

**USO DE REDES NEURAS NA CORREÇÃO DE ERROS DE TRANSMISSÃO EM SISTEMAS DE
COMUNICAÇÃO VIA SATÉLITE**

GUILHERME PICOLO SALAZAR COSTA

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**USO DE REDES NEURAIIS NA CORREÇÃO DE ERROS DE
TRANSMISSÃO EM SISTEMAS DE COMUNICAÇÃO VIA
SATÉLITE**

GUILHERME PICOLO SALAZAR COSTA

ORIENTADOR: MARCO ANTONIO BRASIL TERADA

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.DM – 382/09

BRASÍLIA/DF: JULHO/2009

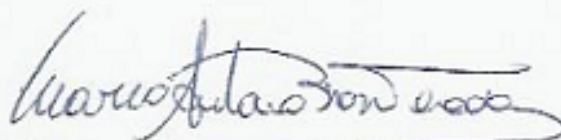
UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

USO DE REDES NEURAIS NA CORREÇÃO DE ERROS DE
TRANSMISSÃO EM SISTEMAS DE COMUNICAÇÕES
VIA SATÉLITE

GUILHERME PICOLO SALAZAR COSTA

DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

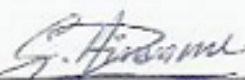
APROVADA POR:



MARCO ANTÔNIO BRASIL TERADA, Dr., ENE/UNB
(ORIENTADOR)



ALEXANDRE RICARDO SOARES ROMARIZ, Dr., ENE/UNB
(EXAMINADOR INTERNO)



GEORGES AMVAME - NZE, Dr., FGA/UNB
(EXAMINADORA EXTERNA)

BRASÍLIA, 03 DE JULHO DE 2009.

FICHA CATALOGRÁFICA

COSTA, GUILHERME PICOLO SALAZAR

Uso de Redes Neurais na Correção de Erros de Transmissão em Sistemas de Comunicação Via Satélite.

xvii, 155p., 210 x 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Redes Neurais

2. Correção de Erros de

Transmissão

3. Sistemas de Comunicação Via Satélite

I. ENE/FT/UnB

II. Título

REFERÊNCIA BIBLIOGRÁFICA

COSTA, G. P. S. (2009). Uso de Redes Neurais na Correção de Erros de Transmissão em Sistemas de Comunicação Via Satélite. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM – 382/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 155p.

CESSÃO DE DIREITOS

AUTOR: Guilherme Pícolo Salazar Costa

TÍTULO: Uso de Redes Neurais na Correção de Erros de Transmissão em Sistemas de Comunicação Via Satélite

GRAU: Mestre

ANO: 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.



Guilherme Pícolo Salazar Costa

UNB – Faculdade de Tecnologia – Departamento de Engenharia Elétrica

CEP 70910-900 – Caixa Postal 04591 - Brasília – DF - Brasil

AGRADECIMENTOS

Agradeço a Deus, Força Suprema do Universo.

Agradeço a minha família, pelos ensinamentos e convívio.

Agradeço a minha mulher Lysia, fonte incansável de estímulo.

Agradeço a minha filha Gabriela, eterna fonte de alento.

Agradeço a meus companheiros de trabalho, pela paciência e aprendizado.

E, finalmente, aos meus mestres, em especial meu orientador, professor Terada, pela paciência e sabedoria com que têm me mostrado o caminho.

RESUMO

USO DE REDES NEURAIIS NA CORREÇÃO DE ERROS DE TRANSMISSÃO EM SISTEMAS DE COMUNICAÇÃO VIA SATÉLITE

Autor: Guilherme Picolo Salazar Costa

Orientador: Marco Antonio Brasil Terada

Programa de Pós-graduação em Engenharia Elétrica

Brasília, julho de 2009

O uso de redes neurais é cada vez mais amplo em nossa sociedade. Especificamente neste trabalho, a técnica de redes neurais é utilizada na demodulação de um sistema de comunicações via satélite, onde o objetivo a ser atingido é o de a rede neural corrigir dois fenômenos interferentes: o AWGN e a atenuação causada por chuva.

Não é intenção determinar qual o melhor tipo de rede neural para cumprir esta finalidade, mas sim, identificar um tipo de rede neural capaz de permitir o desenvolvimento do estudo.

Para validação dos parâmetros utilizados no estudo teórico, são utilizados dados coletados do sistema de comunicações via satélite atualmente em uso pelo CINDACTA I (Primeiro Centro Integrado de Defesa Aérea e Controle do Espaço Aéreo).

Os resultados são bastante promissores, indicando que pelo menos teoricamente o desempenho da técnica utilizada com redes neurais, supera a atual técnica de demodulação em uso pelo sistema TELESAT.

Como o reconhecimento de padrões pela rede neural está limitado a erros de fase menores que 45° , técnicas de correção antecipada de erros (FEC) continuam sendo indispensáveis e complementares.

ABSTRACT

NEURAL NETWORKS FOR TRANSMISSION ERROR CORRECTION IN SATELLITE COMMUNICATION SYSTEMS

Author: Guilherme Picolo Salazar Costa

Supervisor: Marco Antonio Brasil Terada

Electric Engineer Post-graduation Program

Brasília, July of 2009

Neural networks are in widespread use with communication systems. In this work, neural networks techniques are employed to demodulate signals in satellite communication systems, more specifically targeting to compensate two interfering phenomena: AWGN and attenuation caused by rain.

It is not herein intended to achieve the best possible type of neural network, but rather identify a type able to accomplish the objective efficiently.

In addition, real data is collected from satellite communication systems used by CINDACTA I (First Integrated Center of Air Defense and Air Traffic Control) in order to validate the parameters used in the computer simulations.

The results indicate that neural networks can be better than the demodulation technique currently implemented at CINDACTA I, although the use of Forward Error Correction schemes (FEC) is deemed appropriate and complementary.

SUMÁRIO

1 – INTRODUÇÃO.....	1
1.1 – MOTIVAÇÃO E DELIMITAÇÃO.....	3
1.2 – CONTRIBUIÇÃO.....	5
1.3 – UM POUCO MAIS SOBRE O CINDACTA I.....	5
2 – ESPECIFICAÇÕES DE PROJETO.....	7
2.1 – PARÂMETROS DE <i>LINK BUDGET</i>.....	14
3 – FUNDAMENTAÇÃO TEÓRICA.....	16
3.1 – TOPOLOGIA DA REDE VSAT.....	16
3.2 – ALOCAÇÃO (DESIGNAÇÃO) DE CANAIS.....	17
3.3 – MÉTODOS DE ACESSO [6], [13], [19] e [20].....	17
3.3.1 – Métodos de acesso por divisão de canal.....	18
3.3.1.1 – Capacidade dos métodos de acesso por divisão de canal.....	19
3.3.2 – Métodos de acesso aleatório.....	20
3.3.2.1 – Ethernet.....	21
3.3.3 – Métodos de revezamento.....	21
3.4 – TRANSMISSÃO EM BANDA BASE [24].....	21
3.4.1 – Problema da transmissão banda base em linhas telefônicas.....	22
3.5 – MODEM.....	22
3.5.1 – Diferença entre baud e bps.....	23
3.5.2 – Modulação multinível.....	24
3.6 – MODULAÇÃO.....	25
3.6.1 – Sinal analógico.....	27
3.6.1.1 – Sinal de domínio discreto (ou sinal amostrado).....	28
3.6.1.2 – Sinal discreto de variável independente contínua (ou sinal quantizado).....	28
3.6.2 – Sinal digital.....	29
3.6.3 – Conversão analógico/digital.....	31
3.6.4 – Conversão digital/analógico.....	31
3.6.5 – Modulação com portadora analógica e sinal modulante digital (modulação digital).....	31
3.6.5.1 – PSK (<i>Phase Shift Keying</i>)	32

3.6.5.2 – BPSK (<i>Bi Phase Shift Keying</i>).....	35
3.6.5.3 – QPSK (<i>Quadrature Phase Shift Keying</i>).....	36
3.7 – INTERMODULAÇÃO.....	39
3.8 – TÉCNICAS PARA MELHORIA DO SINAL (EQUALIZAÇÃO, CODIFICAÇÃO E DIVERSIDADE DO CANAL).....	39
3.8.1 – Equalização.....	39
3.8.2 – Codificação.....	40
3.8.3 – Diversidade.....	44
3.9 – TIPOS DE SERVIÇO.....	45
3.10 – CODIFICAÇÃO GRAY.....	46
3.11 – CÁLCULO SIMPLIFICADO DE ENLACE.....	47
3.11.1 – EIRP na estação terrena.....	49
3.11.2 – Atenuação no espaço livre (<i>uplink</i>).....	50
3.11.3 – Ocupação de banda.....	50
4 – REDES NEURAIS.....	52
4.1 – O QUE É REDE NEURAL.....	56
4.2 – TOPOLOGIA OU ARQUITETURA.....	56
4.2.1 – Neurônio booleano de Mc Culloch.....	56
4.2.2 – Perceptrons.....	57
4.2.3 – Perceptron multicamada.....	59
4.3 – ALGORITMO DE TREINAMENTO (APRENDIZADO).....	61
4.4 – FUNÇÕES DE ATIVAÇÃO.....	63
4.5 – TÉCNICAS DE DESENVOLVIMENTO DE REDE NEURAL.....	65
4.5.1 – Coleta de dados e separação em conjuntos.....	65
4.5.2 – Configuração da rede.....	65
4.5.3 – Treinamento.....	65
4.5.4 – Teste ou validação.....	67
4.5.5 – Integração.....	67
5 – EFEITOS INTERFERENTES.....	69
5.1 – AWGN.....	69
5.1.1 – Fundamentação de processos estocásticos [17] e [25].....	71
5.2 – ATENUAÇÃO CAUSADA POR CHUVA.....	72

5.2.1 – Modelo ITU-R baseado no modelo DAH.....	74
5.2.2 – Modelo SAM.....	79
6 – LEVANTAMENTO DE DADOS.....	81
6.1 – LEVANTAMENTO DE DADOS COM MODELAGEM AWGN.....	81
6.1.1 – Análise imediata dos dados obtidos.....	83
6.2 – NOVO LEVANTAMENTO COM MODELAGEM AWGN.....	84
6.3 – LEVANTAMENTO DE DADOS METEOROLÓGICOS.....	85
6.3.1 – Análise imediata dos dados obtidos.....	90
6.4 – NOVO LEVANTAMENTO DE DADOS METEOROLÓGICOS.....	90
7 – ESTUDO DE CASO.....	92
7.1 – CONSIDERAÇÕES INICIAIS.....	92
7.1.1 – A escolha da arquitetura.....	92
7.1.2 – A escolha do algoritmo de treinamento.....	93
7.1.3 – Funções de ativação.....	94
7.2 – MODELANDO UM CANAL AWGN.....	94
7.2.1 – Geração do campo amostral (módulo simuladorqpskv4.m).....	94
7.2.2 – Módulo de treinamento (módulo perceptreino.m).....	95
7.2.3 – Módulo de validação (módulo percepronto1.m).....	99
7.2.4 – Análise imediata dos resultados.....	101
7.2.5 – Compatibilizando as grandezas do canal AWGN.....	102
7.2.6 – Nova bateria de testes para o canal AWGN.....	106
7.2.6.1 – Testes com 1.000 símbolos.....	106
7.2.6.2 – Testes com 10.000 símbolos.....	109
7.2.6.3 – Testes com 100.000 símbolos.....	112
7.2.6.4 – Testes com 1.000.000 símbolos.....	113
7.2.6.5 – Análise imediata da nova bateria de testes.....	115
7.3 – MODELANDO UM CANAL ATENUADO.....	116
7.3.1 – Análise imediata dos resultados.....	119
7.4 – MODELANDO UM CANAL AWGN ATENUADO.....	119
7.4.1 – Testes com 1.000 símbolos.....	119
7.4.2 – Testes com 10.000 símbolos.....	122
7.4.3 – Testes com 100.000 símbolos.....	124

8 – CONCLUSÕES.....	128
8.1 – SUGESTÕES PARA TRABALHOS FUTUROS.....	130
REFERÊNCIAS BIBLIOGRÁFICAS.....	131
APÊNDICES	
A – SIMULADORQPSKV4.M.....	134
B – PERCEPTREINO1.M.....	136
C – PERCEPPRONTO1.M.....	137
D – SIMULADORQPSKV5.M.....	139
E – PERCEPTREINO2.M.....	141
F – PERCEPPRONTO2.M.....	142
G – SIMULADORQPSKV6.M.....	144
H – PERCEPTREINO3.M.....	146
I – PERCEPPRONTO3.M.....	147
J- SIMULADORQPSKV7.M.....	150
K – PERCEPTREINO4.M.....	152
L – PERCEPPRONTO4.M.....	153

LISTA DE TABELAS

Tabela 2.1 – Parâmetros da placa TIM (<i>Terrestrial Interface Module</i>) [1].....	9
Tabela 2.2 – Parâmetros de modem [1].....	10
Tabela 2.3 – Características do MPA (<i>Medium Power Amplifier</i>) [1].....	12
Tabela 2.4 – Características do LNA (<i>Low Noise Amplifier</i>) [1].....	12
Tabela 2.5 – Características gerais do sistema [1].....	14
Tabela 2.6 – Correspondência E_b/N_0 versus BER de acordo com FEC [1].....	15
Tabela 2.7 – Características do transponder [1].....	15
Tabela 3.1 – Alternativas angulares recomendadas [10].....	36
Tabela 3.2 – Codificação Gray [22].....	47
Tabela 5.1 – Parâmetros k e α [18].....	75
Tabela 5.2 – Índices pluviométricos por região [18].....	76
Tabela 6.1 – Previsão de interferência para março [26].....	82
Tabela 6.2 – Dados obtidos pelo analisador.....	83
Tabela 6.3 – Novos dados parciais obtidos pelo analisador.....	85
Tabela 6.4 – Precipitação real no dia 26/03/2009.....	86
Tabela 7.1 – Tabela de Q em função de $Q(z)$ [18].....	104
Tabela 7.2 – Tabela de $\text{erfc}(x)$ em função de x [18].....	105
Tabela 7.3 – Tabela comparativa entre o BER teórico e o BER com FEC 3/4.....	106
Tabela 7.4 – Resultados consolidados.....	115
Tabela 7.5 – Repetição da tabela 7.3.....	115

LISTA DE FIGURAS

Figura 1.1 - Box do satélite com abertura de $0,23^\circ$ [1].....	3
Figura 1.2 – Brasão do CINDACTA I.....	3
Figura 1.3 – Área de responsabilidade do CINDACTA I.....	4
Figura 2.1 – Diagrama esquemático de usuários [1].....	7
Figura 2.2 – Diagrama esquemático de uma estação [1].....	8
Figura 2.3 – MMC (<i>Monitoring & Management Center</i>) [1].....	11
Figura 2.4 – NCS (<i>Network Control Center</i>) [1].....	11
Figura 2.5 – Distribuição das estações terrenas.....	14
Figura 3.1 – Topologia da rede VSAT [4].....	16
Figura 3.2 – Terminal VSAT [4].....	17
Figura 3.3 – Transmissão unipolar em a) e b) e bipolar em c).....	21
Figura 3.4 – Exemplo dibit.....	24
Figura 3.5 – Tipos de modulação.....	26
Figura 3.6 – Processos de modulação digital.....	27
Figura 3.7 – Sinal analógico.....	27
Figura 3.8 – Sinal de domínio discreto.....	28
Figura 3.9 – Sinal discreto de variável independente contínua.....	29
Figura 3.10 – Sinal digital.....	30
Figura 3.11 – Conversão analógico/digital.....	30
Figura 3.12 – Modulação PSK.....	33
Figura 3.13 – Modelo de constelação PSK.....	34
Figura 3.14 – Referência dos eixos.....	35
Figura 3.15 – Modelo de mudança de fase.....	35
Figura 3.16 – Constelação BPSK.....	36
Figura 3.17 – Alternativa A.....	37
Figura 3.18 – Mudança de fase na modulação QPSK.....	37
Figura 3.19 – Separação em grupos.....	38
Figura 3.20 – Multiplicação.....	38
Figura 3.21 – Resultado final.....	39
Figura 3.22 – Diversidade espacial tratada por switching [20].....	44
Figura 3.23 – Diversidade espacial tratada por comparadores [20].....	44
Figura 3.24 – Diversidade espacial tratada por controles adaptativos [20].....	45

Figura 3.25 – Disco de codificação mecânica de posição angular [28].....	46
Figura 3.26 – Esboço de um cálculo de enlace.....	48
Figura 3.27 – Diagrama simplificado de emissão terrestre.....	49
Figura 3.28 – Diagrama de enlace.....	50
Figura 4.1 – Neurônio artificial projetado por McCulloch.....	56
Figura 4.2 – Representação matemática do neurônio [9].....	57
Figura 4.3 – Rede de perceptrons proposta por Roseblatt.....	58
Figura 4.4 – Estrutura do método Backpropagation.....	60
Figura 4.5 – Organização em camadas.....	61
Figura 4.6 – Função sigmóide.....	64
Figura 4.7 – Função tangente hiperbólica.....	64
Figura 5.1 – Senóide através de canal AWGN.....	70
Figura 5.2 – Função densidade gaussiana.....	72
Figura 5.3 – Nuvens estratiformes (a) e convectivas (b) [18].....	74
Figura 5.4 – Mapa de distribuição climatológica [18].....	76
Figura 5.5 – Geometria de cálculo do Leff [18].....	77
Figura 5.6 – Cálculo do Leff [18].....	77
Figura 5.7 – Geometria do Leff[18].....	80
Figura 5.8 – Campos e suas componentes copolares e cruzadas [18].....	81
Figura 6.1 – Posição de interferência solar [26].....	81
Figura 6.2 – Imagem satélite para o dia 26/03/2009 [3].....	87
Figura 6.3 – Imagem do radar meteorológico do Gama [3].....	88
Figura 6.4 – Imagem satélite das 20h00 UTC [21].....	91
Figura 7.1 – Arquitetura inicial.....	93
Figura 7.2 – Função transferência tangente hiperbólica [15].....	94
Figura 7.3 – Função transferência linear [15].....	94
Figura 7.4 – Curva do treinamento com traingda.....	96
Figura 7.5 – Curva do treinamento com trainlm.....	97
Figura 7.6 – Curva trainlm com endurecimento.....	98
Figura 7.7 – Constelação com AWGN com $number = 1$	100
Figura 7.8 – Constelação corrigida com $number = 1$	101
Figura 7.9 – Geração de 1.000 símbolos pseudo aleatórios.....	107
Figura 7.10 – Treinamento da rede neural com 1.000 símbolos.....	107
Figura 7.11 – Geração de 1.000 novos símbolos pseudo aleatórios.....	108

Figura 7.12 – Correção sugerida pela rede neural.....	108
Figura 7.13 – Geração de 10.000 símbolos pseudo aleatórios.....	109
Figura 7.14 – Treinamento da rede neural com 10.000 símbolos.....	110
Figura 7.15 – Geração de 10.000 novos símbolos pseudo aleatórios.....	111
Figura 7.16 – Correção sugerida pela rede neural.....	111
Figura 7.17 – Treinamento da rede neural com 100.000 símbolos.....	112
Figura 7.18 – Correção sugerida pela rede neural.....	113
Figura 7.19 – Geração de 1.000.000 de símbolos.....	114
Figura 7.20 – Treinamento da rede neural com 1.000.000 de símbolos.....	114
Figura 7.21 – Símbolos atenuados.....	117
Figura 7.22 – Treinamento da rede.....	117
Figura 7.23 – Novos símbolos atenuados.....	118
Figura 7.24 – Correção sugerida pela rede neural.....	118
Figura 7.25 – Geração de 1.000 símbolos pseudo aleatórios.....	120
Figura 7.26 – Treinamento de rede neural com 1.000 símbolos.....	120
Figura 7.27 – Geração de 1.000 novos símbolos pseudo aleatórios.....	121
Figura 7.28 – Correção sugerida pela rede neural.....	121
Figura 7.29 – Geração de 10.000 símbolos pseudo aleatórios.....	122
Figura 7.30 – Treinamento da rede neural com 10.000 símbolos.....	123
Figura 7.31 – Geração de 10.000 novos símbolos pseudo aleatórios.....	123
Figura 7.32 – Correção sugerida pela rede neural.....	124
Figura 7.33 – Geração de 100.000 símbolos pseudo aleatórios.....	125
Figura 7.34 – Treinamento da rede neural com 15.000 símbolos.....	125
Figura 7.35 – Geração de 100.000 novos símbolos pseudo aleatórios.....	126
Figura 7.36 – Correção sugerida pela rede neural.....	126

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

ADALINE – *Adaptative Linear Network* – Rede Linear Adaptativa

ADPCM – *Adaptive DPCM*

ADSL – *Asymmetric Digital Subscriber Line*

AM – *Amplitude Modulation*

ANATEL – Agência Nacional de Telecomunicações

ASK – *Amplitude Shift Keying*

ATM – *Asynchronous Transfer Mode*

AWGN – *Additive White Gaussian Noise*

BER – *Bit Error Rate*

BPSK – *Bi Phase Shift Keying*

CADE – Conselho Administrativo de Defesa Econômica

CCITT – *Comité Consultatif International de Telegraphie et Telephonie*

CDMA – *Code Division Multiple Access*

CMA – Centro Meteorológico de Aeródromo

CMM – Centro Meteorológico Militar

CMV – Centro Meteorológico de Vigilância

CNMA – Centro Nacional Meteorológico da Aeronáutica

CPTEC – Centro de Previsão de Tempo e Estudos Climáticos

CSMA/CD – *Carrier Sense Multiple Access with Collision Detection*

CST – *Central Signaling Terminal*

CW – *Continuous Wave*

DAH – Dissanayake Haidara Allnutt

DAMA – *Demand Assignment Multiple Access*

DARPA – *Defense Advanced Research Projects Agency*

DPCM – *Differential Pulse Code Modulation*

DSL – *Digital Subscriber Line*

DSP – *Digital Signal Processor*

DTCEA – Destacamento de Controle do Espaço Aéreo

EIRP – *Effective Isotropic Radiated Power*

EMS – Estação Meteorológica de Superfície

ERFC – *Error Function Complementary*

FAX - Facsimile

FDDI – *Fiber Distributed Data Interface*
FDMA – *Frequency Division Multiple Access*
FEC – *Forward Error Correction*
FET – *Field Effect Transistor*
FI – *Frequência Intermediária*
FIR – *Flight Information Region*
FM – *Frequency Modulation*
FSS – *Fixed Satellite Service*
FSK – *Frequency Shift Keying*
GMT – *Greenwich Mean Time*
HPA – *High Power Amplifier*
HUB – *Estação central da rede*
IDU – *InDoor Unit*
IEEE – *Institute of Electrical and Electronics Engineers*
IF – *Intermediate Frequency*
INNS – *International Neural Networks Society*
INPE – *Instituto Nacional de Pesquisas Espaciais*
ISDN – *Integrated Services Digital Network*
ISP – *Internet Service Provider*
ISI – *InterSymbol Interference*
ITU – *International Telecommunications Union*
LGT – *Lei Geral de Telecomunicações*
LMS – *Least Mean Square*
LNA – *Low Noise Amplifier*
LPCD – *Linha Privada de Comunicação de Dados*
MCPC – *Multiple Channels per Carrier*
MMC – *Monitoring & Management Center*
MPA – *Medium Power Amplifier*
MSE – *Mean Square Error*
MSS – *Mobile Satellite Service*
NCC – *Network Control Center*
NCS – *Network Control System*
ODU – *OutDoor Unit*
PAM – *Pulse Amplitude Modulation*

PAMA – *Permanent Assignment Multiple Access*
PCM – *Pulse Code Modulation*
PDS – *Processamento Digital de Sinais*
PM – *Phase Modulation*
PPM – *Pulse Position Modulation*
PSK – *Phase Shift Keying*
PWM – *Pulse Width Modulation*
QAM – *Quadrature Amplitude Modulation*
QPSK – *Quadrature Phase Shift Keying*
RDA – *Região de Defesa Aeroespacial*
REDEMET – *Rede de Meteorologia do Comando da Aeronáutica*
RF – *Rádiofrequência*
RLS - *Recursive Least Square*
RMS – *Root Mean Square*
RN – *Rede Neural*
RTT – *Remote Traffic Terminal*
SAM – *Simple Attenuation Model*
SCPC – *Single Channel per Carrier*
SCU – *Signalling & Control Unit*
SER – *Symbol Error Rate*
SISCEAB – *Sistema de Controle do Espaço Aéreo Brasileiro*
SNR – *Signal to Noise Ratio*
SSMA – *Spread Spectrum Multiple Access*
TCM – *Trellis Coded Modulation*
TDM – *Time Division Multiplex*
TDMA – *Time Division Multiple Access*
TELESAT – *Canalização determinística satelital do Comando da Aeronáutica*
TIM – *Terrestrial Interface Module*
TOPNET – *Canalização determinística fornecida por empresa de telecomunicação*
TWT – *Traveling Wave Tube*
UCEA – *Unidade de Controle do Espaço Aéreo*
UTC – *Universal Time Coordinated*
VSAT – *Very Small Aperture Terminal*

1 - INTRODUÇÃO

Desde tempos imemoriais, o céu exerce grande fascínio sobre a raça humana. Seja pela sua imensidão, seja pela alternância entre o dia e a noite, seja pelos fenômenos que ocorrem de maneira magnífica.

Certamente, a evolução de nossa espécie, está intimamente ligada ao maior conhecimento que vamos paulatinamente galgando acerca do espaço e dos fenômenos que nele ocorrem.

A onipresença da informação é fundamental para o nosso desenvolvimento cada vez mais acelerado, sendo esta uma barreira a ser vencida a qualquer custo.

Talvez a primeira pessoa a expor a idéia da utilização de satélites para o uso em comunicações, tenha sido o britânico Arthur C. Clarke, que em 1.945 publicou no periódico do Reino Unido “*Wireless World*”, o *paper* intitulado “*Extra-terrestrial Relays*”, com o primeiro conjunto de princípios de utilização de satélites de comunicações localizados em órbitas geoestacionárias (a 36.000 km de altitude) [27].

Obviamente que, com o desenvolvimento de meios de comunicações cada vez mais complexos, há necessidade de regulamentação e fiscalização, tanto a nível nacional como mundial.

A ANATEL (Agência Nacional de Telecomunicações) é autarquia especial criada pela LGT (Lei Geral de Telecomunicações), sendo administrativamente independente, financeiramente autônoma, não se subordinando hierarquicamente a nenhum órgão de governo. Do Ministério das Comunicações, a ANATEL herdou os poderes de outorga, regulamentação e fiscalização. Dentre as atribuições da ANATEL, merecem destaque [2]:

- implementar, em sua esfera de atribuições, a política nacional de telecomunicações;
- expedir normas quanto à outorga, à prestação e à fruição dos serviços de telecomunicações no regime público;
- administrar o espectro de radiofrequências e o uso de órbitas, expedindo as respectivas normas;
- expedir normas sobre prestação de serviços de telecomunicações no regime privado;

- expedir normas e padrões a serem cumpridos pelas prestadoras de serviços de telecomunicações quanto aos equipamentos que utilizarem;
- expedir ou reconhecer a certificação de produtos, observados os padrões e normas por ela estabelecidos;
- reprimir infrações dos direitos dos usuários; e
- exercer, relativamente às telecomunicações, as competências legais em matéria de controle, prevenção e repressão das infrações da ordem econômica, ressalvadas as pertencentes ao Conselho Administrativo de Defesa Econômica (Cade).

A ITU (International Telecommunication Union) é a principal agência das Nações Unidas para assuntos referentes a tecnologias de comunicações. Por aproximadamente 145 anos a ITU tem coordenado o compartilhamento global do espectro de rádio, promovendo a cooperação internacional através da designação de órbitas de satélite. Dentre as atribuições da ITU, merecem destaque [10]:

- permitir o desenvolvimento sustentável das telecomunicações e redes de informação;
- facilitar o acesso universal a estas tecnologias;
- desenvolver padrões de uso;
- gerenciar o espectro de rádio-frequência e órbitas de satélites.

Sistemas de comunicações via satélite, são constituídos em sua maioria por satélites geoestacionários que variam dentro de um box, terminais terrenos e estações *hub*.

Os terminais terrenos são conhecidos como VSATs (*Very Small Aperture Terminals*), sendo terminais constituídos por antenas de pequeno diâmetro (variando de 80 cm a 2 m aproximadamente).

As estações *hub* são estações principais, com antenas maiores, que se comunicam com todas as VSATs, funcionando como ponto de interconexão.

O uso de tecnologias de comunicações via satélite são justificáveis em aplicações que exigem espalhamento da mesma informação em uma região geográfica extensa, em localidades remotas que não dispõem de infra-estrutura terrestre adequada, em locais que necessitam de rápida implantação, em situações estratégicas e serviços que exigem

redundância e alta disponibilidade. Na Figura 1.1 temos o exemplo de como a posição orbital de um satélite varia dentro de uma determinada área.

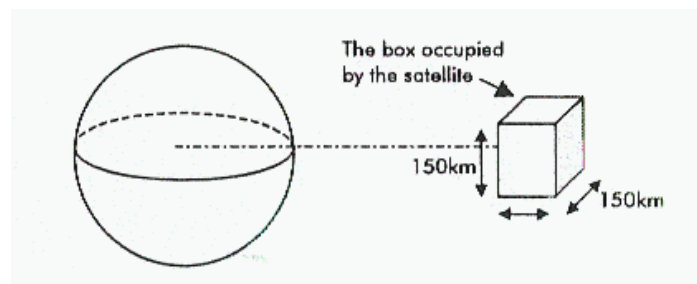


Figura 1.1 – Box do satélite com abertura de $0,23^\circ$ [1].

A primeira banda a ser explorada comercialmente foi a banda C (utiliza o espectro de frequência em torno de 4 GHz para downlink e 6 GHz para uplink). Esta banda apresenta elevada interferência com outros sistemas terrestres, dificultando principalmente a recepção, já que há compartilhamento com os enlaces de microondas. Com o aumento da demanda e a saturação do espectro da banda C, houve uma rápida expansão para o uso das bandas Ku e Ka.

1.1 – MOTIVAÇÃO E DELIMITAÇÃO

A motivação desta Dissertação de Mestrado, é a concatenação de conhecimentos acadêmicos e práticos adquiridos ao longo de anos de trabalho no CINDACTA I (Primeiro Centro Integrado de Defesa Aérea e Controle de Tráfego Aéreo), Organização Militar do Comando da Aeronáutica localizada em Brasília, responsável pela Defesa Aérea da RDA-1 (Região de Defesa Aeroespacial 1) e responsável pelo Controle de Tráfego Aéreo da FIR-Brasília (*Flight Information Region* – Brasília).



Figura 1.2 – Brasão do CINDACTA I.

A RDA-1 e a FIR-Brasília, correspondem a áreas que abrangem geograficamente partes da Região Centro-Oeste, Sudeste, Nordeste e Norte, conforme mostrado na Figura 1.3.

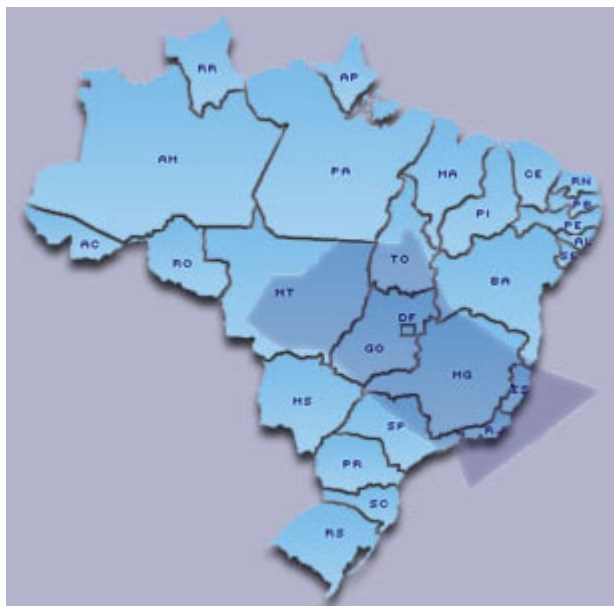


Figura 1.3 – Área de responsabilidade do CINDACTA I.

Com a finalidade de cumprir a sua missão precípua, o CINDACTA I dispõe de canalização “autônoma” via satélite (sistema TELESAT) e canalização contratada junto a empresas de telecomunicações (sistema TOPNET). Estas canalizações são determinísticas e de alto custo, sendo complementares e oferecendo uma solução com alta disponibilidade e redundância.

Como visto, as comunicações via satélite são fundamentais na manutenção de atividades essenciais, diretamente relacionadas com a soberania aeroespacial. Esta canalização é utilizada no tráfego de informações (dados e voz), cujos produtos mais nobres são a visualização radar (síntese de dados de diversas fontes, capazes de permitir a visualização do posicionamento espacial de aeronaves), radiocomunicações (climatização de voz de equipamentos VHF/UHF de diversas fontes) e telefonia.

O sistema satelital “autônomo” (sistema TELESAT) utiliza modulação QPSK com FEC de Viterbi, e vislumbra-se a possibilidade de uso de redes neurais na correção de erros de transmissão em sistemas de comunicações via satélite. A intenção é pesquisar o uso de

redes neurais no processo de demodulação, em substituição aos filtros convencionais atualmente utilizados. Em resumo, pretende-se simular a validade de uso de redes neurais na equalização (uma das técnicas para melhoria do sinal em *links wireless*) do canal [11] e [23].

1.2 - CONTRIBUIÇÃO

Através da simulação e desenvolvimento de modelos computacionais através do MATLAB 7.0 (versão 7.0.0.19920 (R14) de 06/05/2004) [15] e [7], além da realização de medições e levantamentos de informações meteorológicas, pretende-se contribuir de maneira positiva no avanço do uso de redes neurais na correção de erros de transmissão, em especial no sistema utilizado pelo CINDACTA I, que utiliza modulação QPSK.

1.3 - UM POUCO MAIS SOBRE O CINDACTA I

Para que seja possível ao CINDACTA I cumprir sua missão, há necessidade de que a sede localizada em Brasília, receba informações remotas de seus 18 DTCEAs (Destacamentos de Controle do Espaço Aéreo) e 17 UCEAs (Unidades de Controle do Espaço Aéreo).

O DTCEA é uma unidade assistida com grande infra-estrutura que, geralmente possui radar. A UCEA é uma unidade não assistida com menor infra-estrutura.

DTCEAs:

- DTCEA-AN: Anápolis
- DTCEA-BQ: Barbacena
- DTCEA-BR: Brasília
- DTCEA-BW: Barra do Garças
- DTCEA-CC: Cachimbo
- DTCEA-CF: Confins
- DTCEA-CY: Cuiabá
- DTCEA-GA: Gama
- DTCEA-GI: Chapada dos Guimarães
- DTCEA-LS: Lagoa Santa
- DTCEA-PCO: Pico do Couto

- DTCEA-PIE: Piedade
- DTCEA-SRO: São Roque
- DTCEA-STA: Santa Teresa
- DTCEA-TNB: Tanabi
- DTCEA-TRM: Três Marias
- DTCEA-YS: Pirassununga
- DTS: Destacamento de Telecomunicações por Satélite

UCEAs:

- UCEA-APR: Alto Paraíso
- UCEA-BRT: Buritis
- UCEA-KP: Campinas
- UCEA-CMP: Campos
- UCEA-CNA: Canarana
- UCEA-JT: Jataí
- UCEA-GP: Gurupi
- UCEA-IPM: Ipameri
- UCEA-MCL: Montes Claros
- UCEA-PAN: Porto Alegre do Norte
- UCEA-RQ: Ressaquinha
- UCEA-SLN: São Luis do Norte
- UCEA-SOI: Sorriso
- UCEA-TSR: Tangará da Serra
- UCEA-TEO: Teófilo Otoni
- UCEA-UBE: Uberaba
- UCEA-VAR: Varginha

2 – ESPECIFICAÇÕES DE PROJETO

Na sua concepção original, o sistema TELESAT foi desenvolvido para substituir o antigo sistema de tropodifusão do então Ministério da Aeronáutica, permitindo o carregamento da canalização do SISCEAB (Sistema de Controle do Espaço Aéreo Brasileiro) no satélite BRASILSAT B3, em 1999. A Figura 2.1 exemplifica uma topologia típica do sistema.

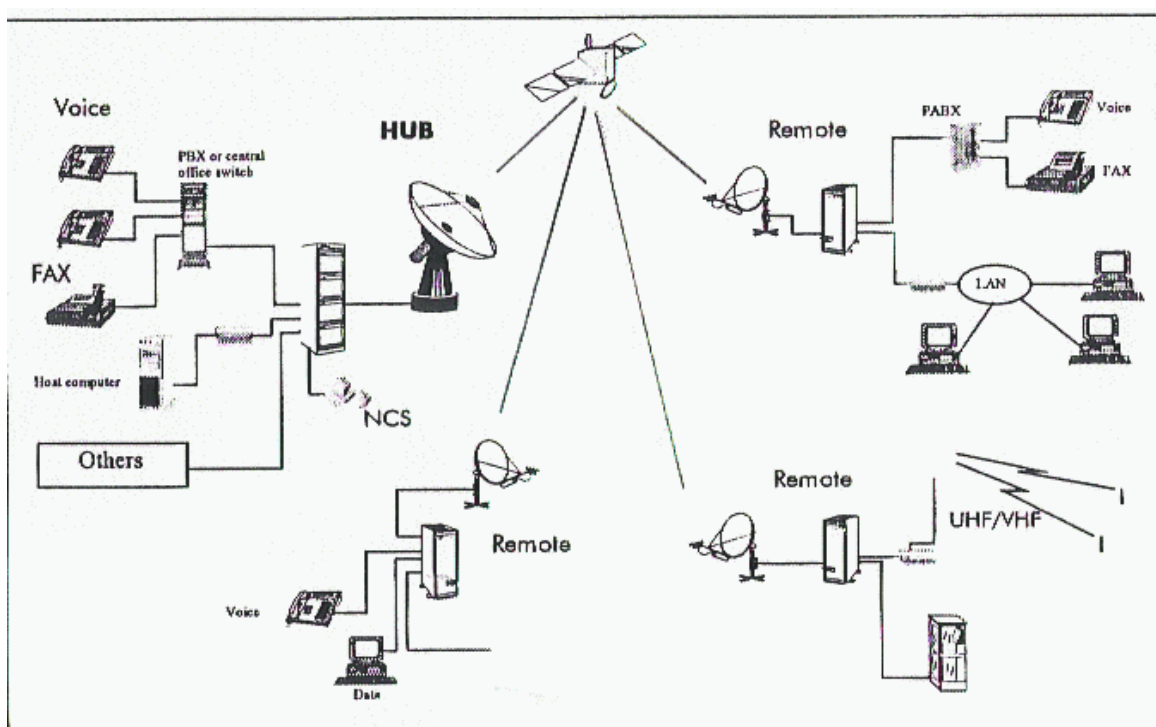


Figura 2.1 – Diagrama esquemático de usuários [1].

O sistema TELESAT, trabalha com a alocação (designação) de canal tipo PAMA (*Permanent Assignment Multiple Access*) e DAMA (*Demand Assignment Multiple Access*).

A alocação tipo PAMA garante o tráfego ponto a ponto através de canais reservados, para aplicações prioritárias de voz e dados, sendo os canais multiplexados através de técnicas FDM (*Frequency Division Multiplex*).

A Figura 2.2 mostra de maneira resumida, a composição de uma estação terrestre.

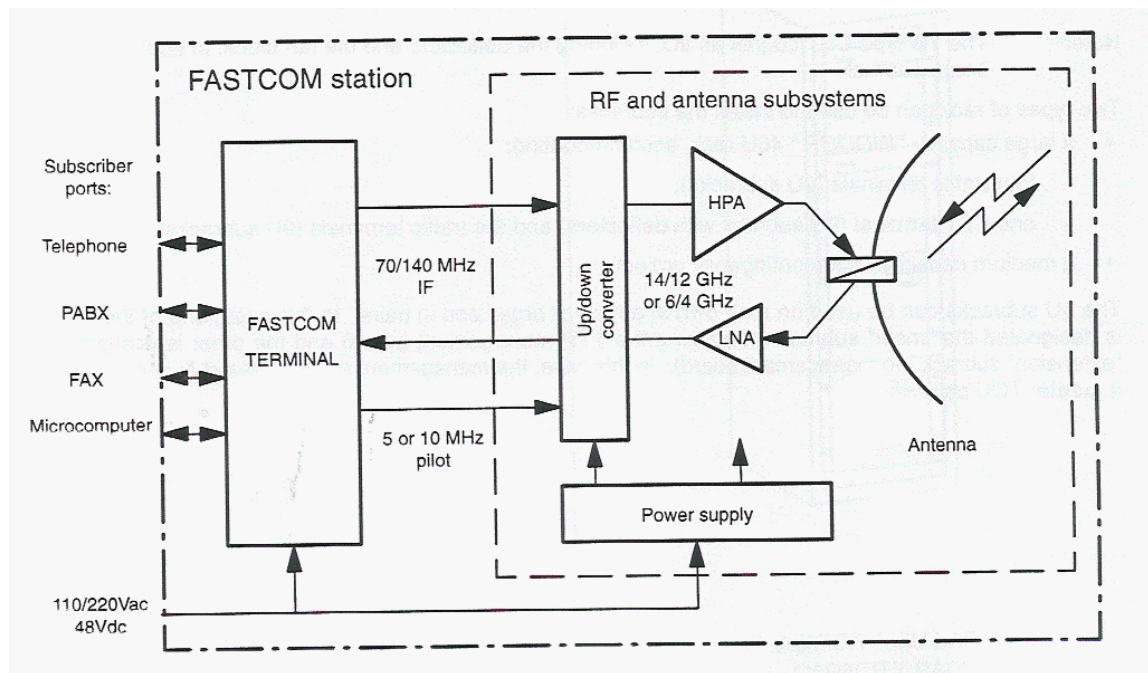


Figura 2.2 – Diagrama esquemático de uma estação [1].

Uma estação é dividida em 3 partes, **terminais digitais** (FASTCOM), **unidade de RF** e **sistema de antena**:

1. Os **terminais digitais** estão alojados em gabinetes, possuindo as interfaces 4 fios + E&M, 2 fios, e interface de dados V.24 com RS232 ou RS422 (síncrono ou assíncrono), podendo operar nos modos SCPC (*Single Channel per Carrier*) ou MCPC (*Multiple Channels per Carrier*). São partes componentes dos terminais digitais:

- as placas TIM (*Terrestrial Interface Module*) suportam fisicamente as interfaces, possuindo ainda como funções a possibilidade de cancelamento de eco (ITU-T Rec. G165), compressão de voz (ITU-T Rec. G711, G721, G729), processamento de FAX, multiplexação e empacotamento e processamento dos sinais de E&M. A Tabela 2.1 exhibe parâmetros da placa TIM.

Tabela 2.1 – Parâmetros da placa TIM (*Terrestrial Interface Module*) [1].

4 Wire Access TIM Input	
Input	
Frequency range	300-3400 Hz
Impedance	300, 600 or 1200 Ω balanced
Level	- 20 to 0 dBm
Nominal level	0 dBmo = 0 dBm (600 Ω)
Line Signaling	
E&M access. Pulse or change in state	E&M type 1, 2, 3, 4, 5

2 Wire Access TIM input (Telephone Set Interface)	
Base band	300-3400 Hz
Impedance	300, 600 or 1200 Ω balanced
Signaling	decadic or Q23
Supply	48 V (battery reversal on-off hooking)
Ringing	70 to 80 V AC / 50 Hz
Metering pulses (option)	12 or 16 kHz
Nominal level	0 dBmo = 0 dBm (600 Ω)

4 Wire / 2 Wire Access TIM Output	
Frequency range	300-3400 Hz
Impedance	300, 600, 1200 Ω balanced
Level	- 7 to + 2 dBm
Nominal level	0 dBmo = 0 dBm (600 Ω)

Speech encoder	
Software selectable 64 kbit/s PCM	ITU-T Rec. G711
32 kbit/s ADPCM	ITU-T Rec. G721
16 kbit/s LD-CELP	ITU-T Rec. G728
8 kbit/s CELP (option)	ITU-T Rec. G729
Echo canceller	
Specifications	ITU-T Rec. G 165
Echo path attenuation	≥ 6 dB

- Os modems utilizam modulação QPSK, compatíveis com SCPC-FDMA e MCPC-FDMA. Para a sinalização dos canais é utilizado TDM e TDMA. São capazes de realizar conversões de frequências, possuindo um DSP (*Digital Signal Processor*). Aqui são setados o FEC (*Forward Error Correction*) com algoritmo de Viterbi que pode ser 1/2, 3/4 ou 7/8. A Tabela 2.2 mostra alguns parâmetros possíveis do modem.

Tabela 2.2 – Parâmetros de modem [1].

FAX	FAX Group 3 V27ter (4800 bps), V29 (9600 bps), V17 (14400 bps),
Data interfaces	
Synchronous data	9.6 - 19.2 - 64 - 128 kbit/s
Asynchronous data	≤ 9.6 kbit/s - 19.2 kbit/s
MODEM	
Information rate	7.3 to 95.3 kbit/s (FEC 1/2) 11.1 to 133.5 kbit/s (FEC 3/4) 13 to 116.6 kbit/s (FEC 7/8)
Transmitted rate	16 to 192 kbit/s
FEC	convolutional 1/2, 3/4, 7/8 K=7
Decoding	soft quantified, Viterbi decoding
IF	140 ± 36 MHz, or 70 ± 18 MHz
Stability of transmitted carrier	± 10 Hz max / month
Frequency response	10 ⁻⁶ per year
IF input/output impedance	50 Ω
Tx frequency agility	1 Hz steps
Rx frequency agility	1 Hz steps
Interfaces	
Dry loops	4 tele-control outputs 4 tele-signaling inputs
Clock synchronization	
Internal :	10 ⁻⁶ per year
external (with E1 TIM board) :	5 or 10 MHz or 2048 kHz
Help for installation	DC Voltage for Antenna pointing and tracking available.

- O SCU (*Signalling & Control Unit*) é o responsável pela configuração das placas TIM e dos modems, bem como o acionamento da estação.
- O módulo de controle e monitoração, composto pelo MMC (*Monitoring & Management Center*) e NCS (*Network Control Center*). O MMC, através de um CST (*Central Signaling Terminal*), deve informar o estado dos equipamentos das estações e mudar parâmetros (tais como EIRP e frequência do canal). O NCS, através de um CST (*Central Signaling Terminal*) que conversa com SCU (*Signalling & Control Unit*), RTT (*Remote Traffic Terminal*) e NCC (*Network Control Center*), deve assegurar o gerenciamento e operação do sistema, através da configuração da rede (criação de estações, definição de frequências disponíveis, definição de canais de controle, tipo de tráfego em cada canal, alocação PAMA e DAMA). As Figuras 2.3 e 2.4 mostram os módulos MMC e NCS.

