



UNIVERSIDADE FEDERAL DO ABC

PATRICIA YUMI OZONO

**Estudo e Análise de Desempenho da Equalização Turbo em
Sistemas PLC**

Santo André

2017

PATRICIA YUMI OZONO

**Estudo e Análise de Desempenho da Equalização Turbo em
Sistemas PLC**

Relatório apresentado à Universidade Federal do ABC como requisito parcial para obtenção do título de bacharela no curso de Engenharia de Informação, sob orientação do Prof. Doutor Murilo Bellezoni Loiola.

Santo André

2017

Dedicatória

Dedico este trabalho à minha família e namorado, que sempre incentivaram e apoiaram minha formação em engenharia.

Agradecimentos

Aos meus familiares e namorado, pelo incentivo, apoio e amor incondicional que me tem fornecido e fortalecido por toda a vida.

Ao meu professor orientador Murilo Bellezoni Loiola, por toda a paciência, auxílio e cooperação oferecidos ao longo desse 1 ano de realização do trabalho.

Aos docentes da Universidade Federal do ABC, por todos esses anos de batalha e coleguismo me transmitindo seus conhecimentos essenciais para que eu tenha uma sólida formação e me torne uma grande profissional.

Aos meus colegas, por cada um ser uma peça importante no meu crescimento, seja no auxílio em estudos ou no passeio realizado em dias de feriado ou finais de semana.

*“A felicidade não se resume na ausência de problemas,
mas sim na sua capacidade de lidar com eles.”*

(Albert Einstein)

RESUMO

A comunicação utilizando a rede elétrica é intitulada como *Power Line Communication* (PLC). A primeira parte deste trabalho apresenta um levantamento teórico dessa tecnologia e, logo em seguida, é apresentada uma simulação do canal PLC em ambiente computacional. Outra consideração importante a ser feita em ambientes PLC é a presença de ruídos impulsivos. Portanto, foram estudados modelos matemáticos que pudessem modelá-los e serem simulados em computador. Tendo o canal e o ruído definidos, as seções seguintes deste trabalho focaram no estudo de alguns processos básicos que compõem um sistema de transmissão de informação: codificação, equalização de canal e decodificação. A fim de melhorar o desempenho na recuperação da mensagem original enviada para o receptor, foi escolhido o método de Equalização Turbo, pois estudos indicam que a combinação iterativa entre equalizador e decodificador, além das decisões suaves (probabilísticas), melhora a tomada de decisão do bit recuperado. Foram escolhidos três algoritmos para realizar a equalização: MAP, Max-Log-MAP e SFE, e um para a decodificação: Log-MAP. Os primeiros testes foram feitos apenas considerando o canal PLC e adição de ruído AWGN. Seus resultados foram satisfatórios, pois a mensagem original pôde ser recuperada e todos os algoritmos alcançaram taxas de erro de cerca de 1×10^{-5} , ou seja, apenas 1 bit errado a cada 100.000 transmitidos, para relações sinal-ruído menores que 7 dB. Outros testes foram feitos considerando a adição de ruído impulsivo ao invés do AWGN, e os resultados não foram bons, pois as taxas de erro ficaram sempre em 100%. A solução proposta para tentar melhorar tal desempenho foi adicionar uma rede neural, do tipo MLP, antes da equalização turbo, mas não houve melhora no desempenho do sistema. Uma possível explicação para isso vem do fato de que o equalizador turbo não vê mais o canal PLC original, mas uma combinação deste com a rede neural. Essa hipótese e outras soluções para os problemas enfrentados com o ruído impulsivo podem ser abordadas em trabalhos futuros.

Palavras-chave: Equalização turbo, PLC, Ruído impulsivo, Redes neurais.

ABSTRACT

Communication using power grid is known as Power Line Communication (PLC). The first part of this document presents a theoretical survey of this technology and after this, a simulation of this channel in a computational environment is presented. Another important consideration to be made in PLC environments is the presence of impulsive noises. Therefore, mathematical models were studied to model and simulate this kind of noise in a computer. Having the channel and the noise defined, the following sections of this document focused on the study of some basic processes that compose an information transmission system: encoding, channel equalization and decoding. In order to improve performance in the retrieval of the original message sent to the receiver, the Turbo Equalization technique was chosen because studies indicate that the iterative combination between equalizer and decoder, besides the soft decisions (probabilistic) generated by them, improves the decision of the bit recovered. Three algorithms were chosen to perform the equalization: MAP, Max-Log-MAP and SFE, and one for the decoding: Log-MAP. The first tests were done only considering the PLC channel and the addition of AWGN noise. Their results were satisfactory as the original message could be retrieved and all algorithms reached error rates of up to 1×10^{-5} , i.e. only 1 wrong bit per 100,000 transmitted. Other tests were done considering the addition of impulsive noise rather than the AWGN, and the results were not so good, with error rates about 100%. The proposed solution to try to improve such performance was to add an MLP neural network, before the turbo equalization, but there was no improvement in turbo equalization performance. A possible explanation for this is the fact that the turbo equalization doesn't see the original PLC channel, but a combination between this one and the neural network. This hypothesis as well as other solutions to the problems faced with the impulsive noise can be approached in future studies.

Keywords: Turbo equalization, PLC, Impulsive noise, Neural networks.

Sumário

1. Introdução.....	12
2. O que é um canal de comunicação PLC?	13
2.1. Simulação de um canal PLC	16
3. Ruído Impulsivo	19
3.1. Distribuição alfa-estável simétrica	20
4. Processamento da informação pelo transmissor	23
4.1. Codificador RSC	24
5. Equalização de canal.....	30
5.1. Redes Neurais Artificiais	32
6. Decodificação de Canal.....	35
7. Equalização turbo.....	40
7.1. Decodificadores Atuando como Equalizadores	42
7.1.1. Equalizador MAP	42
7.1.2. Equalizador Max-Log-MAP	42
7.1.3. Equalizador SFE	44
8. Redes Neurais e Equalização Turbo.....	45
9. Simulações, Resultados e Discussões	46
10. Conclusões.....	53
Referências Bibliográficas.....	55
Apêndice I – Simulação canal PLC	57
Apêndice II – Simulação da Rede Neural.....	58
Apêndice III – Simulação do Ruído Impulsivo (Distribuição alfa-estável)	59

LISTA DE FIGURAS

Figura 1. Esquema resumido do processo de transmissão e recepção de informação.	12
Figura 2. Topologia adotada para modelagem do canal PLC, sendo Γ o coeficiente de reflexão no ponto indicado (B ou D) e T o coeficiente de transmissão. Fonte: [04].	14
Figura 3. Modelagem do canal PLC de 15 caminhos feita em [04].	16
Figura 4. Resposta em frequência do canal PLC com 15 caminhos multipercurso.	18
Figura 5. Exemplificação do comportamento do ruído AWGN (simulado em Matlab) e do ruído Impulsivo [17], no qual $n_b(t)$ representa a intensidade, ao longo do tempo.	20
Figura 6. Distribuições alfa-estáveis com $1 < \alpha < 2$	23
Figura 7. Diagrama das etapas consideradas no processo de transmissão e recepção de informação. Cada letra denota a forma como a mensagem se encontra dentro do sistema: m - forma original, c - codificada, c' - codificada e entrelaçada, r - passada pelo canal, c' - equalizada, c - desentrelaçada e m - recuperada no receptor.	24
Figura 8. Exemplo de codificador convolucional não sistemático.	26
Figura 9. Diagrama de Estados do codificador da Figura 8.	27
Figura 10. Diagrama em Trelça do codificador da Figura 8.	28
Figura 11. Representação dos tipos de combinações possíveis de uma codificação turbo. Fonte: [11].	29
Figura 12. Exemplo de codificador RSC. Fonte: [12].	30
Figura 13. Codificador RSC utilizado no projeto.	30
Figura 14. Esquema básico do processo de equalização adaptativa supervisionada. Fonte: [06].	31
Figura 15. Exemplo de uma rede neural biológica (esquerda) e artificial (direita). Fonte: [07].	32
Figura 16. Definição básica de um perceptron. Fonte: [07].	33
Figura 17. Modelo genérico de um neurônio artificial.	34
Figura 18. Exemplo de estrutura geral de um MLP (esquerda) e operação do algoritmo de treinamento. Fonte: [07].	35
Figura 19. Comparação entre decodificação abrupta e suave. Fonte: [11].	37

Figura 20. Estruturas básicas e iterativas da decodificação turbo. Fonte: [11].	39
Figura 21. Diagrama do sistema completo de transmissão adotado para este trabalho. As caixas tracejadas indicam a relação entre CODECs.	40
Figura 22. Comportamento de $\ln(1 + e - \delta_1 - \delta_2)$. Fonte: [11].	43
Figura 23. Estrutura do algoritmo SFE. Fonte: [11].	44
Figura 24. Receptor proposto com a combinação de MLP e equalização turbo para ambientes com ruído impulsivo.	46
Figura 25. Escolha da quantidade de coeficientes do canal PLC.	48
Figura 26. Espectro do canal PLC que foi ajustado para as limitações enfrentadas na simulação.	49
Figura 27. Desempenho da equalização turbo em canal PLC e com adição de ruído.	51
Figura 28. Desempenho da equalização turbo em canal PLC e ruído impulsivo com adição de rede neural na equalização.	52

LISTA DE SIGLAS

AI	Artificial Intelligence
AWGN	Additive White Gaussian Channel
BCJR	Bahl Cocke Jelinek Raviv
BER	Bit Error Rate
BPSK	Binary Phase Shift Key
CODEC	Codificador-Decodificador (do inglês, Encoder-Decoder)
ISI	Intersymbol Interference
LMS	Least Mean Squares
MAP	Maximum a Posteriori Probability
MLM	Max-Log-Map
MLP	Multi-Layer Perceptron
PDF	Probability Density Function
PLC	Power Line Communication
RF	Radio Frequency
RLS	Recursive Least Squares
RNA	Redes Neurais Artificiais
RSC	Recursive Systematic Convolutional
SaS	Symmetric Alpha-Stable
SER	Symbol Error Ratio
SFE	Soft-Feedback Equalizer
SISO	Soft-Input, Soft-Output
SNR	Relação Sinal-Ruído (do inglês, Signal-to-Noise Ratio)

1. Introdução

Agora no século XXI, o mundo nunca esteve tão conectado. É possível realizar uma conversa, ou tomar conhecimento de um acontecimento, entre pessoas que se encontram em lados opostos do globo terrestre e de uma maneira prática, eficiente e rápida. Diversos são os meios empregados que permitem a transmissão de informação como, por exemplo, cabos coaxiais, fibras óticas, sinais de RF (*Radio Frequency*). Cada um tem a sua topologia, estrutura e teoria que permitem o seu funcionamento. Um meio de transmissão que não foi primordialmente desenvolvido para transmissão de informação, mas que desde a metade dos anos 90 tem ganhado mais atenção e estudos voltados para este intuito, é a Rede Elétrica.

A comunicação utilizando a rede elétrica recebeu o nome de PLC (do inglês, *Power Line Communication*). A intenção é fazer com que o sistema elétrico não apenas distribua energia elétrica, mas também voz e dados. Uma vantagem na utilização dessa tecnologia é o fato de ser uma rede já amplamente projetada e implementada, sendo, então, de mais fácil acesso.

Como em qualquer sistema de comunicação, os sistemas PLC estão sujeitos a vários fenômenos que degradam seus desempenhos. Alguns exemplos são: interferência intersimbólica e presença de ruído impulsivo, os quais são fenômenos que ocorrem durante a passagem do sinal transmitido pelo canal, conforme pode ser visto no esquema da Figura 1. Tendo a intenção de mitigar esses efeitos deletérios, o presente trabalho irá estudar a eficiência de técnicas de equalização turbo, tanto linear quanto não-linear, no processo de recuperação da informação no receptor do sistema.

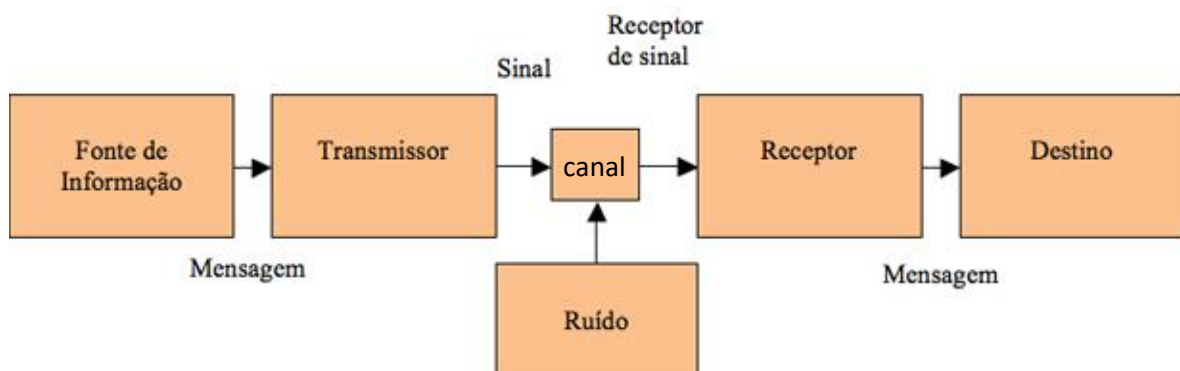


Figura 1. Esquema resumido do processo de transmissão e recepção de informação.

Para melhor compreender o funcionamento e limitações físicas do sistema PLC, a primeira parte deste trabalho apresenta um levantamento teórico dessa tecnologia. Além disso, também é feito um estudo sobre como operam decodificadores e equalizadores lineares e não-lineares, pois juntos realizam a equalização turbo.

Tendo o embasamento teórico, foi utilizado o software Matlab® para projetar um sistema PLC de transmissão e recepção de informação e simular a equalização turbo operando na recepção de dados que trafegaram pelo canal PLC projetado. Por meio dos resultados das simulações, foram averiguados quão eficientes podem, ou não, ser os equalizadores turbo na recuperação da mensagem original enviada por um canal PLC sujeito a ruídos impulsivos.

2. O que é um canal de comunicação PLC?

Os sistemas PLC começaram a receber maior atenção e pesquisas científicas por volta da metade da década de 90, mas já no início do século XX engenheiros suíços tentaram realizar comutação entre subestações e controle de carga na rede elétrica [01, 02]. Nas pesquisas atuais, o acesso à internet é o foco de grande parte dos trabalhos realizados nessa área.

Para começar os estudos desses sistemas, é interessante pesquisar e caracterizar o comportamento do canal de comunicação PLC. Como é dito por Parson em [03], e citado por Amirshahi em [01], o canal de propagação é o principal causador de muitos problemas e limitações enfrentados durante a transmissão de informação, pois suas características modificam a mensagem que por ele trafega, provocando interferência entre símbolos (ISI, do inglês *Intersymbol Interference*), atenuação de potência, entre outros.

Utilizando como referência o que Zimmermann e Dostert escreveram em [04], uma maneira de modelar o canal PLC seria considerando-o uma caixa-preta e descrevendo uma função de transferência equivalente $H(f)$. A estrutura do modelo é fundamentada em efeitos físicos, analisados por meio de várias medições, e não por meio de propriedades dos componentes que formam a rede, como cabos, conectores, etc., pois seria preciso conhecer as características e comportamentos físicos de todos esses componentes para poder configurar matrizes correspondentes às propriedades.

Na prática, é praticamente impossível determinar a quantidade necessária de parâmetros com suficiente precisão.

A modelagem feita em [04] é iniciada com a definição da topologia das principais redes e chega-se à conclusão que o melhor modelo a ser adotado é a topologia de barramento. Este tipo de topologia segue o princípio de que os equipamentos periféricos devem ser ligados a um ramo central que conduzirá as suas informações em certo sentido e para outras redes. De modo a simplificar a modelagem de um canal PLC, a topologia foi reduzida a apenas um ponto de acesso (Figura 2), pois foi considerado que a mesma ideia pode ser aplicada aos demais acessos da rede.

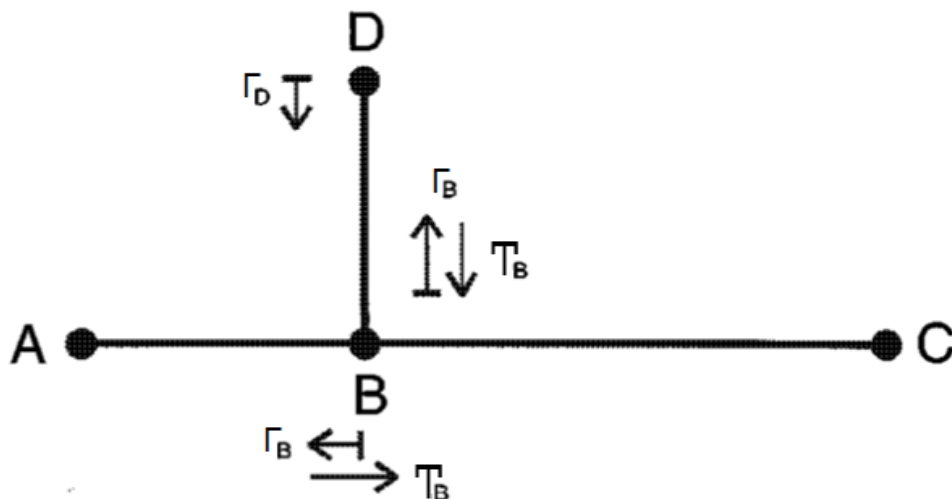


Figura 2. Topologia adotada para modelagem do canal PLC, sendo Γ o coeficiente de reflexão no ponto indicado (B ou D) e T o coeficiente de transmissão. Fonte: [04].

