Códigos Cíclicos

Felipe Mourad Pereira
femp1999@gmail.com

Ocimar Mota dos Santos Filho
ocimar.acad@gmail.com

I. Resumo

O presente relatório tem a finalidade de reportar as atividades realizadas pelos autores durante o segundo laboratório de ELE-32. A atividade consistiu de compreender, reproduzir e analisar a codificação e decodificação de sinais binários, com a presença de ruído no canal de comunicação, agora fazendo-se uso de codificação cíclica.

II. Introdução

Em telecomunicações, uma mensagem binária pode ser representada em forma de polinômio utilizando-se cada dígito como coeficiente do polinômio. Assim, uma mensagem $\boldsymbol{v} = [d_0 d_1 d_2 \dots d_{n-1}]$ pode ser representada pelo polinômio $v(D) = d_0 + d_1 D + d_2 D^2 + \dots + d_{n-1} D^{n-1}$, em que D é a variável *Dummy*.

Dado a mensagem binária $\boldsymbol{v} = [d_0 d_1 d_2 \dots d_{n-1}]$, define-se $\boldsymbol{v}^1 = [d_{n-1} d_0 d_1 \dots d_{n-2}]$ como sua primeira rotação e $\boldsymbol{v}^i = (\boldsymbol{v}^1)^{i-1}$ sua $i-ésima$ rotação. Dada essa definição, uma codificação é então considerada cíclica quando, se $\boldsymbol{v}$ é uma palavra-código, então $\boldsymbol{v}^1$ também o é [1].

Define-se também como polinômio gerador $g(D)$ aquele associado à palavra-código de grau mínimo. Assim, uma palavra-código $\boldsymbol{v}$ é obtida de uma palavra-informação $\boldsymbol{u}$ por $v(D) = g(D) * u(D)$, em que $*$ representa a operação produto (ou convolução) entre dois polinômios.

Para a codificação cíclica, a síndrome de um dado padrão de erro é o resto de uma palavra-código com esse erro por $g(D)$. A síndrome sempre será a mesma para um mesmo padrão de erro. [1].

III. Metodologia

Inicialmente, para cada $n \in \{10, 11, ..., 20\}$ e $k = \lfloor \frac{4}{7} n \rfloor$ obteve-se todos os polinômios geradores $(g_{n,k})_i$, em que i é o índice do polinômio gerador $g_{n,k}$ obtido, utilizando-se do comando *cyclpoly* do *Matlab*. Para cada $(g_{n,k})_i$ gerou-se também todas as palavras-informação $\boldsymbol{u}$ de tamanho k possíveis e, fazendo $v(D) = (g_{n,k})_i * u(D)$, todas as palavras códigos $\boldsymbol{v}$ possíveis. Em seguida, para cada $(g_{n,k})_i$, calculou-se a menor distância entre os $\boldsymbol{v}'s$ obtidos.

De posse desse resultado, selecionou-se os 5 (cinco) n e $(g_{n,k})_i$ que levaram às maiores distâncias mínimas para se analisar as eficácia da codificação e decodificação após a passagem de um canal binário simétrico (*BSC*).

Enquanto para a codificação basta realizar o produto entre dois polinômios ($v(D) = g(D) * u(D)$), a decodificação após a passagem pelo canal binário simétrico necessita de um passo a passo maior, de modo a remover possíveis erros de transmissão.

Escolhido um polinômio gerador $g(D)$, para realizar a decodificação deve-se gerar as síndromes de erro no primeiro bit que corrigirão a palavra recebida na transmissão para a palavra-código mais próxima, segundo o conceito de distancia de Hamming [2]. Se a distância mínima entre duas palavras-códigos possíveis para $g(D)$ for d_{min}, então há $d_{min} - 1$ grupos de palavras (ou camadas) entre elas, cada grupo (camada) contendo as palavras com a mesma distância para as duas palavras-códigos corretas. Portanto, deve-se corrigir no máximo $\lfloor \frac{d_{min}-1}{2} \rfloor$ bits, para que a palavra recebida seja corrigida para a palavra-código mais próxima (e por isso mais provável). Assim, gerou-se apenas as síndromes associadas a padrões de erros com erro na primeira posição e com um total de $\lfloor \frac{d_{min}-1}{2} \rfloor$ bits errados.