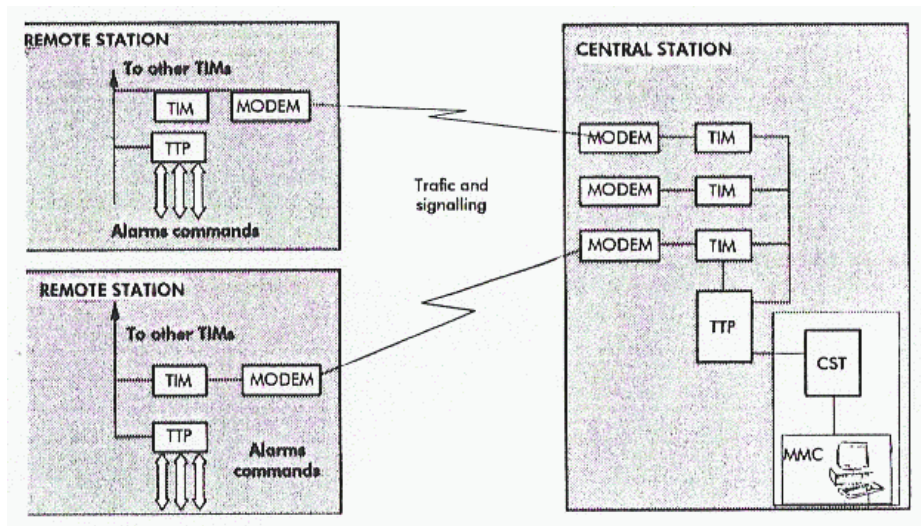


Figura 2.3 – MMC (*Monitoring & Management Center*) [1].

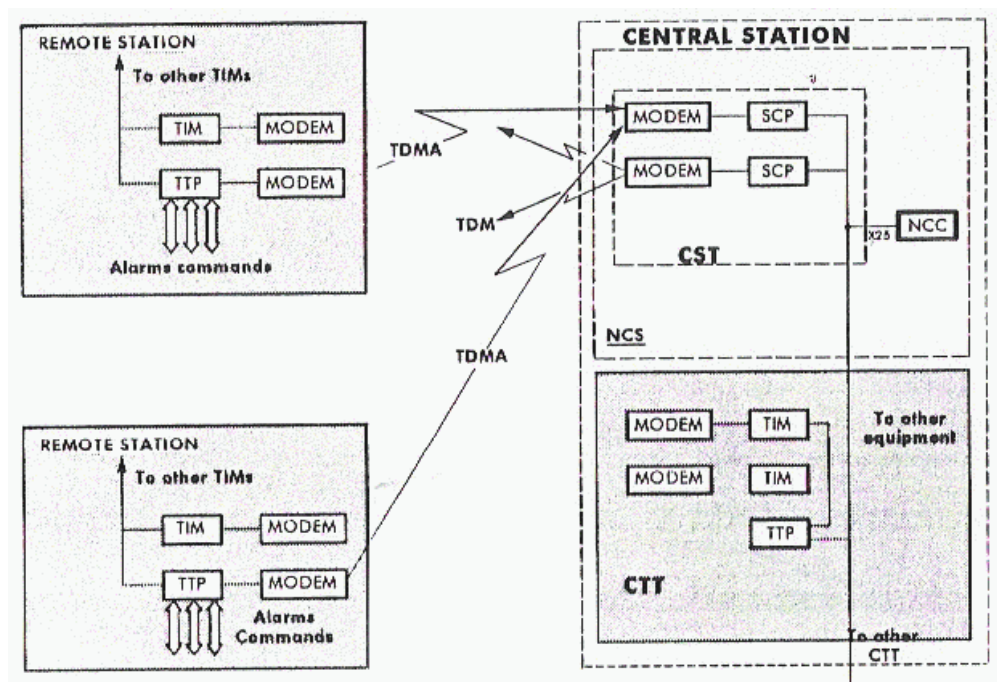


Figura 2.4 – NCS (*Network Control Center*) [1].

2. **unidades de RF** incluem *Up/Down converters* e amplificadores:

- Up-converter: responsável pela conversão das portadoras de FI (Frequência Intermediária) de 70 ou 140 MHz em uma portadora de RF (Rádio Frequência) dentro da faixa de 5,850 a 6,425 GHz.
- Down-converter: responsável pela conversão da portadora de RF recebida na faixa de 3,625 a 4,200 GHz, em uma portadora de FI de 70 ou 140 MHz.

- HPA: na verdade o HPA (*High Power Amplifier*) corresponde a um MPA (*Medium Power Amplifier*), válvula de 400W (56 dBm) de saída, com ganho mínimo de 75 dB, conforme a Tabela 2.3.

Tabela 2.3 - Características do MPA (*Medium Power Amplifier*) [1].

Output Power	
•TWT	56.0 dBm min. (400 watts)
•flange	55.4 dBm min. (350 watts min.)
Bandwidth	800 MHz (1225 MHz optional)
Gain	
•at rated power	75 dB min.
•small signal	78 dB min.

- LNA: o LNA (*Low Noise Amplifier*) é um amplificador de vários estágios de FET (*Field Effect Transistor*), cujas características estão expressas na Tabela 2.4.

Tabela 2.4 – Características do LNA (*Low Noise Amplifier*) [1].

- Bandwidth	: 3.625 - 4.200 GHz
- Gain adjustment range	: $22 \leq G_{nom} \leq 30$ dB
- Noise factor (for $22 \leq G \leq 30$ dB)	: ≤ 6.5 dB
- Variation in gain across the band	: ≤ 1 dBp-p
- Gain slope	: ≤ 0.02 dB/MHz
- Maximum gain variation between $25 \pm 10^\circ\text{C}$: ± 0.25 dB
- Input VSWR	: $\leq 1.3 : 1$
- Output VSWR	: $\leq 1.3 : 1$
- Interception point (for $22 \leq G \leq 30$ dB)	: $\geq +27$ dBm
- Output level with 1 db of compression for $21 < G < 29$ dBm	: > 18 dBm
- Input connector	: SMA female
- Output connector	: SMA female
- Power requirements	: + 12V / 165 mA - 12V / 25 mA

3. **antenas:** o diâmetro depende diretamente da posição dentro da arquitetura e consequentemente, da potência requerida para manutenção dos parâmetros de projeto. Existem antenas com 2,4 m, 3,7 m, 4,6 m, 6,2 m e 7,3 m.

O sistema foi dimensionado para utilizar 3 transponders (8A, 9A e 10A do satélite BRASILSAT B3) de 36MHz, *uplink* a 6 GHz e *downlink* a 4 GHz, com uma topologia mista de estrela e *mesh*.

Dentro do mesmo transponder, as estações dotadas com antenas de 4,6m são capazes de se comunicar através de um único *hop*, as estações dotadas com antenas de 2,4m e/ou em transponders diferentes, são capazes de se comunicar através de duplo *hop* através da *hub*, a fim de poupar a potência do transponder. Um *blocking rate* (probabilidade da linha DAMA estar ocupada) de 0,1% foi a especificação inicial do segmento satelital.

Os *sites* remotos são equipados com *transceivers* capazes de operar em apenas um par de RF (radiofrequência). Assim sendo, cada *site* remoto é capaz de se comunicar através de um único transponder utilizando uma frequência de *up-converter* e uma frequência de *down-converter*. As estações *hub* (6 ao todo) são compostas por *transceivers* capazes de abrigar 3 frequências de *up-converter* e 3 frequências de *down-converter*.

O gerenciamento de todo o sistema está localizado em Brasília, onde se encontra o NCC (*Network Control Centre*) e o CST (*Control & Signaling Terminal*).

As estações estão distribuídas no território nacional conforme mostra a Figura 2.5.

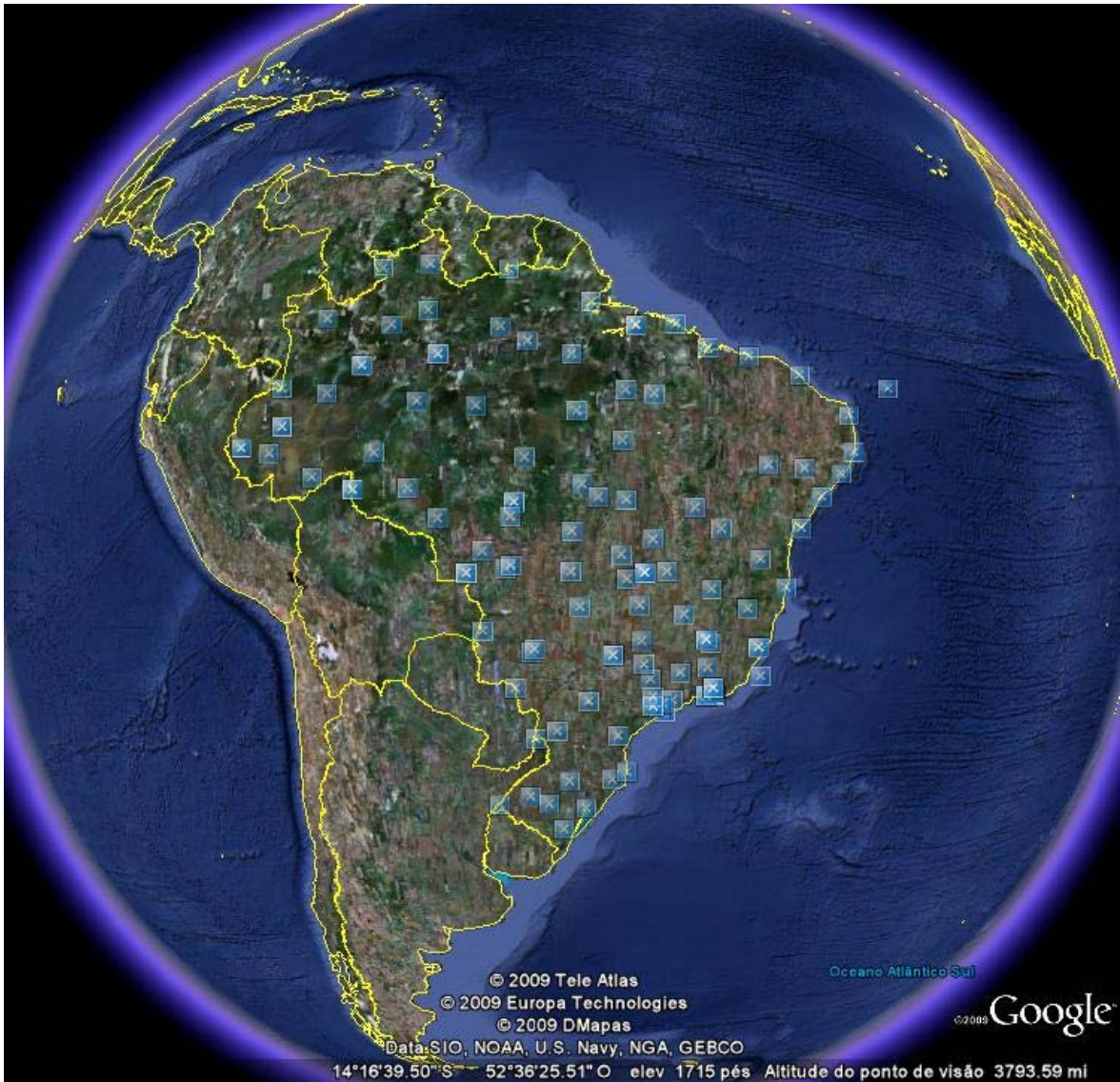


Figura 2.5 – Distribuição das estações terrenas.

2.1 – PARÂMETROS DE *LINK BUDGET*

Alguns parâmetros serão essenciais no desenvolvimento deste trabalho, principalmente os da Tabela 2.6, onde temos a correspondência E_b/N_0 versus BER.

Tabela 2.5 – Características gerais do sistema [1].

	FASTCOM
Modulation	QPSK
FEC	1/2 , 3/4 or 7/8
Voice encoding	16, 32 or 64 kbit/s
Data	<ul style="list-style-type: none"> • Up to 19.2 kbit/s async. • From 9.6 kbit/s up to 128 kbit/s sync.

Tabela 2.6 – Correspondência Eb/No versus BER de acordo com FEC [1].

	Composite Eb/No		
	FEC 1/2	FEC 3/4	FEC 7/8
Threshold	5.0 dB	3.5 dB	2.5 dB
BER = 1 x 10 ⁻³	4.8 dB	5.8 dB	6.3 dB
BER = 1 x 10 ⁻⁴	5.5 dB	6.6 dB	7.1 dB
BER = 1 x 10 ⁻⁵	6.3 dB	7.3 dB	7.8 dB
BER = 1 x 10 ⁻⁶	6.9 dB	8.2 dB	8.5 dB
BER = 1 x 10 ⁻⁷	7.7 dB	9.0 dB	9.2 dB

Estes valores são descritos para um canal em *loop-back*.

Tabela 2.7 - Características do transponder [1].

SATELLITE	BRASILSAT B3
Location	84° West
Tx Frequency Band	5850-6425 MHz
Rx Frequency Band	3625-4200 MHz
Polarisation	Linear
Orthogonal isolation	33 dB
Transponder Bandwidth	36 MHz
Coverage	Brazil
EIRP (EOC)	36.7 dBW
G/T (EOC)	- 2.5 dB/K
SFD (EOC)	From - 78 dBW/m ² to - 92 dBW/m ² adjustable in 2dB steps
Stability	0.1°(E/W); 0.1°(N/S)
Spacing	3.0°
Up-link advantage	From 0 dB (EOC) to 6 dB Please refer to fig 10
Down-link advantage	From 0 dB (EOC) to 3 dB Please refer to fig. 11
EMBRATEL recommendation for multi-user operation:	
Nominal SFD (EOC)	- 86 dBW/m ²
Nominal Input Back-off	4.0 dB
Nominal Output Back-off	3.4 dB
Nominal Intermodulation density	- 24 dBW/4kHz
Margin for external interferences	3.6 dB

Requisitos de qualidade:

- BER melhor que 10⁻⁷ durante mais que 98% do tempo (no pior mês do ano).
- BER melhor que 10⁻³ durante mais que 99,8% do tempo (no pior mês do ano).

3 – FUNDAMENTAÇÃO TEÓRICA

3.1 – TOPOLOGIA DE REDE VSAT

As redes VSAT são constituídas pelo segmento satelital, pelos terminais VSAT e pelas estações *hub*.

O satélite é uma estação que recebe os sinais e os envia novamente para a Terra. As VSATs são terminais constituídos por antenas de pequeno diâmetro. As estações *hub* são estações principais (com antenas maiores) que podem se comunicar com todas as VSATs, funcionando como ponto de interconexão [9].

As redes VSAT, conforme mostra a Figura 3.1, podem ser do tipo estrela, onde as VSATs se comunicam exclusivamente com a estação *hub*, ou do tipo malha (*mesh*), onde as VSATs podem comunicar-se entre si. É importante observar, que no caso da topologia estrela, para que duas VSATs comuniquem-se entre si, há necessidade do duplo salto, uma vez que a *hub* fará toda a intermediação da troca de mensagens.

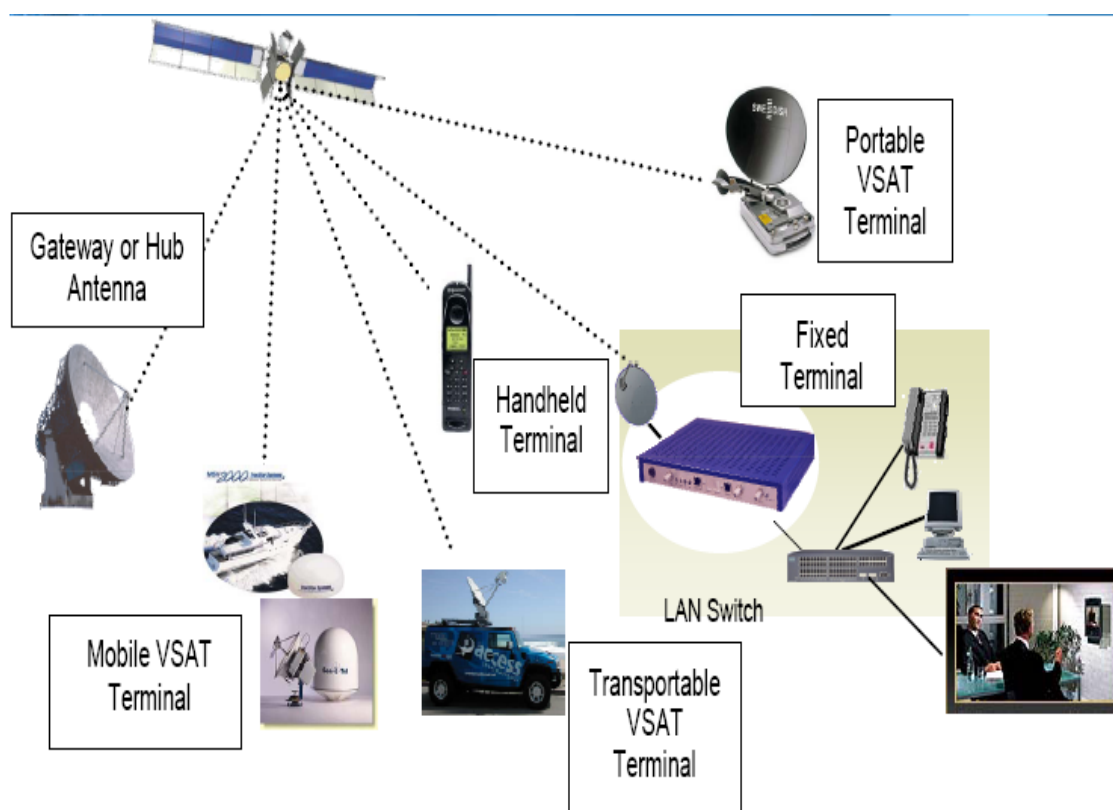


Figura 3.1 - Topologia de rede VSAT [4].

A estação VSAT propriamente dita, é composta de 2 unidades físicas distintas, a ODU (*OutDoor Unit*) e a IDU (*InDoor Unit*), conforme mostrado na Figura 3.2. Na ODU ficam a antena e o alimentador, podendo ficar também a parte de amplificação. Na IDU fica o modem, podendo ficar também a parte de RF.



Figura 3.2 – Terminal VSAT [4].

3.2 – ALOCAÇÃO (DESIGNAÇÃO) DE CANAIS

Para que uma estação se comunique, é necessária a alocação de um canal de RF, que pode ser permanente ou temporária. O método de alocação PAMA (*Permanent Assignment Multiple Access*) permite uma alocação permanente. No DAMA (*Demand Assignment Multiple Access*) há uma alocação dinâmica, onde o *pool* de canais administrados são alocados na medida da necessidade.

3.3 – MÉTODOS DE ACESSO [6], [13], [19] e [20]

Nas comunicações via satélite, em especial, as portadoras podem ser utilizadas no modo SCPC (*Single Channel per Carrier*) ou MCPC (*Multiple Channels per Carrier*).

O SCPC refere-se ao uso exclusivo de uma portadora para cada canal. Ou seja, a largura de banda de uma portadora, é utilizada para uma única fonte. Isto indica que não existe multiplexação em subportadoras. Como vantagens imediatas temos a simplicidade, robustez e baixo custo de implementação. As desvantagens são as limitações de um uso eficiente no caso de transmissões em *burst*. O SCPC é utilizado em aplicações que se valem de alocação tipo DAMA.

No MCPC vários canais são combinados através de técnicas de multiplexação, em um único *bitstream*, antes de serem modulados em uma portadora. O MCPC é utilizado em aplicações que se valem de alocação tipo PAMA.

3.3.1 – Métodos de acesso por divisão de canal

Os métodos de múltiplo acesso utilizados nas comunicações, visam otimizar o uso da canalização utilizada, de forma a reduzir o custo total do sistema de comunicação, devendo ser transparentes ao usuário final. A utilização desses métodos permite que um único canal seja compartilhado por vários usuários.

TDMA (*Time Division Multiple Access*): acesso múltiplo por divisão de tempo, onde cada usuário recebe um *time slot* (intervalo de tempo) específico, permitindo o compartilhamento da frequência disponível para o canal. O TDMA compartilha o canal de transmissão no tempo, eliminando colisões. Contudo, a velocidade de transmissão é sempre limitada, mesmo que haja um único usuário ativo, pois cada um deve esperar sua vez na seqüência de transmissão.

FDMA (*Frequency Division Multiple Access*): acesso múltiplo por divisão de frequência, onde cada usuário recebe um *frequency slot* (intervalo de frequência) específico. O FDMA compartilha o canal de transmissão através da divisão do canal em frequência eliminando colisões contudo, a velocidade de transmissão é sempre limitada mesmo que haja um único usuário ativo.

Os métodos acima descritos consistem basicamente na partição do canal, vinculando cada usuário a um subcanal. Pode-se perceber que essa metodologia não permite uma utilização ótima da canalização, uma vez que ao vincularmos um usuário a um *time slot* ou *frequency slot*, estaremos provavelmente desperdiçando os períodos de silêncio do usuário, onde não há necessidade da reserva do subcanal.

CDMA (*Code Division Multiple Access*): acesso múltiplo por divisão de código, onde cada usuário recebe um código. Por vezes o CDMA é chamado como SSMA (*Spread Spectrum Multiple Access*), uma vez que ele utiliza o espalhamento espectral como meio de acesso. As bandas do CDMA são divididas em canais de RF, onde cada canal consiste de um par de frequências (TX e RX) com 1,25 MHz de banda cada. O acesso múltiplo de canais que compartilham uma mesma banda de frequências, é feito pela utilização de códigos diferentes, o sinal contendo a informação é codificado utilizando-se uma chave de código que provoca o seu espalhamento espectral em uma banda transformando-o aparentemente em ruído. Quanto mais usuários utilizam o canal, maior o ruído, aumentando a interferência para os canais que utilizam a mesma banda até um limiar quando não é mais possível decodificar os canais. Esta interferência também é tanto maior, quanto maior for a potência individual de cada canal transmitido naquela banda. Um dos fatores que contribui para a grande capacidade alcançada por sistemas CDMA é a possibilidade de utilização de reuso de 1, ou seja, a mesma frequência de portadora é reutilizada em todas as células de sistemas de telefonia móvel.

3.3.1.1 – Capacidade dos métodos de acesso por divisão de canal

Considerando um canal ideal AWGN (*Additive White Gaussian Noise*) de largura de banda total W , com um único usuário com potência média P onde $\frac{1}{2} * N_0$ é a densidade espectral de potência, temos que sua capacidade C de transmissão de dados é dada por:

$$C = W * \log_2 \left(1 + \frac{P}{W * N_0} \right) \quad (3.1)$$

No FDMA, considerando K usuários onde a potência média por usuário é dada por P , sabemos que a capacidade pelo número de usuários K é dada por:

$$C_K = \frac{W}{K} * \log_2 \left[1 + \frac{P}{(W/K) * N_0} \right] \quad (3.2)$$

No TDMA, a capacidade pelo número de usuários K é dada por:

$$C_K = \left(\frac{1}{K} \right) * W * \log_2 \left[1 + \frac{K * P}{W * N_0} \right] \quad (3.3)$$

Isso corresponde numericamente ao resultado do FDMA, o importante a se ressaltar, no entanto, é que dificilmente o transmissor suporta uma potência elevada $K \cdot P$ quando o número de usuários K é muito elevado!

No CDMA, a capacidade do sistema depende do nível de cooperação entre os K usuários. No pior caso, temos usuários não cooperativos, que desconhecem as formas de onda de espalhamento dos demais usuários. Nesse caso a capacidade pelo número de usuários K é dada por:

$$C_K = W * \log_2 \left[1 + \frac{P}{W * N_o + (K - 1) * P} \right] \quad (3.4)$$

3.3.2 - Métodos de acesso aleatório

Um método de acesso aleatório permite que um usuário transmita sempre à taxa total do canal. Quando há uma colisão, cada usuário retransmite repetidamente sua informação até ser aceito. Quando a informação do usuário sofre colisão, ela não é retransmitida de imediato, aguardando um tempo aleatório antes de tentar nova transmissão.

Slotted ALOHA: diferentemente da partição do canal, esse método de acesso permite que um único usuário transmita informação à taxa total do canal. É parcialmente descentralizado por permitir que cada usuário detecte colisões e decida quando retransmitir, no entanto, requer que haja uma sincronização entre todos os usuários.

A eficiência desse tipo de protocolo é baixa (37%), pois quando temos vários usuários ativos, há um grande número de colisões (26%), além um número grande de intervalos de silêncio (37%), fruto da política probabilística de transmissão onde os usuários evitarão transmitir.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection): acesso múltiplo por detecção de portadora, onde o usuário deve detectar a portadora (escutar o canal antes de transmitir, pois se o canal já estiver em utilização deve aguardar um período de tempo aleatório) e detectar colisão (se um usuário começa a utilizar o canal ao mesmo tempo, deve haver a interrupção da transmissão).

3.3.2.1 – Ethernet

A Ethernet é um padrão de redes de computadores, atualmente também designado por IEEE 802.3 (*Institute of Electrical and Electronics Engineers*, entidade que designou um comitê no mês 2 de 1980 para padronizar a operação de uma rede CSMA/CD). Essa tecnologia de rede superou outras tecnologias tais como as Token Ring, FDDI e ATM.

3.3.3 – Métodos de revezamento

Método de Polling: esse método requer que um usuário seja designado como mestre. O usuário mestre escolhe cada um dos nós de maneira logicamente circular, o que elimina as colisões mas, em caso de falha do mestre o canal ficará inoperante.

Método de Passagem de Permissão: nesse caso não há mestre, mas existe uma permissão (*token*) passada sequencialmente entre os usuários

3.4 – TRANSMISSÃO EM BANDA BASE [24]

A transmissão de um sinal em banda base, consiste em enviar o sinal de forma digital através da linha, ou seja, enviar os bits conforme a necessidade, de acordo com um padrão digital, como por exemplo a codificação manchester, utilizada em redes locais. A transmissão digital pode ser unipolarizada ou bipolarizada, como mostra a Figura 3.3, onde os itens a) e b) são unipolarizados e o item c) é bipolarizado.

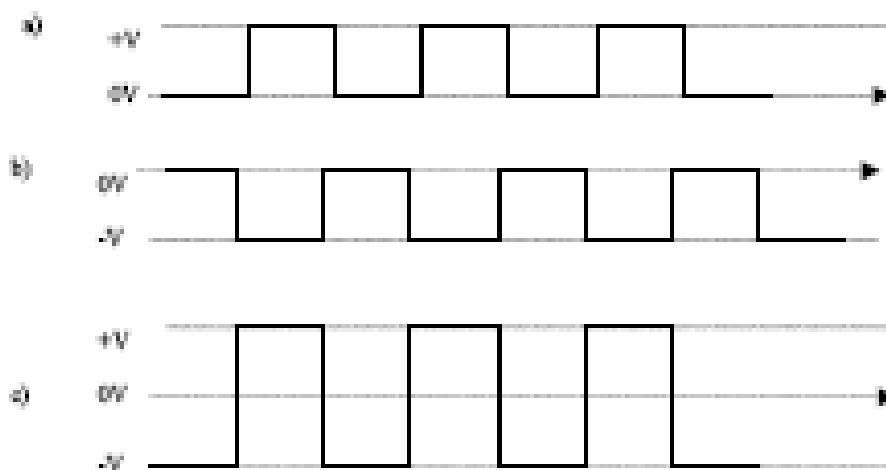


Figura 3.3 – Transmissão unipolar em a) e b) e bipolar em c).

Uma característica da transmissão digital é que ocupa uma alta largura de banda, conforme a análise de sinais e transformada de Fourier. Assim, a transmissão banda base é muito usada quando tem-se largura de banda disponível, tipo com LPCD (Linha Privativa de Comunicação de Dados).

3.4.1 - Problema da transmissão banda base em linhas telefônicas

Como visto, a transmissão em banda base necessita uma alta largura de banda para ser transmitida, porém, as linhas telefônicas possuem uma limitação de 4 KHz na sua largura de banda. Então, como fazer para transmitir os dados através destas linhas?

A solução para este problema é utilizar um novo conceito de transmissão de dados, a transmissão com modulação, cujo objetivo é transformar o sinal digital a ser enviado em um sinal analógico dentro da faixa de 4 KHz, transmitir este sinal através da linha telefônica e remontar o sinal no seu destino. O equipamento que faz esta tarefa é conhecido como modem, que é a contração da palavra modulador e demodulador. O sinal é digital entre o computador e o modem, e analógico na linha telefônica.

3.5 – MODEM

A informação digital dos computadores, precisa ser convertida adequadamente para o tráfego pela rede telefônica pública, construída originalmente para o tráfego de voz e não para os sinais digitais dos computadores. O enlace digital entre o computador e o modem é transformado por este último em um enlace analógico, que chega até uma central telefônica. O enlace entre centrais é feito de forma digital, exceto em centrais muito antigas. Grandes instituições alugam linhas digitais, podendo comunicar-se a grandes velocidades.

Inicialmente os padrões foram adotados como recomendações da CCITT (*Comité Consultatif International de Telegraphie et Telephonie*), posteriormente renomeada ITU. Diversos padrões para comunicação de dados sobre rede telefônica foram desenvolvidos, esses padrões estão nomeados com siglas que começam com a letra V e, por isso são conhecidos como padrões e recomendações da série V [10].

Atualmente, a palavra modem é também utilizada para designar dispositivos usados em transmissão exclusivamente digital, como por exemplo, aqueles que recebem as

informações digitais de um computador e os adequam para uma linha telefônica digital como a ISDN (*Integrated Services Digital Network*).

Considerando um canal de voz telefônico com banda de 3.000 Hz e uma relação sinal ruído de 35 dB chegamos a uma taxa de transmissão teórica de 35 Kbps para os modems analógicos convencionais. O ruído tem várias causas, dentre elas a que mais interessa é o ruído proveniente da quantização, que surge quando se digitaliza o sinal analógico para entrar na rede pública telefônica. Sempre comete-se um erro ao transformar um sinal analógico em um sinal digital e esse erro tem papel semelhante ao ruído, sendo por isso chamado de ruído de quantização. O processo inverso, a transformação o sinal digital em analógico não introduz ruído.

Normalmente os provedores Internet (ISP – *Internet Service Provider*) conectam-se à rede telefônica pública através de linhas digitais, onde não se faz a quantização. Assim, no caminho ISP para o *end user* o ruído é bem menor, podendo chegar a 56 Kbps. Já no caminho inverso, antes do sinal analógico do usuário entrar na rede pública, é feita uma conversão de analógico para digital, aumentando a quantidade de ruído.

Atualmente existem técnicas mais modernas que permitem o aumento das taxas de transmissão (xDSL – *Digital Subscriber Line*).

3.5.1 - Diferença entre baud e bps

A velocidade de modulação (medida em bauds) é o número de possíveis variações da onda portadora por segundo, ou seja, o número de vezes que se imprime a informação digital na onda portadora por segundo (a cada possível variação na onda portadora está se imprimindo uma informação digital).

A velocidade do sinal (medida em bps), indica o número de bits efetivamente transmitidos pelo canal de comunicação por segundo.

Não necessariamente a portadora carrega um bit por ciclo de onda. Na transmissão, pode-se considerar que estão sendo transmitidos símbolos, e os símbolos podem significar um ou mais bits. Uma analogia é imaginar os símbolos como caixas, e dentro dessas caixas podem ser acomodados um ou mais bits.

Em uma modulação monobit, a portadora carrega um bit a cada baud. Desta forma, temos que na modulação monobit, o número de bauds é sempre igual ao número de bps. Em uma modulação multinível, a cada baud (ou variação da onda portadora), pode-se imprimir mais de um bit na onda portadora.

3.5.2 - Modulação multinível

O objetivo da modulação multinível é transmitir a informação com maior velocidade, e para conseguir isto é impresso mais de um bit por variação da onda portadora. Por exemplo, na modulação ASK (*Amplitude Shift Keying*) dibit, em vez de criar-se uma tabela para os bits "1" e "0", cria-se uma tabela de amplitudes diferenciadas para os bits "00", "01", "10", e "11". Assim, cada variação na onda portadora carrega consigo mais de um bit.

A figura a seguir mostra um exemplo de ASK dibit, onde a seguinte tabela foi estabelecida como padrão entre transmissor e receptor:

- "00": 2V
- "01": 4V
- "10": 6V
- "11": 8V

Além disto, o padrão convencionou que a portadora tivesse 2400 Hz e fosse variada a cada ciclo (2400 bauds). A Figura 3.4 ilustra a forma de onda resultante para transmitir a seqüência de bits "100011010011", onde $F_p = 2400$ Hz, $V_s = 4800$ bps e $V_m = 2400$ bauds.

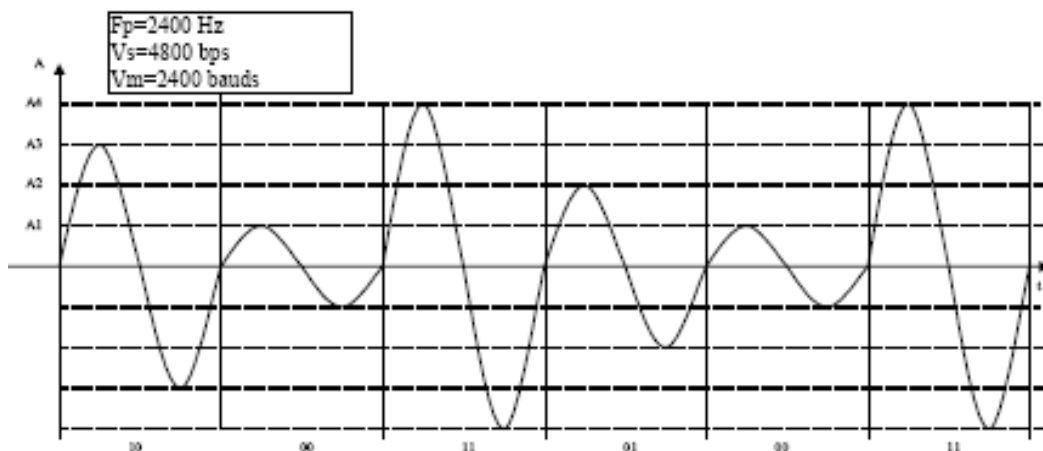


Figura 3.4 - Exemplo dibit.

Para aumentar mais ainda a velocidade do sinal, pode-se utilizar mais de dois bits por baud.

3.6 – MODULAÇÃO

A modulação é o processo pelo qual alguma característica da forma de onda da portadora (*carrier*) é variada (modulada), de acordo com a variação de um outro sinal de interesse (modulador / modulante). Os atributos a serem modificados na portadora podem ser a amplitude, polarização, frequência e/ou fase.

O processo de modulação surge da necessidade em se adequar o sinal de interesse ao canal disponível, da melhor maneira possível. Desse modo cabe-nos observar as seguintes necessidades de adequação:

- transmissões a longas distâncias necessitam de ondas eletromagnéticas, fios, fibras ópticas ou guias de onda;
- o tamanho de antenas eficientes é da ordem de $0,5 \lambda$ e considerando-se um sinal de 3 KHz, observa-se que ele exige uma antena de 50 km;
- circuitos eletrônicos, tem custo e complexidade de hardware menores se a largura de faixa fracionária (largura de faixa do sinal, dividida pela sua frequência central) for da ordem de 0,01 a 0,1;
- necessidade de redução da influência dos ruídos através de técnicas de modulação;
- alocação de faixas de frequência através do uso de portadoras diferentes.

Pelo exposto observamos que temos de atender questões de meio de propagação, hardware, ruído e distribuição de espectro. Além disso, um processo de modulação deve modificar o formato da informação elétrica com o objetivo de transmiti-la com:

- menor potência possível;
- a menor distorção possível;
- facilidade de recuperação da informação original;
- o menor custo possível.

Os processos de modulação podem geralmente ser classificados em categorias básicas:

- de acordo com a portadora: modulação de onda contínua (cw de *continuous wave*) e modulação de pulsos.
- de acordo com o sinal modulador (modulante): modulação analógica quando o sinal modulador é contínuo, e modulação digital quando o sinal modulador é pulsado.

Os vários tipos de modulação podem ser combinados, e além disso, um mesmo sistema pode utilizar vários deles.

É importante observar que, de modo geral, referimo-nos a sinais analógicos como sendo do tipo AM, FM ou PM; e referimo-nos a sinais digitais como sendo do tipo FSK, ASK, PSK e QAM. Contudo essa forma de referência leva em conta apenas o sinal modulante, sem levar em consideração a portadora, conforme mostrado nas Figuras 3.5 e 3.6.

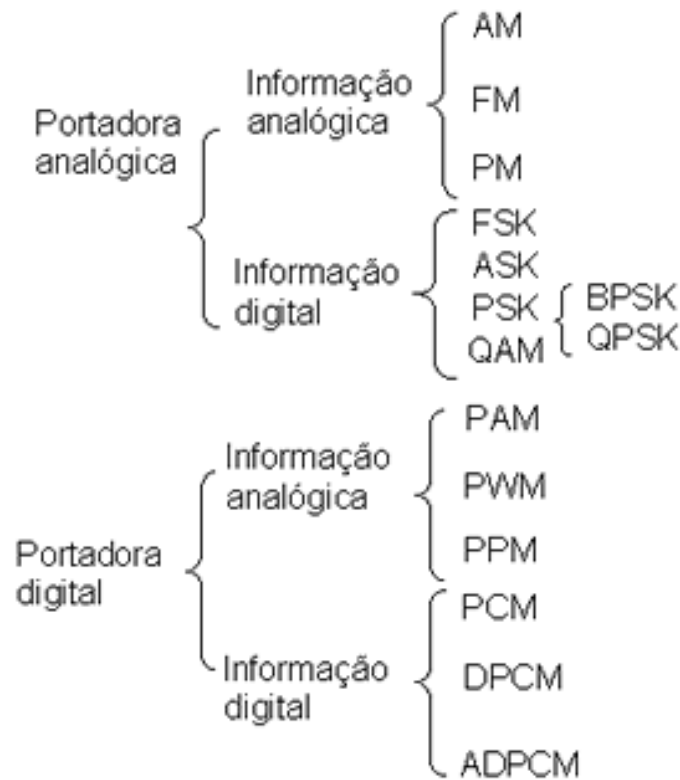


Figura 3.5 – Tipos de modulação.

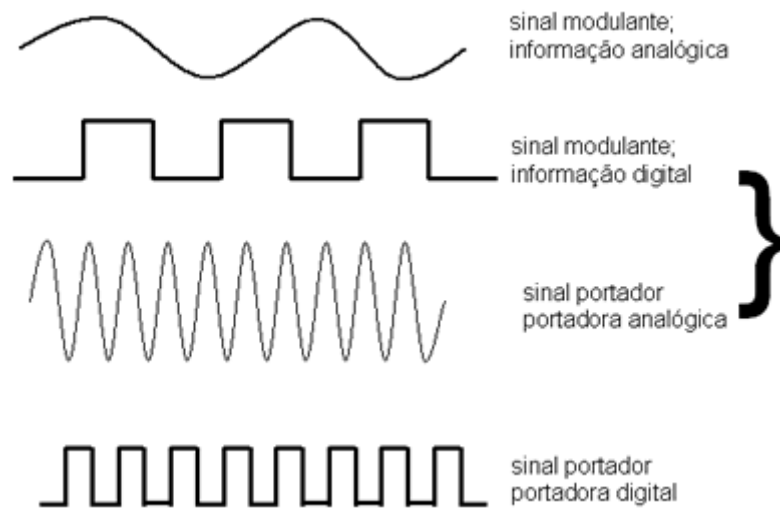


Figura 3.6 – Processos de modulação digital.

3.6.1 – Sinal analógico

O sinal analógico (ou contínuo), é aquele que possui uma resposta contínua, dada por uma variável contínua. Ou seja, a função contínua que descreve o comportamento do sistema, permite obter qualquer valor em amplitude (eixo das ordenadas) em função de qualquer valor da variável independente (eixo das abscissas), como exemplificado pela Figura 3.7.

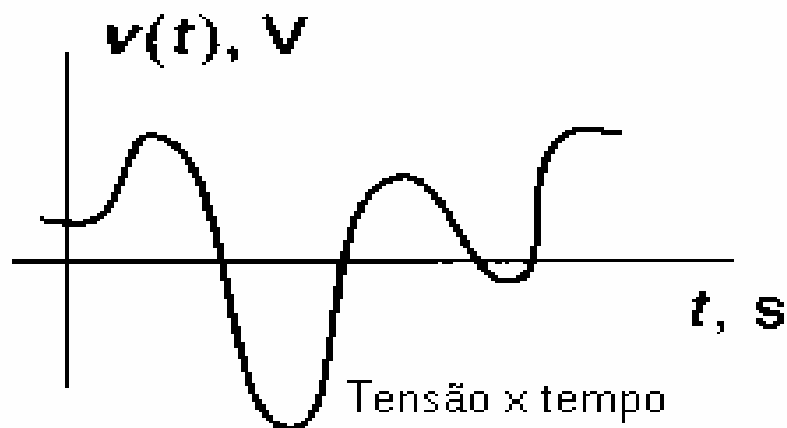


Figura 3.7 – Sinal analógico.

3.6.1.1 – Sinal de domínio discreto (ou sinal amostrado)

O sinal de domínio discreto é aquele que possui resposta contínua, dada por uma variável discreta (amostrada) conforme Figura 3.8. Ou seja, a função contínua que descreve o comportamento do sistema, permite obter qualquer valor em amplitude (eixo das ordenadas) em função de uma variável independente que é discreta (eixo das abscissas). A variável independente pode ser definida a intervalos regulares e ser representável por uma seqüência numérica. Observe que a amostragem é uma etapa do processo de conversão do sinal analógico em sinal digital.

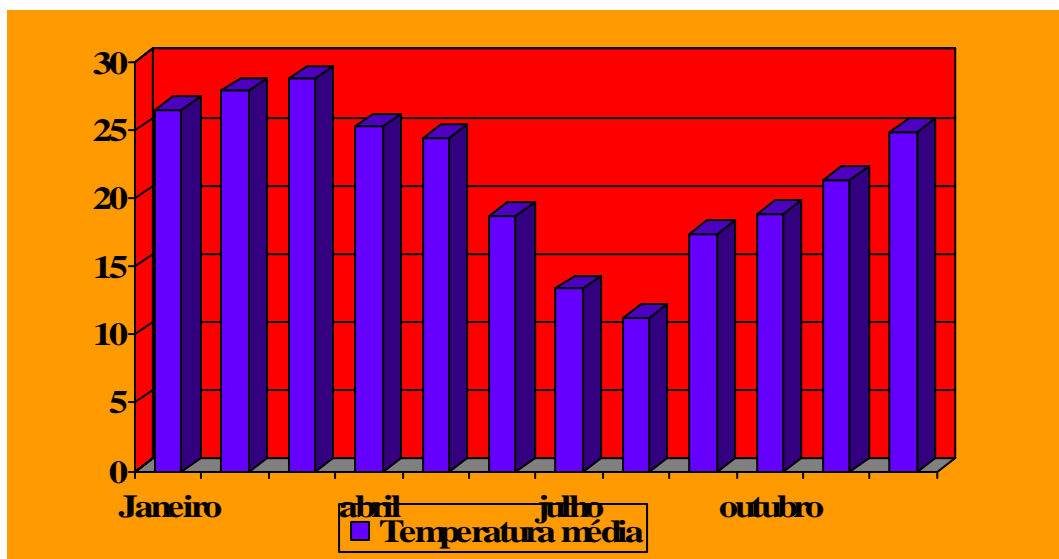


Figura 3.8 – Sinal de domínio discreto.

3.6.1.2 - Sinal discreto de variável independente contínua (ou sinal quantizado)

O sinal discreto de variável independente contínua, é aquele que possui resposta quantizada, dada por uma variável contínua. Ou seja, a Figura 3.9 exibe uma função discreta de variável contínua que descreve o comportamento do sistema, permite obter determinados valores em amplitude (eixo das ordenadas) em função de variável independente contínua (eixo das abscissas). Observe que a quantização também é uma etapa do processo de conversão do sinal analógico a sinal digital.

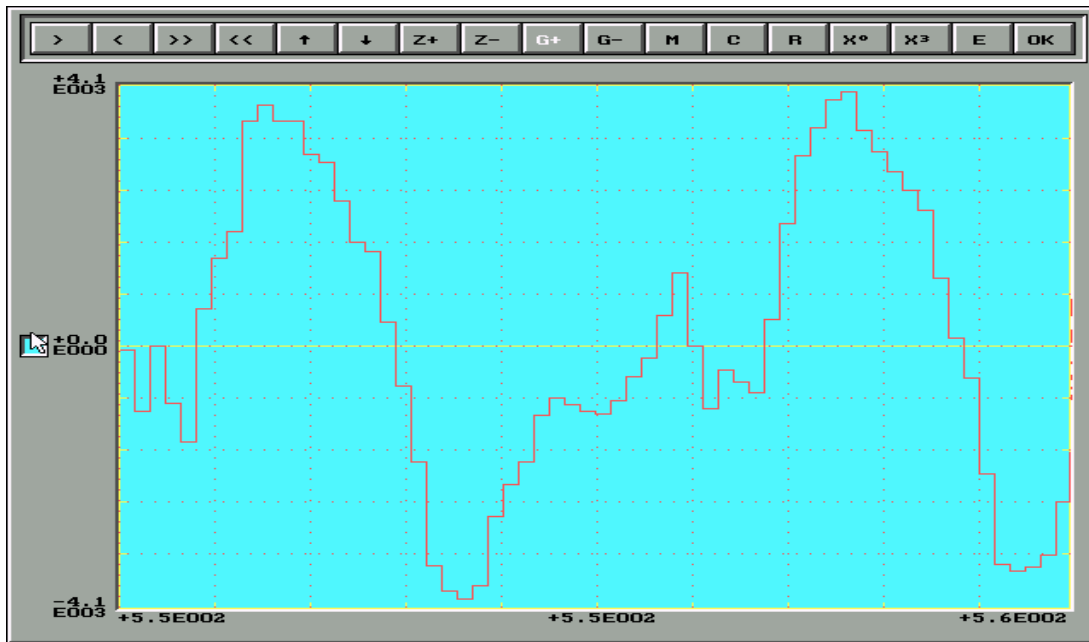


Figura 3.9 – Sinal discreto de variável independente contínua.

3.6.2 – Sinal digital

O sinal digital é aquele que possui resposta discreta (quantizada), dada por variável independente discreta (amostrada), conforme a Figura 3.10. Ou seja, a função discreta de variável independente discreta (amostrada) que descreve o comportamento do sistema, permite obter determinados valores em amplitude (valores quantizados no eixo das ordenadas) em função de variável independente discreta (valores amostrados no eixo das abscissas). As amplitudes são codificadas numericamente, usualmente de forma binária.

O processo de digitalização (também chamado de conversão analógico-digital ou A/D) requer como visto a discretização da variável independente (amostragem) e a discretização em amplitude (quantização) além da codificação numérica.

No mundo digital, cada símbolo pode ser representado por uma seqüência de bits que quantificam esse símbolo.