Pelo fato da topologia ser composta por linhas de transmissão, outras importantes considerações feitas foram a impedância característica dos cabos da rede (Z_L), que, por normalmente ser somente resistiva, tem apenas parte real, e a impedância da rede interna de uma residência vista pela rede ($Z_H(f)$), que é um valor complexo pelo fato de ser composta por equipamentos não somente resistivos. A presença de diferentes impedâncias na topologia implica na ocorrência de reflexão, total ou parcial, do sinal viajante, ocasionando o fenômeno de multipercurso. Também pode haver casamento entre impedâncias e nada ser refletido. Para mensurar o quanto do sinal é refletido ou transmitido existem dois parâmetros conhecidos, respectivamente, como Coeficiente de Reflexão (Γ) e Coeficiente de Transmissão (T) [13].

O casamento de impedâncias é considerado como o caso ideal quando se trabalha com linhas de transmissão, dado que a reflexão do sinal é nula, mas na prática nem sempre é possível possuir tal configuração. Portanto, é comum a ocorrência de multipercurso do sinal transmitido, podendo chegar a infinitos caminhos de propagação. Assim, é definido o primeiro fator que caracteriza a função $H(f)$: g_i . O fator g_i representa o produto dos coeficientes de reflexão e transmissão de cada caminho (i) possível de existir. Como as reflexões e transmissões de sinais, que dão origem a esses coeficientes, ocorrem nas junções de cabos, em [04] foi evidenciado que a impedância resultante nessas junções é sempre menor que a impedância característica do cabo principal. Portanto, tanto os coeficientes de reflexão, quanto os de transmissão, terão magnitude sempre menor ou igual a 1¹. A partir dessa explicação, ao analisar numericamente o fator g_i , infere-se que ele é, em geral, complexo e tem o módulo sempre menor ou igual a 1.

Ao trabalhar com multipercurso, outro parâmetro relevante é o *delay* (atraso) sofrido pelo sinal que percorreu o caminho i. Este atraso é dado por:

$$\tau_i = \frac{d_i \sqrt{\epsilon_r}}{c} = \frac{d_i}{v_p}, \quad (1)$$

onde, c é a velocidade da luz ($\sim 3 \times 10^8 \text{ m/s}$), d_i o comprimento do caminho i, ϵ_r a constante dielétrica do isolante que compõe a linha de transmissão e v_p a velocidade de propagação da onda eletromagnética no cabo.

Também advindo do estudo de linhas de transmissão, sabe-se que todo sinal pode sofrer atenuações ($A(f, d)$) ao trafegar por uma linha de transmissão. Esta atenuação aumenta com o comprimento do cabo e com a frequência do sinal. Após uma explicação acerca do parâmetro de atenuação, Zimmermann e Dostert concluem que este pode ser caracterizado como [04]:

$$A(f, d) = e^{-(a_0 + a_1 f^k)d}, \quad (2)$$

¹ Da teoria de Linhas de Transmissão, seja Z_0 a impedância característica e Z_c a impedância da carga, podendo ambas serem complexas. Os coeficientes de transmissão e reflexão são dados como:

$$\Gamma = \frac{Z_c - Z_0}{Z_c + Z_0} \quad T = 1 + \Gamma$$

onde, $a_0(m^{-1})$ indica quanto de atenuação pode ocorrer por metro, $a_1(s/m)$ indica quanto de atenuação a frequência de operação pode causar por metro, k é o expoente do fator de frequência, que segundo [02] tem valores entre 0,5 e 1, e d é o comprimento do cabo. Os parâmetros de atenuação a_0, a_1 e k são geralmente derivados de medições específicas, pois mesmo conhecendo as características do cabo em estudo, não é possível determiná-las [02].

Após todas as considerações e análises, a função de transferência do canal PLC é composta pelos parâmetros g_i e os apresentados em (1) e (2). A expressão matemática geral e simplificada da resposta em frequência do canal PLC para N multipercursos é exposta na equação (3).

$$H(f) = \sum_{i=1}^N g_i \cdot e^{-(a_0 + a_1 f^k) d_i} \cdot e^{-j2\pi f \left(\frac{d_i}{v_p}\right)} \quad (3)$$

2.1. Simulação de um canal PLC

Utilizando a equação (3) e os valores apresentados na Tabela 1, considerando 15 multipercursos, Zimmermann e Dostert obtiveram o espectro de amplitude do canal apresentado na Figura 3 a seguir.

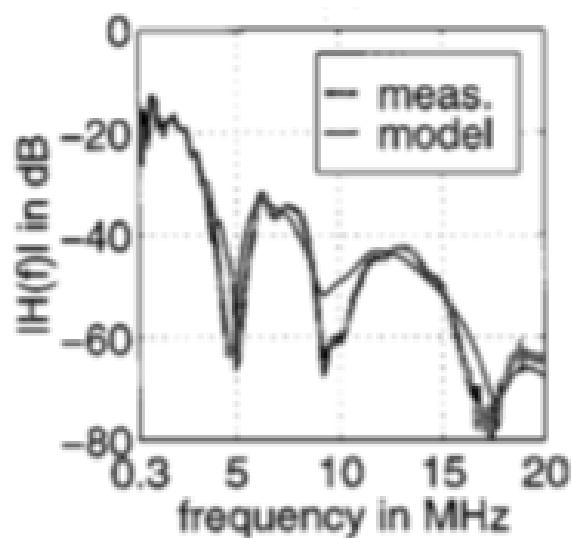


Figura 3. Modelagem do canal PLC de 15 caminhos feita em [04].

Vale a pena lembrar que os parâmetros de atenuação a_0, a_1 e k são determinados com base em análises de dados medidos, reais. Portanto, os valores apresentados na Tabela 1 foram adotados em [02] para que o espectro do canal PLC proposto em (3) pudesse estar o mais ajustado possível ao espectro obtido por meio de dados reais. Apesar do modelo proposto ser simples e dependente de certos ajustes, ele pode ser usado para simular o comportamento de um canal PLC, pois evidencia as principais características que são próprias deste tipo de canal.

Tabela 1. Parâmetros do modelo de 15 caminhos. Fonte: [04].

attenuation parameters					
$k=1$	$a_0=0$	$a_1=7.8 \cdot 10^{-10}$ s/m			
path-parameters					
i	g_i	d_i/m	i	g_i	d_i/m
1	0.029	90	9	0.071	411
2	0.043	102	10	-0.035	490
3	0.103	113	11	0.065	567
4	-0.058	143	12	-0.055	740
5	-0.045	148	13	0.042	960
6	-0.040	200	14	-0.059	1130
7	0.038	260	15	0.049	1250
8	-0.038	322			

A fim de simular o canal PLC, foi desenvolvida uma rotina em Matlab (Apêndice I) considerando (3), a Tabela 1 e o valor de ϵ_r igual a 4, seguindo a recomendação de [14] para o tipo de cabo NAYY 150 que é utilizado no artigo [04]. O resultado está apresentado na Figura 4.

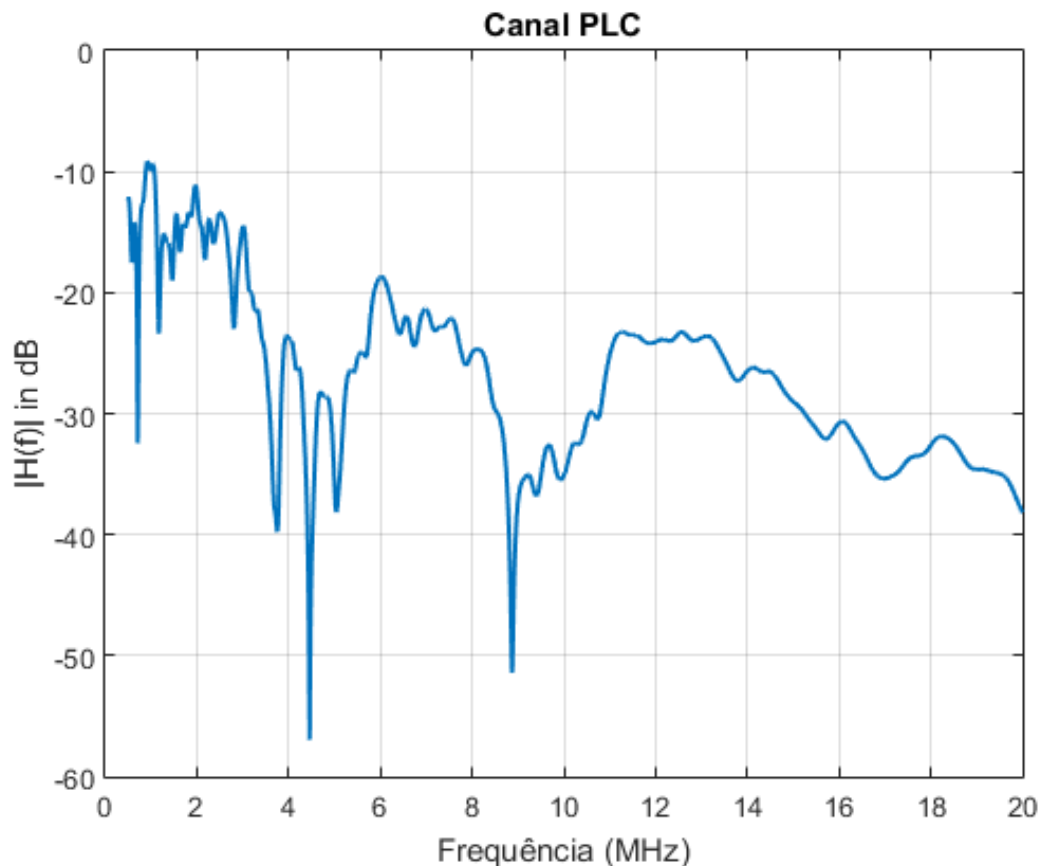


Figura 4. Resposta em frequência do canal PLC com 15 caminhos multipercurso.

Comparando esta figura com a Figura 3, notou-se que a modelagem em Matlab segue comportamento semelhante ao gráfico teórico até, aproximadamente, 15MHz, porém com amplitude superior. Realizando simulações com valores diferentes de ϵ_r , a_0 e a_1 foi possível ajustar a resposta do canal simulado para um formato mais semelhante ao espectro de canal PLC modelado em [04] e apresentado na Figura 3 quando ϵ_r era maior do que 4 e a_0 próximo de $0,01 \text{ m}^{-1}$. Com isso, inferiu-se que o tipo de cabo utilizado no artigo poderia ser diferente do considerado na simulação e que talvez o parâmetro a_0 não fosse realmente 0. Como se trata de um modelo que tenta traduzir a realidade por equações matemáticas e considerações iniciais, também seria válido considerar um resultado que utilizou valores próximos àqueles considerados inicialmente, dado que o resultado é semelhante e os novos valores refletem características de equipamentos reais. Portanto, neste trabalho foi considerado o canal apresentado na Figura 4 para realizar as simulações.

3. Ruído Impulsivo

Além das características intrínsecas do canal que distorcem o conteúdo do sinal enviado do transmissor para o receptor, outro fenômeno importante de ser estudado é o ruído, compreendido como um sinal aleatório indesejado presente no ambiente que acaba sendo somado à mensagem original. Sua origem pode se dar pela agitação térmica de cargas, quando o canal é um condutor elétrico, descargas eletromagnéticas causadas por fenômenos naturais ou equipamentos elétricos externos. Como seu comportamento é imprevisível, os estudos envolvendo processamento de sinal costumam generalizar e adotar o ruído branco – AWGN (do inglês, *Additive White Gaussian Channel*), que é compreendido como um sinal aleatório que afeta todas as frequências e modelado por meio da distribuição gaussiana (Figura 5), para realizar análises dos seus efeitos e, então, propor soluções.

Ao estudar o sistema PLC, notou-se que pelo fato desta tecnologia utilizar a rede elétrica como meio de transmissão, a interferência eletromagnética causada por dispositivos externos, ou fenômenos naturais, é sentida de forma mais intensa, como, por exemplo, um impulso elétrico de curta duração e de grande intensidade gerado por simples ligar/desligar da lâmpada de uma sala. A partir deste cenário surgiram os estudos sobre o *ruído impulsivo*, pois seu comportamento é mais específico e não mais gaussiano.

A Figura 5 exemplifica bem o que foi descrito no parágrafo anterior: à esquerda está apresentado um ruído com distribuição tipicamente gaussiana, no qual a amplitude do sinal varia continuamente no tempo, porém com intensidade bem inferior à amplitude máxima alcançada pelo ruído impulsivo; e à direita um ruído tipicamente impulsivo, no qual há a presença de somente alguns picos esporádicos de ruído, os quais têm grande amplitude se comparados ao nível máximo de intensidade que o ruído gaussiano alcança.

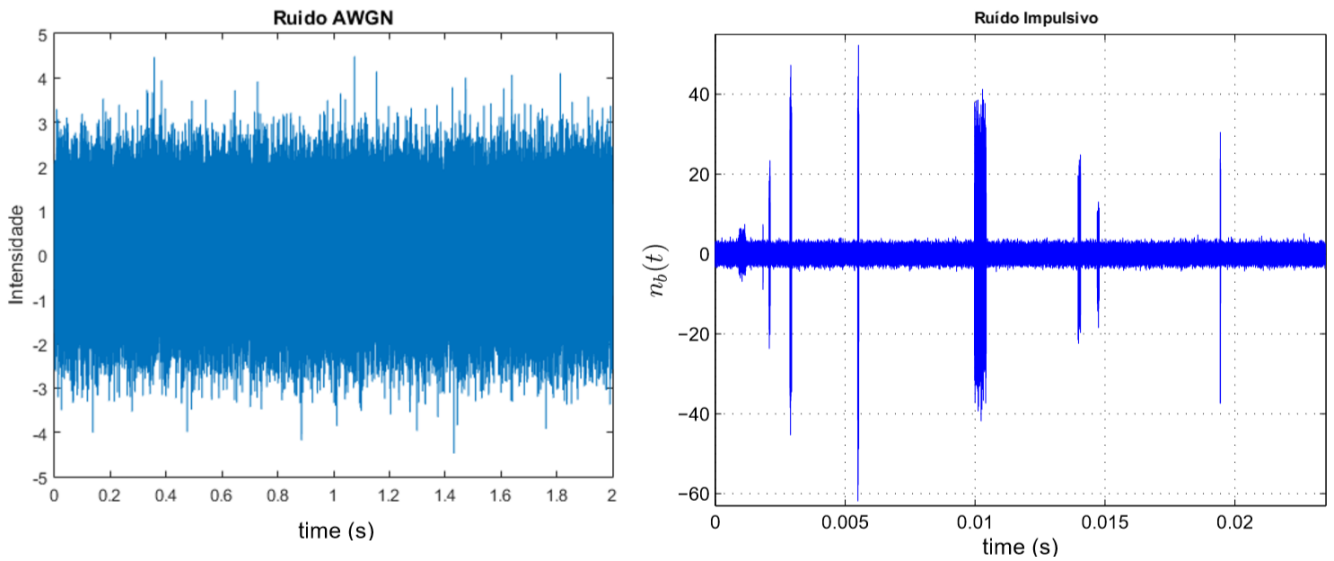


Figura 5. Exemplificação do comportamento do ruído AWGN (simulado em Matlab) e do ruído Impulsivo [17], no qual $n_b(t)$ representa a intensidade, ao longo do tempo.

Alguns dos estudos de modelagem matemática do ruído impulsivo foram feitos por Middleton, Zimmermann *et al*, Ghosh, Wang *et al* e Tsihrintzis *et al* em seus respectivos artigos [15, 16, 17, 18 e 19]. Como este trabalho não focou em um estudo mais profundo sobre a modelagem deste tipo de ruído, mas sim nos efeitos que tal fenômeno causaria na equalização turbo, foi escolhido o modelo de distribuição alfa-estável simétrica (SaS, do inglês, *Symmetric Alpha-Stable*) de Tsihrintzis *et al*, pelo fato de já ter sido estudado no trabalho de graduação do colega Marcelo Lopes [20] e seus resultados também terem sido analisados no período de levantamento bibliográfico do trabalho atual.

3.1. Distribuição alfa-estável simétrica

Nos estudos de probabilidade, a distribuição de uma função de densidade de probabilidade (*pdf*, do inglês, *Probability Density Function*) é dita como estável se uma variável aleatória com tal pdf satisfaz a seguinte propriedade [21]:

Definição 1: Uma variável aleatória X é estável se para outras variáveis independentes de mesma pdf X_1 e X_2 e para quaisquer constantes a e b positivas e não-nulas, $aX_1 + bX_2$ tem a mesma distribuição que $cX + d$, sendo c outra constante positiva e diferente de zero e $d \in \mathbb{R}$. Indo

além, se $d = 0$, então essa variável também pode ser classificada como simétrica.

