

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL**  
**FACULDADE DE ENGENHARIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGE**

**SINCRONISMO PARA SISTEMAS DE TRANSMISSÃO  
DIGITAL MULTIPORTADORA BASEADO EM REDES  
NEURAS ARTIFICIAIS**

**CLÓVIS RODRIGO RIBEIRO**

Porto Alegre, Junho de 2007.

**CLÓVIS RODRIGO RIBEIRO**

**SINCRONISMO PARA SISTEMAS DE TRANSMISSÃO DIGITAL  
MULTIPORTADORA BASEADO EM REDES NEURAIAS  
ARTIFICIAIS**

**Dissertação apresentada como  
requisito para obtenção do grau de  
Mestre, pelo Programa de Pós-  
Graduação em Engenharia Elétrica  
da Faculdade de Engenharia,  
Pontifícia Universidade Católica do  
Rio Grande do Sul.**

**Orientador: Prof<sup>a</sup>. Dra. Maria  
Cristina Felippetto de Castro.**

**Porto Alegre**

**2007**

**SINCRONISMO PARA SISTEMAS DE TRANSMISSÃO DIGITAL  
MULTIPORTADORA BASEADO EM REDES NEURAIAS  
ARTIFICIAIS**

**Clóvis Rodrigo Ribeiro**

Esta dissertação foi julgada para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

---

Maria Cristina Felippetto de Castro, Dra.  
Orientador

---

Daniel Coutinho, Dr.  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

---

Maria Cristina Felippetto de Castro, Dra.  
Presidente

---

Fernando César Comparsi de Castro, Dr.  
FENG – PUCRS

---

Fabício Ourique, Dr.  
FENG – PUCRS

Dedico esse trabalho a minha esposa Gislaine pela compreensão da dedicação necessária e a meu filho Rodrigo, que me distrai e alegra as poucas horas vagas que possuo.

# Agradecimentos

*Aos professores Fernando e Maria Cristina pela confiança em meu trabalho.*

*Aos meus colegas de mestrado e trabalho, pelas horas de descontração.*

*A todos que contribuíram para a conclusão dessa dissertação de forma direta ou indireta.*

*Aos meus pais pelo apoio e cobrança necessários para tornar-me um vencedor.*

*A Deus por permitir chegar ao final de mais uma grande etapa de minha vida.*

*Clóvis*

# Resumo

Um dos problemas de difícil solução em transmissão digital – e talvez o principal deles – é o sincronismo entre transmissor e receptor. Sem um sistema de sincronismo não é possível sequer iniciar o processo de demodulação no receptor. Abordaremos neste trabalho a utilização de Redes Neurais Artificiais do tipo RBF (Funções de Base Radial), para a solução de um aspecto deste problema – o sincronismo de *frame* em sistemas multiplexada, propondo, assim, um novo método de sincronismo que compararemos com o clássico método de sincronismo de *frame* por correlação. Os resultados demonstraram uma melhora considerável na FSER (*Frame Synchronization Error Rate*) do método proposto em relação ao método de sincronismo por correlação, para qualquer duração de intervalo de guarda (IG).

# Abstract

Synchronization plays a crucial role on digital transmission due to its importance for the demodulation in the receiver. This paper proposes a new OFDM frame synchronization method based on a Radial Basis Function (RBF) Artificial Neural Network (ANN). We compare the new approach with the correlation method proposed by Shinsuke Hara and Ranjee Prasad for pilot symbols detection. The obtained results have shown a significant FSER (Frame Synchronization Error Rate) increase for any cyclic prefix.

# Sumário

<b>1. INTRODUÇÃO.....</b>	<b>12</b>
<b>2. SISTEMAS DE COMUNICAÇÃO DIGITAL.....</b>	<b>14</b>
2.1. INTRODUÇÃO .....	14
2.2. <i>ADVANCED TELEVISION SYSTEMS COMMITTEE – ATSC [4]</i> .....	15
2.2.1. <i>Codificador</i> .....	16
2.2.1.1. <i>Sincronizador de Quadro</i> .....	16
2.2.1.2. <i>Aleatorizador de Dados</i> .....	16
2.2.1.3. <i>Codificador Reed Solomon</i> .....	16
2.2.1.4. <i>Entrelaçador de Dados</i> .....	17
2.2.1.5. <i>Codificador de Treliça</i> .....	17
2.2.2. <i>Estruturador de Quadro</i> .....	18
2.2.2.1. <i>Multiplexador</i> .....	18
2.2.2.2. <i>Insersor de Pilotos</i> .....	19
2.2.3. <i>Modulador 8VSB</i> .....	20
2.2.3.1. <i>Modulador AM-DSB-SC (Amplitude Modulated – Double Side Band-Suppressed Carrier)</i> 20	
2.2.3.2. <i>Filtro de VSB (Vestigial Side Band)</i> .....	20
2.2.3.3. <i>Filtro de Nyquist</i> .....	20
2.3. <i>DIGITAL VIDEO BROADCASTING – DVB [5]</i> .....	21
2.3.1. <i>Codificador</i> .....	22
2.3.1.1. <i>Divisor de Sinais (Splitter)</i> .....	22
2.3.1.2. <i>Adaptação do MUX e Dispersão de Energia</i> .....	22
2.3.1.3. <i>Codificador Externo</i> .....	22
2.3.1.4. <i>Entrelaçador Externo</i> .....	23
2.3.1.5. <i>Codificador Interno</i> .....	23
2.3.1.6. <i>Entrelaçador Interno (Inner Interleaver)</i> .....	24
2.3.1.6.1. <i>Demultiplexador</i> .....	24
2.3.1.6.2. <i>Interleaver de Bit</i> .....	25
2.3.1.6.3. <i>Interleaver de Símbolo</i> .....	25
2.3.2. <i>Estrutura do Quadro</i> .....	25
2.3.2.1. <i>Mapeador</i> .....	26
2.3.2.2. <i>Adaptação de Quadro</i> .....	27
2.3.3. <i>Modulador</i> .....	27
2.3.3.1. <i>Modulador OFDM</i> .....	27
2.3.3.2. <i>Inserção do Intervalo de Guarda</i> .....	28
2.3.3.3. <i>Conversor D/A (Digital/Analógico)</i> .....	28
2.4. <i>INTEGRATED SERVICES DIGITAL BROADCASTING – ISDB [6]</i> .....	29
2.4.1. <i>TS Remultiplexador</i> .....	30
2.4.2. <i>Codificação de canal</i> .....	31
2.4.2.1. <i>Codificador Reed Solomon</i> .....	31
2.4.2.2. <i>Divisor de TS</i> .....	32
2.4.2.3. <i>Dispensor de energia</i> .....	32
2.4.2.4. <i>Ajuste de Delay</i> .....	32
2.4.2.5. <i>Interleaver de Byte</i> .....	32
2.4.2.6. <i>Código Convolutacional</i> .....	33
2.4.3. <i>Modulação</i> .....	33
2.4.3.1. <i>Interleaver de Bit e Mapeador</i> .....	33
2.4.3.2. <i>Combinador de Níveis Hierárquicos</i> .....	35
2.4.3.3. <i>Interleaver de Tempo</i> .....	36
2.4.3.4. <i>Interleaver de Freqüência</i> .....	37

2.4.3.5.	<i>Estruturação do Frame</i> .....	39
2.4.3.6.	<i>Intervalo de Guarda</i> .....	39
2.5.	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO .....	39
<b>3.</b>	<b>REDES NEURAIAS ARTIFICIAIS .....</b>	<b>40</b>
3.1.	INTRODUÇÃO .....	40
3.2.	<i>RADIAL BASIS FUNCTION – RBF</i> .....	42
3.2.1.	<i>Estrutura da RBF</i> .....	42
3.2.2.	<i>Heurística de Treino de uma RNA RBF</i> .....	43
3.3.	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO .....	47
<b>4.</b>	<b>CANAIS DE TRANSMISSÃO WIRELESS.....</b>	<b>48</b>
4.1.	CANAIS <i>BRAZIL</i> .....	49
4.1.1.	<i>Canal Brazil A</i> .....	49
4.1.2.	<i>Canal Brazil B</i> .....	50
4.1.3.	<i>Canal Brazil C</i> .....	51
4.1.4.	<i>Canal Brazil D</i> .....	52
4.1.5.	<i>Canal Brazil E</i> .....	53
<b>5.</b>	<b>DESENVOLVIMENTO .....</b>	<b>55</b>
5.1.	INTRODUÇÃO .....	55
5.2.	PREÂMBULO.....	55
5.3.	DEFINIÇÃO DA RNA .....	57
5.4.	MÉTODO PARA VALIDAÇÃO.....	57
5.5.	LOCALIZAÇÃO DO INÍCIO DO <i>FRAME</i> .....	58
5.6.	RESULTADOS OBTIDOS .....	59
<b>6.</b>	<b>CONCLUSÃO .....</b>	<b>72</b>
<b>7.</b>	<b>BIBLIOGRAFIA .....</b>	<b>73</b>

# Lista de Figuras

FIGURA 2.1 – DIAGRAMA DE BLOCOS DO SISTEMA ATSC [4] .....	15
FIGURA 2.2 – ENTRELAÇADOR DE DADOS [4] .....	17
FIGURA 2.3 – ENTRADA E SAÍDA DOS DADOS DO BLOCO CODIFICADOR DE TRELIÇA [4] .....	18
FIGURA 2.4 – ESTRUTURA DO <i>FRAME</i> .....	19
FIGURA 2.5 – FILTRO DE INTERFERÊNCIA, CODIFICADOR DE TRELIÇA E MAPEADOR DE SÍMBOLOS 8VSB .....	19
FIGURA 2.6 – RESPOSTA DO FILTRO DE NYQUIST .....	20
FIGURA 2.7 – DIAGRAMA EM BLOCOS DO SISTEMA DVB [5] .....	21
FIGURA 2.8 – FIGURA DO <i>INTERLEAVER E DEINTERLEAVER</i> [5] .....	23
FIGURA 2.9 – ENTRELAÇADOR INTERNO PARA MODULAÇÃO QPSK [5] .....	24
FIGURA 2.10 – ENTRELAÇADOR INTERNO PARA MODULAÇÃO 16QAM [5] .....	24
FIGURA 2.11 – ENTRELAÇADOR INTERNO PARA MODULAÇÃO 64QAM [5] .....	25
FIGURA 2.12 – CONSTELAÇÃO QPSK [5] .....	26
FIGURA 2.13 – CONSTELAÇÃO 16QAM [5] .....	26
FIGURA 2.14 – CONSTELAÇÃO 64QAM [5] .....	27
FIGURA 2.15 – FIGURA DO SÍMBOLO OFDM COM O INTERVALO DE GUARDA .....	28
FIGURA 2.16 – FIGURA DA SEGMENTAÇÃO DO CANAL DE 6MHZ .....	29
FIGURA 2.17 – DIAGRAMAS DE BLOCO DO SISTEMA ISDB [6] .....	30
FIGURA 2.18 – SISTEMA ISDB DIVIDIDO EM TRÊS BLOCOS .....	30
FIGURA 2.19 – TS'S MULTIPLEXADOS (TSP) .....	31
FIGURA 2.20 – PACOTES DE SAÍDA DO CODIFICADOR <i>REED SOLOMON</i> .....	31
FIGURA 2.21 – INTERLEAVER DE BIT PARA MODULAÇÃO DQPSK .....	33
FIGURA 2.22 – MAPEAMENTO PARA UMA MODULAÇÃO DQPSK .....	33
FIGURA 2.23 – INTERLEAVER DE BIT PARA MODULAÇÃO QPSK .....	34
FIGURA 2.24 – MAPEAMENTO PARA UMA MODULAÇÃO QPSK .....	34
FIGURA 2.25 – INTERLEAVER DE BIT PARA MODULAÇÃO 16QAM .....	34
FIGURA 2.26 – MAPEAMENTO PARA UMA MODULAÇÃO 16QAM .....	34
FIGURA 2.27 – INTERLEAVER DE BIT PARA MODULAÇÃO 64QAM .....	35
FIGURA 2.28 – MAPEAMENTO PARA UMA MODULAÇÃO 64QAM .....	35
FIGURA 2.29 – COMBINADOR DE NÍVEIS HIERÁRQUICOS .....	36
FIGURA 2.30 – INTERLEAVER DE TEMPO .....	37
FIGURA 2.31 – GRÁFICO DE DISPERSÃO DAS PORTADORAS DENTRO DE UM SEGMENTO NO MODO 2K .....	38
FIGURA 2.32 – GRÁFICO DE DISPERSÃO DAS PORTADORAS DENTRO DE UM SEGMENTO NO MODO 4K .....	38
FIGURA 2.33 – GRÁFICO DE DISPERSÃO DAS PORTADORAS DENTRO DE UM SEGMENTO NO MODO 8K .....	38
FIGURA 3.1 – MODELO DE UM NEURÔNIO NÃO-LINEAR [7] .....	41
FIGURA 3.2 – ARQUITETURA DE UMA REDE RBF [7] .....	42
FIGURA 4.1 – MAGNITUDE DA RESPOSTA AO IMPULSO DO CANAL <i>BRAZIL A</i> NO DOMÍNIO DO TEMPO. ....	50
FIGURA 4.2 – MAGNITUDE DA RESPOSTA AO IMPULSO DO CANAL <i>BRAZIL B</i> NO DOMÍNIO DO TEMPO. ....	51
FIGURA 4.3 – MAGNITUDE DA RESPOSTA AO IMPULSO DO CANAL <i>BRAZIL C</i> NO DOMÍNIO DO TEMPO. ....	52
FIGURA 4.4 – MAGNITUDE DA RESPOSTA AO IMPULSO DO CANAL <i>BRAZIL D</i> NO DOMÍNIO DO TEMPO. ....	53
FIGURA 4.5 – MAGNITUDE DA RESPOSTA AO IMPULSO DO CANAL <i>BRAZIL E</i> NO DOMÍNIO DO TEMPO. ....	54
FIGURA 5.1 – PREÂMBULO NO TEMPO COM 64 PONTOS .....	56
FIGURA 5.2 – PLATÔ DO INTERVALO DE GUARDA DO SÍMBOLO DE TAMANHO 1/4 .....	58
FIGURA 5.3 – CURVA DE FSER DA IDENTIFICAÇÃO POR RNA E POR CORRELAÇÃO PARA IGS E SNR DIFERENTES .....	59
FIGURA 5.4 – MONTAGEM DA JANELA DA RNA .....	61
FIGURA 5.5 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL A</i> .....	62
FIGURA 5.6 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL B</i> .....	62
FIGURA 5.7 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL C</i> .....	63
FIGURA 5.8 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL D</i> .....	63
FIGURA 5.9 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL E</i> .....	64