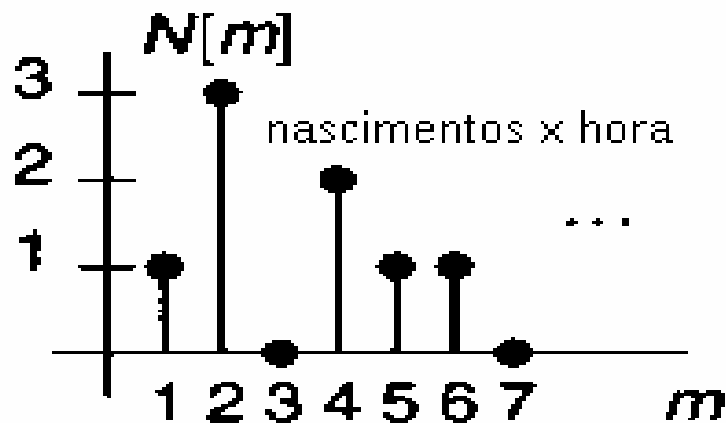


Figura 3.10 – Sinal digital.

Logo, o processamento digital de sinais passa pelo seguinte processo ilustrado pela Figura 3.11:

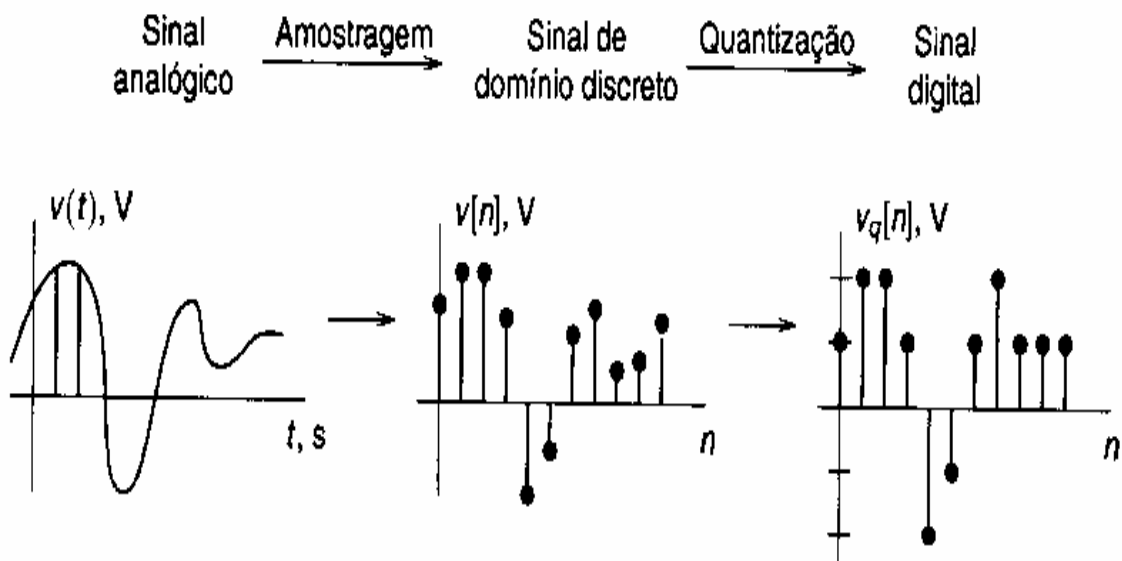


Figura 3.11 – Conversão analógico/digital.

Algumas vantagens do PDS (Processamento Digital de Sinais): maior imunidade a ruído, imunidade a variação de temperatura, precisão.

Desvantagens: custo alto e tempo de processamento.

3.6.3 – Conversão analógico/digital

A conversão A/D (digitalização) é o processo que converte o sinal analógico em sinal digital. Para isso, o sinal analógico deve ser discretizado ao longo do tempo através de amostragem, e discretizado em amplitude através da quantização.

Teorema da Amostragem:

- O Teorema de Nyquist descreve como deve ser o processo de discretização do sinal analógico ao longo do tempo (amostragem), para que seja possível digitalizar um sinal. Para que um sinal seja digitalizado e seja recuperado sem distorções, a taxa de amostragem deve ser no mínimo o dobro da máxima frequência do sinal analógico de interesse.

Se considerarmos que a voz tenha uma frequência máxima de 4 KHz, segundo Nyquist devemos amostrar esse sinal no mínimo 8.000 vezes por segundo ou seja, a taxa de amostragem deve ser de 8 KHz. Se formos considerar também o processo de discretização dos níveis do sinal (quantização do sinal) em 256 níveis possíveis para representar sua quantização através de seqüência binária composta por bits 0s e 1s, devo ter 8 bits ($2^8 = 256$ estados possíveis). Assim eu preciso de um canal capaz de suportar um tráfego de 8.000 amostras onde cada amostra é definida por 8 bits, ou seja, o canal deve suportar 64 Kbps (8.000 amostras por segundo x 8 bits por amostra).

3.6.4 – Conversão digital/analógico

A conversão D/A é o processo que converte o sinal digital em sinal analógico, a partir das discretizações existentes. Ou seja, a partir das amostras e quantizações existentes.

3.6.5 – Modulação com portadora analógica e sinal modulante digital (modulação digital)

A modulação digital apresenta vantagens sobre a modulação analógica tais como maior imunidade a ruído, facilidade de multiplexação, possibilidade de acomodar técnicas de detecção e correção de erros, possibilidade de implementar encriptação.

A performance de um tipo de modulação pode ser medida em termos de sua eficiência de potência e eficiência da largura de banda.

A eficiência de potência descreve a habilidade da técnica de modulação em preservar a fidelidade da mensagem digital com baixos níveis de potência. A eficiência de potência de uma modulação digital descreve a habilidade em se preservar a fidelidade (probabilidade aceitável de erro de bit). É expressa pela razão E_b/N_0 onde E_b é a energia por bit do sinal e N_0 é a densidade espectral de potência do ruído.

A eficiência de largura de banda descreve a habilidade da técnica de modulação em se manter dentro de uma determinada faixa de frequência, sendo uma razão entre a transferência de dados em bps e a ocupação da largura de banda alocada pelo sinal modulado. Temos então a razão R/B (em bps/Hz).

Shannon demonstrou através de seu teorema, o limite superior (teórica máxima) para a eficiência de largura de banda para uma probabilidade de erro arbitrariamente pequena, onde a máxima eficiência é limitada pelo ruído do canal sendo dada pela razão $C/B = \log_2 (1 + S/N)$ onde C é a capacidade do canal em bps, B é a largura de banda em Hz e S/N é a relação sinal/ruído (que nesse caso não é dada em dB).

3.6.5.1 – PSK (*Phase Shift Keying*)

A modulação por deslocamento de fase é a técnica de modulação que altera a fase da portadora em função do sinal digital a ser transmitido. Para este processo são usados pulsos bipolares.

De modo geral, em sistemas via-satélite são utilizadas modulações PSK, uma vez que modulações que envolvem a amplitude da portadora como a ASK, são em princípio inadequadas, já que o canal via-satélite é extremamente não linear e sua atenuação é variável com o tempo. A modulação FSK tem sido pouco utilizada pela sua baixa eficiência espectral.

Os diversos tipos de modulação podem ser comparados, em termos de sua robustez ao ruído e interferências, através da relação entre a taxa de bits errados BER (*Bit Error Rate*) e a correspondente razão energia de bit/densidade espectral de ruído (E_b/N_0).

Quando ocorrer uma transição de nível lógico do sinal digital a ser transmitido (sinal modulante), haverá uma mudança de 180 graus na fase da onda portadora com relação ao ângulo anterior. A transição observada pode ser tanto de nível lógico "0" para "1" como de nível lógico "1" para "0".

Para este tipo de modulação, conforme a Figura 3.12, deve se usar a detecção síncrona, já que esta tem como base o conhecimento preciso a respeito da fase da onda portadora recebida, bem como da sua frequência. Esta técnica de modulação, devido ao fato mencionado, envolve circuitos de recepção (demodulação) mais sofisticados; em compensação oferece melhor desempenho que as técnicas ASK e FSK. Na detecção síncrona, no processo da demodulação, a onda modulada recebida pode ser detectada fazendo o produto entre o sinal recebido e um sinal cosenoidal localmente gerado. O oscilador local, gerador da onda cosenoidal utilizada na demodulação, deve estar em sincronismo com a onda portadora tanto em fase como em frequência.

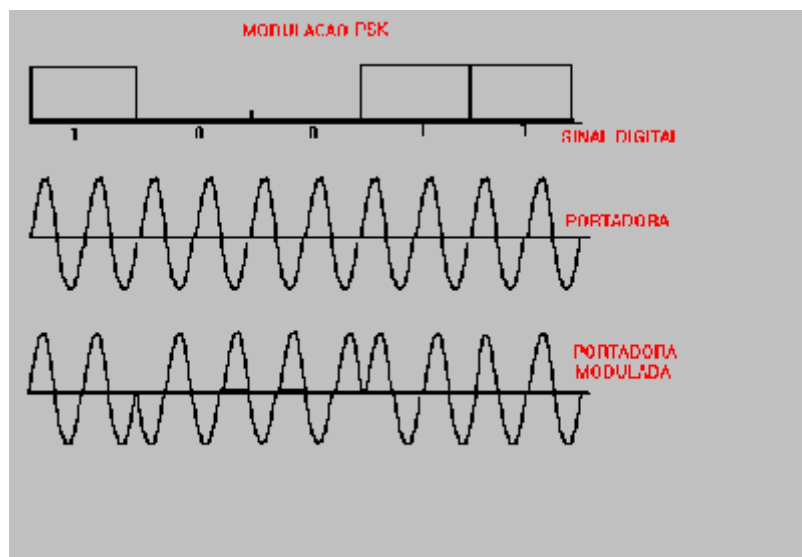


Figura 3.12 – Modulação PSK.

As técnicas de modulação digital são utilizadas em aplicações que necessitem uma maior eficiência de ocupação da banda. Para efeito de estudo pode-se classificar as modulações em técnicas monobit e técnicas multinível. As técnicas monobit utilizam apenas um símbolo por bit e as técnicas multinível utilizam um símbolo para um conjunto de bits.

É importante salientar que no caso monobit, designado por BPSK, que possui apenas dois símbolos, não existe a necessidade do eixo de quadratura. No caso do QPSK, temos quatro

símbolos e a cada símbolo temos associado um conjunto de dois bits e finalizando temos um sistema 8PSK, com 8 símbolos e a cada símbolo temos associado três bits. Observe que se utilizássemos mais bits por símbolo, haveria uma aproximação entre eles que dificultaria sua detecção e conseqüentemente aumentaria a probabilidade de erro, vide Figura 3.13. Outro aspecto importante a ser observado é que cada símbolo está associado a um conjunto de bits de forma que haja a variação de apenas um bit entre os símbolos adjacentes.

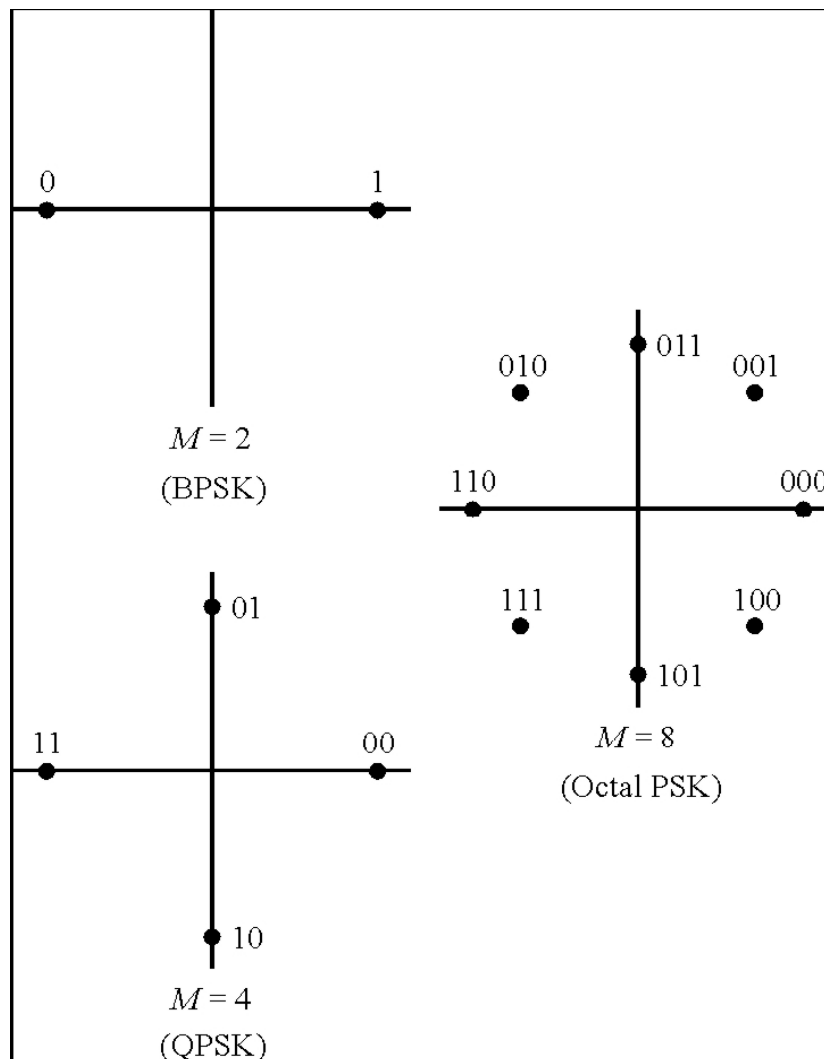


Figura 3.13 – Modelo de constelação PSK.

A constelação PSK é a forma de representar por vetores um sinal com modulação digital. O vetor I (*In-Phase*) indica que não houve alteração na fase da portadora ou houve mudança de 180°. O vetor Q indica que houve uma defasagem de 90° na fase da portadora.

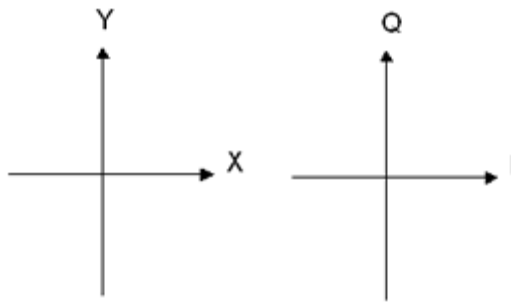


Figura 3.14 – Referência dos eixos.

3.6.5.2 – BPSK (*Bi Phase Shift Keying*)

A técnica de modulação por deslocamento de fase bipolar, é o processo pelo qual se altera a fase da onda portadora senoidal em função de um sinal digital modulante binário. Desta forma o sinal modulado chaveará entre duas fases acompanhando o sinal de entrada.

A forma mais usual de implementação da modulação BPSK é termos fase de 0° e 180° (inversão de fase de um estado para o outro), conforme mostra a Figura 3.15. Esse tipo de modulação é também conhecido como PRK (*Phase Reversal Keying*).

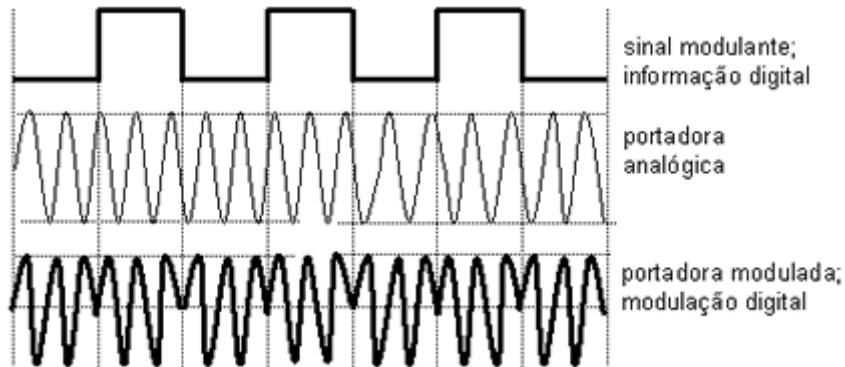


Figura 3.15 – Modelo de mudança de fase.

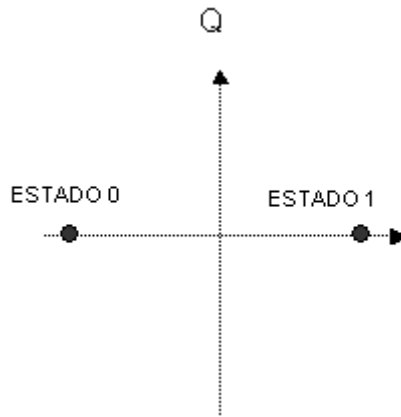


Figura 3.16 – Constelação BPSK.

3.6.5.3 – QPSK (*Quadrature Phase Shift Keying*)

A modulação QPSK (modulação por deslocamento de fase quaternário) é derivada do PSK onde são utilizados parâmetros de fase e quadratura da onda portadora para modular a informação. Em QPSK cada estado (símbolo) é formado por 2 bits: 00, 01, 10, 11.

Temos duas alternativas recomendadas pelo ITU *Radiocommunication Sector*, mostradas na Tabela 3.1 e exemplificadas pelas Figuras 3.17 e 3.18:

Tabela 3.1 – Alternativas angulares recomendadas [10].

Nível	Dibit	Mudança de fase	
		Alternativa A	Alternativa B
0	00	0	45 graus
1	01	90 graus	135 graus
2	10	180 graus	225 graus
3	11	270 graus	315 graus

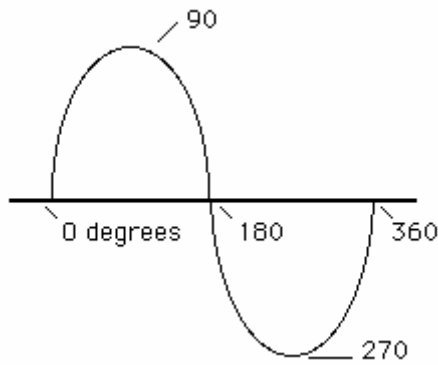


Figura 3.17 – Alternativa A.

QPSK

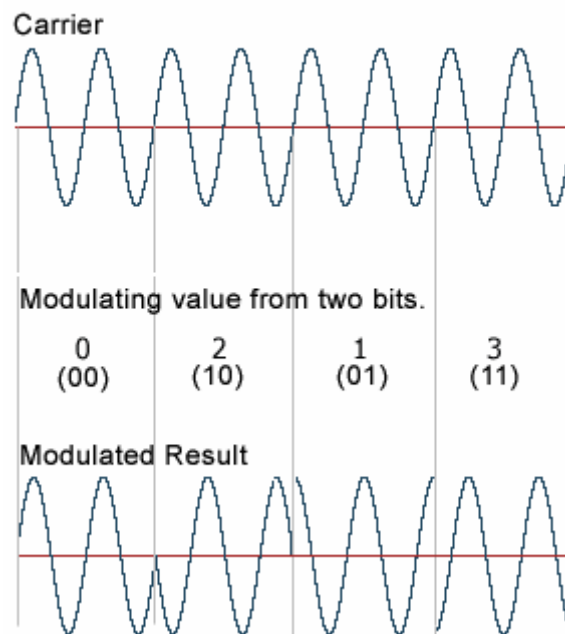


Figura 3.18 – Mudança de fase na modulação QPSK.

A técnica de modulação por deslocamento de fase quaternário, é o processo pelo qual se altera a fase da onda portadora senoidal em função de um sinal digital modulante binário agrupado em conjuntos de 2 bits (dibit). Sendo assim tenho 4 estados possíveis (ou seja 2^2). A cada dibit ocorre uma inversão na fase do sinal da portadora senoidal.

Em seguida, a informação a ser enviada é separada em grupos de dois bits. Assim, por exemplo, a *stream*: 11 00 11 10 00 10 00 01 00 ...

A seqüência é então convertida para sinal polar (converte 0 para -1). Fica: (11) (-1-1) (11) (1-1) (-1-1) (1-1) (-1-1) (-11) (-1-1).

E cada símbolo é separado em dois sinais que serão juntados posteriormente.

Esses canais são o I e o Q. Enviando alternadamente bits para os canais I e Q, tem-se:

- I: 1 -1 1 1 -1 1 -1 -1 -1
- Q: 1 -1 1 -1 -1 -1 -1 1 -1

O sinal de I e Q é filtrado para diminuir o ruído. Para facilitar, será mostrado em onda quadrada na Figura 3.19:

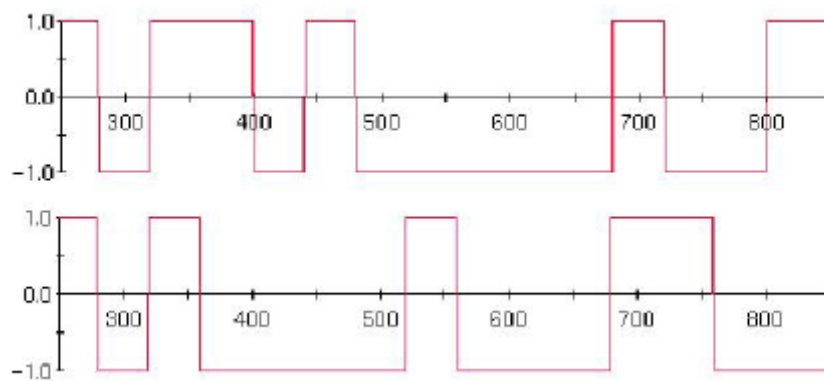


Figura 3.19 – Separação em grupos.

Multiplicando o canal I pelo seno, e o canal Q pelo cosseno, tem-se o resultado visto a seguir na Figura 3.20:

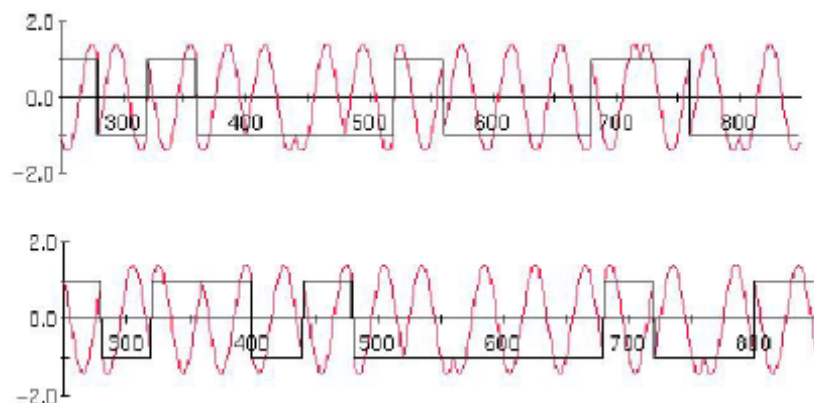


Figura 3.20 – Multiplicação.

Somando-se os dois canais, tem-se o sinal a ser transmitido, que é mostrado na Figura 3.21 a seguir:

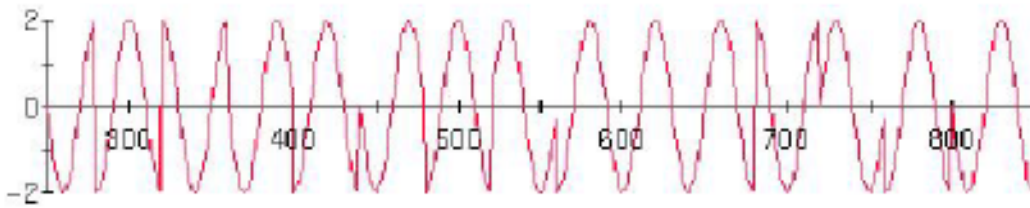


Figura 3.21 – Resultado final.

3.7 – INTERMODULAÇÃO

No enlace satélite há dois pontos onde se produzem intermodulações: o amplificador de potência da estação terrena (*uplink*) e o amplificador de potência do transponder (*downlink*).

Quando uma única portadora é amplificada ocorre a chamada distorção não linear (produtora de harmônicos). Quando várias portadoras compartilham o mesmo amplificador não-linear, ocorre a distorção de intermodulação.

3.8 – TÉCNICAS PARA MELHORIA DO SINAL (EQUALIZAÇÃO, CODIFICAÇÃO E DIVERSIDADE DO CANAL)

3.8.1 - Equalização

A ISI (*InterSymbol Interference*) é o maior obstáculo para alta taxas de transmissão, sendo causada por transmissão multipercurso.

Uma vez que o fading em canais móveis é aleatório e variante no tempo, usamos equalizadores adaptativos, mais utilizados em comunicações digitais onde os dados são segmentados em “*short time blocks*”, principalmente em sistemas TDMA. Na verdade os equalizadores são filtros responsáveis pela diminuição dos erros introduzidos pelo multipercurso.

Como algoritmos existentes para os filtros adaptativos podemos citar: LMS (*Least Mean Square*) onde o critério utilizado é a minimização do MSE (*Mean Square Error*), o RLS (*Recursive Least Square*) e o *Zero Forcing Algorithm*.

A finalidade da equalização é reconstruir o sinal original através de filtros lineares, filtros não lineares, adaptativos.

3.8.2 - Codificação

Codificação é a representação de sinais por símbolos mais adequados a uma transmissão eficiente. Podem ser utilizados codificadores de fonte e codificadores de canal.

A codificação de canal é uma das técnicas para a melhoria da qualidade do sinal com a conseqüente diminuição do BER (*Bit Error Rate*).

A codificação do canal acrescenta bits redundantes à mensagem, sendo usada para detectar (código de detecção de erro) ou corrigir erros (código de correção de erro). É aplicada antes da modulação, sendo a decodificação aplicada depois da demodulação, por isso a codificação é considerada uma técnica de detecção posterior. Sendo assim é tratada de modo independente da modulação. Contudo técnicas de modulação codificada Trellis, combinam codificação e modulação.

Em 1.948 Shannon demonstrou que, a fim de manter uma codificação da informação, os erros induzidos pelos ruídos de um canal podem ser reduzidos a qualquer nível desejado sem sacrificar a taxa de transferência desejada. Shannon usou o Teorema de Nyquist para considerar a influência do ruído na transmissão de dados. A medida que se aumenta o número de bits para quantização, mais difícil é a distinção entre sinal e ruído pois menor a diferença entre os níveis.

Esta relação é também conhecida como Lei de Hartley-Shannon e mostra que, dado certo canal com largura de faixa fixa, quanto maior for a relação sinal/ruído, maior será a capacidade do canal, ou seja, maior será a capacidade de transmissão de informações em bps.

A largura de faixa (B) é a diferença entre as frequências máxima e mínima do sinal. Em telefonia temos uma largura de faixa de 3.100 Hz (faixa de 300 a 3.400 Hz), em rádios AM

a largura de faixa do áudio é de 5kHz, em rádios FM a largura de faixa do áudio é de 15 kHz.

$$C = B \times \log_2 \left(1 + \frac{P}{N_0 \times B} \right) = B \times \log_2 \left(1 + \frac{S}{N} \right) \quad (3.5)$$

Onde:

- C é a capacidade do canal em bps;
- B é a largura de banda em Hz;
- P é a potencia do sinal recebido em Watts;
- N_0 é a densidade de potência do ruído em Watts/Hz;
- $P = E_b \times R_b$ onde;
- E_b é a média de energia por bit;
- R_b é a razão de transmissão por bit.

A finalidade básica da detecção e correção de erros é introduzir redundâncias nos dados para melhorar a performance do *link wireless*. A introdução de bits redundantes acarreta uma maior largura de banda. Isso leva a uma baixa eficiência do *link* com alta SNR, mas produz excelente performance BER com baixos valores de SNR.

Há dois tipos de codificação do canal: *block codes* e *convolutional codes*.

Block Codes são códigos de *Forward Error Correction* (FEC) que permitem a detecção e correção de um número limitado de bits, sem que haja retransmissão. Os *Block Codes* são utilizados para prover o aumento de performance de sistemas de comunicação, quando não há possibilidade de aumentar a potência de transmissão ou a utilização de técnicas de modulação mais sofisticadas. As seqüências de informações são agrupadas em blocos e codificadas.

Block Codes: Hamming Codes, Hadamard Codes, Golay Codes, Cyclic Codes, BCH Codes, Reed-Solomon Codes.

Códigos Reed-Solomon podem ser implementados em hardware e/ou software. Implementações em hardware são rápidas, mas não são aplicáveis para um grande número

de variações deste código. Esse código de correção de erro usa 188 bytes de cada conjunto de 204 bytes para transmissão da informação original.

Convolutional Codes são códigos de *Forward Error Correction* (FEC) que permitem o mapeamento de uma seqüência contínua de informações. Esse mapeamento é altamente estruturado, oferecendo uma melhor performance que o *Block Code*. A informação é passada através de um registrador de deslocamento de estados. A codificação desse código pode ser representada de diversas formas: *Generator Matrix*, *Generator Polynomials*, *Logic Table*, *State Diagram*, *Tree Diagram* e *Trellis Diagram*.

São exemplos de codificação convolucional os seguintes algoritmos: Viterbi, Fano's *Sequential Decoding*, *Stack Algorithm*, *Feedback Decoding*.

A TCM (*Trellis Coded Modulation*) é uma técnica que combina as técnicas de codificação e modulação.

O código de Viterbi é uma codificação de FEC expressa por uma razão onde o numerador indica o número de símbolos originais de entrada no codificador e o denominador indica o número total de símbolos que saem do codificador.

O código convolucional é particularmente útil na correção de erros provocados pelo ruído térmico, gerados em função da temperatura e da qualidade dos componentes eletrônicos utilizados.

FEC (*Forward Error Correction*) é uma técnica de correção antecipada de erros, que consiste na inserção de símbolos adicionais ao conteúdo original da mensagem. Este símbolo adicional é responsável pela indicação de erro.

Os resultados mostraram que uma melhoria de 3,3 dB sobre um sinal original (sem correção) poderia ser obtida através do uso de técnicas de correção de erros. Em outras palavras, um enlace de satélite que, normalmente, precisaria de uma antena de 1,8 metros de diâmetro para receber um sinal sem correção, poderia utilizar uma antena menor (ganho 3,3 dB menor), se o sinal fosse adicionado de técnicas de correção de erros, com a mesma qualidade anterior.

Quando se diz que códigos de FEC estão concatenados, significa dizer que a saída de um tipo de codificador está ligada à entrada de outro tipo de codificador, onde o primeiro código é chamado de código interno e o segundo de código externo.

Obviamente não se pode aumentar indefinidamente o número de bits de correção, uma vez que há um custo envolvido.

Em sistemas satelitais, existem fatores como o tipo de modulação e o fator de espalhamento dos filtros do satélite, que determinam a taxa máxima de símbolos a serem transportados.

Para a modulação QPSK e para a maioria dos filtros hoje utilizados, temos a seguinte aproximação:

- Taxa Máxima de Símbolos (MSps) = Largura de Banda (MHz) / 1,2

Portanto, para um dado transponder de 33 MHz, temos uma Taxa Máxima de Símbolos de $33 / 1,2 = 27,5$ MSps.

Como o QPSK utiliza 2 bits por símbolo, a Taxa de Transmissão de Bits é dada por $27,5$ MSps x 2 = 55 Mbps.

Considerando-se uma codificação de Viterbi de 3/4, temos que a Taxa de Transmissão de Bits Corrigida deste tipo de FEC será de 55 Mbps x 3/4 = 41,26 Mbps.

Concluimos que utilizando-se um transponder de 33 MHz e um FEC de 3/4, conseguimos transportar 41,26 Mbps; caso não utilizássemos nenhum tipo de FEC, conseguiríamos um transporte de 55 Mbps.

Os codificadores de fonte permitem escolher a qualidade do sinal digital através do número de bits por amostra. Para a comunicação de voz com faixa de 300 a 3.400 kHz e amostragem de 8 kHz, podemos ter sistemas de comunicação digital com diferentes taxas de dados por amostra. Ou seja, considerando que posso, por exemplo, utilizar 8 bits para identificar o nível de saída do meu sistema, o que quer dizer que posso estratificar a minha saída (o meu sinal quantizado) em 256 níveis de quantização disponíveis para melhor identificar meu sinal, uma vez que $2^8 = 256$.

Os codificadores de canal visam essencialmente à proteção contra erros no canal, através de códigos detectores de erro ou detectores combinados com corretores de erros.

3.8.3 - Diversidade

A diversidade é uma técnica poderosa que permite prover o aumento de confiabilidade da comunicação a um custo relativamente baixo. Ao contrário da técnica da equalização, não exige treinamento algum da rede. A diversidade se vale da própria característica da rádio propagação, procurando utilizar caminhos independentes ou altamente descorrelacionados e que, portanto, não estejam susceptíveis ao mesmo tipo de interferência.

A diversidade espacial (também conhecida como diversidade em antena) é uma das técnicas mais conhecidas em sistemas *wireless*, e pode ser tratada de diversas formas conforme as Figuras 3.22, 3.23 e 3.24. Para isso, as antenas devem ter uma separação mínima de $1,5 \lambda$.

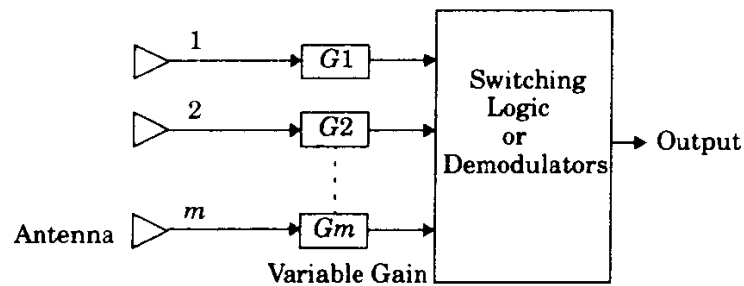


Figura 3.22 – Diversidade espacial tratada por switching [20].

Os sinais obtidos pela diversidade espacial podem ser tratados de diversas formas.

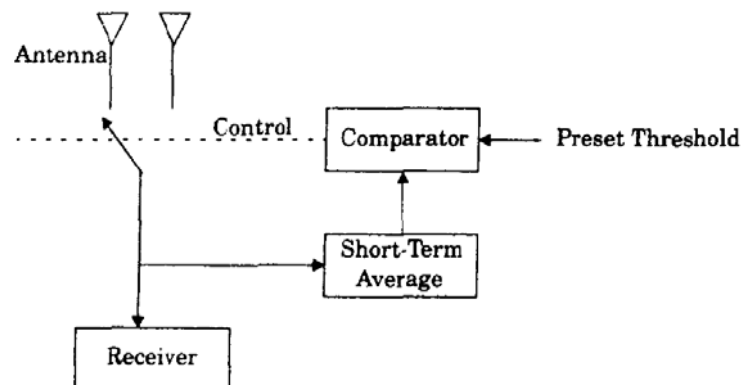


Figura 3.23 – Diversidade espacial tratada por comparadores [20].

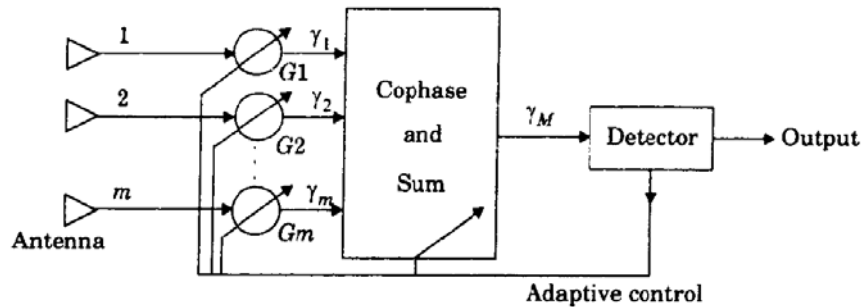


Fig 3.24 – Diversidade Espacial tratada por controles adaptativos [20].

A diversidade em polarização consiste na utilização de polarizações ortogonais entre si com a finalidade de explorar a diversidade.

A diversidade em frequência transmite informação em mais que uma portadora.

A diversidade em tempo reposiciona (espalha) os bits da informação ao longo do tempo, de forma a minimizar os efeitos do fading ou ruído em rajada (*noise burst*). O *interleaving* pode utilizar uma estrutura em blocos matriciais (*block*) ou uma estrutura convolucional (usada idealmente em conjunto com códigos convolucionais). Obviamente há um atraso inerente neste tipo de técnica, uma vez que os blocos devem estar completos para que seja aplicada a técnica do reposicionamento.

3.9 – TIPOS DE SERVIÇOS

Os diversos tipos de serviços de Telecomunicações via satélite, são agrupados de acordo com:

- Serviço Fixo Via Satélite ou FSS (*Fixed Satellite Service*)
- Serviço Móvel Via Satélite ou MSS (*Mobile Satellite Service*)

O Serviço Fixo caracteriza-se por estações terrenas de posição fixa e especificada. No Serviço Móvel, as estações terrenas não têm posição especificada, podendo ser móveis ou transportáveis.

3.10 – CODIFICAÇÃO GRAY

Em certas situações, a ordem crescente da representação decimal dos números binários, não é conveniente para representação de dados.

No nosso caso específico, considerando a modulação QPSK, a codificação Gray assegura que fases adjacentes diferem de apenas um bit. Ou seja, para que o ruído provoque um erro de 1 símbolo (SER), com um erro de 2 bits (BER), há de se ter um erro de fase maior que 135° . Ruídos que causam um erro de fase de 45° a 135° causarão um erro de apenas 1 bit.

Imaginemos que um sistema mecânico está sendo usado para codificar a posição de um marcador sobre um disco. Na Figura mostrada, os campos cinza representam um bit alto na leitura, e um campo branco representa um bit baixo.

A codificação digital é obtida à medida que o disco mostrado pela Figura 3.25 gira. Neste caso, utilizou-se a seqüência crescente da representação decimal dos números binários.

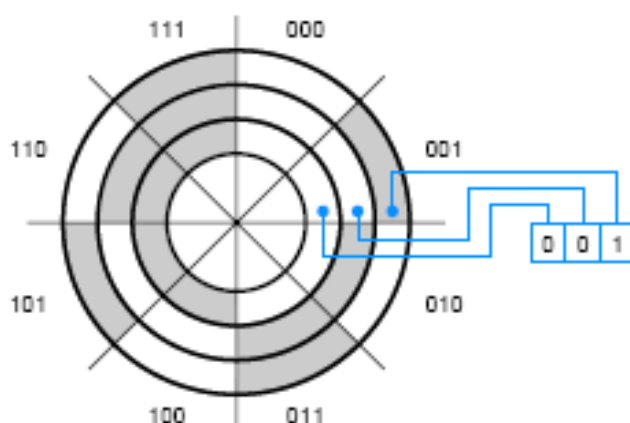


Figura 3.25 – Disco de codificação mecânica de posição angular [28].

O problema deste tipo de representação está certamente em transições em que vários bits mudam de estado ao mesmo tempo. Em uma transição da posição 111 e a posição 000, por exemplo, caso o sistema de leitura passe momentaneamente pelo estado 101 (porque talvez o mecanismo de leitura do segundo bit funcione ligeiramente mais devagar) há uma indicação incorreta de posição.

Desta forma é mais eficiente um sistema em que posições vizinhas tenham mudança de apenas um bit por vez.

De modo recursivo, o código de Gray é construído a partir da definição para 1 bit (que é 0 e 1). A partir daí aplica-se a seguinte regra para construção de um código de n+1 bits a partir de um código de n bits:

- As primeiras 2^n palavras de um código de Gray com (n + 1) bits são iguais às palavras do código de n bits, com um '0' adicional à esquerda;
- As últimas 2^n palavras de um código de Gray são iguais às palavras do código de n bits em ordem reversa, com um '1' adicional à esquerda.

Tabela 3.2 – Codificação Gray [22].

Posição	Código Binário	Gray 1 bit	Gray 2 bits	Gray 3 bits
0	000	0	00	000
1	001	1	01	001
2	010		11	011
3	011		10	010
4	100			110
5	101			111
6	110			101
7	111			100

3.11 - CÁLCULO SIMPLIFICADO DE ENLACE

O cálculo de enlace visa permitir o correto dimensionamento da rede, sendo fundamental para garantir o desempenho do enlace dentro dos parâmetros técnicos e orçamentários.

A intenção aqui, é de apenas esboçar como se faz um cálculo deste tipo. Obviamente, para um *link budget*, deve-se ter um profundo conhecimento do sistema em todas as suas fases, pois só assim todos os fatores relevantes serão considerados.

A Figura 3.26 nos mostra alguns parâmetros importantes utilizados em um cálculo simplificado. Observe que a atenuação no espaço livre é sempre maior para frequências maiores; em conseqüência, usamos para o *uplink* a maior frequência dentro de uma faixa, a fim de poupar o transponder no *downlink*, uma vez que os recursos em termos de potência do transponder, são bastante limitados

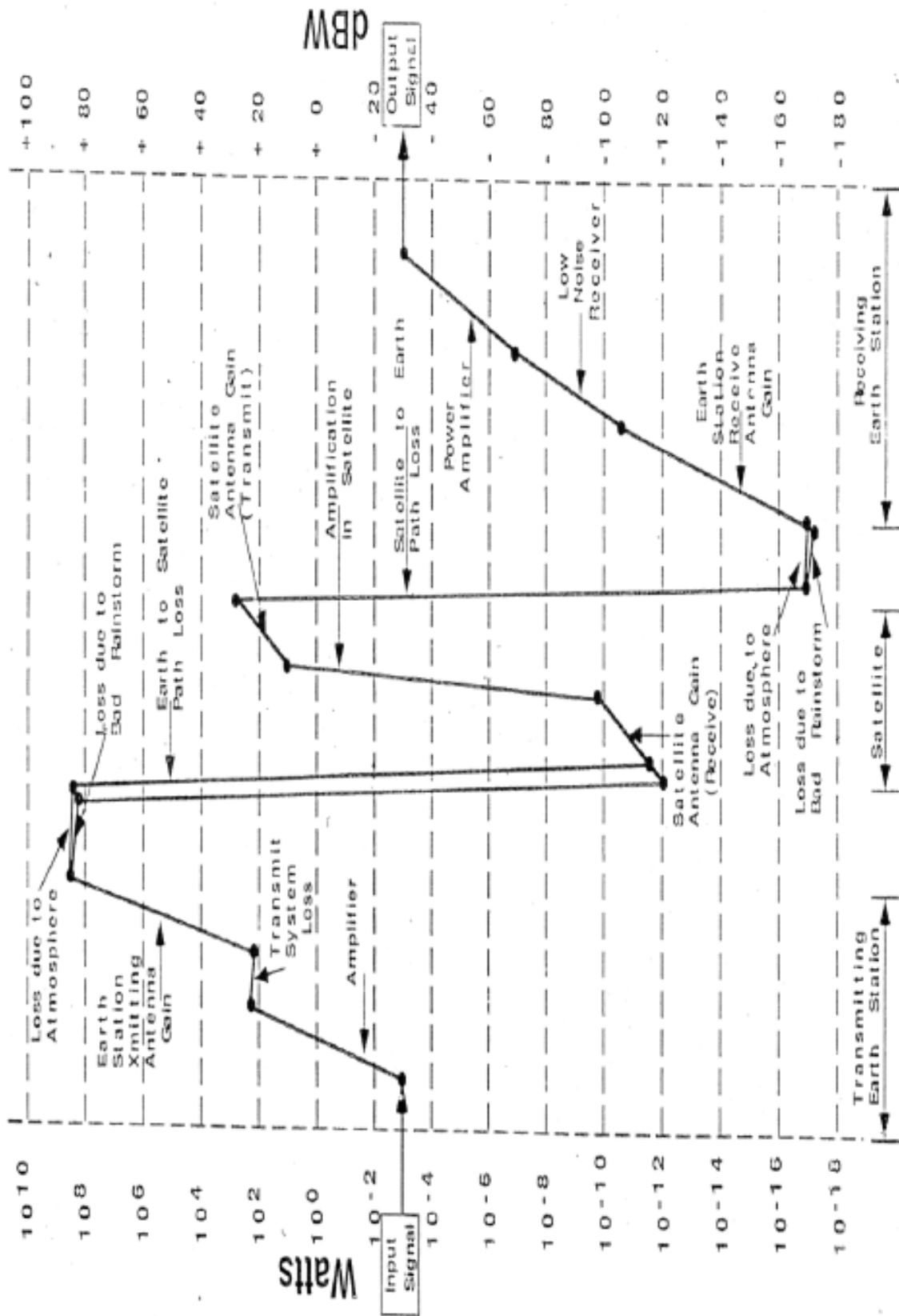


Figura 3.26 – Esboço de um cálculo de enlace.

Apenas lembrando, uma relação de potências $R_p = 10 \log (P_{OUT} / P_{IN})$ em dB, pode ser dada por:

- $10 \log (P / 1\text{mW})$ em dBm;
- $10 \log (P / 1\text{W})$ em dBW;

3.11.1 – EIRP na estação terrena

O valor que inicialmente para nós, é de extrema importância, é o valor de EIRP (*Effective Isotropic Radiated Power*) na saída da antena terrena, conforme Figura 3.27. A potência equivalente isotrópica irradiada é dada por:

$$\text{EIRP} = P_T + G_T \quad (3.6)$$

Onde:

- P_T é a potência que chega à antena, dada em dBW;
- G_T é o ganho de transmissão na direção considerada, dado em dBi;
- EIRP é a potência equivalente isotrópica irradiada, expressa em dBW.

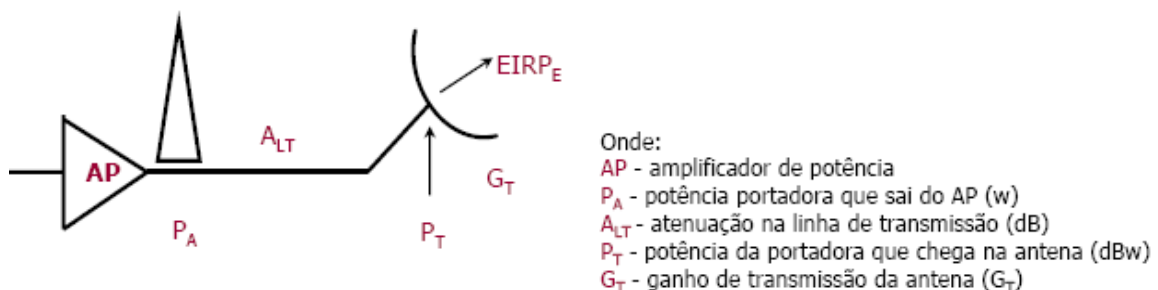


Figura 3.27 – Diagrama simplificado de emissão terrestre.

No nosso caso específico temos:

- $P_A = 45 \text{ W}$;
- $A_{LT} = 2 \text{ dB}$;
- $G_T = 46 \text{ dBi}$ para 6 GHz no *uplink*.

$$\text{EIRP} = 10 \cdot \log (45\text{W} / 1 \text{ W}) - 2 \text{ dB} + 46 \text{ dBi} = 60,53 \text{ dBW} \quad (3.7)$$

3.11.2 – Atenuação no espaço livre (*uplink*)

Atenuação no espaço livre, exemplificada pela Figura 3.28, é dada por $A_{EL} = 32,4 + 20 \log F + 20 \log D$, onde A_{EL} é dada em dB, F em MHz e D em km. É importante lembrar que no *uplink*, sempre usamos as maiores frequências, uma vez que elas sofrem uma maior atenuação e, teoricamente, os equipamentos terrestres possuem maior disponibilidade de potência e espaço para maiores antenas.