As distribuições estáveis foram caracterizadas por Paul Lévy, em 1925 [22], após seus estudos sobre as somas de variáveis independentes e identicamente distribuídas. A partir de então, ficou evidente que distribuições deste tipo podem possuir características assimétricas e longas caudas (valores que se distanciam da média), mas suas propriedades matemáticas ainda são complexas e intrigantes, dificultando a elaboração de fórmulas fechadas. Apesar das dificuldades, há três casos especiais de distribuições estáveis que puderam ser bem determinados, dois deles antes mesmo dos estudos de Lévy, e são eles: Gaussiano (distribuição normal), distribuição de Cauchy e distribuição de Lévy.

A distribuição gaussiana pôde ser definida por meio do Teorema Central do Limite, pois este enuncia que a soma normalizada de um grande número de variáveis aleatórias independentes, igualmente distribuídas e de variância finita converge para uma distribuição gaussiana. Ao considerar variâncias tanto finitas quanto infinitas, o teorema apresentado anteriormente pode ser generalizado como Teorema Central do Limite Generalizado e o resultado da soma normalizada é uma distribuição alfa-estável [20]. Mesmo com a segunda constatação, não há expressões analíticas para as pdf's das distribuições alfa-estáveis, porém é possível definir a função característica $\varphi(t)$ de uma variável aleatória X , segundo [20]:

$$\varphi(t) = E \exp(itX) = \begin{cases} \exp(-\gamma^\alpha |t|^\alpha [1 - i\beta \operatorname{sign}(t) \tan(\frac{\pi\alpha}{2})] + i\delta t), & \text{para } \alpha \neq 1 \\ \exp(-\gamma |t| [1 + i\beta \operatorname{sign}(t) (\frac{2}{\pi}) \log|t|] + i\delta t), & \text{para } \alpha = 1 \end{cases} \quad (4)$$

Onde:

- α é um parâmetro que controla o tempo de declínio das caudas da função densidade de probabilidade e é limitado ao intervalo $0 < \alpha \leq 2$. Sendo que quando $\alpha = 2$, a distribuição alfa-estável se reduz à distribuição gaussiana. E quando $\alpha = 1$, o resultado é a distribuição de Cauchy.
- δ é o parâmetro de localização ($-\infty < \delta < \infty$). Corresponde à média para $1 < \alpha \leq 2$ e à mediana para $0 < \alpha \leq 1$.

- γ é o fator de dispersão ($\gamma > 0$), que determina o espalhamento da densidade ao redor do parâmetro de localização.
- β é o índice de simetria ($-1 \leq \beta \leq 1$). Quando $\beta = 0$, a distribuição é simétrica ao redor do parâmetro de localização.

Deixando os estudos mais aprofundados sobre esse modelo matemático para as referências [20] e [21] e focando na simulação de variáveis aleatórias com distribuição alfa-estável, [21] apresenta a seguinte equação:

$$X = \begin{cases} \frac{\sin(\alpha\theta)}{\cos(\theta)^{1/\alpha}} \left[\frac{\cos((\alpha-1)\theta)}{W} \right]^{\frac{1-\alpha}{\alpha}}, & \text{para } \alpha \neq 1 \\ \tan(\theta) & , \text{ para } \alpha = 1 \end{cases} \quad (5)$$

Onde $\theta = \pi(u_1(t) - 0,5)$, $W = -\log(u_2(t))$, $u_1(t)$ e $u_2(t)$ são variáveis aleatórias uniformemente distribuídas no intervalo $[0,1]$ e α é o mesmo coeficiente da equação (4), porém definido no intervalo $1 < \alpha < 2$.

Simulando a equação (5) no Matlab® através da rotina apresentada no Apêndice III, considerando apenas $\alpha > 1$ a fim de evitar o caso especial de Cauchy, foram obtidos os gráficos da Figura 6. Por meio desses gráficos, foi decidido adotar $\alpha = 1.1$ nas simulações deste trabalho, pois é a distribuição que mais se assemelhava como o comportamento de ruído impulsivo apresentado na Figura 5.

Uma questão importante abordada em [20] foi em relação à dispersão da distribuição alfa-estável (γ), pois este parâmetro pode ser usado como uma medida proporcional à potência do ruído. Após apresentar uma equação de SNR deduzida para um caso generalizado (GEbN0) e considerar a energia do sinal (E_b) unitária, Lopes apresentou a seguinte equação para γ [20]:

$$GEbN0 = \frac{E_b}{4\gamma^{\frac{2}{\alpha}}} \quad (6)$$

$$\gamma = \frac{1}{\sqrt{(4GEbN0)^{\alpha}}} \quad (7)$$

O fator de dispersão calculado em (7) será usado na Seção 9 para a definição da relação sinal-ruído do sistema.

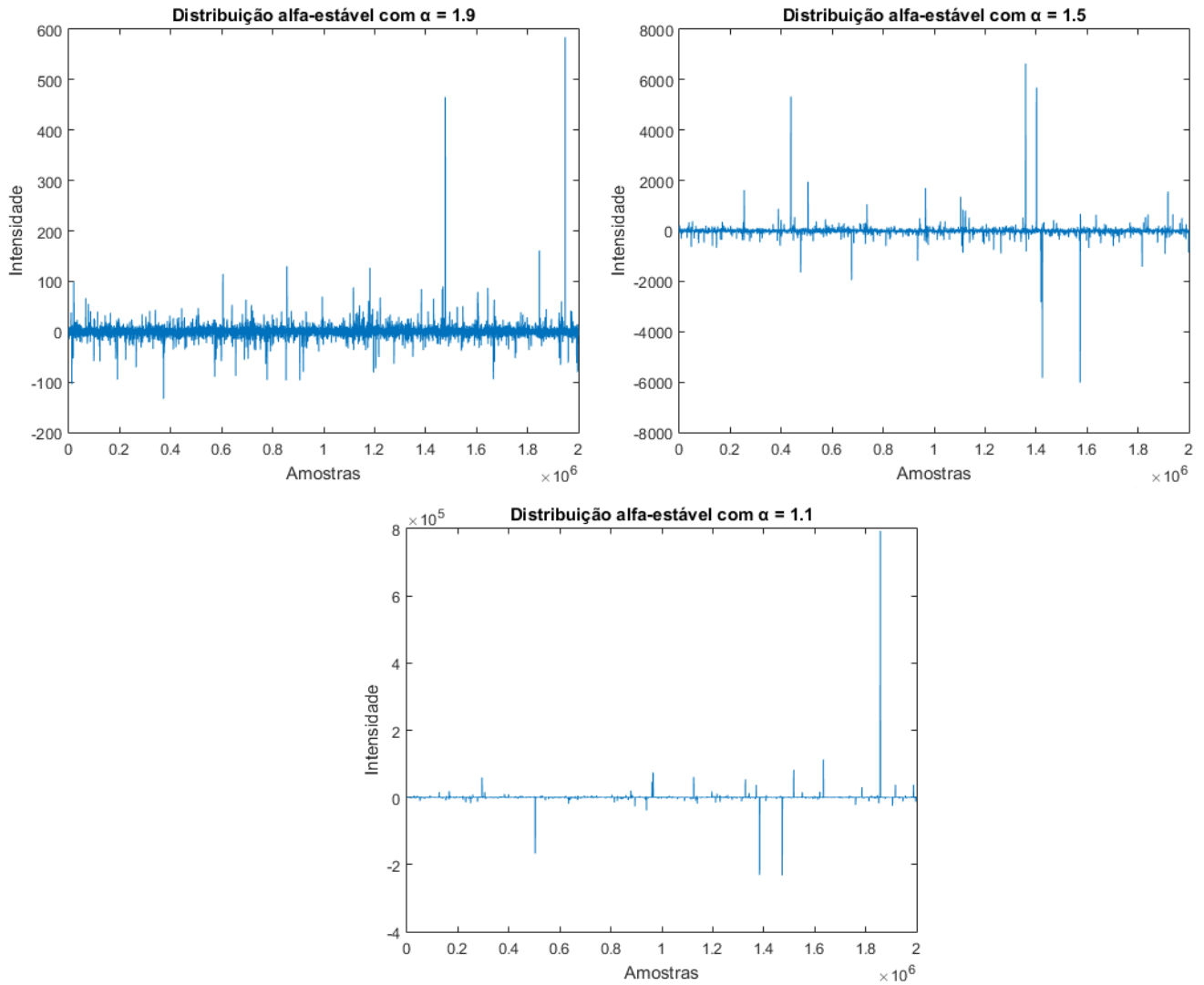


Figura 6. Distribuições alfa-estáveis com $1 < \alpha < 2$.

4. Processamento da informação pelo transmissor

Antes de prosseguir para o estudo da equalização turbo propriamente dita, é importante lembrar algumas etapas do processamento da informação que ocorrem antes do envio da mensagem pelo canal, ou seja, que ocorrem no transmissor. A Figura 7 expõe os dois processos que foram considerados no projeto do transmissor da simulação deste trabalho: Codificação de canal, do tipo RSC (do inglês, *Recursive Systematic Convolutional*), e entrelaçamento de dados.

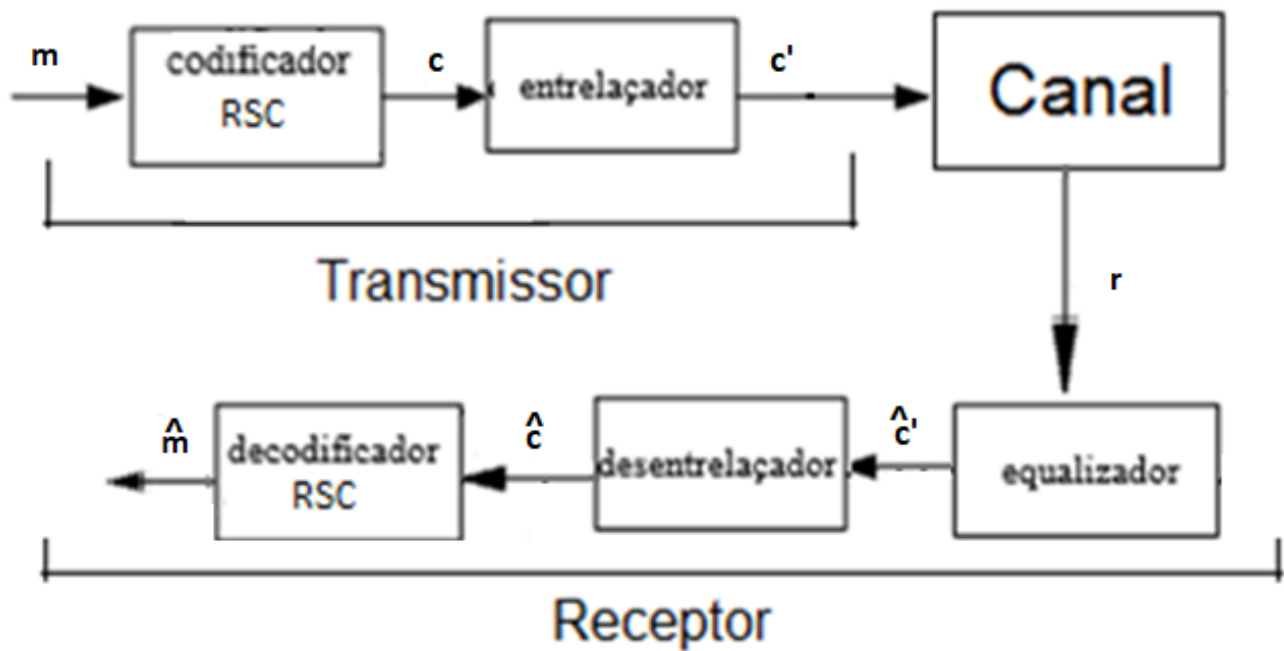


Figura 7. Diagrama das etapas consideradas no processo de transmissão e recepção de informação. Cada letra denota a forma como a mensagem se encontra dentro do sistema: m - forma original, c - codificada, c' - codificada e entrelaçada, r - passada pelo canal, \hat{c} - equalizada, \hat{c} - desentrelaçada e \hat{m} - recuperada no receptor.

Antes de prosseguir com a explicação sobre o codificador e entrelaçador, vale a pena ressaltar que em um processo de transmissão de informação é imprescindível a presença de um modulador para enviar os dados fisicamente pelo canal, pois é nele em que ocorre a transposição da informação codificada para as ondas eletromagnéticas por meio da manipulação de sua fase, amplitude e/ou frequência. Porém, como este não era o foco deste estudo, assumiu-se que a modulação, bem como a demodulação no receptor, ocorria sem erros e era do tipo BPSK (do inglês, *Binary Phase Shift Key*). Como o próprio nome da modulação já diz, só há dois símbolos possíveis em sua constelação. Portanto, é possível notar que neste trabalho as definições de bit e de símbolo se mesclam.

4.1. Codificador RSC

Em sistemas que trabalham com o processamento digital de sinais, um vídeo, um som ou um texto são “traduzidos” para uma sequência de bits (1 e 0) para que possam ser tratados por equipamentos eletrônicos. Esse processo recebe o nome de codificação de fonte. Em um mundo ideal, se há o desejo de compartilhar esse

vídeo ou texto, bastaria enviar para o destinatário essa mesma sequência de bits (comumente intitulada por *mensagem*), mas, como já visto na seção 2, o canal por onde tráfegará essa mensagem, seja ele qual for, lhe provocará interferências, ou seja, alterações no conteúdo, e ainda há a presença de ruídos externos. A fim de proteger a mensagem desses efeitos, foi desenvolvida a codificação de canal. Este tipo de codificação realiza a inserção de bits redundantes para que possam ajudar na detecção e correção de erros dos bits recebidos pelo receptor. Há duas principais classes de códigos corretores de erros: os códigos de bloco e os códigos convolucionais. Maiores informações sobre códigos de bloco podem ser encontradas em [10] e [12], pois neste trabalho foi estudada apenas a parte de codificação convolucional.

Com base na explicação feita por Proakis em [10], a codificação convolucional pode ser compreendida como a passagem dos bits de informação por um registrador composto por K estados finitos, implicando em uso de memória, e n funções algébricas lineares. Para cada k -bits recebidos na entrada do codificador, n bits são devolvidos para recompor a mensagem e, portanto, é possível associar uma taxa R para um codificador convolucional dada por:

$$R = \frac{k}{n} \quad (8)$$

Utilizando como exemplo um codificador formado por dois elementos de memória (elementos de atraso, $m(k - K)$), representados pelos quadrados da Figura 8, duas funções algébricas que utilizam a adição módulo-2, com apenas 1 bit de entrada e considerando que os elementos de memória estão inicialmente zerados, qual seria a codificação da sequência 10100? A Tabela 2 apresenta quais os valores de $m(k)$, $m(k-1)$, $m(k-2)$, $c1$ e $c2$ a cada passagem de tempo k para que a sequência de entrada fosse codificada.

Tabela 2. Passo-a-passo do processo de codificação.

k	Bit de entrada $m(k)$	$m(k-1)$	$m(k-2)$	$c1$ $(m(k-1) \oplus m(k-2))$	$c2$ $(m(k) \oplus m(k-2))$	Codificação de $m(k)$
0	1	0	0	0	1	01
1	0	1	0	1	0	10
2	1	0	1	1	0	10

3	0	1	0	1	0	10
4	0	0	1	1	1	11
5	-	0	0	-	-	-

Por meio da Tabela 2 é fácil visualizar o deslocamento dos bits pela memória do codificador e concluir que a sequência codificada seria: 0110101011. Indo mais além, pelo fato deste codificador possuir somente 2 elementos de memória, há 4 estados possíveis pelos quais esses elementos juntos podem passar e são eles: 00, 01, 10 e 11. Com isso, também é possível notar que a sequência de bits de mensagem (10100) proporcionou a volta ao estado inicial do codificador ($S_0 = 00$).

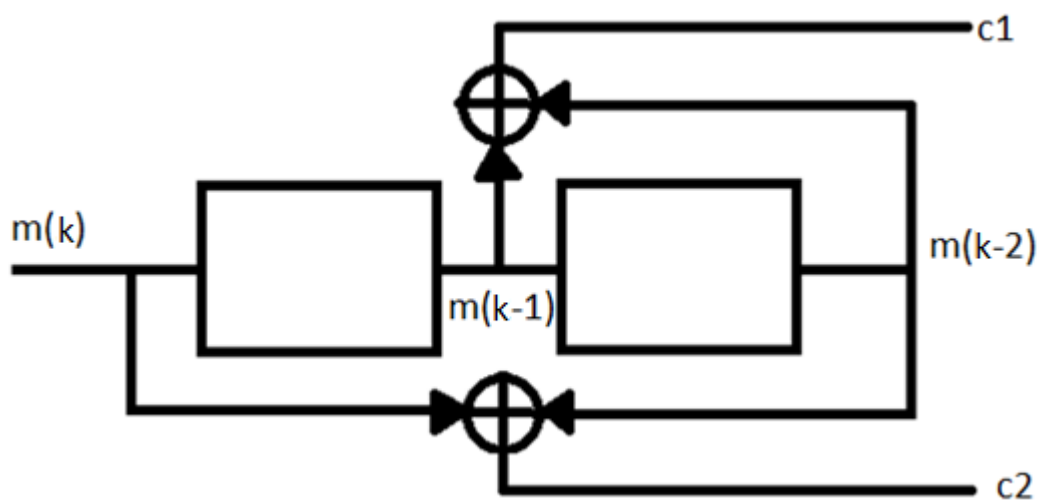


Figura 8. Exemplo de codificador convolucional não sistemático.

