MÓDULOS DE TRELIÇA PARA CÓDIGOS CONVOLUCIONAIS BASEADOS EM CÓDIGOS VARIANTES NO TEMPO

Bartolomeu F. Uchôa-Filho

Grupo de Pesquisa em Comunicações - GPqCom
Departamento de Engenharia Elétrica
Centro Tecnológico
Universidade Federal de Santa Catarina
Florianópolis - SC - 88040-900
E-mail: uchoa@eel.ufsc.br

ABSTRACT

É bastante conhecido que códigos convolucionais invariantes no tempo podem ser representados por uma treliça semi-infinita, formada pela concatenação de cópias idênticas de um módulo regular, chamado de módulo convencional de treliça. Em 1996, McEliece e Lin definiram "a" complexidade de uma treliça como sendo o número de símbolos (no módulo) por bit de informação e, segundo essa medida, determinou que o módulo de menor complexidade, chamado de módulo "mínimo", é aquele baseado na treliça BCJR para códigos de bloco. Dois anos mais tarde, Hole sugeriu uma representação alternativa para códigos convolucionais, baseada num módulo BCJR estendido e numa versão podada do módulo convencional, cuja complexidade é inferior àquela do módulo "mínimo" para vários casos considerados na literatura. Neste artigo, apresentamos exemplos de códigos convolucionais e módulos associados, baseados em códigos variantes no tempo, cuja complexidade, segundo a medida de McEliece e Lin, é bastante inferior àquelas obtidas por Hole e por McEliece e Lin. Estes resultados sugerem que seja dada uma especial atenção para estas represen tações, no intuito de se estabelecer a representação menos complexa para um código convolucional com taxa e distância livre fixadas.

1. INTRODUÇÃO

Considere o corpo de Galois GF$(2) \triangleq F$. Um código convolucional binário $\mathcal{C}$ com comprimento de bloco n, k bits de informação, e memória m (onde m é o menor inteiro para que exista um codificador que gere $\mathcal{C}$ com apenas m unidades de atraso), dito ser um código (n, k, m), é um sub-espaço de dimensão k do espaço $F(D)^n$, onde $F(D)$ é corpo das funções racionais na variável D sobre F [1]. A *distância livre* do código convolucional (ou seja, o peso mínimo de Hamming de qualquer palavra-código) é denotada por d. Assim, diremos que o código (n, k, m) $\mathcal{C}$ de distância livre d é um código (n, k, m, d). A seguir, usaremos a notação adotada em [1] e [2].

Um código (n, k, m, d) $\mathcal{C}$ pode ser representado por uma treliça semi-infinita que, após um breve transiente, consiste na concatenação de cópias idênticas de uma estrutura básica chamada *módulo*. O módulo M é portanto o menor período da treliça semi-infinita. A treliça "convencional" para $\mathcal{C}$ é aquela cujo módulo, denotado por M_{conv}, consiste em 2^m estados iniciais e 2^m estados finais; de cada estado inicial partem 2^k ramos conectando este estado a estados finais (normalmente distintos, mas transições paralelas são admitidas). Cada ramo é rotulado com n bits, correspondentes aos n bits de saida do codificador para $\mathcal{C}$. M_{conv} pode ser obtido diretamente a partir da *matriz geradora canônica* $\mathbf{G}(D)$, que é uma matriz de dimesão $k \times n$ cujos elementos são polinômios $\in F(D)$. Por falta de espaço, e por serem bastante conhecidas, serão aqui omitidas (veja, por exemplo, Lin e Costello [3]) a descrição de $\mathbf{G}(D)$, a relação entre $\mathbf{G}(D)$ e a realização na forma direta do codificador, a seqüência de informação de entrada e a seqüência codificada. Entretanto, precisaremos definir a equivalência entre duas matrizes geradoras (ou dois codificadores). Dizemos que $\mathbf{G}(D)$ e $\mathbf{G}'(D)$ são matrizes geradoras *equivalentes* se e somente se elas geram o mesmo código convolucional (n, k, m, d). Para tanto, é necessário e suficiente que exista uma matriz $k \times k$ não singular $\mathbf{T}(D)$ sobre $F(D)$ tal que [6, Teorema 2.14]:

$$\mathbf{G}'(D) = \mathbf{T}(D)\,\mathbf{G}(D) \tag{1}$$

McEliece e Lin [1] determinaram que o esforço computacional requerido pelo algoritmo de Viterbi para decodificar um bit é proporcional ao número de símbolos (no módulo) por bit de informação. Esta medida é dita ser "a" *complexidade de treliça* (CT) para o módulo M do código

convolucional. É fácil verificar que a complexidade de treliça para o módulo convencional é:

$$CT(M_{conv}) = \frac{n}{k} \cdot 2^{k+m} \qquad (2)$$

símbolos por bit.