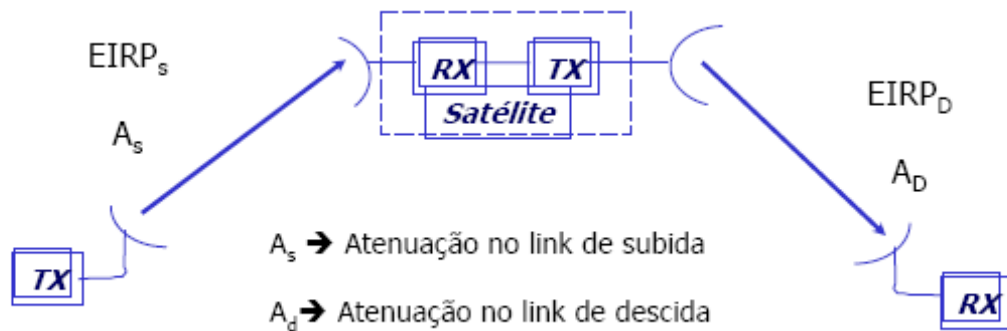


Figura 3.28 – Diagrama de enlace.

No nosso caso específico temos:

- $F = 6.000 \text{ MHz}$;
- $D = 36.000 \text{ km}$.

$$A_{EL} = 32,4 + 20 \log (6.000) + 20 \log (36.000) = 199,08 \text{ dB} \quad (3.8)$$

3.11.3 – Ocupação da banda

A taxa de ocupação da largura de banda (*bandwidth*) é dada pela expressão:

$$BW_{AL} = R_B \cdot \frac{1}{FEC} \cdot \frac{1}{\log_2 m} \cdot f_s \quad \text{onde} \quad (3.9)$$

- BW_{AL} é a *bandwidth*;
- R_B é o *bit rate* em bps;
- FEC é a taxa do código corretor de erros utilizado, no caso estaremos considerando o código de Viterbi a 3/4;

- m é o número de fases do modulador, 2 para BPSK, 4 para QPSK, 8 para 8PSK, 16 para 16QAM, 16 para 16APSK e 32 para 32APSK;
- f_s é o espaçamento de frequências (típico de 1,4).

No nosso caso teremos:

- $R_B = 64$ kbits/s;
- $FEC = 3/4$;
- modulação QPSK (e portanto $m=4$);
- $f_s = 1,3$.

$$BW_{AL} = 64k \times 4/3 \times 1/\log_2 4 \times 1,3 = 55,5 \text{ kHz} \quad (3.10)$$

4 – REDES NEURAIAS

A partir da evidente constatação da evolução das máquinas, de seu uso e influência em nossas vidas, um dos grandes desejos do homem tem sido a criação de uma máquina cognitiva, que possa operar independentemente do controle humano. Uma máquina cuja independência seja desenvolvida de acordo com seu próprio aprendizado e que tenha a capacidade de interagir com ambientes incertos (desconhecidos por ela) [8].

O sucesso de uma máquina autônoma, depende única e exclusivamente de sua capacidade de lidar com uma variedade de eventos inesperados no ambiente em que opera. Esta máquina teria maior capacidade de aprender tarefas de alto nível cognitivo que não são facilmente manipuladas por máquinas atuais, e continuaria a se adaptar e realizar tais tarefas gradativamente com maior eficiência, mesmo que em condições ambientais imprevisíveis. Então, seriam muito úteis onde a interação humana é perigosa, tediosa ou impossível.

Organismos humanos, são uma fonte de motivação para o desenvolvimento desta máquina, e proporcionam diversas dicas para o desenvolvimento de algoritmos de aprendizado e adaptação . Assim, espera-se que algumas das características de organismos biológicos de aprendizado e adaptação estejam presentes.

Enquanto computadores funcionam de modo sequencial, proporcionando maior eficiência na resolução de tarefas perfeitamente delimitadas, o cérebro humano funciona de modo paralelo, e sendo extremamente conectado, é mais eficiente na resolução de tarefas que exigem várias variáveis. O motivo pelo qual máquinas inspiradas na biologia são diferentes das máquinas atuais, encontra-se no fato de que as máquinas atuais baseiam seu processamento explicitamente em modelos matemáticos. Mecanismos de controle baseados em mecanismos neurais, podem coordenar diversos graus de liberdade durante a execução de tarefas manipulativas e em ambientes desestruturados. Eles são capazes de lidar com tarefas complicadas sem que tenham que desenvolver um modelo matemático.

A teoria de redes neurais artificiais vem se consolidando como uma eficiente ferramenta para se lidar com problemas complexos, em que extensa massa de dados deve ser modelada e analisada em um contexto multidisciplinar, envolvendo aspectos estatísticos e computacionais além dos dinâmicos e de otimização.

Redes neurais são concepções computacionais baseadas em unidades de processamento pequenas e simples, onde as interligações formam uma rede. Ou seja, rede neural é uma rede com muitos processadores simples (cada um tendo uma pequena quantidade de memória), conectados por meio de canais de comunicação (conexões) aos quais, usualmente, estão associados valores (pesos) numéricos. Uma rede pode então ser caracterizada por seus processadores básicos (chamados neurônios), pela função de ativação (que representa o estado de um neurônio), pelo padrão de conexão existente entre os neurônios e por seu algoritmo de treinamento (algoritmo de aprendizado).

O comportamento de um neurônio é determinado pelas funções a ele associadas, e pelas conexões por meio das quais recebe entradas e transmite sua saída.

A organização dos neurônios em camadas, bem como os padrões de conexão dentro e entre camadas, caracteriza a arquitetura (topologia) da rede neural.

De acordo com diversas estruturas (topologias ou arquiteturas) neurais e algoritmos de aprendizagem propostos por vários pesquisadores, redes neurais possuem certas características exclusivas de sistemas biológicos. Tais características entram em conflito com os tradicionais métodos computacionais. Sistemas de computação baseados em redes neurais têm a capacidade de receber ao mesmo tempo várias entradas e distribuí-las de maneira organizada. Geralmente, as informações armazenadas por uma rede neural são compartilhadas por todas as suas unidades de processamento. Característica que contrasta com os atuais esquemas de memória, onde a informação fica confinada em um determinado endereço.

Em um sistema de rede neural, a informação pode parecer ter representação redundante, porém, o fato de que ela se encontre distribuída por todos os elementos da rede, significa que mesmo que parte da rede seja destruída, a informação contida nesta parte ainda estará presente na rede, e poderá ser recuperada. Portanto, a redundância na representação de informações em uma rede neural, transforma-se em uma vantagem, pois torna o sistema tolerante a falhas. Os atributos de uma rede neural, tais como aprender através de exemplos, generalizações redundantes, e tolerância a falhas, proporcionam fortes incentivos para a escolha de redes neurais como uma escolha apropriada para aproximação para a modelagem de sistemas biológicos.

A principal força na estrutura de redes neurais reside em sua habilidade de adaptação e aprendizagem. A habilidade de adaptação e aprendizagem pelo ambiente significa que modelos de redes neurais podem lidar com dados imprecisos e situações não totalmente definidas. Uma rede treinada de maneira razoável tem a habilidade de generalizar quando é apresentada a entradas que não estão presentes em dados já conhecidos por ela.

A característica mais significativa de redes neurais está em sua habilidade de aproximar qualquer função contínua não linear. Esta habilidade das redes neurais as tem tornado úteis para modelar sistemas não lineares na combinação de controladores não lineares.

Redes neurais podem ter várias entradas e várias saídas, sendo facilmente aplicáveis à sistemas com muitas variáveis.

Com o avanço em tecnologias de hardware, existem componentes com funções voltadas a sistemas com implementações voltadas para redes neurais, o que traz uma velocidade adicional à computação neural.

As primeiras informações mencionadas sobre a neurocomputação datam de 1943, em artigos de McCulloch e Pitts, em que sugeriam a construção de uma máquina baseada ou inspirada no cérebro humano. Em 1949 Donald Hebb escreveu um livro intitulado "*The Organization of Behavior*" (A Organização do Comportamento), que perseguia a idéia de que o condicionamento psicológico clássico está presente em qualquer parte dos animais pelo fato de que esta é uma propriedade de neurônios individuais. Suas idéias não eram completamente novas, mas Hebb foi o primeiro a propor uma lei de aprendizagem específica para as sinapses dos neurônios, conforme [12] e [16].

Também proveniente deste período de tempo foi a construção do primeiro neuro computador, denominado Snark, por Mavin Minsky, em 1951. O Snark operava com sucesso a partir de um ponto de partida técnico, ajustando seus pesos automaticamente, entretanto, ele nunca executou qualquer função de processamento de informação interessante, servindo de inspiração para novos modelos. O primeiro neuro computador a obter sucesso (Mark I Perceptron) surgiu em 1957 e 1958, criado por Frank Rosenblatt, Charles Wightman e outros. Devido a profundidade de seus estudos, suas contribuições técnicas e de sua maneira moderna de pensar, muitos o vêem como o fundador da

neurocomputação na forma em que a temos hoje. Seu interesse inicial para a criação do Perceptron era o reconhecimento de padrões.

Após Rosenblatt, Bernard Widrow, com a ajuda de alguns estudantes, desenvolveu um novo tipo de elemento de processamento de redes neurais chamado de Adaline. Widrow também fundou a primeira companhia de hardware de neurocomputadores e componentes.

Nos anos 80, muitos dos pesquisadores foram bastante corajosos e passaram a publicar diversas propostas para a exploração de desenvolvimento de redes neurais bem como suas aplicações. Porém, talvez o fato mais importante deste período tenha ocorrido quando Ira Skurnick, um administrador de programas da DARPA (*Defense Advanced Research Projects Agency*) que decidiu ouvir os argumentos da neuro computação e seus projetistas, e divergindo dos caminhos tradicionais dos conhecimentos convencionais, fundou em 1983 pesquisas em neuro computação. Este ato não só abriu as portas para a neurocomputação, como também deu à DARPA o status de uma das líderes mundiais em se tratando de "moda" tecnológica.

Outro pesquisador que emergiu neste período foi John Hopfield, renomado físico de reputação mundial, se interessou pela neurocomputação, e escreveu artigos que percorreram o mundo todo persuadindo centenas de cientistas, matemáticos, e tecnólogos altamente qualificados a se unirem nesta nova área emergente.

Apesar de um terço dos pesquisadores da área terem aderido à mesma pela influência de Hopfield, foi em 1986 que este campo de pesquisa "explodiu" com a publicação do livro "*Parallel Distributed Processing*" (Processamento Distribuído Paralelo) de David Rumelhart e James McClelland.

Em 1987 ocorreu em São Francisco a primeira conferência de redes neurais dos tempos modernos, a *IEEE International Conference on Neural Networks*, e também foi formada a *International Neural Networks Society* (INNS). A partir destes acontecimentos decorreram a fundação do *INNS journal* em 1989, seguido do *Neural Computation* e do *IEEE Transactions on Neural Networks* em 1990.

4.1 – O QUE É REDE NEURAL

Como visto anteriormente, redes neurais são um método geral e prático para o aprendizado de aproximações de funções a partir de exemplos.

Uma Rede Neural pode ser caracterizada por:

- Processadores básicos, chamados neurônios.
- Pelo padrão de conexão (topologia ou arquitetura).
- Pelo algoritmo de treinamento (algoritmo de aprendizado).
- Pela função de ativação.

4.2 – TOPOLOGIA OU ARQUITETURA

Define a organização dos neurônios em camadas, bem como os padrões de conexão dentro e entre camadas.

4.2.1 – Neurônio booleano de McCulloch

Na década de 40, o neurofisiologista Warren McCulloch e o matemático Walter Pitts foram os pioneiros na teoria das redes neurais.

Em essência, o trabalho fazia uma analogia entre células vivas e o processo eletrônico, simulando o comportamento do neurônio natural, onde o neurônio possuía apenas uma saída, que era uma função de entrada da soma do valor de suas diversas entradas. Isto é mostrado nas Figuras 4.1 e 4.2.

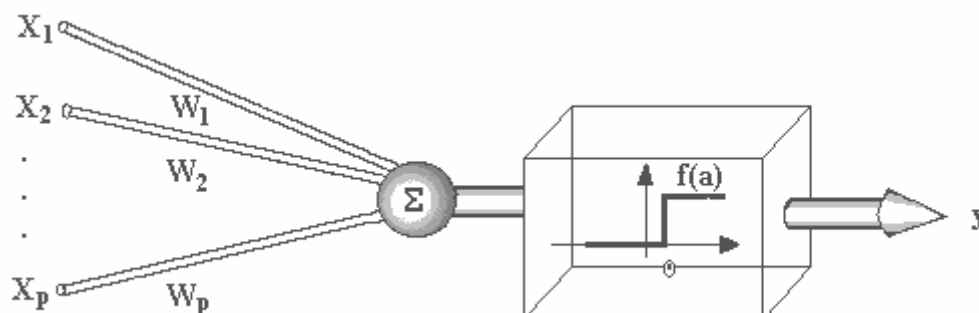


Figura 4.1 - Neurônio artificial projetado por McCulloch.

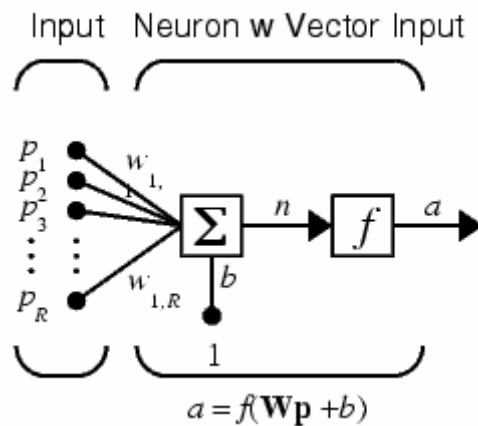


Figura 4.2 – Representação matemática do neurônio [15].

onde

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b \quad (4.1)$$

Neste modelo, existe apenas um neurônio capaz de resolver problemas linearmente separáveis, ou seja, existe um discriminador linear capaz de executar a função.

Embora neurônios singelos não tenham a capacidade de implementar todas as funções booleanas (resolvem apenas as funções E e OU), sempre existirá uma rede de múltiplos neurônios que implementará uma função booleana.

4.2.2 - Perceptrons

No final da década de 50, Roseblatt criou uma rede de múltiplos neurônios dispostos em uma camada de entrada e uma camada de saída. Os neurônios que recebem diretamente as entradas da rede constituem o que se chama de camada de entrada. Os neurônios que recebem como entradas as saídas daqueles neurônios da camada de entrada constituem a camada final que é a camada de saída. As camadas internas que não são nem a de entrada, nem a de saída, usadas em modelos mais sofisticados, são geralmente referidas como camadas ocultas ou escondidas.

Alguns autores referem-se ao vetor de entrada como sendo a camada de entrada, constituído por neurônios de ligação direta.

Neste modelo de arquitetura (topologia), os neurônios são organizados de modo que os pesos das conexões são adaptados a fim de se atingir a eficiência sináptica.

Hebb em 1949, propôs um princípio de aprendizado pelo qual o aprendizado em sistemas nervosos complexos poderia ser reduzido a um processo puramente local, em que a intensidade das conexões sinápticas é alterada apenas em função dos erros detectáveis localmente. Este princípio local pode ser traduzido pela regra [12]:

$$w_i^{\text{nov}} = w_i^{\text{velho}} + \Delta w_i \quad (4.2)$$

$$\Delta w_i = \eta \cdot (y_l^d - y_l) \cdot x_{i,l}^d \quad (4.3)$$

onde:

- w_i : é o peso sináptico do neurônio i ;
- η : é o parâmetro que define a taxa de aprendizado, ou seja, como os ganhos são alterados em consequência dos erros. É regra local, na medida em que não depende dos valores das demais variáveis espalhadas pela rede;
- y_l^d : é a saída desejada do l -ésimo exemplo;
- y_l : é a saída real do l -ésimo exemplo;
- $x_{i,l}$: é a entrada do neurônio i em seu l -ésimo exemplo.

Comparativamente com McCulloch e Pitts, pode-se dizer que o Modelo de Rosenblatt proporcionou a extensibilidade a estímulos que podem assumir valores contínuos, em oposição aos valores binários, além disso, introduziu uma lei de treinamento para os perceptrons. Esta evolução é mostrada na Figura 4.3.

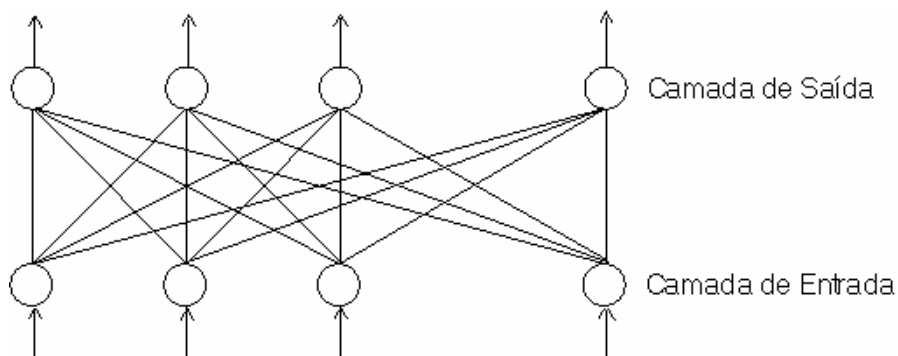


Figura 4.3 - Rede de perceptrons proposta por Rosemblatt.

4.2.3 – Perceptron multicamada

O perceptron multicamada é a generalização do perceptron, onde temos, no mínimo, uma rede de neurônios composta por: camada de entrada, camada escondida e camada de saída.

Rumelhart, Hinton e Williams introduziram a topologia (arquitetura) de perceptrons multicamadas com o uso do algoritmo *Backpropagation* (retropropagação do erro), desenvolvido exatamente para essa arquitetura [12]. Esse método numérico é capaz de realizar a busca com base em gradiente apenas em um espaço de busca associado a redes que tem uma topologia fixa. O treinamento de redes multicamadas usando o *Backpropagation* é útil apenas quando a arquitetura de rede é escolhida corretamente. Se por um lado redes com arquitetura muito pequena não aprendem bem a solução do problema, por outro, redes muito grandes podem levar ao *overfitting* (quando uma rede não é capaz de reconhecer novas instâncias, possivelmente a um treinamento excessivo da rede), e a um desempenho ruim.

Esta classe de rede consiste de múltiplas camadas de unidades computacionais, geralmente interconectadas em uma forma *feedforward*. Isso quer dizer que cada neurônio em uma camada, tem conexões diretas a neurônios da próxima camada. Em muitas aplicações, as unidades dessas redes aplicam uma função sigmóide (em forma de S) como função de ativação.

O teorema de aproximação universal, dita que toda função contínua que mapeia intervalos de números reais a algum intervalo de números reais de saída, pode ser arbitrariamente aproximada com precisão por um perceptron multicamada com somente uma camada oculta. Este resultado só é válido para classes restritas de funções de ativação, por exemplo, funções sigmóides.

Redes multicamadas podem usar um grande número de técnicas de aprendizado (algoritmo), sendo que a mais popular é a propagação reversa. Neste caso os valores de saída são comparados com a resposta correta para computar o valor de alguma função-erro predefinida. Por alguma técnica o erro é então alimentado de volta na rede. Usando essa informação, o algoritmo ajusta os pesos de cada conexão para reduzir o valor da função erro.

O perceptron multicamada foi concebido para resolver problemas mais complexos, os quais não poderiam ser resolvidos pelo modelo de neurônio básico. Um único perceptron ou uma combinação das saídas de alguns perceptrons poderia realizar uma operação XOR, porém, seria incapaz de aprendê-la. Para isto são necessárias mais conexões, as quais só existem em uma rede de perceptrons dispostos em camadas. Os neurônios internos são de suma importância na rede neural, pois provou-se que sem estes, torna-se impossível a resolução de problemas linearmente não separáveis. Em outras palavras pode-se dizer que uma rede é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede, vide as Figuras 4.4 e 4.5.

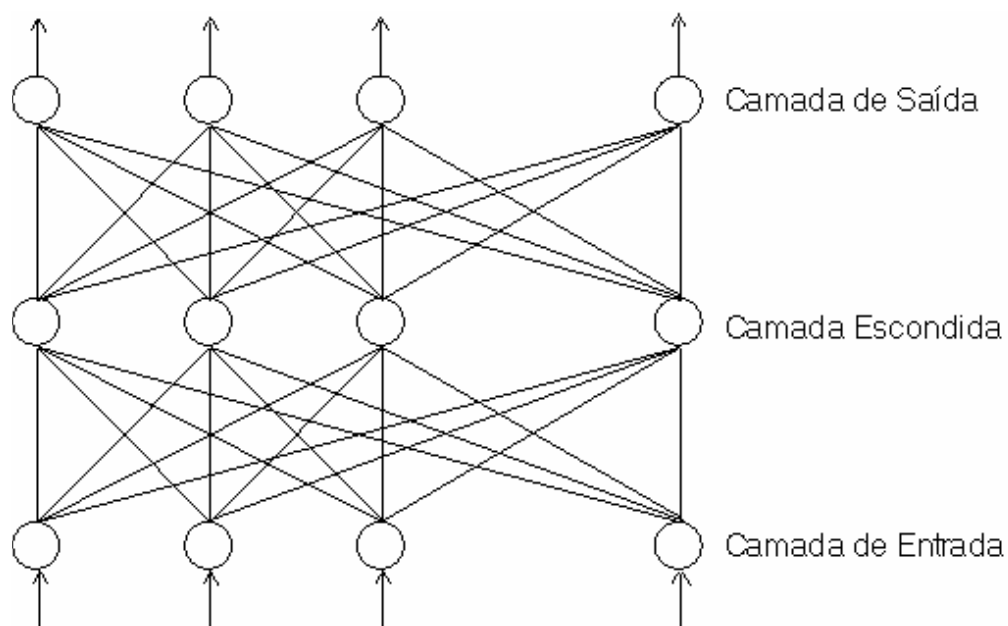


Figura 4.4 - Estrutura do método *backpropagation*.

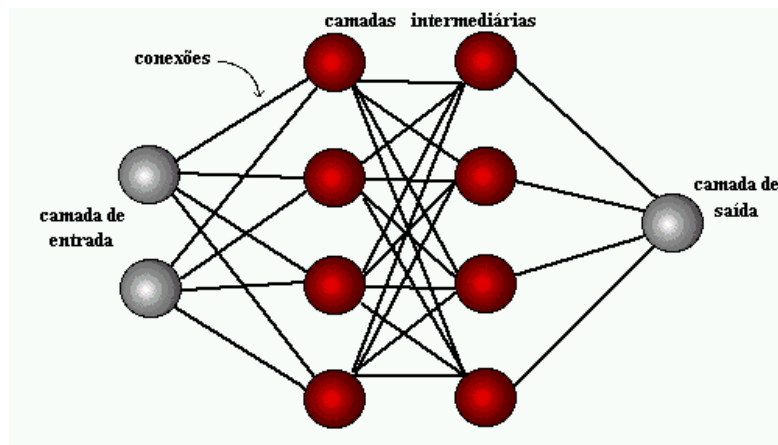


Figura 4.5 - Organização em camadas.

Usualmente as camadas são classificadas em três grupos:

- camada de entrada: onde os padrões são apresentados à rede;
- camadas intermediárias ou ocultas: onde é feita a maior parte do processamento, através das conexões ponderadas. Podem ser consideradas como extratoras de características;
- camada de saída: onde o resultado final é concluído e apresentado.

4.3 – ALGORITMO DE TREINAMENTO (APRENDIZADO)

Conjunto de regras por meio das quais os pesos das conexões são ajustados, usando para o ajuste, um conjunto de exemplos que representa a função a ser aprendida. Esse conjunto de exemplos é geralmente chamado de conjunto de treinamento.

Uma rede neural sempre aprende a partir de um conjunto de treinamento, o aprendizado se reflete na capacidade de generalização que a rede vai exibir, quando usada em novas situações.

Outro fator importante é a maneira pela qual uma rede neural se relaciona com o ambiente. Nesse contexto existem os seguintes paradigmas de aprendizado:

a - por retroação do mundo

Diz respeito a ausência ou presença de realimentação explícita do mundo exterior, ou seja, que em certos intervalos de tempo um agente assinala acertos e erros.

- aprendizado supervisionado (aprendizado usando algoritmo supervisionado): utiliza um agente externo que indica à rede um comportamento bom ou ruim de acordo com o padrão de entrada. A descrição de cada exemplo (vetor ou padrão) de treinamento traz também a classe (ou vetor de saída) que o exemplo representa. Durante a fase de treinamento, o algoritmo de treinamento da rede tenta ajustar os pesos das conexões de maneira que a saída de rede, coincida com a classe (ou vetor de saída) associada ao exemplo, para cada exemplo do conjunto de treinamento.
- aprendizado não supervisionado (auto-organização): não utiliza um agente externo indicando a resposta desejada para os padrões de entrada, utiliza entretanto, exemplos de coisas semelhantes para que a rede responda de maneira semelhante. A RN é treinada sem considerar a informação da classe associada a cada exemplo de treinamento. O aprendizado não supervisionado funciona por meio da modificação de pesos de conexões, de maneira que exemplos semelhantes entre si sejam associados à mesma unidade (ou cluster) de saída.

b - por finalidade do aprendizado:

- auto-associador: é apresentada à rede uma coleção de exemplos para que ela memorize. Quando se apresenta um dos elementos da coleção de exemplos, mas de modo errôneo, a rede deve mostrar o exemplo original, funcionando assim como um filtro.
- hetero-associador: é uma variação do auto-associador, mas que se memoriza um conjunto de pares. O sistema aprende a reproduzir o segundo elemento do par mesmo que o primeiro esteja pouco modificado, funcionando desta maneira como um reconhecedor de padrões.

Finalmente, pode-se dizer que algoritmos de treinamento se dividem em:

- não construtivo: pressupõe a definição de uma arquitetura fixa de rede. São exemplos de algoritmos não construtivos que utilizam a topologia perceptron: Princípio de Aprendizado de Hebb (Hebb, 1949), *Backpropagation* (Rummelhart, 1986), *Self-Organizing Maps* (Mapas Auto-Organizados de Kohonen, por Kohonen, 1972), Memória Associativa (Hopfield, 1982), *Pocket* (Gallant, 1986), *Pocket com Modificação Ratchet* (Gallant, 1986), *Thermal Perceptron* (Frean, 1990) e *Loss Minimization* (Hrycej, 1992).
- construtivo: modelo de aprendizado neural que não pressupõe a definição de uma arquitetura de rede fixa, antes do início do treinamento. A principal característica desse

modelo de aprendizado é a construção dinâmica das camadas intermediárias da rede, à medida que vão sendo necessárias ao treinamento. Essa característica faz com que o processo de determinação da arquitetura da rede fique interligado ao próprio processo de aprendizado, onde ambos os processos, o de treinamento e de construção da rede, acontecem simultaneamente e são interdependentes. São exemplos de algoritmos construtivos *Tower* (Gallant, 1986), *Pyramid* (Gallant, 1986), *Tiling* (Mézard & Nadal, 1989), *Upstart* (Frean, 1990), *DistA1* (Yang et al, 1997) e *Cascade-Correlation* (Fahlman & Lebiere, 1990). Podem ser classificados em duas categorias, os *forward*, que constroem a rede camada por camada a partir da camada de entrada até a camada de saída ser finalmente construída, e os *backward*, que inserem novos neurônios entre a camada de entrada e a última camada construída.

4.4 – FUNÇÕES DE ATIVAÇÃO

A função de ativação do neurônio $g(v)$, possui comportamento monotônico sobre uma faixa do argumento v (faixa dinâmica) e a saturação fora dessa faixa. Existem várias funções de ativação que procuram, em maior ou menor grau, preservar tais características com maior ou menor fidelidade:

- função de ativação do tipo identidade é dada por $g(v) = v$. Neste caso a monotonicidade é preservada desprezando-se o fenômeno da saturação.
- função de ativação do tipo $g(v) = h(v)$ ou degrau unitário. Apenas a saturação é preservada.
- função de Siebert é a função de ativação do tipo $g(v) = \frac{v^K}{v^K + \alpha}$, onde $v \geq 0$. Essa função preserva tanto a característica de saturação como a de monotonicidade na faixa dinâmica, permitindo inclusive ajustar o ganho médio na faixa mediante o parâmetro K .

As funções de ativação mais comumente usadas são as funções sigmóide, cujo tipo mais comum é a função logística e a tangente hiperbólica que, além de exibirem todas as vantagens da função de Siebert, são simétricas e possuem todas as derivadas contínuas:

- a função sigmóide do tipo logística, é dada por $g(v) = (1 + \exp(-\alpha \cdot v))^{-1}$ e sua respectiva derivada é dada por $\frac{d}{dv}(g(v)) = g(v) \cdot (1 - g(v))$. Esta função é exibida na Figura

4.6, e pode assumir todos os valores entre 0 e 1, onde α é o parâmetro de inclinação da função, e v é o valor de ativação do neurônio.

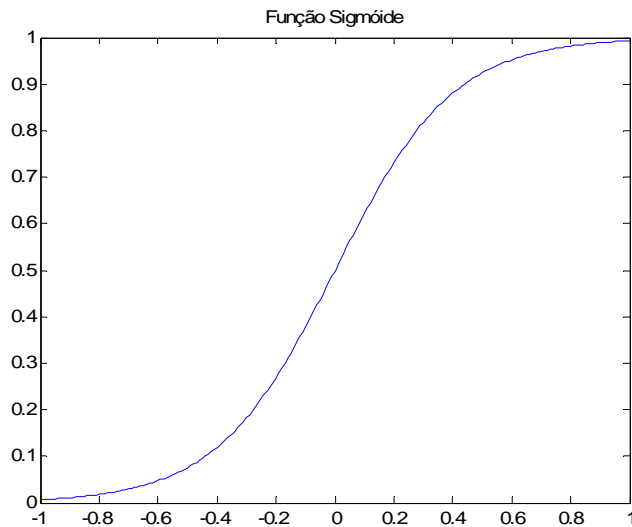


Figura 4.6 – Função sigmóide.

- a função tangente hiperbólica, é exemplificada pela Figura 4.7, sendo dada por

$$g(v) = \tanh(v) = \frac{1 - \exp(-\alpha \cdot v)}{1 + \exp(-\alpha \cdot v)} \quad \text{e sua respectiva derivada é dada por}$$

$$\frac{d}{dv}(\tanh(v)) = (1 - g(v))^2. \quad \text{Observe que } \alpha \text{ é o parâmetro de inclinação da função, e } v \text{ é o}$$

valor de ativação do neurônio.

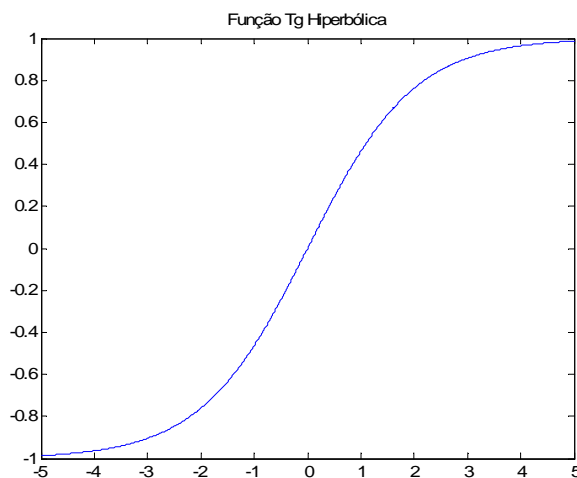


Figura 4.7 – Função tangente hiperbólica.

4.5 – TÉCNICA DE DESENVOLVIMENTO DE REDE NEURAL

4.5.1 – Coleta de dados e separação em conjuntos

Os primeiros passos do processo de desenvolvimento de redes neurais artificiais são a coleta de dados relativos ao problema e a sua separação em um conjunto de treinamento e um conjunto de testes. Esta tarefa requer uma análise cuidadosa sobre o problema para minimizar ambiguidades e erros nos dados. Além disso, os dados coletados devem ser significativos e cobrir amplamente o domínio do problema, não devendo cobrir apenas as operações normais ou rotineiras, mas também as exceções e as condições nos limites do domínio do problema.

Depois de determinados estes conjuntos, eles são geralmente colocados em ordem aleatória para prevenção de tendências associadas à ordem de apresentação dos dados. Além disso, pode ser necessário pré-processar estes dados, através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização na rede.

4.5.2 – Configuração da rede

O próximo passo é a definição da configuração da rede, que pode ser dividido em três etapas:

- seleção do paradigma neural apropriado à aplicação;
- determinação da topologia da rede a ser utilizada, número de camadas, o número de unidades em cada camada, etc...;
- determinação de parâmetros do algoritmo de treinamento e funções de ativação, tendo este passo um grande impacto na performance do sistema resultante.

Existem metodologias que auxiliam a condução destas tarefas. Normalmente estas escolhas são feitas de forma empírica. A definição da configuração de redes neurais é ainda considerada uma arte, que requer grande experiência dos projetistas.

4.5.3 - Treinamento

A propriedade mais importante das redes neurais é a habilidade de aprendizado sobre seu ambiente e com isso a capacidade de melhoria de seu desempenho. Isso é feito através de

um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

A rede neural se baseia nos dados para extrair um modelo geral. Portanto, a fase de aprendizado deve ser rigorosa e verdadeira, a fim de serem evitados modelos espúrios. Todo o conhecimento de uma rede neural está armazenado nas sinapses, ou seja, nos pesos atribuídos às conexões entre os neurônios. De 50 a 90% do total de dados devem ser separados para o treinamento da rede neural, dados estes escolhidos aleatoriamente, a fim de que a rede "aprenda" as regras e não "decore" exemplos. O restante dos dados só são apresentados à rede neural na fase de testes, a fim de que ela possa "deduzir" corretamente o interrelacionamento entre os dados.

O algoritmo de treinamento (ou aprendizado) de uma rede, é estabelecido por um conjunto de regras por meio das quais os pesos das conexões são ajustados usando, para o ajuste, um conjunto de exemplos que representa a função a ser aprendida. Esse conjunto de exemplos é geralmente chamado de conjunto de treinamento. Uma RN sempre aprende a partir de um conjunto de treinamento, o aprendizado se reflete na capacidade de generalização que a rede vai exibir, quando usada em novas situações. Nesta fase, seguindo o algoritmo de treinamento escolhido, serão ajustados os pesos das conexões. É importante considerar, nesta fase, alguns aspectos tais como a inicialização da rede, o modo de treinamento e o tempo de treinamento.

Uma boa escolha dos valores iniciais dos pesos da rede pode diminuir o tempo necessário para o treinamento. Normalmente, os valores iniciais dos pesos da rede são números aleatórios uniformemente distribuídos, em um intervalo definido. A escolha errada destes pesos pode levar a uma saturação prematura.

Quanto ao modo de treinamento, na prática é mais utilizado o modo incremental, onde as correções são feitas a cada amostra) devido ao menor armazenamento de dados, além de ser menos suscetível ao problema de mínimos locais, devido à pesquisa de natureza estocástica que realiza. Por outro lado, no modo *batch* (as correções são feitas após uma década) tem-se uma melhor estimativa do vetor gradiente, o que torna o treinamento mais estável. A eficiência relativa dos dois modos de treinamento depende do problema que está sendo tratado.

Quanto ao tempo de treinamento, vários fatores podem influenciar a sua duração, porém sempre será necessário utilizar algum critério de parada. O critério de parada do algoritmo Backpropagation não é bem definido, e geralmente é utilizado um número máximo de ciclos. Mas devem ser considerados a taxa de erro médio por ciclo e a capacidade de generalização da rede. Pode ocorrer que em um determinado instante do treinamento a generalização comece a degenerar, causando o problema de *over training*, ou seja, a rede se especializa no conjunto de dados do treinamento e perde a capacidade de generalização.

O treinamento deve ser interrompido quando a rede apresentar uma boa capacidade de generalização e quando a taxa de erro for suficientemente pequena, ou seja, menor que um erro admissível. Assim, deve-se encontrar um ponto ótimo de parada com erro mínimo e capacidade de generalização máxima.

4.5.4 – Teste ou validação

O próximo passo é o teste (validação) da rede. Durante esta fase, o conjunto de teste é utilizado para determinar a performance da rede, com dados que não foram previamente utilizados. A performance da rede, medida nesta fase, é uma boa indicação de sua performance real.

Devem ser considerados ainda outros testes, como análise do comportamento da rede utilizando entradas especiais e análise dos pesos atuais da rede, pois se existirem valores muito pequenos, as conexões associadas podem ser consideradas insignificantes e assim serem eliminadas (*prunning*). De modo inverso, valores substantivamente maiores que os outros poderiam indicar que houve *over training* da rede.

4.5.5 - Integração

Finalmente, com a rede treinada e avaliada, ela pode ser integrada em um sistema do ambiente operacional da aplicação. Para maior eficiência da solução, este sistema deverá conter facilidades de utilização como interface conveniente e facilidades de aquisição de dados através de planilhas eletrônicas, interfaces com unidades de processamento de sinais, ou arquivos padronizados. Uma boa documentação do sistema e o treinamento de usuários são necessários para o sucesso do mesmo.

Além disso, o sistema deve periodicamente monitorar sua performance e fazer a manutenção da rede quando for necessário ou indicar aos projetistas a necessidade de novo treinamento. Outras melhorias poderão ainda ser sugeridas quando os usuários forem se tornando mais familiares com o sistema, estas sugestões poderão ser muito úteis em novas versões ou em novos produtos.

5 – EFEITOS INTERFERENTES

Sistemas de comunicações via satélite requerem um nível adequado de sinal entre transmissor e receptor, de modo a proporcionar características de robustez, confiabilidade e disponibilidade do sistema.

Para sistemas digitais, a performance do enlace é estabelecida através do BER (Bit Error Rate) de um determinado período de tempo. Este período de tempo é o que oferece as piores condições de propagação sendo, geralmente, entre 0,04% e 0,5% ao longo do ano [18]. Para sistemas analógicos, a razão C/N (potência da portadora pela potência do ruído) na entrada do demodulador, define o limite de disponibilidade do sistema.

Existem muitos fenômenos que degradam o sinal de um sistema de comunicações via satélite, tais como absorção atmosférica (efeito dos gases da atmosfera), atenuação das nuvens (efeito de partículas de gelo e do aerossol), cintilação (efeitos de refração), rotação de Faraday (efeito ionosférico), atenuação por chuva (efeito de partículas de chuva) e despolarização (efeito de chuva e cristais de gelo). A atenuação por chuva é definitivamente, o fenômeno que causa maiores perdas para frequências superiores a 10 GHz. A atenuação e despolarização vêm da interação entre a propagação eletromagnética e de tudo que exista na atmosfera (íons, átomos neutros, moléculas, partículas que contenham água), podendo ser negligenciados em frequências na faixa de 4 GHz [18].

Outro fenômeno bastante interessante é a interferência solar. Este fenômeno acontece nos meses de março e setembro, quando o sol fica alinhado com o satélite e antena terrestre, dentro do lóbulo principal do diagrama de irradiação da antena terrestre. O sol é fonte poderosa de ruído branco e degrada a relação sinal/ruído.

Comumente um enlace de comunicações via satélite é modelado como sendo susceptível a interferências do tipo AWGN.

5.1 - AWGN

O canal de comunicações via satélite, é geralmente modelado como sendo predominantemente um canal AWGN (*Additive White Gaussian Noise*), onde estão presentes fortes atenuações e muitas vezes grandes variações de fase do sinal. O canal AWGN é utilizado em modelamentos matemáticos para caracterizar aqueles canais onde o

tipo de ruído responsável por degradar a comunicação é um ruído branco (atuante em uma larga faixa de frequência) adicionado ao sinal. O ruído branco é um sinal aleatório e tem um modelamento matemático que o considera como possuindo largura de faixa infinita, média nula e correlação nula entre suas amplitudes tomadas a instantes de tempo distintos, ou seja, o valor da amplitude do ruído em um determinado instante independe daquele observado em outro instante de tempo qualquer. O termo gaussiano se deve ao fato desse tipo de ruído possuir uma função densidade de probabilidade gaussiana com média nula, com desvio padrão igual à sua tensão RMS (*Root Mean Square*), e variância igual à potência dissipada em um resistor de 1W.

Para ilustrar tal fato foi gerado no MATLAB um programa capaz de simular uma senóide através de um canal AWG, conforme a Figura 5.1:

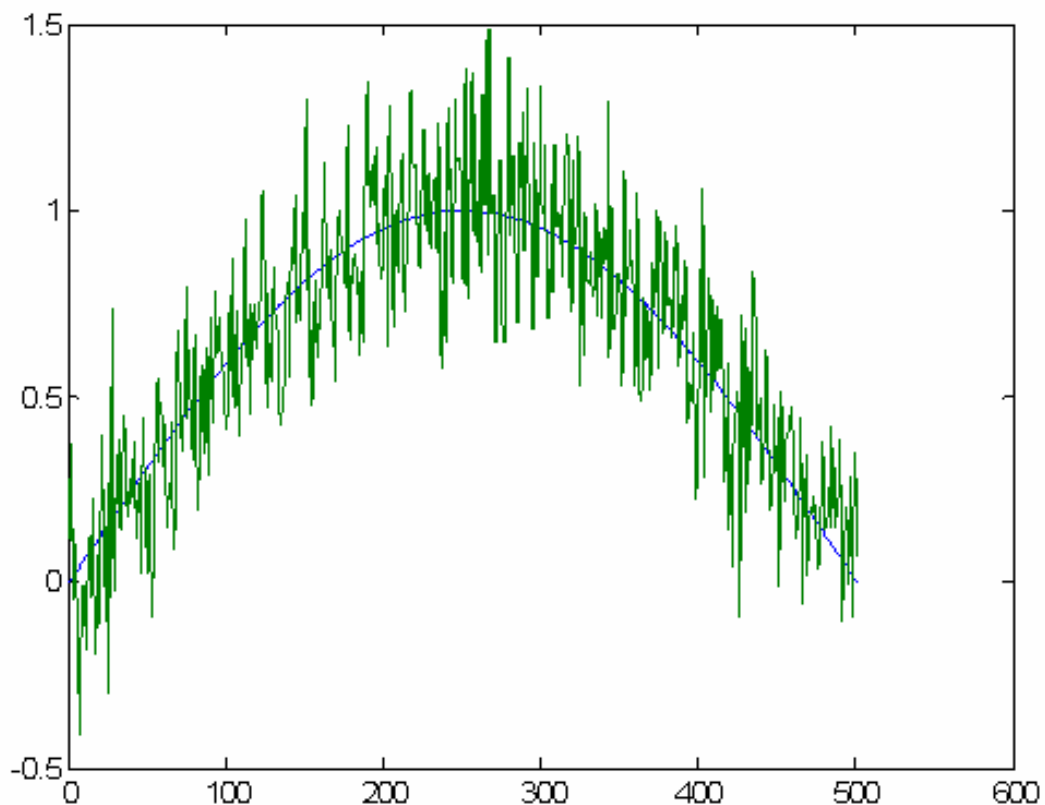


Figura 5.1 – Senóide através de canal AWGN.

5.1.1 – Fundamentação de processos estocásticos [17] e [25]

Distribuição de probabilidade discreta:

- Se uma variável X pode assumir um conjunto discreto de valores X_1, X_2, \dots, X_K , com as probabilidades p_1, p_2, \dots, p_K , respectivamente, sendo $p_1 + p_2 + \dots + p_K = 1$, diz-se que está definida uma distribuição de probabilidade discreta de X . A função $p(X)$ que assume os valores p_1, p_2, \dots, p_K , respectivamente, para $X = X_1, X_2, \dots, X_K$ é denominada função de probabilidade ou função de frequência de X . Como X pode assumir certos valores com dadas probabilidades, ele é frequentemente denominado variável aleatória discreta. A variável aleatória também é conhecida como variável casual ou estocástica.

Distribuição de probabilidade contínua:

- No caso em que a variável X , pode assumir um conjunto contínuo de valores, o polígono de frequência relativa torna-se uma curva contínua, cuja equação é $Y = p(X)$. A área total limitada por essa curva e pelo eixo dos X é igual a 1. A função densidade de probabilidade ou função densidade $p(X)$ define uma distribuição de probabilidade contínua para X . A variável X é então denominada variável aleatória contínua.

Distribuição de frequência:

- Quando se resumem grandes massas de dados brutos, costuma-se frequentemente distribuí-los em classes ou categorias e determinar o número de indivíduos pertencentes a cada uma das classes, denominado frequência da classe. Um arranjo tabular dos dados por classes, juntamente com as frequências correspondentes, é denominado distribuição de frequência ou tabela de frequência. Histogramas e polígonos de frequência são duas representações gráficas de distribuições de frequência

Teoria da correlação:

- A correlação diz respeito ao grau de relação entre as variáveis, ou seja, como as variáveis dependentes estão subordinadas às variáveis correlatas (ou independentes). Se todos os valores das variáveis satisfazem exatamente uma equação, diz-se que elas estão perfeitamente correlacionadas. Se 2 dados honestos são lançados simultaneamente 100 vezes, não há relação entre os pontos correspondentes a cada um deles.

Série temporal:

- Série temporal é um conjunto de observações tomadas em tempos determinados, comumente em intervalos iguais.

Variável aleatória gaussiana:

- A variável aleatória X é chamada de Gaussiana se sua função densidade tem a forma mostrada pela Figura 5.2:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \cdot e^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}} \quad (5.1)$$

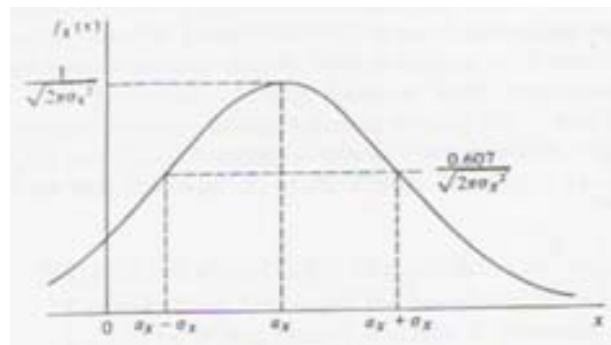


Figura 5.2 – Função densidade gaussiana.

Processo estocástico:

- O *stream* de bits de uma mensagem é randômico. Esse sinal desejável é acompanhado por formas de onda indesejáveis também randômicos (ruído). Portanto, a performance dos sistemas é limitada pela capacidade de reconhecer os efeitos negativos do ruído. Um processo estocástico representa uma variável aleatória para um dado momento. Desta forma, a saída depende da variável aleatória de entrada e do tempo.

5.2 – ATENUAÇÃO CAUSADA POR CHUVA

A atenuação causada por chuva, pode ser quantificada de modo muito simples através da expressão abaixo:

$$A(t) = \Pr(\text{clearsky}) - \Pr \quad (5.2)$$

onde:

- $A(t)$: é a atenuação total em dado instante t , em dB;
- P_r (*clearsky*): é a potência recebida em céu claro em dado instante t , em dB;
- P_r : é a potência efetivamente recebida em dado instante t , em dB.

Em *links* satélite, as atenuações causadas por chuva são as mais importantes para adequar uma margem segura de operação, principalmente para frequências superiores a 10 GHz.

Devem ser seguidos 3 passos para o correto dimensionamento do enlace:

- passo 1: determinar uma correta razão de chuva sobre um período de interesse percentual;
- passo 2: calcular uma atenuação específica do sinal para esta chuva em dB/km;
- passo 3: encontrar um comprimento efetivo sobre o qual a atenuação age.

