



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

EMANUELLE PEREIRA MOURA

**ESTUDO E APLICAÇÃO DE CÓDIGOS CONVOLUCIONAIS COM
DECODIFICADOR DE VITERBI**

Campina Grande, Paraíba
Julho de 2015

EMANUELLE PEREIRA MOURA

ESTUDO E APLICAÇÃO DE CÓDIGOS CONVOLUCIONAIS COM
DECODIFICADOR DE VITERBI

*Relatório de Estágio Integrado submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Área de Concentração: Teoria da Informação e Codificação

Orientador:

Professor Bruno Barbosa Albert, D. Sc.

Campina Grande, Paraíba
Julho de 2015

EMANUELLE PEREIRA MOURA

ESTUDO E APLICAÇÃO DE CÓDIGOS CONVOLUCIONAIS COM DECODIFICADOR DE VITERBI

*Relatório de Estágio Integrado submetido à Unidade
Acadêmica de Engenharia Elétrica da Universidade
Federal de Campina Grande como parte dos requisitos
necessários para a obtenção do grau de Bacharel em
Ciências no Domínio da Engenharia Elétrica.*

Área de Concentração: Teoria da Informação e Codificação

Aprovado em / /

Professor Francisco Marcos de Assis, D. Sc.
Universidade Federal de Campina Grande
Avaliador

Professor Bruno Barbosa Albert, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

AGRADECIMENTOS

Gostaria de exprimir todo o meu reconhecimento ao engenheiro Karim Maouche, que foi o responsável por este estágio. Sou grata por seus ensinamentos, treinamento e conselhos.

Agradeço a todos que compõem a empresa ADEUNIS RF pela receptividade durante os cinco meses de estágio. Principalmente, aos engenheiros que fazem parte da equipe técnica, pelo tempo que dedicaram ao meu treinamento, questionamentos e ajuda nas mais diversas situações inerentes ao trabalho.

Agradeço às instituições de ensino UFCG e ESISAR, responsáveis pela minha formação acadêmica. Sou profundamente grata aos professores Etienne Perret e Bruno Barbosa Albert que me aconselharam sabiamente na realização deste relatório.

Enfim, agradeço a minha família e amigos pela paciência e pelo apoio em todos os momentos.

RESUMO

Em comunicações digitais, o conteúdo da informação enviada pode chegar à recepção modificado ou mesmo perdido durante o seu transporte pelo canal físico. Os códigos de correção foram introduzidos com a finalidade de reduzir o erro na transmissão devido à perturbação do canal.

Este relatório apresenta as atividades de pesquisa desenvolvidas na empresa ADEUNIS RF. Consiste de um estudo detalhado dos códigos corretores de erros, principalmente a classe dos codificadores convolucionais com decodificador de Viterbi. Além disso, apresentamos os resultados das simulações realizadas no MATLAB e os resultados experimentais obtidos a partir de dados emitidos por um rádio modem.

Palavras-chave: Códigos corretores de erros, código convolucional, decodificador de Viterbi, MATLAB.

ABSTRACT

In the digital communication, the information content may arrive on reception modified or lost during its transport through the physical channel. The corrections codes were introduced in order to reduce the transmission errors caused by channel impairments.

This report proposes to present the ADEUNIS RF research activities. It consists of a detailed study of error correction codes, mainly the class of convolutional codes with Viterbi decoder. Furthermore we present the MATLAB simulation results and the experiments results obtained from the data sent from a Radio Modem.

Keywords: Error Correction Codes, Convolutional Code, Viterbi Decoder, MATLAB programming

LISTA DE ILUSTRAÇÕES

Figura 1. Diagrama de blocos de um sistema de comunicação	1
Figura 2. Pontos de venda dos produtos ADEUNIS RF pelo mundo e a sede localizada em Crolles.....	3
Figura 3. Produtos ADEUNIS RF	4
Figura 4. Processo de codificação	5
Figura 5. Codificador com taxa do código $R = 1/2$	7
Figura 6. Codificador com taxa do código $R = 1/3$	8
Figura 7. Codificador $R = 1/2$	9
Figura 8. Codificador $R = 1/2$	9
Figura 9. Diagrama de estados de um código $C(2,1,3)$	10
Figura 10. Representação por treliça	11
Figura 11. Representação em treliça.....	12
Figura 12. Representação em treliça.....	12
Figura 13. Representação em treliça.....	13
Figura 14. Representação em treliça.....	13
Figura 15. Organização do decodificador de Viterbi	18
Figura 16. Desempenho de códigos $C(2,1,5)$ utilizando diferentes decisões.....	19
Figura 17. Desempenho de códigos convolucionais utilizando diferentes rendimentos	20
Figura 18. Desempenho de códigos convolucionais utilizando diferentes comprimentos de restrição.....	21
Figura 19. Gerador de sinais.....	22
Figura 20. Câmara anecóica da empresa	23
Figura 21. Antena Bilog-periódica	23
Figura 22. Esquema da comunicação serial entre o Modem e o computador	24
Figura 23. ARF868 LP e ARF868 ULR Modems.....	24
Figura 24. Configuração da transmissão feita no laboratório. 1- Emissor: gerador de sinais 2- Receptor: modem ARF. 3 - Computador com o software para a aquisição dos dados.....	25
Figura 25. Performances em BER de códigos convolucionais.....	27
Figura 26. Quantidade de erros em cada pacote.....	27

Figura 27. Curvas de BER com o entrelaçamento de bits.....	28
Figura 28. Experiência com condução via cabo no laboratório ADEUNIS. Tx é o GS, Rx é o modem ARF.....	29
Figura 29. Performance em BER do código (2,1,3).....	30
Figura 30. Medições dentro da câmara anecóica.....	30
Figura 31. Performance em BER do código C(2,1,3) para transmissões na câmara anecoica	31
Figura 32. Medição de campo. A esquerda o modem emissor localizado na empresa. A direita, o modem receptor localizado a de 20 Km de distância.....	32

LISTA DE TABELAS

Tabela 1. Setores de aplicação dos produtos ADEUNIS.....	4
Tabela 2. Principais aplicações dos códigos convolucionais	7
Tabela 3. Configurações pertinentes ao experimento.....	26

LISTA DE ABREVIATURAS E SIGLAS

AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem
BER	Bit Error Rate
BM	Branch Metric
BPSK	Binary Phase Shift Keying
CCSDS	Consultative Committee for Space Data Systems
CDMA	IS-95A code division multiple access Interim Standard - 95A
FEC	Forward Error Correction
GS	Gerador de Sinais
ISM	Industrial, Scientific and Medical
LTE	Long Term Evolution
NASA	National Aeronautics and Space Administration
NFC	Near Field Communication
PM	Path Metric
RF	Radiofrequência
UWB	Ultra-wide-band
VA	Viterbi Algorithm

SUMÁRIO

1	Introdução	1
1.1	Objetivos do estágio.....	1
1.2	Empresa	2
1.2.1	Produtos ADEUNIS RF	3
2	Fundamentação Teórica	5
2.1	Os códigos corretores de erros.....	5
2.2	Os códigos convolucionais	6
2.3	Algoritmo de Viterbi.....	11
3	Simulação MATLAB.....	15
3.1	Implementação do decodificador de Viterbi.....	16
3.2	Resultados.....	19
4	Estudo Prático	22
4.1	Material utilizado	22
4.1.1	Gerador de sinais.....	22
4.1.2	Câmara anecóica	23
4.1.3	Antena Bilog-periódica	23
4.1.4	DOCKLIGHT.....	24
4.1.5	Rádio modem ARF868.....	24
4.2	Procedimentos experimentais e resultados	25
4.2.1	Experiência 1 - Emissão irradiada em laboratório	25
4.2.2	Experiência 2- Emissão conduzida via cabo	28
4.2.3	Experiência 3- Emissão em Câmara anecóica.....	30
4.2.4	Experiência 4: Medições de campo.....	31
4	Conclusão.....	33
	Bibliografia.....	34
	APÊNDICE A – Código no MATLAB.....	35

1 INTRODUÇÃO

1.1 OBJETIVOS DO ESTÁGIO

O princípio básico da telecomunicação é transportar uma mensagem de uma fonte emissora a um destinatário através de um canal físico. Assim sendo, um sistema de comunicação é um conjunto de mecanismos que permitem o tratamento e o transporte da informação de um ponto emissor a um ponto receptor, de maneira que o mínimo de informação seja perdida durante a transmissão.

Um esquema de um sistema de comunicação simplificado é ilustrado no diagrama de blocos da Figura 1. Com base na imagem a seguir podemos discutir brevemente a funcionalidade de cada componente do diagrama.

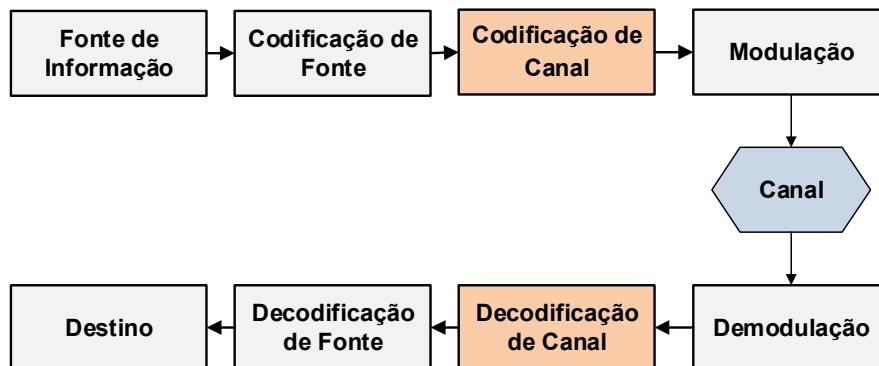


Figura 1. Diagrama de blocos de um sistema de comunicação

Primeiramente, temos a fonte de informação, que fornece a mensagem a ser transmitida. Logo em seguida, a codificação de fonte é feita para eliminar a redundância presente na mensagem que será enviada, sua utilização é necessária para melhorar a eficiência do sistema. Essa mensagem passa então para a etapa de codificação de canal onde é adicionada uma redundância controlada a informação, permitindo a identificação e a correção de eventuais erros ocorridos durante a transmissão.

Na etapa de modulação, as características da portadora (como a frequência ou a amplitude) são modificadas para que a informação seja transmitida. Em seguida, o sinal modulado é enviado pelo canal de comunicação, que é o meio físico (fibra ótica, cabo coaxial, o ar, etc) entre o emissor e o destinatário. Ao chegar à recepção é feito todo o processo inverso com o objetivo de recuperar a mensagem original.