FIGURA 5.10 – PRIMEIRO PICO DA FIGURA 5.5 .....	65
FIGURA 5.11 – SEGUNDO PICO DA FIGURA 5.5.....	66
FIGURA 5.12 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL</i> A IDENTIFICANDO TODOS IGS. ....	67
FIGURA 5.13 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL</i> B IDENTIFICANDO TODOS IGS.....	67
FIGURA 5.14 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL</i> C IDENTIFICANDO TODOS IGS.....	68
FIGURA 5.15 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL</i> D IDENTIFICANDO TODOS IGS. ....	68
FIGURA 5.16 – SAÍDA DA RNA PARA O CANAL <i>BRAZIL</i> E IDENTIFICANDO TODOS IGS. ....	69
FIGURA 5.17 – POSIÇÃO INICIAL DO PRIMEIRO SÍMBOLO OFDM DO CANAL <i>BRAZIL</i> A.....	70
FIGURA 5.18 – POSIÇÃO INICIAL DO SEGUNDO SÍMBOLO OFDM DO CANAL <i>BRAZIL</i> A.....	70

# Lista de Tabelas

Tabela 3.1 – Possíveis Algoritmos de Aprendizado de uma rede RBF

Tabela 3.2 – Heurística de treino de uma RNA

Tabela 4.1 – Multipercursos do canal *Brazil A*

Tabela 4.2 – Multipercursos do canal *Brazil B*

Tabela 4.3 – Multipercursos do canal *Brazil C*

Tabela 4.4 – Multipercursos do canal *Brazil D*

Tabela 4.5 – Multipercursos do canal *Brazil E*

# Lista de Acrônimos

8VSB	<i>8 Vestigial Side band</i>
ANN	<i>Artificial Neural Network</i>
AM-DSB-SC	<i>Amplitude Modulated – Double Side Band-Suppressed Carrier</i>
ATSC	<i>Advanced Television Systems Committee</i>
BER	<i>bit error rate</i>
COFDM	<i>Codec Orthogonal Frequency Division Multiplex</i>
DQPSK	<i>Differential Quadrature Phase-Shift Keying</i>
DVB	<i>Digital Video Broadcasting</i>
DTV	<i>Digital Television</i>
FEC	<i>Forward Error Correction Code</i>
FSER	<i>Frame Synchronization Error Rate</i>
HDTV	<i>High-Definition Television</i>
IFFT	<i>Inverse Fast Fourier Transformer</i>
IQ	<i>In-Phase/Quadrature</i>
ISDB	<i>Integrated Services Digital Broadcasting</i>
OFDM	<i>Orthogonal frequency-division multiplexing</i>
PRBS	<i>Pseudo Random Binary Sequence</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QPSK	<i>Quadrature Phase-Shift Keying</i>
RBF	<i>Radial Basis Function</i>
RNA	Rede Neural Artificial

TPS      *Transmission Parameter Signals*

TS      *Transport Stream*

TVD      *Televisão Digital*

VSB      *Vestigial Side band*

# 1. Introdução

Devido à crescente digitalização dos meios de comunicações e à cada vez maior utilização de técnicas de comunicação digital, surge a necessidade de pesquisas consistentes nesta área, objetivando a contínua melhora do desempenho dos sistemas. Com isto, diminui-se os custos de implantação, fazendo com que esta tecnologia seja acessível à população em geral, ocasionando assim a chamada inclusão digital.

Aproveitando este período de transição em que o país está passando, onde o Governo Brasileiro está estabelecendo uma política de investimento em pesquisas nas mais diversas áreas a fim de obter o *expertise* necessário para o desenvolvimento de produtos com tecnologia nacional, e com possibilidade de exportar tal tecnologia ou produto baseado nesta, apresentamos esta proposta de inovação no sistema de sincronismo para qualquer sistema digital *wireless* multiportadora.

De fato, a idéia surgiu em estudos sobre televisão digital multiportadora (*Orthogonal frequency-division multiplexing* - OFDM), em que nos deparamos com o problema de sincronismo de *frame*.

A presente pesquisa buscou projetar um sistema de sincronismo para comunicação digital multiportadora com a intenção de diminuir a quantidade de símbolos utilizados para sincronizar o processo dentro de um *frame*, aumentando assim a taxa de transmissão de informação.

Para isto elaboraremos um novo cabeçalho para este *frame*, garantindo que haja um padrão que possa ser identificado por uma Rede Neural Artificial (RNA), além de manter informações de configuração do receptor sem a necessidade de portadoras pilotos para a fase de configuração.

A validação dos resultados foi através da comparação com o método de correlação citado por Shinsuke Hara & Ranjee Prasad [1], van Nee, R. & R. Prasad [2] e Proakis, J. G. [3], expondo ambos os métodos as mesmas condições de simulação do meio de transmissão.

No Capítulo 2 será apresentado uma breve introdução aos sistemas de comunicação digital de televisão já existentes, com foco nos sistemas baseados em modulação OFDM, para o qual esta pesquisa foi desenvolvida. No Capítulo 3 veremos como funciona uma rede neural do tipo RBF, a qual viabilizou o desenvolvimento do método, tendo o Capítulo 4 as especificações dos canais de transmissão utilizados para validar o método.

No Capítulo 5 demonstraremos como foi desenvolvido o método de sincronismo baseado em redes neurais artificiais. O Capítulo 6 apresenta nossas conclusões.

## 2. Sistemas de Comunicação Digital

### 2.1. Introdução

A evolução das comunicações sempre foi a principal meta de pesquisas militares, mais intensamente após as guerras mundiais, onde se percebeu que por melhor que fossem os “enigmas” aplicados à informação transmitida através de comunicação analógica, esta não garantia a segurança da informação, a qual podia ser decodificada com um pouco de tempo, dedicação e concentração dos responsáveis por esta tarefa. Por isso houve a necessidade de aprimorar e criar um modelo de comunicação que não pudesse ser decodificada facilmente pelo adversário – eis que começam a surgir as técnicas de comunicação digital, baseadas na teoria digital e com palavras de encriptação que chegam hoje a 128 bits ou mais.

Conforme as pesquisas na área aumentavam, o que continua até hoje, aquilo que já não era mais novidade passava a conhecimento público e transformado em tecnologia do dia a dia para as pessoas das grandes metrópoles. Exemplo típico deste processo é o que ocorreu com a telefonia celular.

Atualmente a tecnologia de comunicação digital está presente em desde um simples controle remoto de televisão até os complicados protocolos de seguranças bancários via Internet. Outra aplicação que agora esta chegando ao Brasil para facilitar o acesso à educação da população é o sistema de transmissão digital de televisão, que é capaz de receber dados das transmissoras, e também enviar dados para elas através de seus canais de interatividade.

Hoje no mundo existem três modelos de transmissão de televisão digital consagrados, o *Advanced Television Systems Committee* (ATSC) [4], que é o sistema Norte Americano, o *Digital Video Broadcasting* (DVB) [5], que é o sistema Europeu e o mais robusto deles que é o *Integrated Services Digital Broadcasting* (ISDB) [6], sistema Japonês, que foi adotado recentemente pelo Brasil como base de seu sistema de Televisão Digital (TVD).

No próximo capítulo veremos como funcionam estes sistemas, apontando fatores relevantes para a execução deste trabalho.

## 2.2. Advanced Television Systems Committee – ATSC [4]

Apesar do sistema ATSC não ser multiportadora (OFDM), o método proposto neste trabalho pode ser adaptado facilmente para resolver o problema de sincronismo deste – basta colocarmos uma seqüência conhecida no início de um *frame* e treinarmos a rede para identificá-la.

As primeiras transmissões de TV digital foram em 1998, quando os Estados Unidos adotaram o ATSC, que foi o primeiro sistema de televisão digital a ser desenvolvido no mundo, com a característica de ser um sistema mono portadora (portadora única) com modulação por amplitude 8 *Vestigial Sideband* (8VSB). O ATSC ocupa os mesmos 6 MHz de banda de um canal do sistema analógico, porém, não é um sistema tão robusto em relação a seus concorrentes DVB e ISDB que são sistemas multiportadoras (OFDM), além de não possuir até 2007 possibilidade de recepção móvel. Também é o sistema menos complexo entre os três citados. Na Figura 2.1 temos o diagrama em blocos do codificador ATSC.

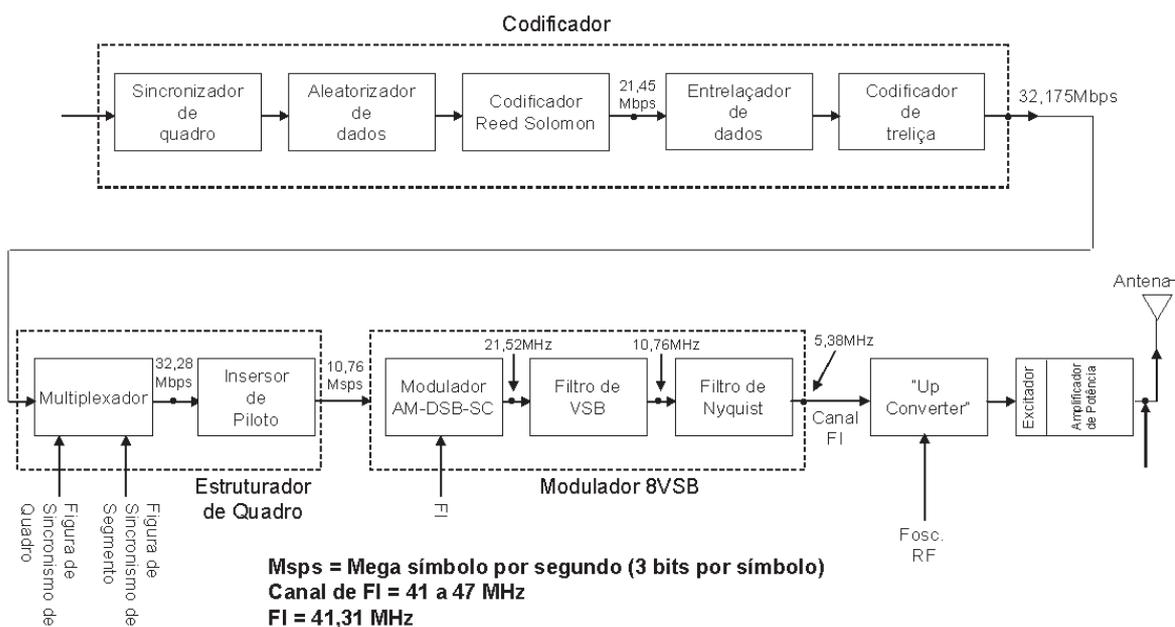


Figura 2.1 – Diagrama de Blocos do sistema ATSC [4]

O codificador de canal ATSC é composto pelos blocos do Sincronizador de Quadro, Aleatorizador de Dados (*Randomizer*), Codificador *Reed Solomon*, Entrelaçador de Dados (*Interleaver*) e o Codificador de Treliça (*Trellis Encoder*).

O estruturador de Quadro é composto pelo Multiplexador e o Inserir de Pilotos, já a etapa de modulação é composto pelo bloco do Modulador AM-DSB-SC (*Amplitude Modulated – Double Side Band – Suppressed Carrier*), Filtro VSB e o Filtro de *Nyquist*.

## **2.2.1. Codificador**

### **2.2.1.1. Sincronizador de Quadro**

Neste bloco chegam pacotes seqüenciais de 188 bytes, dentre os quais 1 byte serve de sincronismo para estes pacotes. A função do Sincronizador de quadro é retirar o byte de sincronismo que não é útil ao resto do sistema, antes de entregá-los ao aleatorizador de dados.

### **2.2.1.2. Aleatorizador de Dados**

A principal função deste bloco é aleatorizar os bytes dos pacotes recebidos do Sincronizador de quadros, garantindo com que não haja energia concentrada em determinadas frequências do espectro, causada por repetições de padrões de dados.

Este bloco é constituído basicamente por um *Pseudo Random Binary Sequence* (PRBS) [4][5][6].