Todos os passos exigem um modelamento complexo e cuidadoso, que depende de um trabalho prévio estatístico das condições atmosféricas reinantes em determinado local. O terceiro passo é bastante interessante, uma vez que existem dois tipos de formação de nuvens, que causam efeitos diversos. A nuvem estratiforme é gerada por camada de nuvem com grande espalhamento de gelo e água, abrangendo grande área, com razão de precipitação de chuva inferior a 10 mm/hora. A nuvem convectiva é gerada por correntes de ar verticais poderosas, que geram chuvas concentradas e de maior intensidade.

A Figura 5.3 nos mostra estas formações estratiformes e convectivas.

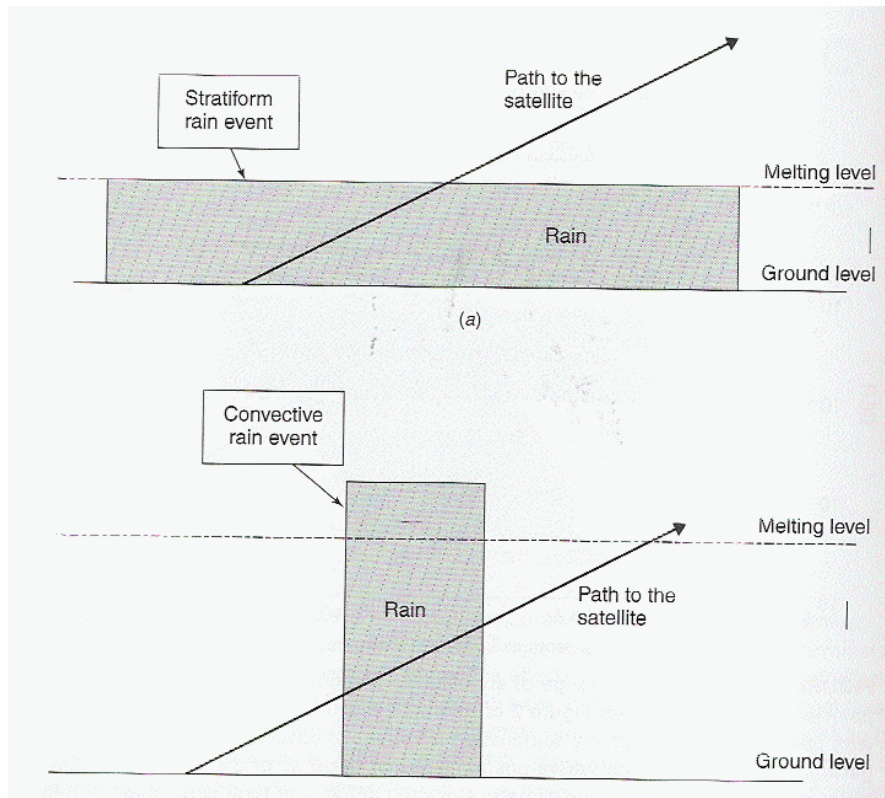


Figura 5.3 – Nuvens estratiformes (a) e convectivas (b) [18].

A estimativa da atenuação por chuva, pode ser obtida através das expressões:

$$A = \gamma_r \cdot L_{\text{eff}} \text{ (dB)} \quad (5.3)$$

- A: é a atenuação total;
- (γ_r) : é a atenuação específica;
- L_{eff} : é o comprimento efetivo percorrido pela onda em camada chuvosa, em km.

Existem diversos modelos preditivos que determinam a atenuação específica tais como o modelo Crane, modelo SAM (Simple Attenuation Model) [18], o modelo DAH (Dissanayake, Haidara, Allnut) e diversos outros modelos publicados pela ITU-R [10].

5.2.1 – Modelo da ITU-R baseado no modelo DAH

O modelo da ITU-R baseado no modelo DAH é, provavelmente, o modelo mais acurado atualmente usado, onde a atenuação específica (γ_r) em 1 km é dada por:

$$\gamma_r = k \cdot (R_{0,01})^\alpha \text{ (dB/km)} \quad (5.4)$$

onde:

- k e α dependem da frequência conforme Tabela 5.1;
- $R_{0,01}$ é o índice pluviométrico relativo a 0,01% de determinado período de tempo, dado em mm/h. O parâmetro $R_{0,01}$ ou alguns outros índices que variam de 10% a 0,001% são obtidos pelo conjunto da Figura 5.4 e Tabela 5.2.

Tabela 5.1 - Parâmetros k e α [18].

Frequency (GHz)	k_H	k_V	α_H	α_V
4	0.000650	0.000591	1.121	1.075
6	0.00175	0.00155	1.308	1.265
8	0.00454	0.00395	1.327	1.310
10	0.0101	0.00887	1.276	1.264
12	0.0188	0.0168	1.217	1.200
20	0.0751	0.0691	1.099	1.065
30	0.187	0.167	1.021	1.000
40	0.350	0.310	0.939	0.929
50	0.536	0.479	0.873	0.868

NOTES:

- (1) The suffices V and H refer to vertical and horizontal polarization, respectively
- (2) Values of k and α at frequencies other than those in the table can be obtained by interpolation using a logarithmic scale for frequency, a logarithmic scale for k , and a linear scale for α .
- (3) Values have been tested and found to be accurate up to a frequency of 40 GHz; values between 40 and 50 GHz are expected to be accurate but have not yet been tested.
- (4) For linear and circular polarization, and for all path geometries, the coefficients in equation (8.13) can be calculated using the values in the above table and the following equations²⁵

$$k = [k_H + k_V + (k_H - k_V)\cos^2\theta \cos 2\tau]/2$$

$$\alpha = [k_H\alpha_H + k_V\alpha_V + (k_H\alpha_H - k_V\alpha_V)\cos^2\theta \cos 2\tau]/2k$$

where θ is the path elevation angle and τ is the polarization tilt angle relative to the horizontal ($\tau = 45^\circ$ for circular polarization).

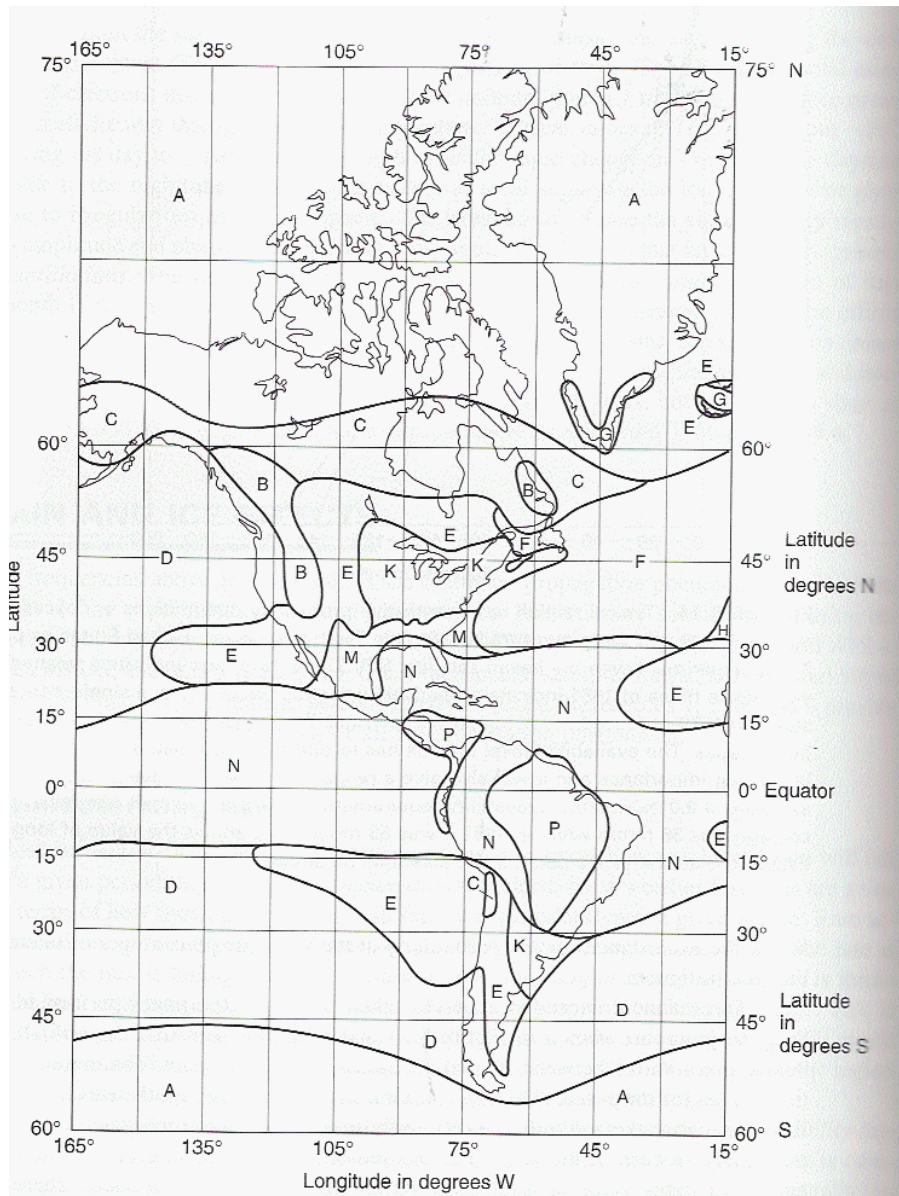


Figura 5.4 – Mapa de distribuição climatológica [18].

Tabela 5.2 – Índices pluviométricos por região [18].

Percentage of Time (%)	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q
10	<0.1	0.5	0.7	2.1	0.6	1.7	3	2	8	1.5	2	4	5	12	24
0.3	0.8	2	2.8	4.5	2.4	4.5	7	4	13	4.2	7	11	15	34	49
0.1	2	3	5	8	6	8	12	10	20	12	15	22	35	65	72
0.03	5	6	9	13	12	15	20	18	28	23	33	40	65	105	96
0.01	8	12	15	19	22	28	30	32	35	42	60	63	95	145	115
0.003	14	21	26	29	41	54	45	55	45	70	105	95	140	200	142
0.001	22	32	42	42	70	78	65	83	55	100	150	120	180	250	170

O cálculo da atenuação total expressa pela equação (5.3), requer ainda a definição do valor de L_{eff} (comprimento efetivo percorrido em camada chuvosa em km), exemplificado pela Figura 5.5.

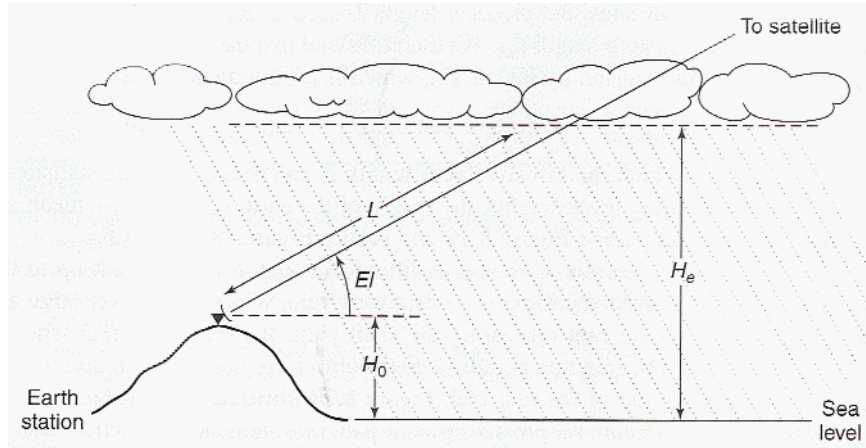


Figura 5.5 – Geometria de cálculo do L_{eff} [18].

A figura 5.6, indica quais os parâmetros apontados pela ITU-R como sendo importantes para a determinação de L_{eff} (comprimento efetivo percorrido em camada chuvosa em km), para uma chuva estratiforme.

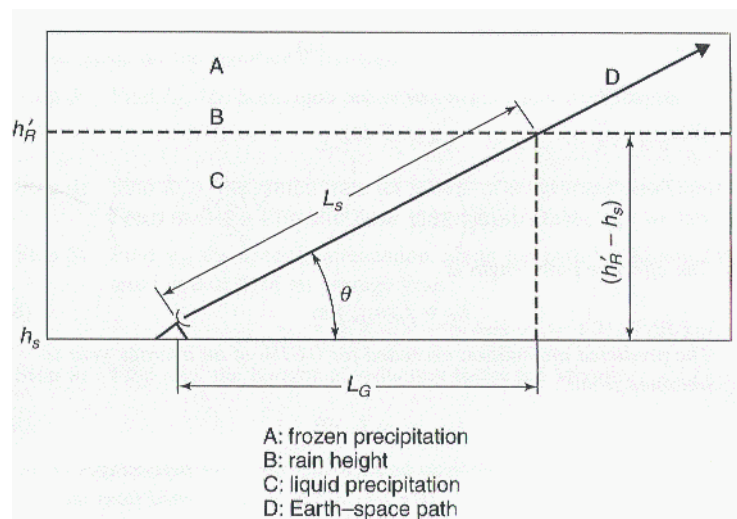


Figura 5.6 – Cálculo do L_{eff} [18].

Para o cálculo do L_{eff} , devemos levantar os seguintes valores:

- L_S : comprimento inclinado e L_G : comprimento horizontal;
- θ : elevação (em graus) e Φ : latitude (em graus);
- h'_R : altitude da base da nuvem chuvosa;
- $r_{0.01}$: fator de redução horizontal;
- $v_{0.01}$: fator de ajuste vertical para 0.01 % do tempo;

para $\theta \geq 5^\circ$

$$L_S = \frac{h'_R - h_S}{\sin \theta} km \quad (5.5)$$

$$L_G = L_S \cos(\theta) km \quad (5.6)$$

$$r_{0.01} = \frac{1}{1 + 0.78 \sqrt{\frac{L_G \cdot \gamma_R}{f}} - 0.38(1 - e^{-2L_G})} \quad (5.7)$$

$$\zeta = \tan^{-1} \left(\frac{h'_R - h_S}{L_G r_{0.01}} \right) \text{deg} \quad (5.8)$$

para $\zeta > \theta$

$$L_R = \frac{L_G r_{0.01}}{\cos \theta} km \quad (5.9)$$

senão

$$L_R = \frac{(h'_R - h_S)}{\sin \theta} km \quad (5.10)$$

se $|\Phi| < 36$

$$\chi = 36 - |\phi| \text{deg} \quad (5.11)$$

senão

$$\chi = 0 \text{deg} \quad (5.12)$$

$$v_{0.01} = \frac{1}{1 + \sqrt{\sin \theta} (31(1 - e^{-(\theta/(1+\chi))}) \frac{\sqrt{L_R \gamma_R}}{f^2} - 0.45)} \quad (5.13)$$

finalmente

$$L_{eff} = L_R \cdot v_{0.01} \quad (5.14)$$

5.2.2 – Modelo SAM

O modelo SAM (Simple Attenuation Model) foi desenvolvido por Warren Stutzman e Keith Dishman. É um modelo simplificado que permite a obtenção rápida de valores aproximados do L_{eff} (comprimento efetivo percorrido em camada chuvosa em km) para qualquer índice pluviométrico e qualquer frequência entre 2,9 GHz e 180 GHz. Como em especificado pela equação (5.4), a atenuação específica (γ_r) pode ser determinada por:

$$\gamma_r = k \cdot R^\alpha \text{ (dB/km)} \quad (5.4)$$

$$k = 4,21 \cdot 10^{-5} \cdot f^{2,42} \quad 2,9 \leq f \leq 54 \text{ GHz} \quad (5.15)$$

$$k = 4,09 \cdot 10^{-2} \cdot f^{0,669} \quad 54 \leq f \leq 180 \text{ GHz} \quad (5.16)$$

$$\alpha = 1,41 \cdot f^{-0,0779} \quad 8,5 \leq f \leq 25 \text{ GHz} \quad (5.17)$$

$$\alpha = 2,63 \cdot f^{-0,272} \quad 25 \leq f \leq 164 \text{ GHz} \quad (5.18)$$

Para a determinação de R, que é o índice pluviométrico relativo a um determinado período de tempo dado em mm/h, faz-se uso da Figura 5.4 e da Tabela 5.2, assim como foi visto no modelo da ITU-R.

Para a determinação da atenuação total conforme a equação (5.3), há necessidade da determinação de L_{eff} (comprimento efetivo).

$$A = \gamma_r \cdot L_{eff} \text{ (dB)} \quad (5.3)$$

$$\text{para } R \leq 10 \text{ mm/h, } L = L_{eff} \quad L = \frac{He - Ho}{\sin(EL)} \text{ km} \quad (5.19)$$

- He é a altitude da base da nuvem chuvosa em km;
- Ho é a altitude da estação em km;
- EL é a elevação da antena da estação em graus.

Existe ainda um fator de correção de He para as latitudes:

- para latitudes $\leq 30^\circ$ He = 4,8 km

- para latitudes $\gt 30^\circ$ $H_e = 7,8 - 0,1 |\text{latitude}| \text{ km}$

Para $R \gt 10 \text{ mm/h}$, calcular a equação 5.19 com as seguintes alterações:

- para latitudes $\leq 30^\circ$ $H_e = 4,8 \text{ km} + \log_{10} (R/10) \text{ km}$
- para latitudes $\gt 30^\circ$ $H_e = 7,8 - 0,1 |\text{latitude}| + \log_{10} (R/10) \text{ km}$

Finalmente, para $R \gt 10 \text{ mm/h}$, o comprimento efetivo (L_{eff}) será dado pela equação (5.20), conforme ilustrado pela Figura 5.7:

$$L_{eff} = \left\{ \frac{1 - \exp \left[-\frac{1}{22} \cdot \alpha \cdot \log_e \left(\frac{R}{10} \right) \cdot L \cdot \cos(El) \right]}{\frac{1}{22} \cdot \alpha \cdot \log_e \left(\frac{R}{10} \right) \cdot \cos(El)} \right\} \text{ km} \quad (5.20)$$

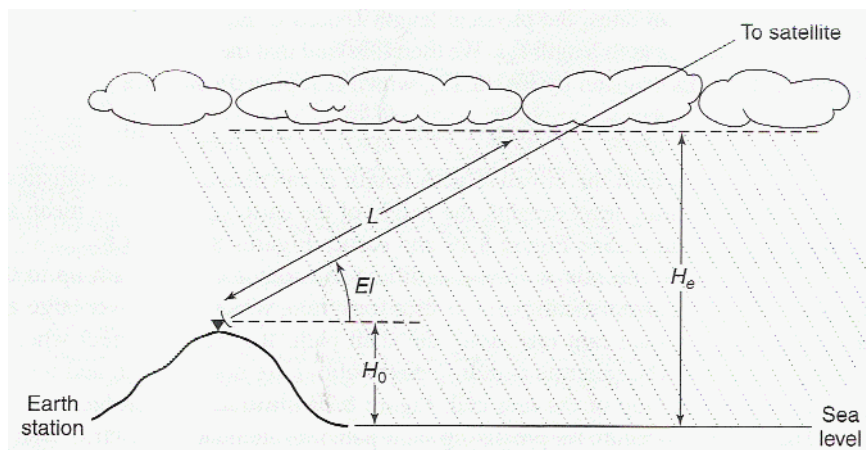


Figura 5.7 – Geometria do L_{eff} [18].

6 – LEVANTAMENTO DE DADOS

O objetivo deste capítulo é registrar a coleta de dados de interesse para uma simulação computacional mais completa. Os fenômenos (efeitos interferentes) considerados de maior interesse, são o efeito AWGN e a atenuação, em especial a atenuação causada por chuva [5].

Para tanto, foi montado um *loop* entre a estação *hub* de Brasília e o satélite Brasilsat B4, modo a se estabelecer qual o nível de Eb/No ao longo de um período de tempo. Observe que o satélite Brasilsat B4 substituiu recentemente o satélite Brasilsat B3, assumindo a posição orbital do B3.

Foram levantadas também informações meteorológicas, a fim de estabelecer parâmetros úteis para o cálculo de atenuação.

6.1 – LEVANTAMENTO DE DADOS COM MODELAGEM AWGN

Inicialmente foi feita consulta ao site da Star One [26], com a finalidade de se averiguar um período com forte interferência solar conforme mostrado pela Figura 6.1. Como este fenômeno é considerado predominantemente AWGN, a intenção inicial foi a de, através de um *loop* entre a estação *hub* de Brasília e o satélite Brasilsat B4, determinar qual o índice BER em um dado período de tempo.

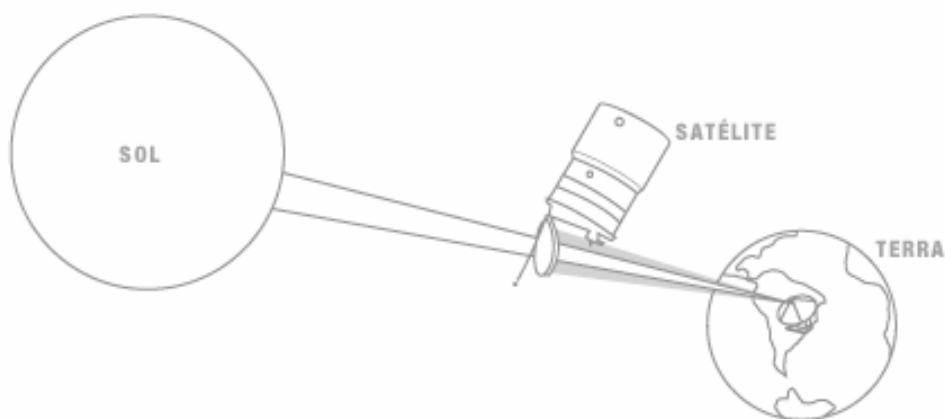


Figura 6.1 – Posição de interferência solar [26].

Os satélites da série Brasilsat trabalham na Banda C, de 5.850 MHz a 6.425 MHz na subida e de 3.625 MHz a 4.200 MHz na descida.

O site [26] permite visualizar a previsão de interferência solar nas estações terrenas. Estas interferências são mais acentuadas nos meses de março e setembro. Conforme já comentado, o satélite B3 cedeu sua posição para o satélite B4 a uma longitude de 84° W.

Os parâmetros setados foram para o satélite B4, localidade de Brasília (e automaticamente a página completa a posição do satélite e da estação) e antena com diâmetro de 6,2 m, conforme podemos ver na Tabela 6.1.

Tabela 6.1 – Previsão de interferência para março [26].



Star One
uma Empresa Brasileira

CÁLCULO DA INTERFERÊNCIA SOLAR NAS ESTAÇÕES TERRENAS DO BRASIL POR CIDADE

PREVISÃO DE INTERFERÊNCIA SOLAR - BRASIL

Satélite: Brasilsat B4

Longitude do satélite +O: 84

Localidade: Brasília (DF)

Latitude da localidade +N: -15.77972

Longitude da localidade +O: 47.92972

Azimute do satélite: 290.449

Elevação do satélite: 44.919

Diâmetro da Antena (m): 6.2

Banda: C

Frequência (GHz): 3.825

Ângulo de Interferência (°): 1.989

Semestre: Primeiro

Ano: 2009

Fluxo Solar: 93.7

INTERFERÊNCIAS PREVISTAS

Data Interf. dd/mm/aaaa	Início GMT hh:mm:ss	Fim GMT hh:mm:ss	Duração mm:ss	Início GMT-3h hh:mm:ss	Fim GMT-3h hh:mm:ss	Degradação máxima em C/N (dB)
22/03/2009	18:01:22	18:08:37	07:15	15:01:22	15:08:37	2.3
23/03/2009	17:58:57	18:10:27	11:30	14:58:57	15:10:27	3.7
24/03/2009	17:57:30	18:11:15	13:45	14:57:30	15:11:15	5.9
25/03/2009	17:56:30	18:11:40	15:10	14:56:30	15:11:40	13.4
26/03/2009	17:55:53	18:11:38	15:45	14:55:53	15:11:38	17.5
27/03/2009	17:55:36	18:11:21	15:45	14:55:36	15:11:21	17.4
28/03/2009	17:55:35	18:10:45	15:10	14:55:35	15:10:45	13.2
29/03/2009	17:55:59	18:09:44	13:45	14:55:59	15:09:44	5.7
30/03/2009	17:56:47	18:08:17	11:30	14:56:47	15:08:17	3.7
31/03/2009	17:58:35	18:05:55	07:20	14:58:35	15:05:55	2.3

De posse destes dados, no dia 26/03/2009, fechou-se um *loop* entre a *hub* e o satélite, de modo a estabelecer uma emissão de bits de *uplink* que foram verificados após o *downlink*.

O equipamento utilizado na medição foi o TSW200E1 [29], um analisador de comunicação digital, utilizado para manutenção e instalação de linhas de comunicação de dados síncronas e assíncronas, e que permite testar *links* de dados em velocidades que variam de 1200 bps até 10 Mbps, com ou sem estrutura de quadros.

O equipamento foi conectado na interface digital do modem e configurado para enviar 64 Kbps. Isto permitiu o levantamento da integridade dos dados na sua chegada, e conseqüente taxa BER. Os dados obtidos estão na Tabela 6.2.

Tabela 6.2 – Dados obtidos pelo analisador.

<u>hora local</u>	<u>bits errados</u>	<u>BER FEC3/4</u>	<u>hora local</u>	<u>bits errados</u>	<u>BER FEC3/4</u>
15h00	0	0	15h10	63	$9,84 \cdot 10^{-4}$
15h01	0	0	15h11	47	$7,34 \cdot 10^{-4}$
15h02	0	0	15h12	0	0
15h03	0	0	15h13	0	0
15h04	41	$6,40 \cdot 10^{-4}$	15h14	0	0
15h05	5.420	0,084	15h15	0	0
15h06	6	$9,37 \cdot 10^{-5}$	15h16	0	0
15h07	0	0	15h17	0	0
15h08	4	$6,25 \cdot 10^{-5}$	15h18	0	0
15h09	4.716	0,073	15h19	7	$1,09 \cdot 10^{-4}$

6.1.1 – Análise imediata dos dados obtidos

Observa-se, pelos dados obtidos através do analisador, que os momentos críticos (abaixo do *threshold*) foram 15h05 e 15h09. Isto foi confirmado pelas indicações do sistema TELESAT, que começaram a alarmar indicando queda de diversos links, principalmente nestes dois momentos.

Considerando que o ano com 365 dias tem 525.600 minutos e que, portanto, 1 minuto de indisponibilidade do sistema corresponde a $1,90 \cdot 10^{-6}$ % de indisponibilidade.

Considerando que o mês de março, com 31 dias tem 44.640 minutos e que, portanto, 1 minuto de indisponibilidade do sistema corresponde a $2,24 \cdot 10^{-5}$ % de indisponibilidade.

Considerando os dados da Tabela 2.6 e que os requisitos de qualidade são:

- BER melhor que 10^{-7} durante mais que 98% do tempo (no pior mês do ano).
- BER melhor que 10^{-3} durante mais que 99,8% do tempo (no pior mês do ano).

Chegou-se a conclusão que os dados obtidos são compatíveis com os requisitos de projeto considerados no Capítulo 2, mesmo extrapolando os períodos de interferências para todo o ano.

Contudo, para efeito de simulação de um cálculo de enlace, onde se pretende demonstrar a validade do uso de uma rede neural na correção de erros de transmissão em sistemas de comunicações via satélite, o valor de BER obtido de $8,05 \cdot 10^{-3}$ durante o período de interferência solar, modelados como um canal AWGN, não é considerado satisfatório.

6.2 – NOVO LEVANTAMENTO COM MODELAGEM AWGN

Uma vez que os dados obtidos em 6.1 não permitem uma adequada simulação, em virtude de suas informações espúrias, foram feitas novas medidas em um novo *loop* entre a *hub* e o satélite B4, com a mesma configuração anterior. O equipamento [29] foi acionado dia 21/05/2009 às 13h25 horário local de Brasília, e ficou colhendo informações até o dia 22/05/2009 às 11h40 horário local, portanto, 22h15 de coleta de informações.

O BER medido pelo equipamento durante este período de 22h15 de coleta de informações, foi de $1,8 \cdot 10^{-7}$. Do histograma apresentado pelo analisador de comunicação TSW200E1, foram extraídos, a título de exemplificação, apenas os primeiros 615 minutos, reproduzidos na Tabela 6.3, indicando que os erros são produzidos por uma seqüência de *bursts* que ocorrem com certa previsibilidade.

Tabela 6.3 – Novos dados parciais obtidos pelo analisador.

$T_0 = 13h25$	BER	$T_0 = 13h25$	BER
+ 3 min	23	+ 341 min	14
+ 36 min	13	+ 352 min	18
+ 112 min	9	+ 418 min	29
+ 156 min	31	+ 440 min	24
+ 189 min	24	+ 451 min	16
+ 221 min	22	+ 462 min	20
+ 254 min	19	+ 527 min	6
+ 276 min	8	+ 549 min	7
+ 298 min	11	+ 615 min	4

De acordo com o BER obtido de $1,8 \cdot 10^{-7}$ e os dados do fabricante disponíveis na Tabela 2.6, novamente reproduzida abaixo, vemos que, com um FEC de 3/4, temos um E_b/N_0 de 9 dB.

Tabela 2.6 – Correspondência E_b/N_0 versus BER de acordo com FEC [1].

	Composite E_b/N_0		
	FEC 1/2	FEC 3/4	FEC 7/8
Threshold	5.0 dB	3.5 dB	2.5 dB
BER = 1×10^{-3}	4.8 dB	5.8 dB	6.3 dB
BER = 1×10^{-4}	5.5 dB	6.6 dB	7.1 dB
BER = 1×10^{-5}	6.3 dB	7.3 dB	7.8 dB
BER = 1×10^{-6}	6.9 dB	8.2 dB	8.5 dB
BER = 1×10^{-7}	7.7 dB	9.0 dB	9.2 dB

Portanto, este parâmetro de E_b/N_0 de 9 dB é que será utilizado na simulação de um canal AWGN.

6.3 – LEVANTAMENTO DE DADOS METEOROLÓGICOS

A fim de complementar o que foi obtido com modelagem AWGN no Capítulo 6.1, foram levantadas também, para aquele lapso temporal em específico, as informações meteorológicas.

Agora a intenção é levantar as informações meteorológicas relativas ao dia 26/03/2009 entre as 15h00 local e 15h30 local, período da interferência solar analisado no Capítulo 6.1.

Para tanto foram colhidas informações junto à EMS (Estação Meteorológica de Superfície) de Brasília. Especificamente, procurou-se a precipitação sobre a vertical do aeroporto de Brasília e, uma vez que a antena do sistema TELESAT encontra-se ao lado do aeroporto, considera-se que os dados são válidos.


A média de precipitação observada no dia 26/03/2009 está na Tabela 6.4. Observe-se que o horário das tabelas está em GMT (Greenwich Mean Time) ou UTC (Universal Time Coordinated), portanto para se obter o horário local basta diminuir 3 horas.

Tabela 6.4 – Precipitação real no dia 26/03/2009.

DADOS DE BRASÍLIA DO DIA 26 MAR 2009		
HORÁRIO (GMT)	PRECIPITAÇÃO (mm)	TOTAL DE NUVENS
1700	2,6	OVC
1800	1,6	BKN
1900	0,3	BKN

TOTAL DE PRECIPITAÇÃO: 4,5mm

LEGENDA:	FEW = 1 a 2 oitavos
	SCT = 3 a 4 oitavos
	BKN = 5 a 7 oitavos
	OVC = 8 oitavos
	T < 0,1mm



A linha de interesse, é aquela que marca o horário de 19h00 GMT, pois ela reflete a média apurada das 18h00 às 19h00 GMT, ou seja, das 15h00 às 16h00 local. Além disso, é indicado que a formação observada era do tipo BKN, de *broken*. Este tipo de formação é caracterizada por uma cobertura de 5/8 a 7/8 do céu. Infelizmente, não temos toda a

informação tridimensional da formação meteorológica, uma vez que não há informação da espessura da camada.

Além das informações de precipitação, foram pesquisadas as imagens das formações meteorológicas reinantes às 18h15 GMT, objetivando o cálculo de atenuação, conforme as Figuras 6.2 e 6.3. Observe-se que, na medição comentada no Capítulo 6.1, estão embutidos, além dos efeitos da interferência solar, os efeitos de atenuação causados pela chuva.

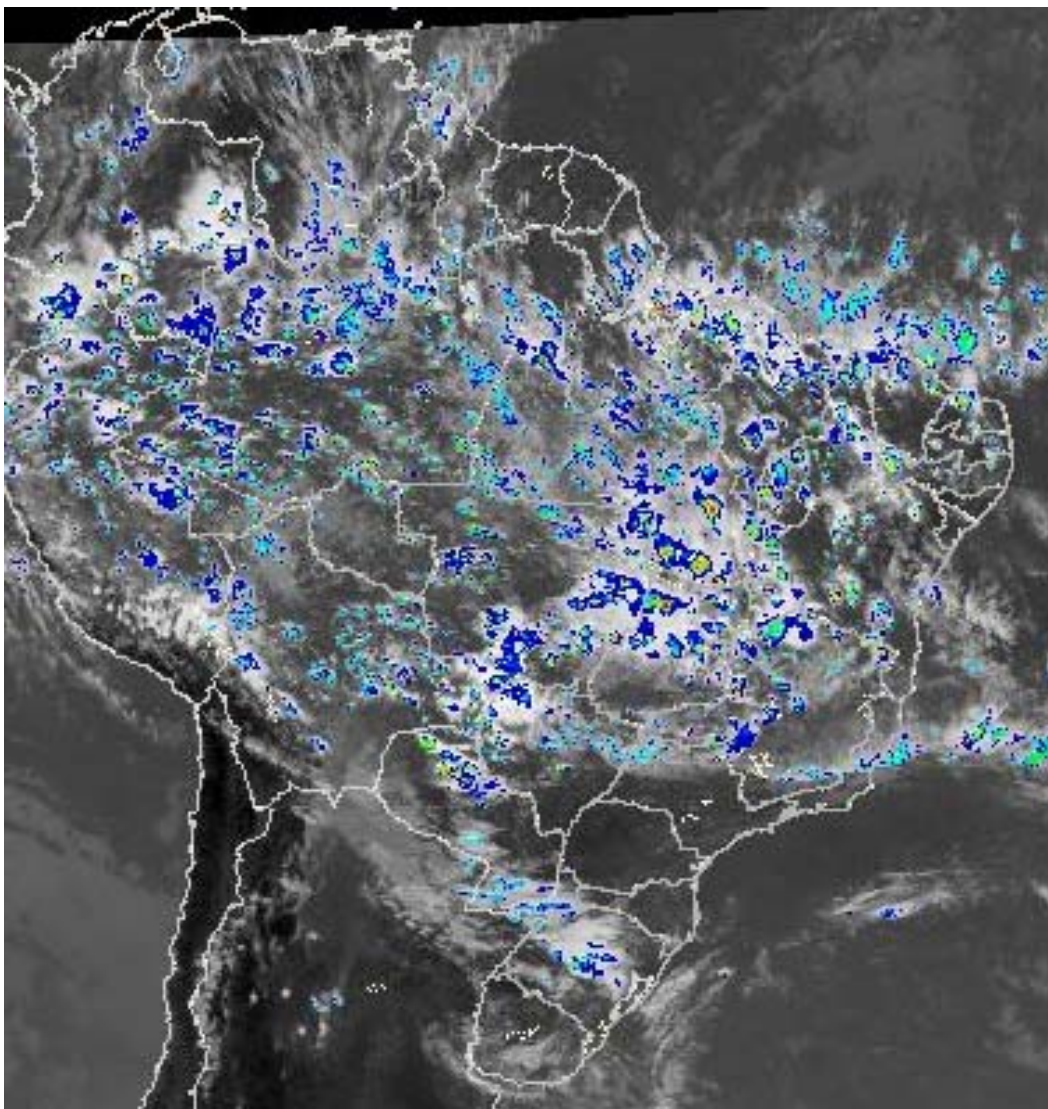


Figura 6.2 – Imagem satélite para o dia 26/03/2009 [3].

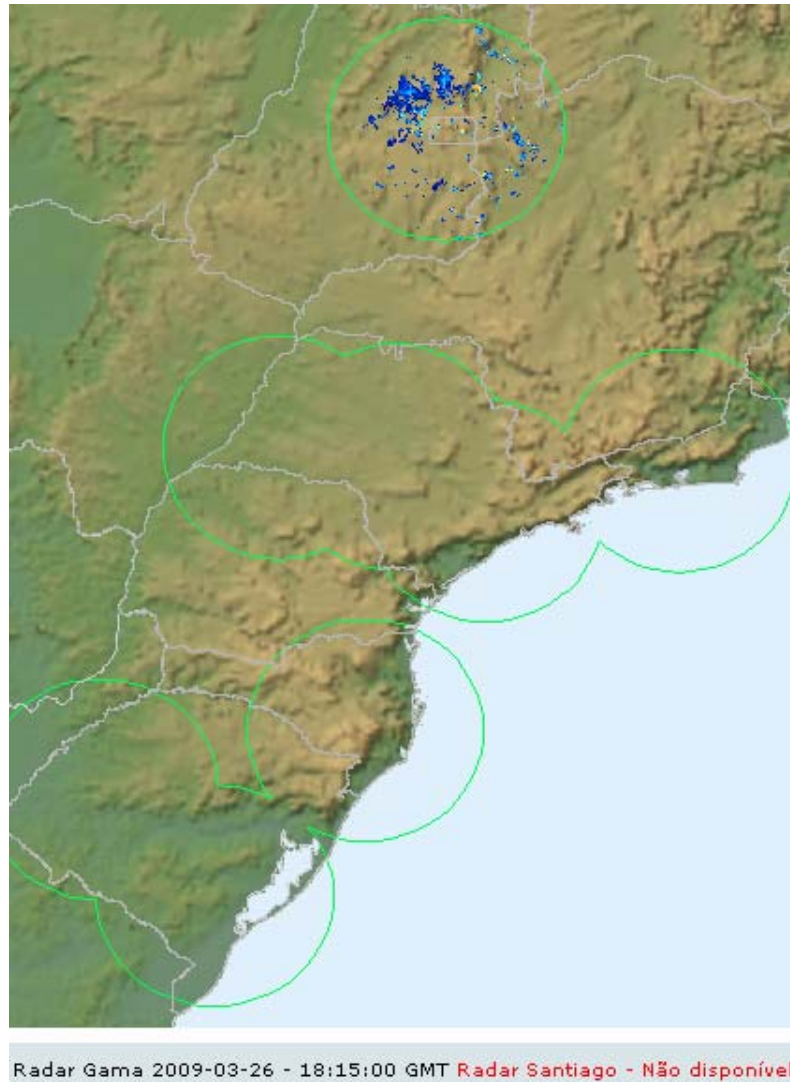


Figura 6.3 – Imagem do radar meteorológico do Gama [3].

As informações visuais, indicam que a ocorrência das formações é bastante oportuna, pois vamos admitir que ocorre justamente na linha imaginária entre o satélite, que está na longitude 84° W e a estação *hub* de Brasília, localizada a uma latitude de $15^{\circ}51'$ S e longitude de $47^{\circ}54'$ W.

Com base nestas informações, é possível estabelecer um modelo de atenuação causado por chuvas baseado no modelo SAM, visto no Capítulo 5.2.2.

Como em (5.4), a atenuação específica (γ_r) pode ser determinada por:

$$\gamma_r = k \cdot R^{\alpha} \text{ (dB/km)} \quad (5.4)$$

$$k = 4,21 \cdot 10^{-5} \cdot f^{2,42} \quad 2,9 \leq f \leq 54 \text{ GHz} \quad (5.15)$$

$$\alpha = 1,41.f^{-0,0779} \quad 8,5 \leq f \leq 25 \text{ GHz} \quad (5.17)$$

Para a determinação de R, que é o índice pluviométrico relativo a um determinado período de tempo dado em mm/h, faz-se uso da Figura 5.4 e da Tabela 5.2, assim como foi visto no modelo da ITU-R. Portanto, considerando $f = 6 \text{ GHz}$, apesar de estar fora da faixa da expressão (5.17) temos:

- $k = 4,21.10^{-5}.f^{2,42} = 3,21 \cdot 10^{-3}$
- $R_{0,1} = 65 \text{ mm/h}$
- $\alpha = 1,41.f^{-0,0779} = 1,226$
- $\gamma_r = 0,53 \text{ dB/km}$

Para a determinação da atenuação total conforme a equação (5.3), há necessidade da determinação de L_{eff} (comprimento efetivo).

$$A = \gamma_r \cdot L_{eff} \text{ (dB)} \quad (5.3)$$

para $R > 10 \text{ mm/h}$

$$L = \frac{H_e - H_o}{\sin(EL)} \text{ km} \quad (5.19)$$

- H_e é a altitude da base da nuvem chuvosa em km;
- $H_o = 1,1 \text{ km}$ (altitude da estação em km);
- $EL = 45^\circ$ (elevação da antena da estação em graus);
- para latitudes $\leq 30^\circ$ $H_e = 4,8 \text{ km} + \log_{10}(R/10) \text{ km}$

$$L = \frac{(4,8 + \log_{10}(65/10)) - 1,1}{\sin(45^\circ)} = \frac{(4,8 + 0,81) - 1,1}{0,707} = 6,37 \text{ km} \quad (6.1)$$

Finalmente, para $R > 10 \text{ mm/h}$, o comprimento efetivo (L_{eff}) será dado por:

$$L_{eff} = \left\{ \frac{1 - \exp\left[-\frac{1}{22} \cdot \alpha \cdot \log_e\left(\frac{R}{10}\right) \cdot L \cdot \cos(EL)\right]}{\frac{1}{22} \cdot \alpha \cdot \log_e\left(\frac{R}{10}\right) \cdot \cos(EL)} \right\} \text{ km} \quad (5.20)$$

$$L_{eff} = \left\{ \frac{1 - \exp[-0,045.1,226.1,87.6,37.0,707]}{0,045.1,226.1,87.0,707} \right\} = 5,09km \quad (6.2)$$

A atenuação total conforme dada pela equação (5.3), é igual a:

$$A = \gamma_r \cdot L_{eff} \text{ (dB)} = 0,53 \text{ dB/km} \cdot 5,09 \text{ km} = 2,69 \text{ dB} \quad (5.3)$$

6.3.1 – Análise imediata dos dados obtidos

Cabe observar, que as informações meteorológicas obtidas em 6.3 através de imagens e índices pluviométricos, foram exclusivamente para determinar se, naquele momento em particular, haveria alguma formação meteorológica.

A atenuação causada por chuva obtida de 2,69 dB, foi alcançada com a utilização do modelo SAM, sendo este valor a ser utilizado em nossa simulação.

6.4 – NOVO LEVANTAMENTO DE DADOS METEOROLÓGICOS

A exemplo do que foi feito no Capítulo 6.3, estaremos agora levantando as reais condições meteorológicas existentes quando das medições descritas no Capítulo 6.2 para um canal AWGN.

Na ocasião desta segunda medição, não foram observadas formações adversas ao longo do período de 13h25 (horário local de Brasília) do dia 21/05/2009, às 11h40 (horário local de Brasília) do dia 22/05/2009.

Portanto, as medições constantes do Capítulo 6.2, foram feitas em condições ideais de céu claro e sem precipitações, conforme mostrado pela Figura 6.4, que destaca o horário das 20h00 UTC do dia 21/05/2009.

Convém destacar novamente, que estas informações de imagens estão sendo utilizadas principalmente para delimitar os fatores que possivelmente estão a influenciar nas medições realizadas para se estabelecer um índice BER de referência.



Figura 6.4 – Imagem satélite das 20h00 UTC [21].

7 - ESTUDO DE CASO

7.1 – CONSIDERAÇÕES INICIAIS

Mediante a intenção de (conforme exposto no Capítulo 1.1 – MOTIVAÇÃO) utilizar redes neurais na correção de erros de transmissão em sistemas de comunicação via satélite, através da construção de uma rede neural capaz de substituir com eficiência, técnicas convencionais de demodulação, optou-se por dividir a tarefa em 3 módulos interdependentes.

O primeiro módulo é responsável por gerar um campo amostral pseudo-randômico, que será usado pelo segundo módulo, para o treinamento supervisionado da rede. Por fim, o terceiro módulo gera um novo campo amostral pseudo-randômico, para validação da rede neural setada pelo segundo módulo.

Estudos preliminares evidenciaram que existe uma correção automática de erros na função `pskdemod` da biblioteca do MATLAB. Portanto, a utilização indiscriminada desta função não permitiria uma correta visualização dos efeitos gerados pela rede neural.

Desta forma, a função `pskdemod` só é utilizada, quando o erro que a rede neural gera na recuperação do sinal se encontra após a 4^a casa decimal (do número complexo). Nesse momento é provocado um arredondamento para valores inteiros, prontos para serem aplicados na função `pskdemod`.

O objetivo é estudar apenas dois tipos de efeitos interferentes:

- interferência do tipo AWGN;
- atenuação causada por chuva.

7.1.1 - A escolha da arquitetura

A arquitetura escolhida foi o perceptron, uma vez que é uma estrutura reconhecidamente eficiente para a resolução de problemas mais complexos. Antes da escolha do número de neurônios necessários, foi necessária reflexão de, qual a topologia que melhor representaria o problema e sua solução.

Assim, foram adotados 2 neurônios na camada de entrada, onde o primeiro neurônio representa a parte real do símbolo modulado e o segundo neurônio representa a parte imaginária do símbolo modulado. Similarmente foram adotados 2 neurônios na camada de saída, onde o primeiro neurônio representa a parte real do símbolo modulado e o segundo neurônio representa a parte imaginária do símbolo modulado. Na camada escondida foram adotados inicialmente 8 neurônios, conforme mostrado na Figura 7.1.

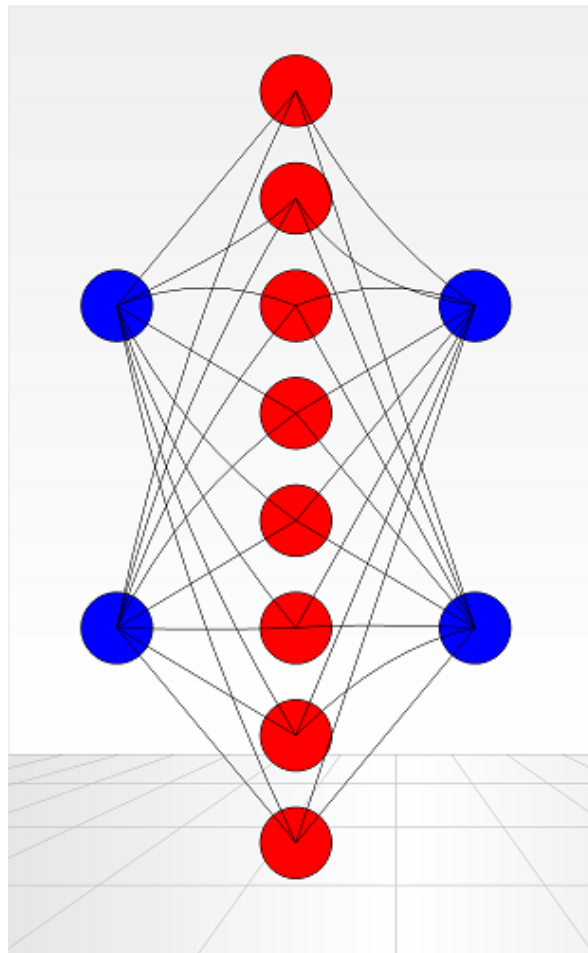


Figura 7.1 – Arquitetura inicial.