Durante esse processo de transmissão da informação os dados originais podem sofrer alterações devido aos distúrbios introduzidos pelo canal, fazendo com que estes cheguem à recepção modificados ou mesmo sejam perdidos. Conforme a teoria da informação proposta por Claude Elwood Shannon em 1948 existe uma técnica de codificação em que a informação pode ser transmitida, a uma taxa de transmissão inferior a capacidade do canal, sem apresentar erros mesmo na presença do ruído.

Este trabalho teve como objetivo aumentar a confiabilidade da transmissão dos dados enviados por um rádio modem, com o propósito de proteger a informação e assim reduzir os problemas provocados pelo canal ruidoso. Concentraremos nosso estudo no codificador e decodificador de fonte e utilizaremos os códigos corretores de erros para tal.

No decorrer deste relatório faremos uma breve apresentação da ADEUNIS RF, empresa em que o estágio foi realizado. Apresentaremos também um estudo teórico dos códigos corretores de erros, mais especificamente a família dos códigos convolucionais.

No capítulo 4 serão apresentadas as simulações de diferentes algoritmos de correção de erros. Esses algoritmos foram implementados usando o software MATLAB e a análise do seu desempenho foi baseada na taxa de erro por bit ou BER (do inglês Bit Error Rate).

Os últimos capítulos são dedicados à parte prática, em que foram realizados vários experimentos. No capítulo 5 mostramos os resultados dos experimentos. Para finalizar, faremos uma breve conclusão do trabalho realizado.

1.2 EMPRESA

Adeunis RF é uma sociedade francesa de capital fechado criada em 1993 por Hervé Vincent e Pascal Champaney, atuais dirigentes da empresa. É especializada na concepção, fabricação e comercialização de sistemas de rádio transmissão de ponta.

Tabela 1. Setores de aplicação dos produtos ADEUNIS

Setor	Aplicação
Energia	Gestão energética de prédios, comando a distância de sistema de iluminação pública, rádio modems para sistemas de irrigação inteligente.
Indústria	Controle e comando de máquinas e equipamentos industriais, detecção de falhas.
Segurança & controle	Segurança de prédios, dispositivos de alarme, paradas de segurança de equipamentos, aplicações domóticas (home automation).
Transporte	Gestão de estacionamentos ou de veículos, sistemas de sinalização, sistemas de geolocalização, módulos de longo alcance embarcados nas MotoGP.
Médico	Captadores sem fio a serviço de pacientes com insuficiência respiratória, medidores de pressão intracraniana, aparelhos portáteis de radiografia digital.
Militar	Controle de pilha de combustível, drones, sistema de visão noturna, vigilância de fronteiras.
Público geral	Sistemas de geolocalização, controle a distância de robôs, etc



Figura 3. Produtos ADEUNIS RF

2 FUNDAMENTAÇÃO TEÓRICA

2.1 OS CÓDIGOS CORRETORES DE ERROS

Um código corretor de erro visa adicionar redundância a informação que será transmitida, de modo que seja possível recuperar a mensagem original na recepção. Seu emprego permitirá a correção de possíveis erros que ocorram no processo de transporte dos dados pelo canal.

A primeira aplicação bem sucedida de um código corretor de erro foi nas comunicações espaciais com a espaçonave Voyager I, esta utilizava um código de Golay binário para transferir fotos de Júpiter e Saturno. Atualmente, os códigos corretores de erros são largamente utilizados em sistemas de comunicação modernos, em sistemas de armazenamento de dados, em processamento digital de imagens, etc.

Existem técnicas de correção de erros em que no caso da ocorrência de erros é solicitada a retransmissão dos dados, em outras técnicas a correção é feita diretamente na recepção da mensagem, sendo conhecidas como Forward Error Correction (FEC). Estas são bastante usadas na transmissão de sinais digitais em duas principais famílias de códigos: os códigos de blocos e os códigos convolucionais.

Os códigos de blocos são caracterizados pelo fato de que a informação é codificada por blocos de k bits gerando palavras código compostas por n bits, logo a redundância adicionada é $n - k$ bits, conforme podemos observar na Figura 4. Os códigos com essa formação são descritos como código de bloco (n, k) .



Figura 4. Processo de codificação

Os primeiros códigos de blocos foram apresentados por Richard Hamming em 1950, tratava-se de um código simples que era capaz de detectar dois erros e corrigir apenas um. Atualmente uma variação mais complexa desse código é usada pela norma Bluetooth. Outra variação dos códigos de Hamming são os códigos BCH capaz de fazer a correção de erros múltiplos. Os códigos convolucionais serão estudados em detalhes no tópico a seguir.

2.2 OS CÓDIGOS CONVOLUCIONAIS

Os códigos convolucionais, introduzidos por Elias em 1955, são caracterizados por serem gerados a partir de um codificador que possua memória. São bastante utilizados em comunicações espaciais onde são necessários codificadores simples, já que estes estarão no espaço, e métodos de decodificação sofisticados obtidos com hardwares mais complexos.

No seu mecanismo de geração, o codificador convolucional é composto de elementos de memória e somadores módulo-2. Dessa maneira, a sequência codificada (também chamada de palavra-código) não depende unicamente dos n símbolos na entrada do decodificador, mas também do conteúdo armazenado nos registradores. Isso nos fornece uma classe de códigos extremamente eficaz.

Os códigos convolucionais são especificados pelos parâmetros k , n , R , m e v . Onde:

- k é o número de bits da informação na entrada do codificador;
- n é o número de bits de informação na saída do codificador;
- $R = \frac{k}{n}$ é a taxa do código;
- m é chamado de memória do codificador;
- $v = (m + 1) * k$ é a restrição de comprimento do codificador.

Um código gerado por um codificador convolucional é definido por $C(n, k, v)$ e sua performance depende dos parâmetros n , k e v . Os principais códigos utilizados terão a taxa de $R = 1/n$. Códigos que apresentam restrição de comprimento maior possuem menor taxa de erro, no entanto são de implementação complexa, tornando-se inviável na medida em que este parâmetro aumenta. Por exemplo, um código $v = 7$ com 2^6 estados é muito mais complexo que um código com $v = 3$ e 2^2 estados.

Levando em conta sua performance, os códigos convolucionais foram escolhidos como objeto de estudo para uma possível aplicação no nosso projeto. Na prática eles são empregados em comunicação fixa e móvel, Wireless LAN's, comunicação por satélite, etc. Na tabela seguinte, temos alguns exemplos de aplicação dessa classe de códigos. Mostramos também os valores da restrição de comprimento e da taxa do código associado a cada aplicação.

Tabela 2. Principais aplicações dos códigos convolucionais

Aplicação	Restrição de comprimento	Taxa do código
GSM (Global System for Mobile Communications)	6	1/2
WiMax	7	1/2
CCSDS/NASA (Consultative Committee for Space Data Systems)	7	1/2
UWB (802.15)	7	1/3
LTE (3GPP - Long Term Evolution)	7	1/3
CDMA IS-95A	9	1/2

Uma das maneiras de representar um codificador convolucional é por meio de diagramas de blocos, como o código C (2, 1, 3) representado na Figura 5. Este codificador utiliza dois elementos de memória (m_1 e m_2) e dois somadores módulo-2.

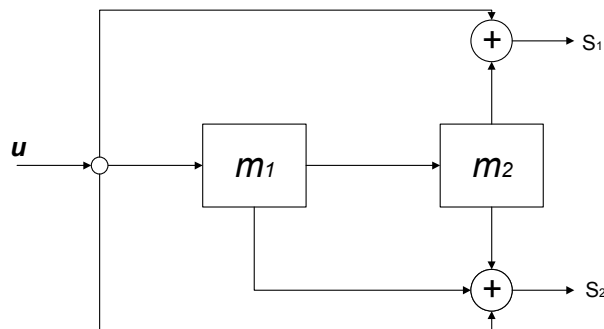


Figura 5. Codificador com taxa $R = 1/2$

Temos que a cada instante de tempo t a entrada u do codificador recebe um bit de informação e gera 2 bits codificados na saída $S = (S_1, S_2)$, portanto sua taxa é $R = 1/2$. As operações para obter a saída S são dadas por:

$$\begin{aligned} S_1 &= u_t + u_{t-2} \\ S_2 &= u_t + u_{t-1} + u_{t-2} \end{aligned} \quad (1)$$

Onde o símbolo + representa adição módulo-2 e u_{t-i} (para $i = 1$ e 2) são as entradas dos estados de tempo anteriores que estão armazenadas nos registradores m_1 e m_2 .

Analogamente, o diagrama de blocos mostrado na Figura 6 representa um código convolucional C (3, 1, 3), cuja taxa do código foi modificada para $R = 1/3$. Neste caso, a saída S será composta por $S = (S_1, S_2, S_3)$. As operações da saída são:

$$\begin{aligned} S_1 &= u_t \\ S_2 &= u_t + u_{t-2} \\ S_3 &= u_t + u_{t-1} + u_{t-2} \end{aligned} \quad (2)$$

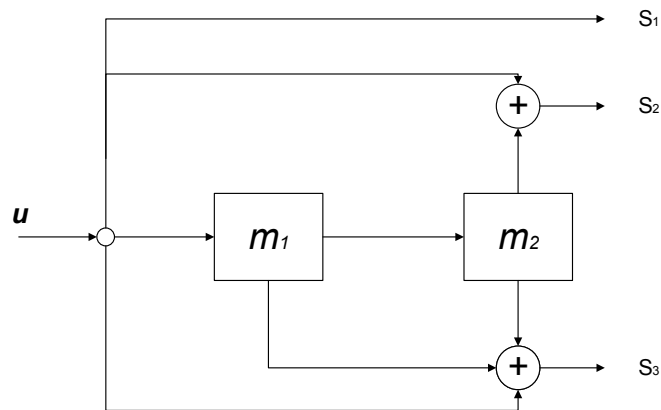


Figura 6. Codificador com taxa do código $R = 1/3$

Para exemplificarmos o funcionamento de um codificador convolucional, tomemos como base o código C (2, 1, 3) da Figura 7. Supondo que inicialmente todos os registradores são inicializados com zero e que na entrada tenhamos a sequência 101b. Como vemos na Figura 7, as saídas correspondentes ao primeiro bit da sequência são $(S_1, S_2) = (1, 1)$, obtidas pelas operações das expressão de (1).