De posse das síndromes desejadas, a decodificação ocorre obtendo-se o resto de palavra $v(D)$ recebida por $g(D)$ e verificando se o resto coincide com uma das síndromes obtidas anteriormente. Caso sim, corrige-se o primeiro bit e obtêm-se o novo resto. Caso contrário, rotaciona-se a palavra analisada e o resto obtido, verificando-o novamente [1]. Esse processo é repetido até que o resto seja nulo (a palavra agora é uma palavra-código válida). Por fim, desfaz-se as rotações realizadas, obtendo-se a palavra-código mais provável de ter sido emitida. O resultado da divisão de $v(D)$ corrigido por $g(D)$ é a palavra-informação $u(D)$.

Um detalhe importante é que, ao ocorrer a transmissão pelo canal com a implementação do erro, é possível obter uma palavra incorrigível para uma palavra-código válida, uma vez que pode possuir mais erros do que as síndromes geradas conseguem corrigir, mesmo com as rotações. Ou seja, os erros ocorridos tornaram a palavra transmitida mais distante (distância de Hamming) do que conseguimos corrigir. Assim, faz-se um caso limite de parada, em que caso ocorra uma rotação completa sem correção, desiste-se de tentar corrigir a palavra, e ela é transmitida erroneamente.

Para análise do desempenho da codificação cíclica implementada em termos de probabilidade de erro na mensagem decodificada após a transmissão, gerou-se, para cada $(g_{n,k})_i$ selecionado inicialmente, L grupos de k bits de informação, de forma que $L \cdot k$ se aproximasse de cem mil. Então codificou-se, transmitiu-se e decodificou-se cada um dos L grupos de bits de informação. Comparou-se a informação final com a original e calculou-se a probabilidade de ter erro sobre a informação recebida (P_b). Realizou-se esse processo para os valores de probabilidade de erro na transmissão da mensagem $p = [0.5, 0.2, 0.1] \cdot 10^{-i}$ com $i = 1, 2, 3, 4$ para cada um dos $g(D)$ selecionados. Por fim, plotou-se em um único gráfico $p \times P_b$ os resultados, como apresentado na Figura 1.

IV. Resultados e discussões

Os cinco n que obtiveram algum polinômio gerador que levaram às maiores distancias mínimas estão expostos na tabela abaixo:

n	12	14	15	17	18
d_{min}	4	3	4	5	4
$\lfloor \frac{d_{min}-1}{2} \rfloor$	1	1	1	2	1

Assim, seguindo a metodologia exposta, obteve-se o gráfico da Figura 1, que compara as codificações cíclicas realizadas para cada g escolhido e a codificação de Hamming.

