MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
SECRETARIA DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM ENGENHARIA ELÉTRICA

CAP ANDRÉ CARLOS GUEDES DE CARVALHO REIS

INFLUÊNCIA DO USO DE CÓDIGOS CONCATENADOS NO DESEMPENHO DE REDES RÁDIO HF

Rio de Janeiro
2002

INSTITUTO MILITAR DE ENGENHARIA

CAP ANDRÉ CARLOS GUEDES DE CARVALHO REIS

INFLUÊNCIA DO USO DE CÓDIGOS CONCATENADOS NO DESEMPENHO DE REDES RÁDIO HF

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia Elétrica do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia Elétrica.

Orientador: Prof. Paulo Roberto Rosa Lopes Nunes - Ph.D.
Co-orientador: Prof. Weiler Alves Finamore - Ph.D.

Rio de Janeiro
2002

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Reis, André C. G. C.
Influência do Uso de Códigos Concatenados no Desempenho de Redes Rádio HF, André Carlos Guedes de Carvalho Reis, Rio de Janeiro, Instituto Militar de Engenharia, 2002.
2f.:il, graf., tab.: - cm
Dissertação (mestrado) - Instituto Militar de Engenharia, 2002
I. Códigos Corretores de Erro

Ao Instituto Militar de Engenharia, alicerce da minha formação e aperfeiçoamento.

AGRADECIMENTOS

Agradeço a todas as pessoas que me incentivaram, apoiaram e possibilitam esta oportunidade de ampliar meus conhecimentos.

Aos meus pais, Professora Maria de Lourdes Guedes de Carvalho Reis e Professor Dr. José de Carvalho Reis, pelo exemplo , incentivo e apoio dados durante toda a vida.

Ao meu Professor Orientador Dr. Paulo Roberto Rosa Lopes Nunes e ao Professor Co-orientador Dr. Weiler Alves Finamore, por suas disponibilidades, atenções, paciências, humanidades e proficiência no campo da engenharia.

Em especial a minha querida esposa Ane Beatriz dos Santos Reis, pelo amor, carinho e paciência nas longas noites e fins de semana em que esteve privada de minha companhia e ajuda para cuidar de nossos três filhos: Ana Carolina (1 ano), Pedro Gabriel (3 anos) e Camille Beatriz (8 anos).

"Felizes os puros de coração, porque verão à Deus"

Jesus Cristo

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SÍMBOLOS

ABREVIATURAS

HF	-	*High Frequency*
HFDLP	-	*High Frequency Data Link Protocol*
SNR	-	*Signal to Noise Ratio*
SR-ARQ	-	*Selective Repeat - Automatic Repeat reQuest*
FEC	-	*Forward Error Correction*
HARQ	-	*Hybrid Automatic Repeat reQuest*
ARQ	-	*Automatic Repeat reQuest*
ISI	-	*Inter Symbol Interference*
RCPT	-	*Rate Compatible Turbo Codes*
RCPC	-	*Rate Compatible Punctured Convolutional*
MAP	-	*Maximum Probability a Posteriori*
ML	-	*Maximum Likelihood*
DMC	-	*Discrete Memoryless Channel*
AWGN	-	*Aditive Wide Gaussian Noise*
BSC	-	*Binary Symmetric Channel*
BER	-	*Bit-Error Rate*
BCC	-	*Binary Convolutional Code*
BCE	-	*Binary Convolutional Encoder*
FIR	-	*Finite Impulse Response*
IIR	-	*Infinite Impulse Response*
BCH	-	*Bose-Chaudhuri-Hocquenghem*
RS	-	*Reed-Solomon*

SÍMBOLOS

R_c	-	Taxa de código
K	-	Comprimento de restrição
M	-	Ordem de memória de um codificador convolucional

LISTA DE SIGLAS

CCSDS	*Consultative Committee for Space Data Systems*
NASA	*National Aeronautics and Space Administration*
DoD	*Department of Defense*

RESUMO

As redes rádio HF (*High Frequency*) utilizam-se do meio de propagação ionosférica como canal para interligação de regiões geograficamente isoladas e por isso mesmo não cobertas por outros sistemas de comunicações. Com a crescente demanda por comunicações de dados, tem-se procurado melhorar o desempenho dos sistemas de comunicações em geral. No caso particular da rede rádio HF, esta melhoria de desempenho tem sido alcançada com pesquisas e desenvolvimento em duas grandes áreas: novos modems equalizadores cegos e novos protocolos de enlace de dados adaptados para redes em HF. Este trabalho tem como ponto de partida o protocolo HFDLP (*High Frequency Data Link Protocol*) atualmente utilizado em redes militares e descrito no MIL-STD-187-721C e no FED-STD-1052 , Apêndice B. Partindo da observação de que os enlaces em HF operam quase sempre em baixas SNR (*Signal to Noise Ratio*) em virtude da atenuação com a distância, verificou-se que a vazão de um protocolo do tipo SR-ARQ (*Selective Repeat - Automatic Repeat reQuest*) como o HFDLP é insatisfatória devido à grande quantidade de retransmissões. Para reduzir o número de retransmissões, incorporou-se ao protocolo HFDLP dois esquemas FEC (*Forward Error Correction*) com códigos concatenados de bom desempenho em baixas energias: o código do CCSDS (*Consultative Committee for Space Data Systems*) usado pela NASA (*National Aeronautics and Space Administration*) para missões espaciais e o código Turbo. Este último é tido como um dos códigos corretores de erro mais poderosos em termos do limite de Shannon. Foi feita uma comparação dos resultados destes dois esquemas híbridos HARQ (*Hybrid Automatic Repeat reQuest*) com o esquema ARQ (*Automatic Repeat reQuest*) puro do HFDLP e obtidos resultados que mostram uma melhoria significativa no desempenho do sistema medido em termos da vazão. A vazão de um sistema com protocolo ARQ corresponde à razão entre o número de símbolos de informação enviados e a soma destes com outros símbolos quaisquer necessários para transmissão da informação. As contribuições deste trabalho foram: o uso de códigos Turbo na camada de enlace, onde não existe informação lateral do canal; o uso de códigos para comunicações espaciais em redes rádio HF e; a indicação de que o algoritmo de decodificação iterativa dos códigos Turbo deve ser adaptado quando os valores a serem decodificados estiverem no domínio da decisão abrupta.

ABSTRACT

HF (High Frequency) radio networks use the ionospheric propagation medium in order to connect regions isolated geographically and so not covered by other types of communications systems. With the increasing demand for data communications, there has been intense work in order to improve the performance of communication systems in general. In the specific case of HF radio networks, an improvement in performance could be achived by means of research and development in two large areas: new modems with blind equalizers and datalink protocols customized for HF networks. The starting point of this work is the HFDLP (High Frequency Data Link Protocol) protocol currently in use by military HF networks and described in MIL-STD-187-721C and FED-STD-1052, Appendix B. Motivated by the fact that HF links usually operates at low SNR (Signal to Noise Ratio) because of the attenuation caused by distance, it was verified that the throughput of a SR-ARQ (Selective Repeat - Automatic Repeat reQuest) based protocol such as HFDLP is poor because of the high number of retransmission requests. In order to reduce the number of such retransmissions, two FEC (Forward Error Correction) schemes using concatenated codes were embedded into the HFDLP protocol both with good performance at low energy regions: the CCSDS (Consultative Committee for Space Data Systems) code used by NASA (National Aeronautics and Space Administration) for deep space probes and the Turbo codes. The last is known as one of the most powerful error control codes in terms of Shannon limit. Several simulations were carried out with the two HARQ (Hybrid Automatic Repeat reQuest) and the HFDLP pure ARQ (Automatic Repeat reQuest) schemes and the results showed a significant improvement in the system throughput performance. Throughput is defined as the ratio of the number of information symbols sent to the sum of this number with the number of any other symbols necessary to send the information over the link. The main contributions of this work are: the use of Turbo codes in the datalink layer with no side information; the use of code for deep space probes in HF radio networks and the indication that the Turbo iterative decoder algorithm must be tunned for use in the hard decision domain.

1 INTRODUÇÃO

Na era digital moderna, a necessidade de armazenagem e transmissão de vídeo, áudio e dados em altas velocidades e com confiabilidade tornaram a aplicação de codificação para proteção de erros onipresente. Além disso, o uso de técnicas de compressão de fontes de informação, aumentando o impacto dos erros no processo de recuperação da informação, tem contribuido ainda mais para aumentar a importância da armazenagem e transmissão sem erros.

Em sistemas de comunicação digitais que utilizam um canal rádio como meio de transmissão da informação, garantir alta velocidade e confiabilidade é uma tarefa difícil. Esta dificuldade vem do fato do comportamento deste tipo de canal depender sobremaneira de fatores externos ao sistema, alguns deles de natureza não-determinística e não relacionada a características ou parâmetros do sistema.

Desta forma, o conhecimento e caracterização da variabilidade do canal de comunicação empregado pelo sistema é de grande importância na determinação do seu desempenho. Esta variação do canal ocorre tanto no tempo como na freqüência utilizada.

No caso de um sistema digital, o seu desempenho é medido pela probabilidade de erro de bit, que indica a confiabilidade com que a informação é transmitida.Na prática, observa-se que o desempenho de um sistema de comunicações é função de uma combinação de características do transmissor, receptor e canal utilizado.

Nesta combinação, o transmissor é responsável por formatar a informação de forma adequada à transmissão pelo canal. As características mais importantes do transmissor para análise de desempenho são o tipo de modulação, tipo(s) de código(s) para canal, tipo de entrelaçamento, potência de transmissão, antena(s), freqüência/faixa de operação, banda passante requisitada e método de acesso ao meio.

O receptor é responsável por receber a informação em forma de sinais de rádio freqüência e processá-la de forma a obter a informação transmitida. As características mais importantes são tipo de filtragem na entrada, tipo de equalizador, tipo de decisor e tipo de decodificador.

O canal é caracterizado pelo conjunto de meios por onde os sinais rádio e suas cópias/ componentes se propagam percorrendo um caminho com origem no transmissor e destino no receptor. Para facilitar o estudo dos canais procurou-se desenvolver modelos es-

tocásticos empíricos ou semi-empíricos que incorporem as características físicas comuns a determinados tipos ou familias de canais.

Fixando-se as características do transmissor/receptor e tipo de canal, é possível estudar a influência do uso de códigos corretores de erro no desempenho do sistema de comunicação digital.

1.1 PROPOSTA DE TRABALHO

De forma geral, neste trabalho procurou-se avaliar a influência do uso de códigos concatenados no desempenho de comunicações em redes rádio HF. Através desta avaliação, é possível identificar quais códigos ou combinações de códigos apresentam melhor desempenho quando usados na camada de enlace de redes rádio HF. No caso da camada de enlace, melhorar o desempenho significa maximizar a vazão de informação.

A rede rádio HF usada como referência para esta análise é aquela definida pelos padrões militares MIL-STD-188-110A e MIL-STD-187-721C do DoD (*Department of Defense*). De forma a preservar a referência e pertinência deste trabalho, procurou-se manter todas as definições e recomendações contidas nos padrões militares já mencionados.

Deste modo, foi desenvolvido um simulador da camada física e de enlace da rede rádio HF com as características definidas nos padrões militares. Nos pontos em que os padrões são omissos foram adotadas soluções consagradas e com desempenho bem conhecido.

A camada fisíca compreende o canal e o modem. O canal será simulado com base no modelo de Watterson considerando os três canais típicos (bom, moderado e ruim) definidos pelo CCIR (ITU-R) conforme a recomendação CCIR 520-2. O modem a ser implementado é o modem $8PSK$ de tom serial definido no padrão MIL-STD-188-110A.

1.2 ORGANIZAÇÃO

O trabalho está dividido em 7 capítulos.

No Capítulo 2 é apresentado o código concatenado em série CCSDS. Neste capítulo é explicado porque o código RS é usado como código externo, o convolucional como interno e qual o papel do entrelaçador entre os dois códigos. No final do capítulo são apresentadas curvas com o desempenho do código em canal AWGN com e sem entrelaçamento.

O Capítulo 3 apresenta o código concatenado em paralelo Turbo. Este capítulo ressalta as duas características importantes do código Turbo que explicam o seu bom desempenho em baixas energias: o uso de códigos recursivos e a decodificação iterativa.

É mostrado ainda que a decodificação iterativa não é uma solução ótima para a solução da decodificação dos códigos Turbo.

O Capítulo 4 apresenta os dois principais modelos de sistema de comunicações: o modelo OSI e o modelo TCP/IP. A função de cada camada no funcionamento da rede é ressaltada e suas interfaces com as camadas adjacentes são descritas. Na última seção é apresentado o modelo em camadas da rede rádio HF a ser analisada neste trabalho.

No Capítulo 5 é apresentado o modelo de Watterson para o canal HF como um caso particular do modelo genérico WSS-US de Bello. É também apresentado o modem de tom serial definido no padrão MIL-STD-188-110A e descrito a implementação do equalizador adaptativo com seqüência de treinamento adotado para este trabalho.

No Capítulo 6 são apresentados os dois esquemas com os códigos CCSDS e Turbo. São discutidos também o mecanismo de propagação do erro da camada física para a camada de enlace. Na implementação do código Turbo na camada de enlace, é mostrado que para energias altas, a parcela com informação do canal deve ter seu peso reduzido antes do começo das iterações.

No Capítulo 7, são apresentadas as conclusões e propostas de trabalhos futuros.

No APÊNDICE 1 são apresentados conceitos básicos da teoria de códigos corretores de erro tais como as modificações que podem ser realizadas sobre os códigos e os tipos de decodificadores: de máxima verossimilhança e de máxima probabilidade *a posteriori*.

No APÊNDICE 2 são apresentados os códigos convolucionais com ênfase naqueles do tipo recursivo por serem empregados como códigos componentes nos códigos Turbo. É também apresentado o algoritmo SISO BCJR que minimiza a probabilidade de erro de bit e que será usado no decodificador iterativo dos códigos Turbo.

O APÊNDICE 3 contém os conceitos de corpo, anel, ideal e grupo necessários para a compreensão de como são construídos códigos não binários tais como o Reed-Solomon (RS). As principais propriedades dos códigos RS são exploradas e um diagrama detalhado de sua decodificação é explicado passo a passo.

O APÊNDICE 4 apresenta as principais definições e expressões de desempenho referentes a sistemas com protocolos ARQ puro e ARQ híbridos (HARQ).

No APÊNDICE 5 são apresentadas as principais características da camada de enlace. As três máquinas de estados do HFDLP são descritas e são ilustrados os formatos dos quadros de controle, dados e séries de dados.

No APÊNDICE 6 é apresentada a relação entre energia de bit e símbolo e no APÊNDICE 7 um diagrama do simulador de canal HF.

2 CÓDIGO CONCATENADO CCSDS

Enlaces espaciais tais como os que envolvem satelites e sondas espaciais são geralmente caracterizados por limitações de potência de transmissão, pois o número e peso de baterias e painéis solares colocados a bordo destes equipamentos contribui significativamente para os custos de lançamento. Como muitas das aplicações que utilizam estes enlaces necessitam taxas de erro de bit de 10^{-5}, há uma extrema necessidade de códigos corretores de erro que operem eficientemente em relações sinal-ruído extremamente baixas.

Pensando nesta necessidade, na interoperabilidade entre sistemas e no direcionamento das pesquisas em teoria dos códigos, foi formado em 1983 o CCSDS. Um ano após, em 1994, foi lançada a recomendação CCSDS 101.0-B-1 para codificação do canal de telemetria. Posteriormente, esta recomendação foi revisada em 1999.

Em redes rádio HF, embora não hája normalmente limitações quanto a potência de transmissão, o nível do sinal no receptor é normalmente baixo em virtude de grandes perdas causadas pela atenuação durante a propagação. Esta constação permite supor que os metodos para controle e correção de erros em enlaces espaciais podem ser eficientes também para enlaces em redes HF.

Este capítulo apresenta o código concatenado em série com entrelaçamento definido pelo CCSDS e de agora em diante denominado simplesmente de código CCSDS.

2.1 CODIFICADOR

O codificador básico do padrão é o convolucional de taxa 1/2, ordem de memória 6, geradores de seqüência ($g_1 = 1111001$, $g_2 = 1011011$) e $d_{free} = 10$ conforme ilustrado na FIG. 2.1. Este código possui distância livre máxima para sua taxa e comprimento restritivo. Antes da sua adoção pelo CCSDS, este código foi também usado como um dos dois códigos convolucionais do *Planetary Standard* da NASA.

Quando o bloco de puncionamento for usado no codificador da FIG. 2.1, o inversor (bloco envolvido por uma linha tracejada) é suprimido. O puncionamento permite que os códigos de taxa 2/3, 3/4, 5/6 e 7/8 sejam obtidos de acordo com a 2.1.

Para aplicações que necessitem de um alto ganho de código, o CCSDS recomenda que o código convolucional seja concatenado serialmente com um codificador RS e intercalados

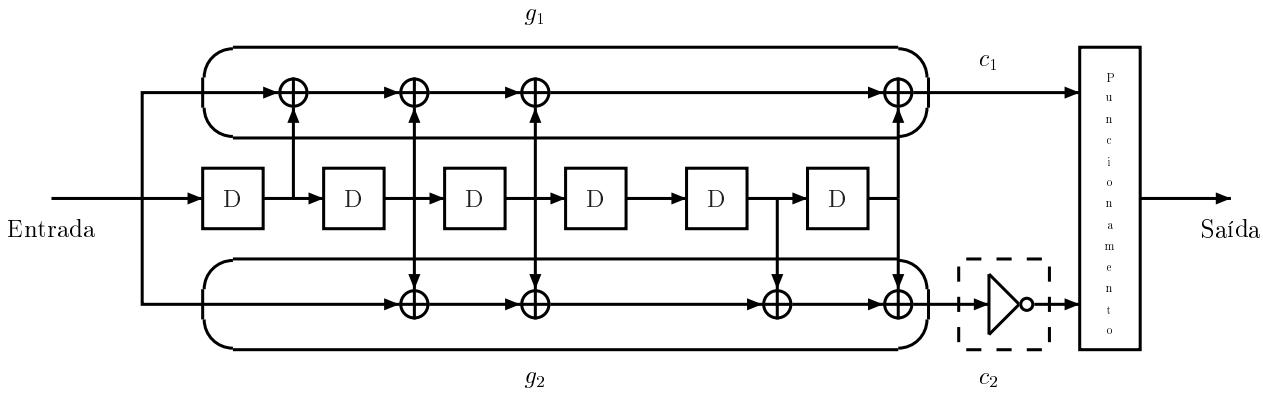


FIG. 2.1: Diagrama em bloco do codificador convolucional

TAB. 2.1: Padrão de códigos puncionados para taxas de códigos convolucionais

Padrão de puncionamento 1 = símbolo transmitido 0 = símbolo não transmitido	Taxa de código	Seqüência na saída
$c_1 : 10$ $c_2 : 11$	2/3	$c_1(1), c_2(1), c_2(2)$
$c_1 : 101$ $c_2 : 110$	3/4	$c_1(1), c_2(1), c_2(2), c_1(3)$
$c_1 : 10101$ $c_2 : 11010$	5/6	$c_1(1), c_2(1), c_2(2), c_1(3), c_2(4), c_1(5)$
$c_1 : 1000101$ $c_2 : 1111010$	7/8	$c_1(1), c_2(1), c_2(2), c_2(3), c_2(4), c_1(5), c_2(6), c_1(7)$

por um entrelaçador conforme pode ser visto na FIG. 2.2. Esta figura mostra o código RS como código do canal externo e o convolucional como código interno.

Os parâmetros do código RS especificado pelo CCSDS são:

1. O número de bits por símbolo é $s = 8$.
2. O número de símbolos por palavra-código é $n = 2^s - 1 = 255$.
3. O número de símbolos que podem ser corrigidos dentro de uma palavra-código é $t = 16$ ou 8
4. O número de símbolos de informação é $k = n - 2t$.
5. O polinômio primitivo $p(x)$ de grau 8 pertencente a $GF(2)[x]$ e gerador do corpo $GF(256)$ é

$$p(x) = x^8 + x^7 + x^2 + x + 1 \tag{2.1}$$

Desta forma, todos os elementos de $GF(256)$, o corpo estendido gerado

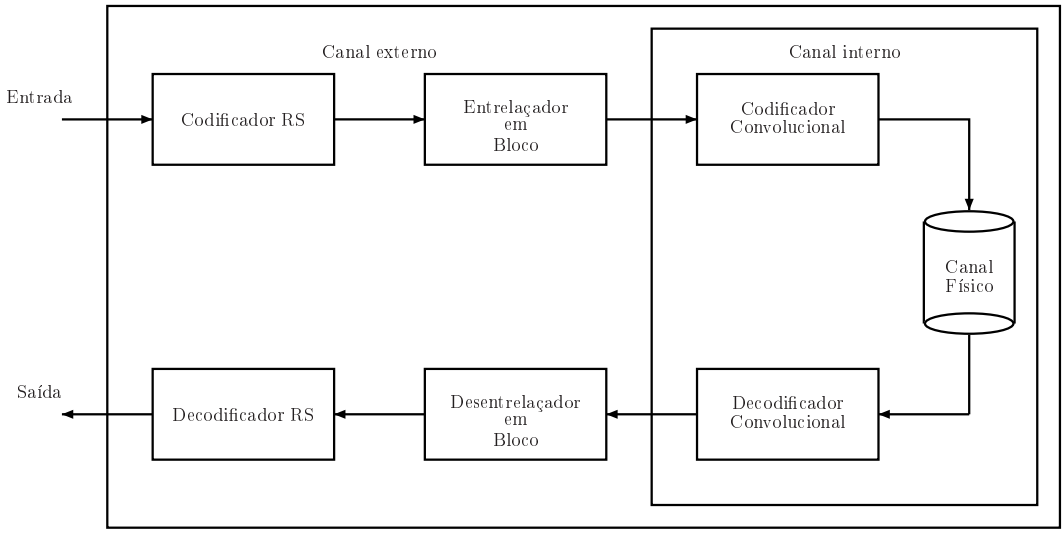


FIG. 2.2: Diagrama em bloco do padrão de telemetria do CCSDS

a partir do corpo fonte $GF(2)$, possuem uma notação polinomial e a multiplicação de dois elementos é feita sempre módulo $p(x)$.

6. O polinômio gerador do código é dado por

$$g(x) = \prod_{j=128-t}^{127+t} (x - \alpha^{11+j}) = \sum_{i=0}^{t} g_i x^i \qquad (2.2)$$

onde α é um elemento primitivo em $GF(256)$.

7. O código é sistemático

Conforme descrito no APÊNDICE 3, os códigos RS são códigos em bloco lineares que constituem um subconjunto dos códigos BCH. Os parâmetros acima especificam um código RS (n, k) com símbolos de s bits onde $n = 255$ e $k = 223$ ou 239. Isto significa que o codificador toma k símbolos de informação cada um com s bits e adiciona símbolos de paridade de modo a formar uma palavra-código com n símbolos. Há então $n - k$ símbolos de paridade com s bits cada O decodificador RS pode corrigir até t símbolos que contenham erros por palavra-código, onde $2t = n - k$.

A FIG. 2.3 ilustra uma palavra-código do código RS $(255, 223)$ com símbolos de 8 bits. Cada palavra-código contém 255 bytes, dos quais 223 são informação e 32 são paridade. Neste caso, o decodificador pode corrigir até 16 erros de símbolo em qualquer lugar da palavra-código. Um erro de símbolo ocorre quando um ou mais bits dentro de um símbolo estiver(em) errados.

Caso haja necessidade, os códigos RS podem ser encurtados igualando a zero alguns símbolos de informação antes do codificador, não transmitindo estes símbolos e reinserindo-os antes do decodificador.

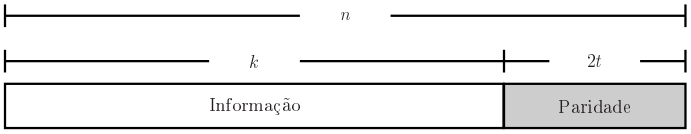


FIG. 2.3: Diagrama em bloco do padrão de telemetria do CCSDS

O FIG. 2.4 mostra o codificador RS $(225, 223)$. No primeiro passo, a mensagem $m(x)$ é multiplicada por x^{n-k}. Esta multiplicação é feita colocando-se as chaves na posição X e alimentando-se o codificador com os símbolos de informação de índices decrescentes. No passo seguinte, as chaves na posição Y e o resultado da multiplicação do passo anterior é dividido por $g(x)$ ficando o resto $d(x)$ desta divisão armazenado nos registradores de deslocamento. Por último, o resto é deslocado para fora dos registradores e acrescentado aos símbolos de informação para formar a palavra-código.

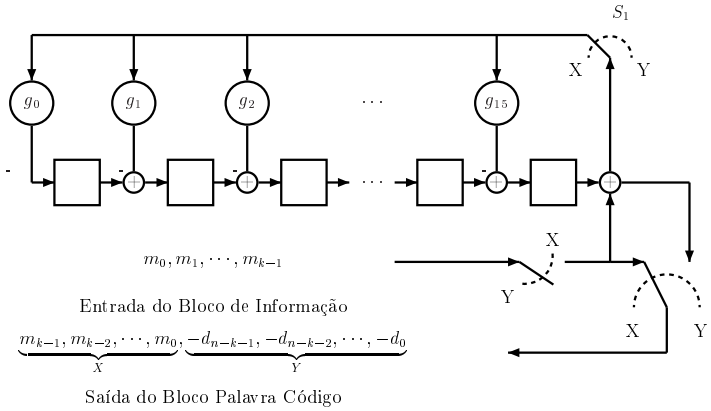


FIG. 2.4: Diagrama em bloco do codificador do código RS do CCSDS

O codificador pode ser usado com entrelaçador. Um entrelaçador é um dispositivo colocado antes do transmissor que mistura no tempo símbolos pertencentes a diversas palavras-código diferentes. Esta ação cria uma diversidade temporal entre os símbolos a serem transmitidos. Quando as palavras-código forem reconstruídas no desentrelaçador, os erros em surto que por acaso tenham sido introduzidos pelo canal serão quebrados e estarão espalhados por várias palavras-código. Isto faz com que a chance do código conseguir corrigir todos os erros dentro de uma palavra-código seja maior.

O entrelaçador recomendado pelo CCSDS é um entrelaçador em bloco com profundidade m conforme ilustrado na FIG. 2.5. Na recomendação, o valor de m varia de 1 a até 5. Neste entrelaçador as chaves S_1 e S_2 se deslocam sincronamente e permanecem em cada posição durante o intervalo de duração de um símbolo RS (8 bits). Deste modo, tem-se a seguinte seqüência na entrada: $d_1^1 \cdots d_1^m d_2^1 \cdots d_2^m \cdots d_k^1 \cdots d_k^m$ seguida de $2tm$ espaços. Estes espaços serão completados com os $2tm$ símbolos de paridade gerados pelos

codificadores: $p_1^1 \cdots p_1^m \cdots p_{2t}^1 \cdots p_{2t}^m$. A palavra-código gerada pelo codificador i terá a seguinte forma: $d_1^i d_2^i \cdots d_k^i p_1^i \cdots p_{2t}^i$

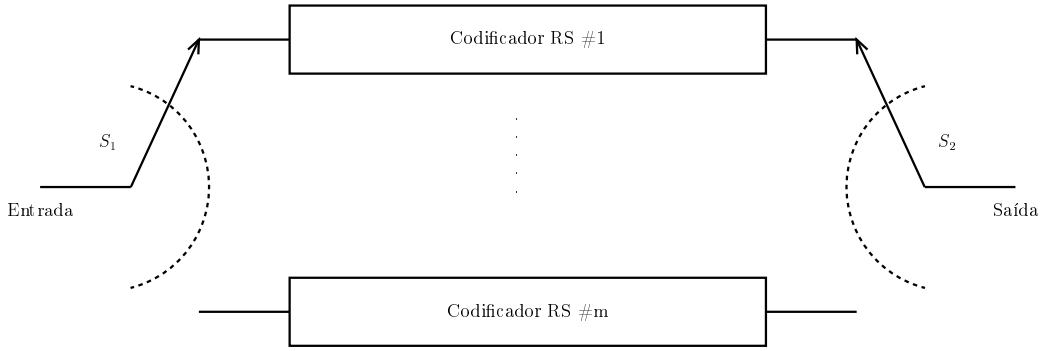


FIG. 2.5: Diagrama em bloco do entrelaçador recomendado pelo CCSDS

2.2 DECODIFICADOR

Em enlaces espaciais, o principal problema é o ruído branco. O decodificador definido pela recomendação para o canal de dentro é o de Viterbi com decisão suave e pelo menos 3 níveis de quantização. A função do decodificador de Viterbi interno é melhorar a qualidade efetiva do canal a ponto do código RS poder ser usado com eficiência. O termo eficientemente aqui empregado significa usar o código RS em canais com erros em surtos. Esta capacidade do código RS de corrigir erros em surto é determinada por duas características: ser MDS e ser não binário.

A escolha do código RS como código externo é natural pois é fato conhecido que o decodificador de Viterbi produz erros em surtos com duração mínima da ordem do comprimento restritivo do código convolucional empregado. Como o código usado nesta recomendação possui $K = 7$, a maior parte dos erros produzidos na saída do decodificador de Viterbi gerarão surtos com comprimento 7 ou 8 e portanto poderão estar confinados em um único símbolo RS.

Se na saída do decodificador de Viterbi, os erros em surto excederem a capacidade de correção do código RS deverá ser usado o entrelaçador para distribuir a carga de controle de erro sobre mais de uma palavra-código.

2.3 DESEMPENHO

O desempenho do código concatendado do CCSDS é determinado pelos dois fatores que seguem:

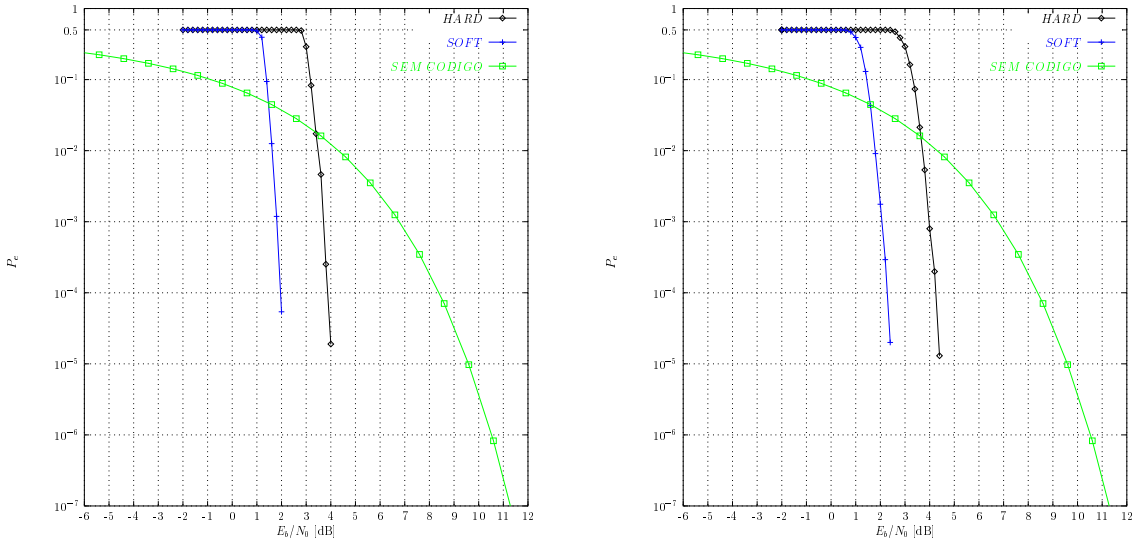
- A distância livre do código convolucional d_H^{free}.
- Tamanho máximo do surto de erro admitido pelo código RS.

Os surtos de erros podem ser produzidos apenas pelo decodificador de Viterbi (Canal AWGN) ou por este em conjunto com o canal que produza erros em surtos como, por exemplo, o canal Rayleigh.

Quando o tamanho máximo do surto de erro admitido pelo código RS for excedido, o uso do entrelaçamento permite melhorar o desempenho do código concatenado CCSDS como um todo. O tamanho do entrelaçador deve ser estimado a partir das estatíticas dos padrões de erro em surto na saída do decodificador de Viterbi.

2.3.1 CANAL AWGN

Para o canal AWGN, a FIG. 2.6 mostra as curvas de probabilidades de erro de bit com decisão suave e decisão abrupta. A modulação usada na simulação foi a BPSK.



(a) Sem entrelaçamento

(b) Com entrelaçamento de tamanho $I = 5$

FIG. 2.6: Curvas de desempenho do CCSDS

Quando o entrelaçamento de tamanho 5 é empregado, a curva FIG. 2.6(b) mostra uma melhoria de cerca $0,4\ dB$ para probabilidade de erro de 10^{-5}.

3 CÓDIGO CONCATENADO TURBO

Conforme destacado no APÊNDICE 1, a teoria de Shannon (SHANNON, 1948) mostrou que todo canal de comunicação possui uma capacidade máxima para transmissão confiável de informações. Transmitindo com um bom código de taxa menor que a capacidade do canal $R_c < C$ é possível atingir uma confiabilidade tão boa quanto se queira. Caso contrário, ainda que seja usado o melhor código disponível, não seria possível garantir a confiabilidade da comunicação.

Aproximadamente 50 anos após o trabalho de Shanon, em 1993, Claude Berrou, Alain Glavieux e Punja Thitimajshima publicaram um trabalho entitulado *Near Shannon limit error-correcting coding and decoding: Turbo Codes* (BERROU, 1993), exibindo um código de taxa $R_c = 1/2$ com uma modulação BPSK em canal AWGN, usando apenas $E_b/N_0 = 0,7\,dB$ para obter uma probabilidade de erro de bit de 10^{-5}. Como a capacidade neste canal e para esta probabilidade de erro de bit está em $E_b/N_0 = 0,0\ dB$ (WICKER, 1995; WELLS, 2000; SHANNON, 1948), este código está a apenas $0,7\ dB$ do limite de Shannon. O código CCSDS está a $2,0\ dB$ deste limite.

As idéias usadas pelo códigos Turbo não são novas pois:

1. A construção de um código com palavras-códigos maiores usando dois ou mais códigos constituintes mais simples já havia sido proposta em 1966 por G. D. Forney (FORNEY, 1966) e seus benefícios conhecidos através dos códigos usados em missões espaciais. O CCSDS é um exemplo.

2. O algoritmo usado no decodificador foi proposto em 1974 por L. R. Bahl (BAHL, 1974) para minimizar a probabilidade de erro de bit.

Entretanto o modo como estes elementos foram organizados permitiu produzir um código corretor de erro com um desempenho nunca antes observado. Dentre as características mais importantes que diferenciam o código Turbo estão a codificação com códigos constituintes convolucionais recursivos em configuração paralela e a decodificação iterativa que serão a seguir descritos.

3.1 CODIFICADOR

O diagrama em blocos de um codificador Turbo sistemático em configuração paralela PCE (*Parallel Concatenated Encoder*) é ilustrado na FIG. 3.1. Nesta figura um bloco de informação $\mathbf{m}$ é codificado pelo codificador C_1 de modo a criar a palavra de paridade $\mathbf{c}_{p1}$. A palavra de informação $\mathbf{m}$ é então permutada pelo bloco entrelaçador Π de comprimento N gerando um versão $\mathbf{m}'$. Esta verão será codificada por C_2 para gerar a palavra de paridade $\mathbf{c}_{p2}$. Caso apenas uma das versões da palavra-informação seja puncionada, o código resultante será $\mathbf{c} = (\mathbf{m}, \mathbf{c}_{p1}, \mathbf{c}_{p2})$ de taxa $R_c = 1/3$. Códigos de taxas mais altas podem ser gerados a partir do puncionamento de símbolos da palavra-informação ou de uma das palavras de paridade.

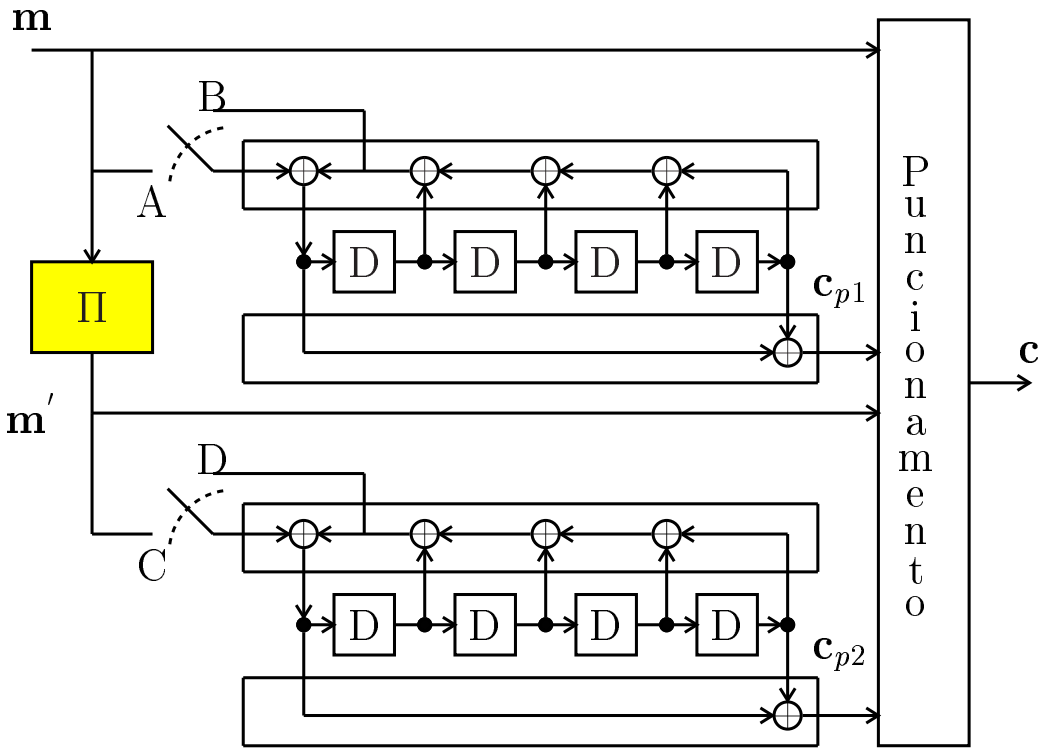


FIG. 3.1: Diagrama do codificador com dois códigos constituintes

Na FIG. 3.1 os códigos constituintes empregados são do tipo convolucional recursivos sistemáticos RSC (*Recursive Systematic Convolutional*) embora pudessem ser também do tipo convolucional não-sistemático NSC (*Nonsystematic Convolutional Code*). É fato bem conhecido que a taxa de erro de bit de um código NSC, para E_b/N_0 grande, é melhor do que a de um código sistemático com mesma ordem de memória. Quando, ao contrário, a razão E_b/N_0 é pequena, o desempenho do código sistemático é melhor. O trabalho de Berrou (BERROU, 1993) mostrou que a classe de códigos RSC pode ser melhor que o

melhor código NSC em qualquer E_b/N_0 para taxas de código altas.

Os códigos usados na FIG. 3.1 são RSC idênticos e têm matriz geradora dada pela EQ 3.1.

$$G_R(D) = \left[1 \quad \frac{f(D)}{g(D)}\right] = \left[1 \quad \frac{1+D^4}{1+D+D^2+D^3+D^4}\right] \tag{3.1}$$

onde:

$f(D)$ é o polinômio de saída

$g(D)$ é o polinômio de realimentação

Os polinômios $f(D)$ e $g(D)$ também podem ser representados na notação octal por 21 e 37 respectivamente. As saídas dos codificadores são dadas por $\mathbf{c}_{p1} = \mathbf{m} \cdot f(d)/g(D)$ e $\mathbf{c}_{p2} = \mathbf{m}^{'} \cdot f(d)/g(D)$.