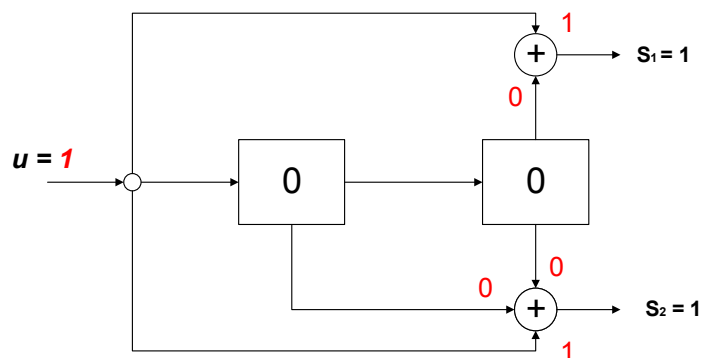


Figura 7. Codificador $R = 1/2$

Conforme surgem novos dados na entrada u , o conteúdo armazenado na memória é atualizado e novas saídas são obtidas a partir das mesmas operações de adição módulo-2 (ver Figuras 8).

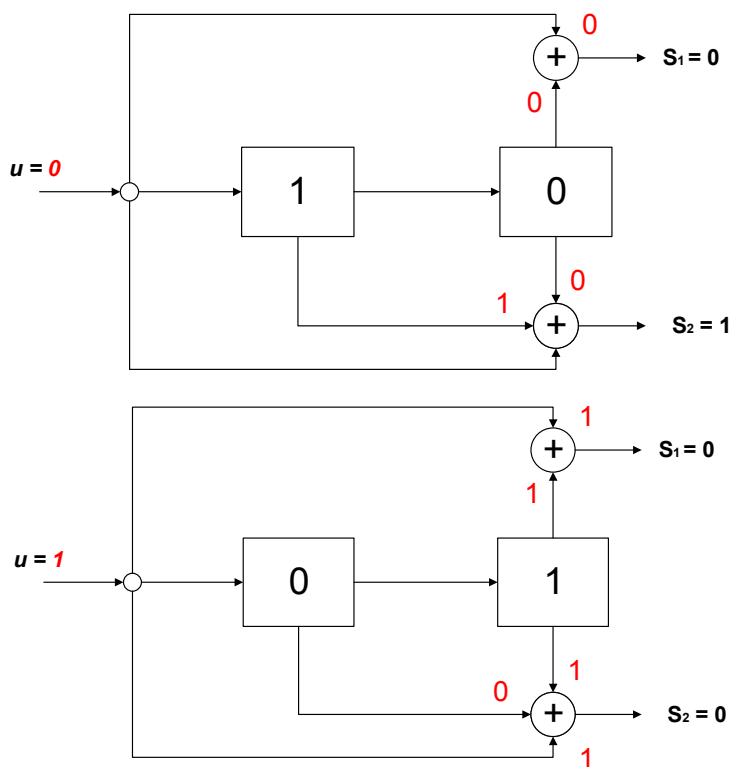


Figura 8. Codificador $R = 1/2$

Ao fim da etapa de codificação da sequência 101b obtemos a sequência codificada 110100b. Para o código $C(3, 1, 3)$ de taxa de codificação $R=1/3$ da Figura 5 e operações representadas em (2) obteríamos: 111 001 100b. Podemos também modificar a restrição de

comprimento adicionando registradores de deslocamento, a lógica de codificação permaneceria a mesma.

Um código convolucional pode ser considerado como uma máquina de estados finita onde os estados dependem tanto dos valores atuais das entradas como também dos valores das entradas anteriores. Desta maneira, podemos representa-los por meio de diagramas de estados, conforme podemos observar na Figura 9 o diagrama de estados do código $C(2, 1, 3)$. Adotamos por convenção neste relatório, que as linhas tracejadas correspondem a uma transição quando o símbolo na entrada é 0 e a linha cheia corresponde ao símbolo um 1 na entrada.

O número de estados possíveis é dado em função da memória do codificador, desse modo para m registradores temos 2^m estados.

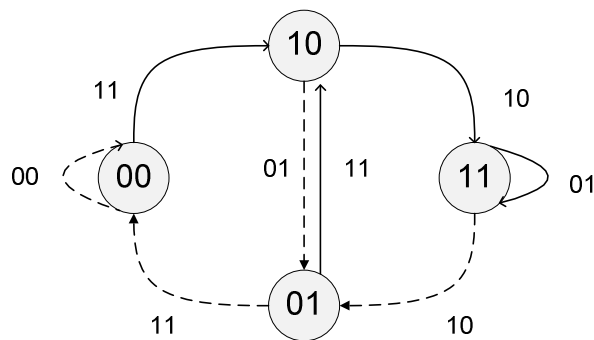


Figura 9. Diagrama de estados de um código $C(2, 1, 3)$

Outra forma bastante utilizada de representação gráfica de um código convolucional é a representação por treliça. Esta coloca em evidência a estrutura temporal, os estados e suas possíveis transições. A treliça da Figura 10 é uma forma de representar o diagrama da Figura 5, onde cada nó representa um estado. Uma sequência de ramos, flecha que liga um estado a outro, forma um caminho. Cada caminho na treliça corresponde a uma única sequência de símbolos de uma mensagem.

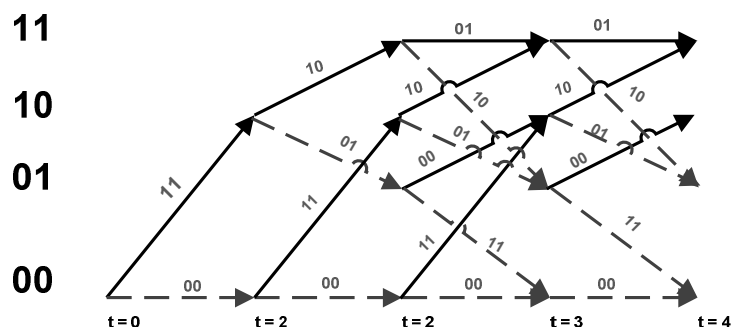


Figura 10. Representação por treliça

2.3 ALGORITMO DE VITERBI

Dentre os decodificadores de códigos corretores de erros o decodificador de Viterbi é um dos mais utilizados em sistemas de comunicação. O Algoritmo de Viterbi (VA) foi proposto por Andrew Viterbi em 1967 e é baseado no princípio de máxima verossimilhança.

Quando uma mensagem ruidosa r é recebida o VA determina uma estimativa y de uma sequência de código válida, entende-se como palavra-código válida uma das possíveis sequências que possam ter sido codificadas e enviadas pelo emissor. A estimativa y é aquela que apresentar a máxima semelhança com r .

Os decodificadores convolucionais podem ser implementados com algoritmos que contam com técnicas ditas de decisão suave (soft decision decoding algorithms) ou de decisão abrupta (hard decision decoding algorithms). Os algoritmos de decisão suave utilizam os valores reais dos sinais recebidos, essa característica apresenta melhoras consideráveis quando se compara com os algoritmos de decisão abrupta em que se utilizam os '0's e '1's detectados no receptor.

Outro conceito importante a ser introduzido para uma melhor compreensão do VA é a distância de Hamming. Podemos definir a distância de Hamming entre duas palavras r e y , como sendo o número de símbolos em que diferem entre si. Palavras-código idênticas possuem uma distância de Hamming nula.

Para entender como o decodificador de Viterbi funciona, mostraremos como ocorre a decodificação da sequência binária 11 01 00. Tal sequência corresponde à sequência discutida e codificada no exemplo da seção anterior.

Observando a treliça da Figura 11, temos que no instante de tempo $t=0$ é recebido o par de símbolos 11b. Calcula-se a distância de Hamming (representada pelos números em azul nos ramos da treliça) entre o par de bits recebidos e as diferentes transições possíveis em cada nó. Inicialmente a treliça parte do estado 00, existindo apenas dois possíveis estados a seguir: o estado cuja saída será "00b" no caso da entrada ter sido um 0 ou o estado cuja saída é "11b" no caso da entrada ter sido um 1.

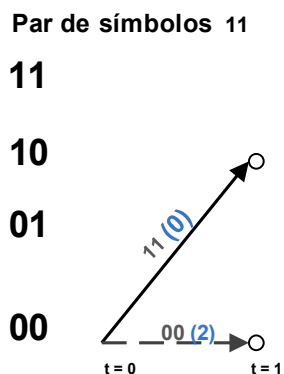


Figura 11. Representação em treliça

A distância de caminho, dada pela distância de Hamming acumulada em cada caminho, é representada pelo número em vermelho em cada nó da treliça representada na Figura 12. Ao fim do último par de bits recebidos obtemos o erro métrico acumulado.

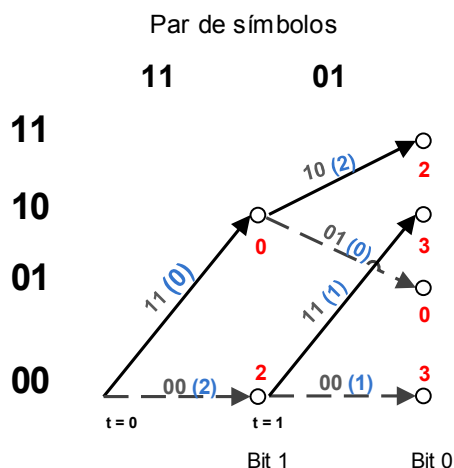


Figura 12. Representação em treliça

No último estado, a sequência retida (a sequência sobrevivente) é a correspondente ao menor erro métrico acumulado, em vermelho na Figura 13. Somente a sequência sobrevivente é

guardada, esta é a sequência mais provável e é utilizada para obter os bits decodificados. As outras sequências são descartadas e a mensagem é recuperada na recepção, no caso estudado a palavra de informação decodificada é 101b. Lembrando que as linhas tracejadas correspondem a uma transição quando o símbolo na entrada é 0 e a linha contínua corresponde ao símbolo ao 1.

