ELE-32 Introdução a Comunicações

Aula 3 - LDPC, variante contínua

May 22, 2024

1 Canal Gaussiano

Até o momento utilizamos o canal BSC por não ter um modelo melhor. Hoje, já sabendo o que é uma modulação 2-PAM (equivalente ao BPSK) e um canal vetorial Gaussiano, podemos extrair mais informação do canal ao utilizarmos o valor recebido por ele ao invés da decisão abrupta (hard decision) sobre o bit transmitido. Por exemplo, ao transmitirmos um símbolo PAM com amplitude $\pm A$ (equiprováveis), sempre decidiríamos por $\hat{s} = +A$ quando o valor recebido $r = s+n$ é positivo, onde n é uma variável aleatória Gaussiana com média 0 e variância σ_n^2. Por outro lado, teríamos mais certeza que $s = +A$ quanto mais positivo r for, isto é, temos mais confiança que $s = +A$ se $r = 10$ do que se $r = 0.1$. Esta confiança pode ser traduzida na probabilidade de $s = \pm A$ dado o valor de r. Além disso, podemos utilizar desta confiança para realizar a decodificação*.

Logo, precisamos de um algoritmo de decodificação para códigos LDPC que utilize as saídas não quantizadas do canal, ou seja, no valor recebido $r = s + n$, sabendo da natureza de n.

1.1 Relaçao sinal ruído

A energia por símbolo transmitido é E_s, calculado pela média dos valores quadráticos dos símbolos da modulação com M símbolos:

$$E_s = \sum_{m=0}^{M-1} P_m ||\mathbf{s}_m||^2 \tag{1}$$

Se os símbolos forem equiprováveis, a entropia da constelação é $log_2[M]$, sendo assim possível atribuir rótulos binários de comprimento $K = log_2[M]$.

A energia por bit é a energia por símbolo normalizada pelo número de bits, isto é, $E_b = E_s/K$, onde Para modulação BPSK, $E_b = E_s$.

A energia do ruído é a variância do canal, σ^2. Esta, por sua vez, tem relação com a densidade espectral de potência constante do ruído AWGN, $N_0/2$. Para comparer sistemas com diferentes larguras de banda e diferentes modulações, utilizamos o parâmetro de relação sinal ruído SNR = Eb/ N0, ou semelhante.

1.2 Valor LLR

O cálculo do grau de confiança e o processo de decodificação são simplificados ao utilizarmos valores LLR (Log Likelihood Ratio), definidos como sendo o valor L em função do valor recebido r:

$$L = ln\left[\frac{P_{S|R}(s = +A|r)}{P_{S|R}(s = -A|r)}\right] \tag{2}$$

*Associamos o valor $s = +A$ ao bit 0 e o valor $s = -A$ ao bit 1

Utilizando o Teorema de Bayes, temos:

$$\begin{aligned} P_{S|R}(s=+A|r) &= \frac{p_{R|S}(r|s=+A)p(s=+A)}{p_R(r)} \\ P_{S|R}(s=-A|r) &= \frac{p_{R|S}(r|s=+A)p(s=+A)}{p_R(r)} \end{aligned} \tag{3}$$

Assumindo que $p(s=+A) = p(s=-A) = 0.5$ e que estamos utilizado símbolos de uma modulação BPSK através de um canal Gaussiano, pode-se demonstrar que:

$$L = \frac{2r}{\sigma^2}, \tag{4}$$

o que permite a obtenção destes valores rapidamente pelo receptor.

Este cálculo precisa ser executado apenas uma vez para cada bit transmitido.

2 Decodificação usando LLR

O uso de LLRs também permite a decodificação de códigos LDPC propagando crenças locais sobre probabilidade (*belief propagation*) dos bits serem iguais a 1 ou 0. Neste caso, as mensagens retratam a crença, ou credibilidade, do bit ser igual a 1 ou 0. Para melhor entendimento vamos utilizar a notação $X_i^w(b)$, onde:

- $i = 1, 2., ..., d$ é o índice de ramos de um node com grau d;
- w indica a direção da mensagem, podendo valer e (entrada), s saida ou f final;
- e $b = 0, 1$ é o valor do bit, se necessário ou apropriado.

A grandeza X depende do contexto. Por exemplo, $P_3^s(1)$ é a mensagem que sai pelo ramo 3 de um node indicando a probabilidade do bit associado àquele ramo valer 1. Por outro lado e L_2^e é a mensagem que entra pelo quarto ramo, nao sendo necessário indicar o valor de b por causa da definição de LLR. Além disso temos também $X_c(b)$ indicando a mensagem que é transmitida pelo canal, seja ela uma probabilidade, valor LLR ou outra coisa.