O codificador PCE da FIG. 3.1 pode ser usado para gerar um código em bloco de tamanho $(n(N+M), N)$. Como os codificadores componentes são recursivos, não é suficiente igualar a zero os últimos M bits de informação para levar o codificador ao estado zero, ou seja, terminar a treliça. A seqüência de terminação da treliça depende do estado de cada codificador componente depois de N bits. Tal fato torna impossível terminar ambos codificadores com os mesmos M bits. Felizmente, o simples estratagema ilustrado na FIG. 3.1 é suficiente para terminar a treliça: as chaves estarão na posições A e C nos primeiros N ciclos de relógio e na posições B e D para os M ciclos adicionais que irão preencher os codificadores com zeros. O decodificador não supõe conhecidos os M bits terminais.

Tradicionalmente, bons códigos Turbo têm sido construídos com os códigos componentes de comprimento restritivo bastante pequenos ($K = 3$ a 5). Concatenações adicionais podem ser feitas com códigos constituintes não idênticos, ou seja, com taxa e comprimento restritivo diferentes. Conforme será visto posteriormente, o objetivo no projeto de um código Turbo é escolher os melhores códigos constituintes através da maximização da distância livre efetiva do código. Para valores altos de E_b/N_0, esta regra é equivalente a maximizar o peso mínimo das palavras-código. Já para valores baixos de E_b/N_0, otimizar a distribuição de pesos das palavras-código é mais importante que maximizar o peso mínimo.

O codificador Turbo da FIG. 3.1 produz dois conjuntos de palavras-código, $\mathbf{c}_1 = (\mathbf{m}, \mathbf{c}_{p1})$ e $\mathbf{c}_2 = (\mathbf{m}^{'}, \mathbf{c}_{p2})$, a partir de cada dos dois codificadores componentes. A distribuição de pesos das palavras-código produzidas por esta concatenação em paralelo vai depender de como as palavras-código de um codificador são concatenadas com as

do outro. Intuitivamente, é importante evitar que palavras-código de baixo peso de um codificador sejam emparelhadas com as de baixo peso do outro. É neste ponto que um entrelaçamento adequado é importante para evitar este emparelhamento. Observe-se que, se os codificadores não forem recursivos, uma palavra-código de baixo peso gerada pela seqüência de entrada $\mathbf{m} = (00\cdots00100\cdots00)$, com um único 1, irá sempre aparecer de novo no segundo codificador, qualquer que seja o entrelaçador usado. Tal fato explica o uso dos codificadores RSC, no qual o principal ingrediente é a recursividade e não o fato de ser sistemático. No exemplo da FIG. 3.1, a seqüência de entrada $\mathbf{m} = (00\cdots0010000100\cdots00)$ produz uma palavra-código de peso mínimo ($w_H(\mathbf{c}_2) = 6$). Se o entrelaçador usado preservar este padrão de entrada, a distância mínima do PCE será 12. O interessante é que este valor seja bem maior.

3.2 DECODIFICADOR

Seja um decodificador ML para um código convolucional de taxa 1/2, recursivo ou não, e um bloco de informação de tamanho N. Caso a estrutura do código fosse ignorada, o decodificador ML teria que comparar 2^N seqüências com a seqüência recebida corrompida pelo ruído, de forma a escolher a seqüência que tivesse a melhor correlação. Claramente a complexidade deste algoritmo seria exorbitante. O Algoritmo de Viterbi permite simplificar bastante este problema pois permite eliminar em cada passo metade das seqüências candidatas.

Como o código Turbo possui um entrelaçador, a estrutura da sua treliça é por demais complexa inviabilizando o uso do algoritmo de Viterbi. A solução adotada para os códigos Turbo foi a estratégia de decodificação cooperativa e iterativa usando mais de um decodificador. Basicamente há duas técnicas que são empregadas na decodificação de códigos Turbo:

- **SOVA**: consiste no algoritmo de Viterbi com saída suave (*Soft-Output Viterbi Algorithm*). Este foi desenvolvido por Hagenauer (HAGENAUER, 1989) em 1989.
- **BCJR modificado**: originalmente proposto por Bahl et al (BAHL, 1974) em 1974 e modificado por Berrou et al (BERROU, 1993) em 1993 para códigos RSC. Por usar informaçã *a priori* é considerado um decodificador MAP.

Dada a estrutura do código turbo ilustrada na FIG. 3.1, a regra de decodificação ótima maximizaria $P(m_k|\mathbf{y}_1, \mathbf{y}_2)$ (regra de mínima probabilidade de erro de bit) ou

$P(\mathbf{m}|\mathbf{y}_1, \mathbf{y}_2)$ (regra de seqüência de máxima verosimilhança) onde $\mathbf{y}_1$ e $\mathbf{y}_2$ são os valores recebidos referentes às saídas dos codificadores $\mathbf{c}_{p1}$ e $\mathbf{c}_{p2}$ respectivamente. Como esta regra é computacionalmente muito complexa, foi adotada uma regra de decodificação sub-ótima que usa separadamente as duas observações $\mathbf{y}_1$ e $\mathbf{y}_2$, conforme ilustrado no diagrama da FIG. 3.2. Neste diagrama cada decodificador é um SISO (*Soft-In Soft-Out*), decodificador com entrada e saída suaves .

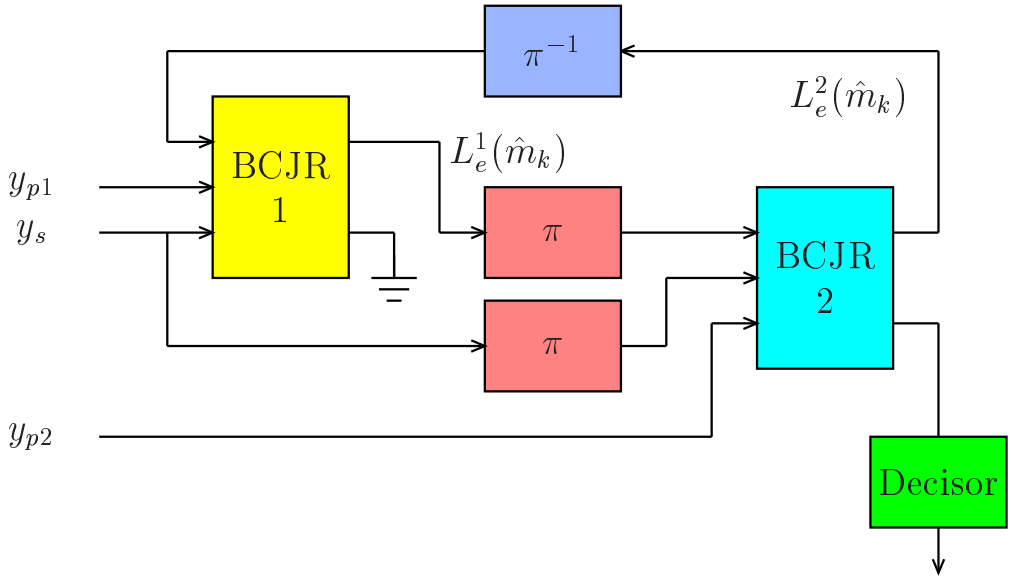


FIG. 3.2: Diagrama de decodificação iterativa com SISO BCJR modificado

FIG. 3.3: Diagrama de decodificador SISO

A FIG. 3.3 ilustra um decodificador SISO. Este decodificador tem como entrada, caso estejam disponíveis, os valores suaves *a priori* $L(m_k)$ de todos os bits de informação m_k e os valores dos símbolos $L_c \cdot y_k$ da palavra-código recebida. A saída do decodificador símbolo a símbolo, MAP, BCJR, são as quantidades $L(\hat{m}_k)$ e $L_e(\hat{m}_k)$. A $L(\hat{m}_k)$ é a razão (logaritmica) de probabilidades *a posteriori* LAPP (*Logaritmic A Priori Probability*) para um símbolo transmitido $+1$ ou -1 do bloco de informação, conforme a EQ 3.2.

$$L(\hat{m}_k) \triangleq L(m_k|\mathbf{y}) = \log\left(\frac{P(m_k = +1|\mathbf{y})}{P(m_k = -1|\mathbf{y})}\right) \tag{3.2}$$

onde $\mathbf{y}$ é a seqüência de todos os bits codificados recebidos.

Na saída deste codificador tem-se os valores suaves de todos os símbolos de informação e ainda a informação extrínseca $L_e(\hat{m}_k)$ que contém a informação de saída suave de todos os outros símbolos da palavra-código e não é influenciado pelos valores em $L(m_k)$ e $L_c \cdot y_k$ do símbolo corrente. Para códigos sistemáticos, a saída suave para o símbolo de informação m_k é representada por três termos aditivos conforme indicado na EQ 3.3.

$$L(\hat{m}_k) = L_c \cdot y_k + L(m_k) + L_e(\hat{m}_k) \tag{3.3}$$

As três parcelas da EQ 3.3 representam três estimativas independentes da LAPP do símbolo m_k: do valor recebido do canal $L_c \cdot y_k$, do valor *a priori* $L(m_k)$ e do valor $L_e(\hat{m})$ obtido a partir das restrições do código.

Assumindo que os símbolos de informação são igualmente prováveis, não há informação a priori disponível para a primeira parte da primeira iteração executada pelo decodificador $BCJR_1$. A decodificação do código $\mathcal{C}_1$ começa usando $\mathbf{y}_s$ e $\mathbf{y}_{p1}$. A informação extrinseca do código $\mathcal{C}_1$ é dada pela EQ 3.4.

$$L_e^1(\hat{m}_k) = L^1(\hat{m}_k) - L_c \cdot y_k \tag{3.4}$$

Esta estimação independente de m_k será usada na segunda parte da primeira iteração como um valor *a priori* para decodificar $\mathcal{C}_2$ conforme mostrado na EQ 3.5. Esta informação extrínseca será usada na próxima iteração como novo valor a priori.

$$L_e^2(\hat{m}_k) = L^2(\hat{m}_k) - (L_c \cdot y_k + L_e^1(\hat{m}_k)) \tag{3.5}$$

É possível observar que após a primeira iteração os valores de $L_e^1(\hat{m}_k)$ e $L_e^2(\hat{m}_k)$ são estatíticamente independentes mas que, a medida que as iterações forem se sucedendo, seus valores ficarão cada vez mais correlatados e a melhoria através das iterações será apenas marginal. Na última iteração, os valores da informação extrínseca dos dois códigos deverão ser combinadas com os valores recebidos conforme EQ 3.6 antes de efetuar a decisão.

$$L(\hat{m}_k) = L_c \cdot y_k + L_e^1(\hat{m}_k) + L_e^2(\hat{m}_k) \tag{3.6}$$

3.3 DESEMPENHO

O excelente desempenho propiciado pelos códigos Turbo pode ser melhor compreendido usando a análise de enumeração de pesos introduzida no APÊNDICE 2. Seja $A_w^{\mathcal{C}_i}(Z)$,

$i = 1, 2$, a IRWEF dos códigos constituintes $\mathcal{C}_1$ e $\mathcal{C}_2$ do PCE. Usando um entrelaçdor uniforme, ou seja, a probabilidade de ocorrer a mesma seqüência na saída do entrelaçador é uniforme entre todas as seqüências possíveis, é possivel estabelecer a IRWEF do PCE através da EQ 3.7.

$$A_w^{PCE}(Z) = \frac{A_w^{C_1}(Z) A_w^{C_2}(Z)}{\binom{K}{w}} \tag{3.7}$$

Usando a EQ 9.48, a taxa de erro de bit do PCE sobre um canal AWGN pode ser majorada pela EQ 3.8.

$$P_b(e) \leq \frac{1}{K} e^{d_{free}^H R_c E_b/N_0} Q\left(\sqrt{2 d_{free}^H R_c E_b/N_0}\right) W \left. \frac{\partial \sum_{w=0}^{K} W^w A_w^{PCE}(Z)}{\partial W} \right|_{W=Z=e^{-R_c E_b/N_0}} \tag{3.8}$$

Antes de substituir a EQ 3.7 na EQ 3.8, é preciso estabelecer uma expressão para cada BCE usado no PCE. Para isto é preciso definir o conceito de evento erro singular.

Definição 3.1 (Evento erro singular) *É a seqüência de saída não nula associada a um percurso no diagrama de estados do BCE que começa e termina no estado zero sem nunca ter passado no estado zero durante o percurso.*

Neste ponto é necessário saber de quantas formas a seqüência de informação associada a um evento erro singular pode ser arrumada dentro do entrelaçador. Caso o tamanho do entrelaçador N seja muito maior que a ordem de memória do BCE constituinte, então o comprimento da seqüência de informação não nula associada ao evento erro singular será muito menor que N. Deste modo, as seqüências de informação associadas a m eventos erro singular podem ser arrumadas aproximadamente de $\binom{N}{m}$ diferentes maneiras dentro do entrelaçador.

Seja então a EQ 3.9, denominada função enumeradora de m eventos.

$$A_w^{(m)}(Z) = \sum_z a_{w,z}^{(m)} Z^z \tag{3.9}$$

onde:

- $a_{w,z}^{(m)}$ é o número de palavras-código formadas pela concatenação de m eventos de erro simples
- w é o peso da palavra-informação
- z é o peso da palavra de paridade

Se m for 1, os coeficientes $a_{w,z}^{(m)}$ são exatamente aqueles obtidos da IRWEF.

A função enumeradora de pesos condicional do BCE pode ser aproximada pela EQ 3.10.

$$A_w^{BCE}(Z) \approx \sum_{m=1}^{m_{max_w}} \binom{K}{m} A_w^{(m)}(Z) \tag{3.10}$$

onde m_{max_w} é o maior número de eventos erro singular que podem ser associados com um seqüência de informação de peso w em ambos os codificadores constituintes ($m_{max_w} \leq w$).

A função enumeradora de pesos condicional do PCE dada pela EQ 3.11 é obtida pela substituição de 3.10 em 3.7.

$$A_w^{PCE}(Z) \approx \sum_{m_1=1}^{m_{max_w}} \sum_{m_2=1}^{m_{max_w}} \frac{\binom{K}{m_1}\binom{K}{m_2}}{\binom{K}{w}} A_w^{(m_1)}(Z) A_w^{(m_2)}(Z) \tag{3.11}$$

Fazendo $\binom{N}{m} \approx \frac{N^m}{m!}$ e $m_1 = m_2 = m_{max_w}$ obtem-se a EQ 3.12.

$$A_w^{PCE}(Z) \approx \frac{w!}{m_{max_w}!^2} K^{2m_{max_w}-w} \left[A_w^{m_{max_w}}(Z)\right]^2 \tag{3.12}$$

Substituindo 3.12 em 3.8 obtem-se a expressão 3.13 de desempenho em canal AWGN.

$$\begin{aligned} P_b(e) &\approx e^{d_{free}^H R_c E_b/N_0} Q\left(\sqrt{2 d_{free}^H R_c E_b/N_0}\right) \\ &\cdot \sum_{w=w_{\min}}^{K} \frac{w.w!}{m_{max_w}!^2} K^{2m_{max_w}-w-1} \\ &\cdot W^w \left[A_w^{m_{max_w}}(Z)\right]^2\Big|_{W=Z=e^{-R_c E_b/N_0}} \end{aligned} \tag{3.13}$$

onde w_{min} é o peso mínimo da informação para eventos de erro simples nos BCE componentes.

Se o BCE for não recursivo, tem-se $w_{min} = 1$ e $m_{max_w} = w$. Neste caso é fácil demonstrar que o tamanho do entrelaçador não tem qualquer influência na taxa de erro de bit (WICKER, 1995).

Caso o BCE seja recursivo, tem-se que $w_{min} = 2$ e $m_{max_w} = \lfloor w/2 \rfloor$(WICKER, 1995). Assim basta analisar os casos em que w seja par (BENEDETTO, 1996).

$$\begin{aligned} P_b(e) &\approx e^{d_{free}^H R_c E_b/N_0} Q\left(\sqrt{2 d_{free}^H R_c E_b/N_0}\right) \\ &\cdot \sum_{j=1}^{\lfloor N/2 \rfloor} 2j \binom{2j}{j} N^{-1} W^{2j} \left[A_{2j}^{(j)}(Z)\right]^2\Big|_{W=Z=e^{-R_c E_b/N_0}} \end{aligned} \tag{3.14}$$

onde $A_{2j}^{(j)}(Z) \approx A_2^{(1)}(Z)^j$.

A análise de $A_{2j}^{(j)}(Z)$ é então reduzida a um exame das palavras-código associadas a palavras-informação de peso 2 na entrada do BCE recursivo. De modo geral, as palavras-código produzidas por este codificador devem possuir um bloco de informação de peso 2 e um bloco de paridade de peso $jz_{ciclo}+t$, onde t é o peso associado ao percurso do estado zero até o ciclo [1], z_{ciclo} é o peso da paridade do ciclo e j o número de voltas completadas no ciclo. Assim, o menor peso de seqüência de paridade gerada por um BCE IIR será $z_{min} = z_{ciclo} + t$ e a função enumeradora de eventos erro singular será dada pela EQ 3.15.

$$\begin{aligned} A_2^{(1)}(Z) &\approx Z^{z_{ciclo}+t} + Z^{2z_{ciclo}+t} + Z^{z_{ciclo}+t} + \cdots \\ &= Z^{z_{min}} + Z^{2z_{min}-t} + Z^{3z_{min}-2t} + \cdots \\ &= \frac{Z^{z_{min}}}{1 - Z^{z_{min}-t}} \end{aligned} \tag{3.15}$$

Substituindo a EQ 3.15 na EQ 3.14 e fazendo $W = Z = H$ tem-se finalmente a EQ 3.16.

$$\begin{aligned} P_b(e) &\approx e^{d_{free}^H R_c E_b/N_0} Q\left(\sqrt{2 d_{free}^H R_c E_b/N_0}\right) \\ &\cdot \sum_{j=1}^{\lfloor N/2 \rfloor} 2j \binom{2j}{j} N^{-1} W^{2j} \left. \frac{(H^{2+2z_{min}})^j}{(1 - H^{z_{min}-2})^{2j}} \right|_{W=Z=e^{-R_c E_b/N_0}} \end{aligned} \tag{3.16}$$

onde a quantidade $2 + 2z_{min}$ é normalmente definida como **distância livre efetiva** do PCE. Da EQ 3.16 observa-se que o desempenho do PCE em canal AWGN depende de dois fatores:

- O tamanho do entrelaçamento N.
- A distância livre efetiva.

Neste último caso, um bom desempenho é condicionado a uma boa escolha do entrelaçador tal que nenhum bloco de informação de peso $w > 2$ venha a ter bloco de paridade de peso menor que z_{min}.

A TAB. 3.1 resume os melhores codificadores IIR de taxa 1/2 (WICKER, 1995).

[1] ciclo é percurso fechado no diagrama de estados do BCE associado ao bloco de zeros existente entre os dois bits 1.

TAB. 3.1: Melhores códigos RSC componentes de taxa 1/2 para PCE de taxa 1/3 com entrelaçador de tamanho 100

M	$g(D)$	$f(D)$	$d_{free,eff}$	d_{free}^{PCE}	w_{free}^{PCE}
1	3	2	4	4	2
2	7	5	10	7	3
3	15	17	14	8	4
4	31	33	22	9	5
4	31	27	22	9	5
5	51	77	38	10	6
5	51	67	38	12	4

3.3.1 CANAL AWGN

A FIG. 3.4 ilustra as curvas de probabilidade de erro de bit nos casos de decisão suave e abrupta em canal AWGN. As simulações foram feitas com blocos de 8184 *bits* e modulação BPSK. É possível observar que em $P_e = 10^{-5}$ ocorre um perda $1,5\ dB$ quando é usada a decisão abrupta em lugar da suave. É importante observar que em códigos convolucionais a perda ao fornecer a decisao abrupta ao decodificador de Viterbi é de cerca de $2,0\ dB$ (VITERBI, 1979) para a mesma probabilidade de erro de bit. Conclui-se então que no caso decodificador Turbo, há um ganho de iteratividade obtido pela decodificação iterativa.

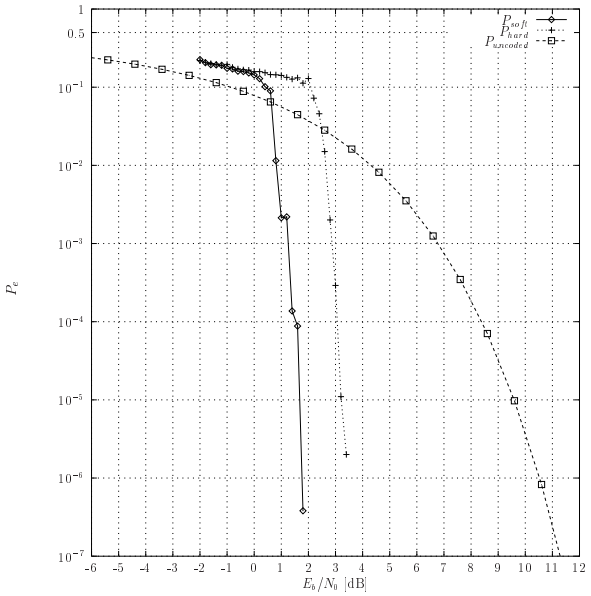


FIG. 3.4: Curvas de desempenho do código Turbo em canal AWGN

4 REDE RÁDIO HF

A disseminação do uso da Internet como meio de comunicação tem proporcionado uma revolução no modo de trabalhar tanto das pessoas como das empresas. Este disseminação da Internet só foi possível graças à flexibilidade de acessar a rede de praticamente qualquer lugar do planeta. Para que este acesso fosse possível houve grande esforço no desenvolvimento de tecnologias que permitissem o acesso à rede através do sistema telefônico fixo e móvel, que juntos constituem a rede de maior cobertura geográfica existente.

Um característica técnica da Internet muito importante para o seu largo emprego e disseminação é fato dela usar um protocolo de comunicação denominado TCP/IP (TANENBAUM, 1989). Este protocolo permite o envio e recebimento de informação através do uso de qualquer tecnologia física de comunicação existente. Desta forma, o TCP/IP funciona como uma aplicação de rede capaz de possibilitar a comunicação entre usuários conectados a redes heterogêneas, ou seja, que utilizem de diferentes tecnologias para transmissão de informação.

Atualmente, tem-se realizado grande esforço no desenvolvimento de sistemas e tecnologias capazes de promover a comunicação móvel pessoal. Tais sistema são mais conhecidos pela sigla inglesa PCS (*Personal Communication Systems*) e operam na faixa de freqüência UHF (*Ultra High Frequency*). Este sistema corresponde ao que se convencionou denominar de 3ª Geração do Sistema Móvel Celular (SMC). Uma das aplicações disponíveis neste sistema é o acesso a serviços da Internet tais como o de mensagens eletrônicas.

O desenvolvimento e implementação do acesso à Internet em sistemas PCS mostra que também é possível e viável fazê-lo usando outras tecnologias de transmissão e outras faixas de freqüências tais como VHF (*Very High Frequency*) e HF (*High Frequency*). Uma rede de rádios operando na faixa HF, por exemplo, permite, com estações transmissoras localizadas em terra, comunicar-se com regiões remotas da Terra sem necessidade de uso de satélites e/ou qualquer outra infraestrutura de comunicações.

A comunicação na faixa HF é muito utilizada por empresas com unidades localizadas em regiões remotas, navios em alto mar e forças armadas. O principal tipo de tráfego destas redes é o de voz. O tráfego de dados nesta faixa é pequeno, pois a velocidade de transmissão é baixa e a taxa média de erros alta. Tal desempenho é determinado

pelo fato do canal HF estar sujeito a efeitos de desvanecimento seletivo tanto no tempo como na freqüência. Este desvanecimento tem origem nos mecanismos de propagação, que nesta faixa ocorre na camada ionosférica.

O uso de uma rede rádio HF não apenas para transmissão de dados, mas também para ter acesso à Internet ou, simplesmente, estabelecer uma rede privativa de dados não é uma idéia nova. Entretanto, o estudo da rede HF ao transportar tráfego TCP/IP pode mostrar quais aspectos das camadas física e de enlace tem maior ou menor influência no desempenho da camada de rede. Um dos aspectos da rede HF que poderia ser melhorado sem necessidade de alterações de equipamentos é a inclusão de códigos corretores de erro no tráfego para melhorar a taxa de erros.

Neste capítulo, serão apresentados alguns modelos de redes de comunicações e diagramado o modelo de rede a ser empregado neste trabalho.

4.1 MODELOS DE REDES DE COMUNICAÇÕES

Para facilitar a análise de uma rede de comunicações, costuma-se dividir a rede em camadas onde cada uma possua uma ou mais tarefas bem definidas. Como exemplo, é comum separar a função de envio de dados da função de gerenciamento de conexão em níveis diferentes. Deste modo, torna-se mais simples a concepção de cada camada e facilita-se o projeto da rede como um todo, pois este será formado por partes cada qual com tarefas bem definidas e distintas.

Normalmente, associa-se a cada camada um ou mais protocolos de acordo com as tarefas atribuídas àquela camada. Um protocolo, segundo a definição em (COMER, 1991) é uma descrição formal de formatos de mensagem e de regras que duas ou mais entidades que desejem se comunicar devem seguir a fim de trocar mensagens entre elas.

Os modelos de rede mais importantes são o modelo da Internet criado pelo DoD (*Department of Defense*) e o modelo OSI (*Open Systems Interconnection*) de sete camadas da ISO (*International Organization for Standardization*). A Internet representa a fusão destes dois modelos.

4.1.1 MODELO TCP/IP

O modelo de quatro camadas TCP/IP foi desenvolvido em 1970 pelo DoD em conjunto com algumas universidades americanas para o projeto da rede do DARPA (*Defense Advanced Research Project Agency*) denominada de ARPAnet. Esta rede viria mais tarde a crescer e se transformar na própria Internet. O núcleo dos protocolos da Internet adere

a este modelo, embora o modelo OSI de sete níveis seja preferido para novos projetos. A FIG. 4.1 apresenta as camadas do modelo TCP/IP.

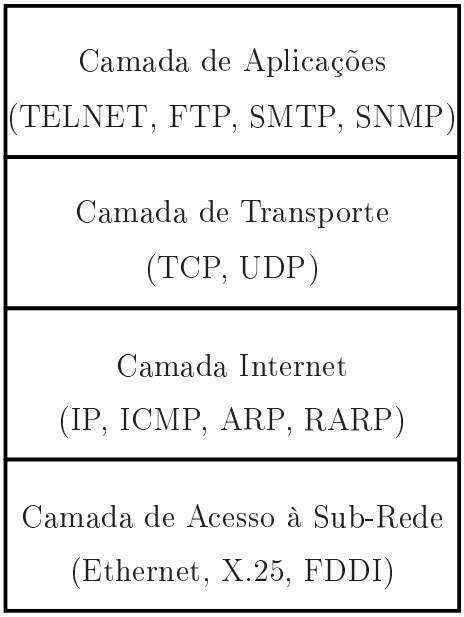


FIG. 4.1: Modelo de quatro camadas DoD

As quatro camadas do modelo TCP/IP, da base para o topo, são:

1. **Camada de Acesso à Sub-Rede** é responsável por enviar dados através do meio físico particular que estiver sendo usado. Diferentes protocolos poderão ser selecionados para este nível, , dependendo do tipo de rede física. Exemplos: Ethernet, Token Ring, FDDI (*Fiber Distributed Data Interface*), X.25, Frame Relay, ATM.

2. **Camada Internet** é responsável por enviar dados através de uma série de diferentes redes físicas que conecte a fonte ao destino da mensagem. Exemplos: IP (*Internet Protocol*), ICMP (*Internet Control Message Protocol*), ARP (*Address Resolution Protocol*), RARP (*Reverse Address Resolution Protocol*).

3. **Camada de Transporte** é responsável pelo controle da conexão, controle de fluxo, retransmissão de dados perdidos e outros controles sobre fluxo de dados. Exemplos: TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), etc.

4. **Camada de Aplicação** é responsável pelas funções a nível de usuário tais como envio de e-mail, transferência de arquivos e acesso remoto. Exemplos: SMTP (*Simple Mail Transfer Protocol*), FTP (*File Transfer Protocol*), TELNET, SNMP (*Simple Network Management Protocol*).

4.1.2 MODELO OSI

Em 1980, a ISO começou a desenvolver um modelo aberto de redes denominado OSI. O modelo era formado de dois grandes componentes: um modelo abstrato de rede e um conjunto de protocolos para estes níveis. O modelo abstrato também é conhecido como modelo de referência básico, ou modelo de sete camadas.

Algumas partes do modelo de referência básico OSI influenciaram nos desenvolvimentos posteriores de protocolos para Internet. Por este modelo, uma rede é dividida em sete camadas conforme ilustrado na FIG. 4.2. Dentro de cada camada, uma ou mais entidades implementam sua funcionalidade. Cada entidade interage diretamente apenas com a camada imediatamente abaixo da sua camada, e provê facilidades usadas pela camada acima dela. Os protocolos permitem que uma entidade em um ponto da rede seja capaz de interagir com uma entidade correspondente na mesma camada em um outro ponto da rede.

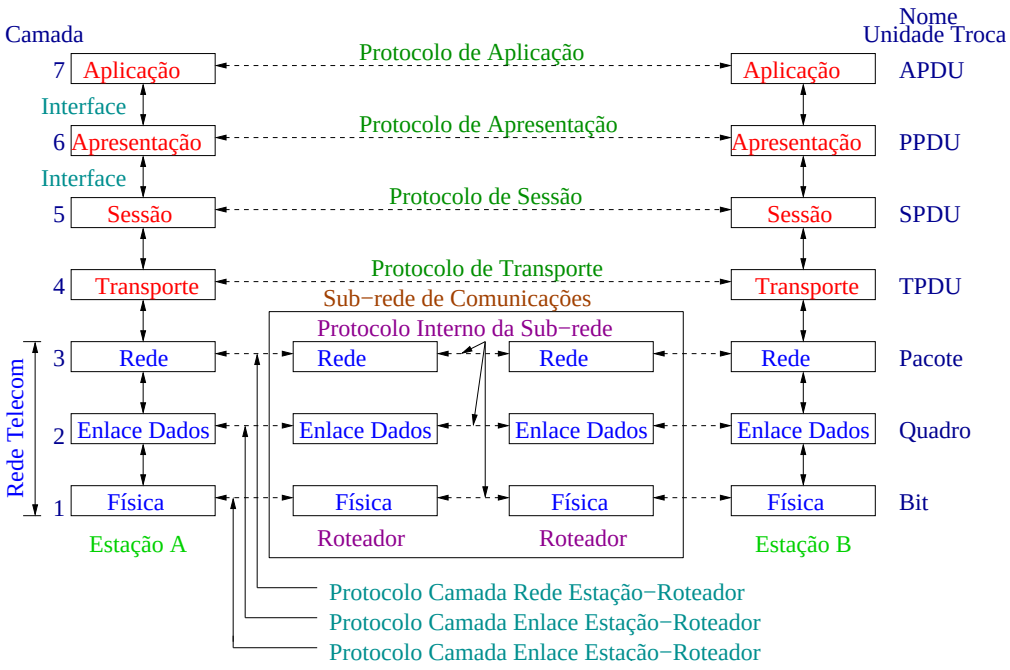


FIG. 4.2: As camadas do modelo de referência OSI

As sete camadas do Modelo de referência básica OSI, da base para o topo, são:

1. **Camada Física** descreve as características mecânicas, elétricas, funcionais e procedimentais dos vários meios de comunicação. Este nível define em uma rede do tipo Ethernet, por exemplo, a interpretação dos sinais trocados, o tamanho do cabo coaxial, o tipo do conector BNC usado e o método de terminação.

2. **Camada de Enlace** descreve a organização lógica dos bits de informação transmitidos em um meio particular. É a última camada do modelo OSI responsável pela comunicação física entre dois pontos da rede. Exemplo: define o enquadramento, endereçamento e checagem de soma de pacotes Ethernet.

3. **Camada de Rede** descreve como um série de trocas de bits sobre vários enlaces pode levar a informação entre dois nós da rede. Exemplo: define o endereçamento e estrutura de roteamento da rede.

4. **Camada de Transporte** descreve a qualidade e a natureza da transmissão da informação. Exemplo: este nível define se e como serão usadas retransmissões para assegurar o envio da informação.

5. **Camada de Sessão** descreve a organização de seqüências de dados maiores do que os pacotes manuseados pelas camadas inferiores. Exemplo: este nível descreve como pacotes de pedido e resposta são atendidos durante chamadas de procedimentos remotos.

6. **Camada de Apresentação** descreve a sintaxe das informações que estão sendo transmitidas. Exemplo: este nível descreve como representações de números em ponto flutuante podem ser trocadas entre sistemas que usem formatos matemáticos diferentes.

7. **Camada de Aplicação** descreve como as aplicações podem ter acesso aos serviços da rede. Exemplo: este nível implementa as operações de sistemas de arquivos.

4.2 MODELO DA REDE RÁDIO HF

O modelo da rede rádio HF a ser estudada neste trabalho inclui apenas as três primeiras camadas do modelo OSI, ou seja, as camadas física, de enlace e de rede. A FIG. 4.3 mostra uma rede TCP/IP que usa como camada física uma rede rádio HF.

A camada física da rede rádio HF é formada por estações rádio compostas cada uma por um rádio transmissor/receptor HF denominado DCE (*Data Communication Equipment*) e um computador denominado DTE (*Data Terminal Equipment*) interligados por meio de uma porta de comunicação serial. Tomado no contexto de uma rede completa (sete camadas), este conjunto de nós constituído por estações HF constitui uma sub rede de comunicações onde cada nó é denominado PSN (*Packet Switch Node*). As funções típicas de um PSN são descritas em (TANENBAUM, 1989).

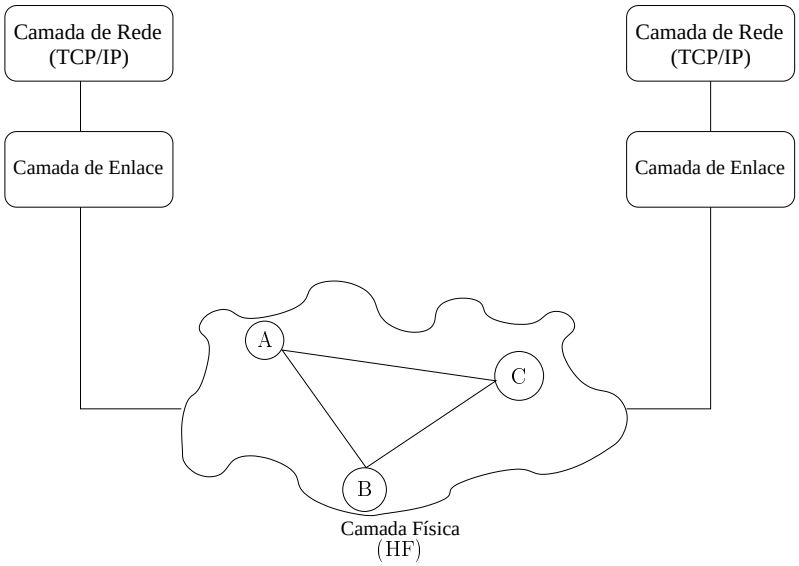


FIG. 4.3: Camadas da rede TCP/IP sobre HF

A FIG. 4.4 ilustra uma estação HF composta por um computador e um rádio com modem ligados por meio de uma interface padrão EIA-TIA RS-232C e MIL-STD-188-184.

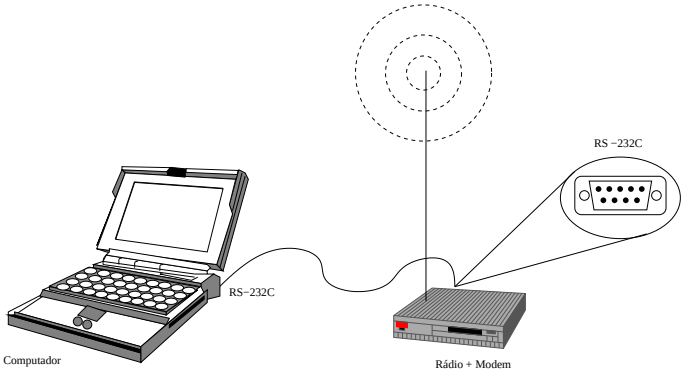


FIG. 4.4: Componentes de uma estação da subrede

O rádio utilizado é constituído de modem externo ou interno, opera na faixa HF e no modo half-duplex. Desta forma, cada vez que se deseje inverter o sentido da comunicação, será necessário aguardar uma indicação de canal livre para então acionar a tecla PTT (*Press To Talk*) para transmissão de voz ou, de modo equivalente, ativar o sinal RTS (*Request To Send*) da porta serial para solicitar o estabelecimento de enlace para transmissão de dados.

O meio de transmissão utilizado pelo rádio é compartilhado pelas estações sempre que estejam sendo usadas as mesmas freqüências e as estações estejam ao alcance umas das outras. Como as transmissões em uma rede HF são quase sempre ponto-a-ponto, as estações da rede não alcançadas em um único salto poderão se comunicar utilizando-se

do recurso de armazenagem para retransmissão (*Store and Forward*) em estações intermediárias. Do mesmo modo, quando não houver uma rota direta ou esta estiver congestionada, a subrede será capaz de rotear a ligação.

A descrição detalhada dos funcionamento e interoperabilidade da sub rede HF é descrita nos padrões MIL-STD-188-110A , MIL-STD-187-721C e MIL-STD-188-141A. A FIG. 4.5 ilustra a relação destes padrões com as amadas do modelo OSI.

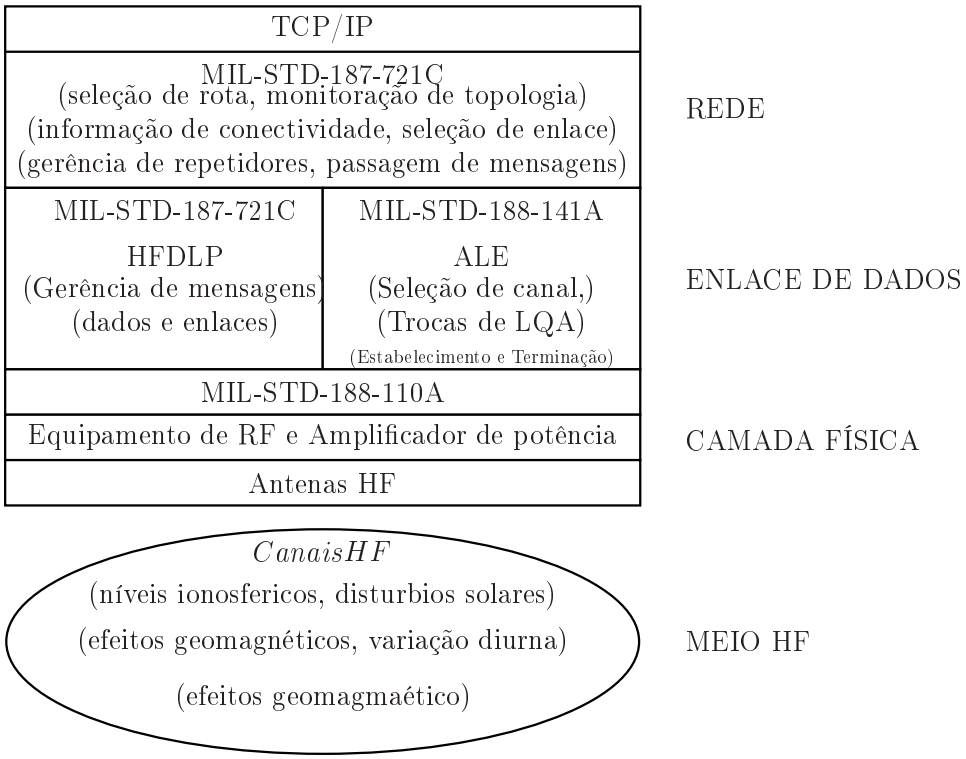


FIG. 4.5: Relação entre padrões militares e camadas do modelo OSI

5 CAMADA FÍSICA

Em qualquer rede de comunicações, a camada física assume um papel preponderante não apenas para o projeto da rede como no desempenho da mesma. No caso de um rede rádio, também denominada rede sem fio, a modelagem eficiente e correta do canal de propagação a ser usado é de fundamental importância na definição das características técnicas do modem a ser empregado. Conjugados, o canal e o modem constituem a camada física da rede.