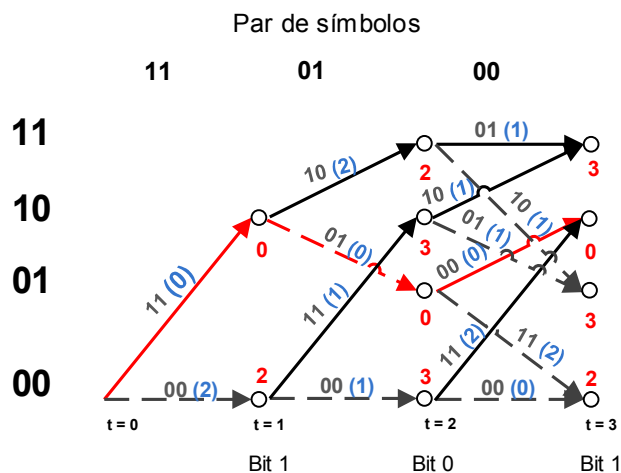


Figura 13. Representação em treliça

Agora, suponha que um erro ocorreu na transmissão do segundo bit 11 11 00 , o algoritmo de Viterbi nos fornece o esquema mais provável e é capaz de retornar corretamente a mensagem decodificada: 101b

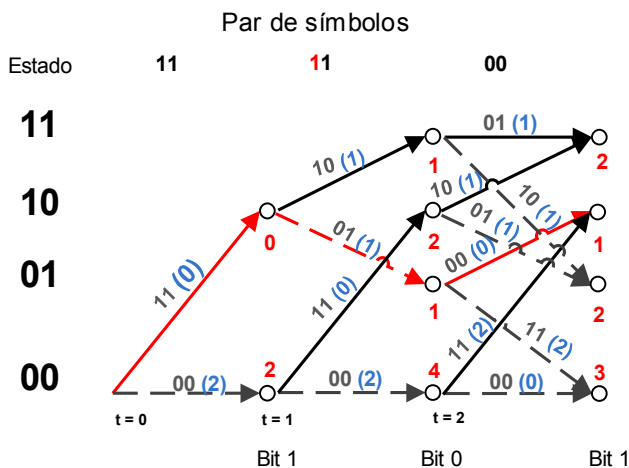


Figura 14. Representação em treliça

A implementação do decodificador de Viterbi, com todas as etapas necessárias para realização do algoritmo será amplamente discutida na seção 3.1 deste relatório. Detalharemos o algoritmo de Viterbi seguindo a estrutura em treliça apresentada nessa seção.

Para atingir melhores performances, os códigos convolucionais são implementados em concatenação com outros códigos. Geralmente são implementados com algoritmos de decodificação uma decisão suave e concatenados com códigos de blocos do tipo Reed-Salomon. Com esta estrutura podemos atingir altas performances que se aproximam da capacidade de transmissão do canal.

3 SIMULAÇÃO MATLAB

Para a simulação do codificador convolucional com o decodificador baseado no algoritmo de Viterbi foi usado o software MATLAB. O objetivo foi de quantificar a performance (pelo cálculo experimental do BER) de um sistema de codificação e decodificação. Para isso, foi feita a simulação de diferentes códigos convolucionais, escolhendo modelos mais simples e relativamente realistas. Estruturamos o programa para simulação seguindo as etapas do Diagrama 1.

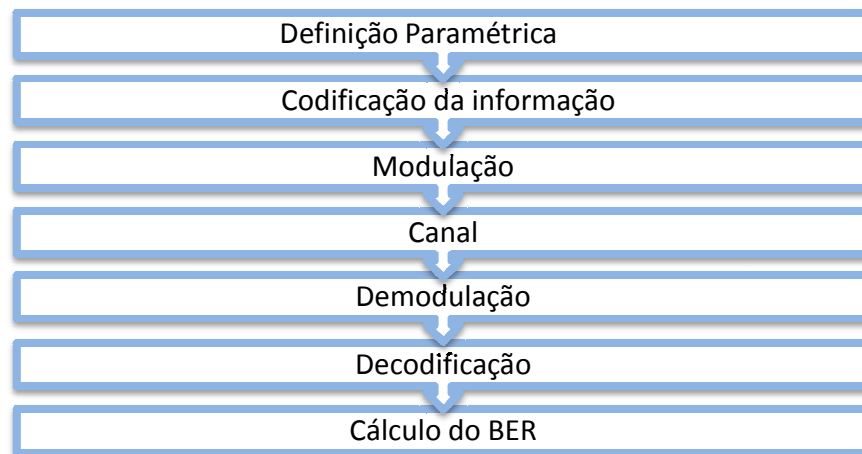


Diagrama 1. Estrutura proposta do código em MATLAB

Na etapa de definição paramétrica a informação é gerada aleatoriamente. Posteriormente, os dados são codificados e modulados (por modulação BPSK) e finalmente são transmitidos por um canal AWGN.

Os blocos de codificação e decodificação são implementados em funções separadas para facilitar sua modificação. A modificação foi feita mudando a taxa ou a restrição de comprimento. Desta maneira, simulamos códigos com taxas $R = 1/2$ para diferentes comprimentos de restrição $v = 3, 5$ e 7 , como também para uma configuração em que $R = 1/3$ e $v = 3$.

No decodificador foram implementamos algoritmos de Viterbi com decisão suave e abrupta. Assim podemos avaliar e comparar a performance de diversos codificadores em cada caso distinto.

A avaliação da performance do sistema foi feita pelo cálculo da taxa por erro de bit ou Bit Error Rate. A curva de BER nos fornece uma boa estimação do comportamento real do sistema, é obtido a partir da seguinte fórmula:

$$BER = \frac{\text{Número total de erros}}{\text{Número total de Bits}}$$

3.1 IMPLEMENTAÇÃO DO DECODIFICADOR DE VITERBI

A implementação do decodificador pelo VA é uma técnica de decodificação relativamente simples quando comparada às outras técnicas de decodificação de códigos convolucionais. No entanto, apesar da simplicidade, esta se mostra muito eficaz e apresenta desempenhos satisfatórios. Em vista disso, reservamos este tópico para discutir o algoritmo de Viterbi em ambiente de programação no Matlab.

Seguindo a estrutura em treliça apresentada anteriormente no tópico 2.3, temos as etapas:

1. Inicialização de todos os registros com 0.
2. A cada unidade de tempo, calcular a distância entre os n bits recebidos e cada caminho possível.
3. A cada unidade de tempo, realizam-se as operações Adicionar-Comparar-Selecionar da seguinte maneira:
 - Adicionar: Calcular das métricas de ramo (Branch Metric-BM) usando a métrica de caminho (Path Metric-PM) do estado anterior e a distância para alcançar o próximo estado.

$$BM = \text{Pre. PM} + \text{distance};$$

- Comparar: A cada nó existem as duas possibilidades de alcançar o próximo estado. Esta operação permite calcular o estado mais provável, a métrica de caminho de cada nó é a métrica de ramo mínima.

$$PM = \min (BM1, BM2) ;$$

- Selecionar: guarda o estado com o PM mínimo que será usado para obter os bits decodificados.

A parte do código em MATLAB correspondente a essas etapas é mostrada a seguir. Uma função completa é dada no anexo.

```

for k = 1:N:length(rvd)/2

    %Passo 1 : Inicialização dos valores

    pm = zeros(4,length(rvd)/2);
    dt = zeros(4,1);

    pm(1,1)=0; pm(2,1)=inf; pm(3,1)=Inf;pm(4,1)=Inf;

    % Como o estado inicial da treliça é zero o PM para esse
    estado é PM = 0 e P = inf para os demais.

    for i = 1:N

        bits = rvd(1, 2*(i+k-1)-1:2*(i+k-1));

        % Passo 2: Cálculo da distância dt. Nesse caso
        utilizaremos a decodificação com decisão suave.

        dt(1,i)=norm([-1,-1]-bits);
        dt(2,i)=norm([-1,+1]-bits);
        dt(3,i)=norm([+1,-1]-bits);
        dt(4,i)=norm([+1,+1]-bits);

        % Passo 3: Adicionar - Comparar - Selecionar

        % Adicionar-Comparar

        pm(1,i+1) = min(pm(1,i)+ dt(1,i), pm(2,i)+dt(4,i));      %
        pm(2,i+1) = min(pm(3,i)+ dt(2,i), pm(4,i)+dt(3,i));      %
        pm(3,i+1) = min(pm(1,i)+ dt(4,i), pm(2,i)+dt(1,i));      %
        pm(4,i+1) = min(pm(3,i)+ dt(3,i), pm(4,i)+dt(2,i));      %

    end

    % Selecionar

    [~,state] = min(pm(:, N + 1));

```

4. Voltar para a etapa 2 até que seja o fim da palavra-código recebida.
5. No fim da treliça teremos armazenados o estado final e as métricas de caminho de todos os estados, eles são utilizados para fazer a etapa *Trace Back* , que é a decodificação dos dados partindo do último bit para o primeiro bit da sequência recebida.
6. Obter a sequência codificada.

O decodificador é organizado como podemos ver no fluxograma da Figura 15.

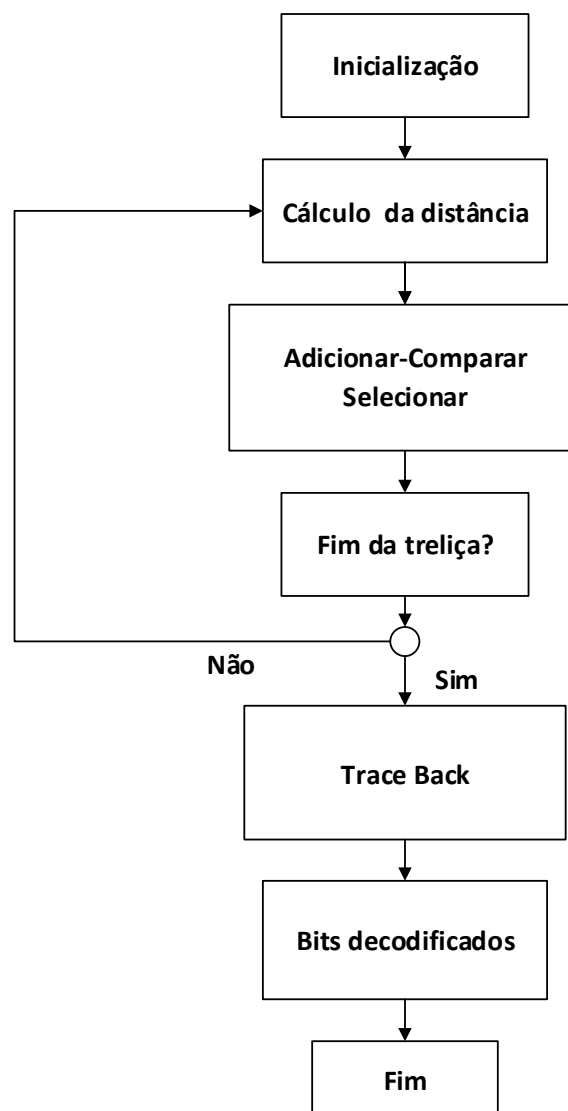


Figura 15. Organização do decodificador de Viterbi

3.2 RESULTADOS

Para avaliar o poder de correção de um determinado tipo de código convolucional, traçamos diversas curvas de BER em função da relação sinal ruído. Comparamos os valores de BER de um sistema cuja informação passa por um processo de codificação de canal a um sistema em que a informação não é codificada. Desta forma, foi possível analisar a melhora de desempenho e a vantagem da codificação da informação.