7.1.2 - A escolha do algoritmo de treinamento

A escolha do algoritmo de treinamento ficou com o *Backpropagation*, um algoritmo de aprendizado supervisionado (que se vale da utilização de agentes indicativos de correção entre valor obtido e valor esperado), cuja direção de fluxo de ativações é *feedforward*. Ou seja, foi escolhido um algoritmo bastante conhecido e eficiente que adota uma arquitetura fixa de rede (não construtivo).

7.1.3 – Funções de ativação

As funções de ativação escolhidas, foram a função transferência sigmóide tipo tangente hiperbólica (no MATLAB ela se chama *tansig*), que atua após a camada escondida; e a função transferência linear (no MATLAB ela se chama *purelin*), que atua após a camada de saída. Esta foi a melhor combinação encontrada para, inicialmente limitar os valores obtidos entre -1 e +1, e posteriormente permitir flexibilidade ao redor dos símbolos gerados. Ambas as funções estão em destaque conforme vemos nas Figuras 7.2 e 7.3.

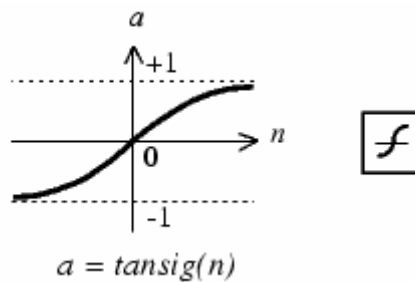


Figura 7.2 – Função transferência tangente hiperbólica [15].

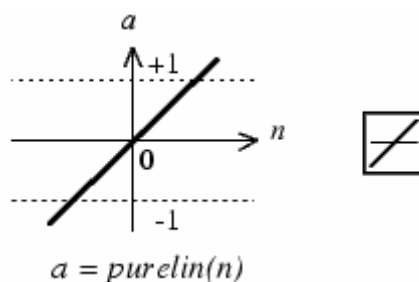


Figura 7.3 – Função transferência linear [15].

7.2 – MODELANDO UM CANAL AWGN

Uma vez que, canais via satélite são comumente modelados como sendo AWGN devido ao seu alto grau de imprevisibilidade, este foi o modelo inicial.

7.2.1 – Geração do campo amostral (módulo *simuladorqpskv4.m*)

O primeiro módulo, denominado *simuladorqpskv4.m* (ou seja, o módulo de simulação qpsk na versão 4), tem por função gerar um campo amostral pseudo randômico de 150

símbolos com quatro valores possíveis distribuídos de 0 a 3, valores estes típicos da modulação qpsk.

O campo amostral randômico sofre a codificação Gray e, posteriormente passa pela modulação qpsk nativa da biblioteca do MATLAB (pskmod). Após isso, finalmente passa por um canal AWGN setado para um SNR de 9 dB, pois verificou-se empiricamente que este valor permitia uma análise conveniente. Além disto, com valores inferiores a este, a dispersão da constelação não permite uma boa análise. A hipótese assumida de um canal AWGN, ou seja, ruído branco aditivo com distribuição de voltagem Gaussiana, de modo geral é válida, considerando apenas o ruído termal.

Os resultados importantes do módulo simuladorqpskv4, são 5 arquivos txt denominados: xgray.txt, xgraymoduladareal.txt, xgraymoduladaimag.txt, xgraymoduladaAWGNreal.txt e xgraymoduladaAWGNimag.txt.

Esses arquivos armazenam os símbolos gerados com a codificação Gray (xgray.txt), armazenam os símbolos devidamente modulados nas suas partes real e imaginária (xgraymoduladareal.txt e xgraymoduladaimag.txt) e finalmente armazenam os símbolos modulados através de um canal AWGN (xgraymoduladaAWGNreal.txt e xgraymoduladaAWGNimag.txt). Esses arquivos serão utilizados no segundo módulo para o treinamento da rede neural.

7.2.2 – Módulo de treinamento (módulo percepreino1.m)

O segundo módulo, denominado percepreino1.m (ou seja, o módulo responsável pelo treinamento do perceptron na versão 1) ficou com a responsabilidade pelo treino da rede, utilizando para isso o campo amostral gerado pelo módulo anterior (simuladorqpskv4.m), através da leitura dos arquivos gerados (xgray.txt, xgraymoduladareal.txt, xgraymoduladaimag.txt, xgraymoduladaAWGNreal.txt e xgraymoduladaimag.txt). A arquitetura adotada foi o perceptron com algoritmo de treinamento backpropagation.

Dentro do *Backpropagation* existem diversas variações. Inicialmente foi escolhida uma, apontada pelo Help do MATLAB como sendo de rápida convergência. Essa variação é a função traingda (*Backpropagation* com *learning rate* variável) que opera no modo *batch*.

Ao longo do desenvolvimento da rede, percebeu-se contudo, que para manter a topologia anteriormente mencionada, esse algoritmo não permitia a escalabilidade do modelo, conforme mostrado na Figura 7.4 (durante a fase inicial de desenvolvimento, o algoritmo foi satisfatório para um conjunto amostral pequeno, mas para conjuntos amostrais com 150 símbolos, por exemplo, a sua convergência tornou-se bastante lenta para os valores setados, sendo necessárias 10.000 épocas para a apresentação de um resultado mediano).

Contudo, esse inconveniente mostrou que uma camada escondida com 16 neurônios foi mais eficiente que uma camada escondida com apenas 8 neurônios.

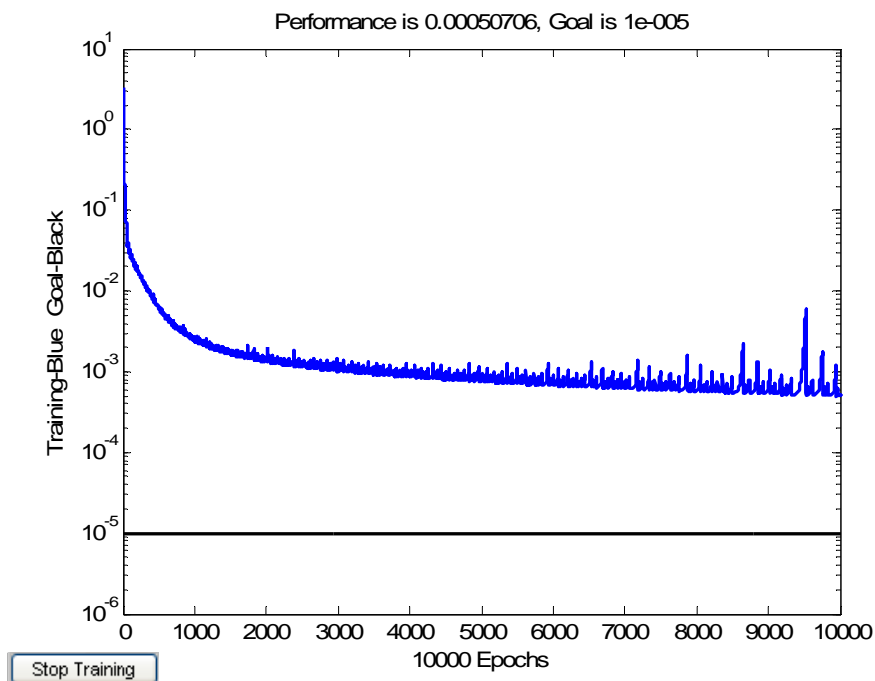


Figura 7.4 – Curva do treinamento com traingda.

Foram necessárias 10.000 épocas e aproximadamente 36 segundos para atingir o valor acima em uma das simulações. O campo amostral utilizado foi de 150 símbolos e 16 neurônios na camada escondida.

Os parâmetros de aprendizado utilizados foram:

- `net.trainParam.lr=0.05;`
- `net.trainParam.lr_inc=1.05;`

- `net.trainParam.epochs=10000`.

Diante da necessidade de melhoria da performance, foi utilizado como guia uma tabela comparativa das diversas variações do *Backpropagation* existente no MATLAB.

Decidiu-se testar a função de treinamento Levenberg-Marquardt (`trainlm`), desenvolvida de modo a dispensar o uso da matriz Hessiana. Mantendo-se os parâmetros anteriormente usados para a `trainda`, foi feita apenas a troca de função para a `trainlm`.

Foi impressionante o resultado obtido, conforme mostrado pela Figura 7.5. A convergência foi extremamente rápida.

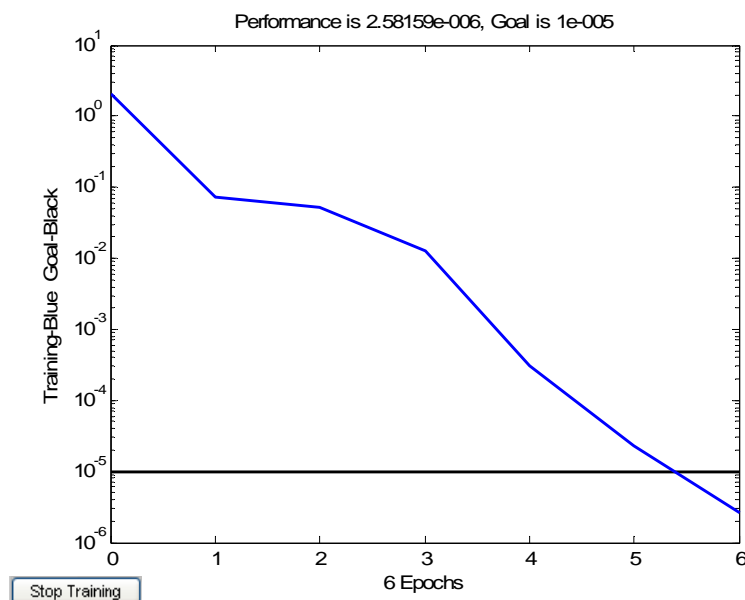


Figura 7.5 – Curva do treinamento com `trainlm`.

Diante do resultado obtido, foi feito um “endurecimento” dos parâmetros anteriores, exigindo-se menor tolerância ao erro:

- `net.trainParam.show=10`;
- `net.trainParam.lr=0.05`;
- `net.trainParam.lr_inc=1.05`;
- `net.trainParam.epochs=1000`;
- `net.trainParam.goal=1e-8`.

Isto é mostrado na Figura 7.6, a seguir:

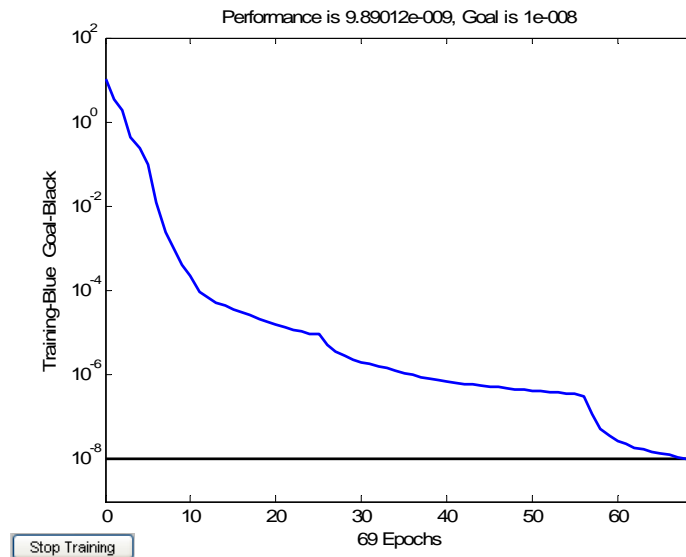


Figura 7.6 – Curva trainlm com endurecimento.

De maneira inquestionável, houve uma enorme melhoria qualitativa da nossa rede, que nos entrega resultados muito mais refinados. Embora tenha ocorrido um aumento do número de épocas de 5 para 69, há de se considerar o “endurecimento” feito no sentido de menor tolerância ao erro, pois passamos de `net.trainParam.goal = 1e-5` para `net.trainParam.goal = 1e-8`.

Uma vez que os módulos são interdependentes, é conveniente lembrar que este módulo se utiliza das informações geradas pelo módulo de geração do campo amostral (`xgraymoduladaAWGNreal.txt`, `xgraymoduladaAWGNimag.txt`, `xgraymoduladareal.txt` e `xgraymoduladaimag.txt`). Esses arquivos fornecem as informações distorcidas pelo canal AWGN e a resposta desejada para essas entradas distorcidas.

As informações realmente úteis produzidas por este módulo (módulo de treinamento), são os pesos (entre a camada de entrada e a camada escondida, e entre a camada escondida e a camada de saída), e os bias (dos neurônios da camada escondida, e dos neurônios da camada de saída).

Assim, o módulo de treinamento produz as saídas: `pesosIW.txt`, `bias1.txt`, `pesosLW.txt` e `bias2.txt`.

- o arquivo `pesosIW.txt` guarda uma matriz de 16 linhas por 2 colunas;
- o arquivo `bias1.txt` guarda uma matriz de 16 linhas por 1 coluna;

- o arquivo pesosLW.txt guarda uma matriz de 2 linhas por 16 colunas;
- o arquivo bias2.txt guarda uma matriz de 2 linhas por 1 coluna.

Estes são os pesos da rede constituída por 2 neurônios na camada de entrada, 16 neurônios na camada escondida e 2 neurônios na camada de saída.

7.2.3 – Módulo de validação (módulo percepronto1.m)

O terceiro módulo, denominado percepronto1.m, é o responsável pela validação da rede estabelecida como válida pelo segundo módulo (perceptreino1.m).

Este módulo gera pseudo randomicamente 150 símbolos com quatro valores possíveis distribuídos de 0 a 3, valores estes típicos da modulação qpsk. Este conjunto amostral será utilizado para validação do modelo.

São utilizados os pesos encontrados no segundo módulo, através dos arquivos pesosIW.txt, bias1.txt, pesosLW.txt e bias2.txt.

Semelhantemente ao primeiro módulo, é gerada uma matriz com os símbolos, os símbolos recebem codificação Gray, ocorre a modulação, e a passagem por um canal AWGN. O sinal resultante do canal AWGN é submetido à rede neural de acordo com os pesos e bias obtidos no segundo módulo.

É exibida uma figura que mostra a constelação com os símbolos gerados, com e sem AWGN.

É exibida também uma figura que mostra as correções feitas pela rede neural, nos símbolos que passam pelo canal AWGN.

O sinal resultante da rede é comparado com sinal original através da função symerr, que verifica o *Symbol Error Rate* (SER). Obviamente, quanto menor o *number* (indica o número de elementos que diferem entre si antes e depois), melhor a precisão da rede. O *ratio* indica a razão entre o *number* e as 150 amostras.

É recomendável lembrar que antes de rodar este módulo, é importante que o *workspace* do MATLAB seja limpo.

Rodando o módulo de validação podemos obter valores como:

- $number = 0$
- $ratio = 0$

ou

- $number = 1$
- $ratio = 0.0067$

Os valores obtidos quando $number = 1$, estão representados graficamente a seguir na Figura 7.7, e indicam que, como o campo amostral de validação da rede é randomicamente gerado, caso haja valores em uma área de conflito (onde o canal AWGN imprime um elevado desvio), a rede obviamente não entende estes valores espúrios. A correção proposta é mostrada na Figura 7.8.

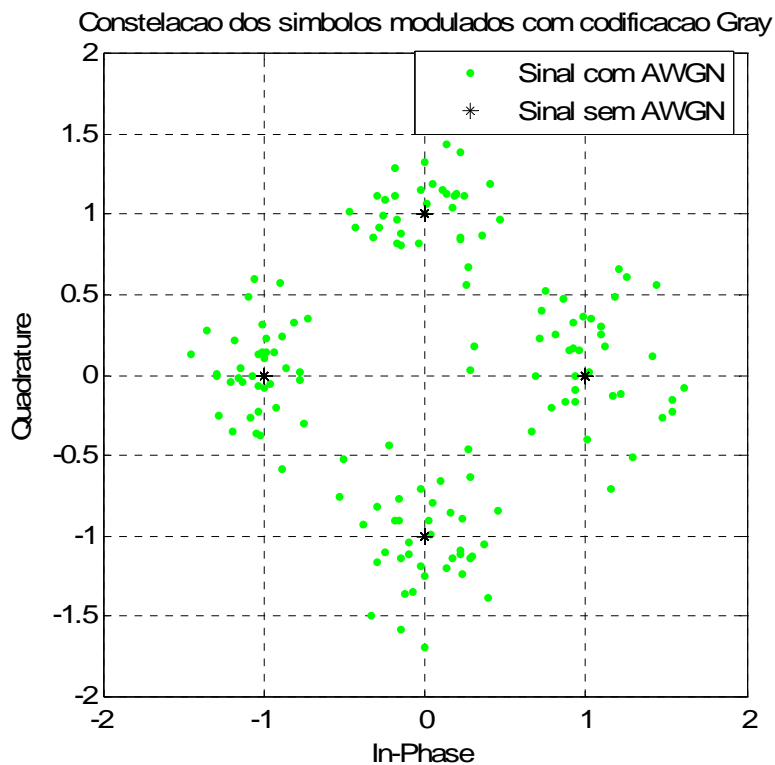


Figura 7.7 – Constelação com AWGN com $number = 1$.

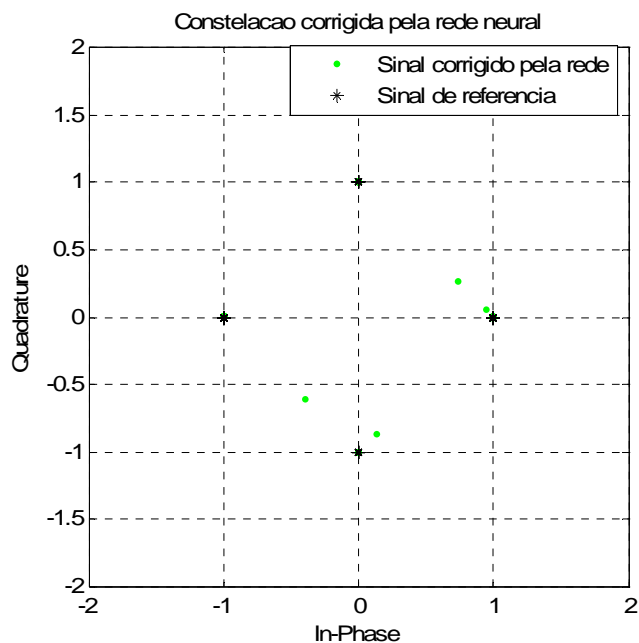


Figura 7.8 – Constelação corrigida com $number = 1$.

Também foram encontrados nas simulações valores de:

- $number = 2$
- $ratio = 0.0133$

e ocasionalmente:

- $number = 3$
- $ratio = 0.0200$

Mas nunca foram encontrados valores superiores a estes.

7.2.4 – Análise imediata dos resultados

Diante dos resultados imediatos, torna-se claro que a rede neural, neste estágio, é capaz de corrigir as pequenas distorções de fase que um canal AWGN pode causar, contudo não está pronta ainda para o uso em casos mais complexos, onde a diferença de fase provocada por um canal AWGN ultrapasse 45° em relação ao sinal de origem.

Torna-se evidente também, a necessidade de compatibilização de grandezas, a fim de aproximarmos o estudo teórico a uma aplicação real.

7.2.5 – Compatibilizando as grandezas do canal AWGN

Até o presente momento, não houve preocupação formal com relação ao fato da relação sinal ruído (SNR) aplicada nos módulos simuladorqpskv4.m, perceptreino1.m e perceppronto1.m, não estar adaptada à grandeza de nosso interesse, E_b/N_0 , que é a razão entre a energia por bit e a densidade espectral de potência de ruído.

A potência relativa de ruído em um canal AWGN é tipicamente descrita por grandezas como:

- SNR (*Signal-to-Noise Ratio*);
- E_b/N_0 (razão entre a energia por bit e a densidade espectral de potência do ruído);
- E_s/N_0 (razão entre a energia por símbolo e a densidade espectral de potência do ruído).

Para um dado E_b/N_0 (em dB), a SNR correspondente será:

$$\text{SNR} = E_b/N_0 + 10 \cdot \log_{10}(k) - 10 \cdot \log_{10}(n_{\text{samp}}) \quad (7.1)$$

Onde k corresponde ao número de bits por símbolo e n_{samp} é o fator de sobreamostragem. No nosso caso $k = 2$ e $n_{\text{samp}} = 1$. Assim sendo, os módulos simuladorqpskv5.m e perceppronto2.m, representam esta necessidade de adequação.

Até o presente momento, também não houve a preocupação com a grandeza de referência (figura de mérito) que indica a qualidade da transmissão, que é o BER (*Bit Error Rate*). Até o momento, utilizou-se o SER (*Symbol Error Rate*) como referência inicial de estudo.

A figura de mérito usual de enlaces de rádio digital é o BER (*Bit Error Rate*). Esta probabilidade de erro de bit, é entendida como a probabilidade de erro de um bit entregue através de um link. Esta figura de mérito é um indicativo da qualidade de um sistema de comunicação digital, assim como a SNR (*Signal to Noise Ratio*) é a figura de mérito indicativa da qualidade de um sistema de comunicação analógico.

Um erro de bit ocorre porque houve um erro de símbolo. Considerando a modulação QPSK, a codificação Gray assegura que fases adjacentes diferem de apenas um bit. Ou

seja, para que o ruído provoque um erro de 2 bits, há de se ter um erro de fase maior que 135° . Ruídos que causam um erro de fase de 45° a 135° causarão um erro de apenas 1 bit.

Como resultado imediato, é altamente provável que um sinal QPSK tenha um número de erros de símbolo igual a um número de erros de bit. Pois é improvável que o canal AWGN gere um erro de fase maior que 135° .

Erros de símbolo aumentam de acordo com o incremento do ruído termal, interferências externas e ISI (*InterSymbol Interference*). Considerando apenas o ruído termal, o SER é calculado a partir da relação E_s/N_0 , ou seja, a energia por símbolo em Joules dividida pela densidade de potência de ruído em W/Hz. Uma alta relação E_s/N_0 implica baixo SER.

O BER em uma modulação QPSK, considerando um canal AWGN (ruído aditivo branco com distribuição de voltagem Gaussiana) é dado por:

$$BER = \frac{1}{2} \operatorname{erfc} \left[\sqrt{\frac{E_b}{N_0}} \right] = Q \left[\sqrt{\frac{2E_b}{N_0}} \right] = Q \left[\sqrt{\frac{C}{N}} \right] \quad (7.2)$$

Onde temos a função $Q(z)$ e a função de erro complementar $\operatorname{erfc}(x)$, expressas através de tabelas desenvolvidas utilizando técnicas de cálculo numérico. Convém destacar ainda, que:

- C/N : relação entre a potência recebida da portadora e a potência de ruído em uma banda alocada (BW em Hertz) em dB;
- C/N_0 : relação entre a potência recebida da portadora e a densidade espectral de potência de ruído em 1 Hz em dBHz;
- $C/N = C/N_0 - 10 \log (BW)$.

A relação entre E_s/N_0 e E_b/N_0 é dada por:

$$E_s / N_0 (dB) = E_b / N_0 (dB) + 10 \cdot \log_{10} (k) \quad (7.3)$$

Onde k é o número de bits por símbolo.

A relação entre E_s/N_0 e SNR para entradas complexas e entradas reais é, respectivamente:

$$Es / No(dB) = 10.\log_{10}(T_{sym} / T_{samp}) + SNR(dB)$$

$$Es / No(dB) = 10.\log_{10}(0,5T_{sym} / T_{samp}) + SNR(dB) \quad (7.4)$$

Onde T_{sym} é o período do símbolo do sinal, e T_{samp} é o período de amostragem do sinal. Assim sendo, o módulo `perceponto2.m` também está adequado a grandeza BER. As Tabelas 7.1 e 7.2 permitem estabelecer uma relação entre valores teóricos e práticos previstos.

Tabela 7.1 – Tabela de Q em função de Q(z) [18].

Table of Q Function Q(z)			
z	Q(z)	z	Q(z)
0	0.5	5.0	2.872 E-7
2.0	2.280 E-2	5.1	1.701 E-7
2.1	1.791 E-2	5.2	9.981 E-8
2.2	1.394 E-2	5.3	5.799 E-8
2.3	1.075 E-2	5.4	3.372 E-8
2.4	8.220 E-3	5.5	1.902 E-8
2.5	6.227 E-3	5.6	1.073 E-8
2.6	4.674 E-3	5.7	6.000 E-9
2.7	3.476 E-3	5.8	3.320 E-9
2.8	2.562 E-3	5.9	1.820 E-9
2.9	1.871 E-3	6.0	9.979 E-10
3.0	1.354 E-3	6.1	5.310 E-10
3.1	9.702 E-4	6.2	2.827 E-10
3.2	6.889 E-4	6.3	1.490 E-10
3.3	4.847 E-4	6.4	7.778 E-11
3.4	3.378 E-4	6.5	4.021 E-11
3.5	2.332 E-4	6.6	2.058 E-11
3.6	1.595 E-4	6.7	1.043 E-12
3.7	1.081 E-4	6.8	5.236 E-12
3.8	7.252 E-5	6.9	2.603 E-12
3.9	4.821 E-5	7.0	1.281 E-12
4.0	3.174 E-5	7.1	6.244 E-13
4.1	2.070 E-5	7.2	3.014 E-13
4.2	1.337 E-5	7.3	1.440 E-13
4.3	8.558 E-6	7.4	6.816 E-14
4.4	5.423 E-6	7.5	3.194 E-14
4.5	3.404 E-6	7.6	1.482 E-14
4.6	2.117 E-6	7.7	6.810 E-15
4.7	1.303 E-6	7.8	3.098 E-15
4.8	7.948 E-7	7.9	2.396 E-15
4.9	4.800 E-7	8.0	6.226 E-16

Tabela 7.2 – Tabela de $\text{erfc}(x)$ em função de x [18].

Table of Function $\text{erfc}(x)$			
x	$\text{erfc}(x)$	x	$\text{erfc}(x)$
0.0	1.00000	2.0	5.167 E-3
0.05	0.94363	2.1	3.267 E-3
0.10	0.88754	2.2	2.029 E-3
0.15	0.83200	2.3	1.237 E-3
0.20	0.77730	2.4	7.408 E-4
0.25	0.72367	2.5	4.357 E-4
0.30	0.67137	2.6	2.515 E-4
0.35	0.62062	2.7	1.426 E-4
0.40	0.57161	2.8	7.932 E-5
0.45	0.52452	2.9	4.331 E-5
0.50	0.47950	3.0	2.321 E-5
0.55	0.43668	3.1	1.220 E-5
0.60	0.39614	3.2	6.297 E-6
0.65	0.35797	3.3	3.187 E-6
0.70	0.32220	3.4	1.583 E-6
0.75	0.28884	3.5	7.713 E-7
0.80	0.25790	3.6	3.687 E-7
0.85	0.22933	3.7	1.729 E-7
0.90	0.20309	3.8	7.951 E-8
0.95	0.17911	3.9	3.587 E-8
1.00	0.15730	4.0	1.587 E-8
1.05	0.13776	4.1	6.889 E-9
1.10	0.11979	4.2	2.932 E-9
1.15	0.10388	4.3	1.224 E-9
1.20	0.08969	4.4	5.012 E-10
1.25	0.07710	4.5	2.013 E-10
1.30	0.06599	4.6	7.925 E-11
1.35	0.05624	4.7	3.060 E-11
1.40	0.04771	4.8	1.159 E-11
1.45	0.04030	4.9	4.303 E-12
1.50	0.03389	5.0	1.567 E-12
1.55	0.02838	5.1	5.596 E-13
1.60	0.02363	5.2	1.959 E-13
1.65	0.01962	5.3	6.727 E-14
1.70	0.01621	5.4	2.265 E-14
1.75	0.01333	5.5	7.476 E-15
1.80	0.01091	5.6	2.420 E-15
1.85	0.00889	5.7	7.680 E-16
1.90	0.00721	5.8	2.390 E-16
1.95	0.00582	5.9	7.291 E-17

Com base nas tabelas teóricas acima (Tabelas 7.1 e 7.2), foi possível estabelecer a Tabela 7.3 comparativa entre os valores teóricos sem FEC e os valores garantidos com o uso do FEC 3/4, de acordo com a Tabela 2.6.

Tabela 7.3 – Tabela comparativa entre o BER teórico e o BER com FEC 3/4.

Eb/No	BER teórico sem FEC	BER garantido FEC 3/4
9dB=7,9432	$3,17.10^{-5}$	10^{-7}
8,2 dB=6,6069	$1,59.10^{-4}$	10^{-6}
7,3dB=5,3703	$4,84.10^{-4}$	10^{-5}
6,6dB=4,5708	$1,35.10^{-3}$	10^{-4}
5,8dB=3,8018	$3,01.10^{-3}$	10^{-3}

7.2.6 – Nova bateria de testes para o canal AWGN

Considerando as novas condições de contorno, decidiu-se pela realização de nova bateria de testes, adequando as simulações com os valores teóricos fornecidos pelo fabricante, de acordo com a Tabela 2.6. Desta forma, a relação Eb/No passou a ter o valor de 9dB, significando dizer que a SNR passou a ter o valor de 12dB.

Tudo isto de acordo com o valor apurado no Capítulo 6.2 para Eb/No.

7.2.6.1 – Testes com 1.000 símbolos

Foram alterados no módulo **simuladorqpskv5.m**, a quantidade de símbolos gerados (de 150 para 1.000) e a relação Eb/No=9dB, o que significa dizer que o SNR=12dB, isto é mostrado pela Figura 7.9.

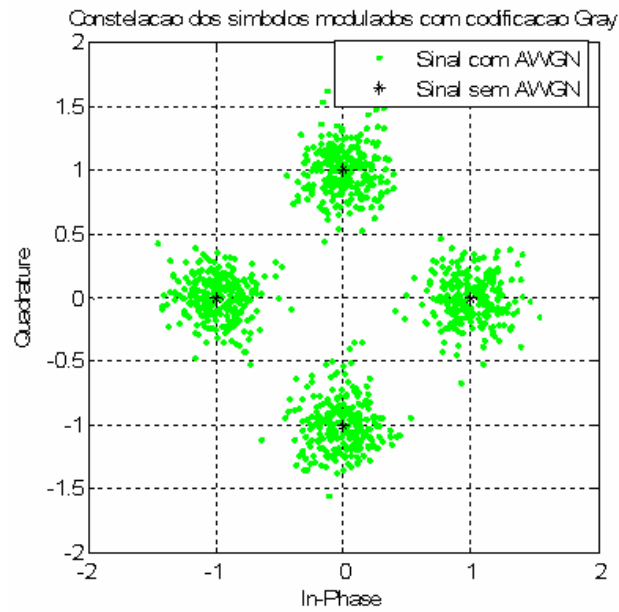


Figura 7.9 – Geração de 1.000 símbolos pseudo aleatórios.

A rede foi treinada nesta condição através do módulo **perceptreino2.m**, conforme vemos na Figura 7.10.

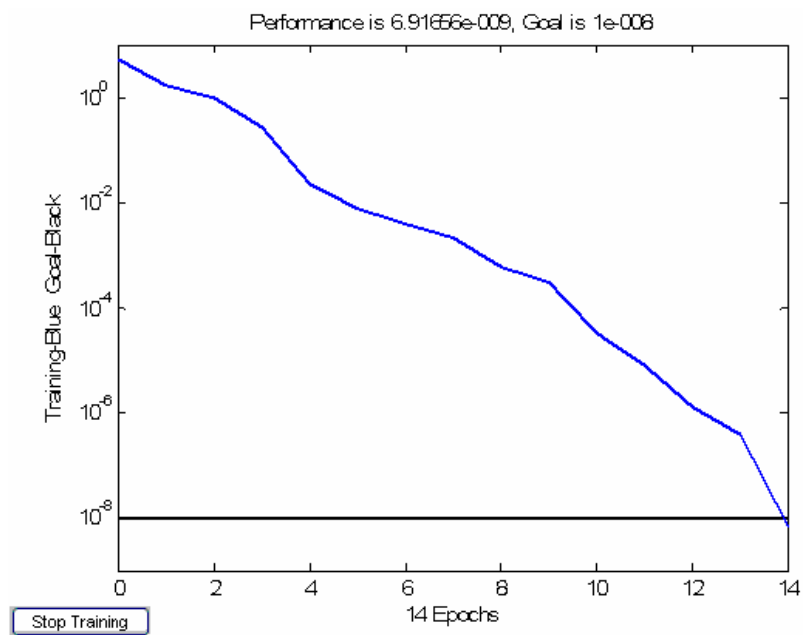


Figura 7.10 – Treinamento da rede neural com 1.000 símbolos.

Utilizou-se o módulo **percepronto2.m** para validação da rede. Conforme a Figura 7.11, foram gerados 1.000 novos símbolos pseudo aleatórios para validação da rede com $EB/No=9dB$.

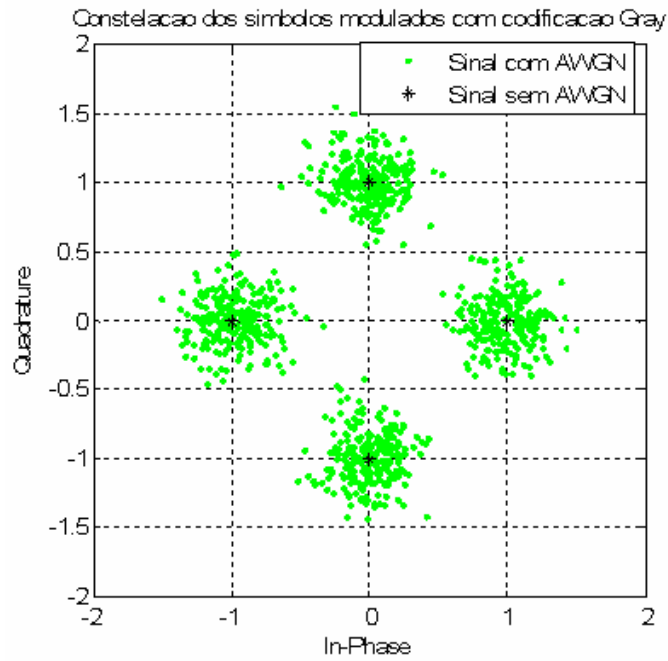


Figura 7.11 – Geração de 1.000 novos símbolos pseudo aleatórios.

A correção fornecida pela rede foi a exibida pela Figura 7.12 abaixo:

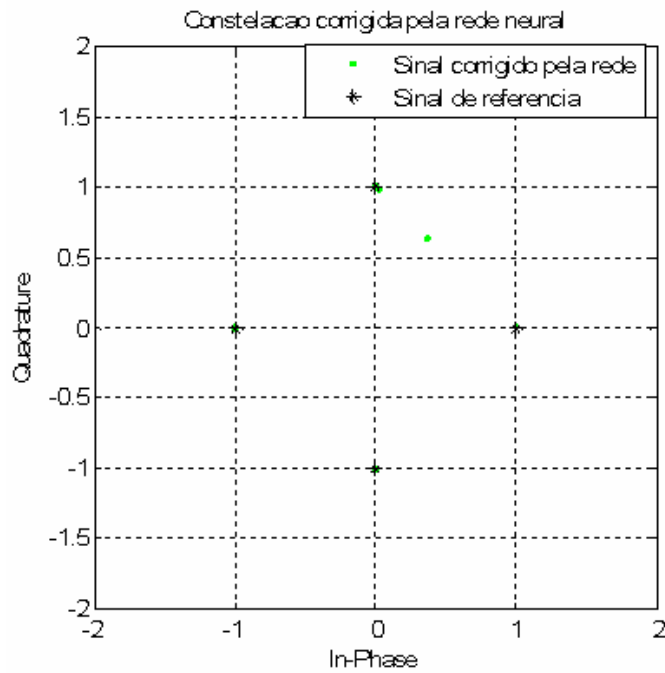


Figura 7.12 – Correção sugerida pela rede neural.

Com os seguintes índices:

- number = 0
- BER = 0
- number = 0
- SER = 0

7.2.6.2 – Testes com 10.000 símbolos

Foram alterados no módulo **simuladorqpskv5.m**, a quantidade de símbolos gerados (de 1.000 para 10.000), mantendo a relação $E_b/N_0=9\text{dB}$, o que significa dizer que o $\text{SNR}=12\text{dB}$. Isto é mostrado pela Figura 7.13.

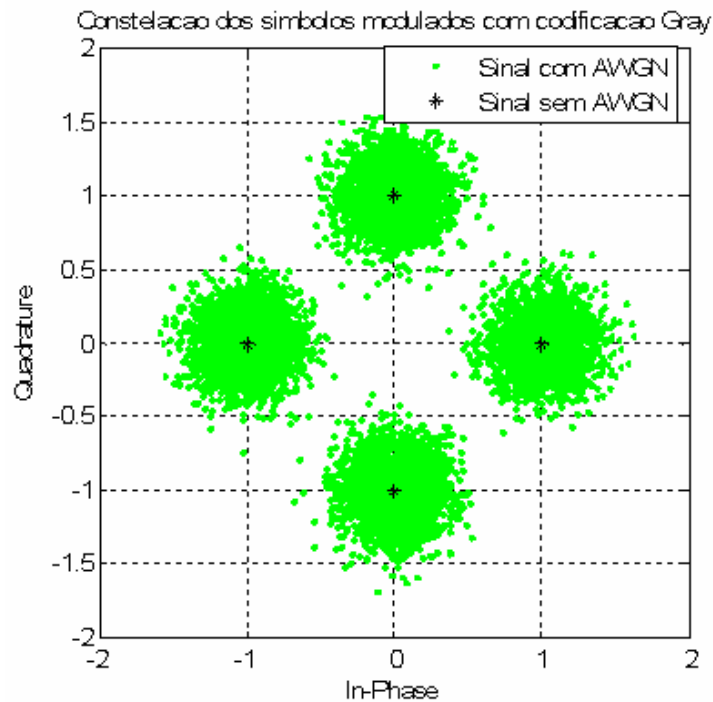


Figura 7.13 – Geração de 10.000 símbolos pseudo aleatórios.

A rede foi treinada nesta condição através do módulo **perceptreino2.m**, conforme mostrado pela Figura 7.14.

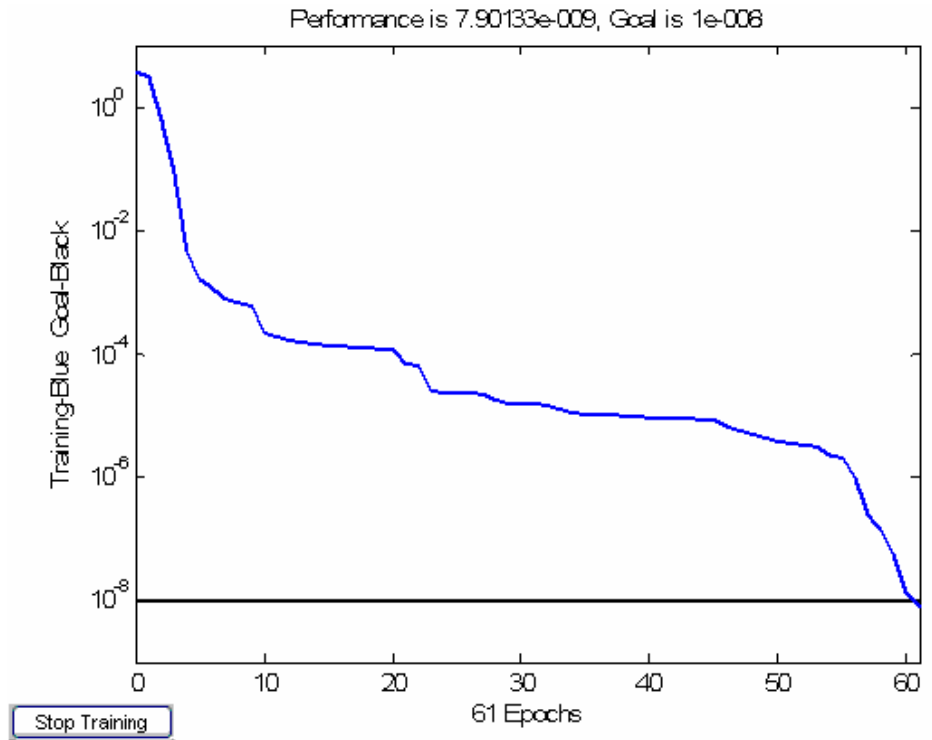


Figura 7.14 – Treinamento da rede neural com 10.000 símbolos.

Nesta etapa, ficou evidente que um maior esforço computacional foi necessário para o treinamento da rede, devido ao acréscimo do campo amostral utilizado para o treinamento.

Conforme a Figura 7.15, utilizou-se o módulo **percepronto2.m** para validação da rede. Foram gerados 10.000 novos símbolos pseudo aleatórios para validação da rede com $EB/No=9dB$.

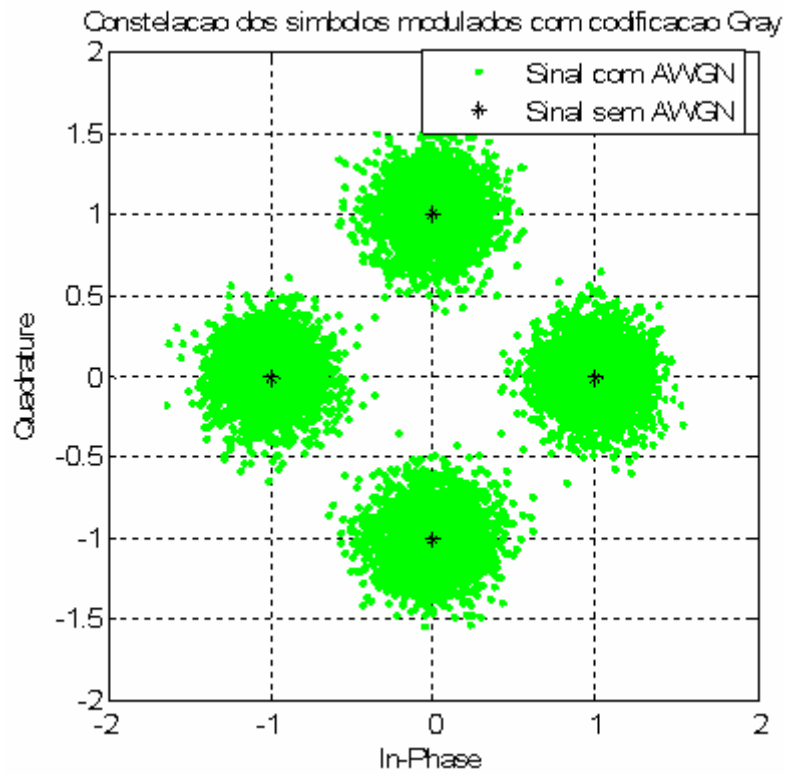


Figura 7.15 – Geração de 10.000 novos símbolos pseudo aleatórios.

A correção fornecida pela rede foi a mostrada pela Figura 7.16.

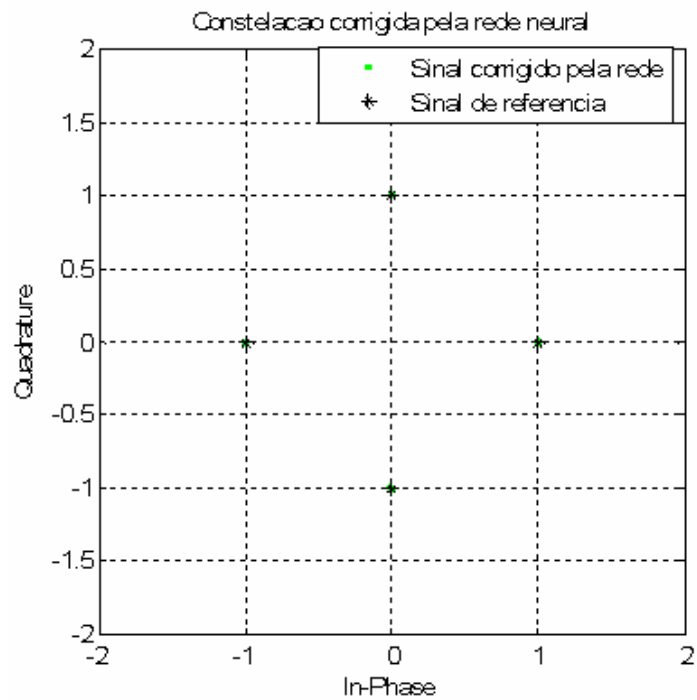


Figura 7.16 – Correção sugerida pela rede neural.

Com os seguintes índices:

- number =3
- BER = 1,5000e-004
- number = 2
- SER = 2,0000e-004

7.2.6.3 – Testes com 100.000 símbolos

Foram alterados no módulo **simuladorqpskv5.m**, a quantidade de símbolos gerados (de 10.000 para 100.000), mantendo a relação $E_b/N_0=9\text{dB}$, o que significa dizer que o $\text{SNR}=12\text{dB}$.

Como vemos na Figura 7.17, a rede foi treinada nesta condição através do módulo **perceptreino2.m**, contudo foram utilizados apenas os 15.000 primeiros símbolos, uma vez que um grande número de amostras, além de aumentar o esforço computacional, não permite a convergência da rede.

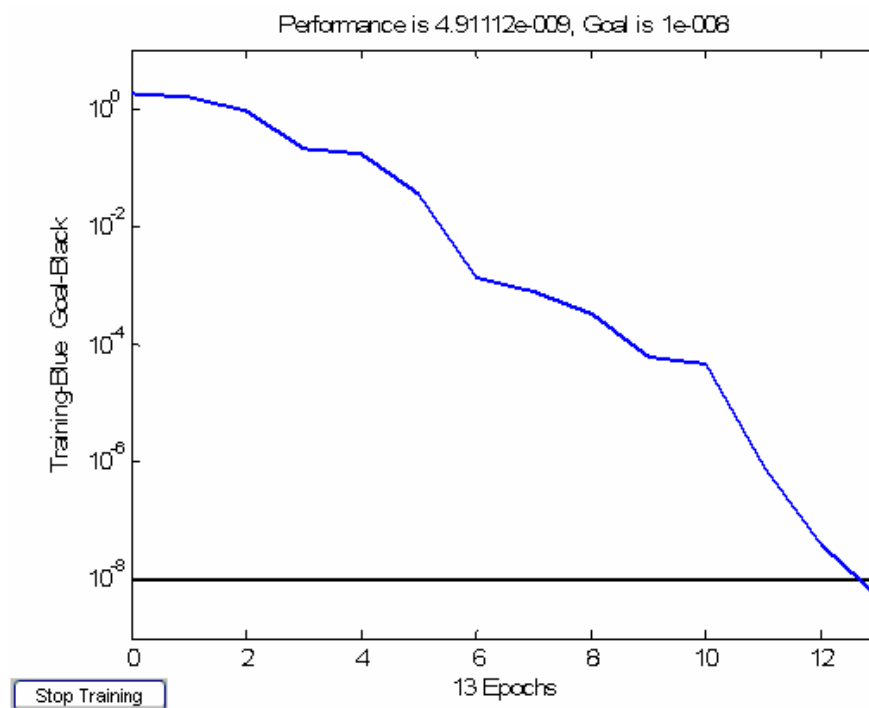


Figura 7.17 – Treinamento da rede neural com 100.000 símbolos.