No caso de um rede rádio HF, a camada fisíca é constituida obviamente pelo canal de propagação ionosférico [2] e pelo modem rádio adotado. No caso deste trabalho, o modelo de canal HF é o de Watterson (WATTERSON, 1970) e o modem rádio de dados adotado é o modem de tom serial 8PSK definido pelo padrão MIL-STD-188-110A (DoD, 1991).

5.1 CANAL

Um canal de comunicações multipercurso com desvanecimento é geralmente caracterizado como um sistema linear e variante no tempo que possui uma resposta impulsiva $h(t,\tau)$ (ou a resposta em freqüência variante no tempo $T(f,t)$) que é um processo estocástico (PE) estacionário no sentido amplo (WSS - *Wide Sense Stationary*) na variável t. Este processo representa a resposta do canal no tempo t devida à aplicação de um impulso no tempo $t-\tau$. A FIG. 5.1 ilustra o sistema formado pelo canal e suas entradas e saídas.

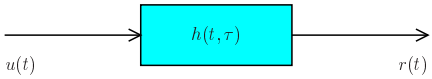


FIG. 5.1: Resposta impulsiva do canal

onde:

$u(t), r(t)$	são os sinais equivalentes passa-baixa da entrada e saída, respectivamente
$\tau = t - t_0$	é a diferença entre o instante atual e o da aplicação do impulso

As relações entre os sinais de entrada e saída passa-baixa e passa-faixa são dadas respectivamente pelas equações 5.1 e 5.2.

[2] Para distâncias entre estações inferiores a 15 Km a propagação por ondas de superfície também deve ser levada em consideração

$$x(t) = Re[u(t)e^{j2\pi f_c t}] \tag{5.1}$$

$$y(t) = Re[r(t)e^{j2\pi f_c t}] \tag{5.2}$$

Variações temporais na resposta impulsiva do canal ou resposta em freqüência resulta no espalhamento em freqüência, geralmente denominado espalhamento Doppler, do sinal transmitido pelo canal. A propagação multipercurso resulta no espalhamento temporal do sinal transmitido. Consequentemente, um canal multipercurso com desvanecimento pode ser caracterizado por um espalhamento duplo no tempo e freqüência.

O multipercurso pode ser tipo discreto ou contínuo conforme os retardos seja fixo ou variável. As equações 5.3 e 5.4 ilustram a forma geral da resposta impulsiva do canal para o caso discreto e contínuo, respectivamente.

$$h(t,\tau) = \sum_{i=1}^{L} \alpha_i(t) e^{-j2\pi f_c \tau_i(t)} \delta[\tau - \tau_i(t)] \tag{5.3}$$

onde:

L	é o número de percursos do sinal pelo canal
$\alpha_i(t)$	é o PE do ganho do percurso i
$\tau_i(t)$	é o PE do retardo do percurso i

$$h(t,\tau) = \alpha(t,\tau) e^{-j2\pi f_c \tau} \tag{5.4}$$

onde $\alpha(t,\tau)$ é o ganho de percurso.

O multipercurso é causado pela existência de espalhadores entre o transmissor e receptor. Quando estes espalhadores possuem movimentos aleatórios, a resposta $h(t,\tau)$ é modelada por um PE Gaussiano complexo de média zero. Casos os espalhadores sejam fixos, o que caracteriza o mecanismo de reflexão, $h(t,\tau)$ é modelada por um PE Gaussiano de média não-nula.

5.1.1 CANAL WSS-US

A teoria usada para estudo de canais variantes no tempo foi introduzida por Bello (BELLO, 1963) em 1963. No seu trabalho, Bello introduziu as funções do sistema canal baseando-se na teoria de processos estocásticos e das transformadas de Fourier. As funções de Bello e suas relações estão ilustradas na FIG. 5.2.

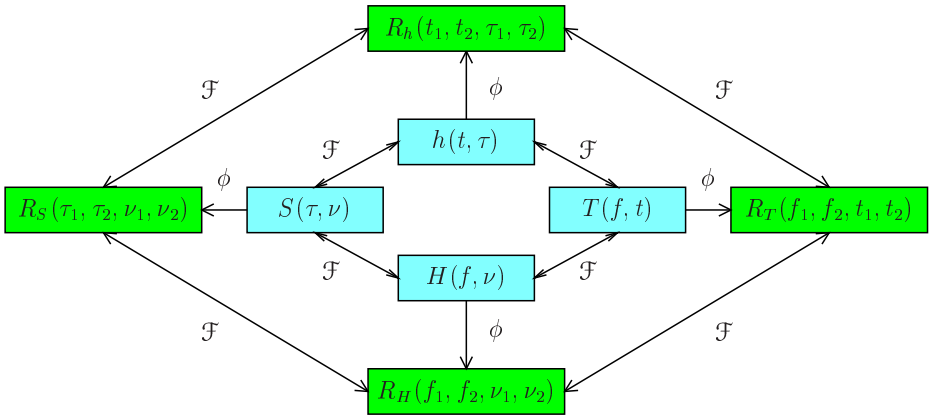


FIG. 5.2: As relações entre as funções de Bello e suas autocorrelações

onde:

t	é o instante de observação
τ	é o retardo em relação ao instante de aplicação do impulso $\delta(t - t_0)$
ν	é denominada de freqüência Doppler ou freqüência deslocada em relação à portadora ou ainda deslocamento Doppler
f	é a freqüência
$h(t, \tau)$	é a resposta impulsiva do canal no instante t_0
$T(f, t)$	é a resposta em freqüência do canal
$H(f, \nu)$	é a transformada de Fourier da resposta em freqüência do canal
$S(\tau, \nu)$	é a função de espalhamento do canal

Na FIG. 5.2 observa-se que as funções externas são autocorrelações das internas conforme mostrado por exemplo na EQ 5.5.

$$R_h(t_1, t_2, \tau_1, \tau_2) = \frac{1}{2} E[h(t_1, \tau_1) \cdot h^*(t_2, \tau_2)] \tag{5.5}$$

e, dentro do conjunto de funções internas e externas, as relações entre funções são obtidas por transformadas de Fourier. A regra do sinal da exponecial é a seguinte: quando a variável transformada envolve mudança do tempo para freqüência o sinal é negativo caso contrário o sinal é positivo. Desta forma tem-se por exemplo a EQ 5.6

$$S(\nu, \tau) = \int_{-\infty}^{\infty} h(t, \tau) e^{-j2\pi t\nu} dt \tag{5.6}$$

Como o processo estocástico complexo $h(t, \tau)$ pode ter suas componentes conjuntamente e individualmente WSS na variável t, as funções de autocorrelação serão dadas pelas equações 5.7 e 5.8.

$$R_h(t_1, t_2, \tau_1, \tau_2)_{WSS} = R_h(\Delta t, \tau_1, \tau_2) \tag{5.7}$$

$$R_h(f_1, f_2, t_1, t_2)_{WSS} = R_T(f_1, f_2, \Delta t) \tag{5.8}$$

Assim a transformada de Fourier da função de autocorrelação de um PE WSS dá origem a uma função não correlatada na variável transformada expressa pelas equações 5.9 e 5.10.

$$R_S(\tau_1, \tau_2, \nu_1, \nu_2)_{WSS} = P_S(\tau_1, \tau_2, \nu)\delta(\nu_1 - \nu_2) \tag{5.9}$$

$$R_H(f_1, f_2, \nu_1, \nu_2)_{WSS} = P_H(f_1, f_2, \nu)\delta(\nu_1 - \nu_2) \tag{5.10}$$

Se o PE que caracteriza o canal for também WSS na variável freqüência, significa que as funções não são dependentes da freqüência e apenas a separação de freqüência é importante. Desta forma, tem-se as equações 5.11 e 5.12.

$$R_H(f_1, f_2, \nu_1, \nu_2)_{WSS} = R_H(\Delta f, \nu_1, \nu_2) \tag{5.11}$$

$$R_T(f_1, f_2, t_1, t_2)_{WSS} = R_T(\Delta f, t_1, t_2) \tag{5.12}$$

E, analogamente, tem-se as autocorrelacões expressas pelas equações 5.13 e 5.14.

$$R_S(\tau_1, \tau_2, \nu_1, \nu_2)_{WSS} = P_S(\tau, \nu_1, \nu_2)\delta(\tau_2 - \tau_1) \tag{5.13}$$

$$R_h(t_1, t_2, \tau_1, \tau_2)_{WSS} = P_h(t_1, t_2, \tau)\delta(\tau_2 - \tau_1) \tag{5.14}$$

Das equações 5.13 e 5.14 conclui-se que canais WSS na freqüência são canais nos quais os espalhamentos são descorrelacionados US (*Uncorrelated Scattering*). Desta forma os canais e suas funções que sejam WSS no tempo e na freqüência são denominados de WSS-US. A FIG. 5.3 ilustra as relações entre as funções de autocorrelação de um canal WSS-US.

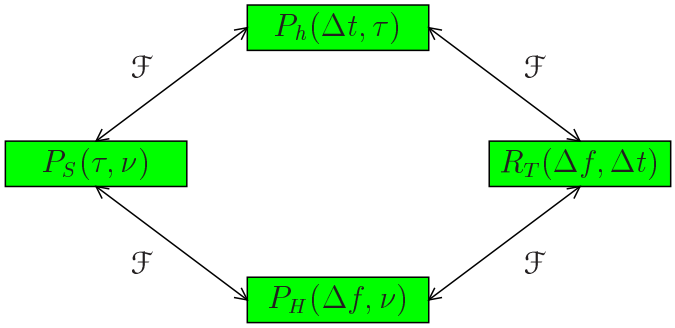


FIG. 5.3: Relações entre as autocorrelações em um canal WSS no tempo e na freqüência

A função de autocorrelação da função de espalhamento do canal $P_S(\tau, \nu)$ representa uma medida de potência espectral no canal com retardo τ e deslocamento de freqüência ν, relativo à freqüência de portadora. A partir desta função, obtem-se o espectro de potência de retardo ou perfil de intensidade de multipercurso dado por 5.15

$$P_S(\tau) = \int_{-\infty}^{\infty} P_S(\tau, \nu) d\nu \tag{5.15}$$

e o espectro de potência Doppler 5.16

$$P_S(\nu) = \int_{-\infty}^{\infty} P_S(\tau, \nu) d\tau \tag{5.16}$$

A faixa de valores sobre os quais o espectro de potência de retardo $P_S(\tau)$ é não-nulo é definido como o espalhamento multipercurso T_m do canal. De modo semelhante, a faixa de valores sobre os quais o espectro de potência Doppler é não-nulo é definido como (banda de) espalhamento Doppler B_d do canal. Quando $P_S(\nu) = \delta(\nu)$ não se observa espalhamento espectral produzido pelo canal $P_S(\tau) = \delta(\tau)$.

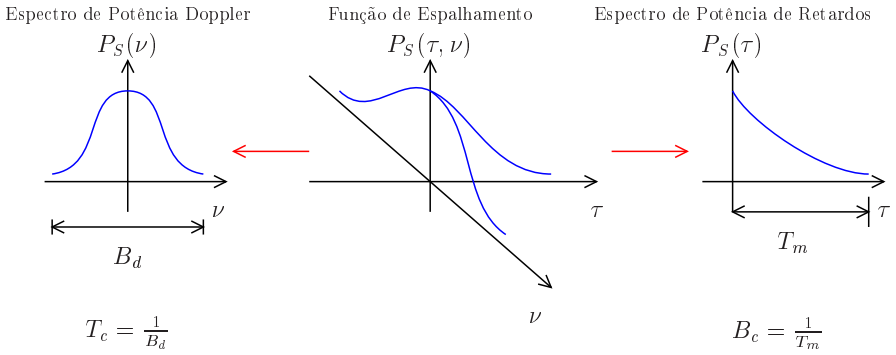


FIG. 5.4: Função de espalhamento do canal

O valor do espalhamento Doppler B_d mede a rapidez de variação da resposta impulsiva do canal com o tempo.

5.1.2 MODELO DE WATTERSON

Os canais ionosféricos HF em geral são não estacionários na freqüência e no tempo, porém se a consideração é restrita a canais limitados em banda (10 Khz, por exemplo) e tempos suficientemente pequenos (10 minutos, por exemplo), a maioria dos canais pode ser adequadamente representada por modelos WSS-US.

Há basicamente três fenomenos responsáveis pelos efeitos de multipercurso e espalhamento Doppler presentes na propagação HF para comunicações BLOS (*Beyond Line Of Sight*):

- Variações da densidade de ionização com a altitude causando a existência de raio alto e baixo.
- Efeito magnetoiônico causando percursos dependentes da polarização (onda ordinária e extraordinária).
- Não-uniformidade das camadas ionosféricas e variações ao longo do tempo.

Cada um destes efeitos, conjugado ou isoladamente, ocasiona diferentes espalhamentos de tempo de retardo T_m e Doppler B_d.

Em 1970, Watterson e Juroshek propuseram um modelo baseado em experimentos estatísticos para o canal HF que foi posteriormente adotado pelo CCIR (CCIR, 1992) como padrão para o desenvolvimento de simuladores de canal HF.

Watterson propôs um modelo baseado em uma linha de retardos com ganhos de derivação (BIGLIERI, 1998). O modelo de Watterson admite que os espectros Doppler dos ganhos de derivação apresentem forma Gaussiana para cada uma das componentes magnetoiônicas (a ordinária e a extraordinária) produzidas pela reflexão ionosférica.

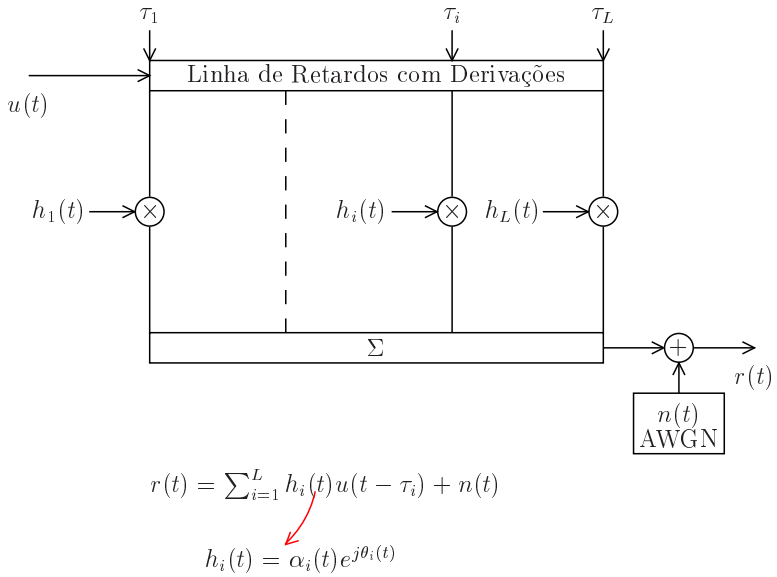


FIG. 5.5: Modelo de canal com linha de retardo e ganhos de derivação

De uma forma geral, os ganhos de derivação deste modelo são dados pela EQ 5.17.

$$h_i(t) = h_{io}(t)e^{j2\pi\nu_{io}t} + h_{ie}(t)e^{j2\pi\nu_{ie}t} \tag{5.17}$$

onde:

$h_{io}(t),\ h_{ie}(t)$ são dois processos estocásticos Gaussianos complexos, WSS, independentes entre si e de média zero que representam respectivamente as componentes ordinária e extraordinária

$\nu_{io},\ \nu_{ie}$ são os deslocamentos de freqüência das componentes ordinária e extraordinária respectivamente

A autocorrelação de cada ganho tem a forma definida pela EQ 5.18.

$$R_{h_i}(\Delta t) \triangleq \frac{1}{2}E[h_i(t)h_i^*(t+\Delta t)] \tag{5.18}$$

Substituindo EQ 5.17 na EQ 5.18, obtem-se a autocorrelação expressa pela EQ 5.19.

$$R_{h_i}(\Delta t) = R_{h_{io}}(0)e^{(-2\pi^2\sigma_{io}^2\Delta t^2 + j2\pi\nu_{io}\Delta t)} + R_{h_{ie}}(0)e^{(-2\pi^2\sigma_{ie}^2\Delta t^2 + j2\pi\nu_{ie}\Delta t)} \tag{5.19}$$

onde:

$R_{h_{io}}(0),\ R_{h_{ie}}(0)$ representam a razão de potência de saída entregue por cada componente pela potência de entrada no canal

$2\sigma_{io},\ 2\sigma_{ie}$ são os espalhamentos de freqüência de cada componente

A transformada de Fourier da autocorrelação da resposta impulsiva do canal dá origem à função de espalhamento.

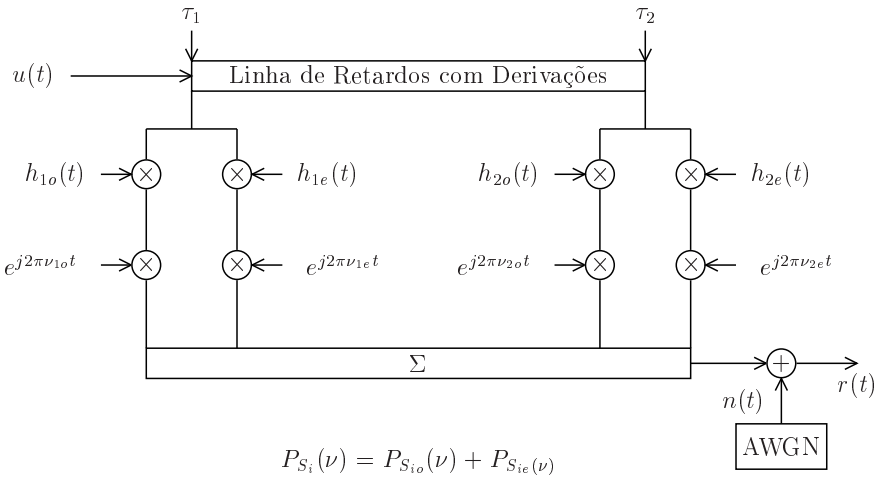
$$P_{S_i}(\nu) = \frac{R_{h_{io}}(0)}{(2\pi)^{1/2}\sigma_{io}}e^{-\frac{(\nu-\nu_{io})^2}{2\sigma_{io}^2}} + \frac{R_{h_{ie}}(0)}{(2\pi)^{1/2}\sigma_{ie}}e^{-\frac{(\nu-\nu_{ie})^2}{2\sigma_{ie}^2}} \tag{5.20}$$

A FIG. 5.6 ilustra um canal de propagação ionosférico que possui dois raios.

5.1.3 CANAIS DO CCIR

De modo a tornar os sistemas de comunicações interoperáveis e eficientes no compartilhamento da faixa HF, o antigo CCIR (atual ITU-R) padronizou a canalização. A banda passante do padrão de canalização adotado pelo CCIR para a faixa HF é de 3 KHz.

Além da canalização, O CCIR padronizou três canais típicos cujos parâmetros são mostrados na TAB. 5.1. Neste caso, o objetivo foi estabelecer um padrão de canal para



$$P_{S_i}(\nu) = P_{S_{io}}(\nu) + P_{S_{ie}(\nu)}$$

$$P_{S_{io}}(\nu) = \frac{R_{h_{io}}(0)}{(2\pi)^{1/2}\sigma_{io}} e^{-\frac{(\nu-\nu_{io})^2}{2\sigma_{io}^2}}$$

$$P_{S_{ie}}(\nu) = \frac{R_{h_{ie}}(0)}{(2\pi)^{1/2}\sigma_{ie}} e^{-\frac{(\nu-\nu_{ie})^2}{2\sigma_{ie}^2}}$$

FIG. 5.6: Canal de propagação HF em faixa-estreita com dois percursos

TAB. 5.1: Canais definidos pelo CCIR

Tipo Canal	Espalhamento de Freqüência Doppler (B_d) $[Hz]$	Espalhamento de Retardo Multipercurso (T_m) $[ms]$
Bom	$0,1$	$0,5$
Moderado	$0,5$	$1,0$
Ruim	$1,0$	$2,0$
Muito Ruim	$2,0$	$2,0$

comparar o desempenho de esquemas de modulação, filtragem, equalização e codificação a serem usados por equipamentos desta faixa.

De posse das informações contidas na TAB. 5.1 e da taxa de sinalização B_s usada pelo modulador é possível prever se determinado canal do CCIR é seletivo em freqüência ou não. Caso a desigualdade 5.21 seja obedecida, o canal pode ser considerado não seletivo e consequentemente não haverá interferência entre símbolos ISI (*Inter Symbol Interference*) causada por multipercurso mas apenas por restrição de faixa.

$$B_d \ll B_s \ll \frac{1}{T_m} \tag{5.21}$$

Considerando os parâmetros da TAB. 5.1 e a desigualdade 5.21, as taxas de sinalização ideais para os canais do CCIR bom, moderado e pobre são da ordem de 200, 100 e 50 *bauds*, respectivamente.

5.2 MODEM

O modem é um dispositivo capaz de transmitir e receber informações digitais através de um canal analógico. Para realizar esta tarefa o modem emprega uma conjunto de técnicas adequadas para um determinado conjunto de aplicações. Expandindo o bloco modulador e demodulador da FIG. 9.1 o diagrama em blocos de um modem genérico pode ser ilustrado pela FIG. 5.7.

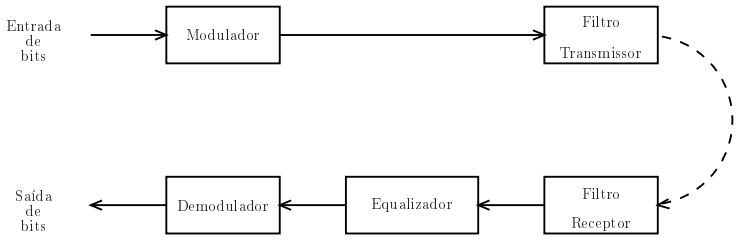


FIG. 5.7: Diagrama em blocos de um modem genérico

O padrão MIL-STD-188-110A (DoD, 1991) define completamente como devem funcionar os blocos do modulador e demodulador. Para o bloco equalizador é imposto que deve usar seqüências de treinamento com tamanho, formação e freqüência de repetição conforme descrito no padrão para equalização adaptativa.

Com relação aos filtros de transmissão e recepção nada é mencionado mas o CCIR definiu a faixa passante do canal HF em 3 KHz. Para este trabalho adotou-se o filtro do tipo raiz quadrada de cosseno levantado com fator de excesso de faixa de $\alpha = 0,25$ (HAYKIN, 2000). O filtro implementado é do tipo FIR com tamanho definido de acordo com a taxa de amostragem que por sua vez é definida de acordo com a maior freqüência contida no sinal modulado.

5.2.1 MODEM DE TOM SERIAL

O padrão MIL-STD-188-110A define as seguintes especificações para o modem de tom serial único:

- A modulação usada pelo modem é $8PSK$.
- O modem opera com taxa de sinalização constante e igual a 2400 *bauds*.
- O modem possui um codificador convolucional embutido com comprimento restritivo $K = 7$ e taxa $R_c = 1/2$.

- O modem possui um entrelaçador de tamanho ajustável de modo a manter o retardo constante para as diferentes taxas nominais de bit.

A Diagrama em blocos funcionais da FIG. 5.8 ilustra simplificadamente o modem serial.

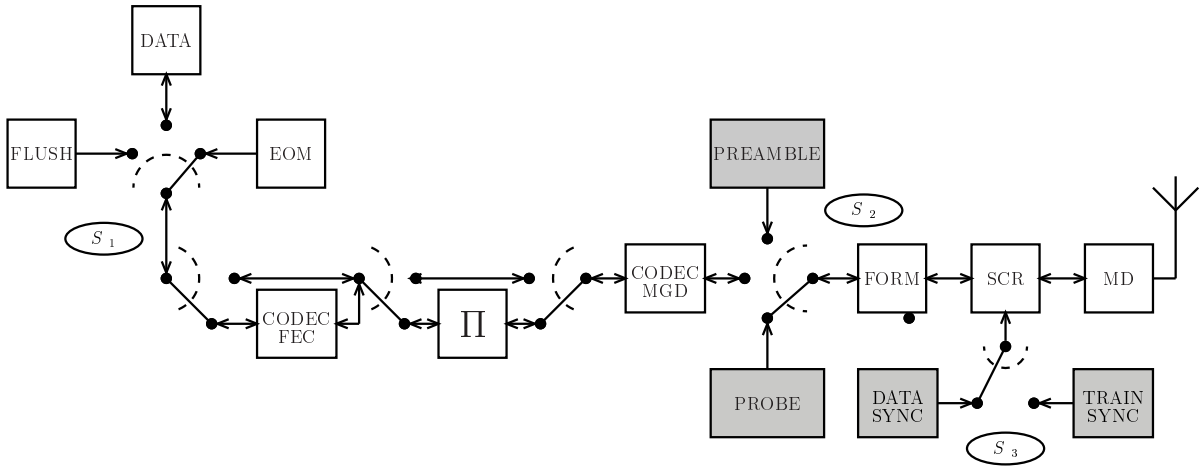


FIG. 5.8: Diagrama em blocos simplificado do modem serial

5.2.1.1 DESCRIÇÃO DA OPERAÇÃO

Quando a camada de enlace possui dados ou mensagens para enviar o sinal de interface RTS é acionado. O modem ao receber este sinal desencadeia as seguintes operações:

1. Acionamento do estabelecimento de automático de enlace ALE (*Automatic Link Establishement*) que possui um modem FSK especifico com taxa de símbolo $R_s = 125 bauds$ capaz de procurar a melhor freqüência para estabelecer o enlace.

2. Após estabelecido o enlace, o modem serial aciona o sinal DCD e envia o sinal CTS para a camada de enlace que comeca a enviar os bits para o buffer do modem.

3. Assim que os dados começam a entrar no buffer, o modem começa a preencher a matriz de entrelaçamento ao mesmo tempo em que envia o preâmbulo para treinamento inicial do equalizador do modem remoto. O tempo para encher a matriz determina o tamanho do preâmbulo e varia de acordo com taxa de bits nominal do modem de modo a o manter o retardo constante.

4. Após preencher a matriz, o modem começa a enviar os dados sempre alternando com pequenas provas de símbolos conhecidos para treinamento do equalizador remoto.

5. A matriz continua sendo continuamente preenchida para formar novas seqüências de dados e treinamento até que seja retirado o sinal RTS.

6. Quando retirado o sinal RTS, a seqüência de fim de mensagem EOM (*End Of Message*) composta por 4 bytes, $4B65A5B2$, é enviada e são completados com zeros os registradores de deslocamento do codificador e a matriz do entrelaçador.

7. Após isto são retirados os sinais CTS e DCD.

8. O enlace é liberado.

A FIG. 5.9 resume os dois tipos de entrelaçamento disponíveis no modem. No caso do entrelaçamento escolhido ser o curto, há a opção sem entrelaçamento algum.

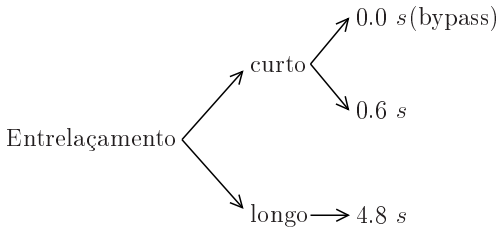


FIG. 5.9: Opções de entrelaçamento do modem

De acordo com o entrelaçamento escolhido e a taxa nominal de bits o tamanho da matriz muda de acordo com a TAB. 5.2.

TAB. 5.2: Dimensões da matrix de entrelaçamento e passo de entrada e saída

	Entrelaçamento longo		Entrelaçamento curto		passo	
$R_b(bps)$	linhas	colunas	linhas	colunas	entrada	saida
4800	...	...	...	...	...	...
2400	40	576	40	72	9	17
1200	40	288	40	36	9	17
600	40	144	40	18	9	17
300	40	144	40	18	9	17
150	40	144	40	18	9	17
75	20	36	10	9	7	7

O preâmbulo é formado por 3 ou 24 repetições de um segmento de duração de 200 ms conforme o entrelaçamento seja curto ou longo. Este segmento é composto por um seqüência de 15 símbolos do canal (tribits) dos quais 10 são conhecidos e 5 não são conhecidos. Esta seqüência é $\{0, 1, 3, 0, 1, 3, 1, 2, 0, D_1, D_2, C_1, C_2, C_3, 0\}$. Os valores de

TAB. 5.3: Modos de operação do modem

	Entrelaçamento longo			Entrelaçamento curto		
	$4,8\ s$			$0,6\ sou 0,0\ s$		
$R_b(bps)$	D_1	D_2	modo	D_1	D_2	modo
4800	01	10	6	11	10	14
2400 (voz)	01	11	7	11	11	15
2400 (dados)	00	00	0	10	00	8
1200	00	01	1	10	01	9
600	00	10	2	10	10	10
300	00	11	3	10	11	11
150	01	00	4	11	00	12
75	01	01	5	11	01	13

D_1 e D_2 são sugeridos pela camada de enlace do modem transmissor ao modem receptor e os valores possíveis estão representados na TAB. 5.3.

Os valores de C_1 , C_2 e C_3 são usados como contadores da ordem da seqüência que está sendo gerada. Desta forma, como a contagem é regressiva, a primeira seqüência será 23 no caso de entrelaçamento longo e 2 no caso de curto. O mapeamento do contador nos símbolos é feito expressando-se o número em um binário com seis digitos, separando os bits dois a dois e acrescentado o 1 na frente para obter os três tribits a serem enviados onde C_1 contém os dois bits mais significativos do contador. Por exemplo $23 \rightarrow (010111) \rightarrow (101,101,111) \rightarrow (C_1 = 5, C_2 = 5, C_3 = 7)$.

Depois de enviado o preâmbulo, o modem fica alternando entre envio de símbolos de dados e novos símbolos de treinamento. O perído desta alternância varia de acordo com a taxa de bits conforme mostrado na TAB. 5.4.

TAB. 5.4: Janelas de envio de símbolos de dados e prova

$R_b(bps)$	Dados (T_s)	Prova (T_s)
4800	32	16
2400	32	16
1200	20	20
600	20	20
300	20	20
150	20	20
75	∞	$\cdots$

Como o modem trabalha com uma taxa de símbolos fixa, à medida que a taxa nominal de bits é reduzida os bits não usados são utilizados para melhorar a taxa de erro dos bits de informação bem como a quantidade bits codificados colocada em cada símbolo do

canal é alterada conforme resumido na TAB. 5.5

TAB. 5.5: Taxa de código do modem serial

$R_b(bps)$	$R_{c,eff}$	Método	$R_{b,out}$	bits por símbolo
4800	$\cdots$	$\cdots$	4800	$3^{(MGD)}$
2400	1/2	1/2	4800	$3^{(MGD)}$
1200	1/2	1/2	2400	$2^{(MGD)}$
600	1/2	1/2	1200	1
300	1/4	1/2 + $\times$2	1200	1
150	1/8	1/2 + $\times$4	1200	1
75	1/2	1/2	150	$2^{(MGD)}$

Como pode-se observar na última coluna da TAB. 5.5 os símbolos formados com dois ou três bits da matriz de entrelaçamento são decodificados pelo código Gray modificado de forma a evitar que o erro entre dois símbolos adjacentes produza mais de um bit errado. O termo decodificado é empregado pois as palavras-código associadas a símbolos adjacentes na constelação do modem possuem distância de Hamming unitária. Estas palavras-código podem ser formadas por dois ou três bits. A TAB. 5.6 mostra o mapeamento MGD para três bits e a TAB. 5.7 para dois bits.

TAB. 5.6: Decodificação modificada Gray para 2400 *bps* e 4800 *bps*

bits de entrada	valor decodificado
000	000
001	001
010	011
011	010
100	111
101	110
110	100
111	101

TAB. 5.7: Decodificação modificada Gray para 75 *bps* e 1200 *bps*

bits de entrada	valor decodificado
00	00
01	01
10	11
11	10

O bloco FORM executa a formação dos símbolos de treinamento e dados a serem enviadas para o modulador. A FIG. 5.10 mostra os pontos da constelação $8PSK$ que são enviados quando símbolo formado possui 1, 2 ou 3 bits. No caso particular dos símbolos de treinamento, os símbolos gerados pelos blocos PREAMBLE e PROBE são ainda mapeados segundo a TAB. 5.8 e repetidos respectivamente 4 e 2 vezes.

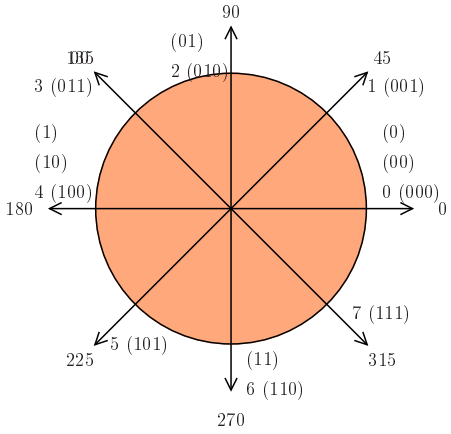


FIG. 5.10: Diagrama da constelação do modem

TAB. 5.8: Mapeamento para formação de tribits a partir de símbolos do canal

símbolo do canal	tribits associados
000	0000 0000
001	0404 0404
010	0044 0044
011	0440 0440
100	0000 4444
101	0404 4040
110	0044 4400
111	0440 4004

A FIG. 5.11 ilustra o modo 8 do modem com 30 seções de dados e prova formadas a partir da matriz de entrelaçamento.

PREAMBLE	DATA 1	PROBE 1	... ...	DATA 29	PROBE 29	DATA 30	PROBE 30
1440	32	16×0		32	$16 \times D_1$	32	$16 \times D_2$

FIG. 5.11: Ilustração da seqüência de símbolos do modo 8 do modem serial

Os símbolos de prova são sempre zero com exeção do penúltimo e último que são respectivamente D_1 e D_2. A finalidade é marcar o fim da matriz de entrelaçamento e, possivelmente, o inicio de uma nova.

As fases de operação do modem são ilustradasseguir.

- **Fase de sincronização do preâmbulo**: as chaves são colocadas nas posições S_1 = DATA, S_2 = PREAMBLE e S_3 = TRAIN SYNC. O preâmbulo é enviado e a matriz de entrelaçamento preeenchida. A FIG. 5.12 ilustra esta fase.

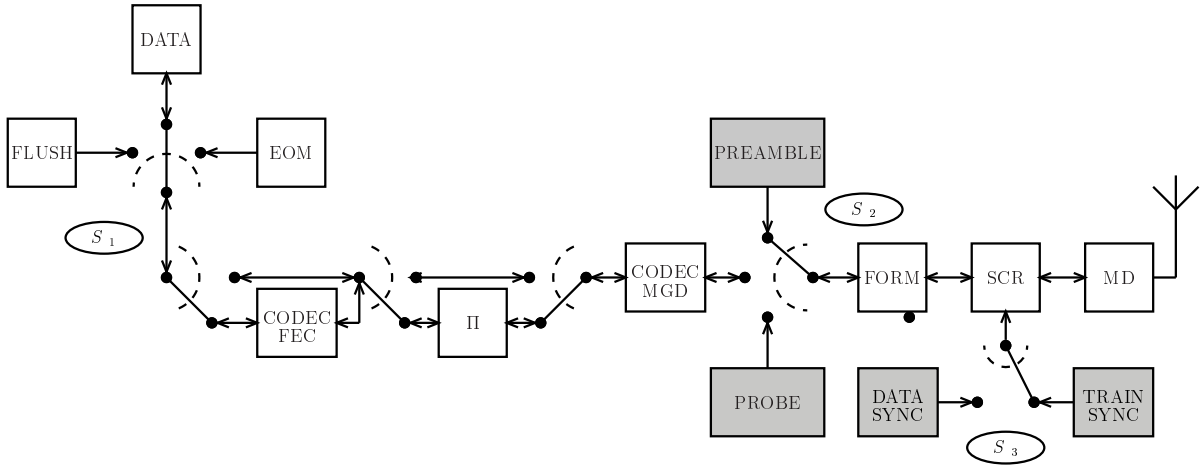


FIG. 5.12: Carregamento do entrelaçador e emissão do preâmbulo

- **Fase de dados**: a chave é colocada na posição S_1 = DATA e as chaves S_2 e S_3 ficam variando entre CODEC MGD / PROBE e DATA SYNC / TRAIN SYNC respectivamente conforme a janela de transmissão seja de símbolos de dados ou de prova. As FIG. 5.13 e FIG. 5.14 ilustra esta fase.

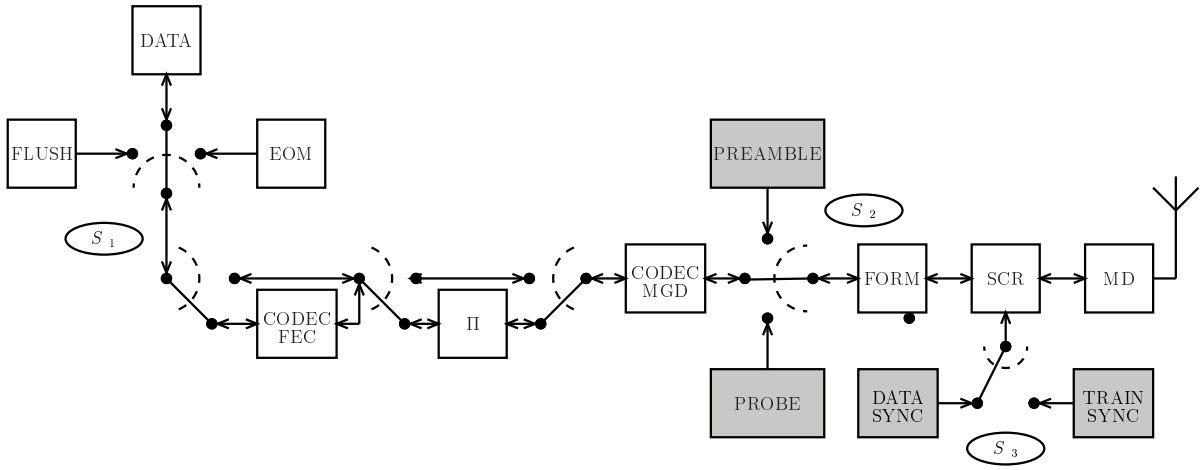


FIG. 5.13: Transmissão dos dados

- **Fase de fim da mensagem**: a chave S_1 na posição EOM para transmissão da seqüência $4B65A5B2$ de 32 bits. A FIG. 5.15 ilustra esta fase.