### **2.2.1.3. Codificador *Reed Solomon***

Este bloco permite ao receptor identificar seqüências de erros no sinal demodulado e conseguir corrigi-los. O Codificador *Reed Solomon* é um *Forward Error Correction Code* (FEC) da família *Bloc Codes* do tipo BCH [15].

No *Reed Solomon* para cada conjunto de 187 bytes de dados que recebe em sua entrada, ele adiciona 20 bytes de redundância, que possibilitará ao receptor corrigir até mesmo 10 bytes errados em cada conjunto de 207 bytes compostos pela sua saída.

### 2.2.1.4. Entrelaçador de Dados

O entrelaçador de Dados (Interleaver) é responsável em embaralhar os bytes da saída do Codificador *Reed Solomon*, garantindo que se houver uma interferência causada por um ruído impulsivo (por exemplo) no canal de transmissão, o seu dual também chamado *De-Interleaver* no receptor, possa espalhar os erros causados por este ruído não permitindo erros seqüenciais, os quais prejudicam o desempenho do código corretor de erros (*Reed Solomon*) [4][5][16].

A Figura 2.2 mostra como funciona este bloco.

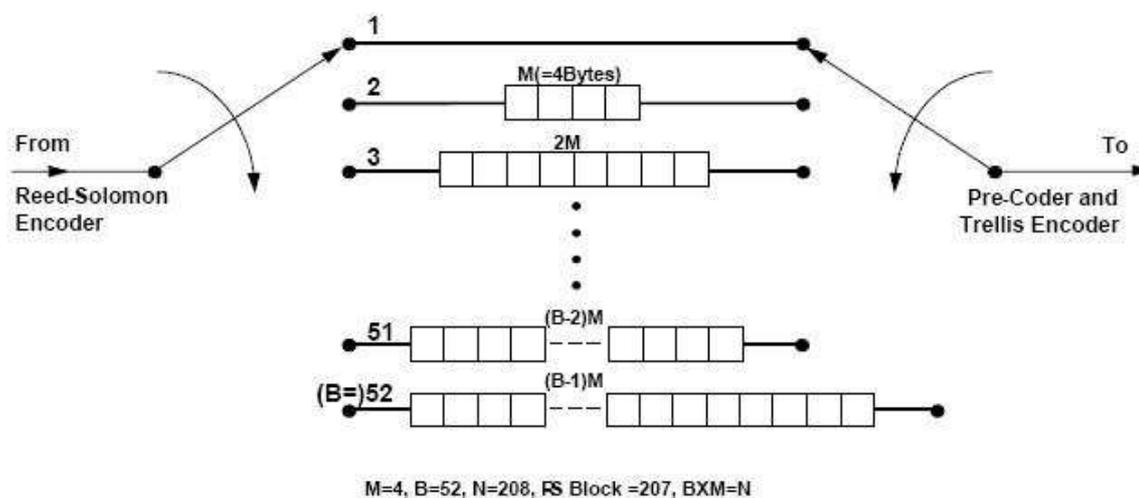


Figura 2.2 – Entrelaçador de dados [4]

Este bloco consegue dispersar os dados sobre um intervalo de 4ms que equivalem a 52 pacotes de 207 bytes que saem do codificador *Reed Solomon*.

### 2.2.1.5. Codificador de Treliça

O Codificador de Treliça [4] tem como principal finalidade à detecção e correção de erros introduzidos pelas interferências no canal de transmissão, servindo de complemento ao codificador *Reed Solomon*.

Neste, para cada dois bits apresentados em sua entrada, são originados três bits de saída, onde o primeiro bit é copiado para a saída e o segundo gera dois bits na saída através de um codificador de razão 1/2. Assim, dos quatro possíveis resultados de entrada,

possuímos oito possíveis resultados de saída, obtendo assim um codificador de Treliça de 2/3.

Cada conjunto de três bits forma um símbolo da modulação digital, logo podem existir oito símbolos diferentes na saída, e cada símbolo representa uma diferente amplitude do sinal modulado no bloco modulador.

A Figura 2.3 mostra o formato dos dados de entrada e de saída neste bloco.

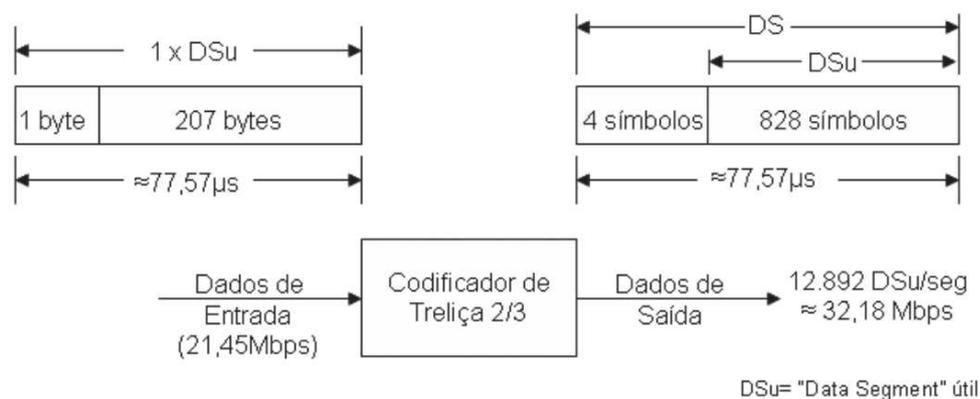


Figura 2.3 – Entrada e Saída dos dados do bloco Codificador de Treliça [4]

Este decodificador minimiza a complexibilidade do processo de decodificação, pois, com sua capacidade de corrigir bits, ele poupa o sistema de ativar o decodificador *Reed Solomon* quando apenas poucos bits são recebidos em erro. O decodificador *Reed Solomon* é um processo bem mais complexo em termos computacionais do que um decodificador de treliça.

## 2.2.2. Estruturador de Quadro

### 2.2.2.1. Multiplexador

A função deste bloco é organizar os 828 símbolos da saída do Codificador de Treliça conforme a Figura 2.4 da estrutura do *frame*.

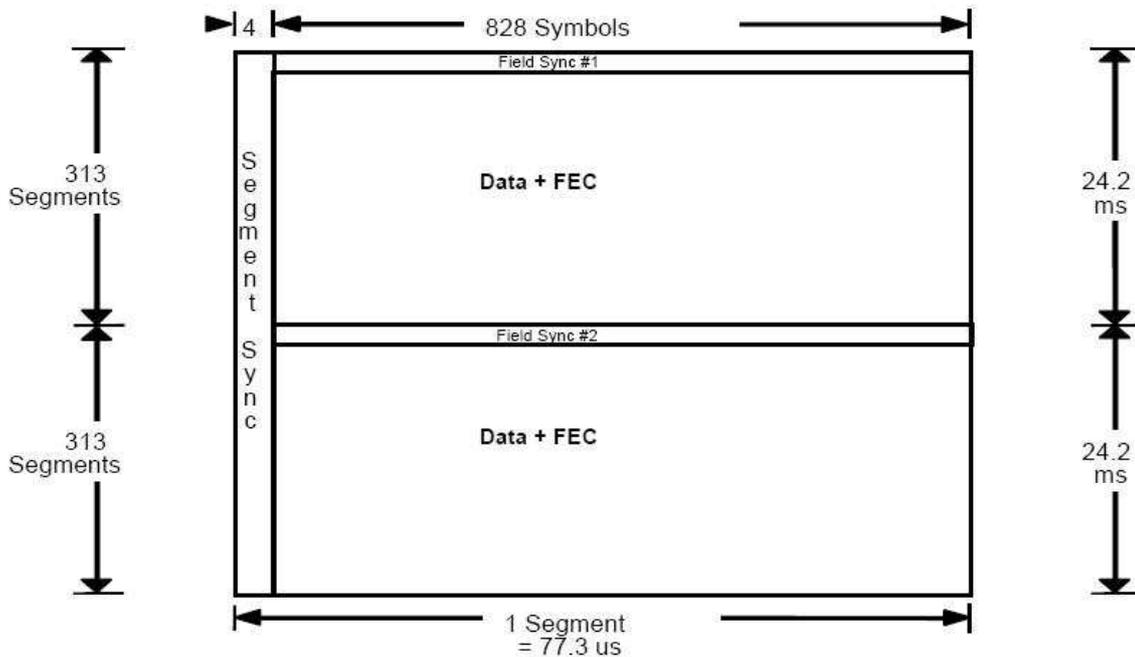


Figura 2.4 – Estrutura do *Frame*

### 2.2.2.2. Inserir de Pilotos

Este bloco converte o sinal digital em um dos oito níveis de tensão conforme a Figura 2.5. Ele acrescenta um sinal DC de +1,25 V a cada símbolo, gerando assim na modulação 8VSB um piloto em fase com a portadora suprimida pelo modulador AM-DSB-SC que sucede o bloco. Este processo permite com que o receptor realize a detecção síncrona do sinal 8VSB [4].

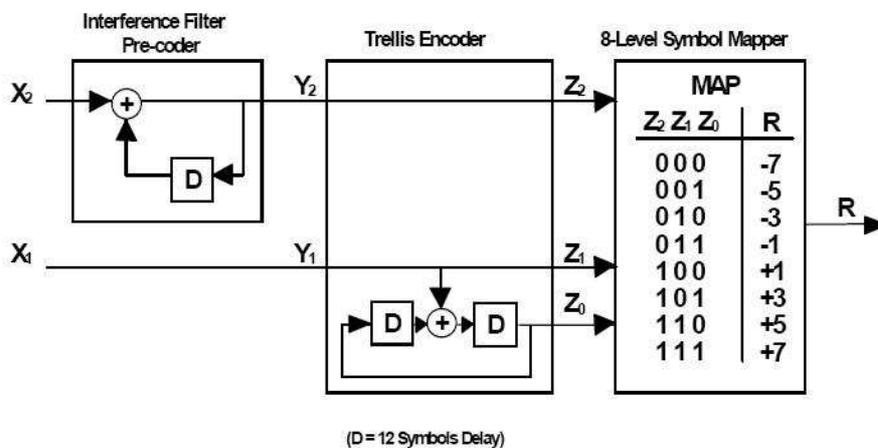


Figura 2.5 – Filtro de Interferência, Codificador de Treliça e Mapeador de símbolos 8VSB

### 2.2.3. Modulador 8VSB

#### 2.2.3.1. Modulador AM-DSB-SC (*Amplitude Modulated – Double Side Band-Suppressed Carrier*)

Tem como função principal modular a portadora senoidal na Frequência Intermediária (FI) com os oito níveis possíveis de amplitude e o componente DC do piloto do sinal de entrada.

Como este é um modulador em amplitude com dupla banda lateral, ele ocupa com o sinal modulado 21,52MHz, ou seja 2 vezes 10,76MHz, sendo a banda útil do sinal modulador igual a sua taxa de símbolos que é de 10,76MHz.

#### 2.2.3.2. Filtro de VSB (*Vestigial Side Band*)

Este bloco filtra o sinal AM-DSB reduzindo assim a banda ocupada por este que é de 21,52MHz, estando muito acima da banda reservada para cada canal de televisão que é de 6MHz, porém, a saída do Filtro VSB apresenta uma banda de 10,76MHz.

#### 2.2.3.3. Filtro de Nyquist

Utilizando um filtro de Nyquist com um *Roll off* próximo de zero, este filtro possui a propriedade de reduzir a banda pela do sinal metade, sem a ocorrência de interferência Intersimbólica, reduzindo assim o sinal para 5,38MHz que é menor que a banda destinada ao canal de televisão de 6MHz, como mostra a Figura 2.6.

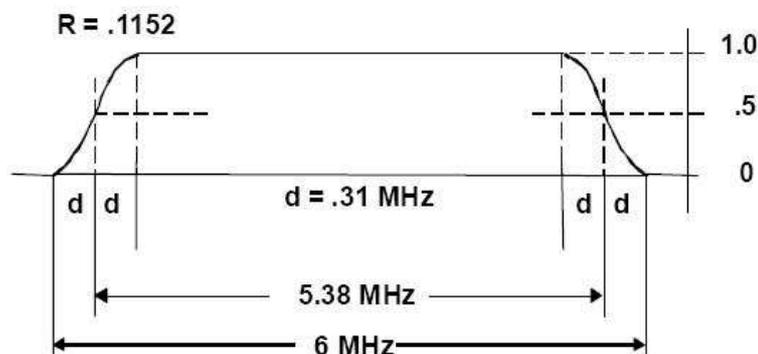


Figura 2.6 – Resposta do Filtro de Nyquist

Com este bloco, acabamos nossa breve explicação do modulador do sistema ATSC, ressaltando os principais blocos e características do sistema.

## 2.3. Digital Video Broadcasting – DVB [5]

Onze anos atrás, em junho de 1996, foram definidos os principais parâmetros para o sistema Europeu de Televisão Digital, por meio do documento *DVB Document AO12 for Digital Terrestrial Television*, o qual, contém de forma generalizada a versão *Standard* e a versão HDTV da TV digital europeia.

O padrão DVB diferencia-se do padrão ATSC principalmente no método de modulação utilizado. O ATSC possui modulação monoportadora e em amplitude *8VSB*, já o DVB é multiportadora modulado em *QPSK* (*Quadrature Phase-Shift Keying*), *16QAM* (*Quadrature Amplitude Modulation*) e *64QAM* multiplexadas por divisão de frequência.