2.1 V-node

Cada v-node está conectado ao canal e a d_v c-nodes. Não é necessário enviar uma mensagem de volta ao canal pois este não altera o seu valor. Já as mensagens enviadas aos c-nodes devem refletir a crença local na probabilidade do local ser igual a um ou zero, ou equivalentemente a sua LLR. Além disso, a mensagem que sai por um ramo não deve depender da mensagem que entra por este ramo, pois isso poderia causar a realimentação de erros. Como o bit é um só, ele deve valer 0 ou 1. Assim, a crença local a ser transmitida por um ramo sobre a probabilidade do bit valer 0 deve ser igual ao produto das crenças recebidas pelos outros ramos sobre a probabilidade do bit valer 0. De forma análoga, geramos a

$$\begin{aligned} P_i^s(0)' &= P_c(0) \prod_{j=1, j\neq i}^{d_v} P_j^e(0) \\ P_i^s(1)' &= P_c(1) \prod_{j=1, j\neq i}^{d_v} P_j^e(1) \end{aligned} \tag{5}$$

onde $P_c(0) = P(b=0|r)$, valor proveniente do canal para o bit representado pelo v-node em questão.

Entretanto, estes dois valores somados não valem 1, exceto situações pontuais, e por isso que temos o $'$ nas equações. Devemos normalizar estes valores pois necessariamente $P_i^s(0) + P_i^s(1) = 1$. Para isso, usamos o fator normalizador $P_i^s(0)' + P_i^s(1)'$, resultando em:

$$\begin{aligned} P_i^s(0)' &= \frac{P_c(0)\prod_{j=1,j\neq i}^{d_v} P_j^e(0)}{P_c(0)\prod_{j=1,j\neq i}^{d_v} P_j^e(0) + P_c(1)\prod_{j=1,j\neq i}^{d_v} P_j^e(1)} \\ P_i^s(1)' &= \frac{P_c(1)\prod_{j=1,j\neq i}^{d_v} P_j^e(1)}{P_c(0)\prod_{j=1,j\neq i}^{d_v} P_j^e(0) + P_c(1)\prod_{j=1,j\neq i}^{d_v} P_j^e(1)} \end{aligned} \tag{6}$$

Este cálculo é extremamente simplificado ao utilizarmos o valor LLR. Inserindo o valor acima na definição de LLR, obtemos simplesmente:

$$L_i^s = L_c + \sum_{j=1,j\neq i}^{d_v} L_j^e \tag{7}$$

isto é, a mensagem LLR de saída por um ramo é igual a soma das mensagens LLR que entram pelos outros ramos mais a mensagem LLR proveniente do canal.

Este cálculo deve ser feito para todo v-node, a cada iteração (mais sobre iterações na seção sobre execução)

2.2 C-node

Cada c-node está conectado ao canal e a d_c v-nodes. Assim como para os v-nodes, a mensagem que sai por um ramo não deve depender da mensagem que entra por este ramo, pois isso poderia causar a realimentação de erros. Diferentemente dos v-nodes, cada ramo corresponde a um bit diferente. Além disso, a soma modulo 2 dos bits deve valer 0. Logo, a probabilidade de um bit associado a um ramo valer zero é igual à probabilidade da soma de todos os outros bits valer 0. Caso $d_c = 2$, a mensagem que sai por um ramo é exatamente igual a mensagem que entra pelo outro ramo. Caso $d_c = 3$, o cálculo da probabilidade exata que sai pelo ramo 1, por exemplo, seria igual a:

$$\begin{aligned} P_1^s(0) &= P_2^e(0)\cdot P_3^e(0) + P_2^e(1)\cdot P_3^e(1) \\ P_1^s(1) &= P_2^e(0)\cdot P_3^e(1) + P_2^e(1)\cdot P_3^e(0) \end{aligned} \tag{8}$$

Por conter todas as combinações possíveis de bits, a princípio não é necessário normalizar os valores acima. Entretanto, limitações numéricas dos computadores podem fazer com que correções sejam, eventualmente necessárias.

É possível realizar essa conta utilizando mensagens LLR de entrada, traduzindo-as para as probabilidades correspondentes. Neste caso, temos, ainda para o caso com $d_c = 3$:

$$L_1^s = ln\left[\frac{exp(L_2^e)\cdot exp(L_3^e)+1}{exp(L_2^e)+exp(L_3^e)}\right] \triangleq L_2^e \boxplus L_3^e \tag{9}$$

onde definimos a L-soma, representada pelo operador $\boxplus$. No caso genérico com um valor de d_c qualquer, a mensagem de saída de um c-node pode ser exatamente calculada como a L-soma de $d_c - 1$ valores LLR:

$$L_i^s = \boxplus_{j=1,j\neq i}^{d_c} L_j^e \tag{10}$$

A expressão acima é computacionalmente custosa. Para simplificar a conta, podemos usar a seguinte aproximação:

$$L_i^s \approx \min_{j=1,j\neq i}^{d_c} |L_j^e| \cdot \prod_{j=1,j\neq i}^{d_c} sign(L_j^e) \tag{11}$$

isto é, a mensagem de saída tem sinal igual ao produto dos sinais das mensagens nos outros ramos e módulo igual ao menor dos módulos das mensagens dos outros ramos.

Este cálculo deve ser feito para todo c-node, a cada iteração.