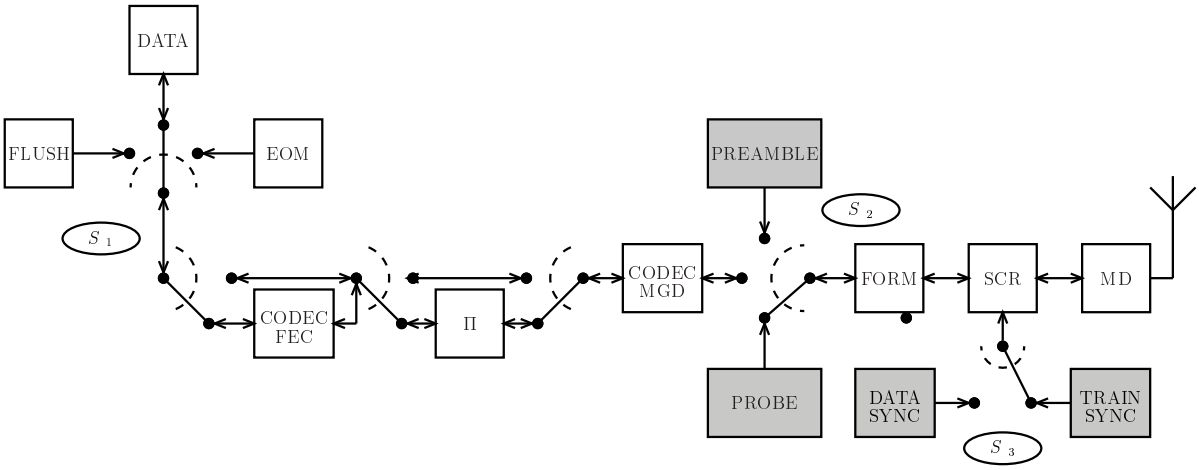


FIG. 5.14: Transmissão da seqüência de treinamento

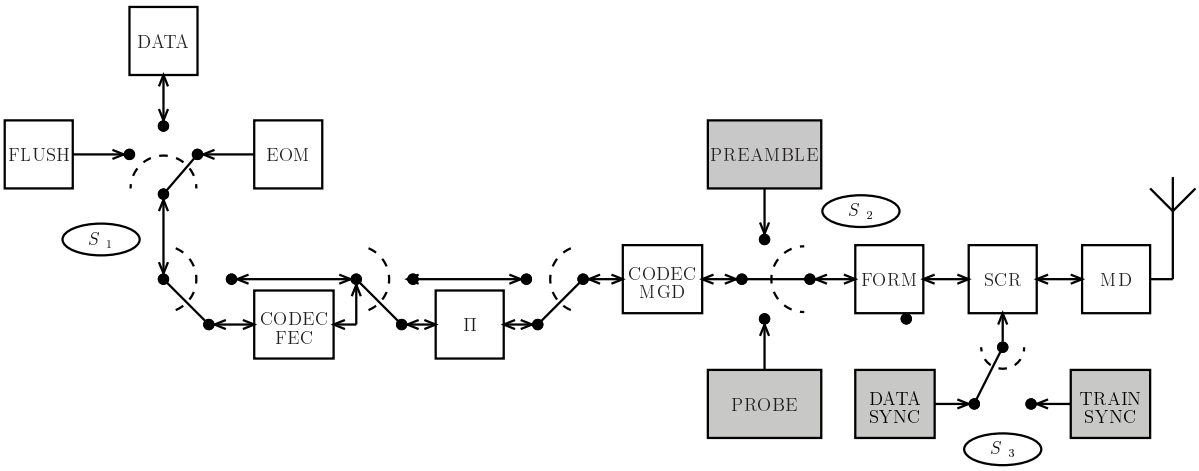


FIG. 5.15: Transmissão da seqüência EOM

- **Fase de esvaziamento de codificador e entrelaçador**: a chave S_1 na posição $FLUSH$ para a emissão de um número suficiente de bits para zerar o estado do codificador e completar a matriz de entrelaçamento. A FIG. 5.16 ilustra esta fase.

5.2.2 EQUALIZADOR

O equalizador é um dispositivo capaz de reduzir a interferência entre símbolos ISI (*Inter Symbol Interference*) causada tanto pela propagação multipercurso como pela restrição de faixa do canal de transmissão.

O MIL-STD-188-110A não padroniza nenhum tipo de equalizador mas apenas indica que deve ser adaptativo e com seqüência de treinamento. Para este trabalho utilizou-se a equalização do tipo não linear com realimentação da decisão DFE (*Decision Feedback Equalization*). A principal vantagem do uso do DFE é o fato desta técnica oferecer o

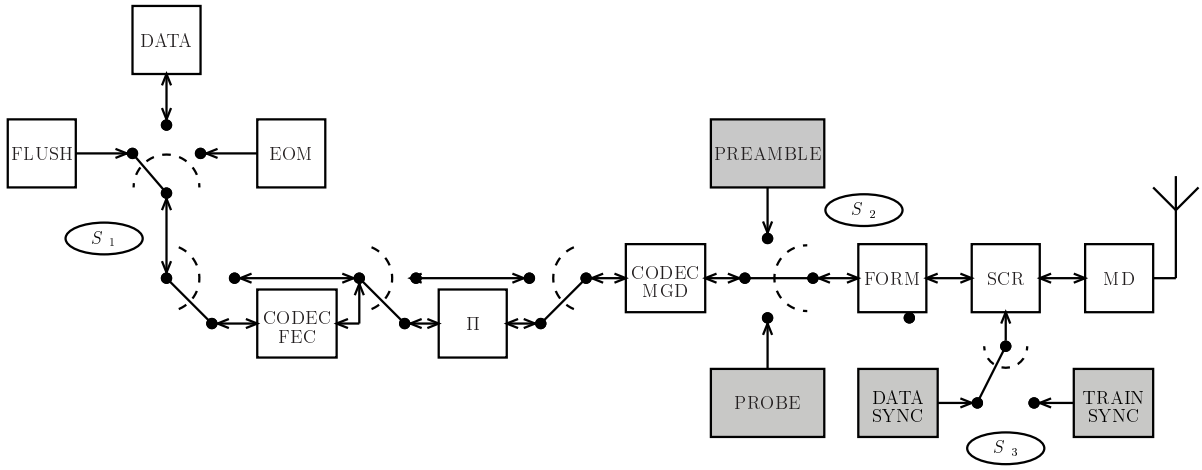


FIG. 5.16: Transmissão do FLUSH do codificador e entrelaçador

cancelamento da ISI com reduzida amplificação do ruído. Na implementação utilizou-se apenas uma única amostra por símbolo BSE (*Baud Spaced Equalizer*) por questão de complexidade das equações.

Boroujeny (FARHANG-BOROUJENY, 1996) entre outros mostraram que a equalização via identificação de canal CEQCID (*Channel EQualization via Channel IDentification*) é mais robusta em canais com variações rápidas como o canal HF do que o DFE puro, ou seja, o DFE usando apenas o acompanhamento adaptativo dos coeficientes.

A identificação de canal foi implementada usando um estimador de canal com algoritmo VS-LMS com passo adaptativo (*Variable Step-Least Mean Square*) (HARRIS, 1986). Após a estimação da resposta impulsiva do canal, a equação de Wiener-Hopf é usada para calcular os coeficientes do filtro avante. O retardo de decisão foi ajustado para $\delta = N_f - 1$, ou seja, o número de coeficientes do filtro avante menos um.

A FIG. 5.17 ilustra o equalizador implementado.

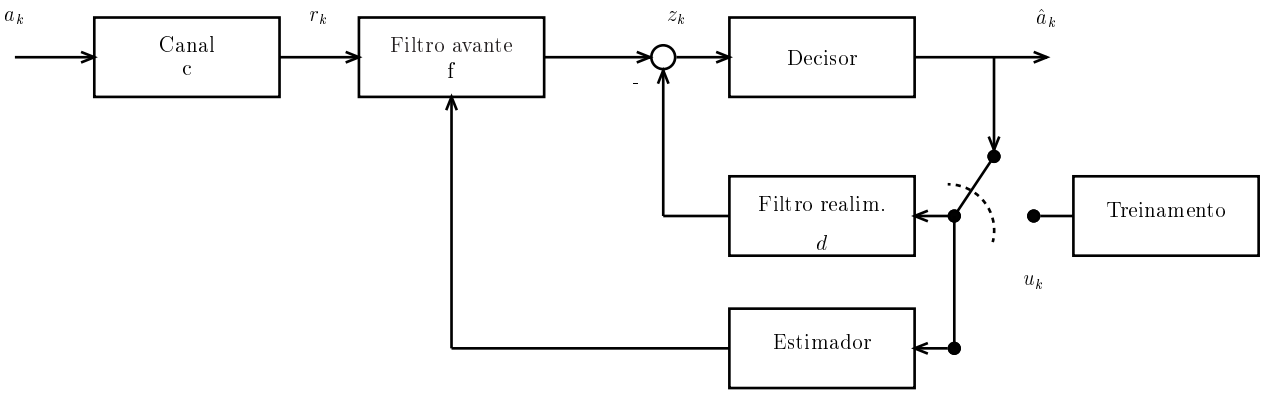


FIG. 5.17: Diagrama do equalizador DFE com estimador de canal LMS

6 USO DE CÓDIGOS CONCATENADOS NA CAMADA DE ENLACE

Conforme visto no Capítulo 4, a camada de enlace é a última camada responsável pela comunicação ponto a ponto segundo o modelo OSI. Por isto ela é também a última camada onde o controle de erros deve ser realizado em redes com camadas físicas pouco confiáveis, ou seja, com altas taxas de erros.

Devido ao fato da comunicação na camada de enlace ser ponto a ponto, o controle de erro nesta camada pode ser otimizado de acordo com características da camada física entre os pontos de origem e destino (ZORZI, 1999) (CHOCKALINGAM, 1999).

Neste capítulo serão abordados os mecanismos de produção e propagação de erros nas camadas física e de enlace e propostos dois esquemas alternativos ao SR-ARQ do HFDLP. Por utilizarem um código para FEC, estes esquemas são denominados de ARQ híbridos ou HARQ (*Hybrid Automatic Retransmission reQuest*) e podem ser tanto do Tipo I como Tipo II, conforme pode ser visto no APÊNDICE 4. Um deles utiliza o código concatenado CCSDS visto no Capítulo 2 e o outro o código concatenado Turbo visto no Capítulo 3.

6.1 ANÁLISE DA PROPAGAÇÃO DE ERRO

Esta seção analisa os mecanismos de produção do erro na camada física e propagação deste erro para a camada de enlace.

De modo geral, um canal de comunicação sem fio causa variações de amplitude, fase e ângulo de chegada no sinal recebido. O diagrama da FIG. 6.1 apresenta uma visão geral das manisfestações da variabilidade associada ao desvanecimento em um canal de comunicação. A variabilidade de um canal rádio é devida a dois tipos distintos de efeitos de desvanecimento: o desvanecimento em larga escala e o em pequena escala. A variabilidade em larga escala está relacionada aos mecanismos de propagação referentes a distância e sombreamento. Já a variabilidade em pequena escala refere-se a mudanças bruscas na amplitude e fase do sinal para o caso de nenhuma variação ou pequenas variações (da ordem do comprimento de onda do sinal) na distância receptor-transmissor. Esta variabilidade é ocasionada por dois fatores: dispersão do sinal transmitido (multipercurso) e variação temporal do canal.

O modelo de Watterson (WATTERSON, 1970) para o canal HF pode ser descrito

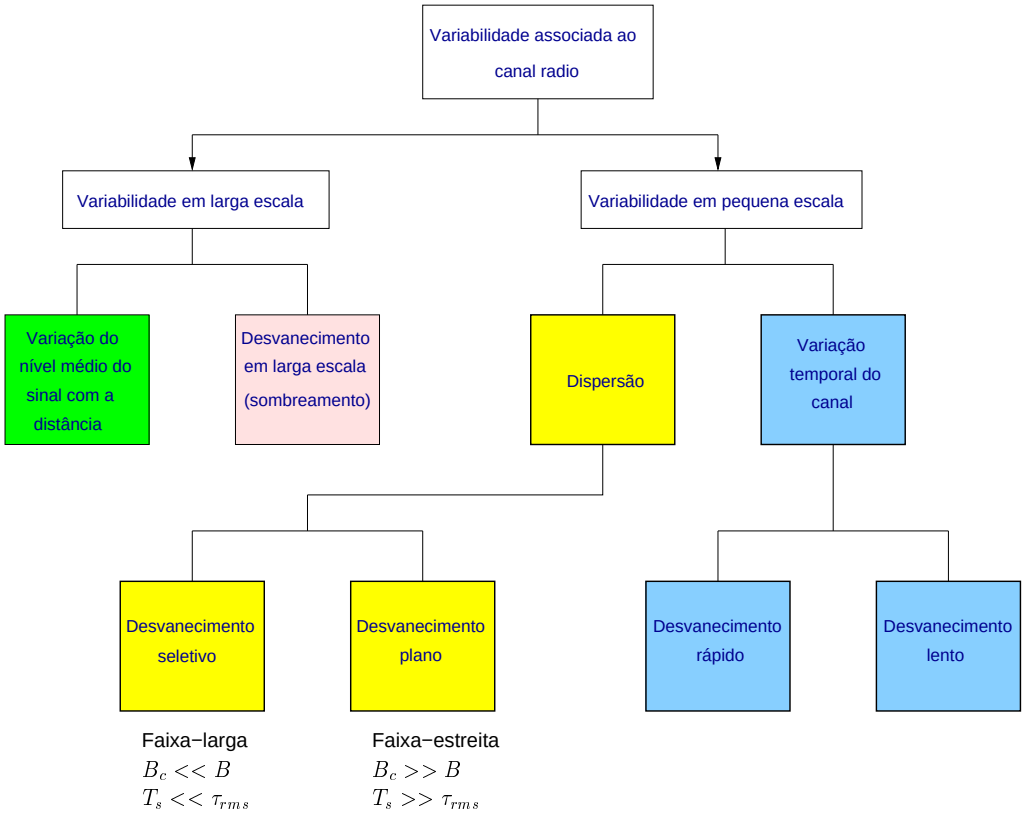


FIG. 6.1: Variabilidade do sinal recebido

resumidamente como um modelo multipercurso onde cada raio sofre um desvanecimento Rayleigh. Neste sistema, o multipercurso é o principal responsável pela interferência entre símbolos ISI embora a restrição de faixa também tenha uma contribuição.

Caso o equalizador descrito no Capítulo 5 fosse capaz de eliminar a interferência entre símbolos, o canal HF poderia ser modelado por apenas um percurso com desvanecimento Rayleigh e ruído branco aditivo. Deste modo, a variabilidade em pequena escala do canal seria devido apenas a variações temporais do canal. Neste caso a técnica de melhoria aplicável seria a diversidade temporal (entrelaçamento) e o uso de códigos corretores de erro FEC.

O modem de tom serial usado neste trabalho e definido no MIL-STD-188-110A já incorpora as duas principais técnicas de combate à variação temporal do canal: entrelaçamento e correção de erro. Conforme descrito no Capítulo 5, há neste modem várias opções de entrelaçamento com tamanhos variáveis de acordo com a taxa nominal de bit do modem. O objetivo deste arranjo é mater o retardo fixo em $0,6\ s$ e $4,8\ s$ para qualquer modo de operação do modem. Comparando os valores de retardo dos entrelaçadores com os tempos de coerência T_c dos canais bom ($0,5\ s$), moderado ($2\ s$) e ruim ($10\ s$), vê-se que, pelo menos teoricamente, ambos os retardos são insuficientes para descorrelacionar

o efeito da variação temporal no canal ruim. Conforme mostrado a seguir, os surtos de erros são bem menores do que aquelas pervistas pelo tempo de coerência do canal.

A FIG. 6.2 mostra as curvas de probabilidade de erro de bit nos canais bom, moderado e ruim definidos pelo CCIR (CCIR, 1992).

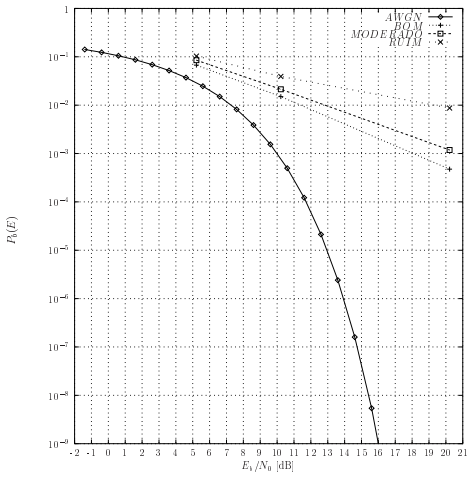


FIG. 6.2: Taxa de erro de bit para o canal CCIR

Os padrões de erro dos canais bom, moderado e ruim para E_b/N_0 podem ser vistos na FIG. 6.3.

Um surto de erro começa e termina em um bit errado e não pode ter mais de min_{EFI} bits corretos dentro do surto. Em todas as figuras que seguem com distribuições de erro, o valor do parâmetro min_{EFI} foi 4. A escolha deste valor é justificável pelo fato de que em 5 bits corretos haverá pelo menos 1 símbolo de canal correto ($8PSK$).

As FIG. 6.4, FIG. 6.5 e FIG. 6.6 apresentam a freqüência relativa dos surtos de erros para, respectivamente, os canais $CCIR$ bom, moderado e ruim com relação E_s/N_0 igual a 10 dB, 15 dB e 25 dB.

As FIG. 6.7, FIG. 6.8 e FIG. 6.9 apresentam a freqüência relativa dos intervalos livres de erros para, respectivamente, as relações E_s/N_0 igual a 10 dB, 15 dB e 25 dB nos canais $CCIR$ bom, moderado e ruim.

Analisando-se as FIG. 6.4 até FIG. 6.9, observa-se que a maior parte dos surtos de erros tem comprimento variando de 5 a 10 bits e que o tamanho dos surtos diminui com o aumento de E_s/N_0. Naturalmente, os comprimentos dos intervalos livres de erros aumentam com o aumento de E_s/N_0 e também com a diminuição da velocidade com que a resposta impulsiva do canal varia dada por B_d. Nesta última situação a explicação é que o estimador que trabalha em conjunto com o equalizador DFE consegue acompanhar mais facilmente as variações do canal.

6.2 CÓDIGO TURBO

Nesta seção é descrito o uso de códigos concatenados Turbo (BERROU, 1993) de taxa $R_c = 1/2$ na camada de enlace em um esquema capaz de obter melhoria na vazão do protocolo HFDLP.

Como a capacidade do canal varia de acordo com a taxa de código (HAYKIN, 2000), todas os resultados obtidos neste trabalho foram referentes à taxa de código $R_c = 1/2$. Para obter esta taxa, usou-se o código Turbo de taxa $R_c = 1/3$ puncionado para a taxa $R_c = 1/2$ (ROWITCH, 2000) e desligou-se o código convolucional da camada de enlace juntamente com o entrelaçamento. Este desligamento consiste apenas na operação do modem no modo 14 conforme pode ser visto na TAB. 5.3.

O desempenho dos códigos Turbo é tanto melhor quanto maior for o tamanho do entrelaçamento conforme pode ser visto pela EQ 3.16. Os resultados que Berrou et al (BERROU, 1993) obteve foram com um entrelaçador de tamanho 65534 bits. Como o HFDLP estabelece o tamanho de quadro máximo em 1023 *bytes* (NCS, 1994) (DoD, 1994), optou-se por respeitar este máximo e usá-lo nas simulações. O trabalho em (CHI, 2000) propõe para pacotes pequenos (cerca de 100 *bits*) o uso da técnica denominada de *tail biting* que consiste na continuação da treliça com os bits de informação do pacote seguinte. Isto garante um ganho de vazão em torno de 8 %. No caso do HFDLP, o ganho obtido por esta técnica seria desprezível em virtude dos quadros. Desta forma optou-se pelo uso do preenchimento com zeros dos registradores dos codificadores RSC, conforme ilustrado na FIG. 3.1.

Os quadros de controle e dados foram codificados de modo diferentes em virtude do tamanho dos quadros de controle ser cerca de uma ordem de grandeza menor que os de dados, conforme resumido na TAB. 9.5. Enquanto que os quadros de dados foram codificados com o código Turbo, os quadros de controle foram codificados com o código CCSDS modificado, onde o código $RS\ (255, 223)$ foi encurtado de acordo com o tamanho (tipo) do quadro. A camada de enlace do receptor usa o CRC do quadro para determinar o tipo de quadro que foi recebido e assim continuar o processamento de seus campos.

O uso de códigos Turbo na camada de enlace supõe que parte da LLR relativa a cada bit foi perdida no demodulador. Apesar desta perda, foi visto no Capítulo 3 que a redução do ganho de código é de apenas $1,5\ dB$ em canal AWGN e modulação BPSK. Esta pequena redução indica que a decodificação iterativa foi capaz de compensar a perda produzida pela decisão abrupta do demodulador.

Para que os bits codificados possam entrar no decodificador Turbo, seus valores

$\{0,1\}$ devem ser mapeados em GF(2) com os elementos $\{+1,-1\}$ de forma que $0 \rightarrow +1$ e $1 \rightarrow -1$ onde o elemento $+1$ é elemento nulo para a adição $\oplus$. A LAPP da soma de dois valores m_1 e m_2 estatísticamentes independentes pertencentes a GF(2) é dada pela expressão 6.1 (HAGENAUER, 1996).

$$\begin{aligned} L(m_1 \oplus m_2) &= \log \frac{1 + e^{L(m_1)} e^{L(m_2)}}{e^{L(m_1)} + e^{L(m_2)}} && (6.1) \\ &\approx \operatorname{sign}(L(m_1)) \cdot \operatorname{sign}(L(m_2)) && (6.2) \\ &\cdot \quad \min(|L(m_1)|, |L(m_2)|) \end{aligned}$$

A soma de dois LAPP é indicada pelo símbolo $\boxplus$ e é definida pela expressão 6.3.

$$L(m_1) \boxplus L(m_2) \triangleq L(m_1 \oplus m_2) \tag{6.3}$$

A seguir será demonstrado em um exemplo simples o funcionamento da decodificação iterativa com os valores abruptos aos invés dos suaves.

Exemplo 6.1 (Código (3,2,2) checador de paridade simples) *A FIG. 6.10 ilustra este exemplo. A codificação de quatro bits de informação através de dois códigos (3,2,2) de checagem de paridade simples com elementos $\{+1,-1\}$ em GF(2) é mostrada na FIG. 6.10(d). Os valores abruptos recebidos são mostrados na FIG. 6.10(b) e nenhuma informação a priori está disponível. Começando pela decodificação horizontal: a informação do bit u_{11} é recebida duas vezes: diretamente por meio de m_{11} e indiretamente por meio de $m_{12} \oplus p_1^-$. Como m_{12} e p_1^- são transmitidos estatisticamente independentes, a LAPP de sua soma é dada pela seguinte expressão:*

$$L(m_{12} \oplus p_1^-) = L(m_{12}) \boxplus L(p_1^-) = 1 \boxplus 1 \approx 1$$

Esta informação indireta sobre m_{11} é chamada de valor extrínseco e é armazenada na FIG. 6.10(e). Os demais valores são calculados de modo idêntico. Quando a tabela com os valores extrínsecos horizontais estiver completa, dar-se-á inicio à decodificação vertical usando os valores L_e^- como valores a priori para a decodificação vertical. Isto significa que depois da decodificação vertical de m_{11} obtém-se as seguintes valores de LAPP para m_{11}:

- *o valor recebido direto $+1$;*

- *o valor a priori* L_e^- *obtido a partir da decodificação horizontal* $+1$*;*
- *o valor extrínseco vertical* $L_e^{\downarrow}$ *usando toda a informação disponível sobre* $m_{21} \oplus p_1^{\downarrow}$*, ou seja,* $(1 + (-1)) \boxplus 1 \approx 0$

De modo análogo, os outros valores da decodificação vertical são colocados na FIG. 6.10(c). Finalizando as iterações, calcularia-se as saídas suaves a partir da equação abaixo conforme mostrado na FIG. 6.10(f).

$$L(\hat{m}) = L_c \cdot y + L_e^- + L_e^{\downarrow}$$

Neste ponto já dispoe-se de boa confiabilidade $|L(\hat{m})|$ *para a decodificação.*

Observou-se que escolha valor de L_c a ser usado pelo algoritmo de decodificação Turbo na camada de enlace é um ponto crítico no que diz respeito ao desempenho do algoritmo principalmente para SNR altos na camada física. A FIG. 6.11 mostra o gráfico dos valores de L_c para os canais AWGN com modulação $8PSK$ e o BSC obtido a partir da aplicação de um demodulador na canal AWGN com modulação $8PSK$.

Heuristicamente, observou-se que o valor de L_c para o algoritmo de decodificação convergir para a palavra-código correta quando $E_b/N_0 = 25\ dB$ é cerca de 10 vezes menor que o valor L_c para o canal BSC.

6.3 CÓDIGO CCSDS

O uso do código RS na camada de enlace é justificado pelo fato da existência na camada física de um decodificador de Viterbi para um código convolucional de comprimento restritivo $K = 7$ conforme descrito no Capítulo 5. É fato bem conhecido da literatura que o decodificador de Viterbi produzir erros em surtos. Este é o principal motivo que originou a ideia de concatenar um código com capacidade de corrigir erros em surto com o RS. Como o tamanho do símbolo RS é de 8 *bits*, a maioria dos erros do decodificador de Viterbi estará confinada a apenas um símbolo RS.

A implementação do código concatenado CCSDS na camada de enlace foi feita usando-se o código convolucional já disponível na camada física que por coincidência corresponde ao definido no padrão. Desta forma, foi necessário apenas implementar o codificador RS na camada de enlace seguido de um entrelaçador conforme ilustrado na FIG. 6.13. O modo de operação correspondente do modem é o de número 8.

6.4 RESULTADOS

As FIG. 6.14, FIG. 6.15 e FIG. 6.16 mostram os resultados da vazão dos protocolos HFDLP+CCSDS, HFDLP+TURBO e HFDLP+CRC nos canais CCIR bom, moderado e ruim respectivamente. Os protocolos HFDLP+CCSDS e HFDLP+CRC foram simulados com tamanho de quadro fixado em 214 *bytes* e 8 quadros por série. No protocolo HFDLP+TURBO foi mantido o tamanho das séries mas aumentado o tamanho dos quadros de dados para 1000 *bytes* em virtude do melhor desempenho dos códigos Turbo para quadros maiores.

No decodificador Turbo foram usados os valores de L_c determinados heuristicamente para o canal AWGN. As simulações começam em $E_b/N_0 = 5\ dB$ em virtude das curvas de probabilidade de erro dos códigos Turbo com decisão suave em canal AWGN e modulação BPSK indicarem a redução da probabilidade de erro a partir de $E_b/N_0 = 4\ dB$.

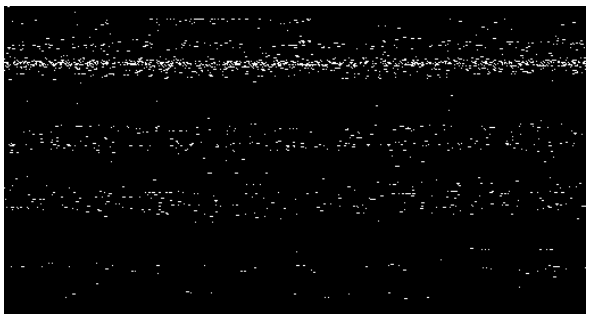
A comparação entre as dos protocolos HFDLP+CCSDS e o HFDLP+CRC em todas figuras mostram que até $E_s/N_0 = 10\ dB$ o esquema de FEC com o CCSDS é melhor que o ARQ puro com CRC. Em canal moderado e $E_s/N_0 = 5\ dB$, a vazão do HFDLP+CCSDS é cerca de 2,5 vezes maior.Para energias maiores, o canal melhora e o FEC torna-se dispensável. A vazão nesta região obtida pelo protocolo HFDLP+CRC poderia ser melhorada aumentando-se o tamanho de quadros e séries.

As figuras mostram que a máxima vazão do protocolo HFDLP+TURBO é de cerca de $0,44$. Como o código Turbo usado possui taxa $R_c = 1/2$, a redundância acrescentada pelo protocolo, considerando desprezível o tempo de processamento dos quadros, é de aproximadamente 12 % quando os quadros usados têm tamanho de 1000 *bytes* e as séries são formadas por 8 quadros.

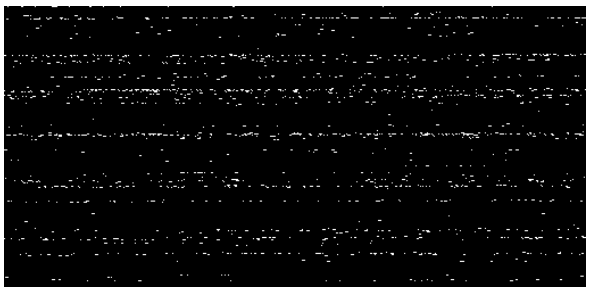
A FIG. 6.15 indica um aumento de 4 vezes na vazão do protocolo HFDLP+TURBO em relação ao protocolo HFDLP+CRC para $E_s/N_0 = 10\ dB$. Este resultado foi obtido usando o parâmetro L_c 100 vezes menor que aquele que seria usado em canal AWGN. Uma explicação para este comportamento em termos da decodificação iterativa seria que a dimunição do peso da informação recebida do canal causaria um aumento no peso da informação extrínseca trocada entre os decodificadores.

Com relação ao desempenho dos códigos Turbo no canal HF onde o desvanecimento símbolo a símbolo é correlacionado as simulações mostraram que o uso ou não de entrelaçamento no canal não altera o desempenho em termo de vazão do protocolo HFDLP+TURBO. Os entrelaçadores usados foram os disponíveis no modem de tom serial, ou seja, com retardo de $0,6\ s$ e $4,8\ s$.

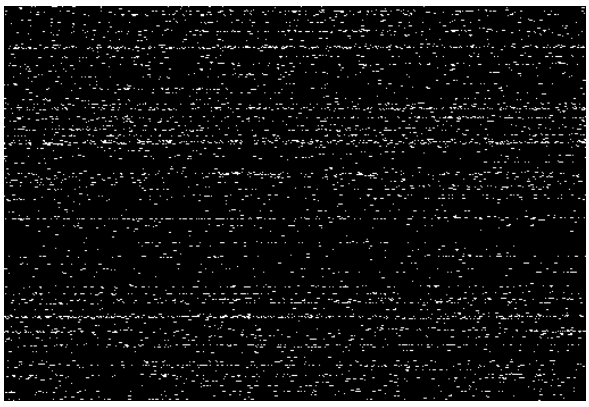
(a) Arquivo original



(b) Canal bom



(c) Canal moderado



(d) Canal ruim

FIG. 6.3: Transmissão de um arquivo de 5000 *bytes* através dos canais bom, moderado e ruim com $E_s/N_0 = 15\ dB$

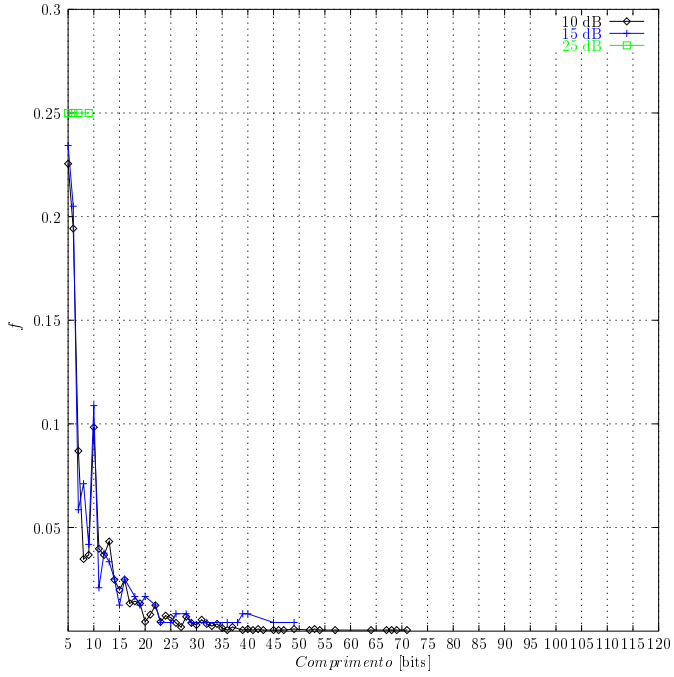


FIG. 6.4: Distribuição de surtos de erros em canal bom.

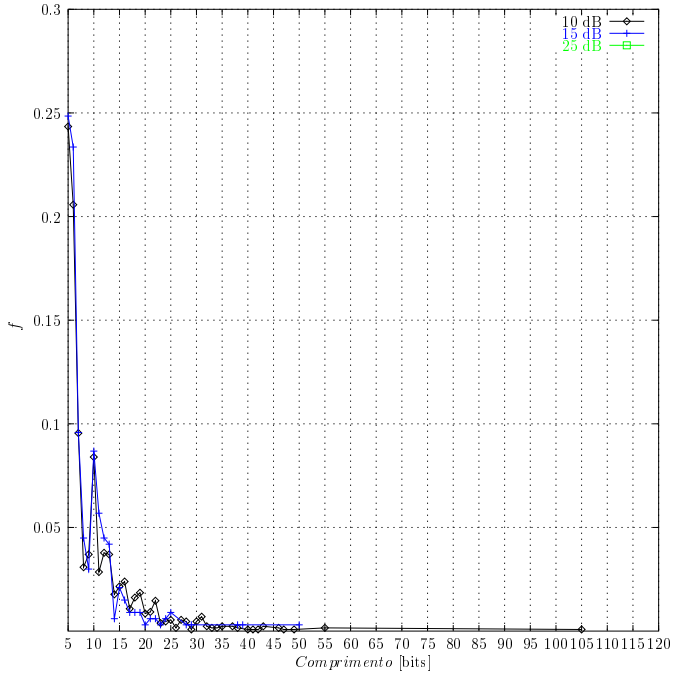


FIG. 6.5: Distribuição de surtos de erros em canal moderado.

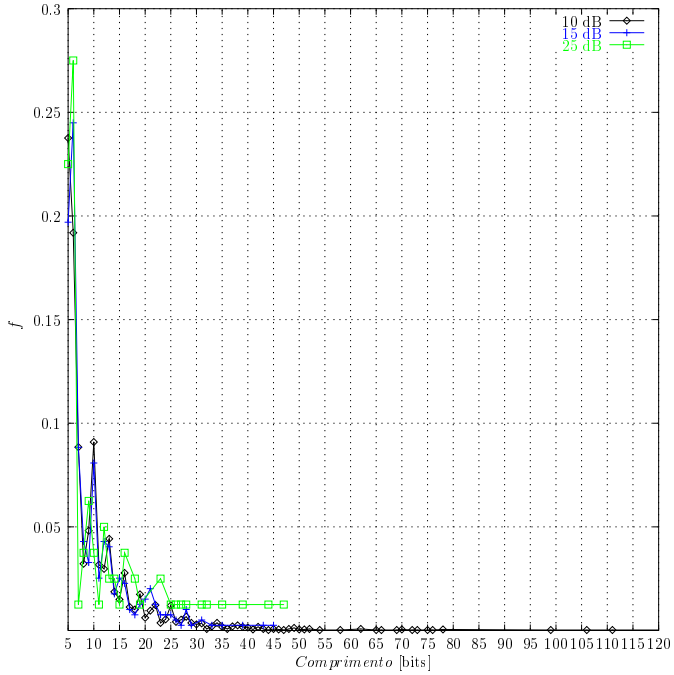


FIG. 6.6: Distribuição de surtos de erros em canal ruim.

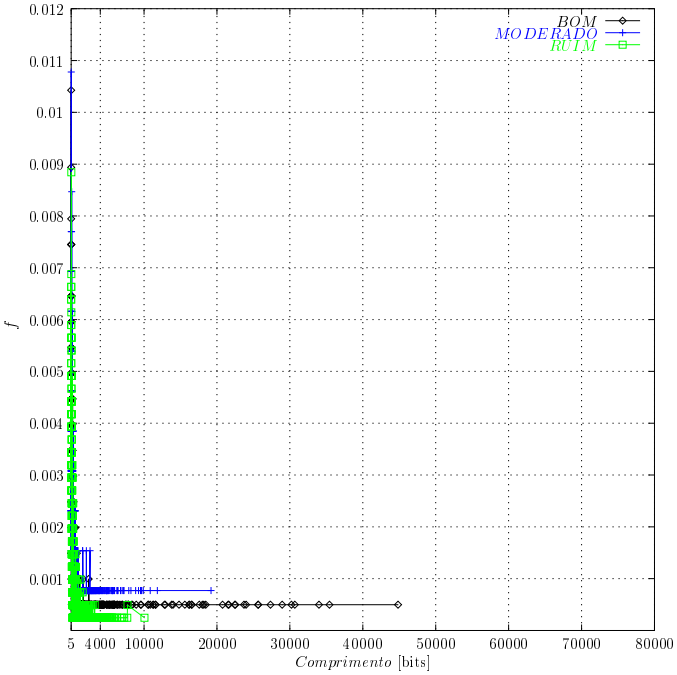


FIG. 6.7: Distribuição de intervalos sem erros com $E_s/N_0 = 10\ dB$.

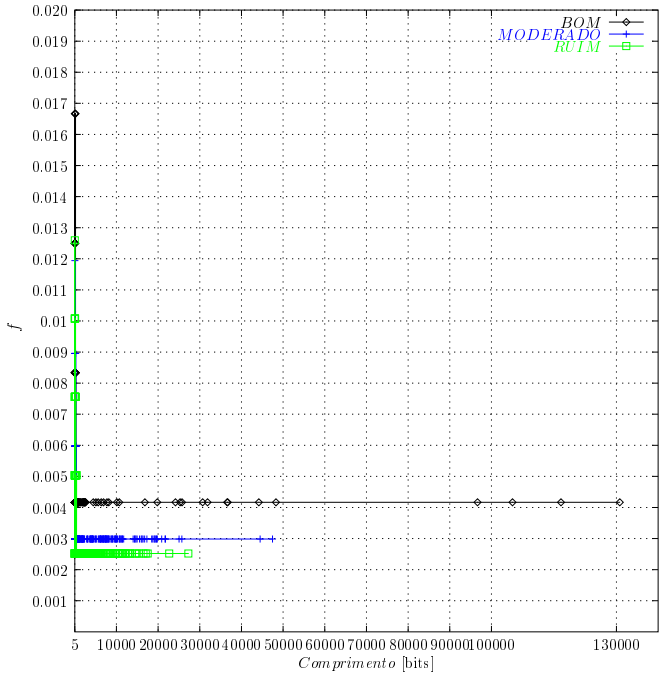


FIG. 6.8: Distribuição de intervalos sem erros com $E_s/N_0 = 15\ dB$.

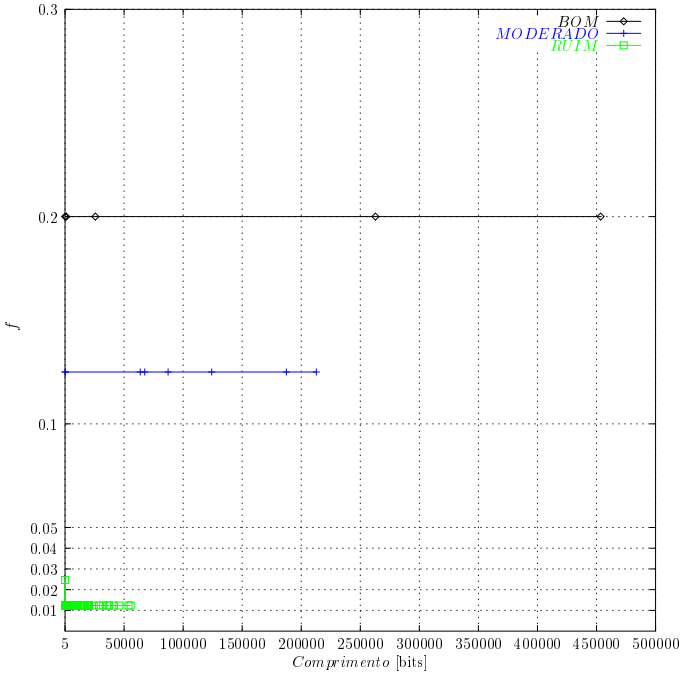


FIG. 6.9: Distribuição de intervalos sem erros com $E_s/N_0 = 25\ dB$.

u_{11}	u_{12}	p_1^-
u_{21}	u_{22}	p_2^-
$p_1^\mid$	$p_2^\mid$	

(a) Código bidimensional

$+1$	$+1$	$+1$
$+1$	$+1$	-1
$+1$	-1	

(b) Valores recebidos $L_c \cdot y$

0	0
$+1$	-1

(c) Informação extrínseca $L_e^\mid$ após a primeira decodificação vertical

$+1$	$+1$	$+1$
$+1$	-1	-1
$+1$	-1	

(d) Valores codificados

$+1$	$+1$
-1	-1

(e) Informação extrínseca L_e^- após a primeira decodificação horizontal

$+2$	$+2$
$+1$	-1

(f) Saída suave após a primeira decodificação horizontal e vertical

FIG. 6.10: Exemplo de decodificação iterativa de um código de taxa 1/2 usando dois códigos de taxa 2/3.

