 利用打掉更多碼的方式使 r 之值提升至 $2/3, 3/4$ 等。此次模擬是將 AWGN 通道之種子參數固定, 而將信號之種子參數輸入十種不同數質再求其平均值。結果如表四所示此模擬值遠比圖五及圖六之 upper bound 低甚多(圖五六以 BPSK 表示(E_b/N), MSK 中 $7dB(E_s/N)$ 相當於此處 $4dB$)。

由 Yasuda [5]的結果可看出 (1)同樣速率的 Punctured code, Constraint length K 每增加 1, coding gain 增加 $0.2\sim 0.5$ dB (2)只要 constraint length K 大於 6, 對高速率的 punctured code 來說, coding gain 大於 3。模擬結果證實了用 constraint length 來解決 rate 增高, coding gain 下降的問題。

伍、討論

由模擬結果得知, 在固定的速率下, 可藉由增加 constraint length 來降低錯誤率, 在固定的 constraint length 下, 當速率增加(即打掉的碼較多時)則錯誤率亦上升。而從模擬結果我們不難發現, 當 $r=2/3$ 、 K 值大於等於 5, 以及 $r=3/4$ 、 K 值大於等於六時與 $r=1/2$ 、 $K=5$ 時的錯誤率幾近相同, 而且還可將速率 9.6Kbps 提升至 12Kbps、12.8Kbps, 以達我們所須之目的, 亦即驗證出在不改變頻寬

的情況下，利用增加 constraint length 的方法，降低錯誤率，以增加傳送的資料量。但如 r 值大於等於 $4/5$ 時，就必須犧牲錯誤率，以提升速率，故僅 $r=2/3$ 和 $3/4$ 時較有實用價值。

若要真正應用於工程上，則必須再考慮某些特定之位元，如在框(V.110' frame)中的位元 E7，是用於決定框的同步。在框中的位元 S 和 X，是用於決定對應數個存在當地使用者裝置和終端調整器的 V.24 內部交換電路之功能。在框中的位元 E1, E2 和 E3，是用於指出真正裝置的數據速率。

至於直接使用 $2/3$ rate 或 higher rate 碼，再 puncture 將可更進一步提升 coding gain，但相對的會增加解碼的複雜度，較不建議使用。

陸、參考文獻

- [1]Ericsson Mobile Communications AB, Mobile Data Division, "Mobile Data communication-Guide to Mobitex,"ESPIDER-164 80 Stockholm, Sweden, Mar. 1993
- [2]M. Sreetharan and R. Kumar "Cellular Digital

柒、附錄

表二、由 $r=1/2$ 碼在 $K=3,5,\dots,9$ 之情況下所產生的 $r=(n-1)/n$ 碼之 puncture 方式

Coding Rate \ K	3	4	5	6	7	8	9
1/2	1 (5) 1 (7)	1 (15) 1 (17)	1 (23) 1 (35)	1 (53) 1 (75)	1 (133) 1 (171)	1 (247) 1 (371)	1 (561) 1 (753)
2/3	10 11	11 10	11 10	10 11	11 10	10 11	11 10
3/4	101 110	110 101	101 110	100 111	110 101	110 101	111 100
4/5	1011 1100	1011 1100	1010 1101	1000 1111	1111 1000	1010 1101	1101 1010
5/6	10111 11000	10100 11011	10111 11000	10000 11111	11010 10101	11100 10011	10110 11001
6/7	101111 110000	100011 111100	101010 110101	110110 101001	111010 100101	101001 110110	110110 101001
7/8	1011111 1100000	1000010 1111101	1010011 1101100	1011101 1100010	1111010 1000101	1010100 1101011	1101011 1010100
8/9	10111111 11000000	10000011 11111100	10100011 11011100	11100010 10011101	11110100 10001011	10110110 11001001	11100000 10011111
9/10	101111111 110000000	101000000 110111111	111110011 100001100	100001111 111110000	111101110 100010001	101100110 110011001	111000101 100111010

註：1 表傳輸(transmitting)，0 表刪除(deleting)，括號內的數字以 8 進位表示之產生器。

表四 Punctured convolutional code 在 $E_s/N_0=7$ 之錯誤率與 K 之關係。其中不等式右邊為圖五及圖六之 Upper bound

Coding Rate \ K	3	4	5	6	7	8	9
1/2			$\epsilon < 1.2 \times 10^{-3}$				
2/3	6.4×10^{-5}	2.5×10^{-5}	$\epsilon < 1.2 \times 10^{-3}$	$\epsilon < 7.1 \times 10^{-4}$	$\epsilon < 1.8 \times 10^{-4}$	$\epsilon < 8.8 \times 10^{-5}$	$\epsilon < 4.17 \times 10^{-5}$
3/4	4.09×10^{-4}	2.01×10^{-4}	1.39×10^{-4}	1.6×10^{-5}	3×10^{-6}	$\epsilon < 5.3 \times 10^{-4}$	$\epsilon < 1.8 \times 10^{-4}$
4/5	1.1×10^{-3}	6.3×10^{-4}	3.85×10^{-4}	1.55×10^{-4}	6.8×10^{-5}	3.4×10^{-5}	1.2×10^{-5}

註： $\epsilon \leq 2 \times 10^{-6}$ 。

Packet Data" Artech House Publishers p.p. 32

- [3]S.M. Redl, M.K. Weber and M.W. Oliphant "An Introduction to GSM" Artech House Publishers
- [4]J.BIBB CAIN & JOHN GEIST "Punctured Convolutional Codes of Rate $(n-1)/n$ and Simplified Maximum Likelihood Decoding" IEEE Trans. On Information Theory, Vol. IT-25, No. 1, January 1979, p.p. 97-100
- [5]Yutaka Yasuda "High Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding" IEEE Trans. On Communications, Vol. Com-32, No.3, March 1984