McEliece e Lin[1] definiram o módulo "mínimo" como sendo aquele baseado na treliça BCJR para códigos de bloco [4]. Seguindo a nota ção em [2], o módulo BCJR será denotado por P. Foi verificado em [1] que $CT(P)$ é siginificativamente menor do que $CT(M_{conv})$ para vários códigos convolucionais. Dois anos mais tarde, Hole [2] sugeriu uma representação alternativa para códigos convolucionais, baseada num módulo BCJR estendido, P_{ext}, e numa versão podada do módulo convencional, M_{prun}. Hole mostrou que a complexidade resultante, $CT(P_{ext}) + CT(M_{prun})$, é inferior àquela do módulo "mínimo" para vários códigos convolucionais. Neste artigo, apresentamos exemplos de códigos convolucionais e módulos associados, baseados em códigos variantes no tempo, cuja complexidade, segundo a medida de McEliece e Lin, é bastante inferior àquelas obtidas por Hole e por McEliece e Lin. Estes resultados sugerem que seja dada uma especial atenção para estas representações, no intuito de se estabelecer a representação menos complexa para um código convolucional com taxa e distância livre fixadas.

Na próxima seção, consideraremos a representação de Hole [2]. Tomaremos como base para a discussão o exemplo do código convolucional (3,2,1,2) adotado em [2]. Veremos que, através de uma transformação $\mathbf{T}(D)$ apropriada, podemos ter uma matriz geradora equivalente que leva naturalmente a um módulo (baseado em códigos variantes no tempo) menos complexo que aquele apresentado por Hole. Na Seção 3, consideraremos o módulo "mínimo" de McEliece e Lin [1]. Adotaremos um exemplo encontrado em [1] para um código convolucional (8,4,3,8). Neste caso, não existe uma transformação $\mathbf{T}(D)$ que possibilite a obtenção de uma menor CT, como no caso da Seção 2. Por outro lado, apresentaremos um código convolucional (8,4,4,8), ou seja, com mesma taxa 4/8 e mesma distância livre 8, porém com um módulo convencional mais complexo[2]. Apesar disto, este código tem um módulo baseado em códigos variantes no tempo cuja CT é bastante inferior à CT "mínima" do código (8,4,3,8) adotado por McEliece e Lin. Por fim, na Seção 4, faremos alguns cometários finais.

2. OS MÓDULOS DE HOLE

Por questão de simplicidade, nesta seção, mostraremos os módulos P_{ext} e M_{prun} de Hole [2] através de um exemplo.

[1] Deve ser citado o trabalho de Sidorenko e Zyablov [5], semelhante ao de McEliece e Lin e publicado anteriormente.

[2] Note que este código tem $m = 4$, enquanto que no exemplo de McEliece e Lin $m = 3$.

A descrição genérica do método de obtenção dos módulos pode ser encontrada em [2], [7] e [8]. Considere o código convolucional de *memória unitária parcial* (veja [9]) cuja matriz geradora canônica é:

$$\mathbf{G}(D) = \mathbf{G}_0 + D \cdot \mathbf{G}_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} + D \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Note que $m = 1$, que é o *rank* da matriz $\mathbf{G}_1$. Se a matriz $\mathbf{G}_1$ é de *rank* cheio, o código é dito ser de *memória unitária* [10]. O módulo convencional, M_{conv}, para este código (3,2,1,2) é mostrado na Fig. 1. Os rótulos sobre

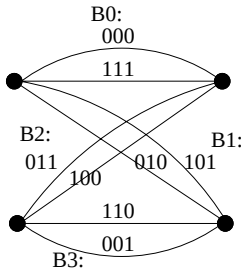

Figure 1: Módulo convencional, M_{conv}, para o código convolucional (3,2,1,2). $B_0 = \{000, 111\}$ é o código de bloco linear relativo às transições do estado inicial zero ao estado final zero. B_1, B_2 e B_3 são os três *cosets* relativos aos demais conjuntos de transições paralelas.