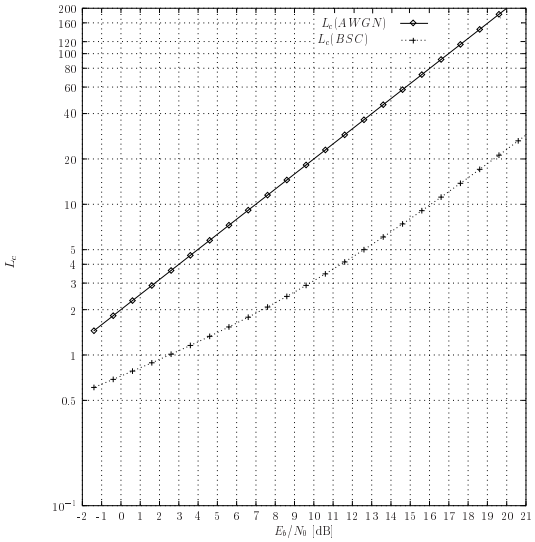


FIG. 6.11: L_c para canal AWGN e BSC

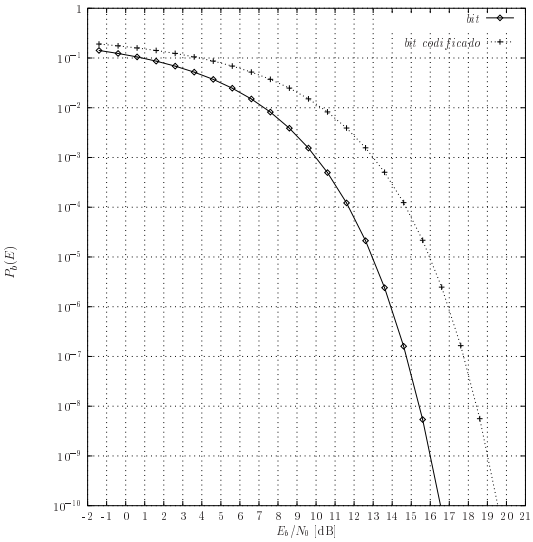


FIG. 6.12: Probabilidade de erro de bit de informação e de código de taxa $R_c = 1/2$ para modulação $8PSK$

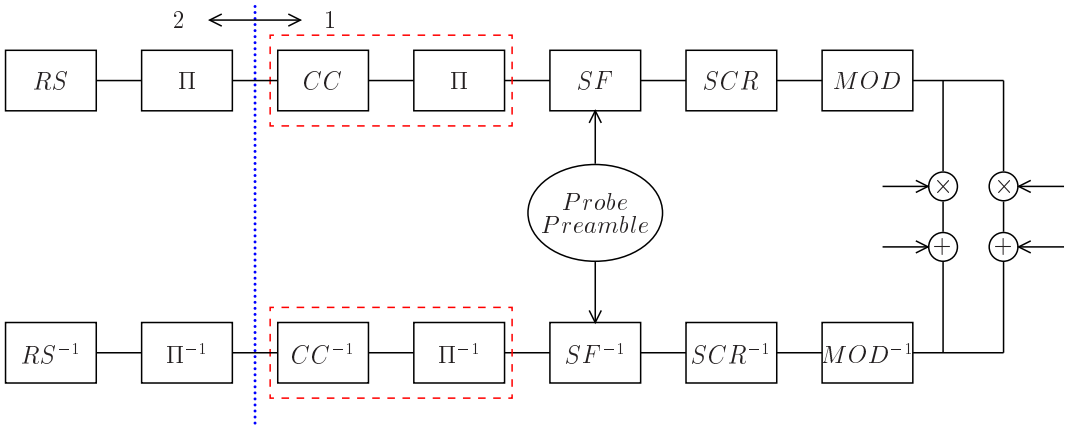


FIG. 6.13: Camada de enlace com RS FEC

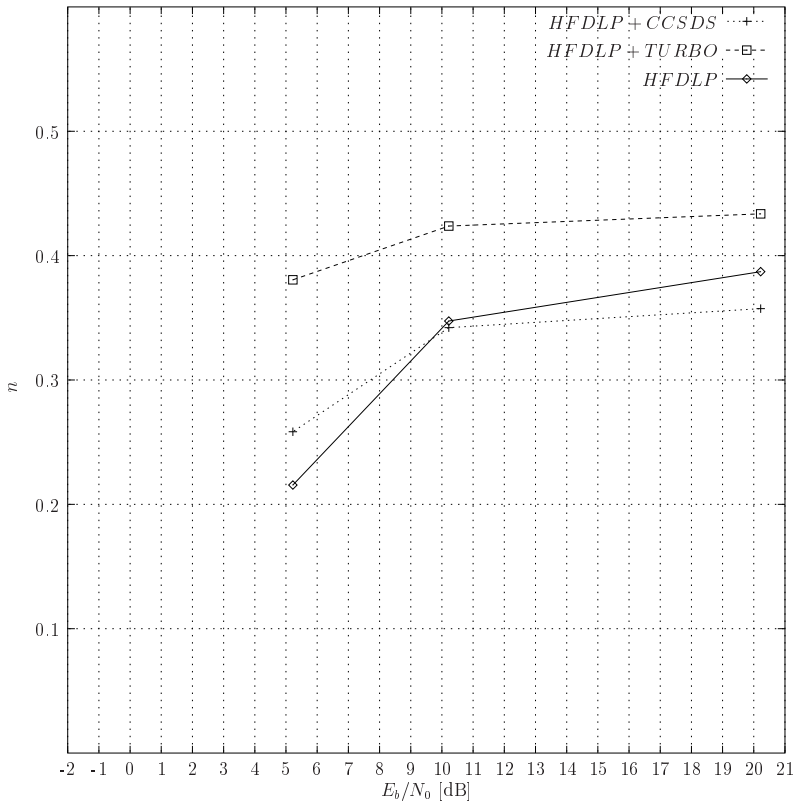


FIG. 6.14: Vazão canal bom

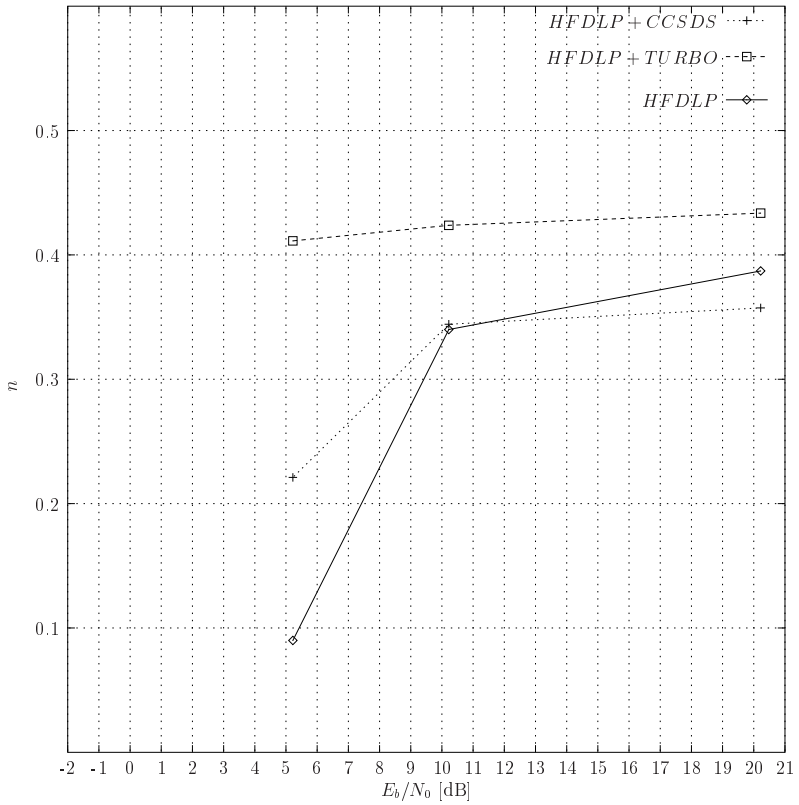


FIG. 6.15: Vazão canal moderado

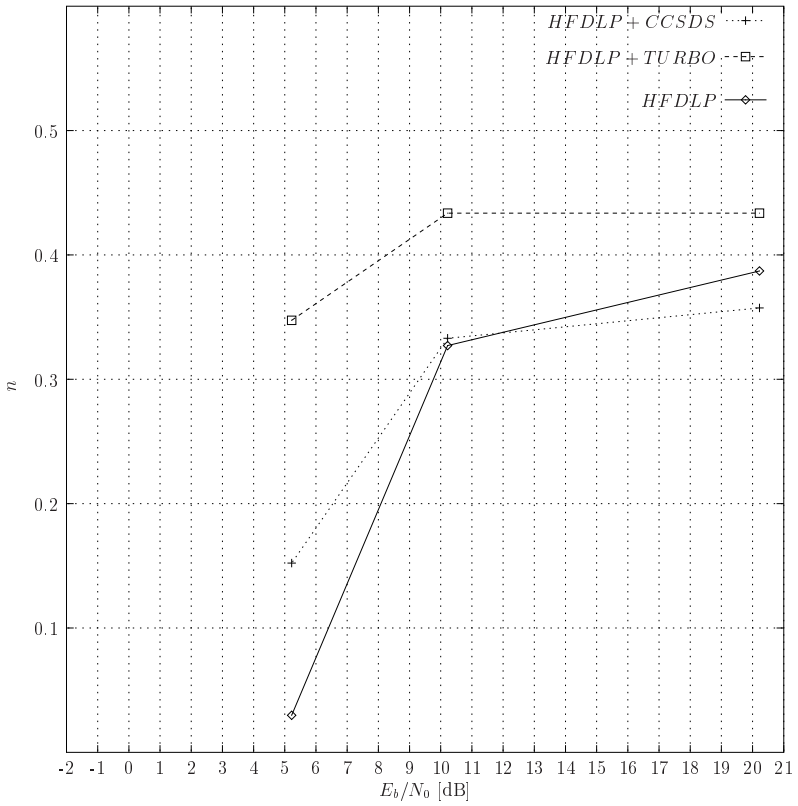


FIG. 6.16: Vazão canal ruim

7 CONCLUSÃO

Neste trabalho, foram apresentados e estudados os códigos concatenados em série CCSDS e os códigos concatenados em paralelo Turbo que são considerados como os melhores códigos para canais AWGN onde a relação sinal-ruído é baixa.

Este estudo mostrou que, além de ter bom desempenho em termos de taxa de erro de bit na camada fisíca onde está disponível a informação lateral a respeito da demodulação dos símbolos recebidos do canal, os códigos concatenados aqui abordados também foram capazes de melhorar bastante a vazão na camada de enlace quando combinados ao protocolo do tipo ARQ puro, no caso o HFDLP.

Cabe ressaltar ainda que os resultados obtidos com ambos os códigos se referem à camada física específica formada por modem e canal HF bem definidos. Mostrou-se que o processo de erro na camada física produz surtos que se propagam para a camada de enlace. Estes surtos, bem como os intervalos, possuem quase sempre os mesmos comprimentos mudando apenas a freqüência com que ocorrem ao longo do tempo, quando o canal passa do tipo bom para o moderado e o ruim.

Os códigos Turbo demonstraram um desempenho supreendente para baixas energias e, de modo geral, robusto em relação a variação da vazão com mudanças de E_s/N_0. O fato do desempenho do código Turbo ser superior em relação ao CCSDS já era esperado em virtude das características de desempenho dos dois códigos, verificadas nos primeiros capítulos desta tese. Entretanto, a comparação do desempenho dos esquemas ARQ híbridos usando os dois códigos mostrou que neste aplicação os códigos Turbo são muito superiores aos códigos RS e convolucional concatenados. Tal resultado foi obtido até mesmo para tamanhos de quadros (entrelaçamento) considerados pequenos para códigos Turbo.

Ainda assim o CCSDS mostrou um bom desempenho em relação ao HFDLP sem FEC para baixas energias. A vantagem do CCSDS é que ele pode ser usado com eficiência para pacotes pequenos, onde o desempenho do códigos Turbo deixa a desejar em virtude do ganho de entrelaçamento ser pequeno. Uma prova deste bom desempenho é que no esquema híbrido com o código Turbo usou-se este apenas para os quadros de dados e o CCSDS para os quadros de controle.

A principal novidade deste trabalho foi sem dúvida o uso de códigos Turbo na camada de enlace, portanto sem a informação suave normalmente disponível nas aplicações até

então desenvolvidas com estes códigos. Para esta aplicação, verificou-se que o principal algoritmo de decodificação de códigos Turbo necessita de alguns ajustes para produzir bons resultados, principalmente para energias medianas e altas. Este trabalho não realizou um estudo profundo sobre este aspecto mas apenas encontrou soluções heurísticas. Descobriu-se que a diminuição do valor do parâmetro L_c para energias altas faz com que o algoritmo log-MAP não introduza erros na seqüência que estiver sendo decodificada.

7.1 PROPOSTAS DE TRABALHOS FUTUROS

Como propostas para trabalhos posteriores decorrentes deste trabalho, colocam-se os seguintes pontos:

- Avaliacão do desempenho de códigos concatenados em série SCCC e códigos LDPC.
- Utilização de outro modelo de canal HF de faixa larga.
- Estudo de um esquema adaptativo para ajuste do passo do estimador de canal VS-LMS e também a adoção de esquemas cegos de equalização, a fim de verificar possíveis melhorias na vazão ou redução da taxa de código
- Desenvolvimento de uma métrica inteira para substituir o logaritmo usado atualmente no decodificador log-MAP do simulador, e assim obter uma redução significativa na complexidade de decodificação.
- Estudar e definir critérios para redução do valor do parâmetro L_c na decodificação iterativa, quando os valores fornecidos ao decodificador forem valores abruptos.
- O estudo de técnicas de combinação de códigos, tais como RCPT (*Rate Compatible Turbo Codes*) e RCPC (*Rate Compatible Punctured Convolutional*), adaptadas para a camada de enlace HF.
- Proposta de um algoritmo para estimação do canal a partir das informações disponíveis na camada de enlace, e assim atuar sobre a taxa de código, modo de operação do modem e tamanho de quadros e séries.

8 REFERÊNCIAS BIBLIOGRÁFICAS

ARRL. *AX.25 Link Access Protocol for Amateur Packet Radio.* ARRL, 2.2 edition, november 1997.

BAHL, L. R., COCKE, J., JEINEK, F. e RAVIV, J. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transaction on Information Theory*, IT-20: 248–287, march 1974.

BELLO, P. A. Characterization of randomly time-variant linear channels. *IEEE Transactions on Communications Systems*, december 1963.

BENEDETTO, S. e MONTORSI, G. Design of parallel concatenated convolutional codes. *IEEE Transactions on Communications*, 44:591–600, may 1996.

BERROU, C. Near shannon limit error-correcting coding and decoding: Turbo codes. Em *Proceedings of ICC'93*, 1993.

BIGLIERI, E., PROAKIS, J. e SHAMAI, S. Fading channels: Information-theoretic and communications aspects. *IEEE Transaction on Information Theory*, 44(6):2619–1640, october 1998.

BLAHUT, R. E. *Information Theory.* Reading, MA: Addison-Wesley, 1988.

CARVALHO, T. C. M. B. *Arquiteturas de Redes de Computadores OSI e TCP/IP.* Makron Books, 1994.

CCIR. *Use of High Frequency Ionospheric Channel Simulators.* CCIR, march 1992.

CHI, Z., WANG, Z. e PARHI, K. High throughput low energy FEC/ARQ technique for short frame turbo codes. Em *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000*, volume 5, p. 2653–2656. IEEE, 2000.

CHOCKALINGAM, A., ZORZI, M. e TRALLI, V. Wireless TCP performance with link layer FEC/ARQ. Em *IEEE International Conference on Communications*, p. 1212–1216, 1999.

COMER, D. E. *Internetworking with TCP/IP: Principles, Protocols and Architecture.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

COSTELLO, JR., D. J. Free distance bounds for convolutional codes. *IEEE Transaction on Information Theory*, p. 356–365, may 1994.

MIL-STD-188-110A: Interoperability and Performance Standards for Data Modems. DoD, september 1991.

MIL-STD-188-141A: Interoperability and Performance Standards for Medium and High Frequency Radio Equipment. DoD, september 1993.

MIL-STD-187-721C: Interface and Performance Standard for Automated Control Appliqué for HF Radio. DoD, november 1994.

MIL-STD-188-220 B:Interoperability Performance. DoD, november 1998. ver2.2.

DORNHOFF, L. L. e HOHN, F. E. *Applied Modern Algebra*. Macmillan, 1980.

FARHANG-BOROUJENY, B. Channel equalization via channel identication: Algorithms and simulation results for rapidly fading HF channels. *IEEE Transactions on Communications*, 44(11):1409–1412, november 1996.

FORNEY, JR., G. D. *Concatenated Codes*. Tese de Doutorado, Cambridge University, 1966.

FORNEY, JR., G. D. Convolutional codes i: Algebraic structure. *IEEE Transactions on Information Theory*, p. 720–738, 1970.

HAGENAUER, J. e HOEHER, P. A viterbi algorithm with soft-decision outputs and its applications. Em *Proceedings GlobeCom*, 1989.

HAGENAUER, J., OFFER, E. e PAPKE, L. Iterative decoding of binary block and convolutional codes. *IEEE Transaction on Information Theory*, 42(2):429–445, march 1996.

HALL, E. K. e WILSON, S. G. Design and analysis of turbo codes on rayleigh fading channels. *IEEE Jornal on Selected Areas in Communications*, 16(2):160–174, february 1998.

HARRIS, R. W., CHABRIES, D. M. e BISHOP, F. A. A variable step (VS) adaptative filter algorithm. *IEEE Transactions on Acoustic Speech and Signal Processing*, 34: 309–316, april 1986.

HAYKIN, S. *Digital Communications*. Prentice Hall, 2000.

JOHNSON, E. E. *Advanced High-Frequency Radio Communications*. Artech House, 1997.

KASAMI, T. K. e T. LIN, S. Linear block codes for error detection. *IEEE Transaction on Information Theory*, 29:131–137, january 1983.

MCELIECE, R. J. *Finite Fields for Computer Scientists and Engineers*. Boston: Kluwer Academic Publishers, 1987.

MOLNÁ, B. G., FRIGYES, I., BODNÁR, Z. e HERCZKU, Z. The WSSUS channel model: comments and a generalisation. Em *Global Telecommunications Conference*, p. 158–162, 1996.

NCS. *FED STD 1052: HF Radio Modems*. NCS, 1994.

PROAKIS, J. G. *Digital Communications*. Mc Graw Hill, 1998.

ROWITCH, D. N. e MILSTEIN, L. B. On performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes. *IEEE Transactions on Communications*, 48(6):948–959, june 2000.

SHANNON, C. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, jan 1948.

SKLAR, B. *Digital Communications Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

TANENBAUM, A. S. *Computer Networks*. Englewood Cliffs,NJ:Prentice-Hall, second edition edition, 1989.

VITERBI, A. J. Error bounds for convolutional codes and an asymptotically optimun decoding algorithm. *IEEE Transaction on Information Theory*, p. 260–269, 1967.

VITERBI, A. J. e OMURA, J. K. *Principles of Digital Communications and Coding*. Mc Graw Hill, 1979.

WATTERSON, C. C. e JUROSHEK, J. R. Experimental confirmation of an HF channel model. *IEEE Transactions on Communication Technology*, COM-18(6):792–803, december 1970.

WELLS, R. B. *Applied Coding and Information Theory for Engineers*. Englewood Cliffs,NJ:Prentice-Hall, 2000.

WICKER, S. B. *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs,NJ:Prentice-Hall, 1995.

YIP, K. W. e NG, T. S. An analytic discrete-time model for a fading dispersive WSSUS channel. Em *IEEE Vehicular Technology Conference*, 1994.

ZORZI, M. e RAO, R. R. Perspectives on the impact of error statistics on protocols for wireless networks. *IEEE Personnal Communications Magazine*, october 1999.

9 APÊNDICES

9.1 APÊNDICE 1: CÓDIGOS CORRETORES DE ERRO

A tarefa da engenharia de sistemas de comunicações digitais é prover um sistema capaz de transmitir informações entre dois pontos quaisquer de modo tão rápido e confiável quanto seja necessário e possível.

Os dois parâmetros disponíveis para o projeto de tal sistema são a potência do sinal transmitido e a largura de faixa do canal de comunicação. Esses dois parâmetros, em conjunto com a densidade espectral de potência do ruído no receptor, determinam a razão E_b/N_0 entre a energia de bit e a densidade espectral de potência. Dada esta razão, a modulação empregada e o canal, é possível determinar a confiabilidade do sistema.

Os códigos corretores de erro, também chamados de códigos para canal, podem ser usados para proteger a informação transmitida de modo a aumentar sua imunidade aos efeitos do ruído de recepção e canal de transmissão. Neste caso, aumentar imunidade significa diminuir a probabilidade de erro da informação que efetivamente foi transmitida. Tal finalidade é alcançada graças à inserção de redundância controlada na informação, tanto pela adição de bits quanto pela expansão da constelação de sinais do canal. Deste modo, o receptor pode tanto detectar como, possivelmente, corrigir os erros.

9.1.1 SISTEMA DE COMUNICAÇÕES COM CÓDIGO

A FIG. 9.1 apresenta o diagrama em bloco de um sistema de comunicações com código. O bloco demodulador contém todas as ferramentas necessárias para receber o sinal, filtrar, remover interferência entre símbolos e decidir qual símbolo foi transmitido. O bloco decodificador é responsável por decidir a partir da palavra-código recebida $\mathbf{r}$, seguindo um critério de decisão estabelecido, qual palavra-informação $\hat{\mathbf{m}}$ foi transmitida.

9.1.2 TIPOS DE DECODIFICADORES

Seja a palavra-informação ou mensagem $\mathbf{m}$ codificada para gerar a palavra-código $\mathbf{c}$ que após passar por um canal sem memória dá origem a palavra recebida $\mathbf{r}$. O problema do decodificador é recuperar a mensagem $\mathbf{m}$ usando para isto um critério que minimize a probabilidade de erro.

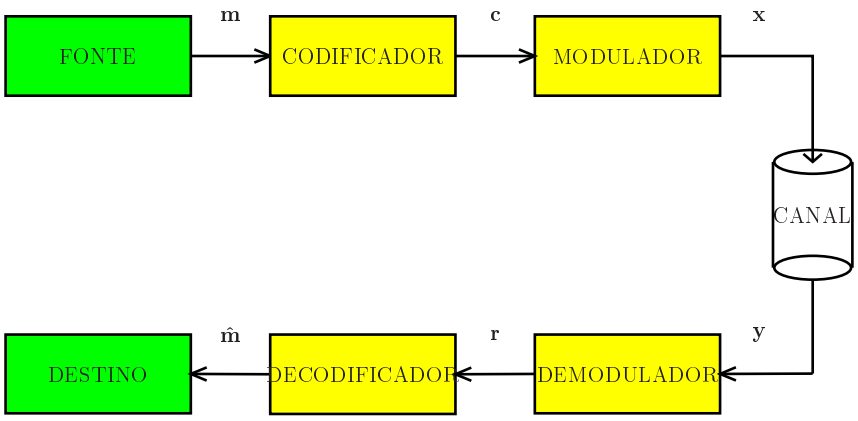


FIG. 9.1: Diagrama de um sistema de comunicações com código corretor de erro

9.1.2.1 DECODIFICADOR DE MÁXIMA PROBABILIDADE A POSTERIORI

A aplicação deste critério, denominado MAP (*Maximum Probability a Posteriori*), exige que exista alguma informação *a priori* disponível a respeito da fonte. O critério MAP permite usar a informação disponível anteriormente para procura a mensagem $\hat{\mathbf{m}}$ que maximiza $p(\mathbf{c}|\mathbf{r})$, ou seja,

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{c}\in\mathcal{C}} p(\mathbf{c}|\mathbf{r}) \tag{9.1}$$

ou

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{m}\in\mathcal{M}} p(\mathbf{m}|\mathbf{r}) \tag{9.2}$$

9.1.2.2 DECODIFICADOR DE MÁXIMA VEROSSIMILHANÇA

Este critério, denominado ML (*Maximum Likelihood*), deve ser usado quando não existe nenhuma informação a priori disponível a respeito da fonte. Quando as mensagens produzidas pela fonte são equiprovavéis, este critério minimiza a probabilidade de erro. O critério ML consiste em procurar a mensagem $\hat{\mathbf{m}}$ que maximiza $p(\mathbf{r}|\mathbf{c})$, ou seja,

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{c}\in\mathcal{C}} p(\mathbf{r}|\mathbf{c}) \tag{9.3}$$

ou

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{m}\in\mathcal{M}} p(\mathbf{r}|\mathbf{m}) \tag{9.4}$$

9.1.3 CONCEITOS BÁSICOS DE CÓDIGOS

Nesta seção serão introduzidos alguns conceitos e definições que auxiliam o entendimento da teoria dos códigos corretores de erro.

9.1.3.1 DEFINIÇÃO

As definições de código e codificador são apresentadas a seguir.

Definição 9.1 (Código) *Um código $\mathcal{C}$ é um conjunto de seqüências de símbolos pertencentes a um alfabeto $\mathcal{A}$ geradas na saída de um codificador. Tais seqüências são usualmente denominadas de palavras-código.*

Definição 9.2 (Codificador) *É um dispositivo que cria um mapeamento biunívoco entre seqüências de entrada e seqüências de saída. As seqüências de entrada são normalmente denominadas de palavras-informação.*

A partir destas definições é possível observar que um dado código pode ser obtido a partir de mais de um codificador.

9.1.3.2 CARACTERÍSTICAS DE UM BOM CÓDIGO

Na busca de códigos práticos e eficientes, há algumas características a serem observadas por um bom código. Entre elas, destacam-se as seguintes:

- **Detecção e correção dos erros introduzidos pelo canal**: é a mais importante característica, pois idealmente o melhor código seria aquele que pudesse corrigir todos os erros introduzidos pelo canal.
- **Transmissão eficiente da informação**: esta característica está relacionada com a eficiência do código, pois indica o fluxo ou vazão da informação pelo canal codificado.
- **Facilidade de codificação e decodificação**: esta característica também está relacionada com a complexidade computacional da implementação do codificador e decodificador em hardware ou software. Em geral, quanto maior o número de erros que um código pode corrigir por palavra-código, maior será a redundância da transmissão, e mais complicada será a codificação e decodificação.

9.1.3.3 EFICIÊNCIA E A TAXA DO CÓDIGO

A função básica de um codificador/decodificador é mapear conjuntos de símbolos do alfabeto de entrada do codificador em conjuntos de símbolos do alfabeto de saída do codificador que normalmente é igual ao da entrada.

Um codificador genérico pode ser caracterizado pelo diagrama da FIG. 9.2. Nesta figura, um bloco de informação $\mathbf{m}$ com k símbolos, $\mathbf{m} = (m_0, \cdots, m_{k-1})$, na entrada do codificador é transformado em um bloco de código $\mathbf{c}$ de tamanho n, $\mathbf{c} = (c_0, \cdots, c_{n-1})$, na saída do codificador.

FIG. 9.2: Diagrama de um codificador

Definição 9.3 *Seja q a cardinalidade do alfabeto da fonte e $q^{'}$ a cardinalidade do alfabeto usado na codificação. A taxa deste código é definida como:*

$$R_c = \frac{\log_q(M)}{\log_{q'}(M')} \tag{9.5}$$

onde M é número de palavras-código no código e $M^{'}$ o número de mensagens possíveis da fonte.

Os codificadores mais comuns recebem palavras-informação de tamanho k e produzem palavras-código de tamanho n ambas de tamanhos fixos. Além disso, em sistemas digitais, é comum o uso de alfabeto binários $\mathcal{A} = 0, 1$. Deste modo, tem-se simplificadamente a EQ 9.6 quando $q = q^{'} = 2$, $M = 2^k$ e $M^{'} = 2^n$. Desta forma, serão necessários n bits de código para transmissão de k bits de informação.

$$R_c = \frac{k}{n} \tag{9.6}$$

A taxa de código constitui uma medida da quantidade de redundância adicionada à informação. Se forem adicionados mais bits de código à palavra-informação, o novo código poderá ser melhor no sentido de que mais erros poderão ser detectados e corrigidos.

Entretanto, esta diminuição da taxa de código fará com que menos bits de informação sejam enviados cada vez que uma palavra-código seja transmitida. Tal relação cria um compromisso entre a capacidade de correção e deteção de erros e o fluxo de informação pelo sistema de comunicação.

Considerando que o melhor código é aquele que minimiza a probabilidade de erro, ficam então as perguntas: Qual seria o melhor código para uma dada taxa de código ? Seria possível construir tal código ? Estas perguntas podem ser respondidas à luz da Teoria da Informação (BLAHUT, 1988). Conceitos tais como entropia, informação mútua, capacidade e taxa de corte podem ajudar a responder estas perguntas. A seguir, estes conceitos serão introduzidos de forma gradual e sem muito formalismo, de modo a ajudar no desenvolvimento deste trabalho. Sendo assim, neste ponto faz-se necessário a introdução do conceito de capacidade.

9.1.3.4 CAPACIDADE

Em 1948, C. E. Shannon publicou um trabalho entitulado *A Mathematical Theory of Communication* (SHANNON, 1948) onde diz que, sempre que a taxa de bits de informação enviada pelo canal for menor que a capacidade do canal, existirá um código corretor de erro capaz de corrigir todo e qualquer erro introduzido pelo canal. Esta afirmação é conhecida como Teorema da Codificação do Canal. É importante notar que este teorema é existencial, ou seja, ele não revela como construir tal código.

A definição de capacidade um canal DMC (*Discrete Memoryless Channel*) (BLAHUT, 1988) é dada através função informação mútua $I(\mathbf{X};\mathbf{Y})$ onde $\mathbf{X}$ e $\mathbf{Y}$ são variáveis aleatórias que assumem valores pertencentes respetivamente aos alfabetos de entrada do canal com cardinalidade J e de saída do canal com cardinalidade K.

Definição 9.4 (Capacidade) *A capacidade de um canal DMC é definida pela maximização da função informação mútua média* $I(\mathbf{X};\mathbf{Y})$ *em qualquer uso simples do canal ou intervalo de sinalização. Esta maximização é feita sobre todas as possíveis distribuições de probabilidade* $\{p(x_j)\}$ *sobre* $\mathbf{X}$.

$$C \triangleq \max_{\{p(x_j)\}} I(\mathbf{X};\mathbf{Y}) \tag{9.7}$$

onde a informação mútua I é definida pela EQ 9.8.

$$I(\mathbf{X};\mathbf{Y}) \triangleq \sum_{j=0}^{J-1}\sum_{k=0}^{K-1} p(x_j,y_k)\log_2\left[\frac{p(x_j|y_k)}{p(x_j)}\right] \tag{9.8}$$

onde:

$p(x_j, y_k)$	é a distribuição de probabilidade conjunta
$p(x_j\|y_k)$	é a distribuição de probabilidade condicional da entrada condicionada a saída do canal
$p(x_j)$	é a distribuição de probabilidade da fonte

A capacidade associada a um canal AWGN (*Aditive Wide Gaussian Noise*) com sinalização antipodal é expressa pela 9.9.

$$C_{AWGN} = W \log_2(1 + \frac{E_s}{N_0}) \tag{9.9}$$

onde:

W	é a banda do canal em Hz
E_s	é a energia média do sinal em cada intervalo de símbolo de duração T_s segundos
N_0	é a variância do ruído branco Gaussiano

A EQ 9.9 pode ser reescrita em função da eficiência espectral $\eta = C/W$.

$$\frac{E_b}{N_0} > \frac{2^\eta - 1}{\eta} \tag{9.10}$$

onde:

η	é a eficiência espectral da modulação empregada medida em $b/s \cdot Hz$
E_b	é a energia média por bit de informação

A capacidade de um canal BSC (*Binary Symmetric Channel*) é definida a partir da probabilidade de erro ou transição do canal p conforme ilustrado na FIG. 9.3. Como um canal BSC é simétrico, a probabilidade de ocorrer um erro é independente do símbolo sendo transmitido.

$$C_{BSC} = 1 + p \log_2 p + (1-p) \log_2 (1-p) \tag{9.11}$$

A confiabilidade de um canal BSC, $1 - p$, é a probabilidade de que nenhum erro ocorra. Normalmente, a confiabilidade de um canal será maior do que meio. Se $p = 1/2$, nenhuma informação poderá ser transmitida pelo canal, pois este terá capacidade nula. Caso $p > 1/2$, bastará inverter os bits recebidos para recair no caso $p < 1/2$.

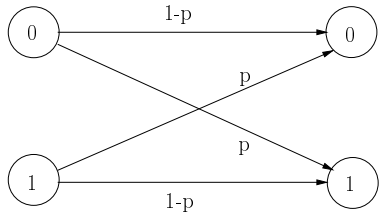


FIG. 9.3: Canal binário simétrico

9.1.3.5 GANHO DE CÓDIGO

Para definir ganho de código é preciso antes definir o conceito de taxa de erro de bit ou BER (*Bit-Error Rate*).

Definição 9.5 (BER) *É a probabilidade que um erro de bit ocorra em um dado tempo.*

De modo geral, o BER diminui com o aumento da razão E_b/N_0 da energia por bit de informação pelo parâmetro da densidade espectral de potência do ruído no receptor.

A definição de ganho de código é dada a seguir.

Definição 9.6 (Ganho de Código) *É a diferença entre o valor de E_b/N_0 necessário para atingir um dado BER em um sistema com código e o E_b/N_0 necessário para alcançar o mesmo BER em um sistema sem código.*

9.1.3.6 PESO E DISTÂNCIA DE UM CÓDIGO

A peso de Hamming de uma palavra é simplesmente o seu número de símbolos não nulos. Deste modo, se palavras de tamanho 5 estiverem sendo usadas, o peso de 01101 será 3, ou seja, $w_H(01101) = 3$.

A distância Hammming entre duas palavras-código $\mathbf{c}_1$ e $\mathbf{c}_2$ pertencentes a $\mathcal{C}$ é o número de posições em que os símbolos diferem entre as duas palavras. Por exemplo, $\mathbf{c}_1 = 01101$ e $\mathbf{c}_1 = 01100$ diferem apenas no último símbolo de modo que a distância de Hamming, ou apenas distância, entre a primeira e segunda é 1. Isto é indicado por $d_H(\mathbf{c}_1, \mathbf{c}_2) = 1$.

A função $d_H(\mathbf{u}, \mathbf{v})$ onde $\mathbf{u}$ e $\mathbf{v}$ são palavras-código definem uma métrica euclidiana. Uma métrica euclidiana é uma função que associa um número a cada dois elementos em um conjunto e que satisfaz os axiomas de Euclides sobre distância.

A partir do conceito de distância de Hamming é possível definir a distância mínima de um código $\mathcal{C}$ como sendo a menor distância de Hamming dentre todos os pares de palavras-código pertencentes ao código conforme ilustrado na EQ 9.12.

$$d_{min} \triangleq \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}} d_H(\mathbf{c}_1, \mathbf{c}_2) \tag{9.12}$$

Deste modo, para corrigir um erro primeiramente será necessário checar se foi recebido uma palavra-código válida. Se a palavra-código recebida não fizer parte do código, será necessário substitui-la pela palavra-código mais próxima usando a métrica acima definida.

Os diagramas a seguir ilustram melhor a situação.

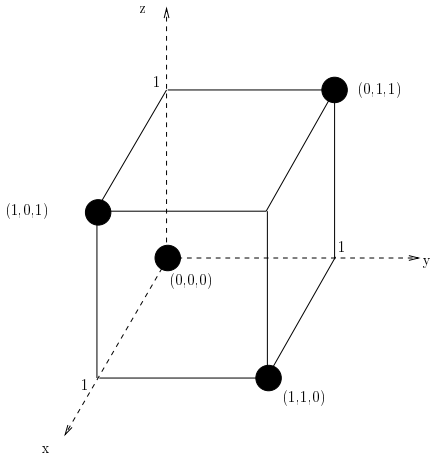


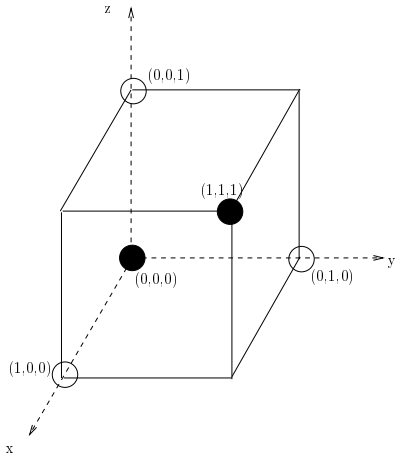
FIG. 9.4: Código para detecção de erro

Na FIG. 9.4, palavras-código de peso dois são colocadas nos vértices do cubo unitário. As palavras-código são representadas por esferas pretas. Logo, neste código há quatro palavras-código : 000, 101, 011 e 110. Caso fosse recebido 111, seria possível detectar a ocorrência de um erro mas não seria possível determinar sua posição. Este código é bom para detecção mas não para correção de erros.

Na FIG. 9.5, apenas 000 e 111 são palavras-código. Se 111 fosse transmitido e um erro fosse introduzido pelo canal, 110 ou 011 ou 101 seria recebido. Como estas seqüências são todas adjacentes ao vértice 111, na recepção, a mensagem mais provável de ter sido transmitida seria 111, pois seria mais provável ocorrer 1 erro do que 2 erros segundo o critério de máxima verossimilhança .

A diferença importante entre estes dois códigos é que o primeiro tem distância 2 entre cada par de palavras-código e o segundo tem distância 3. Em geral, é correto afirmar, segundo o critério acima adotado, que t_d erros podem ser detectados se a mínima distância entre duas palavras-código quaisquer for $t_d + 1$ e que t_c erros podem ser corrigidos se a distância mínima do código for $2t_c + 1$. Logo seguem as desigualdades 9.13 e 9.14.

FIG. 9.5: Código para correção de erros



$$t_d \leq d_{min} - 1 \tag{9.13}$$

e

$$t_c \leq \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \tag{9.14}$$

onde:

$t_d = 2t_c$	é número de símbolos errados que podem ser detetados
t_c	é número de símbolos errados que podem ser corrigidos

9.1.3.7 MODIFICAÇÃO DE CÓDIGOS LINEARES

Em muitas aplicações, podem ser impostas algumas restrições no tamanho da palavra-informação ou no tamanho da palavra-código. Deste modo, pode-se alterar o tamanho destas palavras sem alterar o código $\mathcal{C}$ utilizando-se uma das operações a seguir definidas.

Definição 9.7 (Puncionamento) *Um código é puncionado através da retirada de um de seus símbolos de paridade. Um código* (n, k) *torna-se então* $(n-1, k)$.

Definição 9.8 (Encurtamento) *Um código é encurtado através da retirada de um de seus símbolos de informação. Um código* (n, k) *torna-se então* $(n-1, k-1)$.

Definição 9.9 (Expurgamento) *Um código é expurgado através da retirada de algumas palavras-código. Se* $(q-1)/q$ *palavras-código forem expurgadas de modo que as remanescentes formem um subcódigo linear então um código* (n, k) q*-ário torna-se um código* $(n, k-1)$ q*-ário.*

Definição 9.10 (Extensão) *Um código é extendido através da adição de um símbolo de paridade redundante. Um código* (n, k) *torna-se então* $(n + 1, k)$.

Definição 9.11 (Expansão) *Um código é expandido através da adição de algumas palavras-código. Se o número de palavras-código é incrementado de um fator* q *de modo que o código resultante seja linear então um código* (n, k) q*-ário torna-se um código* $(n, k + 1)$.

Definição 9.12 (Alongamento) *Um código é alongado através da adição de um símbolo de informação. Um código* (n, k) *torna-se então* $(n + 1, k + 1)$.

9.2 APÊNDICE 2: CÓDIGOS CONVOLUCIONAIS

Há basicamente duas grandes categorias de códigos corretores de erro: em bloco e convolucionais. Um código em bloco linear é completamente descrito por dois números inteiros, n e k, e uma matrix ou polinômio gerador. O número k é o número de bits de informação que formam a entrada do codificador em bloco. O número n é o número total de bits associado com a palavra-código na saída do codificador. A característica de um código em bloco linear é que cada palavra-código de tamanho n é unicamente determinada pela palavra-informação de tamanho k. A razão k/n é denominada taxa de código e mede a redundância criada pelo código.

