Relatório 2 - ELE-32
Códigos de Bloco Cíclicos e BCH

Aloysio Galvão Lopes
Departamento de
Engenharia da Computação
Instituto Tecnológico de Aeronáutica
São José dos Campos, São Paulo
Email: aloysiogl@gmail.com

Vitor Pimenta dos Reis Arruda
Departamento de
Engenharia da Computação
Instituto Tecnológico de Aeronáutica
São José dos Campos, São Paulo
Email: vitor_pimenta97@hotmail.com

***Resumo*—Este relatório avalia o desempenho, sobre um canal binário simétrico, de códigos cíclicos com bloco pequeno e de códigos BCH primitivos.**

Para tanto, implementaram-se um canal BSC e codificadores/decodificadores para os códigos mencionados. A taxa de erro de bit de informação foi avaliada estatisticamente para eles, submetendo-os a várias palavras aleatórias para estimar seus desempenhos.

I. Respostas a perguntas (rascunho)

A. Do primeiro roteiro (lab3)

1) Quais foram as maiores dificuldades em implementar o codificador convolucional?: Não houve dificuldade. Modelando o codificador como uma máquina de estado, a implementação foi direta.

2) Quanto tempo a sua solução demora para codificar cada bit de informação? Faça uma média.: Analisando cada um dos 3 códigos convolucionais pedidos no roteiro, observou-se que o primeiro demora 330ns para codificar cada bit de informação; o segundo, 365ns; o terceiro, 443ns.

3) Quais foram as maiores dificuldades em implementar o decodificador convolucional?: Dado que a decodificação tem complexidade exponencial de espaço na quantidade de "memórias"do código convolucional, o maior entrave à implementação do algoritmo foi vislumbrar uma estrutura de dados não-ingênua que permitisse codificar toda a informação necessária sem uso de memória excessiva (por exemplo, liberando espaço de memória ao descartar possíveis caminhos na treliça em meio ao algoritmo).

4) Como a probabilidade de erro de transmissão foi estimada? Qual é o seu valor? Como ela se compara com o valor de p escolhido? Como ela muda com m?: Gráficos necessários

5) Qual é o tamanho final do seu executável?: O programa foi codificado em linguagem Julia, a qual usa uma técnica de compilação "just in time"para converter o código a instruções de uma LLVM ("low level virtual machine") e o executar. Apesar de existirem desenvolvedores contribuindo para tornar possível a compilação prévia (ou seja, tornar possível a geração de um executável "standalone"), o código deste laboratório não pôde ser convertido num executável puro. Ainda assim, pode-se dizer, sobre os arquivos produzidos na linguagem Julia, que nenhum supera 2Kbytes de tamanho e, juntos, não superam 20Kbytes.

6) Quanto tempo a sua solução demora para decodificar cada bit? Faça uma média.: Os decodificadores de cada código convolucional trabalhado demoram, em média, $30\mu s$, $60\mu s$ e $300\mu s$ para decodificar 1 bit de informação.

II. Conclusão

O código cíclico apresenta a vantagem de ser necessário armazenar uma quantidade menor de síndromes e de não ser necessário associar síndrome a erro na decodificação. Isso introduz maior complexidade no passo de decodificação pois não há mapeamento direto entre síndrome e erro. No entanto, no código desenvolvido pelas equipes, foi necessário criar um método para geração de códigos. Dessa forma a complexidade do desenvolvimento do código cíclico é maior no tocante ao entendimento dos conceitos envolvidos e no código do experimento anterior no tocante à criação de um procedimento para a geração da matriz de paridade.

Pôde-se notar do gráfico presente em Figura **??** que o código de Hamming é mais eficiente que quase todos os códigos implementados, com exceção do BCH. Isso era esperado uma vez que nenhum dos outros códigos cíclicos corrige mais bits que Hamming e Hamming possui bloco menor que os demais. Nota-se que os casos de distância mínima dois são ligeiramente piores que o canal não codificado e que o caso de distância mínima três é muito próximo de Hamming.

Pode-se depreender que os códigos cíclicos, da maneira como foram desenvolvidos, são ineficazes na diminuição da distância mínima com o aumento do tamanho do bloco. Em contraste a isso, os códigos do experimento anterior apresentados na Figura **??** são, em sua maioria, mais eficazes que Hamming.

Os tamanhos de códigos cíclicos utilizados variaram de 10 ao maior tamanho requisitado (16) e o método utilizado para gerar o conjunto de síndromes é extensível para tamanhos maiores de códigos. Vale notar que a etapa crítica de geração do conjunto de síndromes só precisa ser realizada uma única vez no preprocessamento, por isso, seu tempo de execução total não é crítico.

A medida entre tamanho do bloco e desempenho é dada pela complexidade da multiplicação e divisão polinomial, nesse caso foram consideradas ambas $\mathcal{O}(n^2)$. Além disso, foram discutidos mais detalhes sobre as complexidades das implementações na explicação do algoritmo.

Por outro lado, o código BCH possui rica estrutura matemática subjacente (corpos), possibilitando completa flexibilidade para gerar tamanhos de bloco arbitrariamente grandes e distâncias mínimas desejadas. Além de ser robusto por construção, ainda se caracteriza por complexidade linear de decodificação, como foi empiricamente validado.

REFERÊNCIAS

[1] Berlekamp, Elwyn R. *Algebraic Coding Theory*. Edição revisada. Singapura: WSPC, 2015.

[2] Sharma, Manish *Aula2 - Códigos Cíclicos*. ELE32-Introdução a Comunicações. 2018.