Chamamos este método de modulação de *COFDM* (*Codec Orthogonal Frequency Division Multiplex*).

As constelações destes tipos de modulações citados serão vistas no bloco do mapeador.

O diagrama em blocos do sistema DVB esta abaixo na Figura 2.7, e terá seus blocos brevemente comentados a seguir.

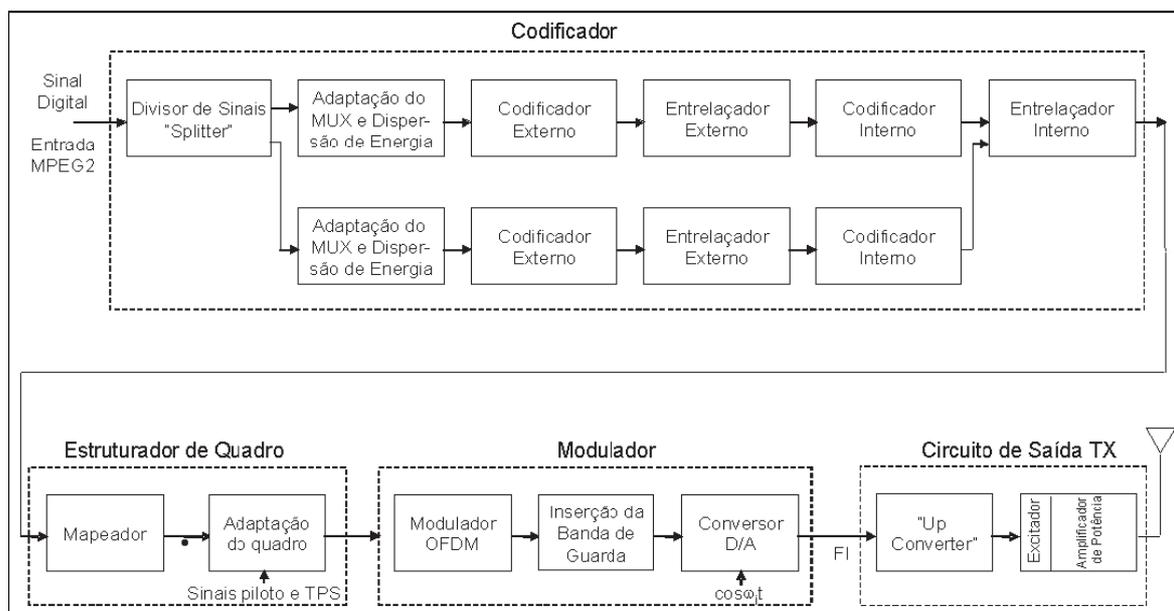


Figura 2.7 – Diagrama em Blocos do Sistema DVB [5]

### 2.3.1. Codificador

Não há muita distinção entre os codificadores dos sistemas citados, sendo o do DVB basicamente o mesmo do ATSC, contendo apenas algumas modificações pelo fato deste sistema ser multiportadora e não monoportadora como o sistema ATSC.

#### 2.3.1.1. Divisor de Sinais (*Splitter*)

O DVB possui a possibilidade de transmissão hierárquica, possibilitando que em um mesmo canal de 8MHz de banda no caso europeu, possa ser transmitido dois canais de conteúdos diferentes, podendo um ser de transmissão de TV *standard* e outro destinado a outro conteúdo digital como interatividade ou outro serviço.

Este bloco serve para dividir o sinal de entrada em dois sinais contendo conteúdos diferentes, os quais podem ser modulados por constelações de sinais de densidade diferente, podendo ser uma QPSK e outra 16QAM por exemplo.

No caso de transmissão de HDTV *High-Definition Television*, é necessária a utilização de toda a banda do canal de 8MHz com uma modulação tão densa como 64QAM. Neste caso, o Divisor de Sinais apresenta dados para transmissão em apenas uma de suas saídas.

#### 2.3.1.2. Adaptação do MUX e Dispersão de Energia

Na saída deste bloco temos sinais digitais compostos por sinal de vídeo, som e dados, o chamado MPEG2.

Os pacotes que contém estes dados na saída do MUX são compostos por 187 bytes de dados úteis e 1 byte de sincronismo, após o MUX, estes 187 bytes de dados são aleatorizados para que não haja concentração de energia causada por repetições de padrões de dados.

#### 2.3.1.3. Codificador Externo

O codificador externo é um codificador do tipo FEC como o *Reed Solomon* do ATSC, porém, ao invés de aumentar 20 bytes de redundância ele apenas aumenta 16 bytes.

Sendo o pacote de entrada num total de 188 bytes, ele entrega a saída um pacote de entrada de 204 bytes.

### 2.3.1.4. Entrelaçador Externo

O Entrelaçador Externo conhecido também por Interleaver de byte, possui a finalidade de aleatorizar os bytes de saída do codificador externo de forma a evitar que uma interferência possa causar perdas consecutivas de dados, dificultando assim no receptor que o código corretor de erros tenha dificuldade ou até mesmo não consiga resolver o problema.

A Figura 2.8 mostra o diagrama do Entrelaçador Externo (*Interleaver*), e de seu dual (*Deinterleaver*) no receptor, com profundidade  $N=12$ , ou seja, é composto por 12 *fifo*s (*first in first out*) de tamanho  $17*(N-1)$  bytes, conforme a Figura.

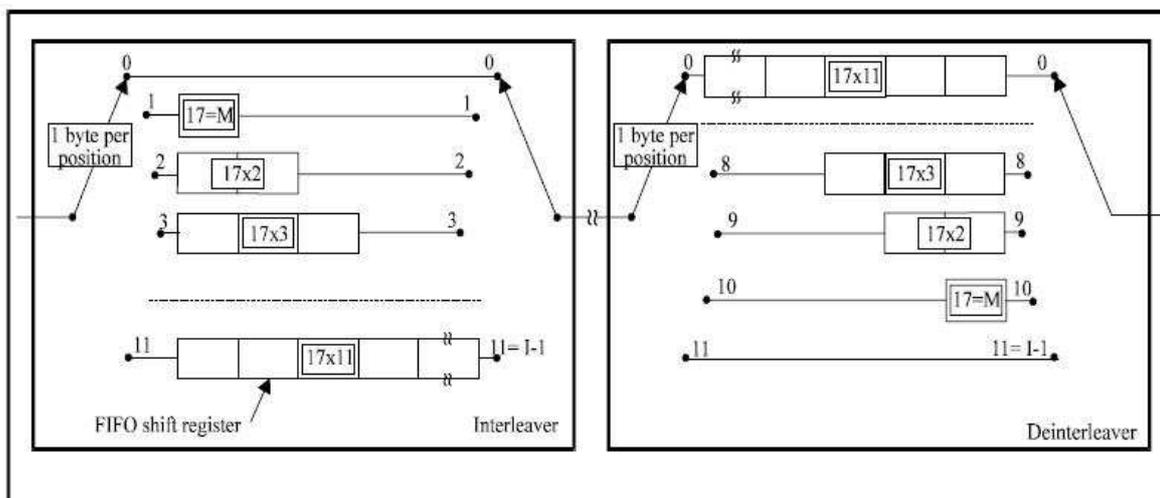


Figura 2.8 – Figura do *Interleaver* e *Deinterleaver* [5]

### 2.3.1.5. Codificador Interno

Este codificador é semelhante ao codificador de treliça do ATSC, porém neste, para cada bit de entrada ele terá dois bits de saída, complementado assim o Codificador Externo, ou seja, garantindo a robustez do sistema para correção de bits que tenham sido invertidos no canal de transmissão, não necessitando a aplicação de um código de complexidade maior para soluções de erros menores.

Outra característica é que este codificador pode ser configurado para trabalhar em outras taxas, tais como  $1/2$ ,  $2/3$ ,  $3/4$ ,  $5/6$  e  $7/8$ .

### 2.3.1.6. Entrelaçador Interno (*Inner Interleaver*)

Este bloco subdivide-se em outros três blocos, o Demultiplexador, o Interleaver de bit e o Interleaver de símbolos, os quais, serão descritos abaixo, porém, sua função é facilitar a utilização do Codificador Interno no Receptor.

#### 2.3.1.6.1. Demultiplexador

Este é responsável por dividir os bits seqüenciais de entrada em bits paralelos, que podem ser de dois em dois para uma modulação QPSK (Figura 2.9) [5], ou de quatro em quatro para uma modulação 16QAM (Figura 2.10) [5], ou ainda de seis em seis bits para uma modulação de 64QAM (Figura 2.11) [5].

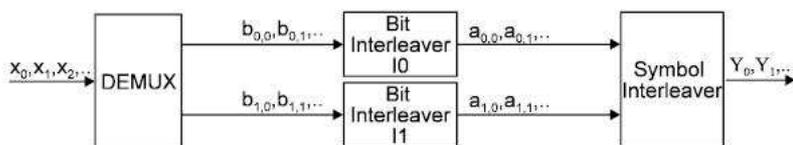


Figura 2.9 – Entrelaçador Interno para modulação QPSK [5]

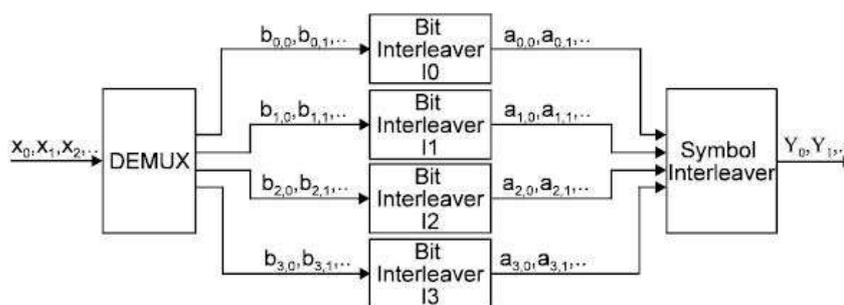


Figura 2.10 – Entrelaçador Interno para modulação 16QAM [5]

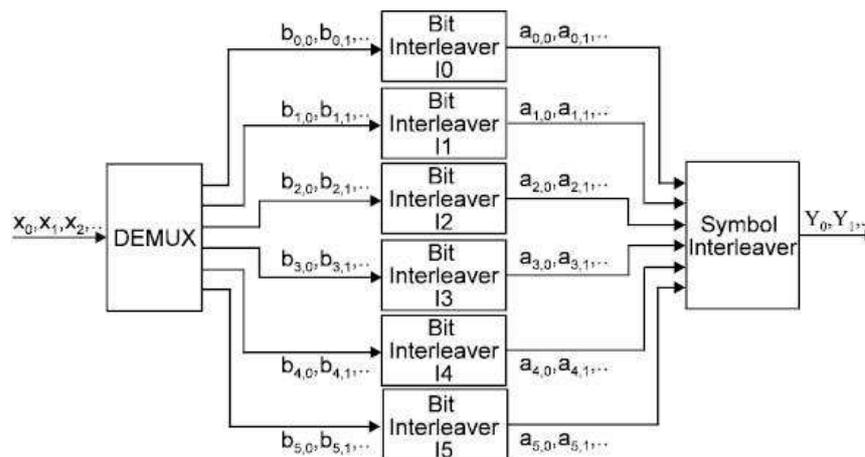


Figura 2.11 – Entrelaçador Interno para modulação 64QAM [5]

### 2.3.1.6.2. Interleaver de Bit

O Interleaver de Bit serve para desordenar a seqüência que está entrando em paralelo no bloco, porém em série em cada *fifo* deste, aumentando assim a robustez do sistema perante a interferências no canal, afastando estes no tempo quando serializados novamente, garantindo assim com que não haja bits invertidos pelo canal agrupados prejudicando assim o desempenho do Codificador Interno.

### 2.3.1.6.3. Interleaver de Símbolo

Este bloco garante que não haja conjuntos de 2, 4 ou 6 bits (conforme a modulação), iguais um após o outro, garantindo assim um afastamento destes nas sub-portadoras do sistema.

## 2.3.2. Estrutura do Quadro

Constituído pelo mapeador e adaptação de quadro é responsável pelo mapeamento e montagem dos quadros contendo os TSP's (*Transmission Parameter Signals*), os quais possuem a configuração do transmissor para que o receptor possa auto configurar-se, além de adicionar os pilotos que possuem as informações de sincronismo de quadro e de frequência, estimação de canal e identificação do modo de transmissão.

### 2.3.2.1. Mapeador

Estabelece uma relação entre os bits oriundos do Interleaver de Símbolos e os estados de fase/amplitude das sub-portadoras da modulação OFDM, em outras palavras, gera os símbolos IQs (*In-Phase/Quadrature*).

As Figuras 2.12, 2.13 e 2.14, mostram as constelações geradas pelo código de Gray para mapear os bits de entrada conforme os estados das modulações QPSK, 16QAM e 64QAM.