Em um código convolucional, a razão k/n tem o mesmo significado que para o código em bloco, ou seja, medida de redundância. Entretanto, n não mais define o tamanho da palavra-código mas apenas quantos bits saem do codificador quando k bits de informação entram. Uma característica importante dos códigos convolucionais, que os difere dos códigos em bloco, é que o codificador possui memória, ou seja, cada n-upla gerada pelo codificador convolucional não é apenas função da k-upla de informação na entrada mas também do estado atual do codificador.

9.2.1 CODIFICADOR

O codificador é o dispositivo que recebe k-uplas na entrada e produz n-uplas na saída onde estas n-uplas pertencem ao conjunto de saída denominado código $\mathcal{C}$. Deste modo, o papel do codificador é estabelecer um mapeamento biunívoco entre palavras-informação e palavras-código.

Quando os símbolos das palavras mapeadas pertencem a $GF(2)$, tanto o código como o codificador são ditos binários. Desta forma, denomina-se BCC (*Binary Convolutional Code*) o código convolucional binário e BCE (*Binary Convolutional Encoder*) o codificador convolucional binário. Suas definições são dadas a seguir.

Definição 9.13 (BCC) *É um conjunto de seqüências produzido na saída de um filtro linear sobre um alfabeto pertencente a um corpo finito binário, GF(2).*

Definição 9.14 (BCE) *É um dispositivo que provê um método de mapeamento de mensagens em palavras-código através da utilização de um filtro linear FIR (Finite Impulse*

Response) ou IIR (Infinite Impulse Response) sobre um alfabeto finito binário, GF(2). A saída do filtro pode ser descrita como a convolução da seqüência de entrada por um polinômio que é função da estrutura do filtro.

O filtro linear associado a um BCE pode ser tanto FIR como IIR conforme ilustrados nos diagramas das FIG. 9.6 e FIG. 9.7. É importante notar ainda que um dado BCC pode possuir mais de um BCE e que um BCE pode ser sistemático ou não-sistemático.

Definição 9.15 (Codificador FIR) *Um codificador é FIR se suas saídas são combinações lineares da entrada corrente e de um número de finito de entradas passadas. Para um codificador com k entradas e n saídas tem-se a seguinte forma para cada uma das saídas:*

$$x_j^p = \sum_{i=1}^{k} \sum_{l=0}^{\nu_i} g_{i,p,l} m_{j-l}^i, \qquad 1 \leq p \leq n$$

onde: $\{g_{i,p,l}\}$ é a seqüência de geradores que relaciona um seqüência particular de entrada $\{m_l^i\}$ com uma seqüência particular de saída $\{x_j^p\}$

A memória de cada uma das k entradas é enumerada pelo vetor de memória $(\nu_1, \nu_2, \cdots, \nu_k)$. Desta forma, o i-ésimo registrador de deslocamento de entrada possui ν_i elementos de memória. A complexidade de estados do codificador é indicada pela memória total do codificador $\nu = \nu_1 + \nu_2 + \cdots + \nu_k$. O número de estados do codificador é 2^ν, enquanto o comprimento da janela é determinado pela ordem de memória $M = \max_{1 \leq i \leq k} \nu_i$.

Outro conceito importante é o de comprimento restritivo K que indica quantos bits de saída poderão ter seus valores afetados por um bit na entrada,ou seja, $K = M + 1$.

Usando a transformada D das seqüências ou palavras-informação, a saída do codificador pode ser expressa em termos de uma matriz geradora polinomial $G(D)$.

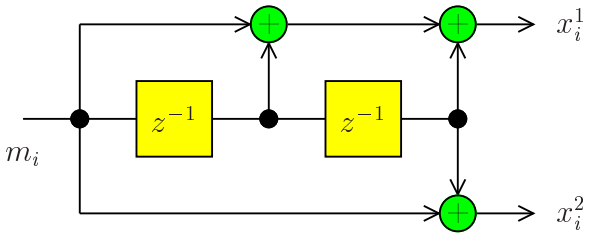
$$\begin{aligned}
\mathbf{x}(D) &= [x_1(D), x_2(D), \cdots, x_n(D)] \\
&= [m_1(D), \cdots, m_k(D)] \begin{bmatrix} g_{1,1}(D) & g_{1,2}(D) & \cdots & g_{1,n}(D) \\ g_{2,1}(D) & g_{2,2}(D) & \cdots & g_{2,n}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k,1}(D) & g_{k,2}(D) & \cdots & g_{k,n}(D) \end{bmatrix} \\
&= \mathbf{m}(D) G(D)
\end{aligned}$$

onde:

$m_i(D) = \sum_j m_j^i D^j$ seqüência temporal da informação na entrada i

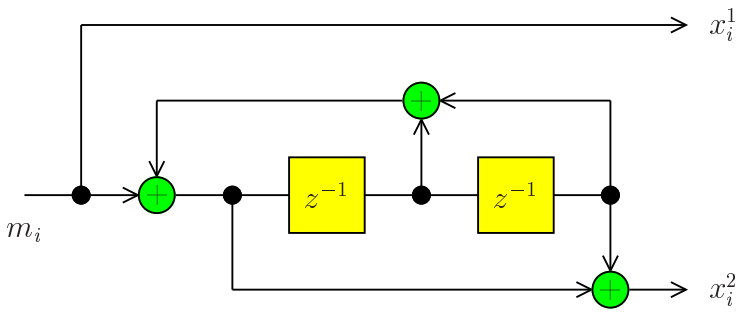
$g_{i,p}(D) = \sum_{j=0}^{\nu_i} g_{i,p,j} D^j$ indica a relação da saída p com a entrada i

Definição 9.16 (Codificador IIR) *Um codificador é IIR se suas saídas são combinações lineares da entrada corrente, de um número finito de entradas passadas e de um número infinito de saídas passadas. Neste caso, os termos $g_{i,p}(D)$ da matriz $G(D)$ serão racionais na variável D.*



$$G^{N-FIR}(D) = [1 + D + D^2 \qquad 1 + D^2]$$

FIG. 9.6: Codificador FIR



$$G^{S-IIR}(D) = [1 \qquad \frac{1+D^2}{1+D+D^2}]$$

FIG. 9.7: Codificador IIR

Como um mesmo BCC pode ser obtido usando-se vários BCE, faz-se necessário definir o conceito de codificador mínimo e apresentar dois teoremas importantes com respeito a este codificador (FORNEY, 1970).

Definição 9.17 (BCE mínimo) *É o BCE para um dado BCC que possui a menor memória total ν.*

Teorema 9.1 *Todo codificador mínimo para um dado BCC possui o mesmo vetor memória.*

Teorema 9.2 *Todo BCC possui um codificador mínimo do tipo FIR e um codificador mínimo sistemático (normalmente IIR).*

Os codificadores mínimos sistemáticos do tipo FIR são pouco usados pois possuem d_{free}^H pequena em comparação com os não-sistemáticos. A 9.2.1 mostra o d_{free}^H máximo para códigos sistemáticos e não-sistemáticos (WICKER, 1995).

TAB. 9.1: Máximo d_{free} para códigos convolucionais de taxa 1/2

Comprimento restritivo K	Máximo d_{free}^{H} sistemático	Máximo d_{free}^{H} não-sistemático
2	3	3
3	4	5
4	4	6
5	5	7
6	6	8
7	6	10

9.2.2 DECODIFICADOR

Para decodificação de códigos convolucionais dispoe-se basicamente de dois algoritmos com suas variações:

- **Algoritmo de Viterbi**: proposto por Viterbi (VITERBI, 1967) em 1967 é um algoritmo ML que minimiza a probabilidade de erro de seqüência em códigos convolucionais. Originalmente, o Algoritmo de Viterbi tem como saídas os valores decididos mas foi modificado por Hagenauer e Hoeher (HAGENAUER, 1989) para produzir valores suaves na saída.

- **Algoritmo BCJR**: proposto por Bahl (BAHL, 1974) em 1974 é um algoritmo MAP que minimiza a probabilidade de erro de bit em códigos convolucionais e em bloco. Foi modificado por Berrou (BERROU, 1993) para códigos convolucionais recursivos sistemáticos.

Nesta seção será abordada apenas a forma do BCJR modificada por Berrou.

A treliça de um codificador convolucional recursivo binário tem a estrutura mostrada na FIG. 9.8 onde as transições de estado tracejadas estão associadas ao símbolo -1 e as contínuas ao símbolo $+1$. Este último é o elemento neutro em relação a GF(2). Ainda na FIG. 9.8, S_i indica o estado do codificador no tempo i e o símbolo u_i esta associado com a transição do tempo $i-1$ ao tempo i. Os estados da treliça no nível $i-1$ e no nível i são indexados respectivamente pelas variáveis $s^{'}$ e s.

O objetivo do algoritmo MAP é produzir a probabilidade *a posteriori* logaritmica LAPP (*Logarithmic A Posteriori Probability*) dada pela EQ 9.15.

$$L(\hat{u}_i) = \log \frac{P(u_i = +1|\mathbf{y})}{P(u_i = -1|\mathbf{y})} = \log \frac{\sum_{S^+} p(s^{'}, s, \mathbf{y})}{\sum_{S^-} p(s^{'}, s, \mathbf{y})} \tag{9.15}$$

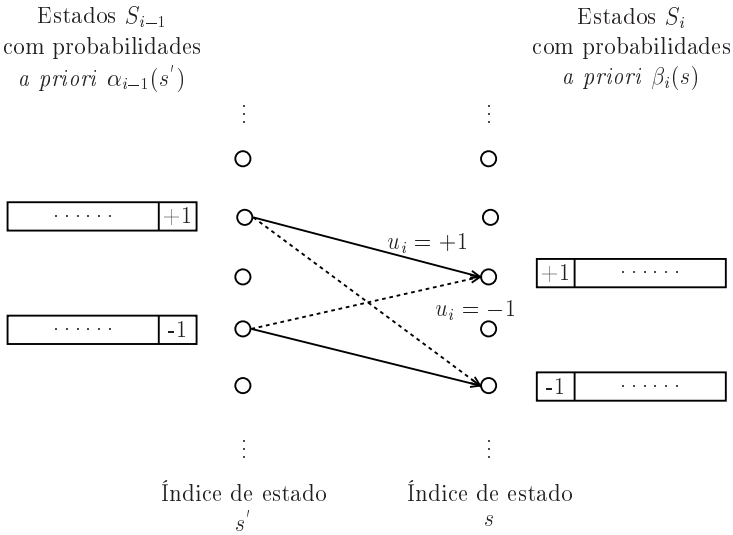


FIG. 9.8: Diagrama da treliça de um código recursivo sistemático

onde S^+ e S^- são os conjuntos de transições de estado $s^{'}$ para s onde, respectivamente, os símbolos $+1$ e -1 estão envolvidos.

O par de índices $s^{'}$ e s determina o símbolo de informação u_i e os símbolos codificados $x_{i,\nu}$ onde $\nu = 2, \cdots, n$. Desta forma, todas as transições podem ser rotuladas como $u_i, x_{i,2}, \cdots, x_{i,n}$ por tratar-se de um código sistemático de taxa $R_c = 1/n$.

Assumindo um canal sem memória, a probabilidade conjunta $p(s^{'}, s, \mathbf{y})$ pode ser expressa como um produto de três probabilidades independentes conforme EQ 9.16

$$\begin{aligned} p(s^{'}, s, \mathbf{y}) &= p(s^{'}, \mathbf{y}_{j<i}) \cdot p(s, y_i | s^{'}) \cdot p(\mathbf{y}_{j>i} | s) \\ &= \underbrace{p(s^{'}, \mathbf{y}_{j<i})}_{\alpha_{i-1}(s')} \cdot \underbrace{P(s|s^{'}) \cdot p(y_i|s, s^{'})}_{\gamma_i(s',s)} \cdot \underbrace{p(\mathbf{y}_{j>i}|s)}_{\beta_i(s)} \end{aligned} \tag{9.16}$$

onde $\mathbf{y}_{j<i}, \forall\ 1 \leq i \leq N$, denota a seqüência de símbolos recebidos y_j do começo da treliça até o tempo $i-1$ e $\mathbf{y}_{j>i}$ é a seqüência do tempo $i+1$ até o fim da treliça. Desta forma a recursão progressiva do algoritmo BCJR tem a forma dada pela EQ 9.17.

$$\alpha_i(s) = \sum_{s'} \gamma_i(s^{'}, s) \cdot \alpha_{i-1}(s^{'}) \tag{9.17}$$

enquanto a recursão regressiva tem a forma da EQ 9.18

$$\beta_{i-1}(s^{'}) = \sum_{s} \gamma_i(s^{'}, s) \cdot \beta_i(s) \tag{9.18}$$

A treliça tem que ter duração finita para que a regra MAP símbolo a símbolo possa ser aplicada de modo ótimo. Deste modo é suposto que todos os percursos começam e

terminam no estado zero. Assim as recursões progressiva e regressiva são inicializadas com $\alpha_0(0) = 1$ e $\beta_N(0) = 1$, onde N é o tamanho da palavra-informação. Sempre que houver transição entre os estados $s^{'}$ e s, a probabilidade de transição deste ramo será dada pela EQ 9.19.

$$\gamma_i(s^{'}, s) = p(y_i|u_i) \cdot P(u_i) \tag{9.19}$$

Usando-se as probabilidades logaritmicas, a probabilidade *a priori* $P(u_i)$ é expressa pela EQ 9.20.

$$\begin{aligned} P(u_i = \pm 1) &= \frac{e^{\pm L(u_i)}}{1 + e^{\pm L(u_i)}} = \left(\frac{e^{-L(u_i)/2}}{1 + e^{-L(u_i)}} \right) \cdot e^{L(u_i)u_i/2} \\ &= A_i \cdot e^{L(u_i)u_i/2} \end{aligned} \tag{9.20}$$

onde:

$$L(u_i) = \log \frac{P(u_i = +1)}{P(u_i = -1)} \tag{9.21}$$

De modo semelhante, o probabilidade condicional $p(y_i|u_i)$ para códigos convolucionais sistemáticos é expressa pela EQ 9.22.

$$p(y_i|u_i) = B_i \cdot \exp \left(\frac{1}{2} L_c y_{i,1} u_i + \frac{1}{2} \sum_{j=2}^{n} L_c y_{i,j} x_{i,j} \right) \tag{9.22}$$

onde:

$$L_c = 4a \cdot \frac{E_s}{N_0} \tag{9.23}$$

Para um canal com desvanecimento, a denota a amplitude de desvanecimento enquanto que para um canal Gaussiano $a = 1$ (HAGENAUER, 1996).

Para um canal BSC, L_c é dado pela EQ 9.24 que representa a razão de probabilidade logarítmica. L_c é denominado valor de confiabilidade do canal (HAGENAUER, 1996).

$$L_c = \log((1 - p)/p) \tag{9.24}$$

onde p é a probabilidade de transição do canal BSC.

Se houver puncionamento, o somatório da EQ 9.22 será apenas sobre os índices j correspondentes aos bits que não tiverem sido puncionados. Os termos A_i e B_i na EQ 9.20 e EQ 9.22 são iguais para todas as transições do nível $i-1$ para o nível i e portanto irão ser canceladas na razão da EQ 9.15. Portanto, a operação de transição de ramo usada na EQ 9.17 e EQ 9.18 é reduzida ao cálculo da EQ 9.25.

$$\exp\left(\frac{1}{2}(L_c y_{i,1} u_i + L(u_i))\right) \cdot \gamma_i^{(e)}(s^{'}, s) \tag{9.25}$$

onde

$$\gamma_i^{(e)}(s^{'}, s) = \exp\left(\frac{1}{2}\sum_{j=2}^{n} L_c y_{i,j} x_{i,j}\right) \tag{9.26}$$

Como a função exponencial na EQ 9.25 é comum em todos os termos das somas na EQ 9.15, é possível dividir todos os termos por aquele e obter a EQ 9.27.

$$L(\hat{u}_i) = L_c y_{i,1} + L(u_i) + L_e(\hat{u}_i) \tag{9.27}$$

onde:

$$L_e(\hat{u}_i) = \log \frac{\sum_{S^+} \gamma_i^{(e)}(s^{'}, s) \cdot \alpha_{i-1}(s^{'}) \cdot \beta_i(s)}{\sum_{S^-} \gamma_i^{(e)}(s^{'}, s) \cdot \alpha_{i-1}(s^{'}) \cdot \beta_i(s)} \tag{9.28}$$

9.2.3 DESEMPENHO

Nesta seção serão apresentadas as principais funções enumeradoras de peso de um código convolucional que em seguida serão usadas para prever o desempenho destes códigos em canal AWGN.

A função enumeradora de pesos WEF (*Weight Enumerating Function*) é importante na análise de desempenho de um codificador pois sintetiza suas principais propriedades estruturais.

Definição 9.18 (WEF) *É uma função cujos argumentos são variáveis do BCE e os coeficientes denotam o número de palavras-código, blocos de informação ou blocos de paridade de pesos particulares.*

Para um BCC de tamanho N, a WEF associada possue a forma da EQ 9.29.

$$A(X) = \sum_{i=0}^{N} a_i X^i \tag{9.29}$$

onde a_i corresponde ao número de palavras-código de peso i.

Como o tamanho das palavras-código de um BCC pode ser infinito, a WEF associada possui a forma da EQ 9.30.

$$A^{'}(X) = \sum_{i=0}^{\infty} a_i X^i \tag{9.30}$$

Apesar da WEF ser determinada a partir de um dado BCE, a WEF vale para qualquer BCE que gere o mesmo BCC.

Entre as WEF mais importantes estão a função enumeradora dos pesos de entrada e de redundância IRWEF (*Input-Redundancy Weight Enumerating Function*) usada em codificadores sistemáticos e a função enumeradora dos pesos de entrada e de saída IOWEF (*Input- Output Weight Enumerating Function*) usada em codificadores não-sistemáticos.

Definição 9.19 (IRWEF) *É a função WEF cujos argumentos são o peso da informação e do bloco de paridade. No caso de codificadores sistemáticos, esta função pode ser usada para separar os pesos de palavras-código em peso de informação e peso da redundância.*

A sua forma geral é dada pela EQ 9.31.

$$A(W,Z) = \sum_{w=0}^{K} \sum_{z=0}^{N-K} a_{w,z} W^w Z^z \tag{9.31}$$

onde $a_{w,z}$ é o número de palavras-código com bloco de informação de peso w e bloco de paridade de peso z.

Entretanto, o peso total da palavra-código é $i = w+z$. Desta forma, os coeficientes da função WEF podem ser calculados a partir dos coeficientes da função IRWEF, conforme EQ 9.32.

$$a_i = \sum_{w=1}^{i} a_{w,i-w} \tag{9.32}$$

Na análise de desempenho, é sempre importante grupar os termos da função IRWEF de acordo com o peso do bloco de informação.

$$A_w(Z) = \sum_{z=0}^{N-K} a_{w,z} Z^z \tag{9.33}$$

As relações entre $A_w(Z)$ e $A(W,Z)$ são dadas pela EQ 9.35 e EQ 9.41

$$A(W,Z) = \sum_{w=0}^{K} A_w(Z) W^w \tag{9.34}$$

$$A_w(Z) = \frac{1}{w!} \cdot \left. \frac{\partial^w A(W,Z)}{\partial W^w} \right|_{W=0} \tag{9.35}$$

Exemplo 9.1 (IRWEF para IIR BCE) *A FIG. 9.9 mostra o grafo orientado do codificador recursivo sistemático da FIG. 9.7. Sua IRWEF é dada por:*

$$\begin{aligned} A'(W,Z) &= \frac{W^2Z^4 + W^3Z^2 - W^4Z^2}{1 - 2W - Z^2 + W^2} \\ &= W^3Z^2 + W^2Z^4 + W^4Z^2 + \cdots \end{aligned}$$

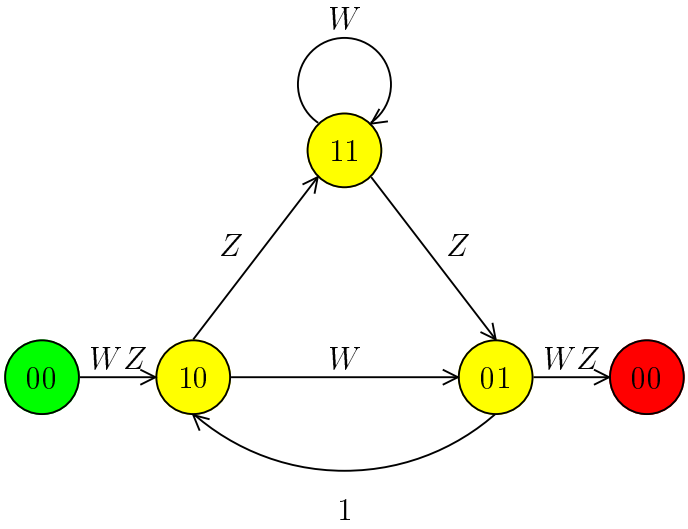


FIG. 9.9: Grafo orientado de um BCE IIR

Definição 9.20 (IOWEF) *É a função WEF cujos argumentos são o peso da informação e o peso da palavra-código.*

A forma geral da IOWEF é dada pela EQ 9.36.

$$A(W,X) = \sum_{w=0}^{K} \sum_{x=0}^{N} a_{w,x} W^w X^x \tag{9.36}$$

onde $a_{w,x}$ é o número de palavras-código com bloco de informação de peso w e palavra-código de peso x.

Exemplo 9.2 (IOWEF para FIR BCE) *A FIG. 9.10 mostra o grafo orientado do codificador recursivo não-sistemático da FIG. 9.6. Sua IOWEF é dada por:*

$$\begin{aligned} A(W,X) &= \frac{WX^5}{(1-WX)^2 - W^2X^2} \\ &= WX^5 + 2W^2X^6 + 4W^3X^7 \\ &+ 8W^4X^8 + 16W^5X^9 + 32W^6X^{10} + \cdots \end{aligned}$$

Exemplo 9.3 (IOWEF para IIR BCE) *A FIG. 9.11 mostra o grafo orientado do codificador recursivo sistemático da FIG. 9.7. Sua IOWEF é dada por:*

$$\begin{aligned} A(W,X) &= \frac{W^2X^6 + W^3X^5(1-WX)}{(1-WX)^2 - X^2} \\ &= W^3X^5 + (W^2 + W^4)X^6 + (3W^3 + W^5)X^7 \\ &+ (W^2 + 6W^4 + W^6)X^8 + (5W^3 + 10W^5 + W^7)X^9 \\ &+ (W^2 + 15W^4 + 15W^6 + W^8)X^{10} + \cdots \end{aligned}$$

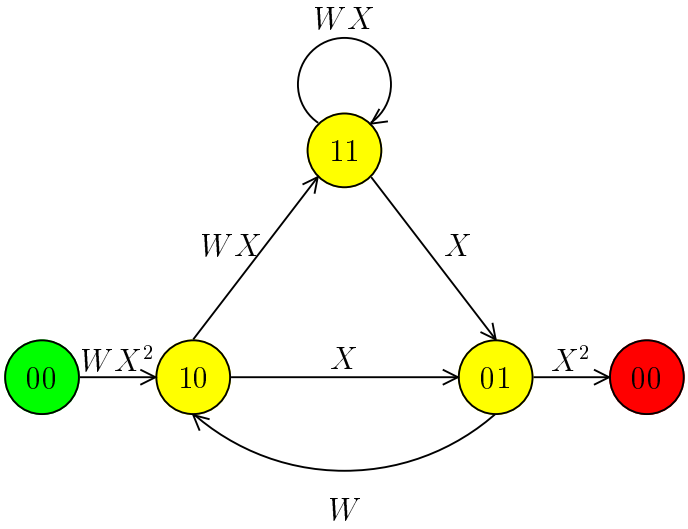


FIG. 9.10: Grafo orientado de um BCE FIR

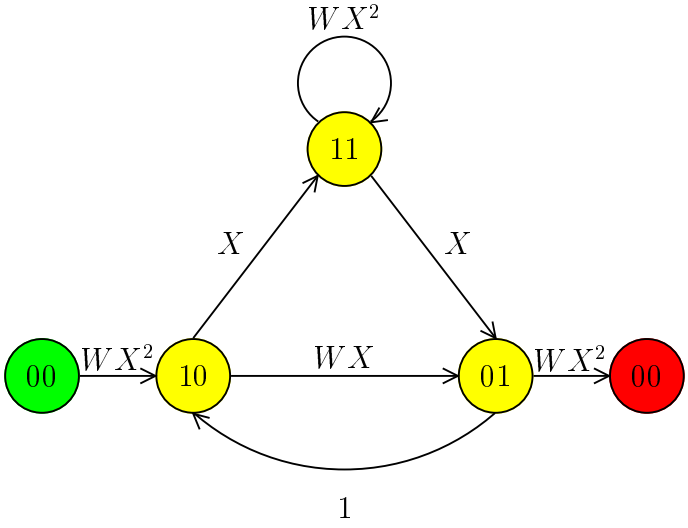


FIG. 9.11: Grafo orientado de um BCE IIR

Para fins de análise de desempenho considera-se um sistema que use a modulação BPSK em canal AWGN. O processo de modulação e transmissão requer que as palavras-código binárias sejam mapeadas em seqüências de valores reais. O mapeamento usado é $\{0,1\} \longmapsto \{1,-1\}$.

Seja $\mathbf{x} = (x_0, x_1, \cdots, x_{N-1})$ a seqüência transmitida e $\mathbf{y} = (y_0, y_1, \cdots, y_{N-1})$ a seqüência recebida. A distância Euclidiana entre estas seqüências é dada pela EQ 9.37.

$$d_E(\mathbf{y}, \mathbf{x}) = \sqrt{\sum_{j=0}^{N-1} (y_j - x_j)^2} \tag{9.37}$$

Devido ao mapeamento utilizado, a relação entre a distância de Hamming e a Euclidiana é expressa pela EQ 9.38.

$$d_E(\mathbf{x_1}, \mathbf{x_2}) = 2\sqrt{d_H(\mathbf{x_1}, \mathbf{x_2})} \tag{9.38}$$

A análise de desempenho começa com a determinação da probabilidade de erro par a par assumindo que a palavra-código **0** tenha sido transmitida. A probabilidade P_d de o decodificador selecionar uma palavra-código $\mathbf{c}$ de peso d em detrimento da palavra **0** é expressa pela EQ 9.39.

$$P_d = Pr\left[\sum_{j=0}^{d-1} y_j \leq 0\right] \tag{9.39}$$

Seja E_b a energia por bit de informação, E_{bc} a energia por bit codificado e $R_c = k/n$ a taxa de código. Tem-se que $E_{bc} = R_c E_b$ e y_j possui uma função densidade de probabilidade condicional Gaussiana de média $\mu = \sqrt{R_c E_b}$ e variância $\sigma^2 = N_0/2$ dada pela EQ 9.40.

$$p(y_j|x_j = 1) = \frac{1}{\sqrt{\pi N_0}} e^{-(y_j - \sqrt{R_c E_b})^2/N_0} \tag{9.40}$$

A soma de d variáveis Gaussianas i.i.d. é também uma variável Gaussiana mas com média $d\mu$ e e variância $d\sigma^2$. P_d pode agora ser calculado usando a função erro $Q(x)$ [3].

$$P_d = P\left\{\sum_{j=0}^{d-1} y_j \leq 0\right\} = Q(\sqrt{2dR_cE_b/N_0}) \tag{9.41}$$

Usando o limitante da união, a probabilidade de erro do decodificador para um código de tamanho N será majorada da forma mostrada pela EQ 9.42.

$$P_e \leq \sum_{d=d_{free}^H}^{N} a_d P_d \tag{9.42}$$

onde a_d é o número de palavras-código de peso d obtido a partir da WEF $A(X)$.

Para obter P_d como potência de d, EQ 9.44, usa-se a relação expressa pela EQ 9.43.

$$Q(\sqrt{x+y}) \leq Q(\sqrt{y}) e^{-x/2} \tag{9.43}$$

$$\begin{aligned} P_d &= Q(\sqrt{2dR_cE_b/N_0}) \\ &= Q(\sqrt{2d_{free}^H R_c E_b/N_0 + 2(d - d_{free}^H)R_c E_b/N_0}) \end{aligned} \tag{9.44}$$

[3] $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-y^2}{2}}\, dy$

$$
\begin{aligned}
&\leq Q(\sqrt{2d_{free}^H R_c E_b/N_0}) e^{-(d-d_{free}^H)R_c E_b/N_0} \\
&= e^{d_{free}^H R_c E_b/N_0} Q(\sqrt{2d_{free}^H R_c E_b/N_0}) (e^{-R_c E_b/N_0})^d
\end{aligned}
$$

Substituindo a EQ 9.44 na EQ 9.42 obtem-se a EQ 9.45, limitante superior da probabilidade de erro do decodificador.

$$
P_e \leq e^{d_{free}^H R_c E_b/N_0} Q(\sqrt{2d_{free}^H R_c E_b/N_0}) \sum_{d=d_{free}^H}^{N} a_d (e^{-R_c E_b/N_0})^d \tag{9.45}
$$

$$
= e^{d_{free}^H R_c E_b/N_0} Q(\sqrt{2d_{free}^H R_c E_b/N_0}) A(X)|_{X=e^{-R_c E_b/N_0}} \tag{9.46}
$$

A partir da EQ 9.45, pode ser obtido também um limitante superior para a probabilidade de erro de bit. Para isto suponha que a palavra-código $\mathbf{c}$ de tamanho N possua um bloco de informação de peso w e um bloco de paridade de peso z. Logo o evento escolha da palavra-código $\mathbf{c}$ em detrimento de $\mathbf{0}$ causará w erros em um total de K bits de informação. A probabilidade par a par associada a este evento é calculada usando a EQ 9.45 calculando as derivadas parciais da IRWEF nos pontos $W = Z = e^{-R_c E_b/N_0}$ onde $d = w + z$. Assim obtém-se a EQ 9.47.

$$
P_b(e) \leq \frac{1}{K} e^{d_{free}^H R_c E_b/N_0} Q\left(\sqrt{2d_{free}^H R_c E_b/N_0}\right) W \left.\frac{\partial A(W,Z)}{\partial W}\right|_{W=Z=e^{-R_c E_b/N_0}} \tag{9.47}
$$

Substituindo-se a EQ 9.34 na EQ 9.47 obtem-se a EQ 9.48.

$$
\begin{aligned}
P_b(e) &\leq \frac{1}{K} e^{d_{free}^H R_c E_b/N_0} Q\left(\sqrt{2d_{free}^H R_c E_b/N_0}\right) W \left.\frac{\partial \sum_{w=0}^{K} W^w A_w(Z)}{\partial W}\right|_{W=Z=e^{-R_c E_b/N_0}} \qquad (9.48) \\
&= \frac{1}{K} e^{d_{free}^H R_c E_b/N_0} Q\left(\sqrt{2d_{free}^H R_c E_b/N_0}\right) \left(\sum_{w=1}^{K} \sum_{z=0}^{N-K} w a_{w,z} W^w Z^z\right)\Bigg|_{W=Z=e^{-R_c E_b/N_0}}
\end{aligned}
$$

O termo $w a_{w,z} W^w Z^z$ na EQ 9.48 representa a contribuição para o BER de todas as palavras-código com bloco de informação de peso w e paridade de peso z.

Quando for necessário comparar vários BCE, será necessário isolar a contribuição ocasionada pelas a_d palavras-código de peso d. Deste modo, define-se:

$$
w_d \equiv \sum_{j=1}^{d} j a_{j,d-j} \tag{9.49}
$$

onde w_d é o número total de informação (bits de informação não nulos) computados considerando todas palavras-código de peso d.

Fazendo a mudança de variável $z = (d - w)$ no limitante de $P_b(e)$, tem-se a EQ 9.50.

$$P_b(e) \leq \frac{1}{K} e^{d^H_{free} R_c E_b/N_0} Q\left(\sqrt{2 d^H_{free} R_c E_b/N_0}\right) \left(\sum_{d=d^H_{free}}^{N} \sum_{w=1}^{d} w a_{w,d-w} W^w Z^{d-w} \right) \Bigg|_{W=Z=e^{-R_c E_b/N_0}} \quad (9.50)$$

$$= \frac{1}{K} e^{d^H_{free} R_c E_b/N_0} Q\left(\sqrt{2 d^H_{free} R_c E_b/N_0}\right) \sum_{d=d^H_{free}}^{N} w_d X^d \Bigg|_{W=Z=e^{-R_c E_b/N_0}} \quad (9.51)$$

A EQ 9.50 é muito útil para projeto de códigos. Para estabelecer o desempenho de um código é necessário saber como influenciam o BER as palavras-código de baixo e alto peso.

Usando as ferramentas previamente desenvolvidas é possível analisar o desempenho de vários tipos de BCE para SNR alto e baixo:.

- **SNR alto**: contribuição será maior das seqüências de baixo peso, ou seja, mais próximas da seqüência transmitida. Deste modo, se d^H_{free} for aumentada, o desempenho em alto SNR será melhor. Para uma dada taxa e ordem de memória, é conhecido que o d^H_{free} de códigos não-sistemáticos é maior do que o dos sistemáticos FIR

- **SNR baixo**: seqüências com peso maior que d^H_{free} irão contribuir mais para o BER como um todo. De fato, as palavras-código de peso alto poderão contribuir mais que as de baixo devido ao maior número das primeiras.

Desta forma, os melhores codificadores são aqueles cujos coeficientes da WEF começam pequenos e crescem lentamente. Há casos em que um código com d^H_{free} menor tem um desempenho melhor que um com d^H_{free} maior dependendo de sua WEF.

Camparando dois codificadores mínimos, um sistemático IIR e outro não-sistemático FIR, observa-se que normalmente o N-FIR em relação ao S-IIR tem desempenho superior em altas energias e inferior em baixas. Tal fato é atribuido ao crescimento mais lento da seqüência w_d para um codificador S-IIR. O exemplo a seguir ilustra esta situação.

Exemplo 9.4 (Desempenho N-FIR $\times$ S-IIR) *A IOWEF do código N-FIR da FIG. 9.6 é dada por:*

$$A(W, X) = \frac{WX^5}{(1 - WX)^2 - W^2X^2}$$

$$
\begin{aligned}
&= WX^5 + 2W^2X^6 + 4W^3X^7 \\
&+ 8W^4X^8 + 16W^5X^9 + 32W^6X^{10} + \cdots
\end{aligned}
$$

enquanto a IOWEF do código S-IIR da FIG. 9.7 é dada por:

$$
\begin{aligned}
A(W,X) &= \frac{W^2X^6 + W^3X^5(1-WX)}{(1-WX)^2 - X^2} \\
&= W^3X^5 + (W^2 + W^4)X^6 + (3W^3 + W^5)X^7 \\
&+ (W^2 + 6W^4 + W^6)X^8 + (5W^3 + 10W^5 + W^7)X^9 \\
&+ (W^2 + 15W^4 + 15W^6 + W^8)X^{10} + \cdots
\end{aligned}
$$

Comparando os codificadores vê-se que ambos possuem $d^H_{free} = 5$ que é a máxima possível para codificadores com ordem de memória $M = 2$ (COSTELLO, 1994). Como mostrado, ambos codificadores geram os mesmos números de palavras-código de peso 5 a 10. Para SNR alta vê-se que o código não-sistemático leva vantagem em relação ao sistemático por associar palavras-informação de peso 1 a palavras- código de peso 5. Para SNR baixo, o sistemático leva vantagem pois as palavras-código de peso 9 em diante têm palavras de informação associadas de peso total menor em comparação ao código não-sistemático.

9.3 APÊNDICE 3: CÓDIGOS BCH-RS

Os códigos BCH (*Bose-Chaudhuri-Hocquenghem*) e RS (*Reed-Solomon*) são códigos de interesse prático no processo de controle de erros. Como exemplo desta importância, os códigos RS são usados em praticamente todas as propostas de transmissão de TV digital na Europa e nos Estados Unidos.

Neste capítulo, serão introduzidos os conceitos necessários à geração e uso destes códigos para a recuperação de erros. A teoria destes códigos depende muito de operações aritméticas realizadas em corpos com um número finito de elementos. Estes corpos são denominados Corpos de Galois. Deste modo, será feita antes uma explanação sobre aritmética em corpos finitos.

9.3.1 CORPO DE GALOIS

Nesta seção serão introduzidos alguns conceitos sobre aritmética em corpos finitos.

9.3.1.1 GRUPO

Seja a seguinte definição algébrica de um grupo.

Definição 9.21 *Um grupo, representado por* $(G, \bigcirc)$*, é um conjunto não vazio* G *munido da operação* $\bigcirc$ *e fechado em relação a ela. Um grupo possui as seguintes propriedades:*

G1. ***Associativa:*** $(a \bigcirc b) \bigcirc c = a \bigcirc (b \bigcirc c), \forall a, b, c \in G$

G2. ***Elemento neutro*** *e:* $\exists e \in G \mid a \bigcirc e = e \bigcirc a = a, \forall a \in G$

G3. ***Elemento inverso:*** $\forall a \in G, \exists a^{'} \in G \mid a \bigcirc a^{'} = a^{'} \bigcirc a = e$

Um grupo é ainda denominado ***comutativo*** *ou* ***abeliano*** *se possui também a seguinte propriedade:*

G4. ***Comutativa:*** $a \bigcirc b = b \bigcirc a, \forall a, b \in G$

O número de elementos de um grupo define sua ordem, $ord(G)$. A ordem de um grupo pode ser finita ou infinita. Quanto for infinita, poderá ser contável ou não contável. Alguns exemplos de grupos são $(\mathbb{Z}, + \, mod\, m)$, $(\mathbb{R}^*, \cdot)$ e $(\mathbb{Z}, \cdot \, mod\, m)$.

9.3.1.2 ANEL

Definição 9.22 *Anel é um conjunto não vazio R munido de duas operações $\oplus$ e $\odot$. Um anel possui as seguintes propriedades:*

R1. *R constitui um grupo comutativo sob $\oplus$, ou seja, forma o grupo $(R, \oplus)$*

R2. *A operação $\odot$ é associativa, ou seja, $(a \odot b) \odot c = a \odot (b \odot c), \forall a, b, c \in R$*

R3. *A operação $\odot$ é distributiva sobre $\oplus$, ou seja, $a \odot (b \oplus c) = a \odot b \oplus a \odot c, \forall a, b, c \in R$*

Adicionalmente, um anel ainda pode ter as seguintes propriedades:

R4. *R é um anel comutativo se a operação $\odot$ comuta, ou seja, $a \odot b = b \odot a, \forall a, b \in G$*

R5. *R é um anel com identidade se a operação $\odot$ possui um elemento identidade e pertencente a R, ou seja, $\exists e \in R \mid a \odot e = e \odot a = a, \forall a \in G$*

Outro conceito importante para a definição de códigos cíclicos é o de ideal principal em um anel.

Definição 9.23 *Seja R um anel. Um subconjunto não vazio $I \subseteq R$ é dito ser um ideal principal se satisfaz:*

I1. *I forma um grupo sob a operação de adição em R.*

I2. *$a \cdot r = b \in I$ para todo $a \in I$ e todo $r \in R$.*

I3. *Existe $g \in I$ tal que todo elemento $c \in I$ pode ser expresso como o produto $m \cdot g$, $m \in R$. Este elemento g é denominado elemento gerador e o ideal gerado por ele é denotado por $\langle g \rangle$.*

9.3.1.3 CORPO

Definição 9.24 *Corpo é um conjunto de elementos F munido de duas operações binárias $\oplus$ e $\odot$. Um corpo possui as seguintes propriedades:*

F1. *F constitui um grupo comutativo sob $\oplus$, ou seja, é o grupo $(F, \oplus)$.*

F2. *F - $\{0\}$ constitui um grupo comutativo sob $\odot$, ou seja, é o grupo $(F - \{0\}, \odot)$.*

F3. *A operação $\odot$ é distributiva sobre $\oplus$, ou seja, $a \odot (b \oplus c) = (a \odot b) \oplus (a \odot c), \forall a, b, c \in F$*

Um exemplo de corpos de Galois são os conjuntos de inteiros de ordem ímpar da forma $\{0, 1, 2, ..., p-1\}$ sob as operações de adição e multiplicação módulo p, onde p é primo. Estes conjuntos são indicados pela notação $GF(p)$. O conjuntos $GF(p^m)$, m inteiro, é considerado um espaço vetorial gerado a partir de $GF(p)$.