Nas simulações que seguem o BER foi obtido utilizando um número de bits da ordem de 10^6 . Cada código teve um tempo de simulação de cerca de 8 horas em um computador da marca Dell com processador Pentium.

Na Figura 16, mostramos duas curvas de BER de um sistema que possua um código C(2, 1, 5). O propósito é comparar o desempenho de um mesmo código com algoritmos de decodificação de decisão suave e decisão abrupta. Neste caso, e em todos os outros apresentados posteriormente, a curva preta nas figuras representam a curva de BER quando a informação não é codificada.

Observando os resultados expostos na Figura 16, nota-se que com a decodificação com decisão suave (curva pontilhada vermelha) tivemos um ganho em relação à decisão abrupta (curva contínua vermelha). Devido a essa melhora apresentaremos os demais resultados das simulações usando a decisão suave.

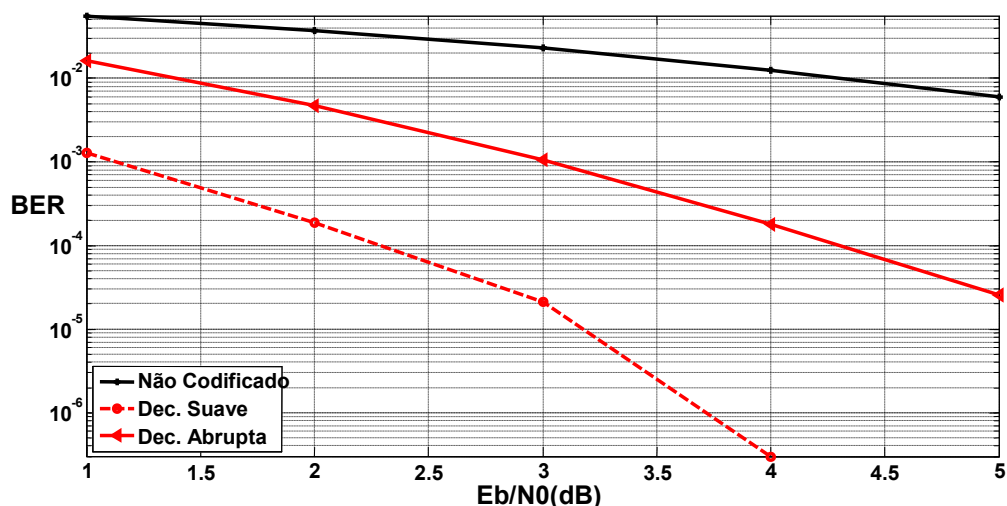


Figura 16. Desempenho de códigos C(2, 1, 5) utilizando diferentes tipos de decisões

A taxa R de um codificador também é um fator importante a analisar, assim fixamos o parâmetro $\nu = 3$ e simulamos sistemas de codificação para $R = 1/2$ e $R = 1/3$. O resultado pode ser visto na Figura 17.

Percebemos que decodificadores com taxa $R=1/3$ (curva pontilhada azul) possui melhor performance em comparação com aqueles que apresentam taxa $1/2$ (curva contínua azul).

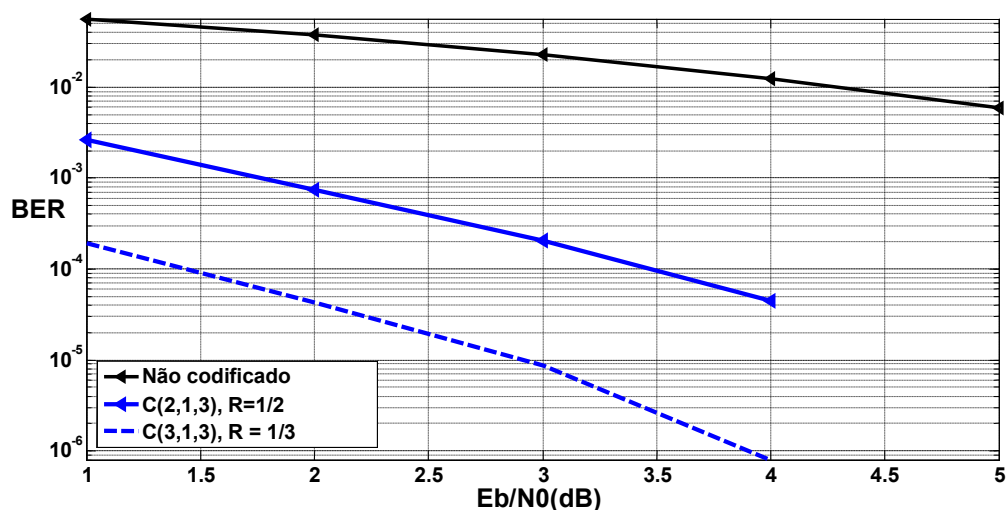


Figura 17. Desempenho de códigos convolucionais utilizando diferentes taxas

Por fim, os resultados apresentados na Figura 18 são obtidos quando fixamos a taxa em $R = 1/2$ e variamos a restrição de comprimento ν . Analisamos o desempenho para diferentes comprimentos de restrição $\nu = 3, 5$ e 7 , onde notamos que códigos com ν maiores possuem melhor desempenho.

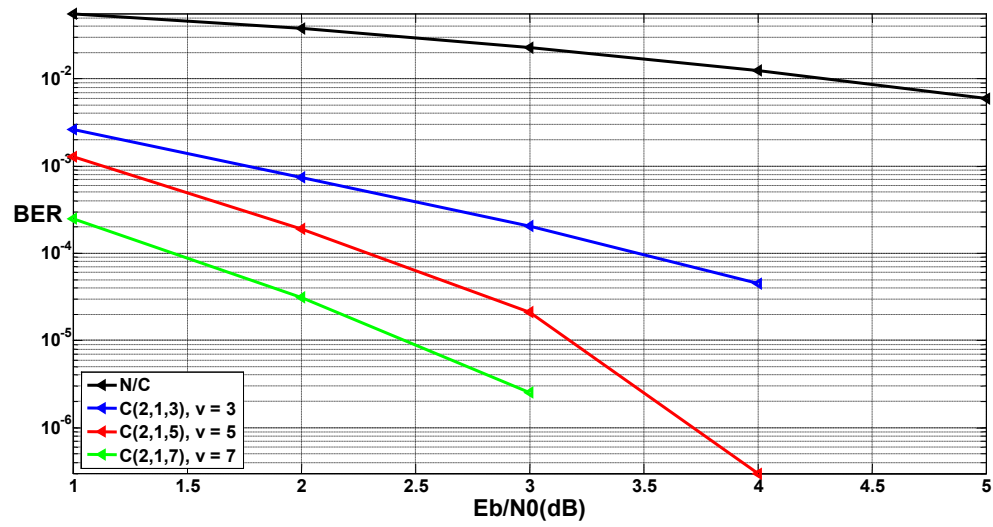


Figura 18. Desempenho de códigos convolucionais utilizando diferentes restrições de comprimentos

4 ESTUDO PRÁTICO

Esse capítulo é destinado à descrição dos procedimentos experimentais, bem como à apresentação dos resultados obtidos em uma situação prática. Trata-se de uma etapa de testes dentro de um contexto realista de utilização dos produtos ADEUNIS RF.

Os ensaios foram realizados nos domínios da empresa em diferentes ambientes de emissão irradiada e conduzida. O objetivo foi de analisar o desempenho dos codificadores estudados anteriormente em situações práticas e também avaliar as eventuais vantagens de sua aplicação.

Antes da realização dos experimentos foi feito um treinamento de alguns dias com os engenheiros da equipe técnica da ADEUNIS RF, que abordava principalmente o funcionamento dos equipamentos listados a seguir.

4.1 MATERIAL UTILIZADO

4.1.1 GERADOR DE SINAIS

O gerador de sinais (GS) fornece um sinal modulado a uma frequência e potência e ajustáveis. Este aparelho foi utilizado para enviar dados conhecidos (previamente codificado) através do canal de transmissão, representado por cada ambiente de realização dos testes.



Figura 19. Gerador de sinais

4.1.2 CÂMARA ANECÓICA

A câmara anecóica é uma sala de experimentação onde as superfícies internas (paredes, teto e piso) são revestidas de um material capaz de absorver as ondas eletromagnéticas. Essa característica permite reproduzir condições de espaço aberto, em que as ondas eletromagnéticas se propagam sem reflexão e são isoladas de fontes externas de ruídos.

O uso da câmara anecóica servirá então para evitar que interferências internas e externas possam modificar os resultados dos testes.

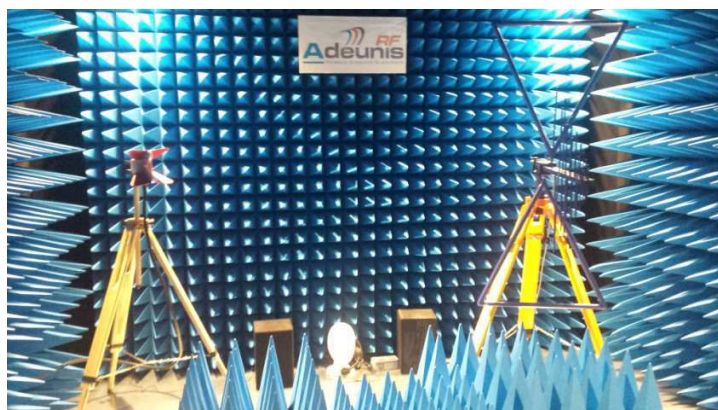


Figura 20. Câmara anecóica da empresa

4.1.3 ANTENA BILOG-PERÍODICA

Esta antena é caracterizada por possuir uma banda passante larga e alta diretividade. Sua gama de frequência de utilização se estende normalmente 200MHz a 1GHz. A antena Bilog-periódica foi escolhida tendo em vista que trabalharemos em frequências abaixo de 1GHz.