Outra informação que pode ser retirada do codificador da Figura 8 é a classificação de *codificador não sistemático*, pois suas saídas não possuem diretamente o bit de entrada. Caso isso acontecesse, o codificador seria classificado como *sistemático*.

Conforme explicado em [11], as saídas c_1 e c_2 do codificador são resultado da convolução entre a entrada m e as respostas ao impulso correspondentes às funções algébricas determinadas para cada uma delas. Para saber qual a resposta ao impulso, basta ter como entrada a sequência $m = 100 \dots$. Voltando para o exemplo exposto na Figura 8, as respostas ao impulso q de cada saída são: $q_1 = [0 \ 1 \ 1]$ e $q_2 = [1 \ 0 \ 1]$.

Como já foram introduzidos os conceitos de estados do codificador, vale a pena também expor sobre outras formas que um codificador pode ser representado, e são eles: o Diagrama de Estados e o Diagrama em Treliça.

Para o diagrama de estados, primeiramente, os estados são evidenciados e setas são utilizadas para indicar todas as possíveis mudanças de estado. Em seguida, ao lado das setas são escritos qual o bit de entrada e o resultado das funções algébricas que regem tal codificador, ficando na forma: (bit)/(resultado). A Figura 9 apresenta o diagrama de estados correspondente ao exemplo de codificador adotado na Figura 8.

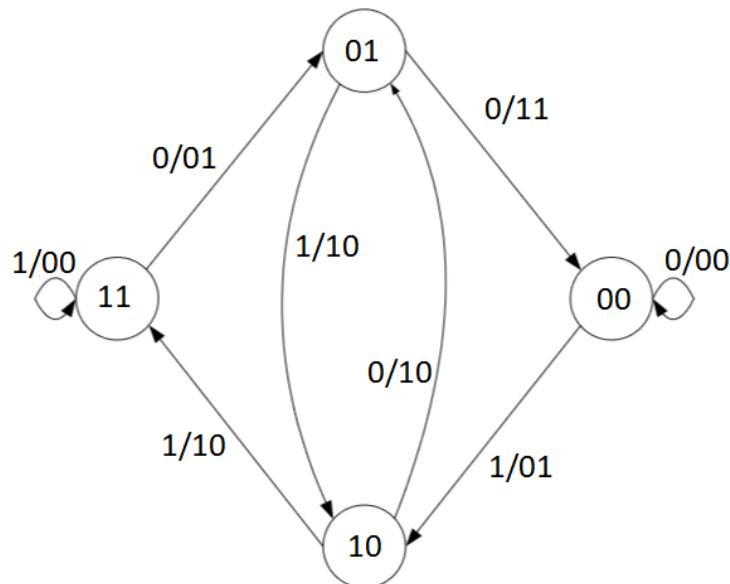


Figura 9. Diagrama de Estados do codificador da Figura 8.

O diagrama de estados resume todas as mudanças de estado e saídas que o codificador pode ter, facilitando, assim, a obtenção da sequência codificada final. Já o diagrama em treliça tem a intenção de exibir o comportamento do codificador ao longo do tempo. De imediato pode parecer redundante e mais complexo do que o diagrama de estados, porém a sua estrutura tem maior valia durante o processo de decodificação. A Figura 10 apresenta o diagrama em treliça do codificador tomado como exemplo. Nele, as retas azuis representam a chegada de um bit 1 e as pretas a entrada de um bit 0 e os valores dispostos ao lado delas são, assim como no diagrama de estados, os resultados das funções algébricas do codificador.

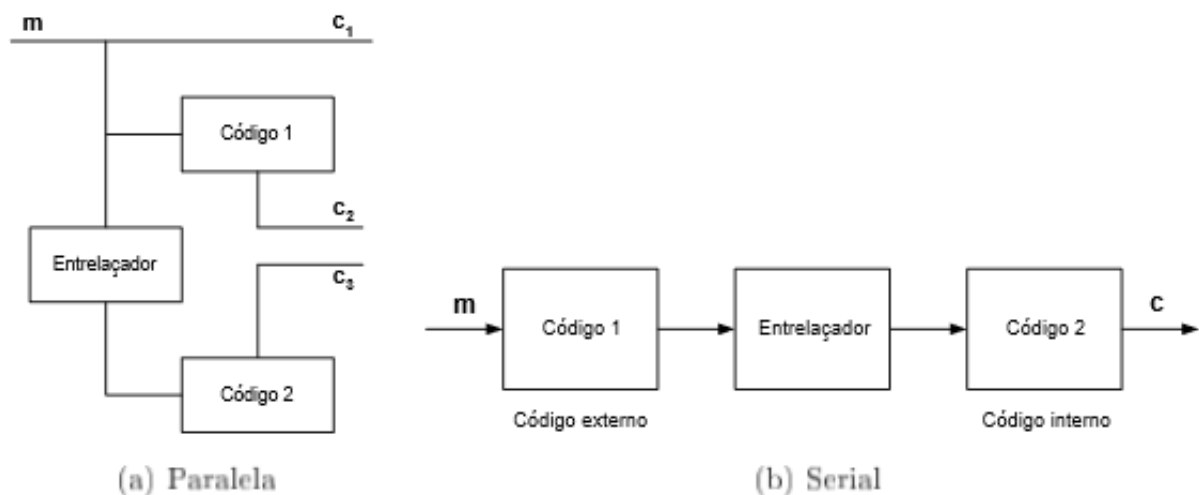


Figura 11. Representação dos tipos de combinações possíveis de uma codificação turbo. Fonte: [11].

A função do entrelaçador é permutar de forma determinística a ordem de uma sequência de bits, ou símbolos, que lhe é passada. Apesar de parecer algo simples, há duas importantes razões que justificam o seu uso, conforme explica Haykin [12]:

- “- Atar os erros que são facilmente cometidos em uma metade do código turbo aos erros que são excepcionalmente improváveis de ocorrer na outra metade.
- Garantir um desempenho robusto em relação à decodificação não equivalente, que é um problema que surge quando as estatísticas do canal não são conhecidas ou foram especificadas incorretamente.”

Eis que neste cenário foram propostos os RSC. Um código RSC é, como o próprio nome já diz, um código convolucional recursivo, ou seja, uma ou mais saídas do codificador alimentam a sua entrada de dados, conforme exemplo da Figura 12. E o ganho que a realimentação proporciona ao codificador é refletido na melhoria de desempenho do ponto de vista de codificação global [12], pois afeta o comportamento dos padrões de erro, impedindo que um erro no bit sistemático seja propagado para os bits de paridade.

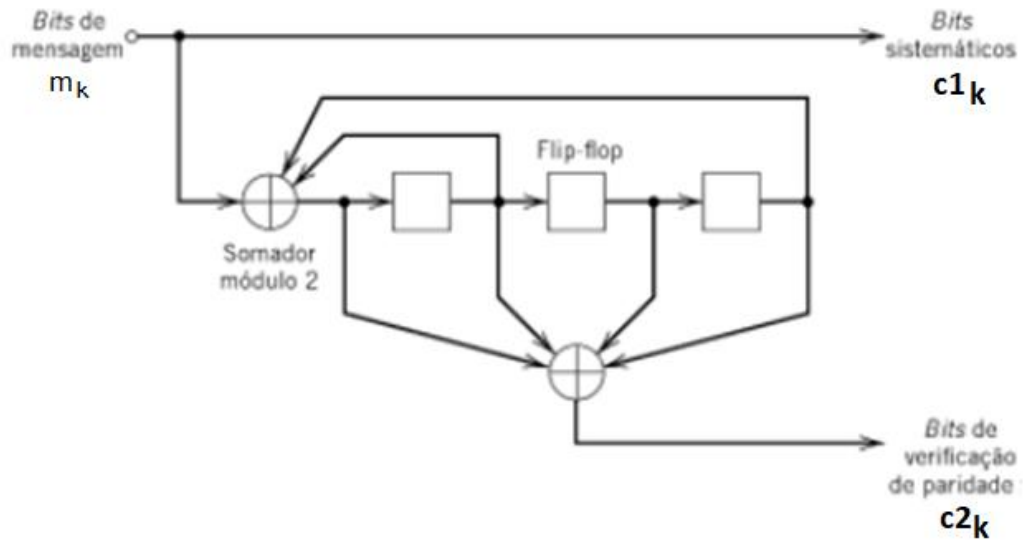


Figura 12. Exemplo de codificador RSC. Fonte: [12].

Neste trabalho foi utilizado um codificador RSC com os polinômios $q_1 = [1 \ 1 \ 1]$ e $q_2 = [1 \ 0 \ 1]$, resultando na estrutura da Figura 13.

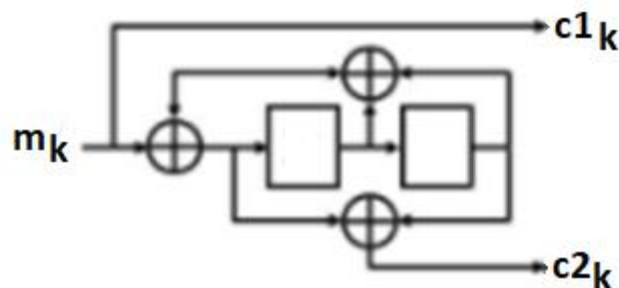


Figura 13. Codificador RSC utilizado no projeto.

Uma vez apresentados os componentes do transmissor, nas seções seguintes serão apresentados os blocos formadores do sistema de recepção considerado neste trabalho.

5. Equalização de canal

Dando prosseguimento às etapas apresentadas na Figura 7, este é o momento de estudar o processo de equalização.

As distorções provocadas no sinal transmitido pelo canal e ruídos externos podem causar entendimentos errôneos dos símbolos quando estes forem

decodificados no receptor. Portanto, faz-se necessário realizar algum processo que diminua, ou até retire, essas distorções. Para realizar uma compensação dos efeitos do canal há os Equalizadores. Em outras palavras, um equalizador deve ser capaz de produzir um sinal $\hat{m}(k)$ que seja o mais idêntico possível ao sinal $m(k)$ que existia antes de passar pelo canal, conforme pode ser visto no esquema da Figura 14 e lembrando que as etapas de codificação foram desconsideradas nessa análise.

Os equalizadores podem ser classificados em dois tipos [06]: adaptativos ou não adaptativos. Os não adaptativos são aqueles que efetuam compensações de forma preestabelecida, ou seja, é formado com informações que já são conhecidas do canal. Já os adaptativos, como o próprio nome sugere, têm a capacidade de verificar as características do canal de transmissão e, então, atualizar os seus coeficientes para conseguir, da melhor forma possível, compensar os efeitos do canal. Um dos modos de adaptação se dá com a comparação entre o sinal equalizado $\hat{m}(k)$ e o sinal original sem que tenha passado pelo canal $m(k)$, de modo que o erro $e(k)$ entre eles tenda à zero, conforme exemplifica o exemplo da Figura 14. Esse tipo de processo recebe o nome de Equalização Adaptativa Supervisionada. Há outros tipos de adaptação, como a autodidata [06].

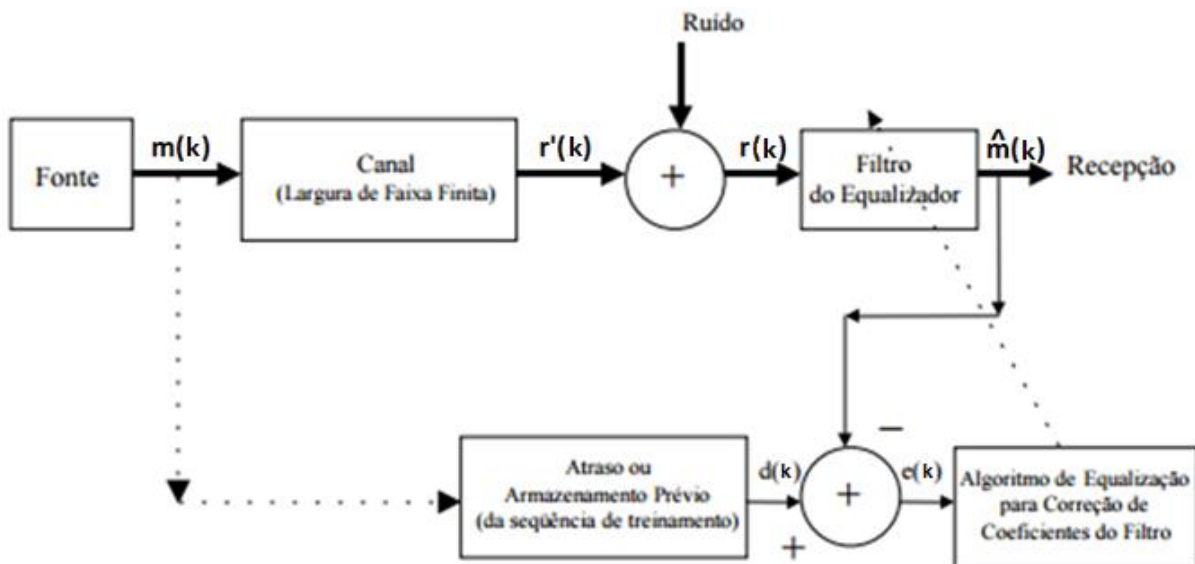


Figura 14. Esquema básico do processo de equalização adaptativa supervisionada.

Fonte: [06].

Dentro dos estudos com estruturas lineares, há algoritmos bem conhecidos, e que já foram codificados em Matlab no trabalho [08], como o LMS (*Least Mean Squares*) e RLS (*Recursive Least Squares*), para adaptar os coeficientes do filtro

equalizador. Com o avanço da computação e seus estudos, houve a criação da teoria de Inteligência Artificial (AI) [24], que tem como pressuposto o funcionamento de mecanismos e softwares de um modo similar à inteligência humana e a sua utilização no processo de equalização pode ser mais eficiente, pois trata o problema com considerações que muitas vezes vão além de considerações lineares. Na próxima subseção será explicada uma das metodologias empregadas em sistemas inteligentes.

5.1. Redes Neurais Artificiais

Dentro da área de sistemas inteligentes, há o *Multi-Layer Perceptron* (Perceptron de múltiplas camadas, em português), abreviado como MLP, e é compreendido como uma generalização de um *perceptron* simples, que pode ser visto como uma representação de um neurônio biológico. Esta inspiração biológica surgiu após a percepção de que animais são capazes de reagir a mudanças em seu meio ambiente, adaptando seu sistema nervoso para modificar seu comportamento, conforme exposto em [07]. Trazendo esta ideia para o processamento computacional, foi elaborada a teoria de “Redes Neurais Artificiais” (RNA), na qual o processamento básico de informação ocorre em diversas unidades simples, denominadas *neurônios artificiais*, que estão interconectadas, conforme pode ser visto na Figura 15.

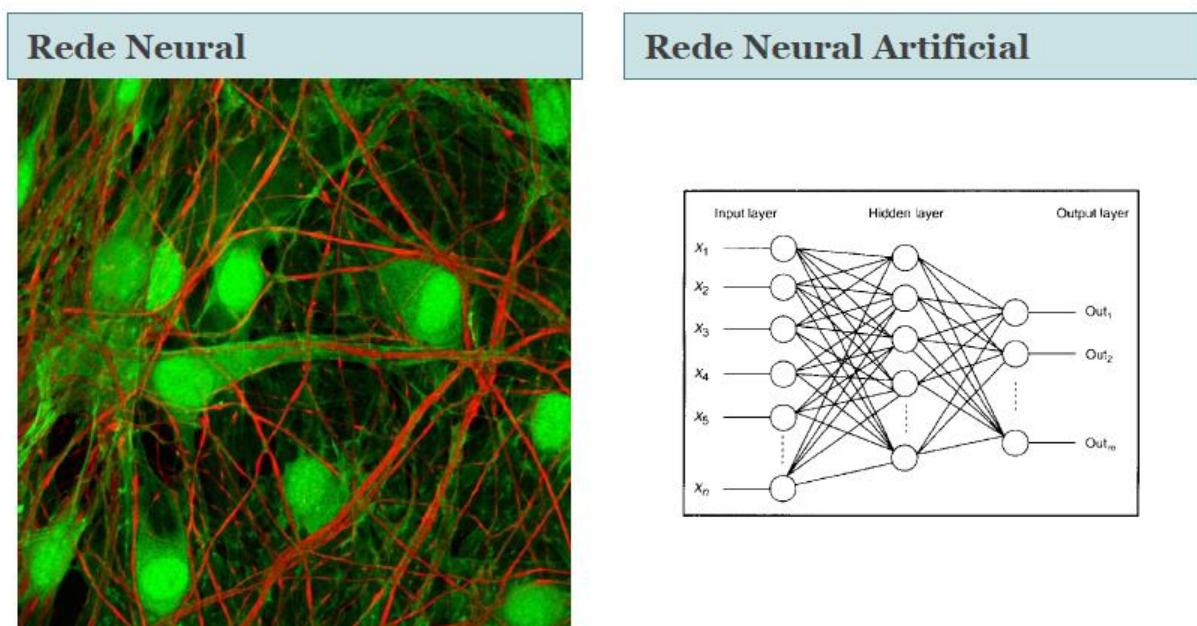


Figura 15. Exemplo de uma rede neural biológica (esquerda) e artificial (direita). Fonte: [07].