os ramos que partem do estado incial zero em direção ao estado final zero constituem um código de bloco linear, denotado por B_0. No exemplo, $B_0 = \{000, 111\}$. Os demais conjuntos de transições paralelas correpondem aos *cosets* $B_1 = \{010, 101\}$, $B_2 = \{011, 100\}$ e $B_3 = \{110, 001\}$. Como os *cosets* representam apenas translações do código B_0, pode-se facilmente estender a treliça BCJR do código B_0 para representar todos os *cosets*, dando origem ao que Hole chamou de módulo estendido, P_{ext}. Este é mostrado na Fig. 2. No seu procedimento de decodificação, Hole também considerou uma versão podada do módulo convencional da Fig. 1, denotado por M_{prun} (*puned*=podado). Neste módulo, apenas um ramo é mantido de cada conjunto de transições paralelas, sendo este ramo rotulado por B_i, o nome do *coset* ao qual ele pertence. O módulo convencional podado do exemplo em questão é mostrado na Fig. 3.

Iniciando com uma métrica zero no estado incial do módulo da Fig. 2, o algoritmo de Viterbi pode ser utilizado neste módulo para determinar as métricas nos quatro estados finais B_0, B_1, B_2 e B_3. Como estes rótulos são os mesmos da Fig. 3, essas métricas podem ser consideradas como as métricas de ramo no módulo M_{prun}, a partir das quais o algoritmo de Viterbi determina as novas métricas de estado da treliça M_{prun}. O procedimento é repetido para os caminhos sobreviventes, sendo que agora a métrica no estado incial do módulo da Fig. 2 deve correponder às métricas

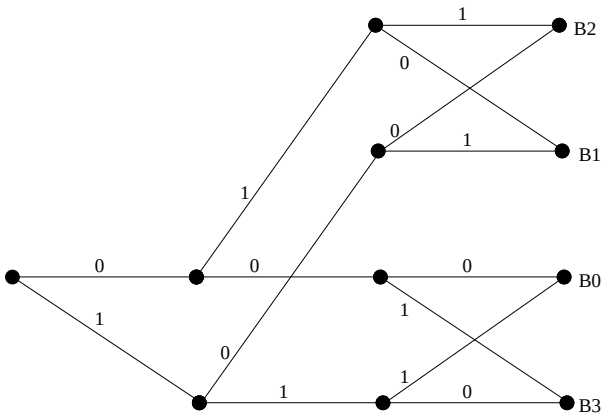

Figure 2: Módulo BCJR estendido, P_{ext}, para todos os *cosets* de $B_0 = \{000, 111\}$.

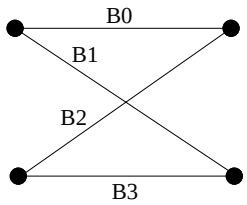

Figure 3: Módulo convencional podado para o código convolucional (3,2,1,2).

acumuladas nos estados finais do módulo M_{prun}. Pode-se verificar facilmente que $CT(P_{ext}) = 14/2 = 7$, pois há 14 símbolos em P_{ext} para $k = 2$ bits de informação. O módulo M_{prun} por sua vêz tem $CT(M_{prun}) = 4/2 = 2$, pois há quatro símbolos para $k = 2$ bits de informação (cada rótulo em M_{prun} é considerado como um único símbolo, pois neste módulo cada métrica de ramo é um único número real previamente computado). A complexidade de treliça resultante é $CT(P_{ext}) + CT(M_{prun}) = 9$.

Sem grandes esforços, podemos constatar que com a matriz

$$\mathbf{T}(D) = \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix}$$

obtemos a matriz geradora canônica equivalente

$$\begin{aligned} \mathbf{G}'(D) &= \mathbf{T}(D)\,\mathbf{G}(D) \\ &= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} + D \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{aligned}$$

De acordo com os trabalhos [11] e [12] sobre códigos puncionados variantes no tempo, a matriz geradora semi-infinita binária [3] deste código pode ser adaptada para uma matriz equivalente de um código puncionado, cujo módulo de treliça corresponde a duas seções, como mostrado na Fig. 4. Claramente, este módulo tem uma $CT = (4+8)/2 = 6$

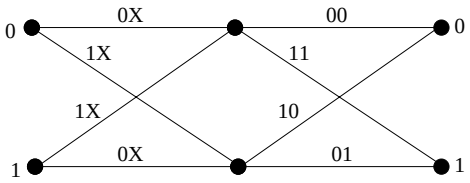

Figure 4: Módulo para o código convolucional (3,2,1,2) baseado em código (puncionado) variante no tempo.

símbolos por bit, que é inferior aos 9 símbolos por bit do exmeplo em [2].