9.3.1.4 POLINÔMIOS SOBRE CORPOS DE GALOIS

Uma estrutura matemática muito importante para definição de corpos são os polinômios $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m$ cujos coeficientes pertencem a um corpo $GF(p)$. A notação empregada para representar um conjunto destes polinômios é $GF(p)[x]$. Pode ser demonstrado que $GF(p)[x]$ juntamente com as operações soma e produto, define um anel comutativo com identidade.

$GF(p)[x]$ pode ser transformado em um corpo de Galois limitando o grau dos polinômios de $GF(p)[x]$. Um dos modos de efetuar esta transformação é definir a operação multiplicação $\odot$ como sendo multiplicação módulo $g(x)$ onde $g(x)$ é um polinômio irredutível de grau n em $GF(p)$. Deste modo, tem-se a expressão 9.52.

$$f_1(x) \odot f_2(x) = [f_1(x) \times f_2(x)] \text{ módulo } g(x) \tag{9.52}$$

O corpo de Galois criado desta forma é denominado corpo de polinômios sobre $GF(p)$ módulo $g(x)$. O número de elementos deste GF é igual a p^n onde p é a ordem do corpo de símbolos $GF(p)$ e n é o grau do polinômio irredutível $g(x)$. Usando este procedimento, é possível construir corpos de Galois tendo p^n elementos a partir de $GF(p)$. É importante observar que para criar o corpo estendido $GF(p^n)$, não é necessário que a cardinalidade do corpo seja prima. Esta ordem também pode ser da forma $q = p^n$.

A determinação de $g(x)$ envolve o conhecimento dos conceitos de ordem de elemento, elemento primitivo, polinômio irredutível e de polinômio primitivo em corpos finitos dados a seguir.

Definição 9.25 (Ordem de um elemento de um corpo de Galois) *Seja β um elemento em $GF(q)$. A ordem de β, $ord(\beta)$ é o menor inteiro positivo m tal que $\beta^m = 1$.*

Definição 9.26 (Elemento primitivo em um corpo de Galois) *Um elemento α com ordem $(q-1)$ em $GF(q)$ é denominado elemento primitivo em $GF(q)$.*

Definição 9.27 (Polinômio irredutível em um corpo de Galois) *Um polinômio $f(x)$ é irredutível em $GF(q)$ se $f(x)$ não pode ser fatorado em um produto de polinômios de menor grau em $GF(q)[x]$.*

Definição 9.28 (Polinômio mínimo) *Seja α um elemento pertencente ao corpo $GF(q^m)$. O polinômio mínimo de α com respeito a $GF(q)$, $m_\alpha(x) \in GF(q)[x]$, é o polinômio não-nulo de menor grau pertencente a $GF(q)[x]$ tal que $m_\alpha(\alpha) = 0$.*

Definição 9.29 (Polinômio primitivo) *Um polinômio irredutível $f(x) \in GF(q)[x]$ de grau m é dito primitivo se o menor inteiro positivo n para o qual $f(x)$ divide $x^n - 1$ é $n = q^m - 1$.*

Quanto à definição de polinômio primitivo, pode ser demonstrado que:

- Qualquer polinômio irredutível $f(x) \in GF(q)[x]$ de grau m deve dividir $x^{q^m-1} - 1$ (MCELIECE, 1987).
- O polinômio mínimo associado a elemento primitivo $\alpha \in GF(q^m)$ tem sempre grau m (WICKER, 1995).
- Existem $\phi(2^n - 1)/n$ polinômios binários de grau n (DORNHOFF, 1980).

Desta forma, um corpo de Galois $GF(q^m)$ pode ser construído por meio de um corpo de polinômios módulo um polinômio primitivo de grau m. Este isomorfismo garante que cada elemento $\alpha^i \in GF(q^m)$ possa ser representado por um polinômio de grau menor que m. A correspondência é dada pela EQ 9.53.

$$x^i \, mod \, g(x) \longleftrightarrow \alpha^i \tag{9.53}$$

onde:

α	é elemento primitivo pertencente a $GF(q^m)$
$g(x)$	é polinômio primitivo
i	grau em x, variando de 0 a $q^m - 2$

Um algoritmo muito simples para buscar um polinômio primitivo seria achar todos os polinômios irredutíveis de grau m e calcular $x^i \, mod \, g(x)$, $(i = 0 \cdots q^m - 2)$. Se forem achados $q^m - 2$ diferentes resultados, $g(x)$ é primitivo.

9.3.2 CÓDIGOS EM BLOCO CÍCLICOS LINEARES

Um código em bloco linear $\mathcal{C}$ (n, k) é dito cíclico se para toda palavra-código $\mathbf{c} = (c_0, \cdots, c_{n-1}) \in \mathcal{C}$, há também uma palavra-código $\mathbf{c} = (c_{n-1}, \cdots, c_0) \in \mathcal{C}$, $c_i \in GF(q)$.

A chave para o entendimento da estrutura dos códigos cíclicos está na associação de um polinômio código $c(x)$ com cada palavra-código de forma que $c(x) = c_0 + \cdots +$

$c_{n-1}x^{n-1}$. A partir desta associação pode ser demonstrado que os polinômios código em $\mathcal{C}$ pertencem a um ideal principal em anéis da forma $GF(q)[x]/x^n - 1$ e que possuem as seguintes propriedades:

1. Dentro do conjunto de polinômios em $\mathcal{C}$, há um único polinômio g(x) com grau mínimo $r \leq n$. $g(x)$ é denominado polinômio gerador de $\mathcal{C}$.

2. Cada polinômio código $c(x) \in \mathcal{C}$ pode ser expresso unicamente como $c(x) = m(x)\ g(x)$, onde $g(x)$ é o polinômio gerador de $\mathcal{C}$ e $m(x)$ é polinômio de grau menor que $(n - r)$ em $GF(q)[x]$.

3. O polinômio gerador $g(x)$ de $\mathcal{C}$ é fator de $x^n - 1$ em $GF(q)[x]$.

9.3.3 CODIFICAÇÃO

Conforme pode ser visto no APÊNDICE 1, a distância mínima de um código dá uma ideia das possibilidades de correção de um código. Os códigos RS são códigos que maximizam a distância mínima, $d_{min} = 2t + 1$, onde t é a quantidade de erros que podem ser corrigidos. Esta propriedades dos códigos RS é conhecida como MDS (Maximun Distance Separable). Já na família dos códigos BCH, d_{min} é igual ou próximo de $\delta = 2t + 1$, conhecido como limitante BCH.

A codificação BCH é bastante simples pois é necessario apenas multiplicar o polinômio informação pelo polinômio gerador em um dado corpo. A dificuldade dos códigos BCH está em determinar o polinômio gerador que é equivalente a construir o código dado a quantidade de erros a serem corrigidos. Para construir um código BCH q-ário de tamanho n para corrigir t erros são necessários os seguintes passos:

1. Encontrar a n-ésima raiz primitiva da unidade α no corpo $GF(q^m)$, onde m é mínimo.

2. Selecionar $(\delta - 1) = 2t$ potências consecutivas de α, começando com α^b, $b > 0$.

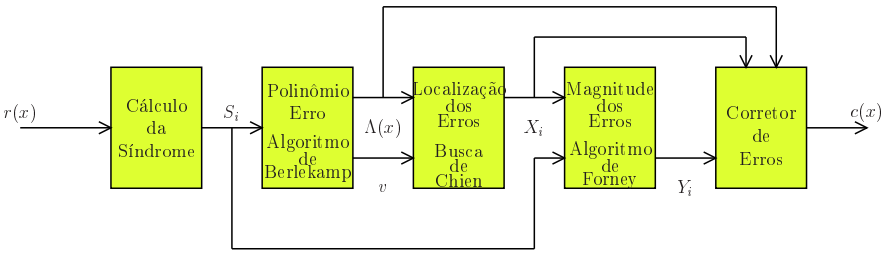
3. O polinômio gerador $g(x)$ será o menor múltiplo comum dos polinômios mínimos associados às potencias selecionadas de α com respeito a $GF(q)$.(Cada polinômio mínimo deverá aparecer apenas uma vez no produto)

Os códigos RS podem ser vistos como um caso particular de códigos BCH conforme definição 9.30.

Definição 9.30 (Código Reed-Solomon) *Um código RS é um código BCH q^m-ário de tamanho $q^m - 1$.*

9.3.4 DECODIFICAÇÃO

A decodificação de códigos BCH/RS envolve uma serie de passos. Todos os passos estão ilustardos no diagrama de fluxo da FIG. 9.12.



Onde:
$r(x)$ = palavra-código recebida
S_i = Síndromes
$\Lambda(x)$ = Polinômio localizador de erros
X_i = Posições dos erros
Y_i = Magnitudes dos erros
$c(x)$ = palavra de informação recuperada
v = número de erros

FIG. 9.12: Diagrama de decodificação de códigos BCH-RS

Nas seções que seguem serão descritos os passos

9.3.4.1 CÁLCULO DAS SÍNDROMES

Assim como na vida real, as síndromes dão indicação de alguma coisa errada. No caso de transmissão de dados, a síndrome é usada para indicar a ocorrência de erros.

Suponha que a palavra-código $c(x)$ foi transmitida e que foi recebido $r(x)$. Pode-se escrever $r(x) = c(x) + e(x)$, onde $e(x)$ representa o polinômio correspondente aos erros que ocorreram durante a transmissão. Sabendo que $c(x) = m(x)g(x)$ e que $g(\alpha^i) = 0$, $i = (0 \cdots 2t)$, é correto escrever que $r(\alpha^i) = e(\alpha^i)$ pois $c(\alpha^i) = 0$.

A partir desta propriedade tem-se o seguinte procedimento para checar a ocorrência de erros na recepção

1. Transformar a palavra-código recebida $\mathbf{r}$ ne sua representação polinomial $r(x)$.
2. Calcular as síndromes $S_i = r(\alpha^i) = e(\alpha^i)$ para $i = 0 \cdots 2t$.
3. Se uma ou mais síndromes não forem nulas, um ou mais erros de símbolos ocorreram no bloco.

9.3.4.2 POLINÔMIO LOCALIZADOR DE ERROS

Neste ponto, existe o conhecimento da ocorrência ou não de erros no bloco recebido. Entretanto, ainda não se sabe quantos símbolos foram afetados nem a localização dos erros.

A localização dos símbolos errados dentro de um bloco é determinado a partir da construção de um outro polinômio denominado polinômio localizador de erros $\Lambda(x)$. Dado então as $2t$ síndromes, o algoritmo de Berlekamp-Massey sintetiza um polinômio cujos valores dos coeficientes indicam as posições dos erros seguindo os seguintes passos:

1. Inicializacao: $k = 0$, $\Lambda^{(0)}(x) = 1$, $L = 0$ e $T(x) = x$
2. Incrementar k=k+1 e calcular $\Delta^{(k)} = S_k - \sum_{i=1}^{L} \Lambda_i^{(k-1)} S_{k-i}$
3. Se $\Delta^{(k)} = 0$, então passo 7
4. Calcule $\Lambda^{(k)}(x) = \Lambda^{(k-1)}(x) - \Delta^{(k)} T(x)$
5. Se $2L \geq k$, então passo 7
6. Faça $L = k - L$ e $T(x) = \Lambda^{(k-1)}(x)/\Delta^{(k)}$
7. Faça $T(x) = x \cdot T(x)$
8. Se $k < 2t$, então passo 2

9.3.4.3 LOCALIZAÇÃO DOS ERROS

Neste bloco, é preciso achar as raízes do polinômio localizador e invertê-las para determinar a localização dos erros.

Para calcular as raizes de $\Lambda(x)$ usa-se, no caso binário, a Busca de Chien (WICKER, 1995). Esta busca consiste basicamente na substituição no polinômio $\Lambda(x)$ de todas a potências do elemento primitivo $\alpha \in GF(2^m)$.

9.3.4.4 MAGNITUDE DOS ERROS

No caso de decodificação de códigos BCH binários é suficiente saber a localização dos erros no bloco. Entretanto, no caso de códigos BCH não-binários e códigos de RS, é necessário também conhecer a magnitude dos erros nas suas respectivas posições.

As magnitudes dos erros são calculadas utilizando-se do algoritmo de Forney (WICKER, 1995). A EQ 9.54 permite o cálculo das magnitudes dos erros.

$$e_{i_k} = \frac{-X_k \Omega(X_k^{-1})}{\Lambda'(X_k^{-1})} \tag{9.54}$$

onde:

X_k	são as raízes do polinômio localizador de erros
$\Omega(x)$	é igual a $\Lambda(x)[1 + S(x)]$
k	número de posições em que o vetor recebido foi corrompido, variando de $1 \cdots \nu$

9.3.4.5 CORREÇÃO DOS ERROS

Finalmente, para corrigir os erros no bloco recebido, monta-se o polinômio erro $e(x)$ com as informações sobre posição e magnitude. Em seguida, efetua-se a soma $c(x) = r(x) + e(x)$ para se obter o polinômio código transmitido e divide-se por $g(x)$ para se obter o polinômio informação.

A condição necessária para o uso de um protocolo com requisição automática de retransmissão ARQ (*Automatic-Repeat-reQuest*) é a existência de um canal com retorno. A função do retorno é trazer informação do receptor para o transmissor. Esta informação, uma vez disponível no transmissor, pode ser usada pelo transmissor para ajustar seus parâmetros de transmissão melhorando a recepção e o desempenho do sistema como um todo. Análises podem ser feitas considerando que o canal de retorno é com ou sem ruído.

A FIG. 9.13 ilustra um sistema ARQ onde uma informação lateral está disponível.

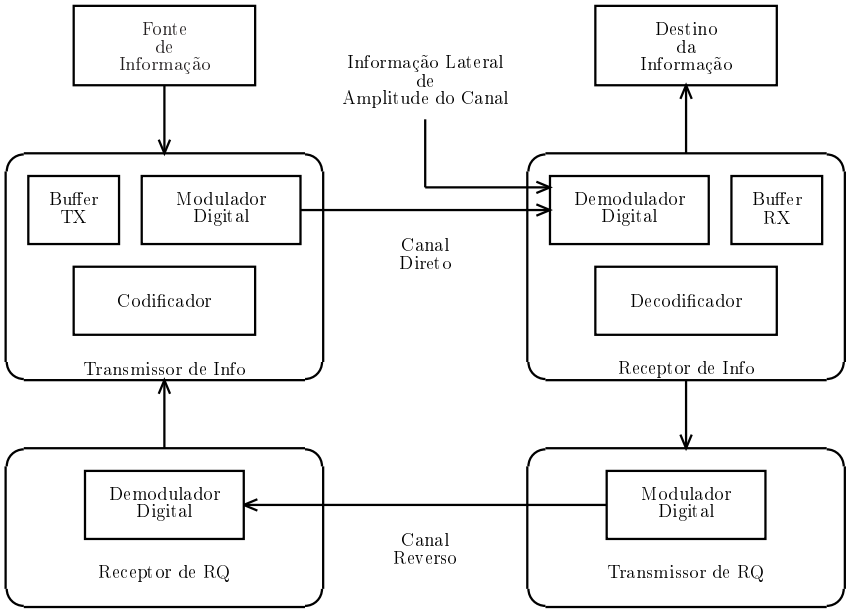


FIG. 9.13: Diagrama de blocos de um sistema com de controle de erros ARQ

Normalmente, o controle de erros para canais com retorno é implementado por meio de um protocolo ARQ. Nestes protocolos, os dados transmitidos são codificados para deteção de erros. Desta forma, será gerado pedido de retransmissão sempre que forem encontrados erros pelo receptor.

Os protocolos ARQ se dividem em dois grandes grupos:

- **Protocolos ARQ puro** são protocolos que fazem uso apenas de códigos detetores de erros. Os protocolos puros são basicamente de três tipos: SW (*Stop-and-Wait*), GBN (*Go-Back-N*) ou SR (*Selective-Repeat*)

- **Protocolos ARQ/FEC** são protocolos também denominados híbridos que se utilizam tanto de códigos para deteção de erro como para correção. Os protocolos híbridos podem ser de dois tipos: Tipo I e Tipo II.

Tomando como referência o modelo OSI, verifica-se que, naturalmente, os protocolos ARQ estão relacionados à camada de enlace pois esta é a primeira camada que cuida da comunicação ponto a ponto. O uso de um protocolo ARQ na camada de enlace é necessário sempre que a correção de erros FEC na camada física não for suficiente para garantir a confiabilidade requerida pela aplicação.

9.4.1 DESEMPENHO

Há basicamente dois parâmetros que medem o desempenho de um protocolo ARQ: a confiabilidade e a vazão. Em sistemas FEC, a confiabilidade é expressa em termos de taxa de erros de bits BER (*Bit Error Rate*) ou de símbolos SER (*Symbol Error Rate*). Já em sistemas ARQ, esta mesma confiabilidade é expressa em termos da probabilidade de aceitação de pacotes errados $P_w(A) = P_a$, onde A corresponde ao evento aceitar um elemento de informação estando este errado. Tanto a BER como a SER podem ser modeladas por variáveis aleatórias e, portanto podem ser expressas por meio das probabilidades $P_b(E)$ e $P_s(E)$, respectivamente. Neste caso, E representa o evento erro após o decodificador.

A probabilidade de aceitação de pacotes errados P_a é definida a seguir.

Definição 9.31 (Probabilidade de aceitação de pacotes errados) *É a porcentagem de pacotes aceitos pelo receptor que contém um ou mais erros de bits/símbolos.*

Antes de desenvolver uma expressão para a probabilidade de aceitação de pacotes errados é necessário introduzir alguns conceitos relacionados a recepção de um pacote. Cada vez que um pacote é recebido, pode ocorrer um dos cinco eventos esquematizados na FIG. 9.14. As probabilidades associadas a cada um dos eventos têm a seguinte notação:

P_c é a probabilidade de que o pacote recebido não possua erros (C_1) ou contenha um padrão de erro corrigível (C_2).

P_u é a probabilidade de que o pacote recebido possua um erro não detetável (U).

P_f é a probabilidade de que o pacote recebido possua um padrão de erro detetável mas não corrigível por falha de decodificação (F).

P_e é a probabilidade de que o pacote recebido possua um padrão de erro detetável mas não corrigível por erro de decodificação (E).

É elementar que $P_c + P_u + P_f + P_e = 1$.

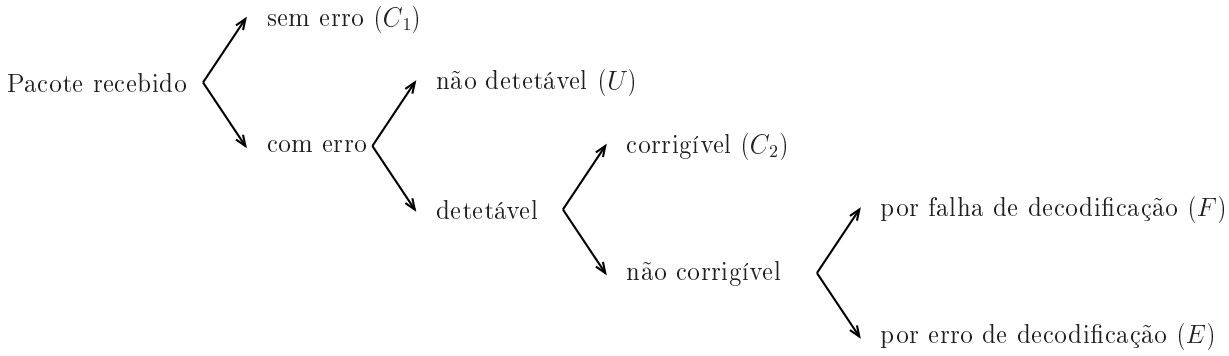


FIG. 9.14: Esquema com os possíveis eventos que podem ocorrer após a chegada de um pacote no receptor

Todo decodificador de um código $\mathcal{C}$ possui uma probabilidade não nula de cometer um erro de decodificação P_e ou de falhar na decodificação P_f. [4] No caso de protocolos ARQ, a probabilidade de um erro ser detetado e gerar um pedido de retransmissão P_r é igual a $P_e + P_f$, ou simplesmente, P_e, no caso de decodificador completo.

Para um protocolo ARQ com decodificador completo, a probabilidade de aceitação de pacotes errados pode ser facilmente calculada sabendo-se que um pacote errado somente é aceito caso contenha um padrão de erro não detetado. Desta forma, P_a é calculado somando as probabilidades de vários eventos que possam resultar na aceitação de um pacote errado. Assumindo que o retorno do canal é livre de erros, a EQ 9.55 permite calcular P_a.

$$P_a = P_u + P_r P_u + {P_r}^2 P_u + \cdots + {P_r}^k P_u + \cdots = P_u \sum_{k=0}^{\infty} P_r^k = \frac{P_u}{1 - P_r} \tag{9.55}$$

onde $P_r < 1$ garante a convergência da série.

Pode ser demonstrado que a probabilidade de aceitação de pacotes errados é independente da qualidade do canal de retorno. A EQ 9.55 mostra que mesmo que a probabilidade de erro não detetado seja fixa, a probabilidade de aceitação de um pacote errado aumenta com a probabilidade de pedido de retransmissão. Tal fato é explicado

[4] Um falha de decodificação acontece apenas quando o decodificador for do tipo incompleto como por exemplo no caso dos códigos BCH/RS

pelo fato de cada retransmissão criar uma nova oportunidade de que ocorra um erro não detetável.

Os melhores códigos detetores de erro (n, k) possuem o limitante superior para a probabilidade de erro (KASAMI, 1983) dado pela EQ 9.56.

$$P_u \leq 2^{-(n-k)}\{1 + (1-2p)^n - 2(1-p)^n\} \tag{9.56}$$

onde p é a probabilidade de transição de um canal BSC.

Como os melhores códigos detetores possuem restrições quanto ao tamanho da entrada do codificador, os códigos CRC são mais usados na prática ainda que sejam subótimos. Para qualquer código CRC de tamanho (n, k) sobre um canal BSC, a probabilidade de erro não detetado aproxima-se de $2^{-(n-k)}$ quando a probabilidade p e a dimensão k do código aumentam. Isto estabelece um limitante inferior do desempenho de um código CRC em um canal com ruído.

Desta forma, é possivel afirmar que a probabilidade de erro não detetado é quase sempre muito baixa e, consequentemente, a confiabilidade é alta. Isto faz com que a medida de desempenho de um protocolo de maior interesse seja a vazão definida a seguir.

Definição 9.32 (vazão) *A vazão η de um sistema com controle de erros ARQ e código (n, k) corresponde à razão entre k e o número médio de pacotes de código de tamanho n aceitos pelo receptor até receber um único pacote de informação de tamanho k.*

Em sistemas com controle de erros FEC, a vazão η é igual a taxa de código $R_c = k/n$.

9.4.2 PROTOCOLOS ARQ PUROS

A vazão de um protocolo ARQ é função do número de vezes que um pacote tem de ser transmitido antes de ser aceito pelo receptor. Seja T_r o número médio de vezes que um pacote tem de ser transmitido antes de ser aceito e P_r conhecido. A determinação de T_r é um problema de valor esperado e sua solução está expressa na EQ 9.57

$$T_r = (1-P_r) + 2P_r(1-P_r) + 3{P_r}^2(1-P_r) + \cdots + kP_r^{k-1}(1-P_r) + \cdots = \frac{1}{1-P_r} \tag{9.57}$$

Um fator que influencia diretamente no desempenho é a maneira como os pedidos de retransmissão serão gerenciados pelo transmissor e receptor. Esta gerência depende diretamente da existência ou não de armazenamento temporário no transmissor / receptor. Os três modos de realizar esta gerência em um protocolo ARQ puro são descritos a seguir.

9.4.2.1 SW-ARQ

No SW-ARQ, não há armazenamento temporário no transmissor e no receptor. Deste modo, o transmissor envia o pacote e fica esperando uma resposta do receptor. Esta resposta pode ser um confirmação de recepção ACK (*acknowledgment*) caso o pacote seja considerado sem erros ou um pedido de retransmissão RQ (*retransmission request*) caso o pacote contenha um padrão de erro detetável. A FIG. 9.15 ilustra este mecanismo.

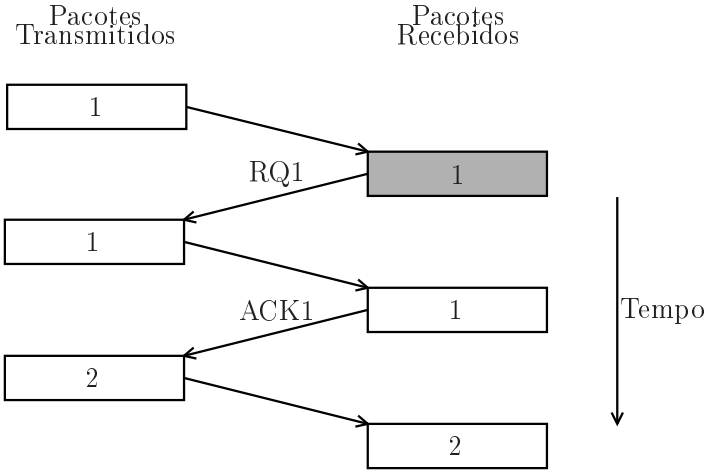


FIG. 9.15: Diagrama de estados do protocolo SW-ARQ

Analisando a FIG. 9.15, observa-se que o tempo que o transmissor fica esperando pelo ACK ou RQ do receptor é função do tempo necessário para o pacote com informação chegar ao receptor, λ_f, do tempo de processamento do receptor para saber se o pacote contem erros, λ_p, e do tempo para a resposta chegar ao transmissor, λ_b. O número de bits que poderiam ter sido transmitidos, Γ, durante o tempo de espera é expresso pela EQ 9.58.

$$\Gamma = R(\lambda_f + \lambda_p + \lambda_b) \tag{9.58}$$

onde R é a taxa de transmissão de bits.

Seja T_r o número de transmissões necessárias antes que o receptor aceite o pacote transmitido. Desta forma, cada pacote com k bits de informação necessita, em média, do tempo necessário para transmissão de $T_r(n + \Gamma)$ bits para ser aceito. A vazão do protocolo SW-ARQ será dada pela EQ 9.59.

$$\eta_{SW} = \frac{k}{T_r(n+\Gamma)} = R_c \left(\frac{1 - P_r}{1 + \Gamma/n} \right) \tag{9.59}$$

onde R_c é a taxa do código detetor de erro.

Observa-se que um protocolo SW-ARQ, possui baixa vazão em redes que possuam Γ elevado tais como redes que se utilizem de enlaces via satelite.

O cálculo da vazão em canais com retorno ruidoso requer algumas considerações a mais. Neste caso, um ACK pode se tornar um RQ, um RQ pode se tornar um ACK ou ainda a resposta pode nunca chegar ao destino. Deste modo, torna-se necessário estabelecer alguns temporizadores para, por exemplo, os seguintes eventos:

- **Transmissão de pacote de informação pelo transmissor**: cada vez que o transmissor envia um pacote, ele espera um tempo determinado para receber uma resposta do receptor. Expirando este tempo, normalmente é assumido que a resposta é um RQ.
- **Transmissão de um RQ pelo receptor**: se uma nova cópia do pacote não for recebida dentro de um tempo adequado, um RQ é enviado novamente.

Para facilitar o calculo da vazão do protocolo SW-ARQ com retorno ruidoso, utiliza-se o grafo orientado da FIG. 9.16 e a formula de Mason (WICKER, 1995). Na FIG. 9.16 o expoente da variável T representa o número de pacotes transmitidos com n bits associados à transição de estado.

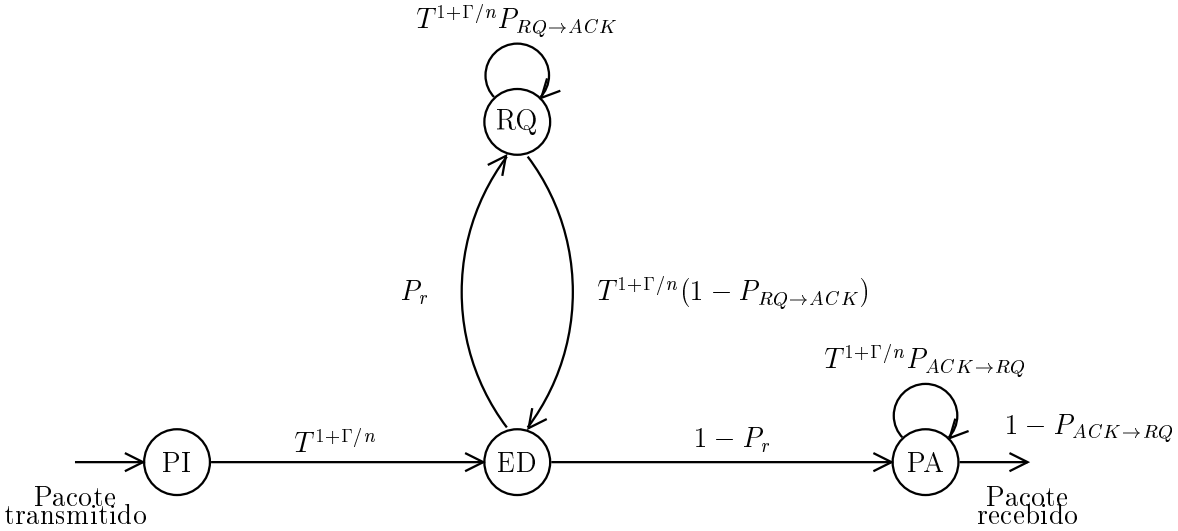


FIG. 9.16: Diagrama de estados do protocolo SW-ARQ com ruído no retorno

Segundo o grafo da FIG. 9.16, a máquina de estados do protocolo SW-ARQ possui os seguintes estados:

PI é o estado no qual a informação disponível é codificada para transmissão.

ED é o estado no qual o receptor deteta a ocorrência de erros no pacote recebido.

RQ é o estado no qual o receptor solicita retransmissão.

PA é o estado no qual o receptor aceita o pacote e envia um ACK. Cada nova copia de um pacote aceito é jogada fora e enviado um novo ACK.

A expressão 9.60 estabelece a vazão do protocolo SW-ARQ com retorno ruidoso (WICKER, 1995).

$$\eta_{SW} = R_c \left[\frac{(1-P_r)(1-P_{ACK\rightarrow RQ})(1-P_{RQ\rightarrow ARQ})}{(1+\Gamma/n)(1-P_rP_{ACK\rightarrow RQ}-P_{RQ\rightarrow ARQ}+P_rP_{RQ\rightarrow ARQ})} \right] \tag{9.60}$$

onde $P_{ACK\rightarrow RQ}$ e $P_{RQ\rightarrow ACK}$ são respectivamente as probabilidades de um ACK tornar-se um RQ e de um RQ tornar-se um ACK.

9.4.2.2 GBN-ARQ

Quando houver a disponibilidade de armazenamento temporário no transmissor, poderá ser usado o protocolo GBN-ARQ mostrado na FIG. 9.17. Neste protocolo, o transmissor envia os pacotes de modo contínuo. Deste modo, quando o receptor deteta um erro em um pacote recebido, ele envia um pedido de retransmissão e descarta todos os pacotes subsequentes até que uma segunda cópia do pacote com erro seja recebida.

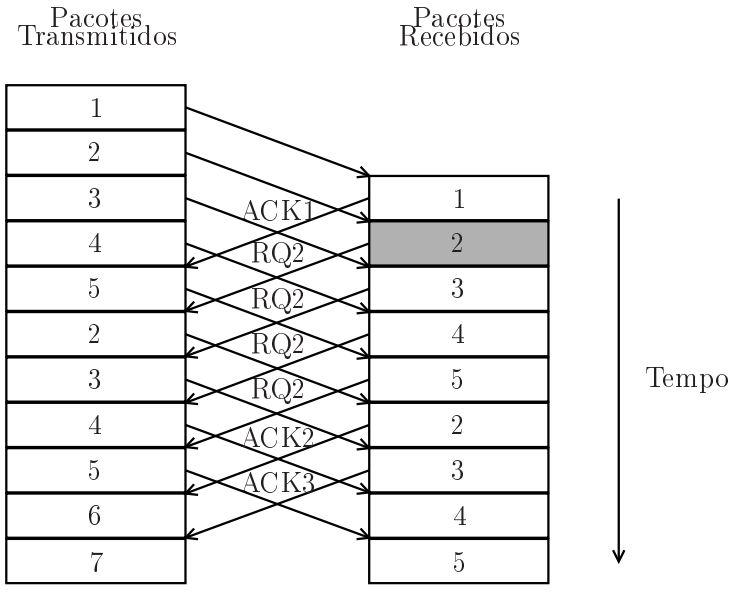


FIG. 9.17: Diagrama de tráfego do protocolo GBN-ARQ

Seja N o número de pacotes que serão necessários retransmitir toda a vez que se detete um pacote com erro. O valor de N é função dos retardos de transmissão λ_f,

recepção λ_b e processamento λ_b. Considerando o retardo total em termos de bits Γ dado pela EQ 9.58, N é simplesmente o menor número inteiro de pacotes que contém um total de no mínimo Γ bits conforme mostrado pela EQ 9.61

$$N = \left\lceil \frac{R(\lambda_f + \lambda_p + \lambda_b)}{n} \right\rceil = \left\lceil \frac{\Gamma}{n} \right\rceil \tag{9.61}$$

Como cada pedido de retransmissão causa a retransmissão de N pacotes, a vazão do protocolo GBN-ARQ é expressa pela EQ 9.62.

$$\eta_{GBN} = \left(\frac{k}{n}\right)\left(\frac{1}{1+(T_r-1)N}\right) = R_c\left(\frac{1-P_r}{1+P_r(N-1)}\right) \tag{9.62}$$

O protocolo GBN-ARQ não é tão sensível ao retardo de propagação pois Γ/n é multiplicado pela probabilidade de retransmissão P_r que normalmente é pequena em sistemas eficientes.

Para o caso de retorno com ruído, usando-se o diagrama da FIG. 9.18 obtém-se a EQ 9.63, vazão do protocolo GBN-ARQ.

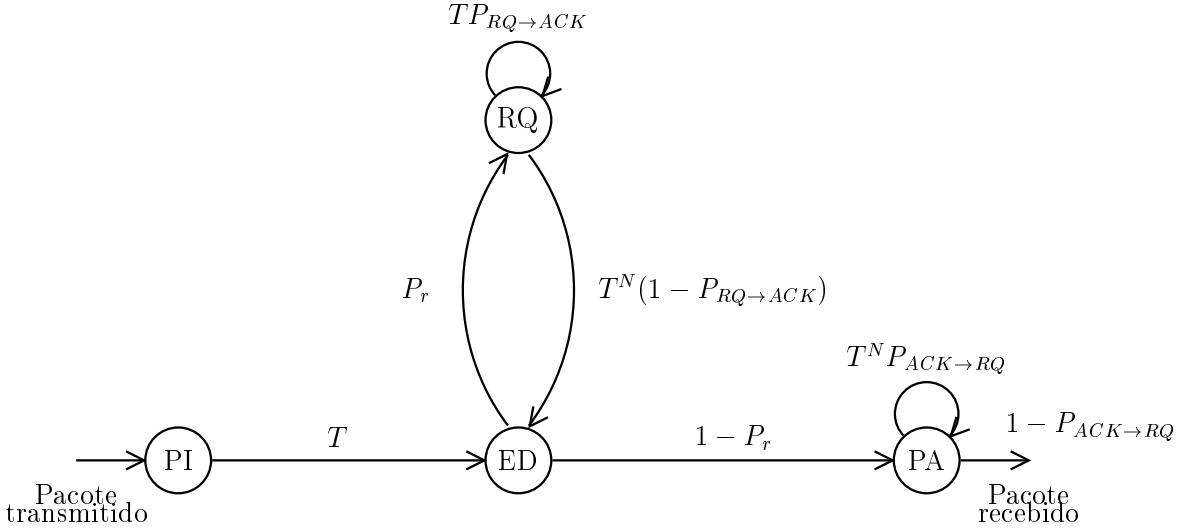


FIG. 9.18: Diagrama de estados do protocolo GBN-ARQ com ruído no retorno

$$\eta_{GBN} = R_c \frac{(1-P_r)(1-P_{ACK\to RQ})(1-P_{RQ\to ARQ})}{\left[\begin{array}{c}(1-P_rP_{ACK\to RQ}-P_{RQ\to ARQ}+P_rP_{RQ\to ARQ}) \\ +(N-1)(P_{ACK\to RQ}+P_r-P_{ACK\to RQ}P_{RQ\to ACK} \\ -P_rP_{RQ\to ARQ}-2P_rP_{ACK\to RQ}+2P_rP_{ACK\to RQ}P_{RQ\to ACK})\end{array}\right]} \tag{9.63}$$

9.4.2.3 SR-ARQ

Este protocolo requer armazenamento temporário tanto no transmissor quanto no receptor tornando mais complicada sua implementação. A FIG. 9.19 ilustra o funcionamento do protocolo SR-ARQ.

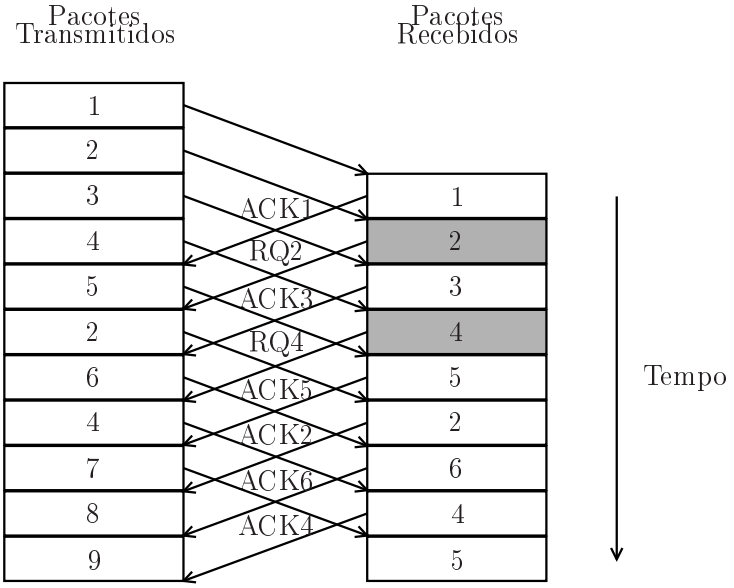


FIG. 9.19: Diagrama de estados do protocolo SR-ARQ

Neste protocolo, o transmissor envia os pacotes de modo contínuo da mesma forma que o GBN-ARQ. A diferença está que na hora de retransmitir um pacote apenas a cópia do pacote que causou a retransmissão é enviado. A vazão do protocolo SR-ARQ pode ser expressa pela EQ 9.64.

$$\eta_{SR} = \left(\frac{k}{n}\right)\left(\frac{1}{T_r}\right) = R_c(1 - P_r) \tag{9.64}$$

No caso de retorno com ruído a EQ 9.65 para vazão do protocolo SR-ARQ pode ser obtida usando-se o diagrama da FIG. 9.20.

$$\eta_{SR} = R_c \frac{(1 - P_r)(1 - P - ACK \rightarrow RQ)}{1 - P_r P_{ACK \rightarrow RQ}} \tag{9.65}$$

9.4.3 PROTOCOLOS ARQ HÍBRIDOS

O principal problema associado com os protocolos ARQ puros é que quando a qualidade do canal piora, o aumento da freqüência dos pedidos de retransmissão tem um impacto severo na vazão. Para contrabalançar este efeito, os protocolos híbridos usam

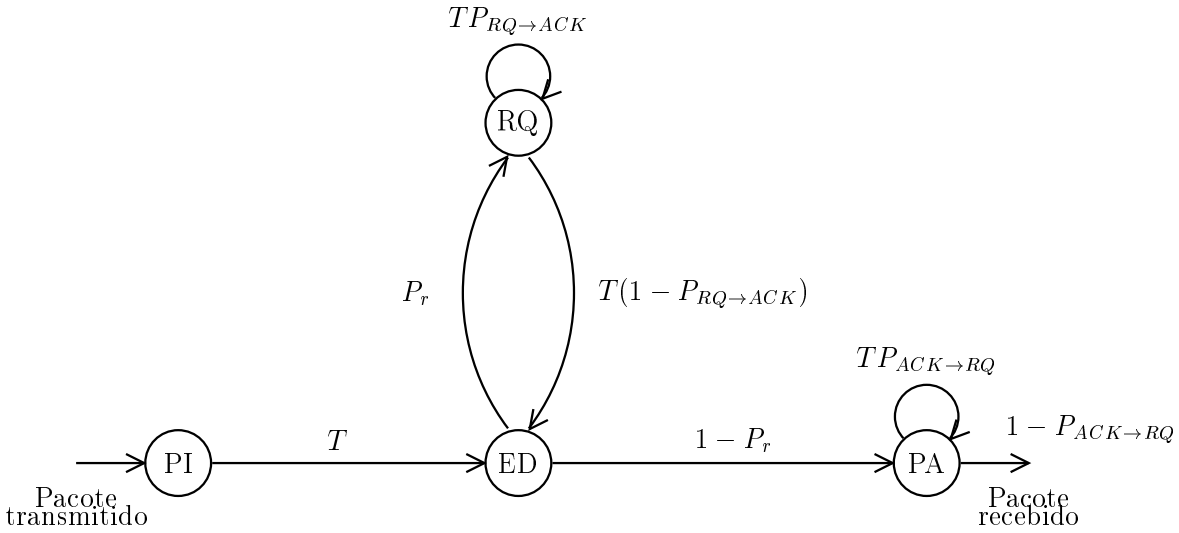


FIG. 9.20: Diagrama de estados do protocolo SR-ARQ com ruído no retorno

códigos corretores de erro FEC em conjunto com deteção de erro. Desta forma, os protocolos ARQ híbridos HARQ (*Hybrid Automatic Repeat reQuest*) possuem uma vazão similar aos sistemas FEC e ao mesmo tempo em que oferecem uma confiabilidade típica de protocolos ARQ.