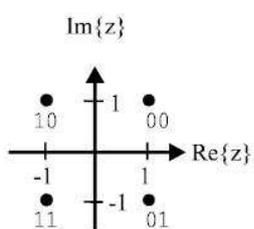


Figura 2.12 – Constelação QPSK [5]

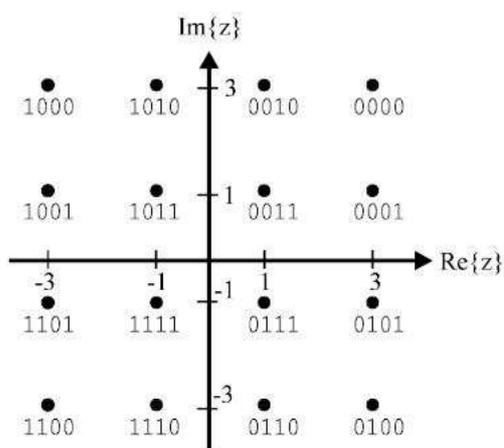


Figura 2.13 – Constelação 16QAM [5]

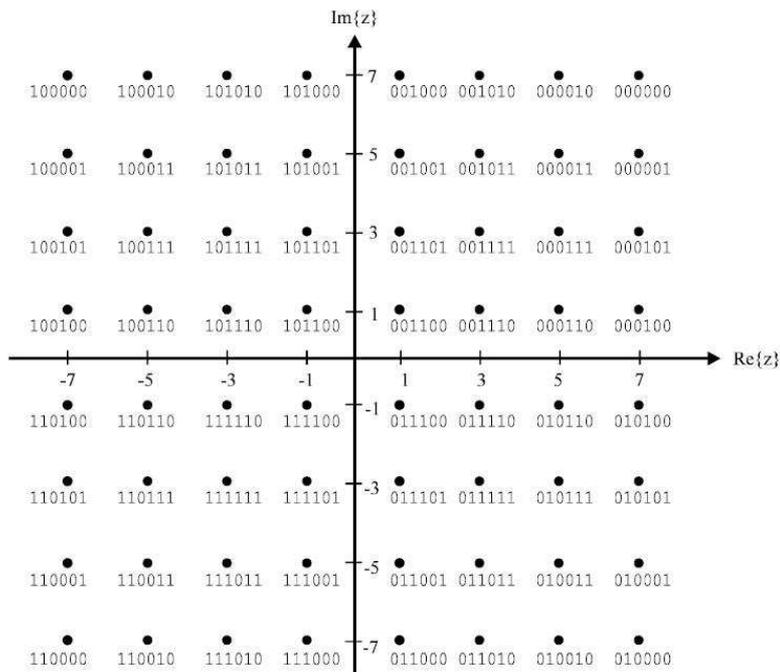


Figura 2.14 – Constelação 64QAM [5]

### 2.3.2.2. Adaptação de Quadro

Neste são acrescentados os TPS's para compor a arquitetura do quadro do COFDM, os quais possuem o número de portadoras, o número da portadora inferior, o número da portador superior, duração do símbolo OFDM e espaçamento entre portadora inferior e superior.

### 2.3.3. Modulador

Esta etapa é composta pelo Modulador OFDM, Inserção do Intervalo de Guarda e o conversor D/A.

#### 2.3.3.1. Modulador OFDM

A modulação OFDM tem a característica de utilizar sub-portadoras ortogonais entre si, ou seja, se calcularmos a integral do produto entre duas sub-portadoras de frequências distintas no intervalo do mesmo símbolo OFDM, o resultado será nulo.

Devido a esta ortogonalidade a interferência intersimbólica entre sub-portadoras é nula.

Agora, cada sub-portadora será modulada conforme o conjunto de bits aplicado à entrada do mapeador, correspondendo a um determinado estado de fase e amplitude.

Para converter as sub-portadoras em frequência para o domínio do tempo, utiliza-se a IFFT (*Inverse Fast Fourier Transformer*), através de processamento digital de sinais.

Após estes passos o Sinal esta pronto para ser passado ao próximo bloco.

### 2.3.3.2. Inserção do Intervalo de Guarda

Devido haver ecos do sinal principal causados pela reflexão do sinal em obstáculos do meio de transmissão, no receptor poderá ocorrer sobreposição de símbolos OFDM, acarretando perda de informações úteis. Para que isto não ocorra, adiciona-se o intervalo de guarda, que é a cópia dos dados do final do símbolo OFDM no tempo para o início deste, fazendo com que o sinal sobreposto por um multipercurso (eco) seja a parte redundante do símbolo.

A Figura 2.15 mostra como fica o símbolo após a inserção do intervalo de guarda.

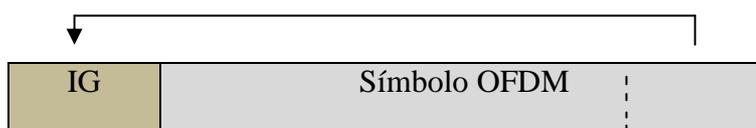


Figura 2.15 – Figura do Símbolo OFDM com o Intervalo de Guarda

O tamanho do intervalo de guarda pode variar de 1/4, 1/8, 1/16 e 1/32 do tamanho do símbolo OFDM, depende do meio de transmissão, se este é muito reflexivo ou não.

### 2.3.3.3. Conversor D/A (Digital/Analógico)

Como todas as operações feitas até aqui foram desenvolvidas através de processamento digital de sinais, basta agora converter o sinal digital em analógico, para que este possa ser confinado dentro da banda de um canal de TV de 6MHz, sendo esta a função do conversor D/A.

Com este bloco encerramos a descrição básica do funcionamento do sistema europeu (DVB), o qual ajudará ao leitor compreender melhor a proposta no capítulo do desenvolvimento deste trabalho, bastando agora apenas citar e explicar o funcionamento do sistema Japonês (ISDB), no qual o sistema Brasileiro esta utilizando como base.

## 2.4. *Integrated Services Digital Broadcasting* – ISDB [6]

O ISDB é o sistema Japonês de Televisão Digital, que foi desenvolvido tendo como base o sistema Europeu (DVB), logo, possuem muitas semelhanças, mas o ISDB é mais complexo e mais robusto, permitindo transmissão hierárquica em três níveis diferentes (ao contrario do DVB que permite apenas duas) e segmentando a banda de um canal de 6MHz em 13 segmentos distintos que podem transmitir três programações diferentes na mesma banda. Note que existe apenas uma programação possível no canal analógico de televisão.

A segmentação é dividida de acordo com a modulação escolhida que pode ser (QPSK, DQPSK, 16QAM e 64QAM) conforme a qualidade da programação a ser transmitida, por exemplo, para uma transmissão de HDTV todos os segmentos devem ser ocupados com uma modulação de 64QAM, garantindo assim a taxa de transmissão de um sinal HDTV.

A Figura 2.16 mostra como fica o canal de 6MHz segmentado no ISDB.

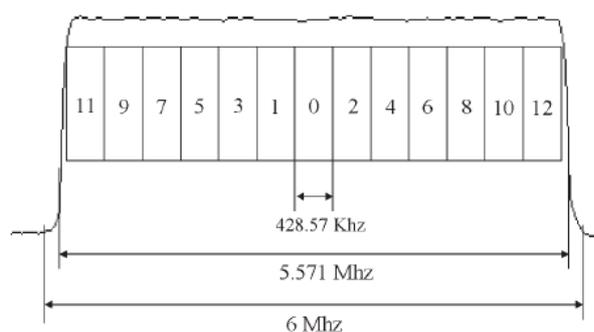


Figura 2.16 – Figura da segmentação do canal de 6MHz

Outra diferença é que no DVB a modulação só pode ocorrer em dois modos o 2K com 2048 sub-portadoras e o 8K com 8192 sub-portadoras, já o ISDB possui três modos de modulação os 2K e 8K como o DVB e mais o modo 4K com 4096 sub-portadoras de dados.

O diagrama de blocos do sistema ISDB esta na Figura 2.17, onde podemos ver diversos blocos semelhantes ao sistema DVB com alguma alteração, mas com a mesma função.

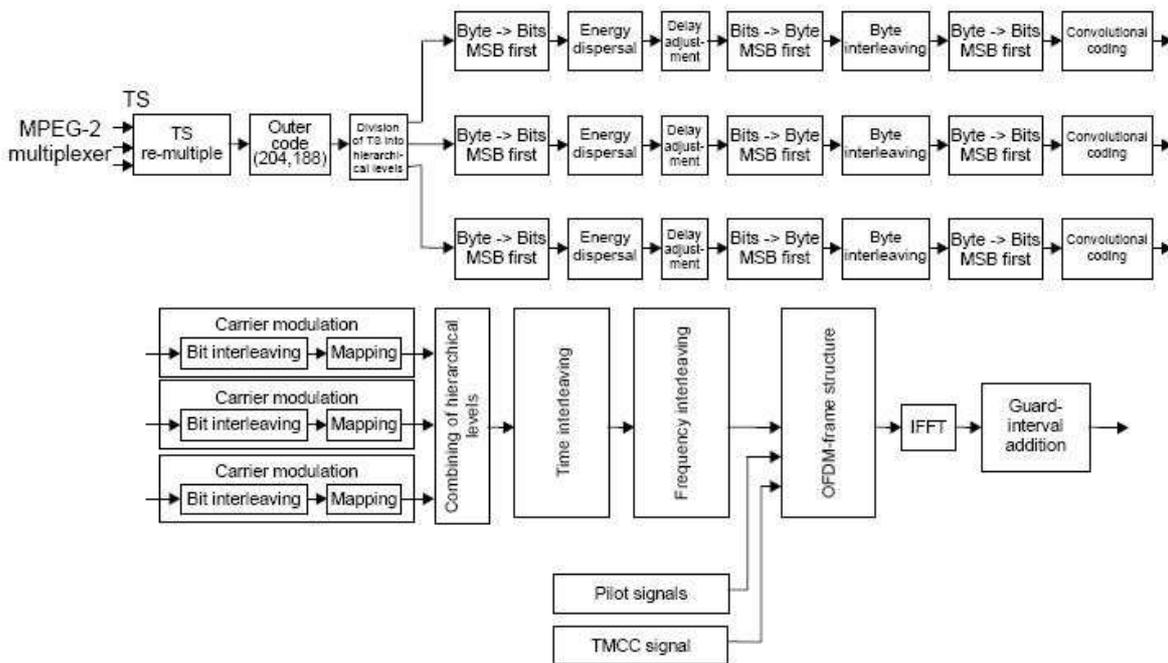


Figura 2.17 – Diagramas de bloco do sistema ISDB [6]

O diagrama de blocos da Figura 2.17 mostra o sistema inteiro, mas que podemos dividir em 3 grandes blocos como mostra a Figura 2.18.

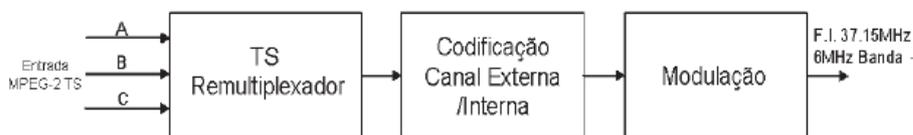


Figura 2.18 – Sistema ISDB dividido em três blocos

### 2.4.1. TS Remultiplexador

O Modulador ISDB recebe em sua entrada três TS (*Transport Stream*), seqüência de dados multiplexados MPEG2 (sinal de vídeo, sinal de áudio e dados), e o remultiplexador tem a função de transformar estes três TS's em apenas um TSP contendo as mesmas informações, serializando os dados para entregar ao codificador *Reed Solomon*. Os pacotes serializados de TSP's podem ser vistos na Figura 2.19, onde o índice A dos TSP's pertence ao canal de dados A e assim por diante, caso não haja TS na entrada de um canal de dados, o Remultiplexador adiciona TSP's nulos, também podemos ver que um quadro multiplexado contém 1152 TSP's.

No codificador de canal estes TSP's serão distribuídos conforme seu sinal de controle (A, B e C) em seus respectivos níveis hierárquicos, este sinal de controle é informado pelo remultiplexador.

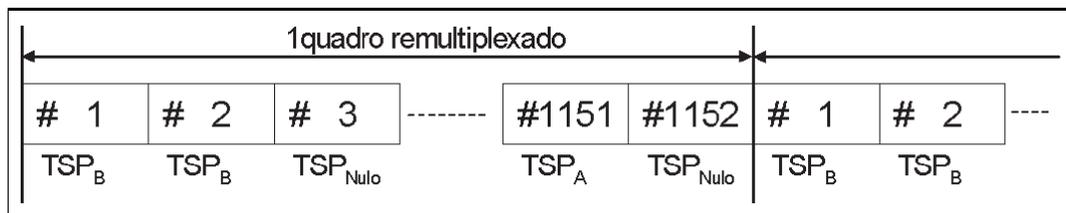


Figura 2.19 – TS's multiplexados (TSP)

## 2.4.2. Codificação de canal

Todos os blocos que ficam entre o codificador *Reed Solomon* e o mapeador pertencem ao codificador de canal, o qual possui muita semelhança com o codificador de canal do sistema DVB.

### 2.4.2.1. Codificador Reed Solomon

Possui a mesma característica e finalidade que no sistema DVB, recebendo pacotes de 187 bytes de dados e 1 byte de sincronismo adicionando então mais 14 bytes de redundância, tendo assim em sua saída 204 bytes para cada 188 bytes de entrada, como mostra a Figura 2.20.