3. O MÓDULO "MíNIMO"

Considere o código convolucional de memória unitária parcial apresentado por Lauer [9] cuja matriz geradora canônica é:

$$\mathbf{G}(D) = \begin{bmatrix} 11111111 \\ 11101000 \\ 10110100 \\ 10011010 \end{bmatrix} + D \begin{bmatrix} 00000000 \\ 11011000 \\ 10101100 \\ 10010110 \end{bmatrix}$$

Note que $m = 3$, que é o *rank* da segunda matriz. A distância livre desse código é 8. O módulo convencional, M_{conv}, para este código (8,4,3,8) tem $CT = 8/4 \cdot 2^{(3+4)} = 256$ símbolos por bit. Esta matriz geradora foi modificada por McEliece e Lin [1], através de uma matriz $T(D)$ apropriada, produzindo um código (8,4,3,8) equivalente para o qual a complexidade de treliça do módulo "mínimo" BCJR é 104 símbolos por bit. Neste caso, não existe uma transformação $\mathbf{T}(D)$ que possibilite a obtenção de uma CT ainda menor, como no caso da Seção 2. Por outro lado, obtemos um resultado surpreendente ao considerar o código convolucional (8,4,4,8) com mesma taxa 4/8 e mesma distância livre 8 gerado por:

$$\mathbf{G}(D) = \begin{bmatrix} 11110110 \\ 00111001 \\ 00001110 \\ 00000011 \end{bmatrix} + D \begin{bmatrix} 11000000 \\ 11110000 \\ 11101100 \\ 10111011 \end{bmatrix}$$

Note que o *rank* da segunda matriz é 4, logo $m = 4$. Apesar de o módulo convencional para este código de memória unitária ser mais complexo (CT=512) que o do seu concorrente (CT=256), a CT do seu módulo baseado em códigos variantes no tempo é consideravelmente inferior à CT "mínima" do código (8,4,3,8) adotado por McEliece e Lin [1], como veremos a seguir. Uchôa-Filho e Palazzo [11], [12] consideraram códigos convoluionais puncionados quando o código *mãe* (i.e., aquele em cujas seqüências codificadas é aplicado o puncionamento) é periodicamente variante no tempo. Naqueles trabalhos o objetivo foi encontrar códigos com a mesma distância livre d obtida no caso invariante no

tempo, porém com um menor peso total de informação associado a seqüências codificadas de peso de Hamming igual a d, visando um melhor desempenho em termos de probabilidade de erro de bit. Adotando as técnicas em [11], [12], podemos construir um módulo de treliça consistindo em 4 seções, ou seja, o menor período da treliça semi-infinita é 4. Cada seção do módulo tem 16 estados, e de cada estado partem dois ramos, cada um contendo 2 bits codificados. Portanto, CT é:

$$\frac{16 \text{ (est./sec.)} \times 2 \text{ (ram./est.)} \times 2 \text{ (bits/ram.)} \times 4 \text{ (sec.)}}{4 \text{ (bits de informação)}}$$

Ou seja, $CT = 64$ símbolos por bit. Este módulo é construido a partir da matriz geradora semi-infinita binária [3], que é obtida trivialmente a partir da matriz $\mathbf{G}(D)$ acima. O procedimento é ilustrado na Fig. 5. Por fim, devemos

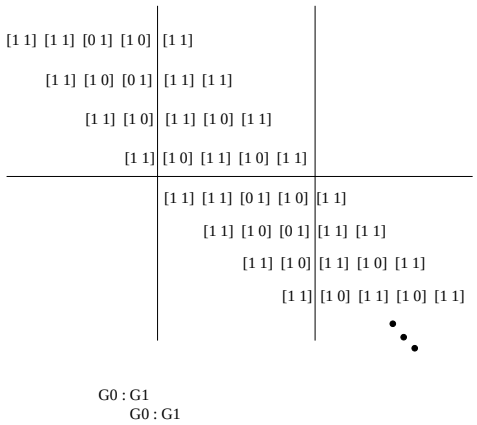

Figure 5: Obtenção do código variante no tempo (período 4) equivalente a partir da matriz geradora semi-infinita binária do código convolucional (8,4,4,8) [11], [12]. Os espaços em branco denotam zeros.

mencionar que a matriz geradora (ou codificador) $\mathbf{G}(D)$ acima, que gera um código de memória unitária, é *inteiramente equivalente*[3] [12] a um codificador variante no tempo obtido por Pil Lee [13].