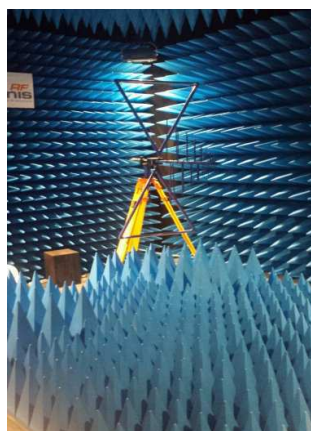


Figura 21. Antena Bilog-periódica

4.1.4 DOCKLIGHT

O DOCKLIGHT é uma ferramenta de teste, análise e simulação de protocolos de comunicação serial RS232 e RS422/485.

O software permite a recepção de dados, assim como o armazenamento em diferentes formatos (ASCII, HEX, DEC, binário). Além disso, proporciona o controle do número de mensagens enviadas ou recebidas pelo dispositivo que está sendo monitorado.

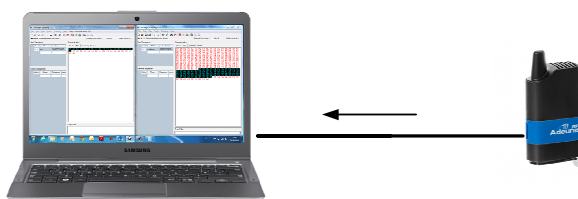


Figura 22. Esquema da comunicação serial entre o Modem e o computador

4.1.5 RÁDIO MODEM ARF868

A gama ARF868, de fabricação da Adeunis RF, é composta por quatro modems que possibilitam a transmissão de dados à distância em half duplex. São produtos de baixo consumo, com funcionamento entre 25mW (ARF868 LP) e 500mW (ARF868 ULR). Este último é capaz de realizar transmissões com um alcance de até 20 km.

Dedicados a comunicação sem fio de ponto a ponto ou multipontos, são projetados no padrão europeu para funcionar na banda de frequência 863-870 MHz. Além disso, possuem a tecnologia de transmissão em banda estreita permitindo a transmissão em vários canais de comunicação, mais de 500 canais de transmissão de 14dBm/2.4kbps.



Figura 23. ARF868 LP e ARF868 ULR Modems.

4.2 PROCEDIMENTOS EXPERIMENTAIS E RESULTADOS

4.2.1 EXPERIÊNCIA 1 - EMISSÃO IRRADIADA EM LABORATÓRIO

Os primeiros ensaios em emissão irradiada foram feitos no laboratório da empresa conforme o esboço representado na Figura 24. Nestas medições utilizamos o gerador de sinais como emissor e o modem ARF como receptor, de maneira que a recuperação dos dados recebidos foi feita por meio do software DOCKLIGHT (faz a comunicação serial entre o modem e o computador).

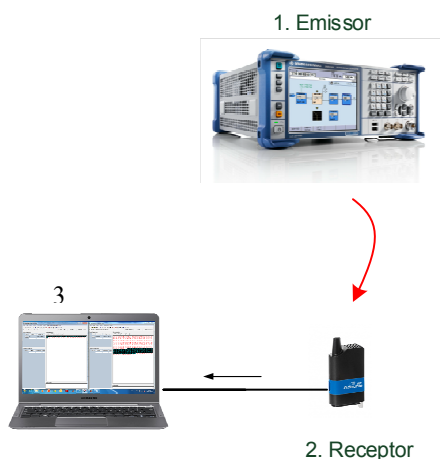


Figura 24. Configuração da transmissão feita no laboratório. 1- Emissor: gerador de sinais 2- Receptor: modem ARF. 3 - Computador com o software para a aquisição dos dados

O emprego do gerado de sinais como emissor permite fazer alterações facilmente na configuração dos dados que desejamos enviar. Assim sendo, os dados foram tratados para que possuíssem a mesma configuração de emissão de um modem ARF, de maneira que foi possível simular uma transmissão entre dois modems.

A principal vantagem do uso do gerador é que podemos ajustar o nível de potência da emissão. No caso em que o gerador emite um sinal com baixa potência estamos simulando situações em que temos um canal muito ruidoso, ou ainda uma situação em que os modems estão muito distantes um do outro. Em ambos as situações a taxa de erros na recepção é alta. Em um caso contrário, onde aumentamos a potência de emissão do sinal emitido pelo GS, teremos uma baixa taxa de erro ou, na maioria das medições, sem erros produzidos pelo canal.

Tendo em vista a impossibilidade de modificar os parâmetros de modulação do modem, utilizaremos então algoritmos de decodificação de decisão abrupta, pois o modem fornece apenas valores de 0's e de 1's.

A mensagem de teste possuía 64 bytes de dados codificados que eram enviados repetidamente. Cada medição gerava arquivos com 500 mensagens idênticas. A Tabela 3 a seguir resume algumas configurações pertinentes a esse experimento.

Tabela 3. Configurações pertinentes ao experimento.

Quantidade de mensagens	500
Tamanho da mensagem	64 bytes
Modem	ARF868 LP
Sensibilidade de recepção	-109 dBm
Alcance ARF868 LP	Até 1km

Devido às interferências do ambiente do experimento alcançamos uma sensibilidade de somente -80 dBm ao invés de -109dBm. As medições foram realizadas entre -76dBm e -78dBm, a este nível de potência temos a taxa de erro por bit situada na região de interesse entre 10^{-2} e 10^{-3} . Para cada nível de potência realizamos 6 medições.

Os resultados em BER desse experimento são apresentados na Figura 25. A curva em preto na figura corresponde ao BER de dados não codificados (N/C). A curva em azul e verde corresponde à curva de BER do codificador C (2, 1, 3) e C(2, 1, 7) respectivamente, ambos os códigos dobram a quantidade de informação a ser enviada. Já a curva de cor vermelha representa a curva de BER do código C(3, 1, 3) que triplica a quantidade de informação a ser enviada.

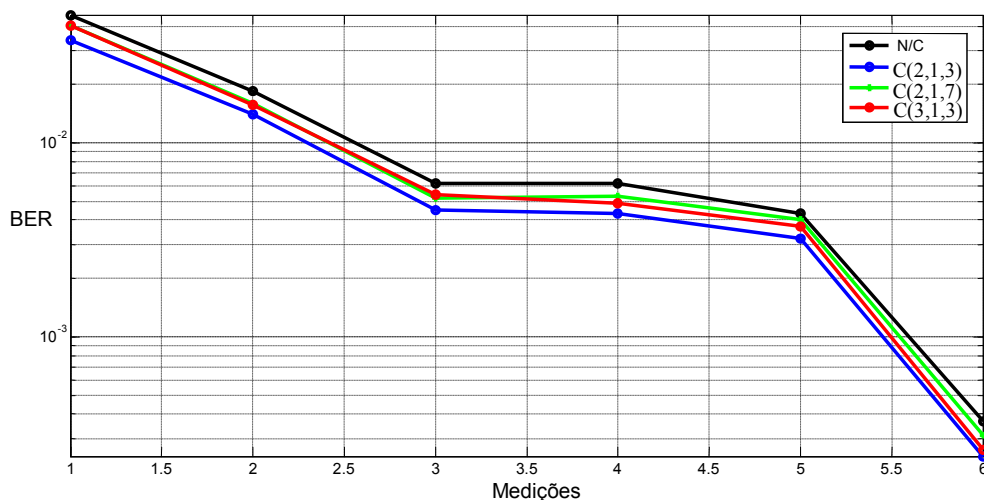


Figura 25. Performances em BER de códigos convolucionais

Constatamos, pela observação dos resultados, que com a codificação dos dados não tivemos um ganho consideravelmente satisfatório. Para compreendermos tais resultados analisamos na Figura 26 o padrão dos erros introduzidos pelo canal. A análise foi feita para 500 mensagens de 512 bits cada, onde na ordenada é representado a quantidade de bits errôneos presente em cada mensagem.

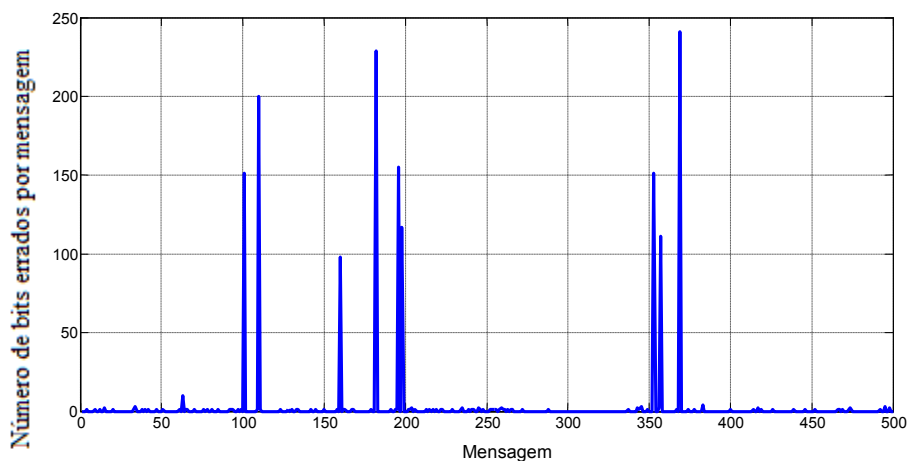


Figura 26. Quantidade de erros em cada pacote.

O padrão de erros mostra que o canal introduz grande quantidade de erros na mensagem, erros sucessivos na maioria dos casos. Essa quantidade de erros excede a capacidade de correção dos códigos convolucionais, visto que a performance do decodificador de Viterbi é fortemente dependente a ocorrência de erros em rajadas (bursts). Dessa maneira o decodificador é incapaz de

achar a sequência correta dos dados, o que nos fornece uma taxa de erro por bit que se aproxima do BER do canal, o que é um comportamento indesejado.

Para minimizar os efeitos dos erros em rajadas pode-se utilizar a técnica de entrelaçamento de bits ou *interleaving*. Esta Consiste em trocar a ordem dos bits antes de sua emissão, de tal forma que na ocorrência de *burst* os erros serão afastados um dos outros, facilitando o processo de decodificação. Isso seria uma possível solução para o nosso problema, no entanto em virtude do número de erros em rajadas, sua utilização não trás melhoras consideráveis (ver Figura 27).