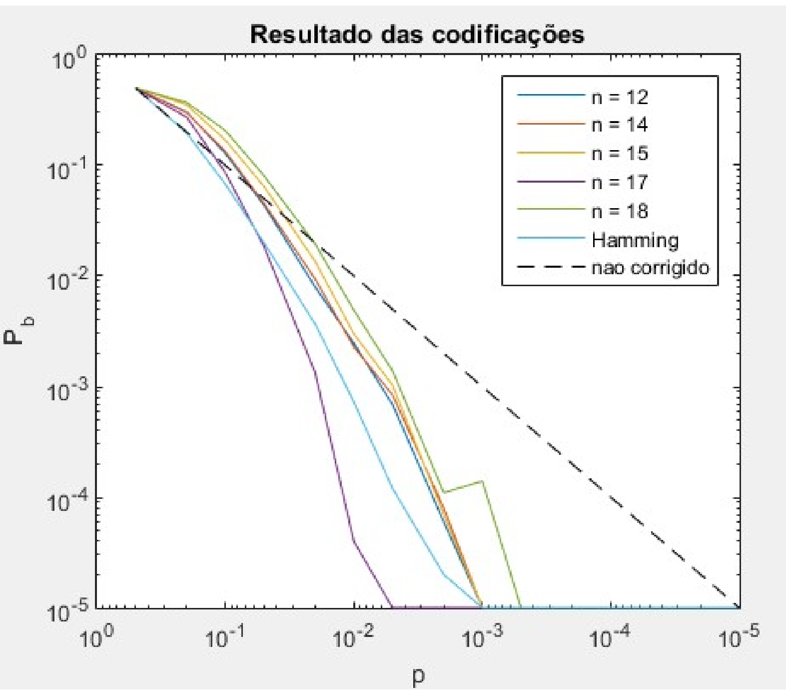

Fig. 1. Resultados obtidos com a decodificação de códigos cíclicos e comparação com codificação de Hamming, em escala logarítmica.

Podemos perceber que para $n = 17$, obteve-se o melhor resultado, com o melhor índice de correções de erro, inclusive melhor que Hamming, uma vez que é o único que possui a capacidade de correção de 2 bits, dado seu valor de $\lfloor \frac{d_{min}-1}{2} \rfloor = 2$.

Para os demais, quanto maior o tamanho, pior o resultado da codificação, uma vez que com mais bits, há a maior probabilidade de ocorrência de múltiplos erros na mesma palavra.

A seguir, algumas perguntas sugeridas para a explicação do experimento:

1. Quais foram as maiores dificuldades em implementar o código, o codificador e o decodificador para os códigos cíclicos?

As principais dificuldades na implementação se encontraram na compreensão do funcionamento da decodificação, principalmente da rotação do resto, e da relação entre a síndrome e a distância da palavra a ser corrigida e a palavra código correta, ou seja, como definir as camadas de correção.

2. Como a dificuldade de projeto de um código cíclico se compara à dificuldade do projeto do código inventado por você no laboratório anterior?

O código cíclico teve uma maior dificuldade de projeto que o código inventado no laboratório anterior, uma vez que houve mais conceitos para se aprender e o código implementado possuía uma maior quantidade de tarefas a serem executadas para a correta decodificação, quando também comparado com o projeto passado, que possuía uma ampla liberdade na definição livre do código utilizado.

3. Compare o desempenho dos códigos criados neste relatório com o desempenho do código de Hamming (que também é um código cíclico).

Apenas o código de $n = 17$ obteve desempenho melhor que o código de Hamming. Isso deve-se ao fato de o código de $n = 17$ ser capaz de corrigir códigos com 2 bits errados, enquanto o código de Hamming, se limita a apenas 1 erro. Assim, a codificação cíclica de $n = 17$ torna-se melhor por corrigir mais palavras recebidas. As outras codificações (para $n = 12, 14, 15$ e 18), por possuírem tamanho maior, com maior probabilidade de múltiplos erros de bit por palavra em comparação com Hamming ($n = 7$), perdem em questão de desempenho de correção, uma vez que esses corrigem apenas um bit por palavra.

Além disso, podemos perceber a perda de velocidade computacional, com uma maior complexidade de tempo, em que não foi sequer possível realizar o código para um tamanho total de entrada da ordem de 10^6, enquanto que para a codificação de Hamming isto levava poucos minutos. Em contrapartida, há uma excelente otimização de espaço, uma vez que a decoficação cíclica realizada utiliza do armazenamento de poucas síndromes fundamentais associadas a erros no primeiro bit e a correção passo a passo, enquanto que na codificação de Hamming armazenava-se um padrão de erro associado a cada síndrome.

4.1. Qual foi o método utilizado para encontrar o codificador de maior distância mínima? Esse método é extensível para qualquer tamanho de bloco?