Utilizou-se o módulo **percepronto2.m** para validação da rede. Foram gerados 100.000 novos símbolos pseudo aleatórios para validação da rede com $E_b/N_0=9\text{dB}$. A correção exibida pela Figura 7.18, sugere um claro entendimento por parte da rede neural, que os quatro símbolos formam um quadrado.

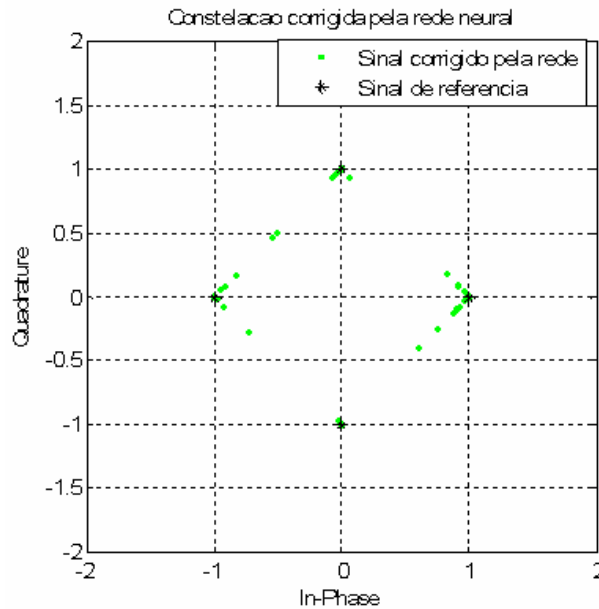


Figura 7.18 – Correção sugerida pela rede neural.

Com os seguintes índices:

- number = 15
- BER = 7,5000e-005
- number = 9
- SER = 9,0000e-005

7.2.6.4 – Testes com 1.000.000 de símbolos

Finalmente, a Figura 7.19 mostra a alteração no módulo **simuladorqpskv5.m**, da quantidade de símbolos gerados (de 100.000 para 1.000.000), mantendo a relação $E_b/N_0=9\text{dB}$. A Figura 7.20 mostra o treinamento realizado pelo módulo **perceptreino2.m**, contudo, foram utilizados apenas os 15.000 primeiros símbolos para o treinamento, uma vez que um grande número de amostras, além de aumentar o esforço computacional, impede a convergência da rede.

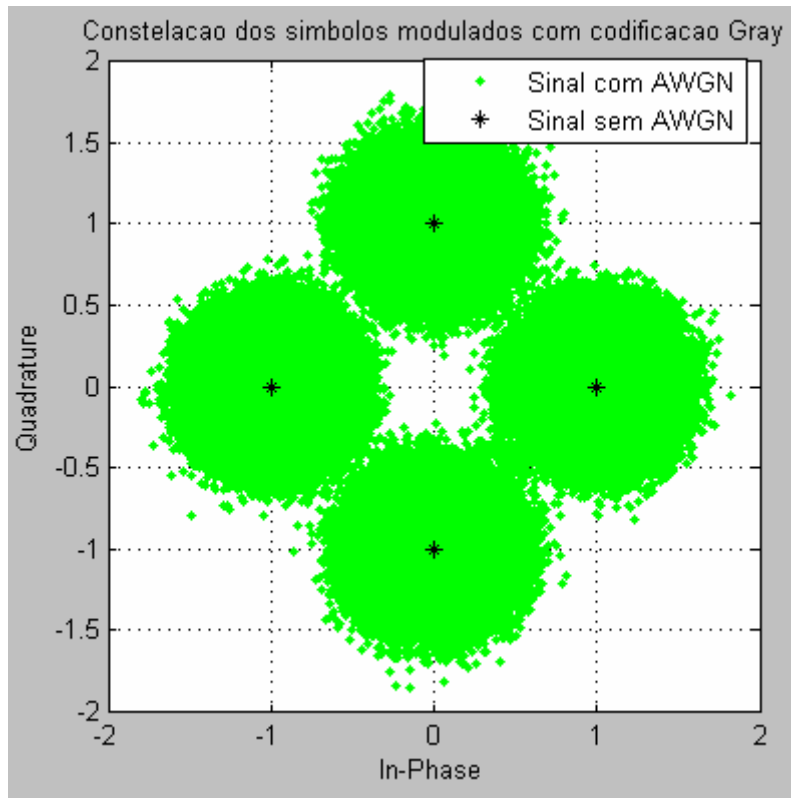


Figura 7.19 – Geração de 1.000.000 de símbolos.

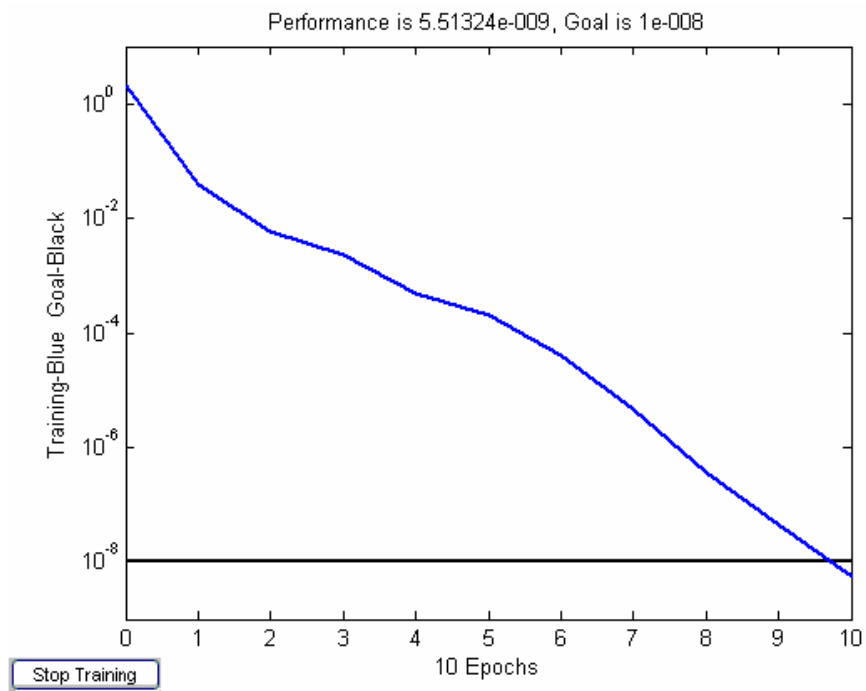


Figura 7.20 – Treinamento da rede neural com 1.000.000 de símbolos.

Utilizou-se o módulo **perceppronto2.m** para validação da rede. Foram gerados 1.000.000 de novos símbolos pseudo aleatórios para validação da rede com $E_b/N_0=9\text{dB}$, com os seguintes resultados:

- number = 186
- BER = 9,3000e-005
- number = 152
- SER = 1,5200e-004

7.2.6.5 – Análise imediata da nova bateria de testes

Assim obtivemos os seguintes resultados consolidados na Tabela 7.4:

Tabela 7.4 – Resultados consolidados.

número de símbolos	BER	SER
1.000	0	0
10.000	1.5e-4	2.0e-4
100.000	7.5e-5	9.0e-5
1.000.000	9.3e-5	1.5e-4

Tabela 7.5 – Repetição da Tabela 7.3.

E_b/N_0	BER teórico sem FEC	BER garantido FEC 3/4
9dB=7,9432	$3,17 \cdot 10^{-5}$	10^{-7}
8,2 dB=6,6069	$1,59 \cdot 10^{-4}$	10^{-6}
7,3dB=5,3703	$4,84 \cdot 10^{-4}$	10^{-5}
6,6dB=4,5708	$1,35 \cdot 10^{-3}$	10^{-4}
5,8dB=3,8018	$3,01 \cdot 10^{-3}$	10^{-3}

Desta forma, de acordo com a Tabela 7.4, para um campo amostral de 1.000.000 símbolos, tivemos um BER de 9.3e-5. Isto, de acordo com a Tabela 7.3, deixa claro que para um E_b/N_0 de 9dB, os valores obtidos pela rede neural ficam aquém dos valores obtidos para

um canal que utiliza técnicas de correção de erro FEC 3/4 (BER de 10^{-7}), como era de se esperar, e muito próximo do valor teórico previsto.

O que é interessante de se observar, é que a rede neural se presta bem para fazer a filtragem necessária, a fim de reconhecer os símbolos que estão dentro da curva de Gauss, e portanto, próximos do valor teórico esperado para um enlace submetido a um AWGN (ruído aditivo branco com distribuição de voltagem Gaussiana), que não aplique técnicas de correção de erros.

Diante deste fato, fica evidente que o uso de rede neural é eficaz apenas quando os erros de fase são menores de 45° . Fica claro também que as técnicas de codificação são complementares ao uso de redes neurais na demodulação.

7.3 – MODELANDO UM CANAL ATENUADO

Agora será feita a implementação de uma rede neural, capaz de corrigir os efeitos de um canal que aplica uma atenuação na informação transmitida.

O procedimento será bastante semelhante ao tomado no estudo de um canal AWGN, ou seja, continuaremos utilizando 3 módulos, devidamente modificados em novas versões, para gerar um campo amostral (simuladorqpskv6.m), treinar a rede neural de acordo com o campo amostral gerado (perceptreino3.m) e validar os resultados (perceppronto3.m).

O módulo simuladorqpsvv6.m, gera 1.000 símbolos pseudo aleatórios de 0 a 3. Gera também 1.000 números pseudo aleatórios distribuídos uniformemente de 0 a 0,1259. Estes 1.000 números pseudo aleatórios distribuídos uniformemente de 0 a 0,1259 vão proporcionar uma atenuação máxima de 0,58 dB (um valor qualquer escolhido) conforme vemos na Figura 7.21; uma vez que $-0,58 \text{ dB} = 10 \log ((1 - 0,1259) / 1)$.

Serão realizadas, a título de ensaio, apenas simulações com 1.000 símbolos e atenuação de 0,58 dB.

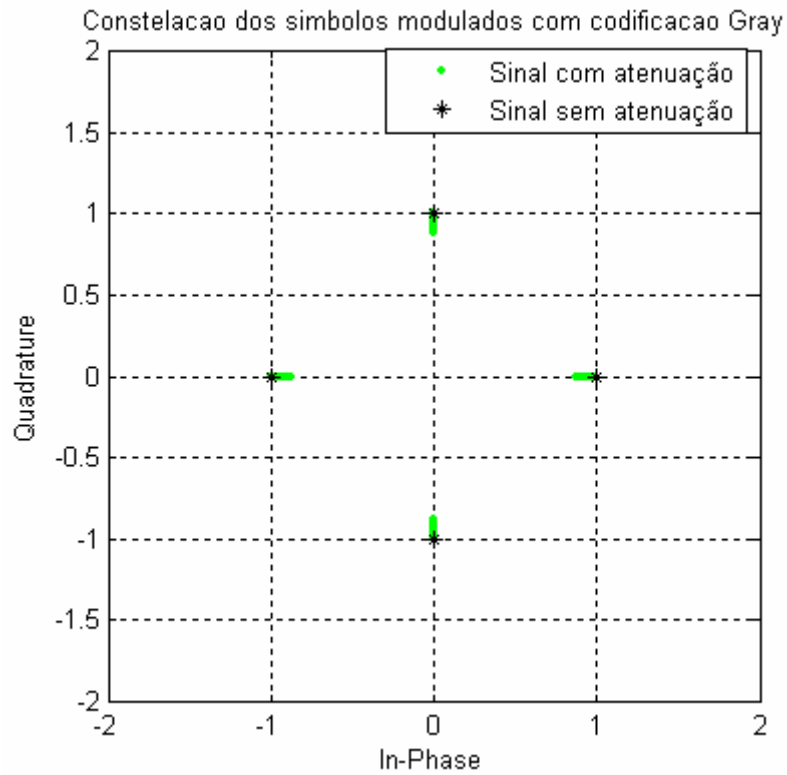


Figura 7.21 – Símbolos atenuados.

A seguir, a Figura 7.22 mostra o treinamento executado pelo módulo percepreino3.m.

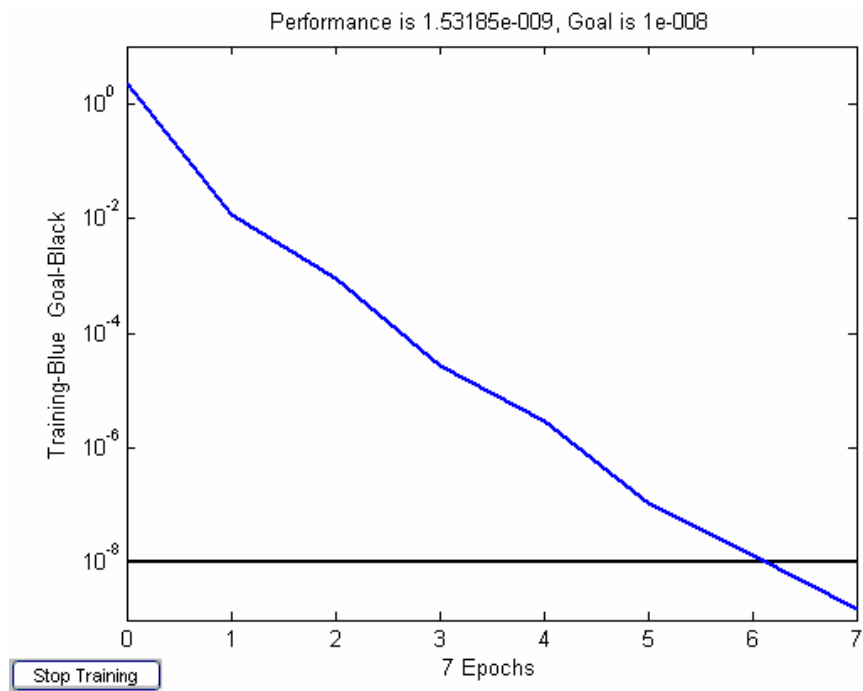


Figura 7.22 – Treinamento da rede.

O módulo percepronto3.m, como responsável pela validação, vai gerar novos símbolos e novos números distribuídos uniformemente (Figura 7.23), e sugerir a melhor correção (Figura 7.24) de acordo com o treinamento da rede.

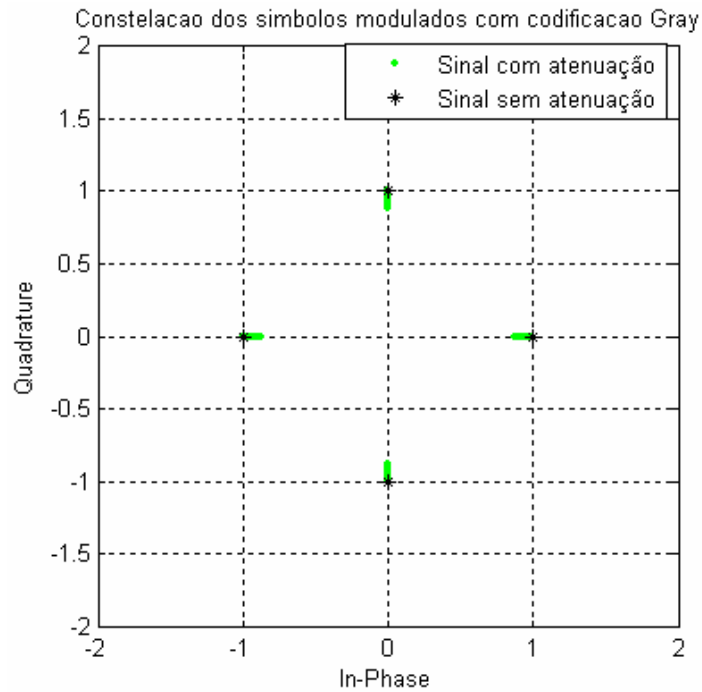


Figura 7.23 – Novos símbolos atenuados.

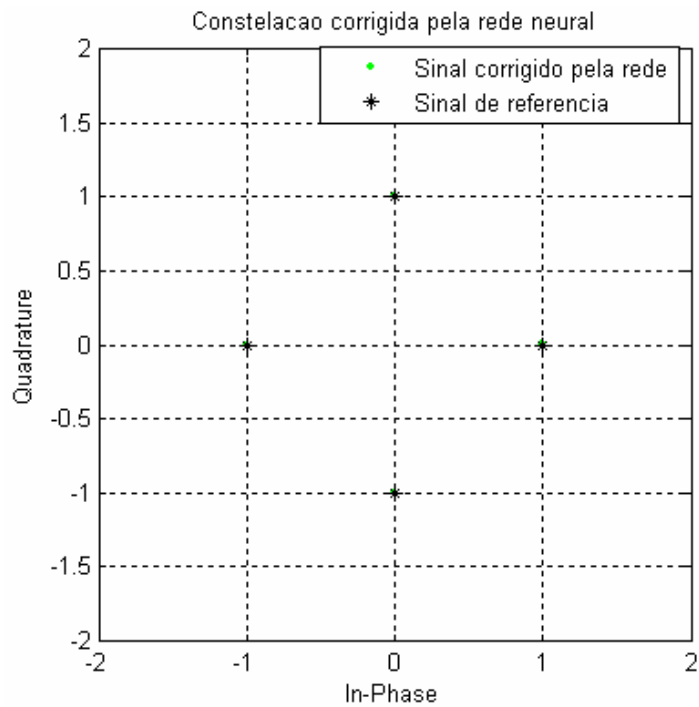


Figura 7.24 – Correção sugerida pela rede neural.

Estes foram os resultados obtidos:

- number = 0
- BER = 0
- number = 0
- SER = 0

7.3.1 – Análise imediata dos resultados

Ficou evidente que a atenuação causada por chuva tem um tratamento muito mais fácil, uma vez que, não ocorrendo deslocamento de fase neste caso, a rede neural aceita uma elevada atenuação.

7.4 – MODELANDO UM CANAL AWGN ATENUADO

Neste capítulo será feita a nossa simulação final, onde modelaremos um canal com AWGN e atenuado.

Os parâmetros a serem utilizados, são os levantados no Capítulo 6. Portanto utilizaremos um E_b/N_0 de 9 dB (obtido em 6.2) no canal AWGN, com uma atenuação causada por chuva de 2,69 dB (obtida em 6.3). Observe que, embora a atenuação causada por chuvas no dias 21 e 22/05/2009 seja nula, para um E_b/N_0 obtido de 9 dB mostrado em 6.2, julgou-se pertinente introduzir na simulação mais este grau de dificuldade. Os dados meteorológicos apontados em 6.4, serviram apenas para mostrar que o *loop* fechado entre estação terrena e satélite não sofreram atenuação causada por chuva.

Diversas simplificações foram feitas, como por exemplo, primeiro foi aplicado o AWGN em todos os símbolos gerados e após isto, foi aplicada a atenuação.

Semelhantemente ao feito anteriormente, foram desenvolvidos 3 módulos para a simulação: `simuladorqpskv7.m`, `perceptreino4.m` e `perceppronto4.m`.

7.4.1 – Testes com 1.000 símbolos

A Figura 7.25 nos mostra uma constelação com 1.000 símbolos, que serão submetidos a treinamento.

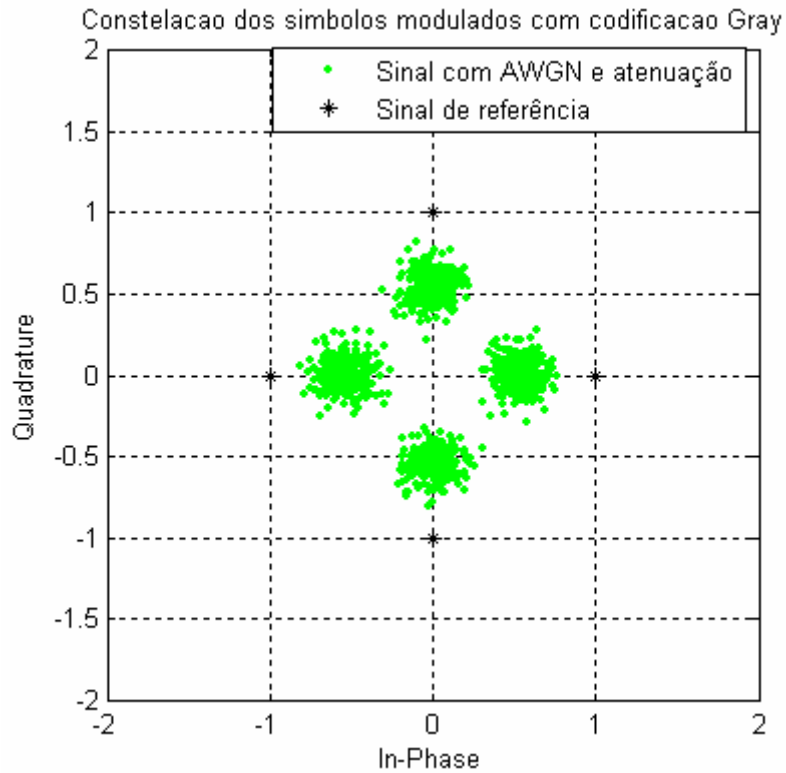


Figura 7.25 – Geração de 1.000 símbolos pseudo aleatórios.

Os símbolos gerados acima, foram submetidos a treinamento conforme mostrado pela Figura 7.26, e conforme a Figura 7.27, foram gerados novos símbolos para validar a rede.

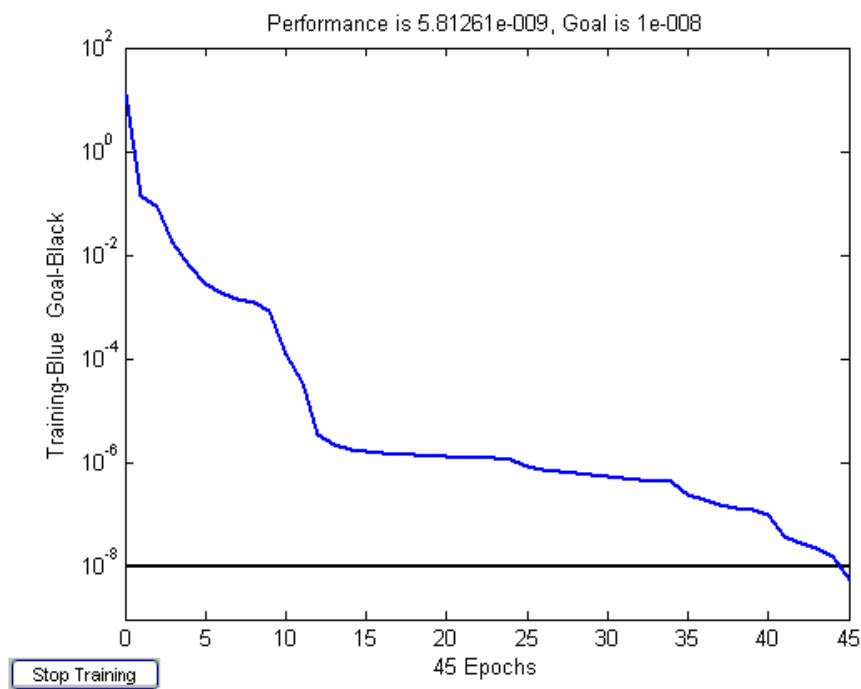


Figura 7.26 – Treinamento da rede neural com 1.000 símbolos.

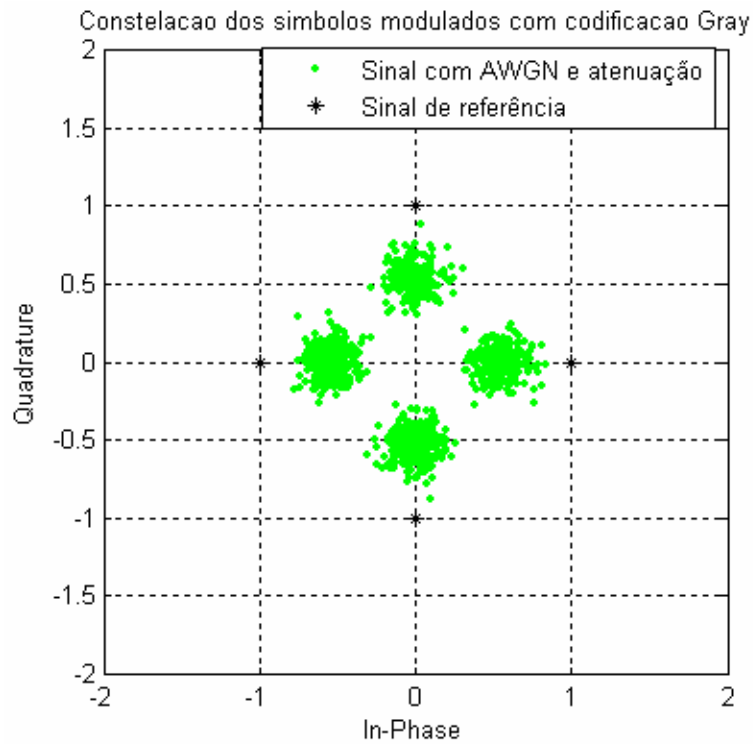


Figura 7.27 – Geração de 1.000 novos símbolos pseudo aleatórios.

Depois do treinamento, a rede neural sugeriu a correção mostrada na Figura 7.28.

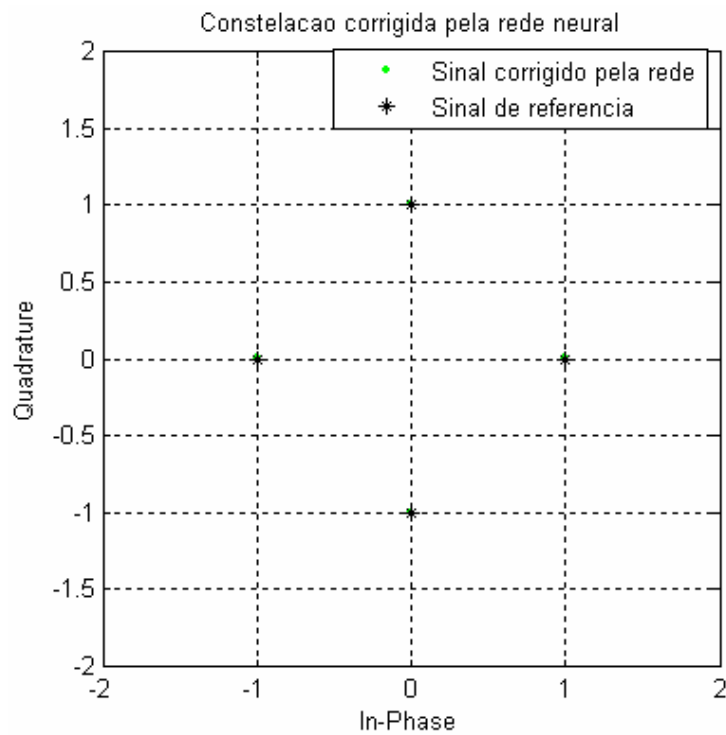


Figura 7.28 – Correção sugerida pela rede neural.

Com os seguintes índices:

- number = 0
- BER = 0
- number = 0
- SER = 0

7.4.2 – Testes com 10.000 símbolos

Nesta fase, foi notado um significativo aumento do custo computacional em todas as etapas. Isto foi decorrente, do método vetorial utilizado para aplicar a atenuação em cima dos símbolos oriundos do canal AWGN.

Desta forma, a Figura 7.29 nos mostra a geração de 10.000 símbolos a serem submetidos a treinamento conforme a Figura 7.30.

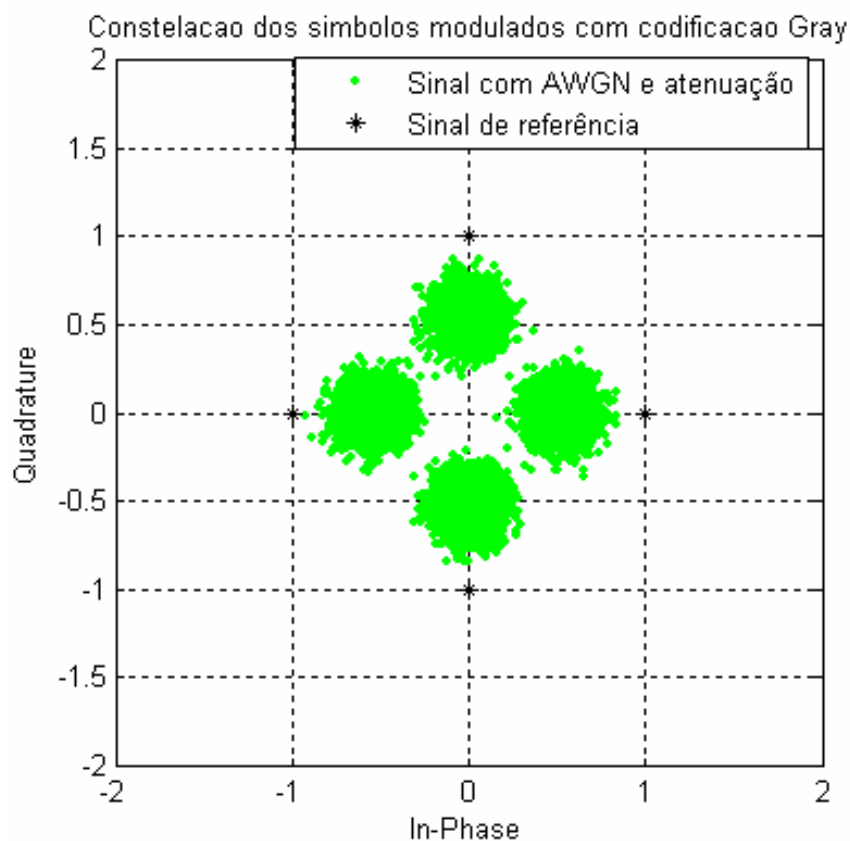


Figura 7.29 – Geração de 10.000 símbolos pseudo aleatórios.

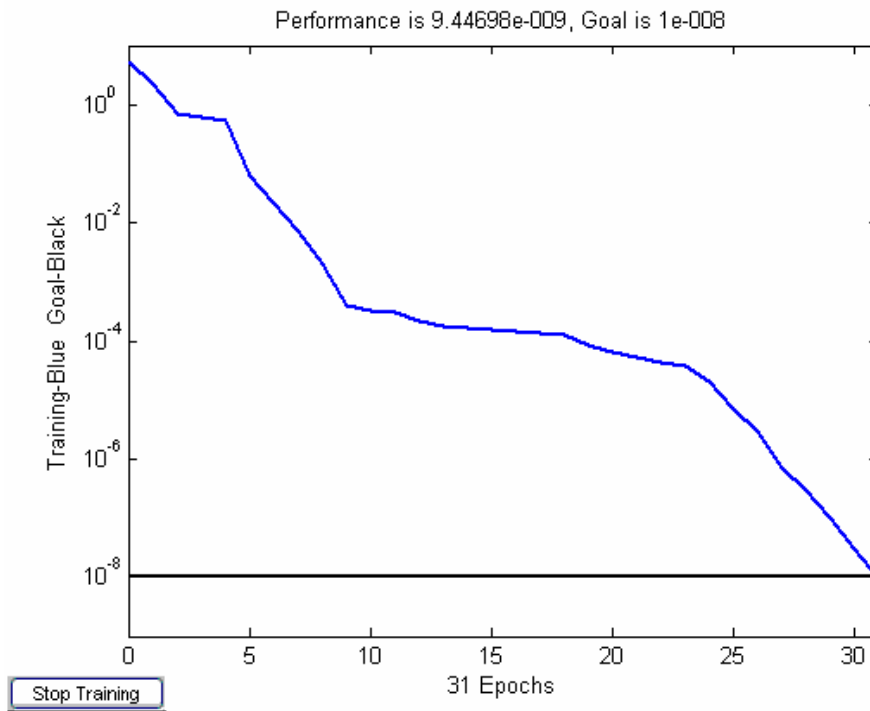


Figura 7.30 – Treinamento da rede neural com 10.000 símbolos.

Após o treinamento, segundo a Figura 7.31, foram gerados novos símbolos a fim de validar a rede neural criada.

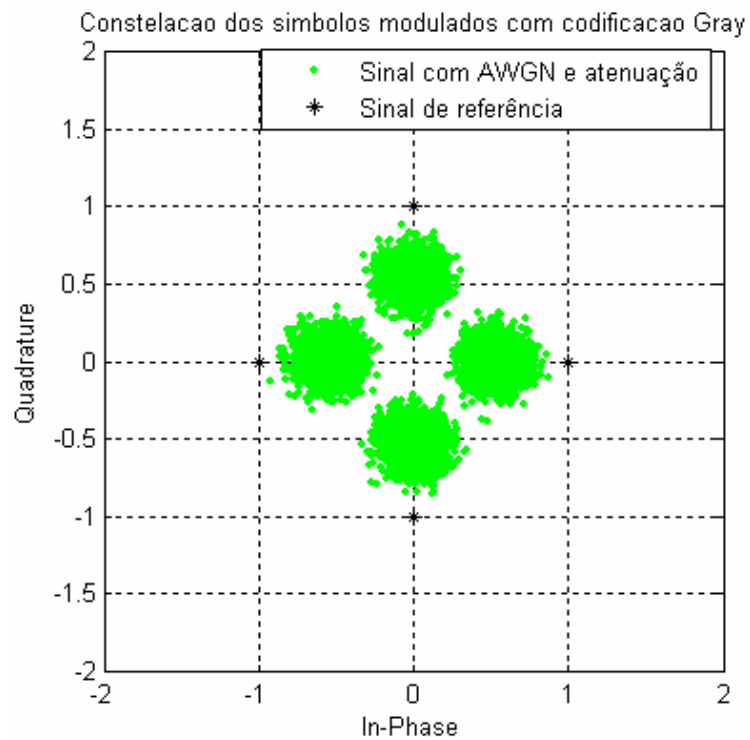


Figura 7.31 – Geração de 10.000 novos símbolos pseudo aleatórios.

A correção sugerida pela rede neural é mostrada na constelação da Figura 7.32.

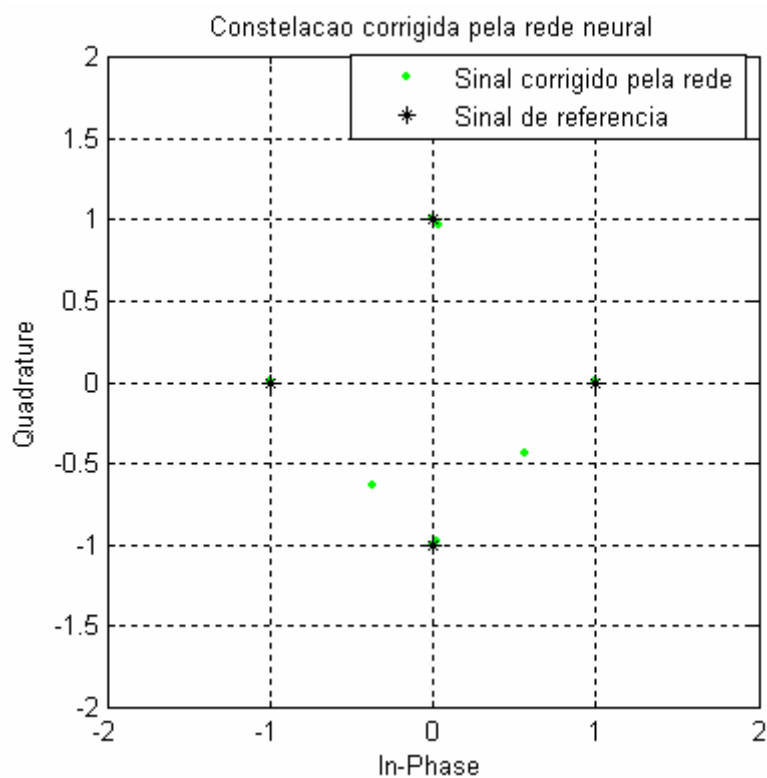


Figura 7.32 – Correção sugerida pela rede neural.

Com os seguintes índices:

- number = 4
- BER = 2,0000e-004
- number = 3
- SER = 3,0000e-004

7.4.3 – Testes com 100.000 símbolos

O custo computacional aumentou ainda mais, e cada fase da simulação demorou aproximadamente 15 minutos para ser concluído.

A Figura 7.33 nos mostra os 100.000 símbolos gerados e o posterior treinamento da rede, exibido pela Figura 7.34. Conforme feito anteriormente, utilizamos apenas os primeiros 15.000 símbolos para treinamento da rede, uma vez que um maior número de exemplos dificulta a convergência da rede, aumentando o custo computacional envolvido.

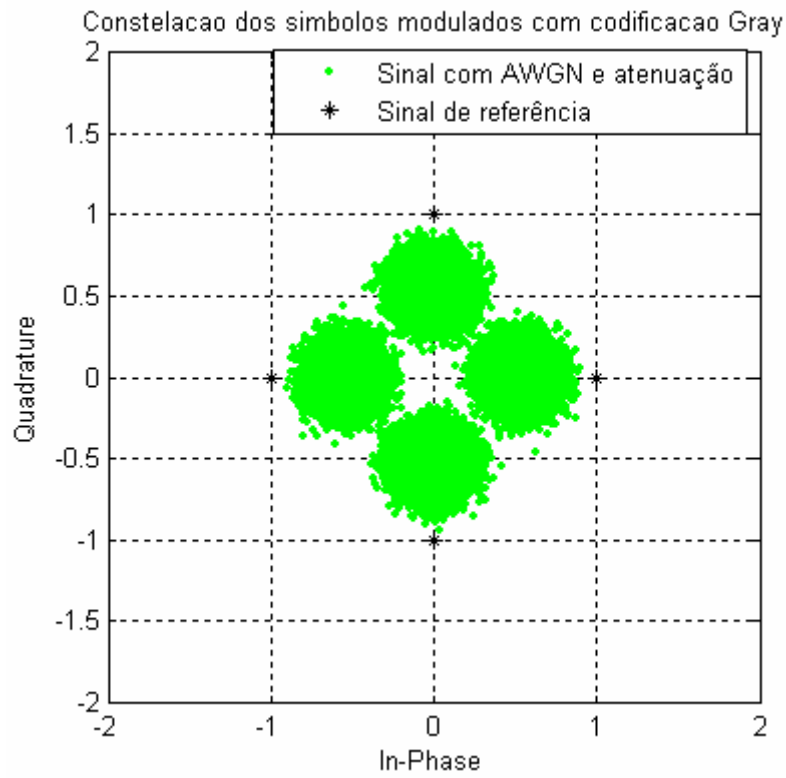


Figura 7.33 – Geração de 100.000 símbolos pseudo aleatórios.

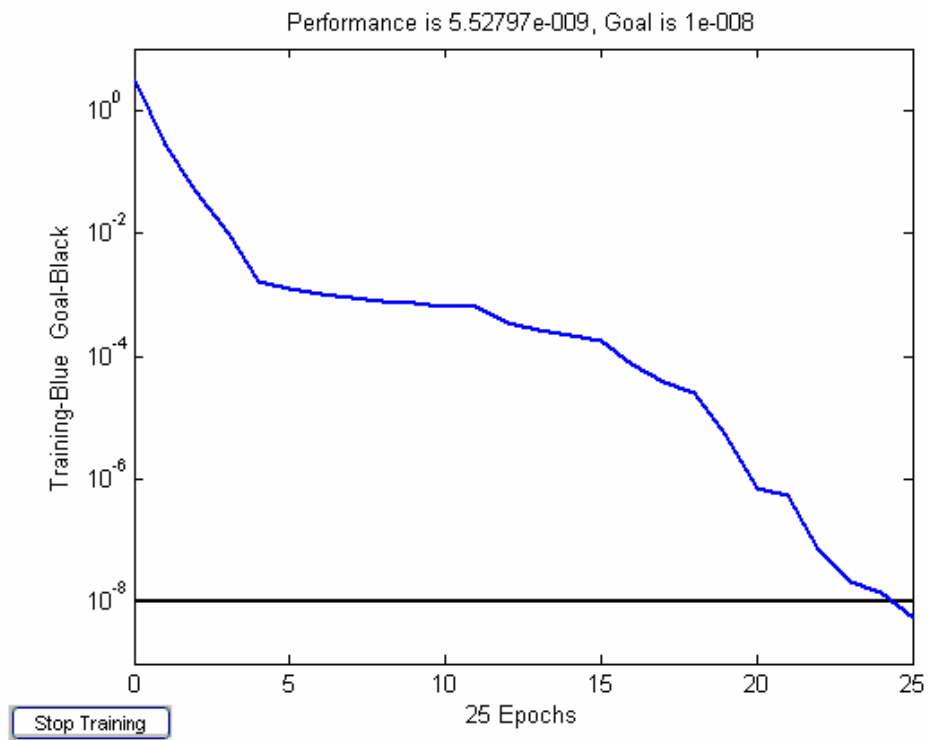


Figura 7.34 – Treinamento da rede neural com 15.000 símbolos.

A seguir, vemos na Figura 7.35 que novos símbolos são gerados, com a posterior sugestão de correção exibida pela Figura 7.36.

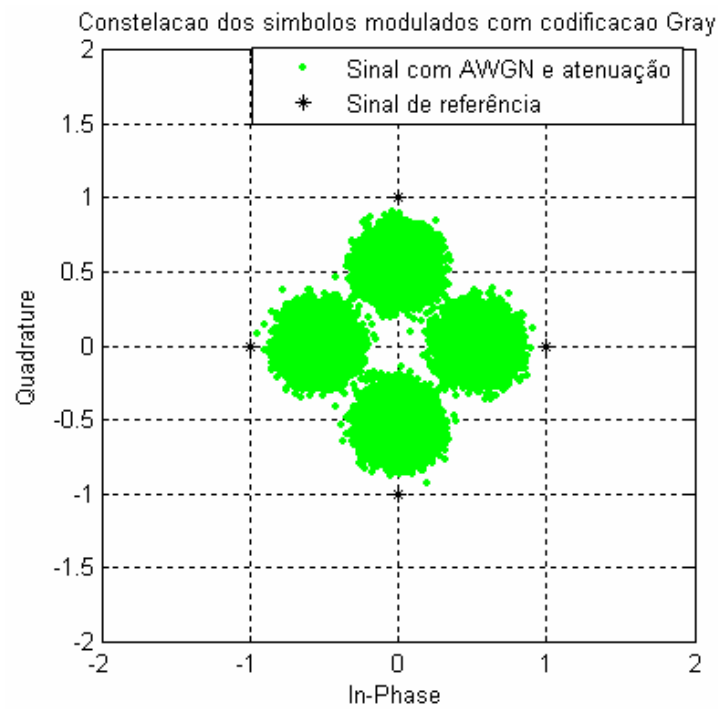


Figura 7.35 – Geração de 100.000 novos símbolos pseudo aleatórios.

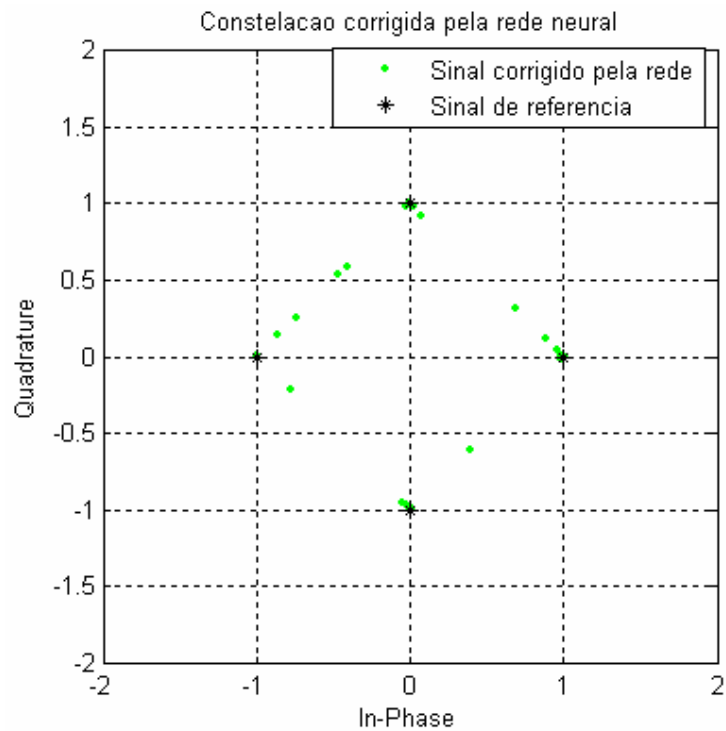


Figura 7.36 – Correção sugerida pela rede neural.

Com os seguintes índices:

- number = 21
- BER = 1,0500e-004
- number = 16
- SER = 1,6000e-004

8 - CONCLUSÕES

Conforme delimitado no início deste trabalho, o objetivo era o de concatenar conhecimentos acadêmicos e conhecimentos práticos adquiridos ao longo de anos de trabalho no CINDACTA I (Primeiro Centro Integrado de Defesa Aérea e Controle de Tráfego Aéreo). A possibilidade de extrair dados reais de um sistema de alta criticidade em produção foi essencial para uma melhor compreensão do estudo.

Para alcançar o objetivo, foi escolhido o foco em uso de redes neurais na correção de erros de transmissão em sistemas de comunicação via satélite.

Nunca foi intenção, obter a melhor rede para o cumprimento desta missão, mas sim, uma rede capaz de mostrar a viabilidade da idéia. Por isto, foi utilizada uma rede bastante simples com apenas uma camada escondida.

A dissertação foi extremamente proveitosa no sentido de proporcionar uma extensão de conhecimentos anteriormente estudados, bem como permitir a aquisição de novos conhecimentos nesta área complexa que é a de comunicações satelitais.

A rede neural foi utilizada em substituição a técnicas convencionais de demodulação, funcionando como um filtro adaptativo capaz de reconhecer e corrigir erros de fase menores que 45° .

Neste ponto, é importante deixar claro que técnicas de correção antecipada de erros (FEC), continuam sendo técnicas complementares e indispensáveis, pois elas é que permitem que os erros de fase maiores que 45° sejam detectados e corrigidos.

Embora a simulação tenha considerado apenas interferências tipo AWGN e atenuação causada por chuva, sem a utilização de técnicas de codificação do canal com FEC (*Forward Error Correction*), foi considerada bastante válida e teoricamente superior comparativamente com as técnicas atualmente em uso pelo CINDACTA I (Primeiro Centro Integrado de Defesa Aérea e Controle de Tráfego Aéreo).

O custo computacional observado nas diversas fases de geração de símbolos, treinamento da rede e validação, crescem de acordo com o aumento do campo amostral. Portanto, para o uso em um caso real, a rede terá de ser previamente treinada e customizada de acordo com

as condições de contorno, pois não há tempo disponível para um treinamento em tempo real. Em resumo, dependendo dos fenômenos interferentes levados em consideração, podemos ter diversas redes, cada uma adequada a cada caso.