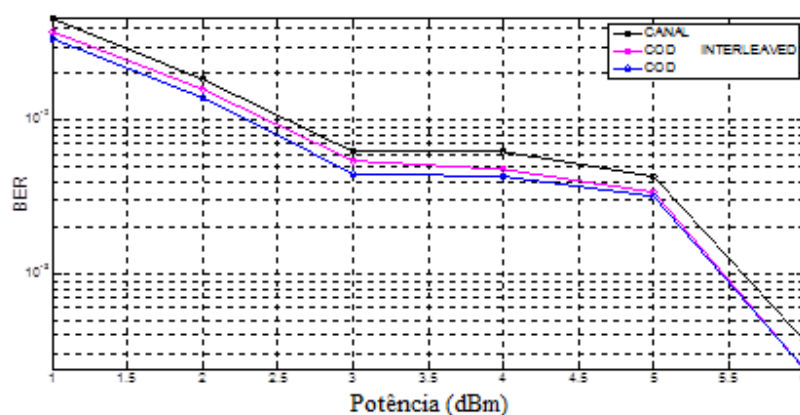


Figura 27. Curvas de BER com o entrelaçamento de bits

4.2.2 EXPERIÊNCIA 2- EMISSÃO CONDUZIDA VIA CABO

A diferença deste experimento com relação aos ensaios precedentes é que os dados são enviados do GS para o modem ARF via um cabo condutor. Com essa configuração podemos obter uma transmissão mais estável e atingir a sensibilidade do modem ARF (-122dbm).

A Figura 28 ilustra o esquema prático da montagem para as medições no laboratório da empresa. O modem empregado nas medições foi o ARF868 URL, que possui um alcance de até 20 km.

Os parâmetros em uso nesse experimento são:

Quantidade de pacotes de dados	200
Tamanho do pacote de dados	64 bytes
Modem	ARF868 URL
Sensibilidade de recepção	-122 dBm
Alcance	Até 20km

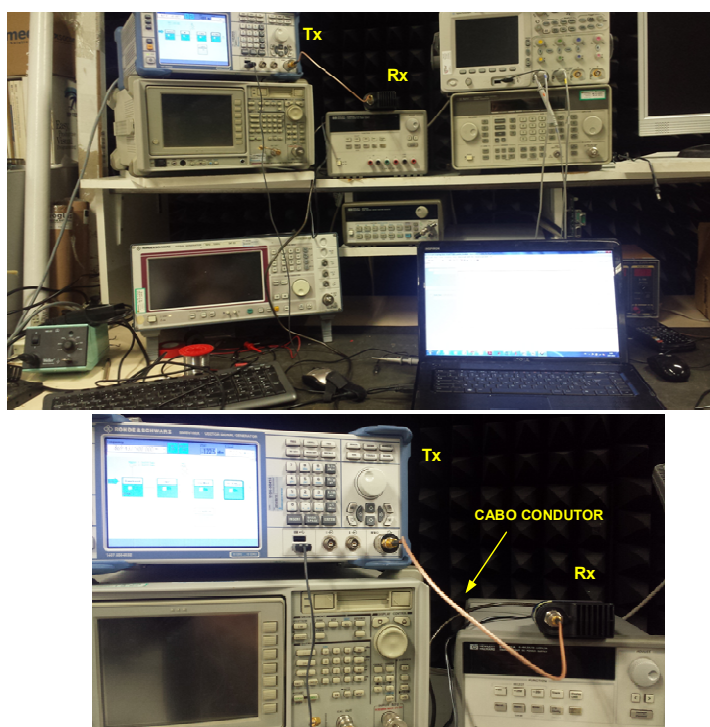


Figura 28. Experiência com condução via cabo no laboratório ADEUNIS. Tx é o GS, Rx é o modem ARF.

Os resultados são apresentados na Figura 29 em BER em função do nível da potência de emissão. As medições foram concentradas na faixa de potência entre -118 dBm e -123dBm e para cada nível de potência realizamos 6 medições, de maneira que o BER foi obtido utilizando $6,144 \times 10^5$ bits. Procedemos dessa maneira a fim de aumentar a confiabilidade dos resultados e diminuir o tempo de processamento no DOCKLIGHT e no MATLAB.

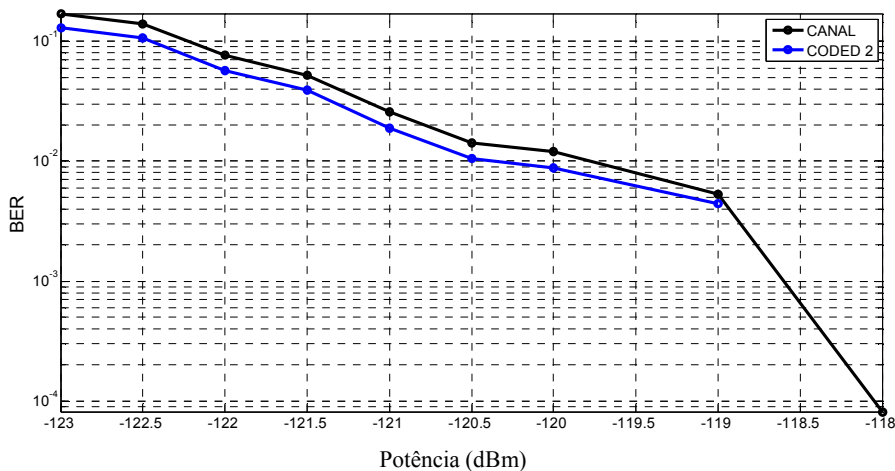


Figura 29. Performance em BER do código C (2, 1, 3).

4.2.3 EXPERIÊNCIA 3- EMISSÃO EM CÂMARA ANECÓICA

Esse experimento consiste em posicionar o modem ARF em visibilidade direta com a antena de emissão e realizar medições em vários níveis de potência. A distância entre o modem e a antena é de 3m o que corresponde a uma atenuação do sinal de -36dBm, a interferência dos cabos são levados em consideração dentro desse valor.

Para emitir o sinal utilizamos o GS conectado à antena Bilog localizada no interior da câmara (ver Figura 30). Essa configuração também permite realizar transmissões mais estáveis.

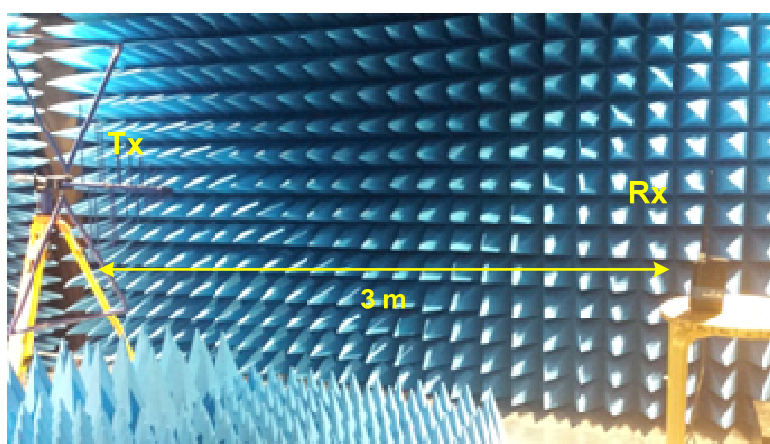


Figura 30. Medições dentro da câmara anecóica

Os parâmetros em uso nesse experimento são:

Quantidade de pacotes de dados	1000
Tamanho do pacote de dados	64 bytes
Modem	ARF868 URL
Sensibilidade de recepção	-122 dBm
Alcance	Até 20km

A partir da análise dos resultados obtidos, mostrados na Figura 31, podemos constatar que o ganho em BER é semelhante ao obtido no experimento 2.

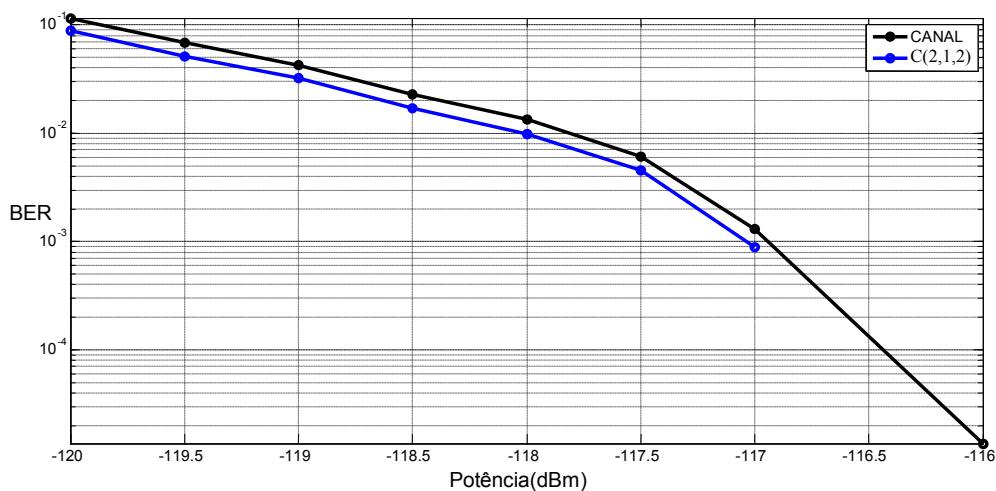


Figura 31. Performance em BER do código C (2, 1, 3) para transmissões na câmara anecóica

4.2.4 EXPERIÊNCIA 4: MEDIÇÕES DE CAMPO

Finalizando a parte experimental, analisamos as transmissões realizadas entre os modems ADEUNIS RF. Os modems foram distanciados a uma distância suficientemente grande para que ocorressem erros na transmissão. O primeiro modem permaneceu nos domínios da empresa, enquanto que o segundo foi deslocado de 10 km e posteriormente de 20 km de distância. Dessa vez utilizamos o modem ARF como emissor e não mais o GS.