A definição inicial do *perceptron*, que é uma unidade simples da rede, dizia que as entradas desse neurônio eram “calibradas” por pesos w , somadas e depois uma função de ativação, que era linear, determinava a saída, conforme mostra a Figura 16. Seu aprendizado, refletido na atualização dos pesos w , seguia a regra:

$$w(n + 1) = w(n) + \eta * e * x(n), \quad (9)$$

onde $e = \hat{m}(n) - m(n)$.

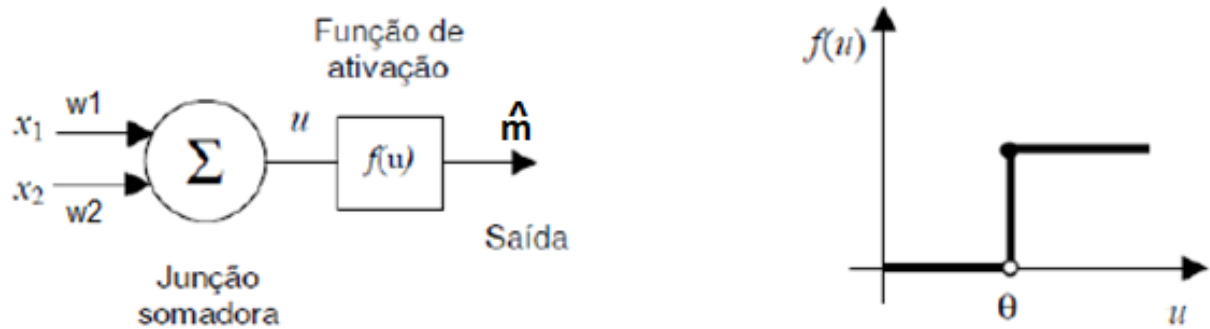


Figura 16. Definição básica de um perceptron. Fonte: [07].

Adotando um modelo mais geral de perceptron (Figura 17), que é a premissa do MLP, o que muda é o tipo de função de ativação do neurônio e há o acréscimo de um limiar b_k que tem o papel de aumentar ou diminuir a influência do valor da entrada líquida, ou seja, após todas as entradas somadas. Essa nova determinação da função tem como propósito limitar a saída do neurônio e introduzir não linearidade no modelo.

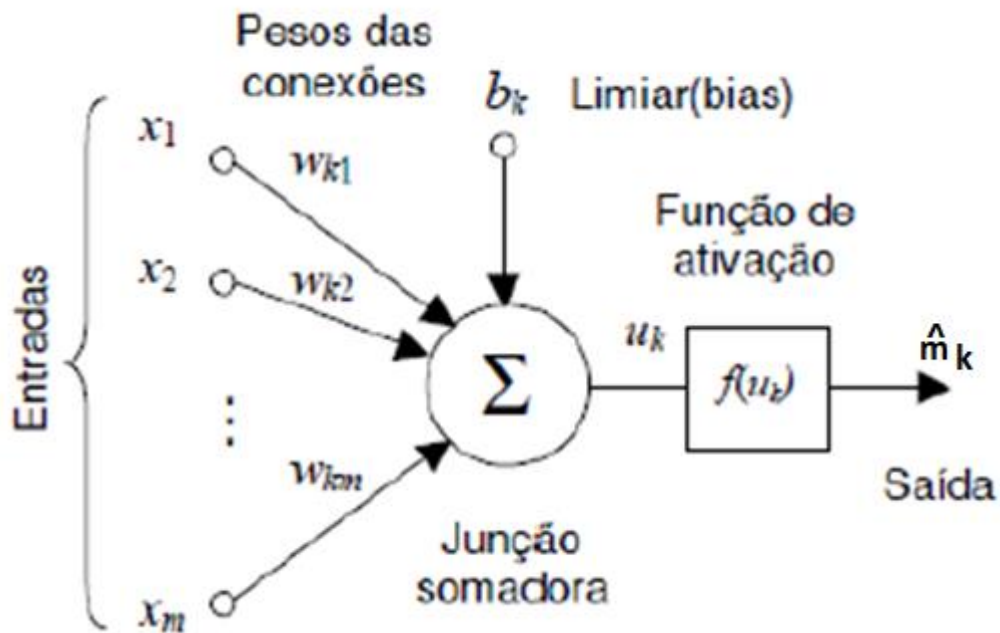


Figura 17. Modelo genérico de um neurônio artificial.

Após estas apresentações, vale comentar que o conhecimento da rede neural está armazenado nos pesos das conexões, ou seja, nas variáveis w . Então, o processo de aprendizagem corresponde a identificar um conjunto apropriado de pesos de forma que a rede se comporte como desejado [07]. A ideia de atualização dos pesos w segue o mesmo princípio dos algoritmos estudados em teoria de filtragem adaptativa, o que diferencia é a utilização de funções não-lineares após a combinação dos sinais de entrada do neurônio.

Voltando para as considerações do MLP, esta rede possui três características principais:

- A rede neural pode ser formada por camadas intermediárias e cada camada pode ter mais de um neurônio.
- Os neurônios das camadas intermediárias possuem uma função de ativação não-linear do tipo sigmoidal como, por exemplo, tangente hiperbólica.
- A rede possui um alto grau de conectividade.

A seguir um exemplo da estrutura geral de um MLP:

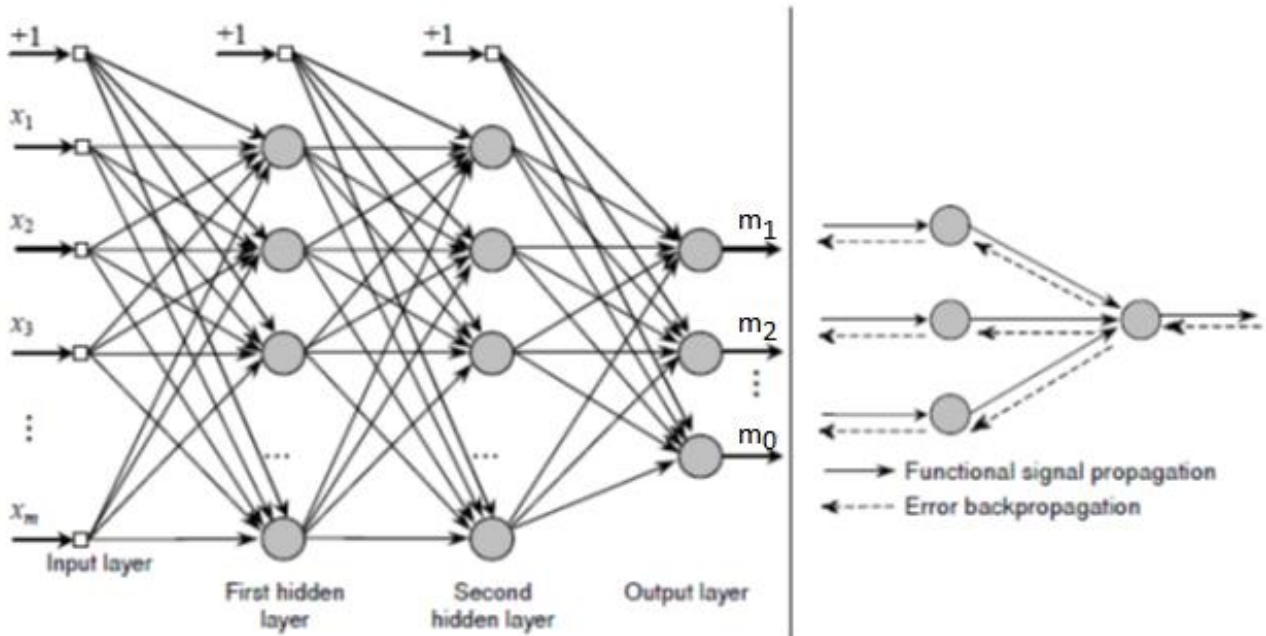


Figura 18. Exemplo de estrutura geral de um MLP (esquerda) e operação do algoritmo de treinamento. Fonte: [07].

No Apêndice II, há uma breve apresentação sobre funções específicas existentes no Matlab que permitem criar e organizar uma rede neural de maneira simples e rápida.

6. Decodificação de Canal

Após o tratamento do sinal recebido pelo equalizador, é chegada a etapa de decodificação. O intuito básico desta etapa é ser capaz de recuperar a mensagem original com o auxílio dos bits de paridade inseridos na codificação.

Em se tratando de códigos convolucionais, um dos critérios mais comumente utilizado para sua decodificação é conhecido como MAP (do inglês, *Maximum a Posteriori Probability*). Nele tem-se a intenção de minimizar a probabilidade de erro de bit entre a palavra código deduzida (\hat{c}) e a palavra código advinda do transmissor (c). Em [11], ainda é explicado que tal relação pode também ser atendida quando a escolha de \hat{c} é feita de forma a maximizar a probabilidade a posteriori (estimativa feita após o recebimento de toda a sequência em análise) de c , conforme descrito na equação (10).

$$\hat{c} = \arg \max_c P(c|r), \quad (10)$$

onde r é a informação advinda do canal, podendo ou não estar equalizada.

Com o auxílio da regra de Bayes (11), a equação (10) pode ser reescrita como a equação (12).

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (11)$$

$$\hat{c} = \arg \max_c p(r|c)P(c), \quad (12)$$

onde $p(r|c)$ é uma típica função de verossimilhança, a qual permite inferir sobre o seu valor a partir de um conjunto de informações e em um sentido inverso à probabilidade condicional.

A implementação computacional do decodificador MAP mais usada atualmente foi proposta por Bahl et al em [23]. Portanto, o algoritmo de implementação MAP é conhecido por BCJR (*Bahl, Cocke, Jelinek e Raviv*, nome dos autores do algoritmo). Antes de dar continuidade ao estudo do BCJR, é importante explicar brevemente sobre a diferença entre decodificação abrupta (*Hard Decoding*) e decodificação suave (*Soft Decoding*). A primeira trabalha com entradas e saídas quantizadas (bits), enquanto que na segunda as entradas e saídas não são quantizadas, mas sim associadas às informações probabilísticas de serem ou não certo símbolo.

A partir da explicação anterior e do gráfico apresentado em [11], comparando o desempenho desses dois tipos de decisão (Figura 19), pode-se notar que a decodificação suave tem melhor desempenho do que a abrupta, pois alcança menor taxa de erro para um mesmo valor de SNR (E_b/N_0 , do inglês, *Signal-to-Noise Ratio*), que neste caso relaciona a energia dos bits de mensagem com a densidade espectral de potência do ruído.

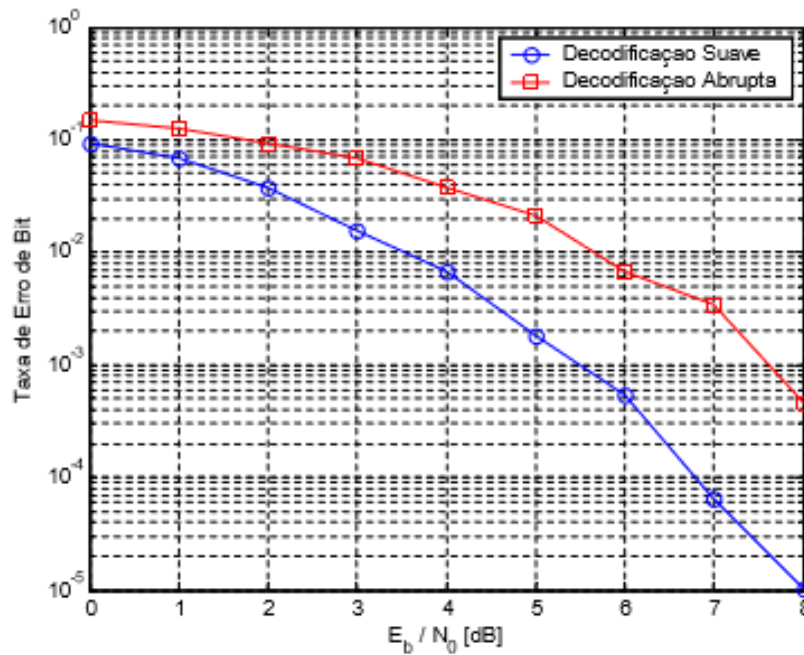


Figura 19. Comparação entre decodificação abrupta e suave. Fonte: [11].

Outra forma de calcular as decisões suaves é por meio da LLR (do inglês, *Log-Likelihood Ratio*), também compreendida como o logaritmo da razão de verossimilhança, conforme expõe a equação (13) [11]:

$$L(m_k) \triangleq \ln \frac{P(m_k=+1|r)}{P(m_k=-1|r)}, \quad (13)$$

onde m_k é o bit da mensagem no instante k , r é a sequência recebida e os valores +1 e -1 vêm do fato do trabalho considerar apenas a modulação BPSK.

Voltando para o algoritmo MAP, ou BCJR, ele aceita como entrada estimativas a priori dos bits que chegam e os próprios bits que se deseja decodificar e retorna uma LLR. Com isso, sua implementação na decodificação tem origem na equação (13), que pode ser reescrita como [11]:

$$L(m_k) \triangleq \ln \frac{\sum_{\{(s',s):m_k=+1\}} p(s',s,r)}{\sum_{\{(s',s):m_k=-1\}} p(s',s,r)}, \quad (14)$$

sendo s' e s os estados do codificador nos instantes $k - 1$ e k do seu diagrama em treliça e $p(s', s, r)$ a probabilidade conjunta entre os termos s' , s e r .

Em [11] foi explicado que as probabilidades conjuntas podem ser escritas como o produto de três termos independentes, uma vez usadas as propriedades de um processo de Markov, conforme exposto abaixo [11]:

$$\begin{aligned} p(s', s, r) &= p(s', r_{j < k}) \cdot p(s, r | s') \cdot p(r_{j > k} | s) \\ &= \underbrace{p(s', r_{j < k})}_{A_{k-1}(s')} \cdot \underbrace{P(s | s') \cdot p(r_k | s', s)}_{C_k(s', s)} \cdot \underbrace{p(r_{j > k} | s)}_{B_k(s)}, \end{aligned} \quad (15)$$

Nesta equação, $r_{j < k}$ representa a sequência de símbolos recebidos do instante inicial até o instante $k-1$ e $r_{j > k}$ é a sequência entre o instante $k+1$ e o final da treliça.

Aplicando (15) em (14), chegou-se à seguinte expressão da LLR para o algoritmo MAP:

$$L(m_k) = \ln \frac{\sum_{\{(s', s): m_k = +1\}} A_{k-1}(s') \cdot C_k(s', s) \cdot B_k(s)}{\sum_{\{(s', s): m_k = -1\}} A_{k-1}(s') \cdot C_k(s', s) \cdot B_k(s)}, \quad (16)$$

Desmembrando cada um dos três coeficientes [11]:

$$A_k(s) = \sum_{s'} C_k(s', s) \cdot A_{k-1}(s') \quad (17a)$$

$$B_{k-1}(s') = \sum_s C_k(s', s) \cdot B_k(s) \quad (17b)$$

$$C_k(s', s) = P(s | s') \cdot p(r_k | s', s) \quad (17c)$$

As equações (17a) e (17b) representam cálculos recursivos *forward* e *backward*, respectivamente, $P(s | s')$ é a probabilidade de transição de estados e $p(r_k | s', s)$ a probabilidade condicional de r_k dado os estados s' ou s . Análises e discussões detalhadas a respeito das equações constituintes do decodificador BCJR podem ser encontradas em [11].

Indo um pouco mais além, com a intenção de reduzir a complexidade computacional do algoritmo MAP (que é exponencial em relação ao número de elementos de memória do codificador), foi proposto o decodificador Log-MAP, o qual

utiliza logaritmos ao longo do algoritmo. Essa simples consideração já proporciona que multiplicações se tornam adições e isso acelera o processamento.

Lembrando que a codificação considerada neste trabalho é do tipo turbo e que a sua estrutura concatena um ou mais códigos, conforme explicado anteriormente e exemplificado na Figura 11, esse mesmo princípio deve ser adotado na decodificação, visto que deve ser mantida a paridade CODEC (Codificador-Decodificador) para que ocorram as traduções corretas, ou seja, equivalentes, da informação e consiga-se obter uma mensagem que seja a mais parecida possível com a original. Com isso, as estruturas básicas de um decodificar turbo são apresentadas na Figura 20, e já acrescida nestas estruturas está a consideração de um processo iterativo, ou de realimentação da informação. Conforme explicado em [11], a realimentação auxilia no refinamento da decisão final de todo o conjunto do decodificador, por conta da concatenação de códigos feita no codificador.