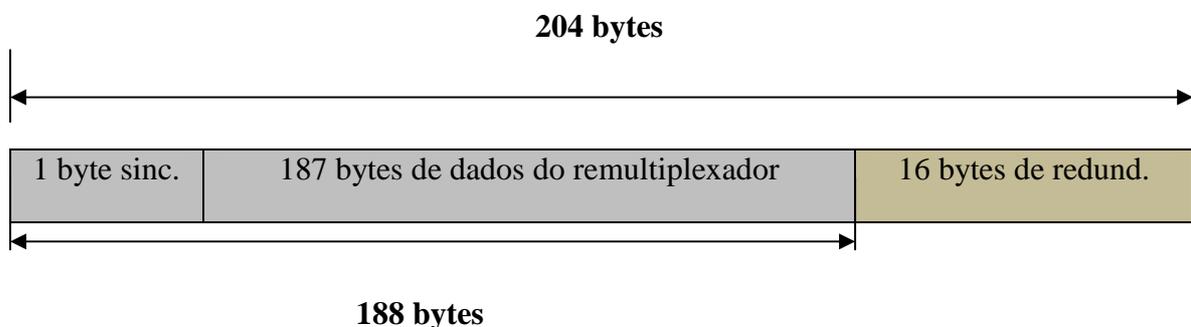


Figura 2.20 – Pacotes de saída do codificador *Reed Solomon*

#### **2.4.2.2. Divisor de TS**

O bloco Divisor de TS tem a função de receber o pacote agora com 204 bytes do codificador *Reed Solomon* (TS) e direcioná-lo para seu respectivo *layer hierárquico* (nível hierárquico) conforme indicação do remultiplexador.

Caso ele receba algum TS nulo, este é descartado, não seguindo adiante na codificação.

#### **2.4.2.3. Dispensor de energia**

Tem a mesma função do aleatorizador de dados da seção 2.2.2. É composto por um PRBS, o qual, distribui as possíveis seqüências de zeros e uns que possam estar agrupados durante a transmissão.

#### **2.4.2.4. Ajuste de Delay**

O Ajuste de delay serve para garantir que no receptor a decodificação comece em todas suas etapas no início real de um *frame*, ou seja, quando o receptor inicia a decodificação ele contém dados inválidos em suas *fifos*, seja por uma mudança de canal feita pelo usuário ou por uma inicialização do sistema, logo o ajuste de delay insere dados nulos os quais farão com que quando dados válidos chegarem aos blocos intermediários, estes estejam na posição inicial sem necessitar de sinais de sincronismos de entre blocos.

#### **2.4.2.5. Interleaver de Byte**

O interleaver de Byte é um bloco que todos os sistemas possuem e cuja finalidade é a mesma de embaralhar os bytes no tempo para que não ocorram *burst* de dados devido a interferência, facilitando assim a correção de erros pelo codificador *Reed Solomon*.

A particularidade deste bloco é apenas a profundidade e o tamanho das *fifos* que ele contém, aumentado a robustez em relação a um sistema com profundidade de interleaver menor, pois espalha mais os erros no tempo.

Sendo o DVB a base do sistema ISDB a profundidade do interleaver de Byte dos dois é a mesma, a Figura 2.8 da seção 2.3.1.4 representa este bloco.

### 2.4.2.6. Código Convolutacional

O código convolutacional do ISDB é um código de puncionamento como o Codificador Interno da seção 2.3.1.5 do sistema DVB com taxa 1/2 e 64 estados, funcionando da mesma forma.

## 2.4.3. Modulação

### 2.4.3.1. Interleaver de Bit e Mapeador

A parte de modulação do sistema ISDB começa pelo embaralhador de bits, que possui diferenças em relação ao do sistema do DVB. O Interleaver de Bit do ISDB funciona como o Interleaver de Byte, com *fifos* de profundidade igual ao número de bits da modulação desejada em tamanhos múltiplos de uma potência de dois do número de bits da modulação. As figuras abaixo mostram os blocos do Interleaver de Bit para cada modulação, e o mapa das constelações que derivam na saída do mapeador que esta conectado diretamente a ele, parecendo assim, que os dois formam um único bloco.

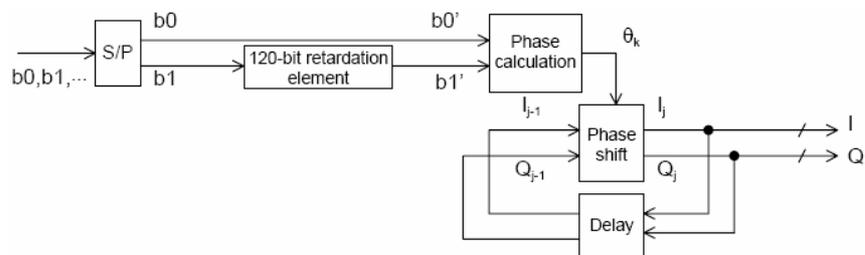


Figura 2.21 – Interleaver de Bit para modulação DQPSK

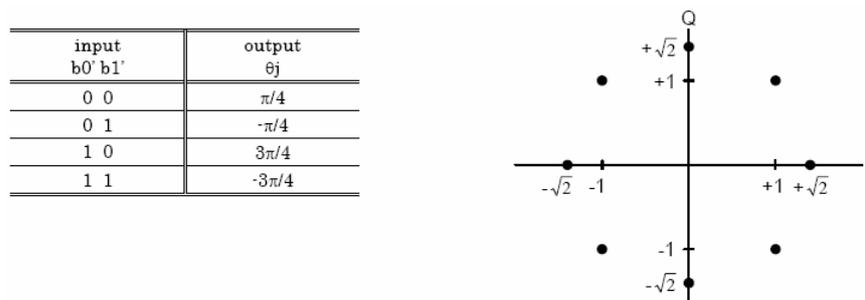


Figura 2.22 – Mapeamento para uma modulação DQPSK

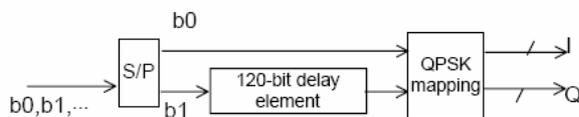


Figura 2.23 – Interleaver de Bit para modulação QPSK

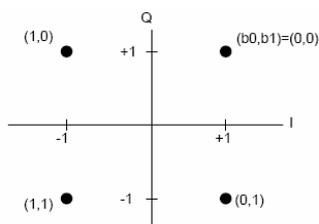


Figura 2.24 – Mapeamento para uma modulação QPSK

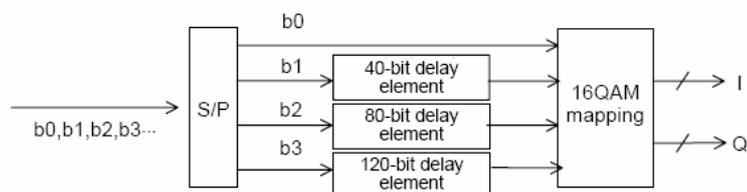


Figura 2.25 – Interleaver de Bit para modulação 16QAM

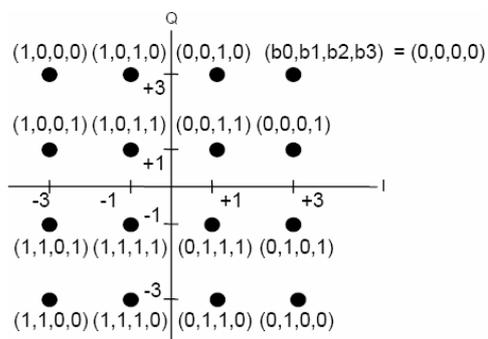


Figura 2.26 – Mapeamento para uma modulação 16QAM

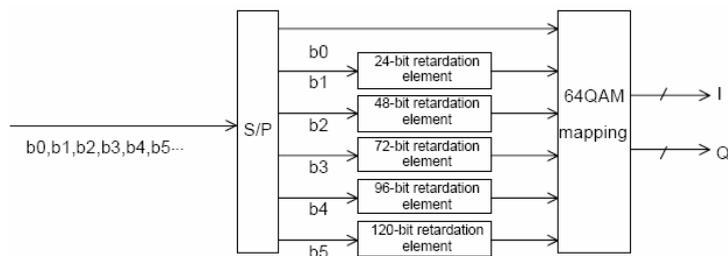


Figura 2.27 – Interleaver de Bit para modulação 64QAM

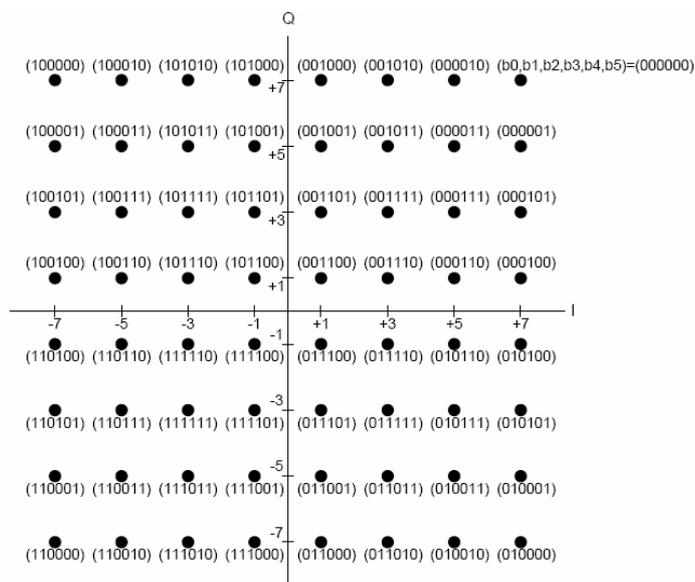


Figura 2.28 – Mapeamento para uma modulação 64QAM

#### 2.4.3.2. Combinador de Níveis Hierárquicos

Este tem a finalidade de serializar os dados que chegam à sua entrada vindos dos níveis hierárquicos A, B e C (quando os três se aplicarem, ou dois destes ou ainda apenas um dependendo da configuração do transmissor), o quais já encontram-se mapeados em símbolos IQs.

O Combinador é formado por *buffers* de tamanho suficiente para evitar que ocorra a sobreposição de dados úteis provenientes dos níveis hierárquicos, já que podem estar chegando mais informações de um nível hierárquico do que em outro, isto é fácil de entender através da Figura 2.29.

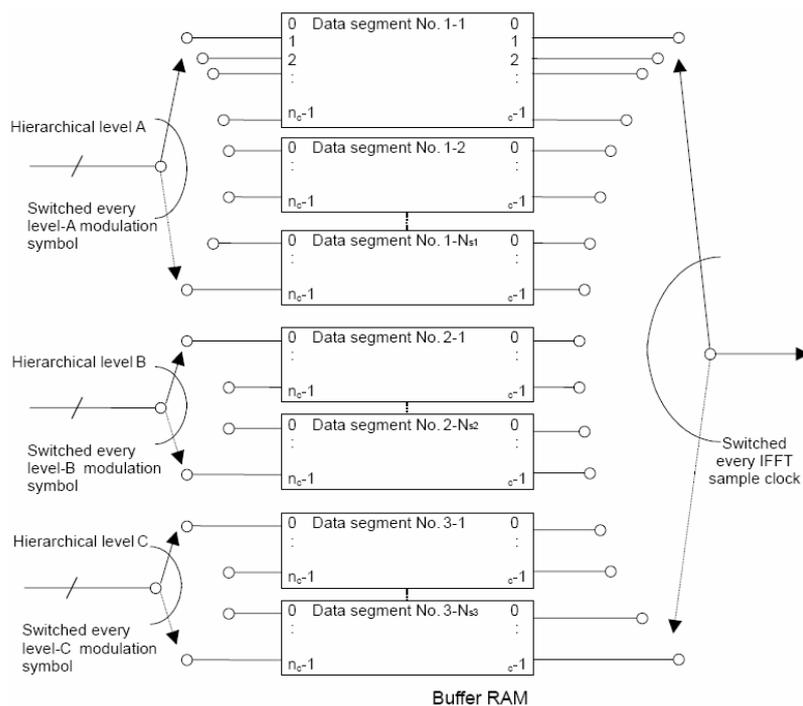


Figura 2.29 – Combinador de Níveis Hierárquicos

Sendo  $n_c$  igual ao número de símbolos IQs por segmento (96, 192 ou 384 de acordo com o modo de transmissão 2K, 4K ou 8K respectivamente), temos no total  $n$  posições diferentes para cada chave.

Como podemos ver as chaves de entrada e de saída são independentes e todas possuem diferentes números de posições para percorrer. A chave de saída respeita o *clock* da IFFT, que é menor que os *clocks* das chaves de entrada deste bloco.

Necessitando assim de *buffers* de tamanhos consideravelmente grandes para cada linha do Combinador de Níveis Hierárquicos, evitando assim a sobreposição de dados que possa acontecer se a velocidade de entrada destes dados for maior do que a velocidade de saída dos mesmos.

### 2.4.3.3. Interleaver de Tempo

Eis um dos dois blocos que distinguem bem o sistema ISDB do sistema DVB em questão de robustez. O Interleaver de Tempo é responsável por “embaralhar” símbolos IQs no tempo, garantindo assim que não haja perda de símbolos IQs subsequentes, o que aumentaria a BER (*bit error rate*) no receptor.

O Interleaver de Tempo é um *interleaver* convolucional como o *interleaver* de byte, com profundidade igual ao número de segmentos e tamanhos diferentes de *fifo*s para cada segmento de transmissão.

A Figura 2.30 apresenta o diagrama de blocos do Interleaver de Tempo.