Para tamanhos de bloco entre $n = 10$ e $n = 20$, como sugerido, obtivemos todos os g's para cada n e k com a função *cyclpoly* sendo k definido pela taxa desejada ($k = \lfloor \frac{4}{7} n \rfloor$), e escolhemos o g em cada n que gere palavras-código com a maior distância mínima. Para encontrar a distância mínima associada a cada $g(D)$, encontra-se as palavras-código associadas a todas entradas (palavras-informação) possíveis, e calculou-se suas distâncias entre si, escolhendo-se a menor delas. Entre os g's obtidos, pode-se identificar o que gera a maior distância mínima, que no caso foi para $n = 17$, com $d_{min} = 5$. Isto é extensível para qualquer intervalo de n, desde que haja polinômios g's para o *range* de taxa desejado. Para n's grandes, muito provavelmente será possível, uma vez que para um dando *range* de taxa, há vários k's possíveis, sendo assim muito provável que para algum desses k's a função *cyclpoly* gere algum $g(D)$.

4.2. Qual foi o método utilizado para encontrar o código maior (palavra-código de maior tamanho)? Esse método é extensível para qualquer tamanho de bloco?

Para encontrar uma palavra-código v de tamanho n para uma dada informação u de tamanho k, basta fazer *conv* com $g(D)$ de tamanho $n - k + 1$, que possui complexidade $O((k)log(k))$. Como $k = \lfloor \frac{4}{7}n \rfloor$, temos $O(nlog(n))$. Assim, o método é extensível para qualquer n, embora possa ficar lento para valores muito grandes.

5. Qual é a relação medida entre o tamanho do bloco e o desempenho?

Podemos perceber que a medida que o tamanho do bloco aumenta, o desempenho cai, uma vez que aumenta-se a probabilidade de ocorrência de múltiplos erros (o que torna-se incorrigível) no mesmo bloco. Isto é válido em blocos que possuem o mesmo valor de $\lfloor \frac{d_{min}-1}{2} \rfloor$. Podemos perceber que, para $n = 17$, em que há um número maior de bits corrigíveis por palavra-código (2 bits), há o melhor desempenho possível, como esperado.

6. Qual a complexidade de codificação e decodificação do seu sistema?

A codificação depende apenas de um produto de polinômios (uma convolução discreta) entre u e g, de tamanhos k e $n - k + 1$, respectivamente, onde n é o tamanho da palavra-código. Portanto, possui complexidade $O(k(n - k))$. Para as taxas utilizadas, temos $O(n^2)$.

Para a decodificação, no pior caso é necessário fazer faz-se $n - 1$ rotações em cada camada, em que cada rotação possui complexidade $O(n)$. Haverá, porém, também no pior caso, $\lfloor \frac{d_{min}-1}{2} \rfloor$ camadas percorridas e em cada camada, faz-se uma divisão de polinômio, que possui complexidade $O(n^2)$, com n o tamanho da palavra-código. Portanto, a complexidade da decodificação é $O(\lfloor \frac{d_{min}-1}{2} \rfloor \cdot ((n-1) \cdot n + n^2)) = O(d_{min} \cdot n^2)$.

V. Conclusão

Com as atividades realizadas, foi possível entender os procedimentos de codificação e decodificação de códigos cíclicos, com a definição de polinômios geradores g, como selecionar o melhor deles, maximizando a distância mínima das palavras-código geradas, e como realizar as rotações da palavra-código e de seu resto da divisão por $g(D)$ a fim de corrigir o erro de transmissão no primeiro bit, ao longo das camadas. Podemos perceber que, com um aumento de complexidade temporal em relação a codificação de Hamming, podemos aumentar a qualidade da correção, obtendo correções de até 2 bits, além da otimização do espaço, armazenando pouquíssimas síndromes fundamentais, atreladas apenas a erros no primeiro bit.

Desta forma, foi possível abstrair, compreender e assimilar como podemos realizar codificação, tentativa e correção e decodificação de sinais binários cíclicos e seus métodos de implementação, o que julgamos ser muito útil para o aprendizado prático em telecomunicações.

References

[1] https://www.ft.unicamp.br/~leobravo/TT%20081/codsec08.pdf

[2] https://en.wikipedia.org/wiki/Hamming_distance