4. COMENTÁRIOS FINAIS

É óbvia a necessidade de se realizar estudos adicionais no sentido de se estabelecer a representação menos complexa para um código convolucional com taxa e distância livre fixadas. É importante observar que a memória m não precisa ser fixada, já que ela se refere especificamente à complexidade da treliça segundo o critério (aparentemente obsoleto) do número de estados. Com base nos resultados das Seções 2 e 3, podemos afirmar que há casos especiais nos quais a treliça BCJR não apresenta complexidade mínima. A estratégia *two-step* de Hole [2] também aponta para esta direção. Aparentemente, para uma matriz geradora $\mathbf{G}(D)$ qualquer, é correto afirmar que, em geral, uma transformação $\mathbf{T}(D)$ existe tal que a treliça BCJR é a menos complexa. Porém, pode haver um codificador equivalente cujo módulo baseado em códigos variantes no tempo seja o menos complexo, como aconteceu no exemplo da Seção 2. Por outro lado, caso não exista uma transformação $\mathbf{T}(D)$ que atuando na mesma matriz $\mathbf{G}(D)$ proporcione uma estrutura como a da Fig. 5, se for encontrada uma outra matriz $\mathbf{G}'(D)$ que produza a mesma taxa e mesma distância livre porém com uma estrutura como a da Fig. 5, provavelmente teremos a menor CT, como aconteceu no exemplo da Seção 3. A dificuldade, que não é nova quando se trata de códigos convolucionais, é a aparente falta de estrutura algébrica dessa classe de códigos, que torna difícil a elaboração de uma teoria determinando em que condições essa ou aquela abordagem produzirá o melhor resultado. Em todos os trabalhos citados acima, foi inevitável uma busca computacional. Ainda assim, acreditamos que seja possível em breve a construção de uma tabela, como disse McEliece, "no espírito de Brouwer e Verhoeff [14]", listando um bom número de códigos convolucionais com as respectivas representações verdadeiramente mínimas.

[3] Dois codificadores (ou matrizes geradoras) são inteiramente equivalentes [12] se para uma mesma seqüência de informação os codificadores (ou as matrizes geradoras) produzem a mesma seqüência codificada.

AGRADECIMENTOS

Esse trabalho teve o apoio financeiro do Conselho Nacional de Desenvolvimento Cietífico e Tecnológico (CNPq), processos 301593/96-5 e 462450/00-7.

5. REFERENCES

[1] R. J. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Trans. on Inform. Theory*, vol. 42, no. 6, pp. 1855-1864, Nov. 1996.

[2] K. J. Hole, "A comparison of trellis modules for binary convolutional codes," *IEEE Trans. on Inform. Theory*, vol. 46, no. 10, pp. 1245-1249, Oct. 1998.

[3] S. Lin and D. J. Costello, Jr., *Error Control Coding, Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, 1983.

[4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Inform. Theory*, vol. IT-20, pp.284-287, Mar. 1974.

[5] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Trans. on Inform. Theory*, vol. 40, no. 5, pp. 1663-1666, Sept. 1994.

[6] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, New York, NY, 1999.

[7] M. F. Hole and O. Ytrehus, "Two-step trellis decoding of partial unit-memory convolutional codes," *IEEE Trans. on Inform. Theory*, vol. 43, pp. 324-330, Jan. 1997.

[8] M. F. Hole and K. J. Hole, "Low-complexity decoding of partial unit-memory convolutional codes on precoded partial-response channels," *IEEE Trans. on Inform. Theory*, vol. 43, pp. 1052-1058, May. 1997.

[9] G. S. Lauer, "Some optimal partial unit-memory codes," *IEEE Trans. on Inform. Theory*, vol. IT-25, pp. 240-242, Mar. 1979.

[10] L. N. Lee, "Short unit-memory byte-oriented convolutional codes having maximal free distance," *IEEE Trans. on Inform. Theory*, vol. IT-22, pp. 349-352, May 1976.

[11] B. F. Uchôa Filho, R. Palazzo, Jr., A. Said, and C. de Almeida, "New unit-memory codes obtained by puncturing periodically time-varying convolutional codes", in *SBT/IEEE International Telecommunications Symposium*, pp. 534-538, São Paulo, SP, Brazil, Aug. 1998.

[12] B. F. Uchôa Filho and R. Palazzo, Jr., "Unit-memory codes with simplified maximum likelihood decoding", in *ISITA'2000 International Symposium on Information Theory and its Applications*, pp. xxx, Honolulu, HI, U.S.A., Nov. 2000.

[13] P. J. Lee, "There are many good periodically time-varying convolutional codes," *IEEE Trans. on Inform. Theory*, vol. 35, no. 2, pp. 460-463, Mar. 1989.

[14] A. E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. on Inform. Theory*, vol. 39, pp. 662-667, Mar. 1993.