Os parâmetros em uso nesse experimento são:

Quantidade de pacotes de dados	1000 a 10km 50 a 20km
Tamanho do pacote de dados	88 bytes
Modem	ARF868 URL
Sensibilidade de recepção	-122 dBm
Alcance	Até 20km

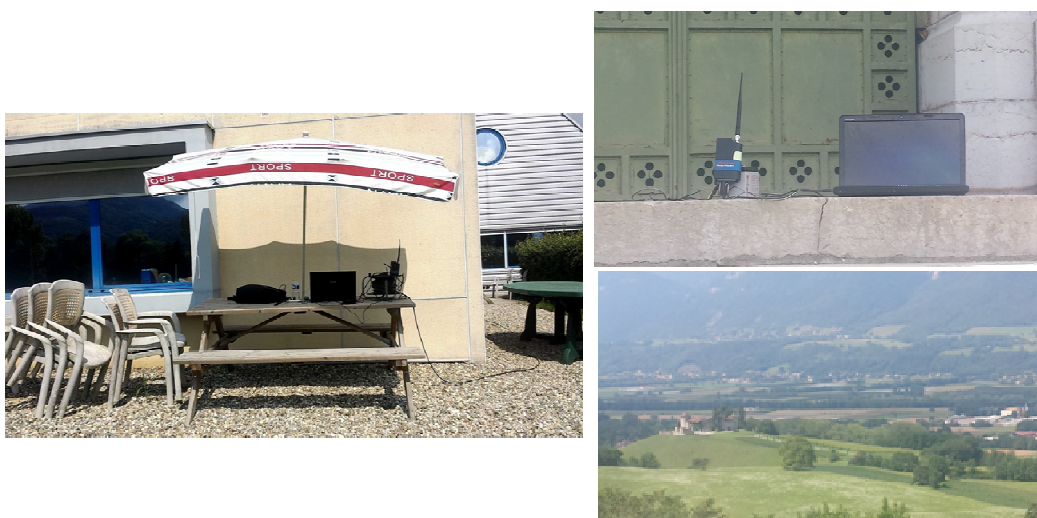


Figura 32. Medição de campo. À esquerda o modem emissor localizado na empresa. A direita, o modem receptor localizado a de 20 km de distância

Os resultados em BER são apresentados na tabela a seguir:

Distância	BER do canal	BER decodificado
10 km	0.0104	0.0076
20 km	0.1405	0.1084

5 CONCLUSÃO

Nesse trabalho apresentamos as atividades realizadas durante o período de estágio na empresa ADEUNIS RF. Discutimos a teoria e o emprego prático de códigos convolucionais. Vimos que ainda que os algoritmos de Viterbi de decodificação suave tenham melhores desempenhos, devido às configurações do modem, acabamos por optar por decodificador de decisão abrupta.

Observando os resultados da taxa de erro de bit das diversas medições realizadas concluímos que, no nosso caso, o emprego apenas do código convolucional não trás melhoras expressivas à eficiência da transmissão, diferentemente do que se foi obtido nas simulações no MATLAB. Isso se explica pelo fato de que o canal simulado (ruído aditivo gaussiano) é bastante diferente da prática, onde os erros chegam em rajadas.

Para chegar a melhores resultados para esse projeto os códigos convolucionais podem ser implementados com a técnica do entrelaçamento de bits e concatenado a outros códigos tais como o código de bloco Reed-Salomon.

Bibliografia

- [1] *Todd K. Moon*, Error Correction Coding Mathematical Methods & Algorithms
- [2] *M. Pednekar, Rohan and B M Dayanand*. Design and Implementation of Convolution Encoder with Viterbi Decoder. International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS).
- [3] Spiral. Software/Hardware Generation for DSP Algorithms. Site disponível em: <http://www.spiral.net/software/viterbi.html> (Página consultada em 26 de maio 2014)
- [4] PROJET MOCAT. Site disponível em: <http://www.irisa.fr/cosi/MOCAT/Travaux/Viterbi/contenu.html#genericite> (Página consultada em 26 de maio 2014)
- [5] MIT 6.02 DRAFT Lecture Notes
- [6] *Infan Ali*. Bit Error Rate (BER) Simulation Using MATLAB. International Journal of Engineering Research and Applications.
- [7] Convolutional code. Site disponível em http://en.wikipedia.org/wiki/Convolutional_code (Página consultada em 18 de agosto de 2014)
- [8] *Mr. Sandesh Y.M et al Int. Journal of Engineering Research and Applications*. Implementation of Convolution Encoder and Viterbi Decoder for Constraint Length 7 and Bit Rate 1/2.
- [9] *D.J. Costello, J. Hagenauer, H. Imai, S.B. Wicker*, Applications of Error Control Coding, IEEE Transactions on Information Theory
- [10] *Henry Hendrix*, Viterbi Decoding Techniques for the TMS320C55x DSP Generation

APÊNDICE A – CÓDIGO NO MATLAB

```
% Simulação de um sistema de transmissão com codificador C(2,1,3) com decodificação por
VA e decisão abrupta.
```

```
clc
```

```
clear all;
```

```
close all;
```

```
tic
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               Declaração Paramétrica
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Eb_N0_dB=1:1:5;
```

```
Eb_N0=10.^(-Eb_N0_dB/20);
```

```
ber=zeros(1,length(Eb_N0));
```

```
% Bit Error Rate
```

```
bl=1E6;
```

```
% Números de bits
```

```
inputx=floor(2*rand(1,bl));
```

```
% Informação randômica
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               Codificador Convolutacional
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
cd = conv_code(inputx);
```

```
fprintf('Encoding completed...\n');
```

```
%----- Modulação BPSK -----%
```

```
%BPSK modulação 0 -> -1; 1 -> 0
```

```
mod_word =2*cd-1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               Canal
%                               Demodulador
%                               Decodificador
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----%
```

```
for k = 1:length(Eb_N0)
```

```
    %Canal AWGN
```

```
    y= mod_word + (Eb_N0(k)/(sqrt(2)))*randn(1,length(mod_word));
```

```

% ----- BER sem decodificação -----%

Y = y > 0;
error_C = sum(cd~=Y);
BERc=error_C/length(Y);
ber_c(1, k)= BERc;

%-----%
%----- Decodificador de Viterbi -----%

decodec = VA_decHard(cd);

% calcula o BER

erreur = sum(decodec~=inputx);
BER(k) = erreur/bl;

end

%----- Plot o BER -----%

theoryBer = 0.5*erfc(sqrt(10.^(Eb_N0_dB/10)));

semilogy(Eb_N0_dB,theoryBer,'k-<','linewidth',2.0) % BER Teórico de um canal AWGN
hold on

semilogy(Eb_N0_dB,BER,'r-<','linewidth',2.0) % Transmissão com codificação
hold on

semilogy(Eb_N0_dB,ber_c,'b-<','linewidth',2.0) % Transmissão sem codificação

axis tight;
grid;
legend('Theory BER','Code (2, 1, 3)','Uncoded');
title('Viterbi decoder performance over AWGN channel for BPSK modulated symbols');
xlabel('Eb/N0 (dB)');
ylabel('BER');

t = toc;
disp(t);

```

conv_code.m

```

% Função do codificador de taxa R = 1/2 e v = 3

function output = conv_code(inputx)

registm1= 0;
registm2= 0;

```

```

b = length(imputx);

output(1, b) = 0;

    for i = 1:1:b

        out2_ = xor(imputx(i), registm1);
        out2 = xor(out2_, registm2);

        out1 = xor(imputx(i), registm2);
        output(1, 2*i-1) = out2;
        output(1, 2*i) = out1;

        registm2 = registm1;
        registm1 = imputx(i);

    end

```

VA_decHard.m

```

% Função do decodificador de taxa R = 1/2 e v = 3 e com algoritmo de decisão abrupta

function decodec = VA_decHard(rvd)

% Inicialização das distâncias e métricas

pm = zeros(4, length(rvd)/2);
metric = zeros(8, length(rvd)/2);

for i=1:length(rvd)/2

    bits = rvd(1, 2*i-1:2*i);
    hd = hamm_dist(bits);

    % Passo 2: Calcula o BM = Pre.PM + Hamming Distance

    metric(1,i) = hd(1,1); % From 00 to 00
    metric(2,i) = hd(4,1); % From 00 to 10
    metric(3,i) = hd(4,1); % From 01 to 00
    metric(4,i) = hd(1,1); % From 01 to 10

    metric(5,i) = hd(3,1); % From 10 to 01
    metric(6,i) = hd(2,1); % From 10 to 11
    metric(7,i) = hd(2,1); % From 11 to 01
    metric(8,i) = hd(3,1); % From 11 to 11

```



```

% Passo 3: Adicionar - Comparar - Selecionar

pm(1, i+1) = min(pm(1, i) + metric(1,i), pm(2,i)+metric(3, i)); % Estado1
00
pm(2, i+1) = min(pm(3, i) + metric(5,i), pm(4,i)+metric(7, i)); % Estado2
01
pm(3, i+1) = min(pm(1, i) + metric(2,i), pm(2,i)+metric(4, i)); % Estado3
10
pm(4, i+1) = min(pm(3, i) + metric(6,i), pm(4,i)+metric(8, i)); % Estado4
11

end

[~,state] = min(pm(:, length(rvd)/2 + 1));

for j=length(rvd)/2:-1:1

    if state == 1

        if (pm(1, j) + metric(1,j)<= pm(2,j)+metric(3, j))
            prev_state = 1;
            bit_dec = 0;
        else
            prev_state = 2;
            bit_dec = 0;
        end
    end
end

if state == 2
    if (pm(3, j) + metric(5,j)<= pm(4,j)+metric(7, j))
        prev_state = 3;
        bit_dec = 0;
    else
        prev_state = 4;
        bit_dec = 0;
    end
end

if state == 3

    if (pm(1, j) + metric(2,j)<= pm(2,j)+metric(4, j))
        prev_state = 1;
        bit_dec = 1;
    else
        prev_state = 2;
        bit_dec = 1;
    end
end

if state == 4

```

```

    if (pm(3, j) + metric(6, j) <= pm(4, j) + metric(8, j))
        prev_state = 3;
        bit_dec = 1;

    else
        prev_state = 4;
        bit_dec = 1;
    end
end

state = prev_state;
decodec(1, j) = bit_dec;

end

```

hamm_dist.m

% Função que retorna a distância de Hamming possível em todos os estados para códigos C(2,1,3)

```

function [HD]=hamm_dist(A)

states = [0 0 0 1 1 0 1 1];
HD=zeros(4,1);
hd = 0;

for ii = 1:1:4
    B = states(1, 2*ii-1:2*ii);
    hd = 0;
    for i = 1:1:2

        if A(1, i) == B(1,i)
            hd = hd;
        else
            hd = hd +1;
        end
    end
end

HD(ii,1)=hd;

end

```