2.3 Execução

Definidas as formas de cálculo das mensagens, o algoritmo pode ser descrito nos seguintes passos:

1. Receba o vetor $\mathbf{r}$ proveniente do canal Gaussiano
2. Sabendo o valor de $\frac{N_0}{2}$, calcule os N valores LLRs do canal usando a expressão 4. Inicie o contador de iterações.
3. Para todos os N v-nodes, calcule todas as mensagens de saída usando a expressão 12, totalizando $N \cdot d_v$ mensagens no total.
4. Teste se o critério de parada foi atingido. Caso positivo, pule para o passo final
5. Para todos os $N-K$ c-nodes, calcule todas as mensagens de saída usando a expressão 11, totalizando $(N-K) \cdot d_c$ mensagens no total.
6. Incremente o contador de iterações e retorne ao passo 3
7. Decida sobre os bits transmitidos.

2.4 Critério de parada

O algoritmo é iterativo: em cada iteração calculamos as mensagens na direção VND → CND e vice versa. Há dois critérios de parada. O primeiro é simplesmente o número de iterações máximo. Uma vez atingido, toma-se uma decisão com base nas mensagens trafegando pelos ramos no momento da interrupção.

O segundo critério de parada é atingido quando uma palavra código válida é detectada. Isso pode ser verificado na camada CND: se todas as equações de verificação de paridade estão sendo satisfeitas, então temos uma palavra pertencente ao código. Um valor negativo de LLR corresponde ao bit 1. Deve haver um número par de bits iguais a 1 em cada c-node. Logo, deve haver um número par de mensagens LLR negativas entrando em cada c-node. Concluindo, um c-node tem sua restrição satisfeita se o produto dos sinais das mensagens que entram nele for positivo.

2.5 Decisão

Diferentemente dos casos anteriores, podemos tomar uma decisão mesmo que uma palavra-código não seja detectada. Para cada v-node, a credibilidade final sobre o bit representado pelo node é obtida somando os valores LLR provenientes de todos os ramos:

$$L^f = L_c + \sum_{i=1}^{d_v} L_i^e \tag{12}$$

Se este valor for maior do que zero, $P(b=0|\mathbf{r}) > P(b=1|\mathbf{r})$ e $\hat{b}=0$; caso contrário, $\hat{b}=1$ [†].

[†]Há discussão ainda sobre a prova matemática dessa afirmação, mas empiricamente parece ser verdade após um grande número de iterações

3 Atividades

1. Implemente um sistema que gera os valores LLR provenientes do canal quando transmitimos símbolos de uma modulação BPSK através de um canal Gaussiano. Os parâmetros do ruído devem ser selecionáveis para permitir a sua utilização para qualquer valor de E_b/N_0 desejado. Lembre-se que $\sigma_n^2 = \frac{N_0}{2}$, e que se V é uma variável aleatória Gaussiana com média zero e variância σ_V^2, a variável $W = kV$ tem variância $\sigma_W^2 = k^2\sigma_V^2$. Sugestão: faça com que os símbolos sempre sejam iguais a ± 1 e altere apenas a variância do ruído.

2. Implemente o algoritmo de decodificação para códigos LDPC utilizando os valores LLR conforme descrito acima.

3. Utilize as suas duas implementações acima para estimar o desempenho do código LDPC projetados no laboratório anterior quando utilizamos a modulação $BPSK$, variando o valor de $\frac{E_i}{N_0}$ e seu correspondente valor de $\frac{E_b}{N_0}$ de 0 a 5dB com intervalo de $0.5dB$. É suficiente faze-lo para os códigos com comprimento $N \approx 1000$, mas é recomendável testa-lo antes para o código com comprimento $N \approx 100$. As simplificações adotadas nos laboratórios anteriores podem ser utilizadas; nominalmente: o fato do código ser linear, e a desigualdade de Chebyshev

4. Obtenha um gráfico da probabilidade de erro em função de $\dfrac{E_b}{N_0}$ para o sistema desenvolvido no laboratório anterior.

4 Entregável (via teste)

1. O gráfico do item 4 da seção anterior

2. O menor valor de E_i/N_0 dentre os valores investigados tal que a probabilidade de erro de bit de informação seja menor do que 10^-4 para o código com $N \approx 1000$.

3. Um arquivo csv contendo $N \approx 1000$ linhas descrevendo seu grafo. Cada linha corresponde a um v-node. A linha i deve conter os inteiros $a_1, a_2, a_3, ..., a_{d_c}$, onde $i = 1, ..., N$ é o índice do v-node e $a_1, a_2, a_3, ..., a_{d_c}$ é o índice dos c-nodes conectados ao v-node i. Exemplo:
   1,2,3
   4,5,6
   ...
   213,17,9

Seu código LDPC com $N \approx 1000$ será testado automaticamente com base nos entregáveis 2 e 3. Quanto mais próximo do valor correto, maior a nota. Há uma margem de tolerância. Os critérios numéricos exatos serão definidos a posteriori. O teste pode conter outras questões.