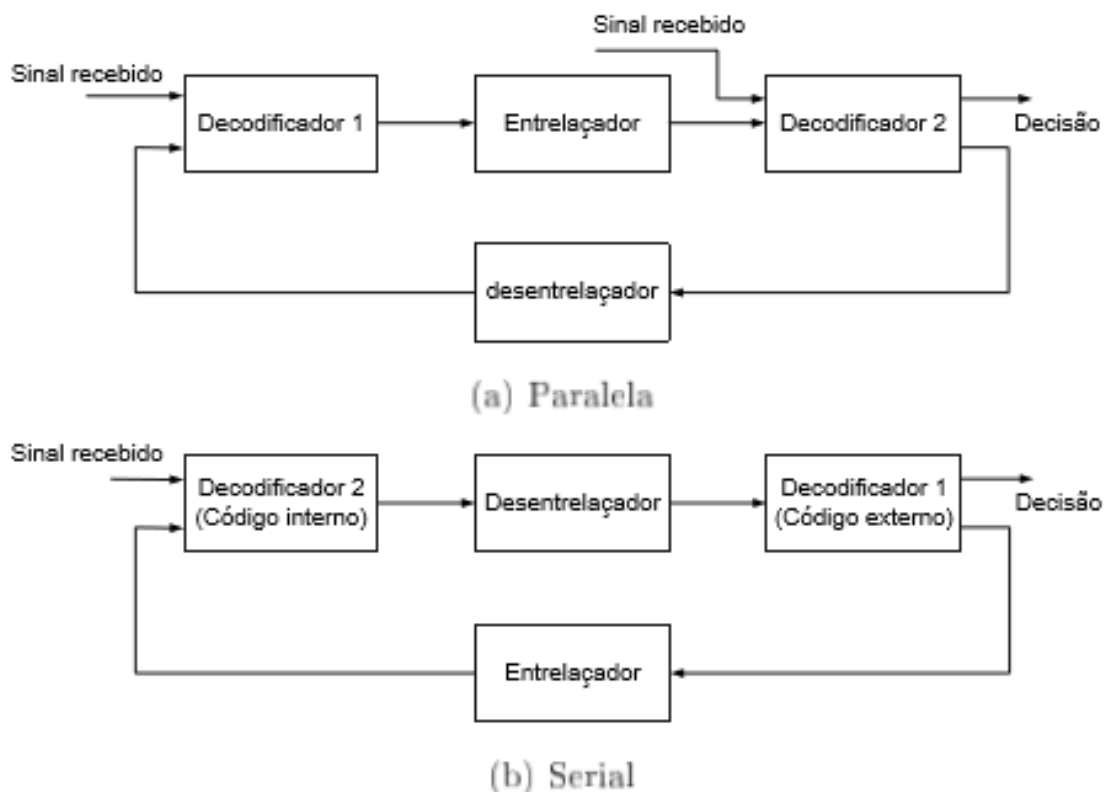


Figura 20. Estruturas básicas e iterativas da decodificação turbo. Fonte: [11].

Após terem sido analisadas todas as etapas mais comuns de tratamento da mensagem do transmissor até o receptor, é chegado, finalmente, o momento de

estudar sobre o processo que intitula este trabalho: a Equalização Turbo, tema da próxima seção.

7. Equalização turbo

Por meio das seções 5 e 6, pode-se estudar o funcionamento dos equalizadores e decodificadores. Em sistemas de comunicação digital convencionais, estes dois componentes trabalham de forma independente. Porém, com a aquisição do conhecimento sobre como os CODECs turbo funcionam, a grande ideia [11] foi notar que tanto o canal quanto o equalizador podem ser considerados como codificador e decodificador. A Figura 21 apresenta, portanto, o diagrama de todos os processos considerados neste trabalho e pelos quais a mensagem do transmissor passa até chegar ao receptor.

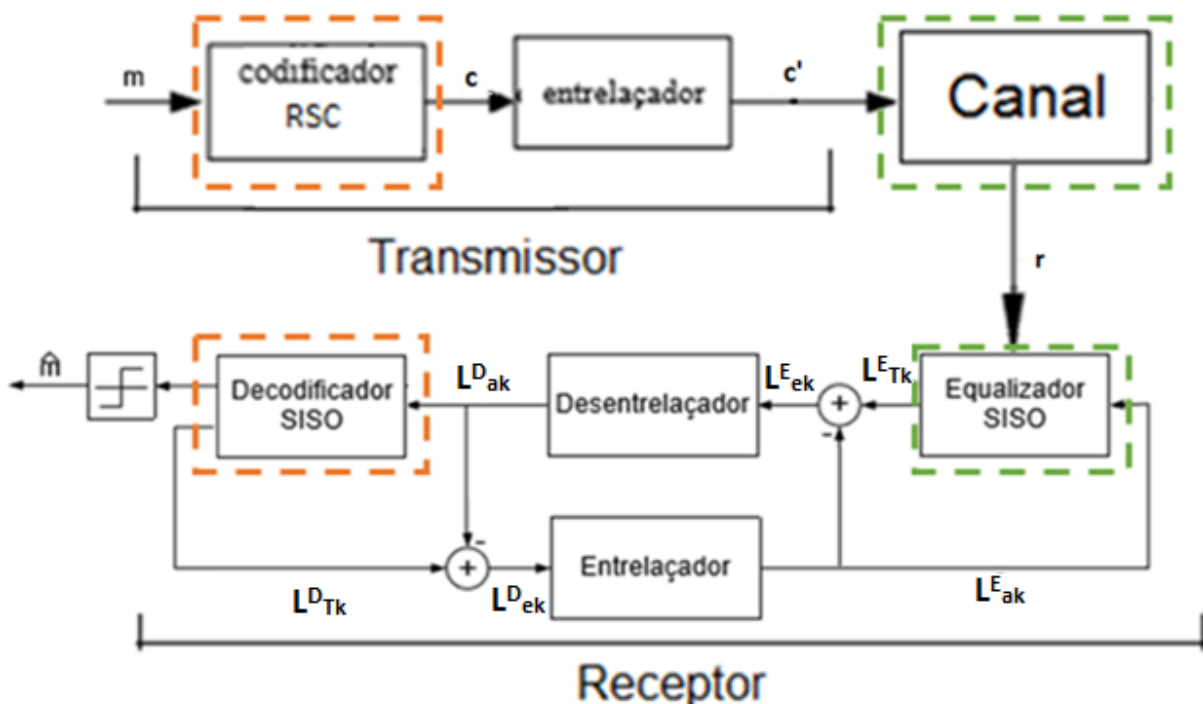


Figura 21. Diagrama do sistema completo de transmissão adotado para este trabalho. As caixas tracejadas indicam a relação entre CODECs.

A combinação iterativa entre equalizador e decodificador é quem recebe o nome de Equalização Turbo.

Outra consideração importante sobre a equalização turbo é referente aos tipos de dados que devem entrar e sair tanto do equalizador quanto do decodificador. Como já visto anteriormente, a melhor decisão feita por um decodificador baseia-se

na decodificação suave. Portanto, o ideal é trabalhar com algoritmos que considerem estimativas probabilísticas tanto na saída quanto na entrada. Esses tipos de algoritmos são conhecidos como SISO (do inglês, *Soft-Input, Soft-Output*) e o MAP-BCJR enquadra-se perfeitamente dentro deste grupo.

Aplicando o que foi explicado no parágrafo anterior na parte do receptor da Figura 21, primeiramente, todas as letras L representam informações suaves na forma de LLR, k um índice temporal, D indica que se trata de uma informação do/para o decodificador e E para informações do/para o equalizador [11]. O equalizador SISO recebe como entrada tanto os dados advindos do canal (r_k) quanto as informações *a priori* dos bits (L_{ak}^E) e tem como resultado uma LLR (L_{Tk}^E), que será explicada nas próximas subseções. Já o decodificador SISO recebe como entrada apenas a informação *a priori* dos bits já parcialmente “equalizados/decodificados” e desentrelaçados (L_{ak}^D), e tem como saída a LLR (L_{Tk}^D) resultante do logaritmo da equação (16).

De modo a garantir maior estabilidade e também melhor desempenho, durante a equalização turbo, é importante que tanto o equalizador quanto o decodificador trabalhem somente com uma informação “nova” sobre o bit m_k , chamada de informação *extrínseca* ($L_{ek}^{D\text{ ou }E}$) [11]. Portanto, durante as iterações, a informação *a priori* da nova análise, que seria intuitivamente a LLR (entrelaçada ou não) do processo anterior, é, na verdade, a diferença entre essa mesma LLR ($L_{Tk}^{D\text{ ou }E}$) e a informação *a priori* ($L_{ak}^{D\text{ ou }E}$) utilizada para gerar tal resultado. Deixando um pouco mais claro:

$$L_{ak}^E = \text{Entrelaçamento}(L_{Tk}^D - L_{ak}^D) \quad (18)$$

$$L_{ak}^D = \text{Desentrelaçamento}(L_{Tk}^E - L_{ak}^E) \quad (19)$$

Da Figura 21, o último componente que falta explicar é a tomada de decisão para os bits da mensagem decodificada (\hat{m}). De forma simples, há o limiar maior ou menor do que zero. Para os valores positivos de L_{Tk}^D , o bit estimado será 1 e para os negativos o bit será 0.

7.1. Decodificadores Atuando como Equalizadores

Após notar que o equalizador também pode ser considerado um decodificador, o algoritmo MAP e uma das suas variações (Max-Log-MAP) foram adaptados para trabalhar como equalizadores e também foi proposta a utilização do algoritmo SFE (do inglês, *Soft-Feedback Equalizer*). Todos os três são brevemente explicados nas próximas subseções.

7.1.1. Equalizador MAP

As principais diferenças entre o decodificador e o equalizador MAP estão no fato de que o segundo calcula as LLRs dos símbolos transmitidos, ou bits no caso do BPSK, na entrada do equalizador e o primeiro calcula as LLRs tanto na entrada quanto na saída do decodificador [11]; e a outra diferença está na forma como uma das probabilidades condicionais é calculada no meio do algoritmo. Em [11] é possível ler maiores detalhes sobre o desenvolvimento matemático, mas no presente trabalho será exibida apenas a equação alterada para o termo $p(r_k | s', s)$, pois a intenção é de apenas ressaltar que mesmo tendo sido feita a consideração de que o equalizador é como um decodificador, os processamentos de cada etapa continuam possuindo certas considerações diferentes.

$$p(r_k | s', s) = \exp\left(-\frac{1}{2\sigma_n^2} \cdot \left|r_k - \sum_{i=0}^{\mu} h_i(t) \cdot x_{k-i}\right|^2\right) \quad (20)$$

sendo $h_i(t)$, $i = 0 \dots \mu$, os coeficientes do canal, σ_n^2 é a variância do ruído AWGN e x_{k-i} são os símbolos associados à transição entre os estados s e s' , conforme descrito em [11].

7.1.2. Equalizador Max-Log-MAP

Lembrando que o codificador Log-MAP é o MAP “melhorado”, pois é considerado o logaritmo das equações e, portanto, são realizadas somas ao invés de multiplicações, o Max-Log-MAP (abreviado como MLM, neste trabalho) também apresenta uma condição que melhora o tempo de processamento do Log-MAP. Essa condição diz que basta ignorar o termo $\ln(1 + e^{-|\delta_1 - \delta_2|})$ do *logaritmo Jacobiano*,

equação (21), que foi utilizado para facilitar o cálculo das expressões mais complexas que envolviam o logaritmo natural da soma de exponenciais, conforme pode ser visto com maiores detalhes em [11].

Como o estudo do comportamento do $\ln(1 + e^{-|\delta_1 - \delta_2|})$ mostrou que esta função decai exponencialmente em função da diferença entre os argumentos das exponenciais, a aproximação feita em (21) permite uma diminuição considerável da complexidade computacional do algoritmo com uma perda de desempenho aceitável, sendo a base para o algoritmo MLM.

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_1 - \delta_2|}) \approx \max(\delta_1, \delta_2) \quad (21)$$

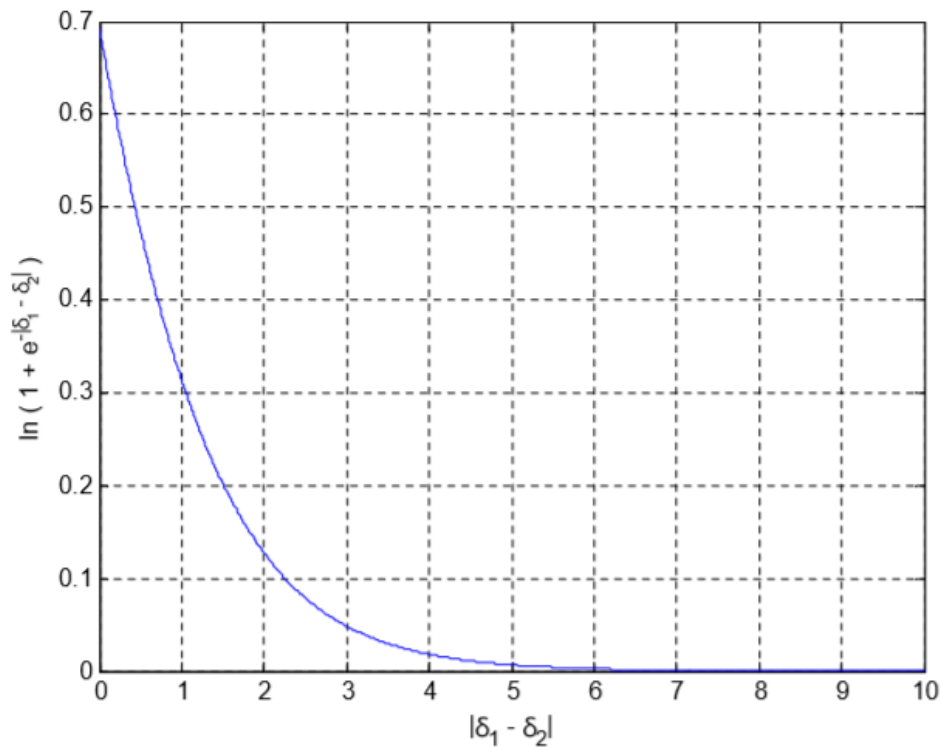


Figura 22. Comportamento de $\ln(1 + e^{-|\delta_1 - \delta_2|})$. Fonte: [11].

Sendo assim, as equações utilizadas para o MLM são [11]:

$$\ln \alpha_k(s) = \max_{s'} \{\ln \gamma_k(s', s) + \ln \alpha_{k-1}(s')\} \quad (22)$$

$$\ln \beta_k(s) = \max_s \{\ln \gamma_{k+1}(s', s) + \ln \beta_{k+1}(s)\} \quad (23)$$

$$L(x_k) = \max_{\{(s',s):s_k=+1\}} \{\ln \alpha_{k-1}(s') + \ln \gamma_k(s',s) + \ln \beta_k(s)\} - \max_{\{(s',s):s_k=-1\}} \{\ln \alpha_{k-1}(s') + \ln \gamma_k(s',s) + \ln \beta_k(s)\} \quad (24)$$

7.1.3. Equalizador SFE

Apesar da redução na complexidade computacional do algoritmo MLM em relação ao BCJR, seu tempo de processamento ainda cresce exponencialmente conforme o aumento no comprimento do canal. Portanto, uma solução é a utilização do algoritmo SFE (do inglês, *Soft-Feedback Equalizer*), que possui complexidade linear em relação ao comprimento do canal e desempenho melhor do que outros algoritmos de mesma complexidade, como apontado em [11]. O SFE possui a estrutura apresentada na Figura 23.

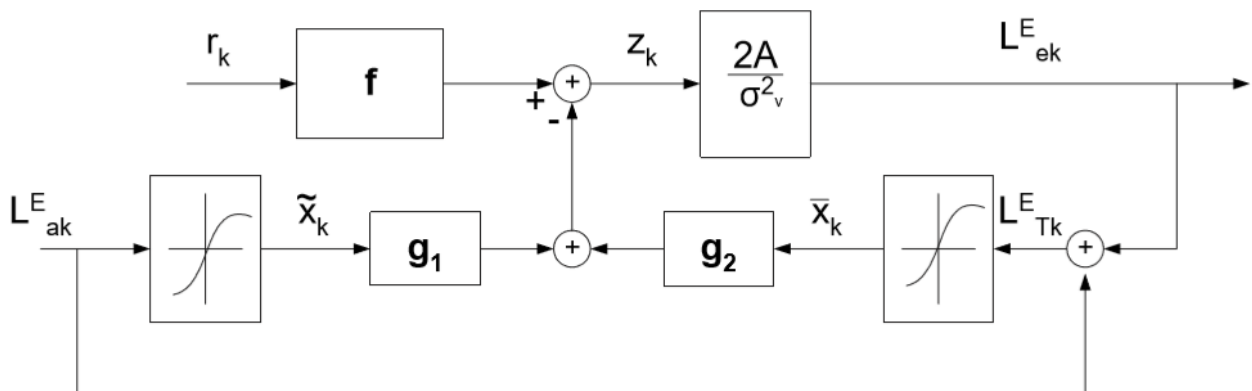


Figura 23. Estrutura do algoritmo SFE. Fonte: [11].

Explicando de forma resumida e sem detalhamento matemático, o funcionamento do SFE tem o princípio de tentar cancelar a interferência causada por uma quantidade de símbolos futuros e outros passados (\tilde{x}_k e \bar{x}_k) ao símbolo que é tratado no instante k (x_k).