Observou-se que o treinamento com 10.000 símbolos, é mais que suficiente para um adequado treinamento da rede, não sendo recomendável o uso de um número maior de símbolos, pois além de aumentar em muito o custo computacional, impede a convergência da rede neural.

Os valores finais obtidos, nos indicam que a simulação obteve um resultado superior ao do sistema de transmissão em uso pelo CINDACTA I.

Na simulação final, para 100.000 símbolos em um canal AWGN atenuado, obtivemos com a utilização da rede neural, um $BER = 1,05 \cdot 10^{-4}$ sem o uso de FEC, com um canal AWGN com $E_b/N_0 = 9$ dB e uma atenuação causada por chuva de 2,69 dB. Consultando os dados do fabricante expressos pela Tabela 2.6, verificamos que o $BER = 1,05 \cdot 10^{-4}$ corresponde a um $E_b/N_0 = 6,6$ dB.

Comparativamente falando, obtivemos para o sistema TELESAT um $BER = 1,8 \cdot 10^{-7}$ com o uso de FEC 3/4 de Viterbi em condições *clear sky*, conforme visto no Capítulo 6.2 e no Capítulo 6.4. Consultando os dados do fabricante expressos pela Tabela 2.6, verificamos que o $BER = 1,8 \cdot 10^{-7}$ corresponde a um $E_b/N_0 = 9$ dB.

Portanto, obtivemos uma diferença de apenas 2,4 dB para condições muito mais severas aplicadas na simulação. Esta comparação, por si só, diz que, caso utilizássemos um FEC, obteríamos resultados semelhantes aos do sistema TELESAT, pois em linhas gerais, o uso de FEC proporciona o mesmo efeito que um aumento de E_b/N_0 de 3 dB. No entanto, além disto, aplicamos na simulação computacional uma atenuação causada por chuva de 2,69 dB, quando as condições de medição no TELESAT eram *clear sky*!

Isto prova que, no caso estudado, o uso de redes neurais na correção de erros de transmissão em sistemas de comunicação via satélite foi bastante eficiente, chegando a superar o atual sistema em uso .

8.1 – SUGESTÕES PARA TRABALHOS FUTUROS

Para trabalhos futuros sugere-se um estudo mais amplo abordando outros efeitos interferentes, tais como a despolarização e demais efeitos não relacionados à chuva e/ou gelo.

Além disto é recomendável também o estudo em outros tipos de modulação, bem como a melhoria da rede neural.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] – Alcatel. Documentos diversos.
- [2] – ANATEL. Disponível em: <http://www.anatel.gov.br>. Acesso em 03 maio 2009.
- [3] – CPTEC / INPE. Disponível em: <http://www.cptec.inpe.br>. Acesso em 18 maio 2009.
- [4] – Cristovam, José Raimundo. “Comunicações Digitais Via Satélite”. UNISAT, 2007.
- [5] – Félix, Luís Eduardo Menezes. “Predição da Indisponibilidade por Chuvas e Redes VSAT no Brasil”. Dissertação de Mestrado, PUC-RJ, 2004.
- [6] – Garg, Vijay K. “Wireless Communications and Networking”. 1ª ed. Ed. Elsevier, 2007.
- [7] – Gilat, Amos. ”MATLAB com Aplicações em Engenharia”, tradução Glayson Eduardo de Figueiredo. 2ª ed. Ed. Bookman, 2006.
- [8] - Haykin, Simon. ” Neural Networks: a Comprehensive Foundation ”. 2ª ed. Ed. Prentice-Hall, 1999.
- [9] - Intelsat. “A Practical Introductory Guide on Using Satellite Technology for Communications”. 2002.
- [10] – ITU. Disponível em: <http://www.itu.int>. Acesso em 03 maio 2009.
- [11] – Kechriotis, G.; Zervas, E., Manolakos, E. S. “Using Recurrent Neural Networks for Adaptative Communication Channel Equalization”. IEEE Transactions on Neural Networks, Vol 5, nº 2, 1994.
- [12] – Kovács, Zsolt László. “Redes Neurais Artificiais: fundamentos e aplicações”. 4ª ed. Ed. Livraria Física, 2006.
- [13] - KUROSE, James F. “Redes de Computadores e a Internet: uma nova abordagem”. Ed. Addison Wesley, 2003.
- [14] - Maral, Gérard; Bousquet, Michel. “Satellite Communications Systems”. 4ª ed. Ed. John Wiley & Sons, 2003.
- [15] – MATLAB 7.0, version 7.0.0.19920 (R14), 2004.
- [16] – Palma Neto, Luiz Garcia et al. “Introdução às Redes Neurais Construtivas”. EdUFSCar, 2005.
- [17]–Peebles, Peyton Z. “Probability, Random Variables, and Random Signal Principles”. 3ª ed. Ed. McGraw-Hill.
- [18] – Pratt, Timothy; Bostian, Charles W.; Allnutt, Jeremy. “Satellite Communications”. 2ª ed. Ed. John Wiley & Sons, 2003.

- [19] – Proakis, John G. “Digital Communications”. 4^a ed. Ed. McGraw- Hill, 2000.
- [20] – Rappaport, Theodore S. “Wireless Communications: principles and practice”. 2^a ed. Ed. Prentice Hall, 2002.
- [21] – REDEMETS. Disponível em: <<http://redemet.aer.mil.br>>. Acesso em 18 maio 2009.
- [22] - Romariz, Alexandre. “Estudo Dirigido”, parte 3. Disponível em: <www.ene.unb.br/~romariz/sd1/edcode3.pdf>. Acesso em 07 janeiro 2009.
- [23] – Saunders, R. S. “Antennas and Propagation for Wireless Communication Systems”. Ed. Wiley, 1999.
- [24] – Silveira, Jorge Luis da. “Comunicações de Dados e Sistemas de Teleprocessamento”. Makron, McGraw-Hill, 1991.
- [25] – Spiegel, M. R. “Estatística”. tradução Carlos A. Crusius. 2^a ed. Ed. McGraw-Hill, 1985.
- [26] – Star One. Disponível em: <<http://www.starone.com.br>>. Acesso em 14 maio 2009.
- [27] – The Arthur C. Clarke Foundation. Disponível em: <<http://www.clarkefoundation.org>>. Acesso em 03 maio 2009.
- [28] – Wakerly, J. F. “Digital Design: Principles and Practices”. 3^a ed. Ed. Prentice Hall, 2000.
- [29] - Wise. Disponível em: <<http://www.wi.com.br/>>. Acesso em 25 março 2009.

APÊNDICES

A – SIMULADORQPSKV4.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%  
%% MODULO GERADOR DE AMOSTRAS%%
```

```
%gera símbolos e mostra  
clear all;  
M=4; %numero de estados possiveis para o QPSK:00 01 10 11  
gerasimbolo=150; %numero de simbolos a serem gerados  
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3  
x;%%  
figure ()  
stem ([1:gerasimbolo],x)  
axis([1,gerasimbolo,0,M-1]);  
xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de simbolos  
aleatorios');  
%mostra constelacao sem codificacao Gray com modulacao PSK  
%scatterplot (pskmod(x,M)),axis([-2,2,-2,2]),grid on;  
%title('Constelacao dos simbolos modulados sem codificacao Gray');  
%mostra constelacao com codificacao Gray com modulacao PSK  
%grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,  
transforma o 2 em 3 e o 3 em 2  
%xgray = [grayencod(x+1)]' %transforma a matriz x para matriz com codigo Gray  
%xgraymodulada = pskmod(xgray,M) %modula a matriz xgray com psk  
%scatterplot(xgraymodulada),axis([-2,2,-2,2]),grid on;  
%title('Constelacao dos simbolos modulados com codificacao Gray');  
%mostra constelacao com codificacao Gray com modulacao PSK com AWGN  
%xgraymoduladaAWGN = awgn(xgraymodulada,1,'measured')  
%scatterplot(xgraymoduladaAWGN),axis([-2,2,-2,2]),grid on;  
%title('Constelacao dos simbolos modulados com codificacao Gray e com AWGN');  
  
%mostra constelacao com codificacao Gray e modulacao QPSK sem e com AWGN  
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,  
transforma o 2 em 3 e o 3 em 2  
xgray = [grayencod(x+1)]';%% %transforma a matriz x para matriz com codigo Gray  
fid=fopen('xgray.txt','w');  
fprintf(fid,'%+5.4f\t',xgray);fclose(fid);  
xgraymodulada = pskmod(xgray,M);%% %modula a matriz xgray com psk  
xgraymoduladareal=real(xgraymodulada);%armazena parte real da xgraymodulada  
fid=fopen('xgraymoduladareal.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladareal);fclose(fid);  
xgraymoduladaimag=imag(xgraymodulada);%armazena parte imag da xgraymodulada  
fid=fopen('xgraymoduladaimag.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladaimag);fclose(fid);  
xgraymoduladaAWGN = awgn(xgraymodulada,9,'measured');%% %insere AWGN na  
xgraymodulada  
xgraymoduladaAWGNreal=real(xgraymoduladaAWGN);%armazena parte real da  
xgraymoduladaAWGN  
fid=fopen('xgraymoduladaAWGNreal.txt','w');
```



```

fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNreal);fclose(fid);
xgraymoduladaAWGNimag=imag(xgraymoduladaAWGN);%armazena parte imag da
xgraymoduladaAWGN
fid=fopen('xgraymoduladaAWGNimag.txt','w');
fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNimag);fclose(fid);
h = scatterplot(xgraymoduladaAWGN,1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao dos simbolos modulados com codificacao Gray');
legend('Sinal com AWGN','Sinal sem AWGN');
axis([-2,2,-2,2]),grid on;
hold off;

%gera simbolos com codificacao Gray com demodulacao PSK com AWGN
%xgraydemoduladaAWGN = pskdemod(xgraymoduladaAWGN,M);%%
%gera simbolos com codificacao Gray com demodulacao PSK
%xgraydemodulada = pskdemod(xgraymodulada,M);%%

%Verifica o SER (Symbol Error Rate)
%[number,ratio] = symerr(xgray,xgraymoduladaAWGN)
%number e o numero de elementos que diferem entre si antes e depois
%ratio e a razao entre number e o numero de elementos totais do vetor

%recupera simbolos demodulados sem codificacao Gray
%graydecod = bitxor(0:M-1,floor((0:M-1)/2));
%xrecuperadosemgray = [graydecod(xgraydemodulada+1)]'
%figure ()
%stem ([1:gerasimbolo],xrecuperadosemgray)
%axis([1,gerasimbolo,0,M-1]);
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de
simbolos aleatorios recuperados sem AWGN');

%recupera simbolos demodulados sem codificacao Gray com AWGN
%graydecodAWGN = bitxor(0:M-1,floor((0:M-1)/2));
%xrecuperadosemgrayAWGN = [graydecodAWGN(xgraydemoduladaAWGN+1)];%%
%figure ()
%stem ([1:gerasimbolo],xrecuperadosemgrayAWGN)
%axis([1,gerasimbolo,0,M-1]);
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de simbolos
aleatorios recuperados com AWGN');

```

B – PERCEPTREINO1.M

```
%%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%%  
%% MODULO DE TREINAMENTO%%  
%este arquivo treina uma rede perceptron  
%utiliza um algoritmo de treinamento backpropagation  
clear all;  
gerasimbolo=150;%numero de simbolos a serem gerados  
neucamadaescondida=16;%numero de neuronios da camada escondida  
fid=fopen('xgraymoduladaAWGNreal.txt');%le arquivo gerado no simuladorqpsk  
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
fid=fopen('xgraymoduladaAWGNimag.txt');%le arquivo gerado no simuladorqpsk  
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
p=[real;imag];%%  
fid=fopen('xgraymoduladareal.txt');%le arquivo gerado no simuladorqpsk  
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
fid=fopen('xgraymoduladaimag.txt');%le arquivo gerado no simuladorqpsk  
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
t=[real;imag];%%  
net=newff(minmax(p),[neucamadaescondida,2],{'tansig','purelin'},'trainlm');  
net.trainParam.show=10;  
net.trainParam.lr=0.05;  
net.trainParam.lr_inc=1.05;  
net.trainParam.epochs=1000;  
net.trainParam.goal=1e-8;  
[net,tr]=train(net,p,t);  
a=sim(net,p);%%  
pesosIW=net.IW{1,1}  
    fid=fopen('pesosIW.txt','w');  
    fprintf(fid,'%+5.4f\t',pesosIW);fclose(fid);  
bias1=net.b{1}  
    fid=fopen('bias1.txt','w');  
    fprintf(fid,'%+5.4f\t',bias1);fclose(fid);  
pesosLW=net.LW{2,1}  
    fid=fopen('pesosLW.txt','w');  
    fprintf(fid,'%+5.4f\t',pesosLW);fclose(fid);  
bias2=net.b{2}  
    fid=fopen('bias2.txt','w');  
    fprintf(fid,'%+5.4f\t',bias2);fclose(fid);
```

C – PERCEPRONTO1.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%
%% MODULO DE VALIDACAO%%
%este arquivo simula uma rede perceptron a partir de pesos obtidos do percepreino1
clear all;
nentrada=2;%numero de neuronios da entrada
nIW=16;%numero de neuronios da camada escondida
nsaida=2;%numero de neuronios da saida
M=4; %numero de estados possiveis para o QPSK:00 01 10 11
gerasimbolo=150; %numero de simbolos a serem gerados
%gera matriz de validacao
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3
%x
%figure ()
%stem ([1:gerasimbolo],x)
%axis([1,gerasimbolo,0,M-1]);
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de
simbolos aleatorios');

%mostra constelacao com codificacao Gray e modulacao QPSK sem e com AWGN
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,
transforma o 2 em 3 e o 3 em 2
xgray = [grayencod(x+1)]; %transforma a matriz x para matriz com codigo Gray
xgraymodulada = pskmod(xgray,M); %modula a matriz xgray com psk
xgraymoduladaAWGN = awgn(xgraymodulada,9,'measured');%insere AWGN na
xgraymodulada

h = scatterplot(xgraymoduladaAWGN,1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao dos simbolos modulados com codificacao Gray');
legend('Sinal com AWGN','Sinal sem AWGN');
axis([-2,2,-2,2]),grid on;
hold off;

preal=real(xgraymoduladaAWGN);
pimag=imag(xgraymoduladaAWGN);
p=[preal;pimag];%%

net=newff(minmax(p),[nIW,nsaida],{'tansig','purelin'},'traingd');
fid=fopen('pesosIW.txt');
net.IW{1,1}=fscanf(fid,'%f\t',[nIW,nentrada]);fclose(fid);
fid=fopen('pesosLW.txt');
net.LW{2,1}=fscanf(fid,'%f\t',[nsaida,nIW]);fclose(fid);
fid=fopen('bias1.txt');
net.b{1}=fscanf(fid,'%f\t',[nIW,1]);fclose(fid);
fid=fopen('bias2.txt');
net.b{2}=fscanf(fid,'%f\t',[nsaida,1]);fclose(fid);
```

```

a=sim(net,p);%%
aarredondado=round(a);
montareal=aarredondado(1,:);
montaimag=aarredondado(2,:);
aarredondado=complex(montareal,montaimag);

%mostra constelação corrigida pela rede antes do arredondamento
h = scatterplot(a',1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao corrigida pela rede neural');
legend('Sinal corrigido pela rede','Sinal de referencia');
axis([-2,2,-2,2]),grid on;
hold off;

%gera simbolos com codificacao Gray com demodulacao PSK com AWGN
%M=4;
xgraydemoduladaAWGN = pskdemod(aarredondado,M)';
%%gera simbolos com codificacao Gray com demodulacao PSK
%%xgraydemodulada = pskdemod(xgraymodulada,M)

%Verifica o SER (Symbol Error Rate)
[number,ratio] = symerr(xgray,xgraydemoduladaAWGN)
%number e o numero de elementos que diferem entre si antes e depois
%ratio e a razao entre number e o numero de elementos totais do vetor

%recupera simbolos demodulados sem codificacao Gray
%graydecod = bitxor(0:M-1,floor((0:M-1)/2));
%xrecuperadosemgray = [graydecod(xgraydemodulada+1)]'
%figure ()
%stem ([1:gerasimbolo],xrecuperadosemgray)
%axis([1,gerasimbolo,0,M-1]);
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de
simbolos aleatorios recuperados sem AWGN');

%recupera simbolos demodulados sem codificacao Gray com AWGN
graydecodAWGN = bitxor(0:M-1,floor((0:M-1)/2));
xrecuperadosemgrayAWGN = [graydecodAWGN(xgraydemoduladaAWGN+1)]';%%
figure ()
stem ([1:gerasimbolo],xrecuperadosemgrayAWGN)
axis([1,gerasimbolo,0,M-1]);
xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de simbolos
aleatorios recuperados com AWGN');

```

D – SIMULADORQPSKV5.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%  
%% MODULO GERADOR DE AMOSTRAS%%
```

```
%gera simbolos e mostra  
clear all;  
M=4; %numero de estados possiveis para o QPSK:00 01 10 11  
gerasimbolo=1000; %numero de simbolos a serem gerados  
EbNo=9;%relacao energia por bit e dens. espectral de pot. ruido  
SNR=EbNo+10*log10(2);%dado EbNo ha ajuste de 3dB para sair o SNR  
%gera matriz de validacao  
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3  
x;%%  
%figure ()  
%stem ([1:gerasimbolo],x)  
%axis([1,gerasimbolo,0,M-1]);  
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de simbolos  
aleatorios');  
  
%mostra constelacao com codificacao Gray e modulacao QPSK sem e com AWGN  
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,  
transforma o 2 em 3 e o 3 em 2  
xgray = [grayencod(x+1)];%% %transforma a matriz x para matriz com codigo Gray  
fid=fopen('xgray.txt','w');  
fprintf(fid,'%+5.4f\t',xgray);fclose(fid);  
xgraymodulada = pskmod(xgray,M);%% %modula a matriz xgray com psk  
xgraymoduladareal=real(xgraymodulada);%armazena parte real da xgraymodulada  
fid=fopen('xgraymoduladareal.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladareal);fclose(fid);  
xgraymoduladaimag=imag(xgraymodulada);%armazena parte imag da xgraymodulada  
fid=fopen('xgraymoduladaimag.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladaimag);fclose(fid);  
xgraymoduladaAWGN = awgn(xgraymodulada,SNR,'measured');%% %insere AWGN na  
xgraymodulada  
xgraymoduladaAWGNreal=real(xgraymoduladaAWGN);%armazena parte real da  
xgraymoduladaAWGN  
fid=fopen('xgraymoduladaAWGNreal.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNreal);fclose(fid);  
xgraymoduladaAWGNimag=imag(xgraymoduladaAWGN);%armazena parte imag da  
xgraymoduladaAWGN  
fid=fopen('xgraymoduladaAWGNimag.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNimag);fclose(fid);  
h = scatterplot(xgraymoduladaAWGN,1,0,'g.');
```

```
hold on;  
scatterplot(xgraymodulada,1,0,'k*',h);  
title('Constelacao dos simbolos modulados com codificacao Gray');  
legend('Sinal com AWGN','Sinal sem AWGN');  
axis([-2,2,-2,2]),grid on;
```

hold off;

E - PERCEPTREINO2.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%  
%% MODULO DE TREINAMENTO%%  
%este arquivo treina uma rede perceptron  
%utiliza um algoritmo de treinamento backpropagation  
clear all;  
gerasimbolo=1000;%numero de simbolos a serem gerados  
neucamadaescondida=16;%numero de neuronios da camada escondida  
fid=fopen('xgraymoduladaAWGNreal.txt');%le arquivo gerado no simuladorqpsk  
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
fid=fopen('xgraymoduladaAWGNimag.txt');%le arquivo gerado no simuladorqpsk  
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
p=[real;imag];%%  
fid=fopen('xgraymoduladareal.txt');%le arquivo gerado no simuladorqpsk  
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
fid=fopen('xgraymoduladaimag.txt');%le arquivo gerado no simuladorqpsk  
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);  
t=[real;imag];%%  
net=newff(minmax(p),[neucamadaescondida,2],{'tansig','purelin'},'trainlm');  
net.trainParam.show=10;  
net.trainParam.lr=0.05;  
net.trainParam.lr_inc=1.05;  
net.trainParam.epochs=500;  
net.trainParam.goal=1e-8;  
[net,tr]=train(net,p,t);  
a=sim(net,p);%%  
pesosIW=net.IW{1,1};  
fid=fopen('pesosIW.txt','w');  
fprintf(fid,'%+5.4f\t',pesosIW);fclose(fid);  
bias1=net.b{1};  
fid=fopen('bias1.txt','w');  
fprintf(fid,'%+5.4f\t',bias1);fclose(fid);  
pesosLW=net.LW{2,1};  
fid=fopen('pesosLW.txt','w');  
fprintf(fid,'%+5.4f\t',pesosLW);fclose(fid);  
bias2=net.b{2};  
fid=fopen('bias2.txt','w');  
fprintf(fid,'%+5.4f\t',bias2);fclose(fid);
```

F – PERCEPRONTO2.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%
%% MODULO DE VALIDACAO%%
%este arquivo simula uma rede perceptron a partir de pesos obtidos do percepreino1
clear all;
nentrada=2;%numero de neuronios da entrada
nIW=16;%numero de neuronios da camada escondida
nsaida=2;%numero de neuronios da saida
M=4; %numero de estados possiveis para o QPSK:00 01 10 11
gerasimbolo=1000; %numero de simbolos a serem gerados
EbNo=9;%relacao energia por bit e dens. espectral pot. ruido
SNR=EbNo+10*log10(2);%dado EbNo ha ajuste de 3 dB para sair o SNR

%gera matriz de validacao
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3
%x
%figure ()
%stem ([1:gerasimbolo],x)
%axis([1,gerasimbolo,0,M-1]);
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de
simbolos aleatorios');

%mostra constelacao com codificacao Gray e modulacao QPSK sem e com AWGN
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,
transforma o 2 em 3 e o 3 em 2
xgray = [grayencod(x+1)]; %transforma a matriz x para matriz com codigo Gray
xgraymodulada = pskmod(xgray,M); %modula a matriz xgray com psk
xgraymoduladaAWGN = awgn(xgraymodulada,SNR,'measured');%insere AWGN na
xgraymodulada

h = scatterplot(xgraymoduladaAWGN,1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao dos simbolos modulados com codificacao Gray');
legend('Sinal com AWGN','Sinal sem AWGN');
axis([-2,2,-2,2]),grid on;
hold off;

preal=real(xgraymoduladaAWGN);
pimag=imag(xgraymoduladaAWGN);
p=[preal;pimag];%%

net=newff(minmax(p),[nIW,nsaida],{'tansig','purelin'},'traingd');
fid=fopen('pesosIW.txt');
net.IW{1,1}=fscanf(fid,'%f\t',[nIW,nentrada]);fclose(fid);
fid=fopen('pesosLW.txt');
net.LW{2,1}=fscanf(fid,'%f\t',[nsaida,nIW]);fclose(fid);
fid=fopen('bias1.txt');
net.b{1}=fscanf(fid,'%f\t',[nIW,1]);fclose(fid);
```



```

fid=fopen('bias2.txt');
net.b{2}=fscanf(fid,'%f',[nsaida,1]);fclose(fid);

a=sim(net,p);%%
aarredondado=round(a);
montareal=aarredondado(1,:);
montaimag=aarredondado(2,:);
aarredondado=complex(montareal,montaimag);

%mostra constelação corrigida pela rede antes do arredondamento
h = scatterplot(a',1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao corrigida pela rede neural');
legend('Sinal corrigido pela rede','Sinal de referencia');
axis([-2,2,-2,2]),grid on;
hold off;

%gera simbolos com codificacao Gray com demodulacao PSK com AWGN
%M=4;
xgraydemoduladaAWGN = pskdemod(aarredondado,M);
%%gera simbolos com codificacao Gray com demodulacao PSK
%%xgraydemodulada = pskdemod(xgraymodulada,M)

%Verifica o BER (Bit Error Rate)
xgraybit=de2bi(xgray,'left-msb');
xgraydemoduladaAWGNbit=de2bi(xgraydemoduladaAWGN,'left-msb');
[number,BER]=biterr(xgraybit,xgraydemoduladaAWGNbit)

%Verifica o SER (Symbol Error Rate)
[number,SER] = symerr(xgray,xgraydemoduladaAWGN)
%number e o numero de simbolos que diferem entre si antes e depois
%ratio e a razao entre number e o numero de elementos totais do vetor

%recupera simbolos demodulados sem codificacao Gray com AWGN
graydecodAWGN = bitxor(0:M-1,floor((0:M-1)/2));
xrecuperadosemgrayAWGN = [graydecodAWGN(xgraydemoduladaAWGN+1)];%%
%figure ()
%stem ([1:gerasimbolo],xrecuperadosemgrayAWGN)
%axis([1,gerasimbolo,0,M-1]);
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de simbolos
aleatorios recuperados com AWGN');

```

G – SIMULADORQPSKV6.M

```
%%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%%  
%%% MODULO GERADOR DE AMOSTRAS PARA CANAL COM ATENUACAO%%%
```

```
%gera simbolos e mostra
```

```
clear all;
```

```
M=4; %numero de estados possiveis para o QPSK:00 01 10 11
```

```
gerasimbolo=1000; %numero de simbolos a serem gerados
```

```
%gera matriz de validacao
```

```
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3
```

```
%mostra constelacao com codificacao Gray e modulacao QPSK sem e com atenuação
```

```
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,  
transforma o 2 em 3 e o 3 em 2
```

```
xgray = [grayencod(x+1)];%transforma a matriz x para matriz com codigo Gray
```

```
fid=fopen('xgray.txt','w');
```

```
fprintf(fid,'%+5.4f\t',xgray);fclose(fid);
```

```
xgraymodulada = pskmod(xgray,M);%modula a matriz xgray com psk
```

```
%inserindo atenuação com numeros pseudo aleatorios distribuidos uniformemente
```

```
coeficienteatenuacaoreal=(0.1259-0)*rand(1,gerasimbolo)';
```

```
coeficienteatenuacaoimag=(0.1259-0)*rand(1,gerasimbolo)';
```

```
atenuacao=complex(coeficienteatenuacaoreal,coeficienteatenuacaoimag);
```

```
atenuacaoreal=real(atenuacao);
```

```
atenuacaoimag=imag(atenuacao);
```

```
xgraymoduladareal=real(xgraymodulada);
```

```
xgraymoduladaimag=imag(xgraymodulada);
```

```
for conta=1:1:gerasimbolo
```

```
    switch xgray(conta,1)
```

```
        case 0
```

```
            xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1)-  
atenuacaoreal(conta,1);
```

```
            xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1);
```

```
xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo  
duladaatenuadaimag(conta,1));
```

```
        case 1
```

```
            xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1);
```

```
            xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1)-  
atenuacaoimag(conta,1);
```

```
xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo  
duladaatenuadaimag(conta,1));
```

```
        case 2
```

```
xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1)+atenuacaoreal(conta,1)
```

```
;
```

```
            xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1);
```

```

xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo
duladaatenuadaimag(conta,1));
    case 3
        xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1);

xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1)+atenuacaoimag(cont
a,1);

xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo
duladaatenuadaimag(conta,1));
    end
end

xgraymoduladareal=real(xgraymodulada);%armazena parte real da xgraymodulada
fid=fopen('xgraymoduladareal.txt','w');
fprintf(fid,'%+5.4f\t',xgraymoduladareal);fclose(fid);
xgraymoduladaimag=imag(xgraymodulada);%armazena parte imag da xgraymodulada
fid=fopen('xgraymoduladaimag.txt','w');
fprintf(fid,'%+5.4f\t',xgraymoduladaimag);fclose(fid);

xgraymoduladaatenuadareal=real(xgraymoduladaatenuada);%armazena parte real da
xgraymoduladaatenuada
fid=fopen('xgraymoduladaatenuadareal.txt','w');
fprintf(fid,'%+5.4f\t',xgraymoduladaatenuadareal);fclose(fid);
xgraymoduladaatenuadaimag=imag(xgraymoduladaatenuada);%armazena parte imag da
xgraymoduladaatenuada
fid=fopen('xgraymoduladaatenuadaimag.txt','w');
fprintf(fid,'%+5.4f\t',xgraymoduladaatenuadaimag);fclose(fid);
h = scatterplot(xgraymoduladaatenuada,1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao dos simbolos modulados com codificacao Gray');
legend('Sinal com atenuação','Sinal sem atenuação');
axis([-2,2,-2,2]),grid on;
hold off;

```

H - PERCEPTREINO3.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%
%% MODULO DE TREINAMENTO CANAL COM ATENUACAO%%
%este arquivo treina uma rede perceptron
%utiliza um algoritmo de treinamento backpropagation
clear all;
gerasimbolo=1000;%numero de simbolos a serem gerados
neucamadaescondida=16;%numero de neuronios da camada escondida
fid=fopen('xgraymoduladaatenuadareal.txt');%le arquivo gerado no simuladorqpsk
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
fid=fopen('xgraymoduladaatenuadaimag.txt');%le arquivo gerado no simuladorqpsk
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
p=[real;imag];%%
fid=fopen('xgraymoduladareal.txt');%le arquivo gerado no simuladorqpsk
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
fid=fopen('xgraymoduladaimag.txt');%le arquivo gerado no simuladorqpsk
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
t=[real;imag];%%
net=newff(minmax(p),[neucamadaescondida,2],{'tansig','purelin'},'trainlm');
net.trainParam.show=10;
net.trainParam.lr=0.05;
net.trainParam.lr_inc=1.05;
net.trainParam.epochs=500;
net.trainParam.goal=1e-8;
[net,tr]=train(net,p,t);
a=sim(net,p);%%
pesosIW=net.IW{1,1};
fid=fopen('pesosIW.txt','w');
fprintf(fid,'%+5.4f\t',pesosIW);fclose(fid);
bias1=net.b{1};
fid=fopen('bias1.txt','w');
fprintf(fid,'%+5.4f\t',bias1);fclose(fid);
pesosLW=net.LW{2,1};
fid=fopen('pesosLW.txt','w');
fprintf(fid,'%+5.4f\t',pesosLW);fclose(fid);
bias2=net.b{2};
fid=fopen('bias2.txt','w');
fprintf(fid,'%+5.4f\t',bias2);fclose(fid);
```

I – PERCEPRONTO3.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%
%% MODULO DE VALIDACAO CANAL COM ATENUACAO%%
%este arquivo simula uma rede perceptron a partir de pesos obtidos do percepreino1
clear all;
nentrada=2;%numero de neuronios da entrada
nIW=16;%numero de neuronios da camada escondida
nsaida=2;%numero de neuronios da saida
M=4; %numero de estados possiveis para o QPSK:00 01 10 11
gerasimbolo=1000; %numero de simbolos a serem gerados

%gera matriz de validacao
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3

%mostra constelacao com codificacao Gray e modulacao QPSK sem e com atenuacao
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,
transforma o 2 em 3 e o 3 em 2
xgray = [grayencod(x+1)]; %transforma a matriz x para matriz com codigo Gray
xgraymodulada = pskmod(xgray,M); %modula a matriz xgray com psk
%inserindo atenuação com numeros pseudo aleatorios distribuidos uniformemente
coeficienteatenuacaoreal=(0.1259-0)*rand(1,gerasimbolo);
coeficienteatenuacaoimag=(0.1259-0)*rand(1,gerasimbolo);
atenuacao=complex(coeficienteatenuacaoreal,coeficienteatenuacaoimag);
atenuacaoreal=real(atenuacao);
atenuacaoimag=imag(atenuacao);
xgraymoduladareal=real(xgraymodulada);
xgraymoduladaimag=imag(xgraymodulada);
for conta=1:1:gerasimbolo
    switch xgray(conta,1)
        case 0
            xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1)-
atenuacaoreal(conta,1);
            xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1);

xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo
duladaatenuadaimag(conta,1));
            case 1
                xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1);
                xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1)-
atenuacaoimag(conta,1);

xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo
duladaatenuadaimag(conta,1));
            case 2

xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1)+atenuacaoreal(conta,1)
;
                xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1);
```

```

xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo
duladaatenuadaimag(conta,1));
    case 3
        xgraymoduladaatenuadareal(conta,1)=xgraymoduladareal(conta,1);

xgraymoduladaatenuadaimag(conta,1)=xgraymoduladaimag(conta,1)+atenuacaoimag(cont
a,1);

xgraymoduladaatenuada(conta,1)=complex(xgraymoduladaatenuadareal(conta,1),xgraymo
duladaatenuadaimag(conta,1));
    end
end

h = scatterplot(xgraymoduladaatenuada,1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao dos simbolos modulados com codificacao Gray');
legend('Sinal com atenuação','Sinal sem atenuação');
axis([-2,2,-2,2]),grid on;
hold off;

preal=real(xgraymoduladaatenuada)';
pimag=imag(xgraymoduladaatenuada)';
p=[preal;pimag];%%

net=newff(minmax(p),[nIW,nsaida],{'tansig','purelin'},'traingd');
fid=fopen('pesosIW.txt');
net.IW{1,1}=fscanf(fid,'%f\t',[nIW,nentrada]);fclose(fid);
fid=fopen('pesosLW.txt');
net.LW{2,1}=fscanf(fid,'%f\t',[nsaida,nIW]);fclose(fid);
fid=fopen('bias1.txt');
net.b{1}=fscanf(fid,'%f\t',[nIW,1]);fclose(fid);
fid=fopen('bias2.txt');
net.b{2}=fscanf(fid,'%f\t',[nsaida,1]);fclose(fid);

a=sim(net,p);%%
aarredondado=round(a);
montareal=aarredondado(1,:);
montaimag=aarredondado(2,:);
aarredondado=complex(montareal,montaimag);

%mostra constelação corrigida pela rede antes do arredondamento
h = scatterplot(a',1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao corrigida pela rede neural');
legend('Sinal corrigido pela rede','Sinal de referencia');
axis([-2,2,-2,2]),grid on;
hold off;

```

```

%gera simbolos com codificacao Gray com demodulacao PSK com atenuacao
xgraydemoduladaatenuada = pskdemod(aarredondado,M)';

%Verifica o BER (Bit Error Rate)
xgraybit=de2bi(xgray,'left-msb');
xgraydemoduladaatenuadabit=de2bi(xgraydemoduladaatenuada,'left-msb');
[number,BER]=biterr(xgraybit,xgraydemoduladaatenuadabit)

%Verifica o SER (Symbol Error Rate)
[number,SER] = symerr(xgray,xgraydemoduladaatenuada)
%number e o numero de simbolos que diferem entre si antes e depois
%ratio e a razao entre number e o numero de elementos totais do vetor

%recupera simbolos demodulados sem codificacao Gray com AWGN
graydecodatenuada = bitxor(0:M-1,floor((0:M-1)/2));
xrecuperadosemgrayatenuada = [graydecodatenuada(xgraydemoduladaatenuada+1)]';%%

```

J – SIMULADORQPSKV7.M

```
%%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%%  
%% MODULO GERADOR DE AMOSTRAS SIMULADORQPSKV7.M%%  
  %AWGN COM ATENUAÇÃO%  
%gera simbolos e mostra  
clear all;  
M=4; %numero de estados possiveis para o QPSK:00 01 10 11  
gerasimbolo=100000; %%numero de simbolos a serem gerados%%  
EbNo=9;%relacao energia por bit e dens. espectral de pot. ruido  
SNR=EbNo+10*log10(2);%dado EbNo ha ajuste de 3dB para sair o SNR  
%gera matriz de validacao  
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3  
x;%%  
%mostra constelacao com cod. Gray e mod. QPSK sem e com AWGN+atenuacao  
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,  
transforma o 2 em 3 e o 3 em 2  
xgray = [grayencod(x+1)];%% %transforma a matriz x para matriz com codigo Gray  
fid=fopen('xgray.txt','w');  
fprintf(fid,'%+5.4f\t',xgray);fclose(fid);  
xgraymodulada = pskmod(xgray,M);%% %modula a matriz xgray com psk  
xgraymoduladareal=real(xgraymodulada);%armazena parte real da xgraymodulada  
fid=fopen('xgraymoduladareal.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladareal);fclose(fid);  
xgraymoduladaimag=imag(xgraymodulada);%armazena parte imag da xgraymodulada  
fid=fopen('xgraymoduladaimag.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladaimag);fclose(fid);  
xgraymoduladaAWGN = awgn(xgraymodulada,SNR,'measured');%% %insere AWGN na  
xgraymodulada  
xgraymoduladaAWGNreal=real(xgraymoduladaAWGN);%armazena parte real da  
xgraymoduladaAWGN  
fid=fopen('xgraymoduladaAWGNreal.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNreal);fclose(fid);  
xgraymoduladaAWGNimag=imag(xgraymoduladaAWGN);%armazena parte imag da  
xgraymoduladaAWGN  
fid=fopen('xgraymoduladaAWGNimag.txt','w');  
fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNimag);fclose(fid);  
atenuacao=-2.69;%%da o valor da atenuacao em dB%%  
atenuacao=10^(atenuacao/10);  
montamatriz=atenuacao;  
for conta=1:1:gerasimbolo  
  atenuacao(conta,1)=montamatriz;  
end  
for conta=1:1:gerasimbolo  
  
modulo(conta,1)=sqrt((xgraymoduladaAWGNreal(conta,1))^2+(xgraymoduladaAWGNimag(conta,1))^2);  
  moduloatenuado(conta,1)=modulo(conta,1)*atenuacao(conta,1);  
end
```



```

for conta=1:1:gerasimbolo
    if xgraymoduladaAWGNreal(conta,1)>0
        if xgraymoduladaAWGNimag(conta,1)>0

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1));
            else

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1));
            end
        elseif xgraymoduladaAWGNreal(conta,1)<0
            if xgraymoduladaAWGNimag(conta,1)>0

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1))+180;
            else

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1))+180;
            end
        else
            angulo(conta,1)=0;
        end
    end
end

for conta=1:1:gerasimbolo

xgraymoduladaAWGNatenuada(conta,1)=moduloatenuado(conta,1)*[cosd(angulo(conta,1)
)+i*sind(angulo(conta,1))];
    end
xgraymoduladaAWGNatenuadareal=real(xgraymoduladaAWGNatenuada);
xgraymoduladaAWGNatenuadaimag=imag(xgraymoduladaAWGNatenuada);

    fid=fopen('xgraymoduladaAWGNatenuadareal.txt','w');%armazena parte real da
xgraymoduladaAWGN
    fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNatenuadareal);fclose(fid);
    fid=fopen('xgraymoduladaAWGNatenuadaimag.txt','w');%armazena parte imag da
xgraymoduladaAWGN
    fprintf(fid,'%+5.4f\t',xgraymoduladaAWGNatenuadaimag);fclose(fid);

h = scatterplot(xgraymoduladaAWGNatenuada,1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao dos simbolos modulados com codificacao Gray');
legend('Sinal com AWGN e atenuação','Sinal de referência');
axis([-2,2,-2,2]),grid on;
hold off;

```

K – PERCEPTREINO4.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%
%% MODULO DE TREINAMENTO AWGN com atenuacao%%
%este arquivo treina uma rede perceptron
%utiliza um algoritmo de treinamento backpropagation
clear all;
gerasimbolo=15000;%numero de simbolos a serem gerados
neucamadaescondida=16;%numero de neuronios da camada escondida
fid=fopen('xgraymoduladaAWGNatenuadareal.txt');%le arquivo gerado no simuladorqpsk
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
fid=fopen('xgraymoduladaAWGNatenuadaimag.txt');%le arquivo gerado no
simuladorqpsk
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
p=[real;imag];%%
fid=fopen('xgraymoduladareal.txt');%le arquivo gerado no simuladorqpsk
real=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
fid=fopen('xgraymoduladaimag.txt');%le arquivo gerado no simuladorqpsk
imag=fscanf(fid,'%f\t',[1,gerasimbolo]);fclose(fid);
t=[real;imag];%%
net=newff(minmax(p),[neucamadaescondida,2],{'tansig','purelin'},'trainlm');
net.trainParam.show=10;
net.trainParam.lr=0.05;
net.trainParam.lr_inc=1.05;
net.trainParam.epochs=500;
net.trainParam.goal=1e-8;
[net,tr]=train(net,p,t);
a=sim(net,p);%%
pesosIW=net.IW{1,1};
fid=fopen('pesosIW.txt','w');
fprintf(fid,'%+5.4f\t',pesosIW);fclose(fid);
bias1=net.b{1};
fid=fopen('bias1.txt','w');
fprintf(fid,'%+5.4f\t',bias1);fclose(fid);
pesosLW=net.LW{2,1};
fid=fopen('pesosLW.txt','w');
fprintf(fid,'%+5.4f\t',pesosLW);fclose(fid);
bias2=net.b{2};
fid=fopen('bias2.txt','w');
fprintf(fid,'%+5.4f\t',bias2);fclose(fid);
```

L – PERCEPRONTO4.M

```
%%SIMULADOR QPSK - DISSERTAÇÃO DE MESTRADO - SALAZAR%%
%% MODULO DE VALIDACAO AWGN com atenuacao%%
%este arquivo simula uma rede perceptron a partir de pesos obtidos do percepreino1
clear all;
nentrada=2;%numero de neuronios da entrada
nIW=16;%numero de neuronios da camada escondida
nsaida=2;%numero de neuronios da saida
M=4; %numero de estados possiveis para o QPSK:00 01 10 11
gerasimbolo=100000; %numero de simbolos a serem gerados
EbNo=9;%relacao energia por bit e dens. espectral pot. ruido
SNR=EbNo+10*log10(2);%dado EbNo ha ajuste de 3 dB para sair o SNR

%gera matriz de validacao
x=randint(gerasimbolo,1,[0,M-1]); %gera matriz [gerasimbolo x 1] com simbolos de 0 a 3
x;
%mostra constelacao com cod. Gray e mod. QPSK sem e com AWGN+atenuacao
grayencod = bitxor(0:M-1, floor((0:M-1)/2)); %lei de formacao da codificacao Gray,
transforma o 2 em 3 e o 3 em 2
xgray = [grayencod(x+1)]; %transforma a matriz x para matriz com codigo Gray
xgraymodulada = pskmod(xgray,M); %modula a matriz xgray com psk
xgraymoduladaAWGN = awgn(xgraymodulada,SNR,'measured');%insere AWGN na
xgraymodulada
xgraymoduladaAWGNreal=real(xgraymoduladaAWGN);%armazena parte real da
xgraymoduladaAWGN
fid=fopen('xgraymoduladaAWGNreal.txt','w');
fprintf(fid,'%+5.4ft',xgraymoduladaAWGNreal);fclose(fid);
xgraymoduladaAWGNimag=imag(xgraymoduladaAWGN);%armazena parte imag da
xgraymoduladaAWGN
fid=fopen('xgraymoduladaAWGNimag.txt','w');
fprintf(fid,'%+5.4ft',xgraymoduladaAWGNimag);fclose(fid);
atenuacao=-2.69;%%da o valor da atenuacao em dB%%
atenuacao=10^(atenuacao/10);
montamatriz=atenuacao;
for conta=1:1:gerasimbolo
    atenuacao(conta,1)=montamatriz;
end
for conta=1:1:gerasimbolo

modulo(conta,1)=sqrt((xgraymoduladaAWGNreal(conta,1))^2+(xgraymoduladaAWGNimag(conta,1))^2);
    moduloatenuado(conta,1)=modulo(conta,1)*atenuacao(conta,1);
end

for conta=1:1:gerasimbolo
    if xgraymoduladaAWGNreal(conta,1)>0
        if xgraymoduladaAWGNimag(conta,1)>0
```

```

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1));
    else

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1));
    end
    elseif xgraymoduladaAWGNreal(conta,1)<0
        if xgraymoduladaAWGNimag(conta,1)>0

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1))+180;
        else

angulo(conta,1)=atand(xgraymoduladaAWGNimag(conta,1)/xgraymoduladaAWGNreal(c
onta,1))+180;
        end
        else
            angulo(conta,1)=0;
        end
    end

for conta=1:1:gerasimbolo

xgraymoduladaAWGNatenuada(conta,1)=moduloatenuado(conta,1)*[cosd(angulo(conta,1)
)+i*sind(angulo(conta,1))];
end
xgraymoduladaAWGNatenuadareal=real(xgraymoduladaAWGNatenuada);
xgraymoduladaAWGNatenuadaimag=imag(xgraymoduladaAWGNatenuada);

h = scatterplot(xgraymoduladaAWGNatenuada,1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao dos simbolos modulados com codificacao Gray');
legend('Sinal com AWGN e atenuação','Sinal de referência');
axis([-2,2,-2,2]),grid on;
hold off;

preal=real(xgraymoduladaAWGNatenuada);
pimag=imag(xgraymoduladaAWGNatenuada);
p=[preal;pimag];%%

net=newff(minmax(p),[nIW,nsaida],{'tansig','purelin'},'traingd');
fid=fopen('pesosIW.txt');
net.IW{1,1}=fscanf(fid,'%f\t',[nIW,nentrada]);fclose(fid);
fid=fopen('pesosLW.txt');
net.LW{2,1}=fscanf(fid,'%f\t',[nsaida,nIW]);fclose(fid);
fid=fopen('bias1.txt');
net.b{1}=fscanf(fid,'%f\t',[nIW,1]);fclose(fid);

```

```

fid=fopen('bias2.txt');
net.b{2}=fscanf(fid,'%f',[nsaida,1]);fclose(fid);

a=sim(net,p);%%
aarredondado=round(a);
montareal=aarredondado(1,:);
montaimag=aarredondado(2,:);
aarredondado=complex(montareal,montaimag);

%mostra constelação corrigida pela rede antes do arredondamento
h = scatterplot(a',1,0,'g. ');
hold on;
scatterplot(xgraymodulada,1,0,'k*',h);
title('Constelacao corrigida pela rede neural');
legend('Sinal corrigido pela rede','Sinal de referencia');
axis([-2,2,-2,2]),grid on;
hold off;

%gera simbolos com codificacao Gray com demodulacao PSK com AWGN
%M=4;
xgraydemoduladaAWGNatenuada = pskdemod(aarredondado,M)';
%%gera simbolos com codificacao Gray com demodulacao PSK
%%xgraydemodulada = pskdemod(xgraymodulada,M)

%Verifica o BER (Bit Error Rate)
xgraybit=de2bi(xgray,'left-msb');
xgraydemoduladaAWGNbit=de2bi(xgraydemoduladaAWGNatenuada,'left-msb');
[number,BER]=biterr(xgraybit,xgraydemoduladaAWGNbit)

%Verifica o SER (Symbol Error Rate)
[number,SER] = symerr(xgray,xgraydemoduladaAWGNatenuada)
%number e o numero de simbolos que diferem entre si antes e depois
%ratio e a razao entre number e o numero de elementos totais do vetor

%recupera simbolos demodulados sem codificacao Gray com AWGN
graydecodAWGN = bitxor(0:M-1,floor((0:M-1)/2));
xrecuperadosemgrayAWGN =
[graydecodAWGN(xgraydemoduladaAWGNatenuada+1)]';%%
%figure ()
%stem ([1:gerasimbolo],xrecuperadosemgrayAWGN)
%axis([1,gerasimbolo,0,M-1]);
%xlabel('Quantidade de simbolos'),ylabel('Simbolos de 0 a 3'),title('Sequencia de simbolos
aleatorios recuperados com AWGN');

```