9.4.3.1 PROTOCOLOS ARQ/FEC TIPO I

O protocolo ARQ/FEC Tipo I é o mais simples protocolo híbrido. Cada pacote é codificado para correção e deteção de erros. Estes protocolos podem ser implementados usando um uńico código para correção e deteção de erro ou dois códigos, um para correção e outro para deteção. O diagrama de estados destes protocolos estão ilustrados nas FIG. 9.21 e FIG. 9.22, respectivamente.

A probabilidade de aceitação de pacote errado é calculada a partir da EQ 9.66.

$$P_a = \frac{P_r^{(FEC)} P_u^{(ED)}}{1 - P_r^{(FEC)} P_r^{(ED)}} \tag{9.66}$$

onde:

$P_u^{(ED)}$ é a probabilidade de erros não detetado do código detetor de erro ED

$P_r^{(ED)}$ é a probabilidade de retransmissão do código detetor de erro ED $(P_e + P_f)$

$P_r^{(FEC)}$ é a probabilidade de retransmissão do código corretor de erro FEC $(P_e + P_u + P_f)$

A vazão deste protocolo é determinada a partir das mesmas expressões dos protocolos ARQ puro substituindo-se apenas P_r por $P_r^{(FEC)} P_r^{(ED)}$.

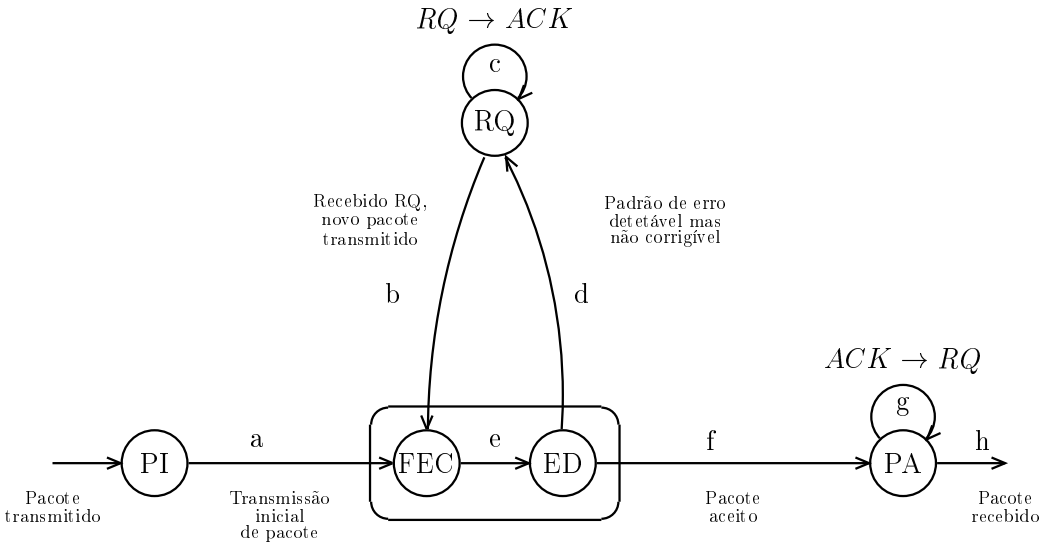


FIG. 9.21: Diagrama de estados de protocolo ARQ/FEC Tipo I com ruído no retorno e dois códigos

9.4.3.2 PROTOCOLOS ARQ/FEC TIPO II

O protocolo ARQ/FEC Tipo II adapta-se às condições variáveis do canal através do uso do incremento da redundância. Desta forma, ao receber um pedido de retransmissão, o transmissor envia bits adicionais de paridade para o receptor. O receptor pode usar então um código com taxa menor e, portanto, maior possibilidade de correção. Tal sistema constitui um sistema de combinação de código.

A FIG. 9.23 ilustra o diagrama de estados de um protocolo ARQ/FEC Tipo II. Nesta figura, o pacote recebido é primeiramente checado para verificar a existência de erros no estado ED_1. Caso possua um padrão de erro detetável, o pacote é enviado para a correção de erros no estado FEC e novamente checado para verificar se os erros foram corrigidos no estado ED_2. Caso os erros persistam, o protocolo passa para o estado de pedido de retransmissão RQ para solicitar os bits de paridade adicionais ou, quando a taxa mínima do código for atingida, uma nova cópia do pacote.

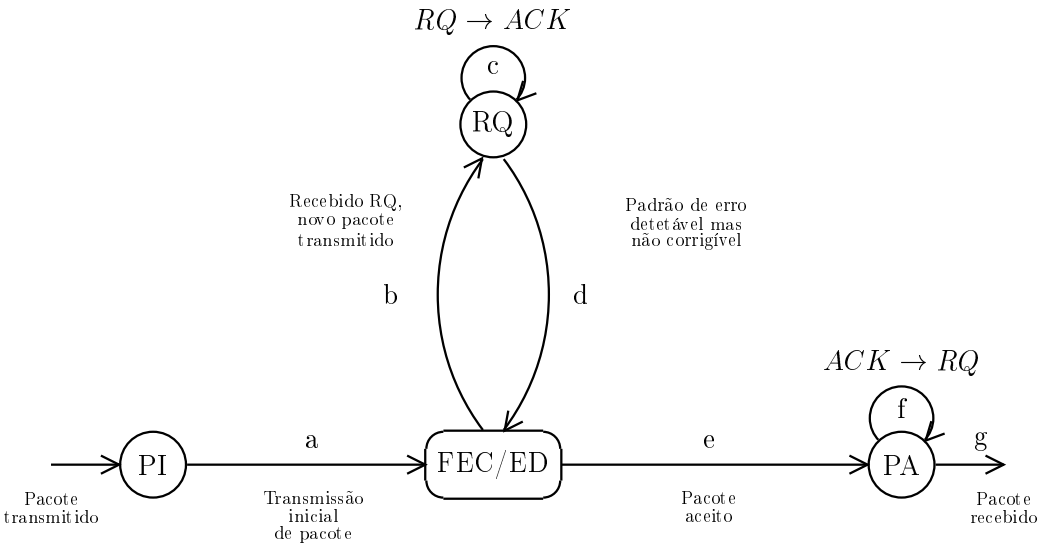


FIG. 9.22: Diagrama de estados de protocolo ARQ/FEC Tipo I com ruído no retorno e um código

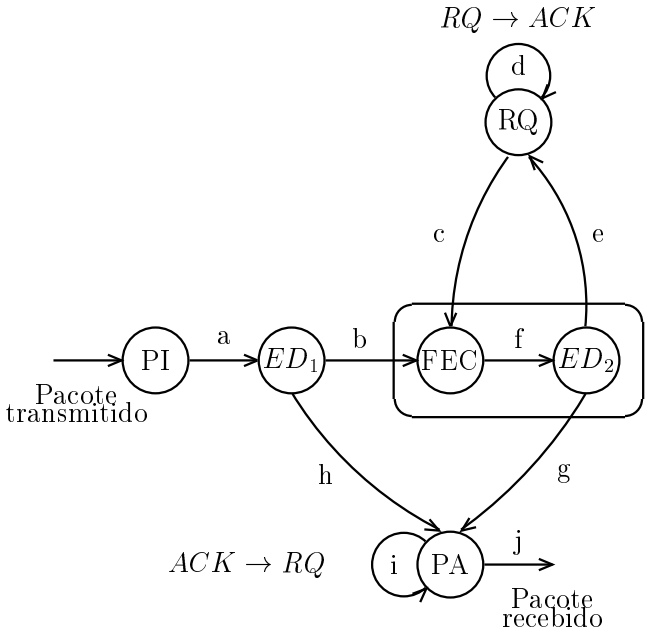


FIG. 9.23: Diagrama de estados de um protocolo ARQ/FEC Tipo II com ruído no retorno com um código

9.5 APÊNDICE 5: CAMADA DE ENLACE DA REDE RÁDIO HF

Em 1992, a NTIA (*National Telecommunications and Information Administration*) requisitou que os fabricantes de equipamentos HF submetessem para padronização um protocolo de enlace de dados adaptado aos modems de dados disponíveis naquela época. Basicamente, os modems eram aqueles descritos no MIL-STD-188-110A .

Em 1994, chegou-se a um concenso e o protocolo de enlace de dados foi incorporado ao MIL-STD-187-721C e ao, seu equivalente civil, o FED-STD-1052, no apêndice B. Estes dois documentos descrevém sucintamente como deve ser o protocolo de enlace de dados HFDLP (*High Frequency Data Link Protocol*) de uma rede rádio em HF.

O HFDLP quando usado em conjunto com um modem de dados apropriado provê um modo de transmitir informação livre de erros por um canal rádio HF. Basicamente, o HFDLP recebe uma mensagem ou bloco de informação de uma camada superior, executa a segmentação em quadros e séries de quadros e gerencia o envio desta mensagem para a outra ponta do enlace HF. Desta forma o HFDLP implementa os serviços usuais de uma camada de enlace tais como envio de múltiplas mensagens por meio de um único enlace com controle de fluxo, deteção de erro e opcionalmente com correção de erro por meio de retransmissões (ARQ).

9.5.1 DESCRIÇÃO DO PROTOCOLO

O HFDLP é um protocolo do tipo ARQ puro com repetição seletiva SR-ARQ. Esta característica faz com que este protocolo tenha um desempenho superior em relação ao AX.25 (ARRL, 1997), muito utilizado no meio civil pelos rádio amadores. Esta superioridade é explicada pelo fato do desvanecimento do canal HF causar freqüêntes retransmissões. Como o AX.25 é do tipo GBN-ARQ, cada retransmissão causa o reenvio de quadros que já haviam sido recebidos corretamente. Estas retransmissões causam uma redução na vazão do AX.25 a praticamente zero em enlaces onde o HFDLP conseguiria passar tráfego com sucesso.

O número de bytes por quadro de dados é variável no HFDLP e pode ser adaptável às condições do canal, de modo a maximizar a vazão pelo canal. Em enlaces com uma alta taxa de erro, o uso de quadros pequenos aumenta a probabilidade de que alguns quadros sejam recebidos sem erros causando um consequente diminuição do número de

retransmissões.

Outro fator que influencia na vazão do HFDLP em um enlace half-duplex é o tempo de inversão do sentido de comunicação. Desta forma, para evitar que sucessivas inversões diminuam a vazão, os quadros são agrupados em séries. O número de quadros por séries de dados é mais uma variável cujo ajuste pode otimizar o desempenho do protocolo.

Uma das características mais importantes do HFDLP é o tratamento de mensagens como unidades de transmissão. Fazendo assim, o HFDLP pode negociar o uso do enlace de dados baseado nas prioridades das mensagens bem como reservar área de armazenamento no terminal receptor informando o tamanho da mensagem. O HFDLP também pode multiplexar várias conexões da camada de rede sobre um único enlace de dados através do uso da identificação de conexão que é acrescida a cada mensagem enviada.

Para implementar estas facilidades o HFDLP utiliza os quadros de controle. O HFDLP pode operar tanto com quadros de controle de tamanho fixo como com quadros de controle de tamanho variável que contém apenas os campos necessários em cada instância da comunicação.

9.5.2 MODOS DE OPERAÇÃO

O HFDLP possui os modos de operação descritos na TAB. 9.2. Estes modos foram delineados para prover uma ampla gama de desempenho com vários graus de complexidade de implementação. O principal modo de operação é o modo ARQ que provê transferências de dados livres de erro ponto a ponto. Os modos secundários são o modo de difusão e o modo ARQ circuito. O modo difusão permite transferência de dados unidirecionais para múltiplos receptores. O modo ARQ circuito permite que um enlace seja estabelecido e mantido mesmo com ausência de tráfego.

O simulador usado nesta tese implementa o modo ARQ com quadros de tamanho fixo e variável.

TAB. 9.2: Modos de operação do HFDLP

Modo	ARQ ?	Tamanho dos Quadros de Controle
ARQ (variável)	SIM	Variável
Difusão	NÃO	Fixo
ARQ circuito	SIM	Variável
ARQ fixo	SIM	Fixo

9.5.3 SUBPROTOCOLOS DO HFDLP

De fato, o HFDLP é constituído por três subprotocolos:

- **Protocolo de Gerência de Mensagens**: trabalha com mensagens completas como unidades. Este protocolo negocia a ordem de transmissão de mensagens sobre o enlace de dados baseado na prioridade das mensagens e provê a multiplexação de várias conexões através de um único enlace de dados. Este protocolo provê mecanismos para a troca de uma mensagem de baixa prioridade por uma mensagem de alta (tanto na direção direta como reversa) e o restabelecimento da mensagem de baixa prioridade após o término da mensagem de alta. O protocolo também provê um meio para que o terminal que está recebendo a mensagem especifique o ponto de retomada ou requeira a completa retransmissão, caso esta tenha sido jogada fora. De certa forma, este protocolo sobrepõe algumas funções de camadas de mais alto nível.

- **Protocolo de Transferência de Dados**: gerencia a transferência de quadros de dados e controle pelo enlace, inclusive confirmações de recebimento e controle de fluxo, além de negociação de parâmetros do modem (taxa de dados, profundidade de entrelaçamento), tamanho de quadros e número de quadros por séries. Dos três protocolos, este protocolo é o que mais se aproxima da noção usual de protocolo de enlace de dados.

- **Protocolo de Gerência de Enlace**: estabelece e monitora continuamente a atividade da camada física que suporta o HFDLP. Exceto no modo de operação ARQ circuito, o enlace de dados é sempre encerrado quando nenhum dos terminais possui mensagems para enviar. A expiração do tempo de enlace ocasiona o término do enlace mesmo que uma mensagem esteja sendo transmitida. Neste caso, o protocolo de gerência de mensagens ou uma camada de nível mais alto pode requerer o restabelecimento do enlace de dados e o restabelecimento da transmissão da mensagem.

Estes protocolos podem ser modelados por máquinas de estados finitos conforme ilustrado a seguir.

9.5.3.1 MÁQUINA DE ESTADO DE GERÊNCIA DE MENSAGENS

O protocolo de gerência de mesagens é funcionalmente descrito pela máquina de estados MMSM (*Message Management State Machine*) mostrada na FIG. 9.24. Está

máquina representa a interface do HFDLP com a camada de rede local. Deste modo, a MMSM de cada terminal reflete o estado corrente daquele terminal, ou seja, livre, transmitindo, recebendo ou tentando transmitir.

Diz-se que um terminal é transmissor ou receptor quando tenha negociado o direito de transmitir ou receber respectivamente. Assim, um terminal só entra no estado de tentativa de transmissão se tiver negociado anteriormente para receber e precisa se tornar transmissor. A existência deste estado assegura que nunca deverá haver dois terminais transmitindo simultâneamente.

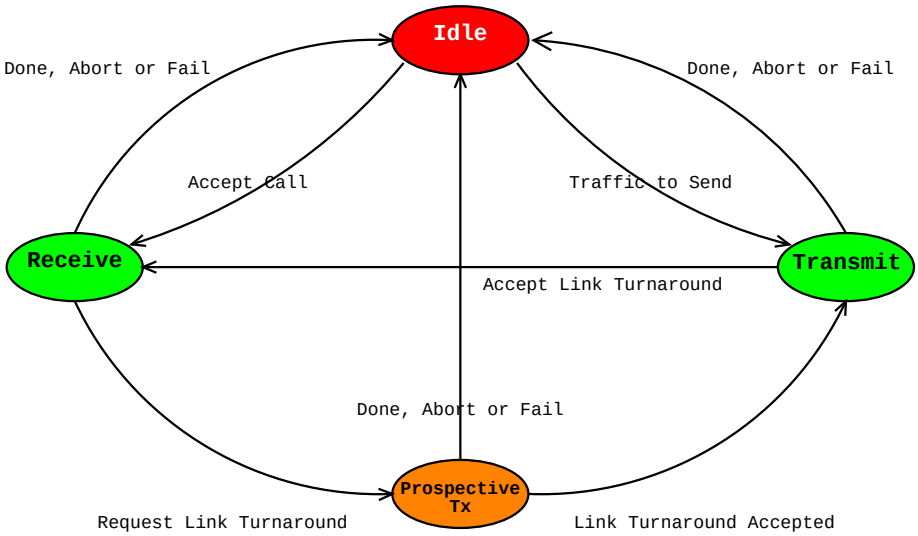


FIG. 9.24: Diagrama da máquina de estados de gerência de mensagens

9.5.3.2 MÁQUINA DE ESTADO DE TRANSFERÊNCIA DE DADOS

A FIG. 9.25 ilustra a DTSM (*Data Transfer State Machine*). Está máquina é implementada através da troca de quadros de controle e de dados através do enlace de dados. Esta troca é efetuada em três fases:

- **Negociação**: esta fase começa com a transmissão de um quadro controle contendo um anunciador (*herald*) que informa a existência de uma série de dados e termina com a confirmação do recebimento dos parâmetros de transmissão contidos no anunciador.

- **Transferência dos dados**:neste fase o terminal transmissor inicia a transferência sem interrupção das series de dados.

- **Confirmação de recebimento**: inicia-se toda a vez que uma serie de dados for

recebida pelo terminal receptor. Consiste na emissão de um quadro de controle informando quais quadros chegaram sem erro.

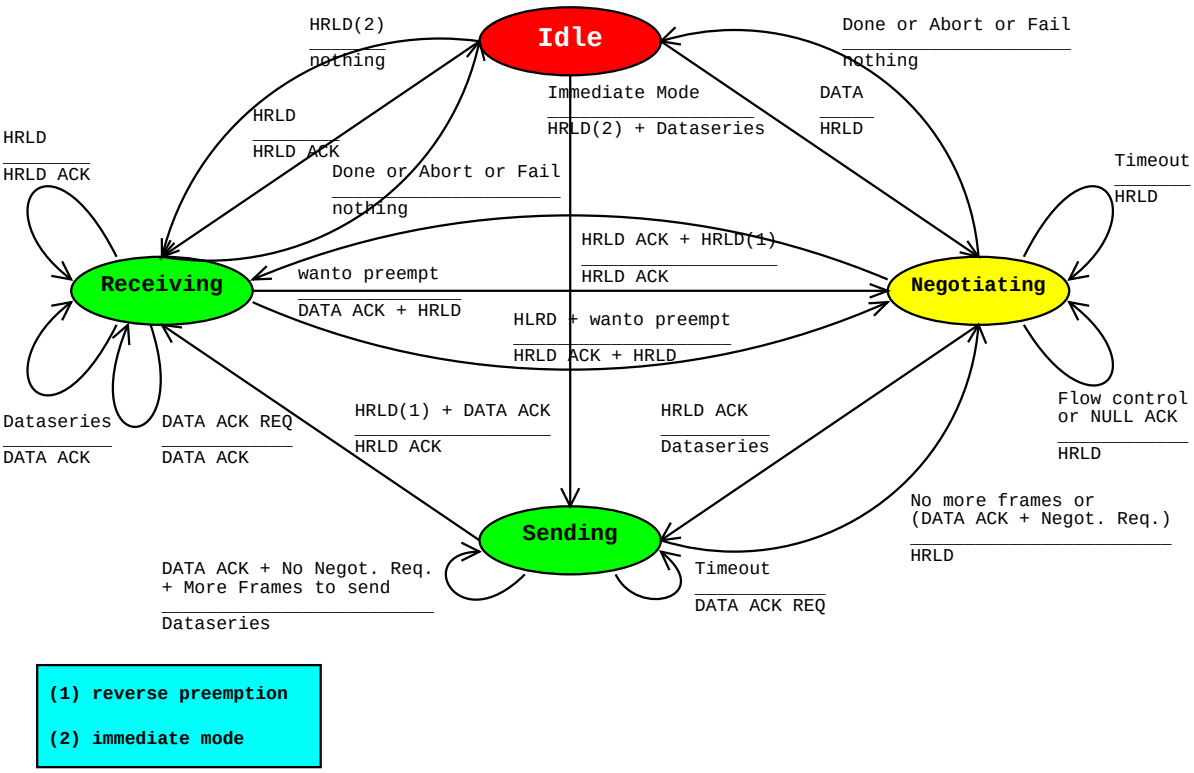


FIG. 9.25: Diagrama da máquina de estados de transferência de dados

9.5.3.3 MÁQUINA DE ESTADO DE GERÊNCIA DE ENLACE

A FIG. 9.26 ilustra a LMSM (*Link Management State Machine*). Este protocolo obriga os dois terminais que desejam estabelecer um enlace a percorrer um série de estados até atingir aquele no qual o enlace é estabelecido (*Link up*). Para que o terminal transmissor possa detetar falhas de enlace, este protocolo emprega o mecanismo de expiração de tempo.

9.5.4 TIPOS DE QUADROS

As implementações do HFDLP trocam dois tipos de quadros: quadros de dados e quadros de controle. Os quadros do primeiro tipo carregam dados da camada de rede com a mínima sobrecarga por quadro enquanto os do segundo tipo são usados para estabelecer enlaces de dados, negociar parâmetros de transferência de dados (inclusive prioridades de mensagens) e confirmar a recepção de quadros de dados.

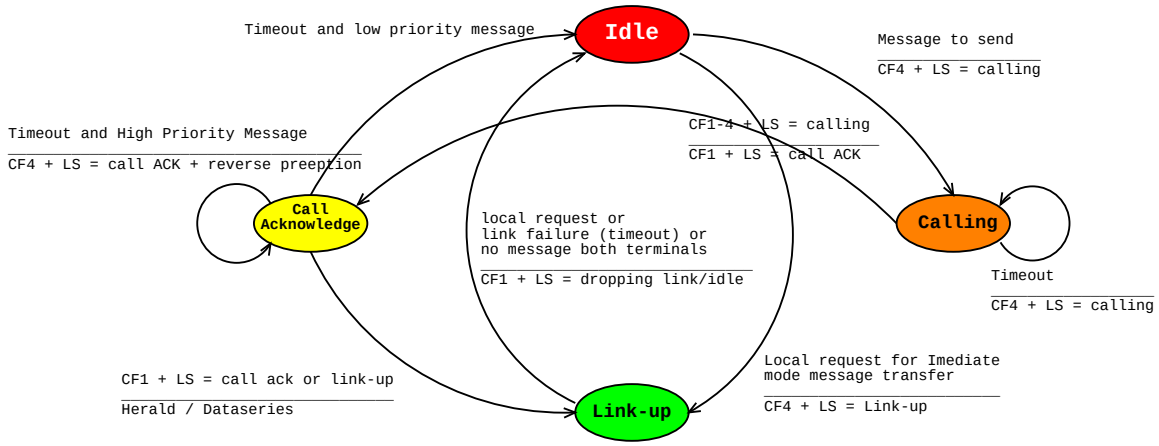


FIG. 9.26: Diagrama da máquina de estados de gerência de enlace

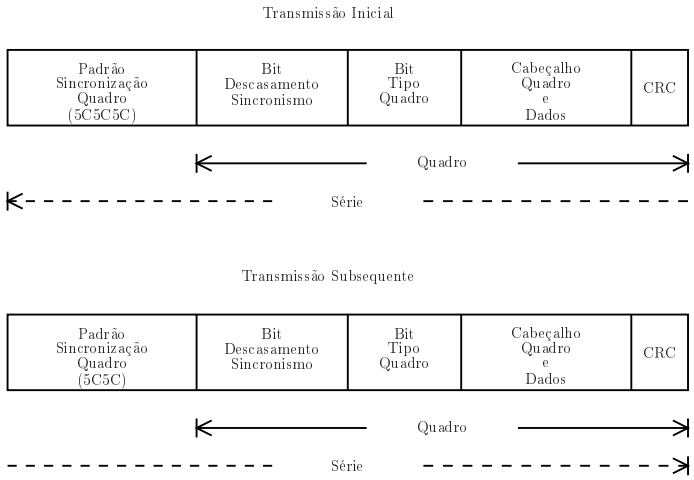


FIG. 9.27: Formato básico do quadro do protocolo

A FIG. 9.27 mostra o formato básico dos quadros e séries de quadros do HFDLP. A descrição da estrutura dos quadros do HFDLP é a seguinte:

- **Padrão de Sincronismo de Quadro**: Cada nova transmissão através do canal físico começa com os três bytes de sincronização de quadro para identificar o tráfego que segue como tráfego HFDLP. Se a transmissão contém mais de um frame, uma seqüência de dois bytes é inserida entre frames adjacentes.

- **Bit de Descasamento de Sincronismo**: o primeiro bit seguindo o padrão de sincronismo é 1 para indicar o fim do padrão de sincronismo e o começo do quadro do protocolo.

- **Bit de Tipo de Quadro**: Este bit é 1 quando o quadro atual é de controle e 0 quando é de dados.

- **Cabeçalho de Quadro e Dados**: Contém os cabeçalhos dos quadros de controle ou os cabeçalhos e dados dos quadros de dados.

- **Controle de Erro CRC**: No final do quadro é acrescido um CRC de 32 bits calculado usando-se todos os bits do quadro começando do bit de descasamento de sincronismo até o último bit do campo de cabeçalho de quadro e dados. O polinômio gerador do código CRC é dado pela EQ 9.67

$$g(x) = x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1 \quad (9.67)$$

9.5.4.1 FORMATO DO QUADROS DE DADOS

A TAB. 9.3 define o formato dos quadros de dados do HFDLP. Para uma descrição detalhada de cada campo o padrão MIL-STD-187-721C (DoD, 1994) deve ser consultado.

TAB. 9.3: Formato do quadro de dados

Nome do Cabeçalho	Nome do Campo	Tamanho (bits)	Possíveis Valores		
Cabeçalho Quadro	Bit Desc Sinc	1	1 (sempre 1)		
	Tipo Quadro	1	0 = quadro de dados		
Quadro Controle Canal Reverso	Formato Taxa Dados	1	0 = taxa de dados absoluta 1 = taxa de dados relativa		
	Taxa de Dados	3	Código	Formato absoluto	Formato relativo
			0	75 bps	÷ 8
			1	150bps	÷ 4
			2	300bps	÷ 2
			3	600bps	não muda
			4	1200bps	× 2
			5	2400bps	× 4
			6	4800bps	× 8
			7	nenhuma recomendação	
	Tamanho do Entrelaçamento	1	0 = entrelaçamento curto 1 = entrelaçamento longo		
	Reservado	1	Setado em 0		
Cabeçalho Quadro Dados	Número de Ordem Quadro Dados	8	Contagem regressiva 255 a 1; identifica o quadro dentro da série		
	Deslocamento da mensagem (byte)	21	Posição do quadro dentro da mensagem (byte 0 é o primeiro byte)		
	Reservado	3	Setado em 0		
Dados		variável	Tamanho variável definido pelo quadro controle		
	CRC	32			

O número máximo de quadros por série varia de acordo com a taxa de dados usada conforme resumido na TAB. 9.4.

9.5.4.2 FORMATO DO QUADROS DE CONTROLE

A TAB. 9.5 mostra os quatro tipos predefinidos de quadros de controle e a TAB. 9.6 define o formato dos campos dos quadros de controle do HFDLP.

TAB. 9.4: Máximo tamanho da série

Taxa de Dados (bps)	Número máximo de quadros por série
75	8
150	16
300	32
600	64
1200	128
2400	255
4800	510

TAB. 9.5: Tipos de quadros de controle

Nome do Cabeçalho	Nome do Campo	Tamanho (bits)	Tipo 1	Tipo 2	Tipo 3	Tipo 4
Cabeçalho do Quadro	Bit Desc Sinc	1	√	√	√	√
	Tipo Quadro	1	√	√	√	√
Cabeçalho do Quadro de Controle	Versão Protocolo	2	√	√	√	√
	Modo Controle	2	√	√	√	√
	Modo Negociação	1	√	√	√	√
	Endereço Extendido	1	√	√	√	√
	Endereço Fonte	16	√	√	√	√
	Endereço Destino	16	√	√	√	√
Gerência Enlace	Estado do Enlace	2	√	√	√	√
	Espiração Enlace	4	√	√	√	√
Transferência de Dados	Tipo ACK/NAK	2	√	√	√	√
	Mapa-bit ACK/Endereço Extendido	256	×	√	√	√
	Alternância Quadros ACK Dados	1	×	√	√	√
	Formato Taxa Dados	1	×	√	√	√
	Taxa de Dados	3	×	√	√	√
	Tamanho do Entrelaçamento	1	×	√	√	√
	Número bytes em Quadros de Dados	10	×	√	√	√
	Número de Quadros na Próxima Série	8	×	√	√	√
Gerência de Mensagens	ID Mensagem TX	8	×	×	√	√
	ID Conexão TX	8	×	×	√	√
	Tamanho Mensagem TX	24	×	×	√	√
	Localização Próximo Byte Mensagem TX	21	×	×	√	√
	Reservado	3	×	×	√	√
	Prioridade Mensagem TX	8	×	×	√	√
	Localização Próximo Byte Mensagem RX	21	×	×	√	√
	Reservado	3	×	×	√	√
Função Extendida	ID usuário	14	×	×	×	√
	Bits de função	50	×	×	×	√
	CRC	32	√	√	√	√
Tamanho Total do quadro			80	360	456	520

TAB. 9.6: Formato do quadro de controle

Nome do Cabeçalho	Nome do Campo	Tamanho (bits)	Possíveis Valores
Cabeçalho do Quadro	Bit Desc Sinc	1	1 (sempre 1)
	Tipo Quadro	1	1 = quadro de controle
Cabeçalho do Quadro de Controle	Versão Protocolo	2	Setado em 0
	Modo Controle	2	0 = modo ARQ, quadro controle tamanho variável 1 = modo difusão, sem ARQ, quadro controle tamanho fixo 2 = modo circuito, ARQ, quadro controle tamanho variável 3 = modo ARQ, quadro controle tamanho fixo
	Modo Negociação	1	0 = negociar apenas quando houver mudanças 1 = negociar antes de qualquer séries de dados
	Endereço Extendido	1	0 = endereço com 2 bytes 1 = endereço com 18 bytes
	Endereço Fonte	16	Hexadecimal de 0000 a FFFF representando os dois bytes menos significativos do endereço da fonte
	Endereço Destino	16	Hexadecimal de 0000 a FFFF
Gerência Enlace	Estado do Enlace	2	0 = chamando 1 = chamada recebida 2 = enlace estabelecido 3 = desligando enlace
	Espiração Enlace	4	Máximo tempo de espera por resposta válida
Transferência de Dados	Tipo ACK/NAK	2	0 = ACK nulo 1 = ACK dados 2 = requisição ACK dados 3 = ACK indicação
	Mapa-bit ACK / Endereço Extendido	256	0 = retransmitir quadro associado 1 = quadro livre de erros
	Alternância Quadros ACK Dados	1	muda de estado para cada novo quadro ACK dados
	Formato Taxa Dados	1	0 = taxa de dados absoluta 1= taxa de dados relativa
	Taxa de Dados	3	Código: 0 — Formato absoluto: 75 bps — Formato relativo: ÷ 8 Código: 1 — Formato absoluto: 150bps — Formato relativo: ÷ 4 Código: 2 — Formato absoluto: 300bps — Formato relativo: ÷ 2 Código: 3 — Formato absoluto: 600bps — Formato relativo: não muda Código: 4 — Formato absoluto: 1200bps — Formato relativo: × 2 Código: 5 — Formato absoluto: 2400bps — Formato relativo: × 4 Código: 6 — Formato absoluto: 4800bps — Formato relativo: × 8 Código: 7 — nenhuma recomendação
	Tamanho do Entrelaçamento	1	0 = entrelaçamento curto 1 = entrelaçamento longo
	Número bytes em Quadros de Dados	10	Decimal de 56 a 1023
	Número de Quadros na Próxima Série	8	Decimal de 1 a 255 0 denota Indicação nula
Gerência de Mensagens	ID Mensagem TX	8	Decimal de 0 a 255
	ID Conexão TX	8	Decimal de 0 a 255
	Tamanho Mensagem TX	24	Tamanho da mensagem em bits
	Localização Próximo Byte Mensagem TX	21	Posição próximo byte mensagem
	Reservado	3	Setado em 0
	Prioridade Mensagem TX	8	Decimal de 0 a 255; 0 menor prioridade
	Localização Próximo Byte Mensagem RX	21	Posição próximo byte mensagem requerida pelo terminal receptor
	Reservado	3	Setado em 0
Função Extendida	ID usuário	14	Sinalizador de linha de serviço
	Bits de função	50	Dados de linha de serviço
	CRC	32	

9.6 APÊNDICE 6: RELAÇÃO ENTRE AS ENERGIAS DE BIT E SÍMBOLO

A relação entre as razões entre a energia do símbolo e do ruído $\frac{E_s}{N_0}$ e a energia do bit de informação e do ruído $\frac{E_b}{N_0}$ é dada por:

$$\frac{E_b}{N_0} = \frac{\frac{E_s}{N_0}}{\log_2 M} \tag{9.68}$$

onde $N_0 = \sigma^2$ é o parâmetro da Densidade Espectral de Potência do ruído passa-baixa.

Após passar por um codificador de taxa $R_c = \frac{k}{n}$, a relação entre a razão da energia do bit de informação codificado pelo ruído $\frac{E_{bc}}{N_0}$ e entre a energia do bit de informação e do ruído $\frac{E_b}{N_0}$ será dada por:

$$\frac{E_{bc}}{N_0} = R_c \, \frac{E_b}{N_0} \tag{9.69}$$

Considerando a energia do bit de informação codificado unitária tem-se a normalização expressa pela equação

$$N_0 = 10^{-\frac{\frac{E_{bc}}{N_0}[dB]}{10}} \tag{9.70}$$

9.7 APÊNDICE 7: DIAGRAMA DO SIMULADOR DE CANAL HF

O diagrama em bloco da FIG. 9.28 ilustra a estrutura utilizada para realizar em computador a simulação do canal HF com dois raios, ou seja, um direto e outro retardado.

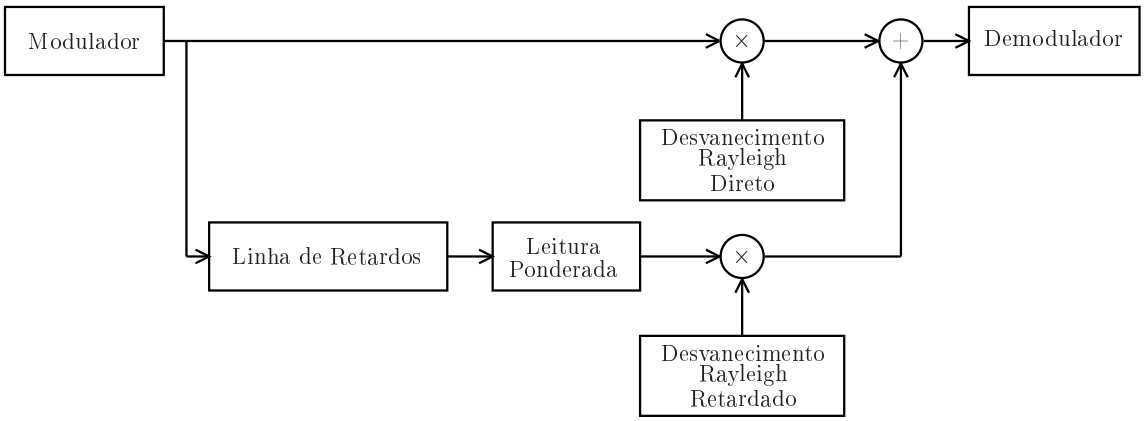


FIG. 9.28: Simulador de Canal HF pelo modelo de Watterson

Na FIG. 9.28 as multiplicações e somas são todas complexas. O raio retardado é obtido através da leitura ponderada de uma linha de retardos. O comprimento da linha de retardos é dado pela EQ 9.71.

$$L = \left\lceil \frac{T_m}{T_s} \right\rceil NPTS \tag{9.71}$$

onde:

L	é o comprimento da linha de retardos
T_m	é o espalhamento de retardo multipercurso
T_s	é a duração do símbolo
$NPTS$	número de amostras por símbolo

O bloco leitura ponderada é necessário em virturde de não haver na simulação nenhuma restrição de que o espalhamento multipercurso T_m seja múltiplo do tempo de símbolo T_s. Deste modo, o valor lido em um dado instante corresponde a média ponderada das amostras anterior e posterior usando como ponderação a distância temporal das amostras ao instante da leitura.

Ainda na FIG. 9.28, os blocos de desvanecimento Rayleigh direto e retardado geram os ganhos complexos que irão multiplicar as amostras dos símbolos. Os valores dos ganhos são gerados através da estrutura ilustrada na FIG. 9.29.

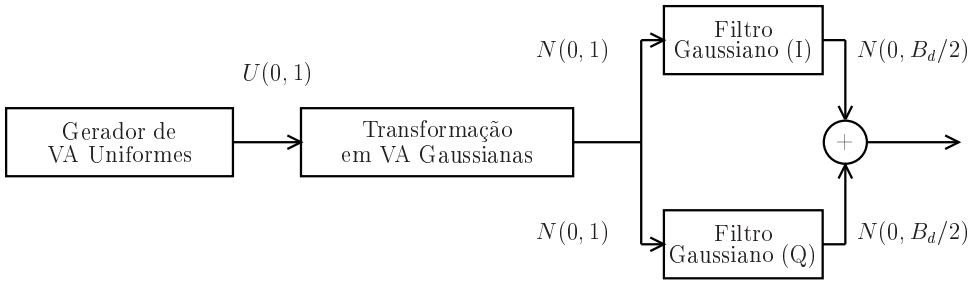


FIG. 9.29: Gerador de desvanecimento

Na FIG. 9.29, o bloco Gerador de VA Uniformes gera duas variáveis uniformes com valores no intervalo $(0, 1)$. Em seguida cada uma destas duas variáveis uniformes são mapeadas em duas variáveis independentes e conjuntamente Gaussianas de média nula e variância unitária pelo bloco Transformação em VA Gaussianas. Por último, os blocos Filtros Gaussianos em fase (I) e quadratura (Q) transformam a seqüência de VAś independentes e variância unitária em uma seqüência de VAś correlatadas com média nula e variância $B_d/2$ de acordo com o modelo de Watterson. Os detalhes da implementação podem ser observados no programa e não serão apresentados neste texto.