Observando a estrutura da Figura 23, é possível notar que os principais acréscimos feitos à estrutura de realimentação para o equalizador SISO da Figura 21 são os filtros g_1 e g_2 , gerados por equações fechadas [11] e com o conhecimento dos coeficientes do canal. Conforme explicado em [11], o filtro g_1 é estritamente anticausal e é quem trata da informação *a priori* fornecida pelo decodificador e ao final tem como

resultado a informação sobre os símbolos futuros. Já o filtro g_2 é estritamente causal e é quem trata a LLR resultante da soma entre as informações *a priori* fornecidas pelo decodificador e as informações *extrínsecas* geradas pelo próprio equalizador, fornecendo, assim, a informação sobre símbolos passados.

A justificativa para o uso da informação de símbolos passados e futuros se deve pelo fato de que a saída do filtro linear f que trata o sinal recebido, r_k , ainda possui ISI [11], e a intenção é tentar cancelar estas interferências entre símbolos, gerando o símbolo z_k . Há uma relação entre os filtros f , g_1 e g_2 que permite minimizar o erro quadrático médio $E[|z_k - x_k|^2]$ entre a saída do equalizador e o símbolo tratado no instante k , e todo o seu detalhamento matemático está descrito em [11].

Para finalizar os processos realizados pelo SFE na equalização turbo, o símbolo z_k é transformado para a forma extrínseca pela equação (25) abaixo que, apesar de parecer simples, também possui todo um desenvolvimento matemático apresentado em [11].

$$L_{ek}^E = \frac{2A \cdot z_k}{\sigma_v^2}, \quad (25)$$

onde, A é o ganho equivalente do canal e σ_v^2 a variância do ruído.

8. Redes Neurais e Equalização Turbo

O ruído impulsivo, modelado pela distribuição alfa-estável, possui características particulares que diferem de uma distribuição gaussiana, conforme explicado na seção 3. Portanto, é provável que técnicas modeladas para situações que envolvam variáveis gaussianas não sejam tão eficazes quando considerado o ruído impulsivo. A ideia proposta para tentar amenizar o impacto do ruído impulsivo sobre o sinal de informação é de realizar uma equalização de canal antes do sinal ser processado pela equalização turbo.

Como foi evidenciada a complexidade do ruído impulsivo, propôs-se utilizar uma rede neural, que no caso seria a MLP, para tratar a situação de uma maneira mais “dinâmica” e com considerações não-lineares. A Figura 24 abaixo exhibe a estrutura do receptor proposto com a adição da MLP.

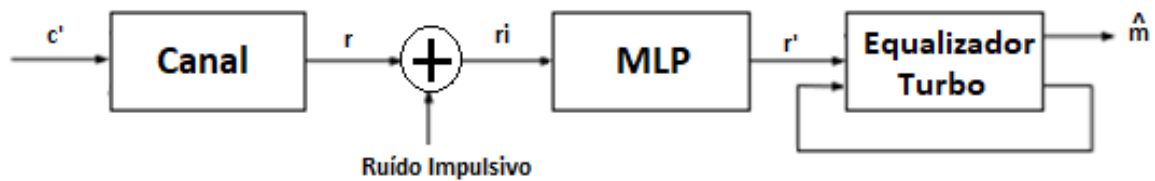


Figura 24. Receptor proposto com a combinação de MLP e equalização turbo para ambientes com ruído impulsivo.

9. Simulações, Resultados e Discussões

As simulações de operação da equalização turbo do presente trabalho foram baseadas nos estudos iniciados em [11], portanto dentre estes estudos foram escolhidos os algoritmos MAP, Max-log MAP (MLM) e SFE como equalizadores SISO em um canal PLC. A justificativa deve-se pelo fato de o primeiro apresentar menores taxas de erro, apesar do maior tempo de processamento; o segundo por apresentar resultados intermediários, mas com rápido processamento; e o terceiro por realizar um tratamento diferente dos dados e também possuir um ótimo tempo de processamento computacional.

Já em relação ao decodificador, a fim de restringir a quantidade de dados a serem analisados, optou-se por utilizar apenas um tipo de algoritmo no decodificador. Lembrando o que foi explicado logo no início da seção 7, ao notar que o equalizador poderia ser um “decodificador” do canal, foi, então, proposta a utilização dos algoritmos de decodificação como equalizadores, tendo sido feitos os pequenos ajustes necessários. Com isso, pôde-se considerar que o desempenho dos algoritmos como equalizadores tenderia a ser o mesmo como decodificadores, uma vez que os processos de tratamento da informação seriam os mesmos. Portanto, a escolha de somente um tipo de decodificador não causaria uma perda significativa de informação sobre o desempenho de vários tipos de algoritmos dentro do processo de equalização turbo. Por fim, a escolha do decodificador foi baseada na intenção de possuir melhores resultados e também pouco tempo de processamento. O algoritmo que melhor se encaixava nestas necessidades era o Log-MAP.

Outras configurações defaults estudadas em [11] e que também foram adotadas no trabalho atual foram:

- Tamanho do frame a ser transmitido: 1024 bits;
- Polinômios geradores do RSC [11]: $q_1 = [1 \ 1 \ 1]$ e $q_2 = [1 \ 0 \ 1]$;
- Taxa de codificação $R = 1/2$;
- Quantidade máxima de bits transmitidos: 1×10^6 ;
- Quantidade de iterações turbo: 15;
- Quantidade de frames errados: 15.

Esta última configuração sobre a quantidade de frames errados foi criada para servir como um segundo critério de parada no envio de bits. A ideia deste critério é assumir que se 15 frames seguidos não foram corretamente decodificados, então a equalização turbo não é mais eficaz na recuperação da mensagem.

Tendo sido feitos os ajustes iniciais, o primeiro desafio encontrado durante a simulação foi determinar uma quantidade mínima suficiente de coeficientes do canal PLC que pudessem representá-lo da mesma forma que é exibido na Figura 4, dado que o tempo de processamento dos algoritmos MAP, Log-MAP e MLM crescia de forma exponencial com o aumento da quantidade de coeficientes do canal. A Figura 25 exhibe as respostas em frequência do canal PLC quando consideradas algumas quantidades de coeficientes. Por meio dela, fica evidente que com 5 a 10 coeficientes já seria possível aproximar a resposta original do canal. Portanto, de modo a otimizar o processamento computacional, escolheu-se utilizar 5 coeficientes do canal original, que possuía 1951 coeficientes.

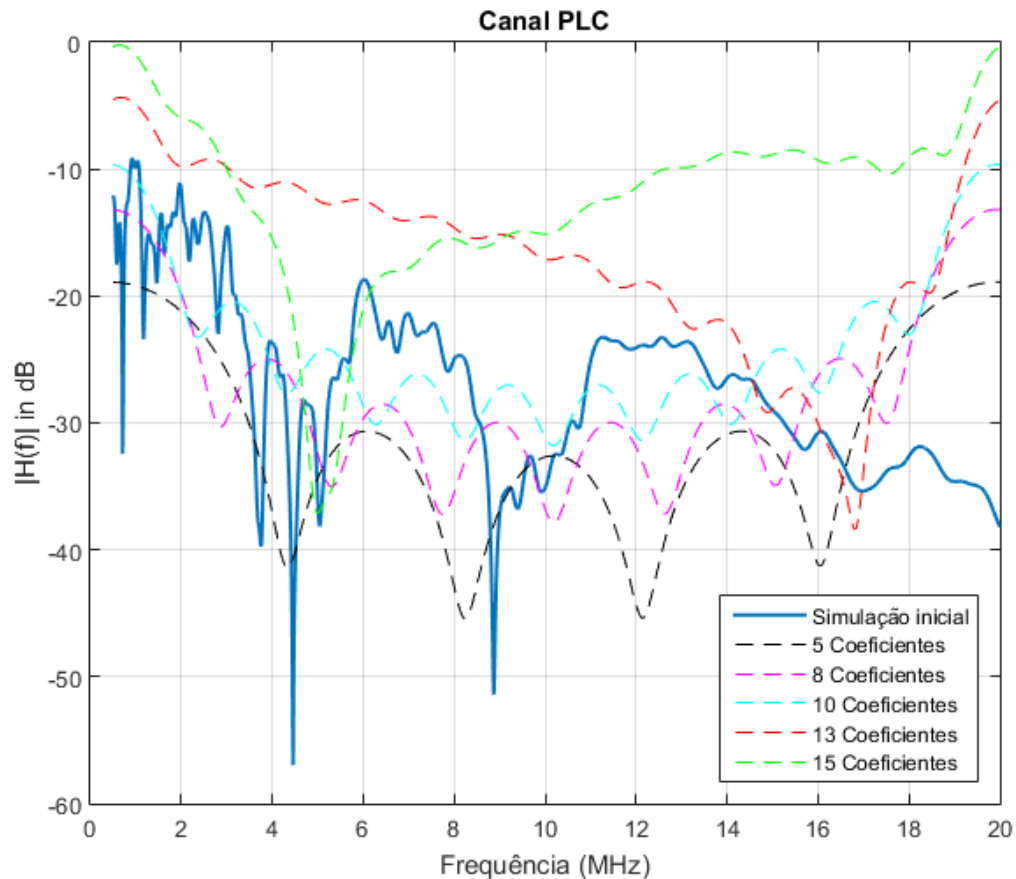


Figura 25. Escolha da quantidade de coeficientes do canal PLC.

Outra consideração acerca do canal PLC aproximado referiu-se a sua energia. O trabalho apresentado em [11] considerava um canal com energia unitária, ou seja, um canal normalizado, pois desse modo era possível simplificar o cálculo da energia do ruído, fosse utilizando a equação (7) para o caso de ruído impulsivo (γ) ou a equação abaixo para o caso de ruído gaussiano (σ) [11]:

$$\sigma = \frac{1}{\sqrt{2 \cdot R \cdot \left(\frac{E_b}{N_0}\right)}} \quad (26)$$

Onde $\frac{E_b}{N_0}$, que é a SNR, deve estar em valor absoluto e não em dB e R (taxa de codificação) que foi definida em (8).

Antes de apresentar a resposta do canal PLC normalizado, outro ajuste necessário foi considerar apenas coeficientes reais para o canal, pois, pelo estudo apresentado em [11], o SFE não considerava valores imaginários em seus cálculos. Portanto, para que todos os equalizadores trabalhassem com o mesmo tipo de canal,

foi considerado o módulo do valor de cada um dos 5 coeficientes escolhidos para caracterizar o canal PLC.

A Figura 26 apresenta a resposta original simulada na seção 2.1 (linha azul contínua), essa mesma resposta normalizada e em módulo (linha azul tracejada) e, finalmente, a resposta normalizada e em módulo dos 5 coeficientes. Analisando este gráfico, nota-se que a energia da resposta em frequência inicialmente simulada para o canal PLC é baixa, ficando, aproximadamente, 10 dB abaixo das respostas normalizadas. Com isso, compreendeu-se que qualquer sinal que trafegue por um canal PLC está sujeito a sofrer consideráveis atenuações na sua potência. Em relação às respostas normalizadas e em módulo, o modelo com apenas 5 coeficientes está muito mais semelhante à resposta da simulação inicial, portanto concluiu-se que o canal estava bem representado.

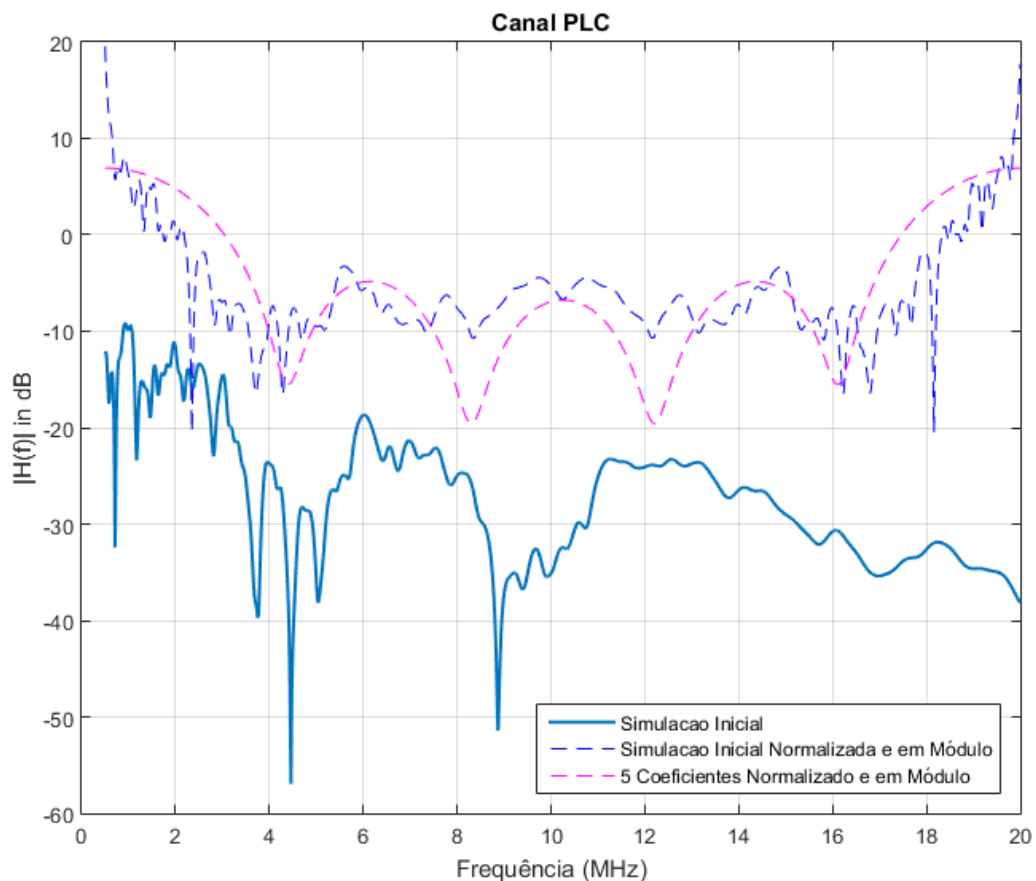


Figura 26. Espectro do canal PLC que foi ajustado para as limitações enfrentadas na simulação.

Seguindo para a etapa de simulações, o parâmetro utilizado para mensurar a qualidade dos resultados da equalização turbo era a BER, que no caso da

modulação BPSK também pode ser analisada como SER (do inglês, *Symbol Error Ratio*). Como a BER representa uma taxa de erro, quanto menor o seu valor, melhor é desempenho do processo.

A Figura 27 apresenta o desempenho dos três tipos de equalizador turbo considerados neste trabalho, tanto com adição de ruído gaussiano quanto com ruído impulsivo. Antes de continuar a análise de desempenho, vale comentar que a quantidade de coeficientes dos filtros g_1 (anticausal) e g_2 (causal) do equalizador SFE foram determinados empiricamente como: $g_1 = 5$ e $g_2 = 4$, dado que eram os valores mais baixos que proporcionavam um desempenho próximo aos demais para o SFE.

Voltando para a análise de desempenho. Para os cenários de canal PLC com adição de ruído gaussiano, os três tipos de equalizador turbo apresentaram um bom desempenho, alcançando taxas de até 10^{-5} , ou seja, 1 bit errado a cada 100.000 bits transmitidos. Portanto, conclui-se que mesmo em um canal cheio de reflexões e imperfeições, é possível recuperar a mensagem original.

O algoritmo MAP teve melhor desempenho dentre os três, pois era quem conseguia alcançar menores valores de BER para uma mesma SNR. Apesar desse ponto positivo, seu tempo de processamento foi maior do que os demais. Como comentário adicional para este algoritmo, já era esperado que seu desempenho fosse melhor pelo menos em relação ao MLM, pois seus processos não sofreram arredondamentos ou outras considerações como no MLM. Em relação ao MLM, seu desempenho mostrou-se intermediário entre o MAP e o SFE, assim como o seu tempo de processamento. Portanto, este seria o algoritmo com melhor “custo-benefício”. O SFE tinha um tempo de execução bem rápido, porém seu desempenho foi o pior entre os três. Também notou-se que para altos valores de SNR, que seria a partir de 5,5 dB, os valores de BER alcançados para os três algoritmos não eram muito distantes. Este fato que pode ser explicado ao notar que uma alta SNR implica em maior potência do sinal que do ruído, facilitando a recuperação dos símbolos enviados.