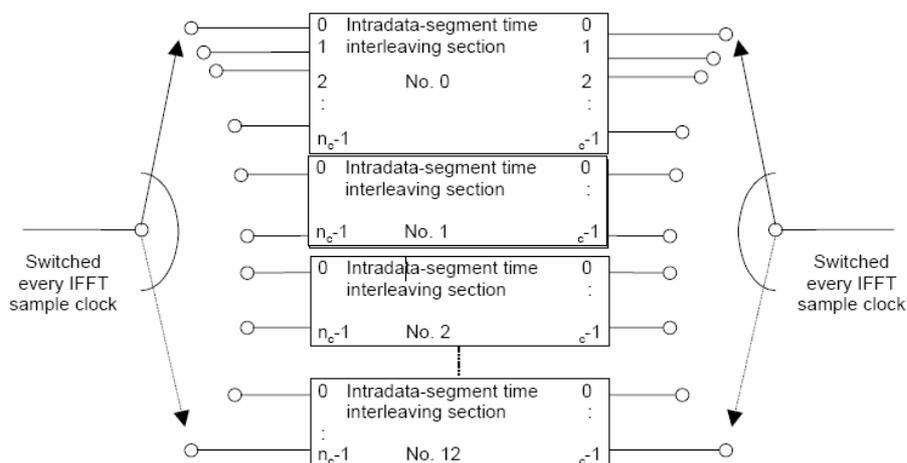


Figura 2.30 – Interleaver de Tempo

Neste bloco para cada símbolo IQ de entrada temos um símbolo IQ de saída e o espaçamento temporal que ele causa aos símbolos pode chegar até 370 ms.

#### 2.4.3.4. Interleaver de Frequência

Este é o outro bloco que distingue bem o sistema Japonês do sistema Europeu, onde o embaralhamento ocorre em frequência, ou seja, ocorre troca dos símbolos IQs de suas portadoras originais para outras, garantindo assim que se um *notch* na função de transferência do canal está degradando a SNR de uma sub-portadora sendo recebida no receptor, não teremos dados perdidos de uma única portadora original. Isto aumenta a robustez do sistema quanto ao multipercurso estático.

O embaralhamento ocorre primeiro entre as sub-portadoras dos treze segmentos que formam um símbolo OFDM composto de  $n$  portadoras (conforme modo de operação). Após, existe mais um embaralhamento dentro de cada segmento obedecendo a uma determinada regra de formação e por último um embaralhamento randomizado também interno a um segmento que segue tabela especificada em norma [6].

Ao final desta operação as sub-portadoras resultam conforme a Figura 2.31 para o modo 2K, Figura 2.32 para o modo 4K e Figura 2.33 para o modo 8K.

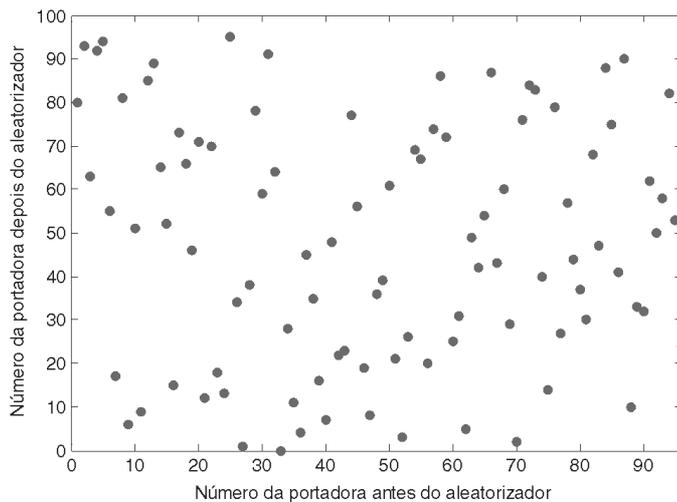


Figura 2.31 – Gráfico de dispersão das portadoras dentro de um segmento no modo 2K

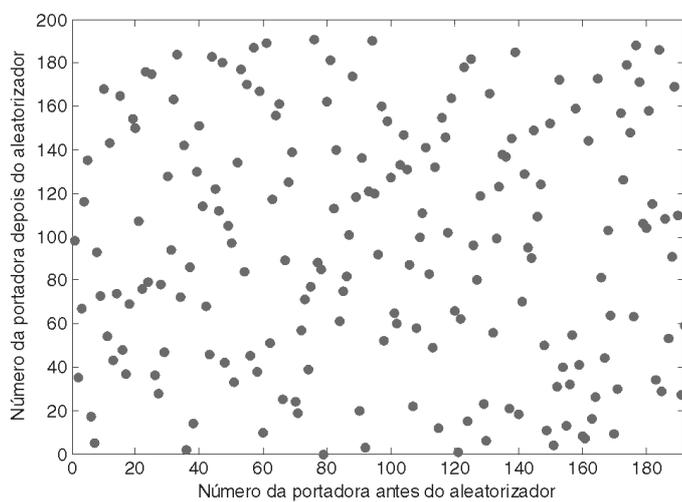


Figura 2.32 – Gráfico de dispersão das portadoras dentro de um segmento no modo 4K

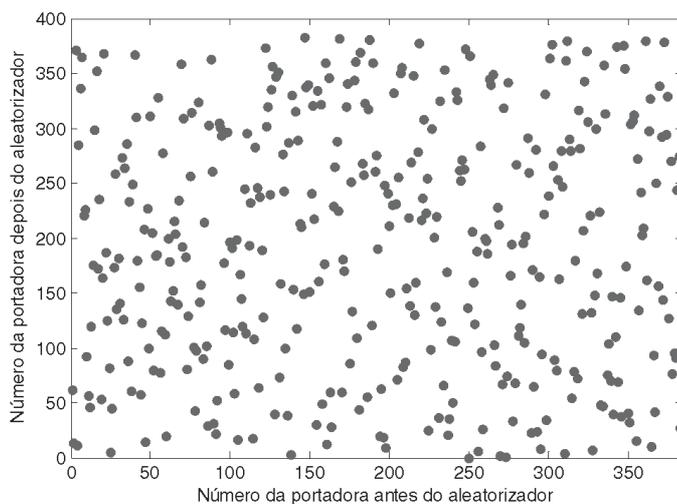


Figura 2.33 – Gráfico de dispersão das portadoras dentro de um segmento no modo 8K

#### **2.4.3.5. Estruturação do *Frame***

A transmissão do sinal do ISDB é feita em quadros formados por 204 símbolos OFDM compostos por 13 segmentos cada.

Neste bloco são adicionados as portadoras pilotos, as quais servem de referência para o receptor e que são transmitidas com uma potência pouco maior que as demais. Neste bloco também são adicionados os parâmetros auxiliares e os TMCCs (Parâmetros de Transmissão, Multiplexação, controle e configuração).

#### **2.4.3.6. Intervalo de Guarda**

O Intervalo de Guarda (IG) ou Prefixo Cíclico (CP), tem as mesmas características e função vista na seção 2.3.3.2 do sistema DVB, até mesmo em seus tamanhos.

## **2.5. Considerações Finais sobre o Capítulo**

Com essa breve descrição dos três sistemas de transmissão digital estudados para a execução deste trabalho e com as características mostradas de cada sistema, poderemos melhor inserir no contexto o método de sincronismo e o ponto de aplicação do mesmo no diagrama de blocos do receptor.

As informações contidas neste capítulo foram baseadas nas respectivas normas dos referidos sistemas. Estas se encontram disponíveis para download em, [www.atsc.org](http://www.atsc.org) para o Sistema Americano, [www.etsi.org](http://www.etsi.org) para o Sistema Europeu e [www.arib.or.jp/english](http://www.arib.or.jp/english) para o Sistema Japonês.

## 3. Redes Neurais Artificiais

### 3.1. Introdução

A dificuldade de solucionar problemas onde não conhecemos a função de transferência de algum sistema foi alvo de pesquisas que iniciaram em 1943 em artigos do neurofisiologista Warren McCulloch, do MIT, e do matemático Walter Pittis da Universidade de Illinois, os quais, fizeram uma analogia entre células nervosas vivas e o processos eletrônicos em um trabalho publicado sobre “Neurônios Formais”, simulando o comportamento de neurônios naturais com apenas uma saída para um processo de soma de suas várias entradas, originando então os estudos em Redes Neurais Artificiais (RNA).

Redes Neurais Artificiais são estruturas de processamento altamente paralelas e fortemente não-lineares, que apresentam a especial característica de aprenderem complicados mapeamentos entrada/saída a partir da observação exaustiva dos dados com que são treinadas, por meio de eficientes algoritmos de aprendizagem. Devido a esta especial característica, as Redes Neurais Artificiais se tornam especialmente adequadas para tratar problemas em que a natureza dos processos envolvidos é indeterminada, não-linear e/ou não estacionária.

A elevada capacidade de aprendizagem das RNA's e a capacidade de se adaptar rapidamente a mudanças são características que sugerem que Redes Neurais Artificiais sejam ferramentas extremamente indicadas para processos que necessitem de aprendizado em tempo real.

Uma RNA é formada por neurônios artificiais, cuja arquitetura básica é mostrada na Figura 3.1.

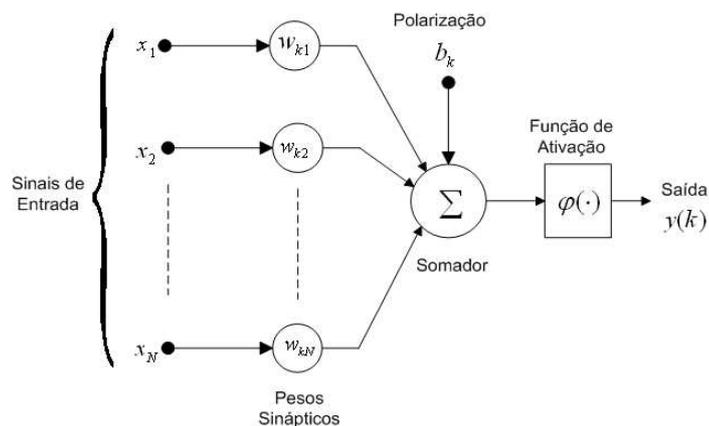


Figura 3.1 – Modelo de um neurônio não-linear [7]

Da Figura 3.1 podemos definir a equação de saída da rede como:

$$y = \sum_{k=0}^{K-1} w_k \varphi_k(x, t_k, \sigma_k^2) + w_b B \quad (3.1)$$

onde  $y$  representa a saída da rede,  $w_k$  os pesos sinápticos de suas respectivas entradas ( $x_n$ ),  $\varphi_k(u, t_k, \sigma_k^2)$  é a função de ativação, e  $w_b B$  o peso do Bias que pode ser utilizado ou não.

RNA's diferem em arquitetura e processo de aprendizado. As RNA's do tipo *Radial Basis Function* (RNA's RBF) possuem a especial característica de rápido aprendizado, o que as torna especialmente adequadas para processos em tempo real.

Hoje em dia, mais evoluídas, as RNA's possuem diversas estruturas distintas, que simulam o funcionamento do cérebro humano para soluções dos mais diversos e complicados problemas de processamento de sinais digitais. Elas diferem principalmente em arquitetura e processo de aprendizado.

As RNA's do tipo *Radial Basis Function* (RNA's RBF) possuem a especial característica de rápido aprendizado, o que as torna especialmente adequadas para processos em tempo real, além disto, possui baixa complexibilidade e baixo custo computacional, fazendo com que seja a mais indicada para o desenvolvimento deste trabalho.

## 3.2. Radial Basis Function – RBF

### 3.2.1. Estrutura da RBF

A RBF é uma rede supervisionada, pois são realimentadas por um erro calculado em função de um valor de referência, consideradas assim como aproximadoras universais.

Esta rede é composta por uma camada escondida [7] de neurônios e sua saída é composta por um único neurônio linear que faz a combinação linear das funções de base radial, ou seja, a saída de uma RBF é a soma das saídas de cada neurônio da camada escondida, como mostra a Figura 3.2.

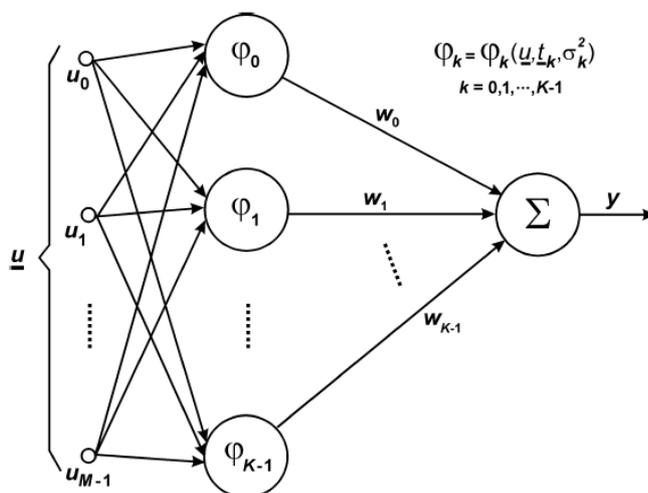


Figura 3.2 – Arquitetura de uma rede RBF [7]

A saída desta rede é dada pela equação 3.2.

$$y = \sum_{k=0}^{K-1} w_k \varphi_k(\underline{u}, t_k, \sigma_k^2) \quad (3.2)$$

onde a função de ativação  $\varphi_k(\underline{u}, t_k, \sigma_k^2)$ , pode ser por exemplo do tipo Lâmina *spline* fina (Equação 3.3), Multi-Quadrática (Equação 3.4), Multi-Quadrática Inversa (Equação 3.5) e a utilizada para este trabalho a Gaussiana (Equação 3.6).

$$\phi(\xi) = \frac{\xi}{\sigma^2} \log\left(\frac{\xi}{\sigma}\right) \quad (3.3)$$

$$\phi(\xi) = \sqrt{\xi^2 + \sigma^2} \quad (3.4)$$

$$\phi(\xi) = \frac{1}{\sqrt{\xi^2 + \sigma^2}} \quad (3.5)$$

$$\phi(\xi) = \exp\left(-\frac{\xi^2}{2\sigma^2}\right) \quad (3.6)$$

Levando em consideração que a rede pode ser formada por mais de um neurônio, a função de ativação  $\varphi_k(\underline{u}, \underline{t}_k, \sigma_k^2)$  representa a  $k$ -ésima função de base radial que compõe a rede RBF, logo, substituindo a Equação 3.6 na Equação 3.2 temos a equação do neurônio baseada em uma Gaussiana obtendo a Equação 3.7.

$$\varphi_k(\underline{u}, \underline{t}_k, \sigma_k^2) = \varphi_k(\underline{u}(n), \underline{t}_k(n), \sigma_k^2(n)) = \exp\left[-\frac{1}{\sigma_k^2(n)} \|\underline{u}(n) - \underline{t}_k(n)\|^2\right] \quad (3.7)$$

onde  $u(n)$  representa o vetor de entrada no instante  $n$ ,  $t_k(n)$  representa o centro da  $k$ -ésima função de base radial  $k = 0, 1, \dots, K - 1$ ,  $K$  é o número de funções de base radial, e  $\sigma_k^2(n)$  é a variância associada a cada uma das funções  $n$ .