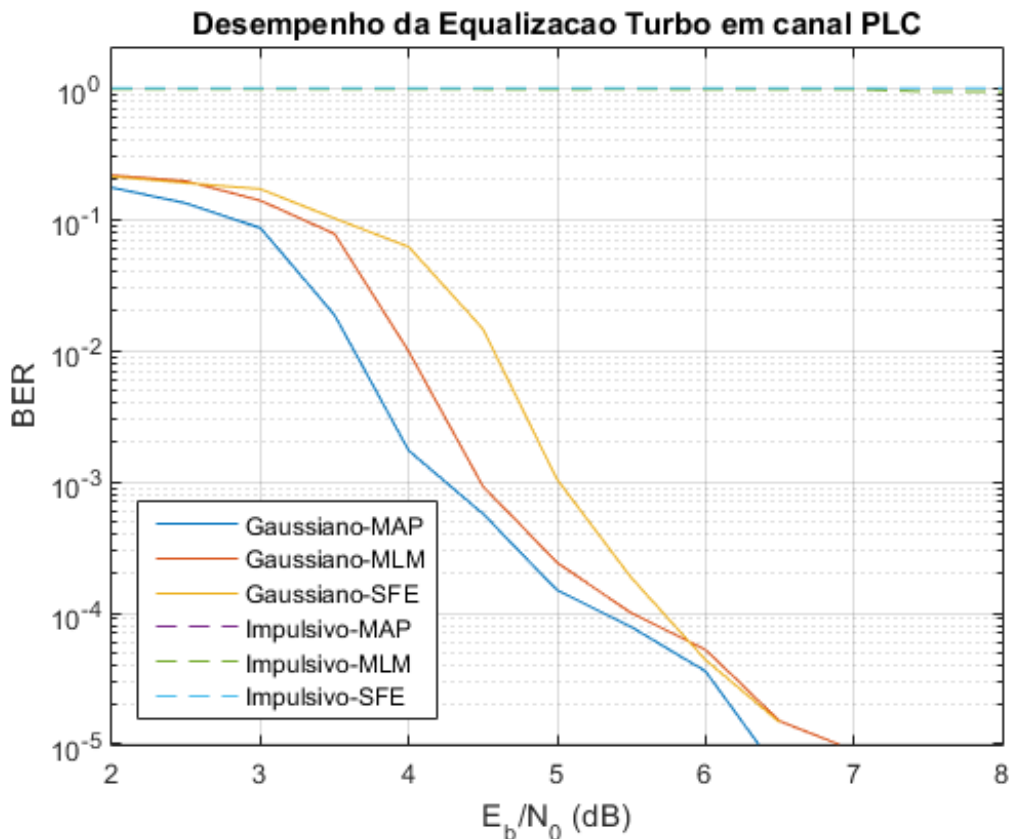


Figura 27. Desempenho da equalização turbo em canal PLC e com adição de ruído.

Ao trocar o tipo de ruído para impulsivo, a equalização turbo não foi capaz de recuperar a mensagem original, dentro do mesmo intervalo de SNR, independentemente o tipo de equalizador utilizado. No gráfico da Figura 27, o desempenho de cada um destes equalizadores resultou em uma linha reta e todas as três restas estão dispostas no valor de $BER = 10^0 = 1$, ou seja, 100% de erro. Portanto, conclui-se que os algoritmos utilizados não estão preparados para trabalhar com o ruído impulsivo.

Colocando em prática as etapas da Figura 24 com uma rede MLP composta por 4 neurônios, com função de ativação sigmoideal, na camada escondida (*hidden layer*), a fim de tentar amenizar os efeitos do ruído impulsivo sobre o sinal para que o equalizador turbo fosse capaz de recuperar a mensagem original, foi obtido o gráfico de desempenho apresentado na Figura 28 abaixo. Ao contrário de tudo o que era esperado, não houve melhora no desempenho da equalização turbo, mas somente um deslocamento para o intervalo entre $3,5 \times 10^{-1}$ e $5,0 \times 10^{-1}$ de BER. Com relação a estes valores de BER, foi concluído que são referentes à correção feita pela MLP no sinal transmitido, pois na simulação também foi calculada uma BER entre os bits antes

de passarem pelo canal com ruído impulsivo e logo após o tratamento pela MLP e as taxas encontradas estavam dentro do mesmo intervalo. Logo, os picos presentes na Figura 28 representam os momentos em que a MLP teve pior desempenho, conseguindo recuperar somente 50% dos bits transmitidos.

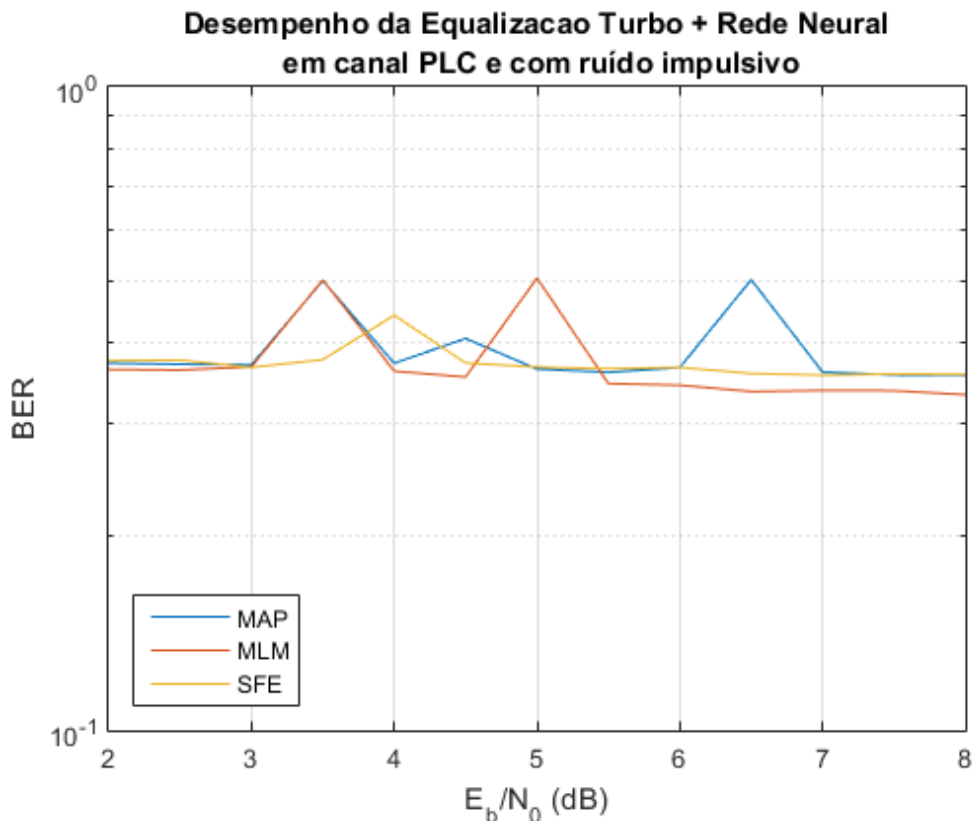


Figura 28. Desempenho da equalização turbo em canal PLC e ruído impulsivo com adição de rede neural na equalização.

A primeira justificativa proposta para esclarecer o comportamento indesejado foi o fato de que a MLP tinha por intenção tentar amenizar os efeitos do ruído impulsivo, mas como os algoritmos foram condicionados a trabalhar em um cenário gaussiano, mesmo que a MLP retirasse 100% do ruído impulsivo.

Outro estudo levantou a hipótese de os algoritmos estarem tentando equalizar um canal que não seja somente o canal PLC simulado no início deste trabalho, mas sim um canal resultante da combinação entre o canal original e a rede MLP e que, então, possuiria menos interferências intersimbólicas. Por se tratar de uma análise mais aprofundada para tentar determinar a resposta do “novo canal” criado pela MLP e o canal PLC e por conta do tempo restante para terminar o presente trabalho, não foi possível testar tal hipótese, mas caso haja oportunidade em continuar

os estudos com a combinação entre canal PLC, rede MLP e equalização turbo, este seria o ponto de partida.

10. Conclusões

O canal PLC é um meio não muito usual para transmissão de informação, mas pelo fato de já existir uma rede amplamente projetada e implantada, não se pode descartar o uso de uma ferramenta que tem o potencial de aumentar a conectividade, nesse mundo tão globalizado. Seu estudo teórico mostrou que se trata de um canal complexo, mas que pode ser traduzido em uma equação matemática e, com a dependência de valores empíricos, simulado computacionalmente.

A proposta de usar equalização turbo em canal PLC mostrou-se ser muito adequada quando considerada somente a presença adicional do ruído gaussiano (ruído normalmente considerado em sistemas de comunicação), pois, neste cenário, todos os três algoritmos propostos como equalizadores SISO (MAP, MLM e SFE), juntamente com o decodificador Log-MAP, foram capazes de recuperar a mensagem original, alcançando taxas de erro de até 1×10^{-5} , significando que há apenas 1 bit errado a cada 100.000 transmitidos. Com isso, conclui-se que mesmo o canal PLC tendo uma resposta em frequência que causa interferência intersimbólica e outras degradações no sinal transmitido, é possível recuperar a mensagem original.

O problema ao trabalhar com o canal PLC é lembrar que neste tipo de canal o ruído mais relevante não é o gaussiano, mas sim o ruído impulsivo. Pelo fato deste ruído ser composto por picos de grande amplitude e de aparição esporádica, a degradação causada no sinal de informação compromete imensamente a recuperação. Os mesmos algoritmos de equalização SISO não foram capazes de recuperar a mensagem original nesse novo cenário, as taxas de erro ficaram sempre em 100%.

Mesmo que haja a justificativa de que a recuperação da mensagem original não ter sido possível pelo fato de os algoritmos terem sido modelados considerando o ruído AWGN, fica evidente que o ruído impulsivo difere do AWGN e é um caso que precisa de maior atenção e cuidado no seu tratamento. A solução proposta neste

trabalho foi adicionar a rede neural, do tipo MLP, antes da equalização turbo, mas não houve melhora no desempenho da equalização turbo.

Apesar de não ter havido melhora no desempenho da equalização turbo, ao final dos estudos foi proposta a hipótese de ser preciso considerar nos algoritmos um canal que seja resultante da combinação entre o canal original PLC e a rede MLP. Essa hipótese e outras soluções para os problemas enfrentados com o ruído impulsivo podem ser abordadas em trabalhos futuros.

Referências Bibliográficas

- [01] Ferreira, H. C., et al., Power Line Communications: Theory and Applications for Narrowband and Broadband Communications over Power Lines. Torquay, UK: Wiley, 2010.
- [02] Dostert, K., Telecommunications over the Power Distribution Grid – Possibilities and Limitations, in International Symposium on Power Line Communications and Its Applications (ISPLC), Essen, Germany, abr. 1997, p. 1-9.
- [03] Parson, J. D., Mobile Radio Propagation Channel, 2nd ed. John Wiley & Sons, Inc., 2000.
- [04] Zimmermann, M., Dostert, K., “A Multipath Model for the Powerline Channel”, IEEE Transactions on Communications, Vol. 50, no. 4, abr. 2002.
- [05] Notas de aula – Codificação de canal. Disponível em: <http://tele1.dee.fct.unl.pt/csf_2005_2006/folhas/Codif_mest.pdf>. Acesso em 28 abr. 2017.
- [06] Haykin, S., Unsupervised Adaptive Filtering, Vol. II: Blind Deconvolution, Wiley, 2000.
- [07] UFABC. Notas de aula Prof. Ricardo Suyama para a disciplina Sistemas Inteligentes. Aulas 2 -4, 8.
- [08] Ozono, P. Y., “Projeto Final: Equalização de Canal por Sistemas Inteligentes (MLP e SVM)”, UFABC, Santo André, Brasil, dez. 2016.
- [09] Santos, A. F., Guimarães, D. A., “Introdução à Equalização Turbo”, Inatel, Departamento de Telecomunicações, Revista Científica Periódica – Telecomunicações, v. 8, n. 1, set. 2005.
- [10] Proakis, J. G., Digital Communications, 4th edition, McGraw-Hill, 2000.
- [11] Loiola, M. B., “Técnicas Autodidatas e Soluções de Baixa Complexidade para Equalização e Estimação Turbo”, Mestrado em Engenharia Elétrica UNICAMP, fev. 2005.
- [12] Haykin, Simon. Sistemas de Comunicação Analógicos e Digitais, Editora Bookman, 4^a edição, 2004.
- [13] Ribeiro, J. A. J.; Engenharia de Micro-ondas–Fundamentos e Aplicações, Érica, 2008.
- [14] Tan, B., Thompson, J., “Parameter Appendix”, mar., 2012.
- [15] Middleton, D., An introduction to statistical communication theory, McGraw-Hill, New York, 1960, vol. 960.
- [16] Zimmermann, M., Dostert, K., “Analysis and Modeling of Impulsive Noise in Broad-Band Powerline Communications”, IEEE Transactions on Electromagnetic Compatibility, vol. 44, no. 1, fev. 2002.

- [17] Khalil, K., et al, "Analysis of the Impact of Impulsive Noise Parameters on BER Performance of OFDM Power-Line Communications", Department OAE, UVHC, Valenciennes, France, 2010.
- [18] Wang, X., Chen, R., "Blind Turbo Equalization in Gaussian and Impulsive Noise", IEEE Transactions on Vehicular Technology, vol. 50, no. 4, pp. 1092-1105, jul. 2001.
- [19] Tsihrintzis, G. A., Nikias, C. L., "Fast estimation of the parameters of alpha-stable impulsive interference", IEEE Transactions on Signal Processing, vol. 44, no. 6, pp. 1492-1503, jun. 1996.
- [20] Lopes, M. R., "Relatório do Trabalho de Graduação - Estudo e análise de desempenho de códigos corretores de erros avançados em sistemas sujeitos a ruídos impulsivos", UFABC, Santo André, Brasil, ago. 2015.
- [21] Nolan, John P. Stable Distributions – Models for Heavy Tailed Data. American University, 2014.
- [22] Lévy, P., Calcul des probabilités, volume unique, Gauthier-Villars et Cie, Paris, 1925.
- [23] L. R. Bahl, J. Cocke, F. Jelinek, e J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", IEEE Transactions on Information Theory, vol. 20, no. 2, pp. 284-287, mai. 1974.
- [24] Russel, S., Norvig, P., Artificial Intelligence: A Modern Approach, 3rd edition, Pearson, New Jersey, 2010.

Apêndice I – Simulação canal PLC

```

%Simulação do canal proposto por Zimmermann e Dostert no artigo "A
%Multipath Model for the Powerline Channel"
%%
clear all

%Variáveis fixas NAYY150
Er = 4; %permissividade relativa
c = 3e8; %velocidade da luz m/s
k = 1; %expoente do fator de atenuação (tipicamente entre 0.5 e 1)
a0 = 0; %parâmetro de atenuação (m^-1)
a1 = 7.8e-10; %parâmetro de atenuação (s/m)
vp = c/sqrt(Er);

%Variáveis experimentais
g = [.029,.043,.103,-.058,-.045,-.04,.038,-.038 .071,-.035,.065,-.055,.042,-.059,.049];
d = [90,102,113,143,148,200,260,322,411,490,567,740,960,1130,1250];

%Intervalo de frequências
f = (.5:.01:20)*1e6;

for i = 1:length(f)
    H(i) = 20.*log10(abs(sum(g.*exp(-(a0+a1.*f(i)^k).*d).*exp(-j*2*pi.*f(i)*d/vp))));
end

plot(f/1e6,H); grid
ylabel '|H(f)| in dB'
xlabel 'Frequência (MHz)'

```

Apêndice II – Simulação da Rede Neural

No trabalho realizado em [08], ao transportar a teoria do MLP para o Matlab, há funções que já criam, treinam e simulam esse tipo de rede. A criação é feita através do comando *newff(x,a,n)*, no qual *x* contém os dados de entrada que serão utilizados para treinamento, *a* a saída desejada e *n* a quantidade de neurônios da camada única camada intermediária. O treinamento utiliza a função *train(net,x,n)*, sendo *net* a variável que guarda o resultado da construção feita por *newff()*. E finalmente a simulação da rede com demais dados de entrada é feita pelo comando *sim(net,x')*, *x'* é uma variável de contém os demais dados de entrada que foram separados apenas para teste da rede já “calibrada”. A seguir um exemplo de implementação:

```
%===== Trainamento MLP =====
for i = 1:5
    x = round(rand(1, L_total-m)); % info. bits
    en_output1 = encoderm( x, g ); % saída do codificador RSC (+1/-1)
    en_output = en_output1(alpha); %entrelaçamento
    saida_canal=filter(canal,[1],en_output);
    r = (saida_canal + sigma*ruido(cont_ruido+1:cont_ruido+length(saida_canal)))/1e4;
    r1(alpha) = r;
    cont_ruido = cont_ruido+length(saida_canal);
    if cont_ruido >= length(ruido)
        cont_ruido = 0;
    end
    if(i==1) net = newff(r1,en_output1,4);
    elseif(i==5)
        MLP = sim(net,r1);
    else
        net = train(net,r1,en_output1);
    end
end
end
```

Apêndice III – Simulação do Ruído Impulsivo (Distribuição alfa-estável)

```
function imp_noise = impulsive_noise(tamanho,alpha)
%% Simulação do ruído impulsivo por meio do modelo alpha-estável
%%Variáveis aleatórias uniformemente distribuídas
u1 = rand(1,tamanho);
u2 = rand(1,tamanho);

%Parâmetros auxiliares
theta = pi*(u1-0.5);
W = -log10(u2);

%Distribuição alpha-estável
imp_noise = (sin(alpha*theta)./cos(theta).^(1/alpha)).*(cos((alpha-1)*theta)./W).^((1-
alpha)/alpha);

figura = figure; plot(imp_noise)
title('Distribuição alpha-estável')
xlabel('Amostras'); ylabel('Intensidade')
xlim([0 20e6])
```