### 3.2.2. Heurística de Treino de uma RNA RBF

Para atingir o objetivo que é de achar a função de transferência de um sistema desconhecido utilizando a RBF, temos que apresentar em sua entrada os valores de entrada dos processo que desejamos estudar e a saída resultante destas. Fazendo isto podemos utilizar algoritmos de aprendizado (Tabela 3.1) [7][8] para ajustar os parâmetros livres da

RBF, que são os vetores de centro  $t_k$ , as variâncias dos centros  $\sigma_k^2$  e os pesos sinápticos  $w_k$ .

Possíveis Algoritmos de Aprendizado para Ajuste dos Parâmetros Livres		
Centros das RBF	Pesos Sinápticos	Variâncias dos centros
Constante: Por conhecimento prévio e inferência a partir do conjunto de vetores de treino.	Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$	Constante: Por conhecimento prévio e inferência a partir do conjunto de vetores de treino.
“Clusterização” pelo algoritmo <i>k-means</i> . Não-supervisionado.	Pseudo Inversa por decomposição em valores singulares: $\underline{w}(n) = \Phi^{-1}(n) \cdot \underline{d}(n)$	Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$
Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$		$\sigma_k^2(n) = \xi_k(n) \cdot \max_{a,b} \left\{ \left\  t_a(n) - t_b(n) \right\ ^2 \right\}$ onde $\xi_k(n)$ é fixo ou ajustado pelo LMS.

Tabela 3.1 - Possíveis Algoritmos de Aprendizado de uma rede RBF

Para a execução utilizamos a heurística de treino proposta por Haykin [8] (no contexto de aproximação de funções), a qual se baseia no algoritmo Gradiente Estocástico.

Neste caso os pesos sinápticos, os centros das funções de base radiais e as variâncias dos centros são atualizados através de um processo de aprendizado supervisionado baseado na minimização da função de custo vista na Equação 3.8.

$$J = E\left\{(d(n) - y(n))^2\right\} \quad (3.8)$$

onde  $d(n)$  representa a saída desejada e  $y(n)$  representa a saída obtida pela RNA.

Esta heurística de treino completa é apresentada pela tabela 3.2 [7][8], onde temos todos os passos de inicialização e Treinamento da RNA RBF.

<b>I - Inicialização</b>	
1.	Subtrair o vetor média do conjunto de N vetores de treino.
2.	Normalizar a i-ésima componente de cada vetor de treino pelo desvio padrão do conjunto de N valores formado pela i-ésima componente de todos os N vetores de treino.
3.	Normalizar o conjunto de N saídas desejadas para o intervalo [-1,+1]
4.	<p>Inicializar os centros das funções de base radial:</p> $t_k(n=0) \rightarrow k - means$ $t_k(n+1) = \begin{cases} t_k(n) + \eta[u(n) - t_k(n)]; & k = \tilde{k}(u) \\ t_k(n); & \text{outros casos} \end{cases}$ $\tilde{k}(u) = \arg_k \min \ u(n) - t_k(n)\ ; \quad k = 0, 1, \dots, K-1$ $\ t_k(n) - t_k(n+1)\  \rightarrow \varepsilon, \forall k$
5.	<p>Inicializar os pesos sináptico</p> $w_k(n=0) \rightarrow 0; \quad k = 0, 1, \dots, K-1$
6.	<p>Inicializar as variâncias dos centros:</p> $\sigma_k^2(n=0) \rightarrow d_{\max}^2(t_i - t_k), \forall k; \text{ com } i, j = 0, 1, \dots, K-1, \text{ onde}$ $d_{\max}^2(t_i - t_j) = \max \left\{ \ t_i - t_j\ ^2 \right\}$
<b>II - Treinamento</b>	
1.	Apresentar vetor $u(n) \in$ ao conjunto de treino à entrada da RBF.
2.	<p>Calcular saída da rede através de</p> $y(n) = \sum_{k=0}^{K-1} w_k(n) \varphi_k(n) = \sum_{k=0}^{K-1} w_k(n) \exp \left[ -\frac{1}{\sigma_k^2(n)} \ u(n) - t_k(n)\ ^2 \right]$
3.	Determinar o erro instantâneo de treinamento na interação $n$ através de

$e(n) = d(n) - y(n)$
<p>4. Atualizar os pesos sinápticos <math>w_k</math> conforme</p> $w_k(n+1) = w_k(n) + \mu_w e(n) \varphi_k(n)$ <p>Atualizar os centros das funções de base radial <math>t_k</math> através de</p> $t_k(n+1) = t_k(n) + 2\mu_t e(n) w_k(n) \varphi_k(n) \frac{u(n) - t_k(n)}{\sigma_k^2(n)}$ <p>Atualizar as variâncias dos centros das funções de base radial <math>\sigma_k^2</math> de acordo com</p> $\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma e(n) w_k(n) \varphi_k(n) \frac{\ u(n) - t_k(n)\ ^2}{(\sigma_k^2(n))^2}$
<p>5. Incrementar <math>n</math> (<math>n = n + 1</math>).</p> <p>Se <math>n</math> é tal que todos <math>N</math> vetores do conjunto de treino foram apresentados à entrada da RBF (constituindo a chamada época de treino), executar o procedimento II.6, caso contrário executar o procedimento II.1.</p>
<p>6. Embaralhar aleatoriamente a ordem dos vetores <math>\in</math> ao conjunto de treino.</p>
<p>7. Determinar e avaliar o Erro Médio Quadrático de Aproximação dado por</p> $MSEA = \frac{1}{N} \sum_{n=0}^{N-1} (d(n) - y(n))^2$ <p>Se <math>MSEA &lt; \epsilon</math> executar II.8, caso contrário, executar II.1.</p>
<p>8. Armazenar os parâmetros da rede (<math>w_k, t_k, \sigma_k^2</math>).</p>
<h3>Observação</h3>
<p>Havendo <i>bias</i>, a Equação para a saída da rede (passo II.2) torna-se:</p> $y(n) = \sum_{k=0}^{K-1} w_k(n) \varphi_k(n) + w_b B$

Onde $w_b$ é a transmitância da sinapse do <i>bias</i> e B é o valor do bias.
Neste caso, a equação para atualização da sinapse do <i>bias</i> ( $w_b$ ) é dada por $w_b(n+1) = w_b(n) + \mu_e e(n) B$

Tabela 3.2 – Heurística de treino de uma RNA

### 3.3. Considerações Finais sobre o Capítulo

Neste capítulo vimos uma breve explicação de Redes Neurais Artificiais, focando a rede do tipo RBF, a qual, foi utilizada para o desenvolvimento e validação deste.

Toda informação contida neste capítulo foram retiradas além das bibliografias citadas, de apostilas de professores que ministram a cadeira de Redes Neurais Artificiais do Programa de Pós Graduação da Faculdade de Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul, informações estas que podem ser encontradas no site dos mesmos ( [www.ee.pucrs.br/~decastro/RNA\\_hp/RNA.html](http://www.ee.pucrs.br/~decastro/RNA_hp/RNA.html) ).

## 4. Canais de Transmissão *Wireless*

Em transmissões *wireless*, principalmente em transmissões digitais *wireless*, os canais de transmissão têm um papel fundamental, pois podem consagrar um sistema desenvolvido ou fazer com que este fracasse.

O canal de transmissão nada mais é que o meio onde o sinal será transmitido, neste caso, o ar.

Um canal de comunicação *wireless* pode ser caracterizado essencialmente como um canal dispersivo no tempo, resultado da propagação do sinal transmitido por múltiplos percursos.

Estes possuem características específicas conforme a geografia (relevo) do terreno onde o sinal será propagado, existindo assim, inúmeros e diferentes tipos de canais existentes.

A questão é, como modelar, desenvolver e aplicar um sistema de transmissão *wireless* sem termos a certeza de que ele funcionará, ou pelo menos, permitirá com que pequenos ajustes resolvam um possível problema?

Lembramos que ao final da década de 90, os três sistemas de Televisão Digital já citados acima foram testados no Brasil e não teve seus resultados aprovados totalmente, devido o meio de transmissão (canal de transmissão) do local onde foram executados os testes (São Paulo, SP).

Logo, respondendo a questão anterior, foram feitos modelamentos de canais que possam simular os mais diversos meios de transmissão para que os sistemas *wireless* possam ser testados e validados em laboratório antes de um teste de campo, facilitando assim análises comparativas (estas que validaram os modelamentos dos canais).

Dentre todos os canais já modelados até hoje, existem cinco que utilizamos para validar nosso trabalho, os quais são considerados pela comunidade científica mundial

como *benchmark* para teste de transmissão digital e que receberam o nome de canais *Brazil* (*Brazil A, B, C, D e E*)[14].

Abaixo veremos algumas características destes canais e as tabelas de seus respectivos modelamentos.

## 4.1. Canais *Brazil*

### 4.1.1. Canal *Brazil A*

O canal *Brazil A* possui seis multipercursos distintos com pequenos atrasos, possuindo assim um eco do sinal que por ele é transmitido fraco, possibilitando diminuirmos o tamanho do intervalo de guarda (seção 2.3.3.2) sem prejudicarmos a recepção.

A tabela 4.1 contém os multipercursos característicos deste canal.

<i>Brazil A</i>	Atraso	Atenuação	Espalhamento Doppler
<b>Percurso 1</b>	0.0 $\mu s$	0.0 <i>dB</i>	0.0 <i>Hz</i>
<b>Percurso 2</b>	0.15 $\mu s$	13.8 <i>dB</i>	0.0 <i>Hz</i>
<b>Percurso 3</b>	2.22 $\mu s$	16.2 <i>dB</i>	0.0 <i>Hz</i>
<b>Percurso 4</b>	3.05 $\mu s$	14.9 <i>dB</i>	0.0 <i>Hz</i>
<b>Percurso 5</b>	5.86 $\mu s$	13.6 <i>dB</i>	0.0 <i>Hz</i>
<b>Percurso 6</b>	5.93 $\mu s$	16.4 <i>dB</i>	0.0 <i>Hz</i>

Tabela 4.1 – Multipercursos do canal *Brazil A* [14]

A Figura 4.1 apresenta a magnitude da resposta ao impulso do canal *Brazil A* no domínio do tempo.

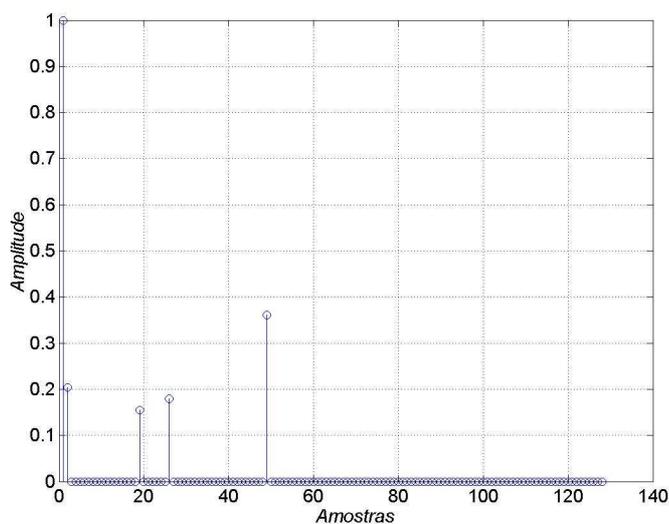


Figura 4.1 – Magnitude da resposta ao impulso do canal *Brazil A* no domínio do tempo.

#### 4.1.2. Canal *Brazil B*

O canal *Brazil B* apresenta ecos de grande atraso relativo, podendo chegar até  $12.7 \mu s$ , e moderadamente fortes, com uma atenuação de apenas  $4 \text{ dB}$ .

Este canal também apresenta seis multipercursos distintos de acordo com a tabela 4.2.

<i>Brazil B</i>	Atraso	Atenuação	Espalhamento Doppler
<b>Percurso 1</b>	$0.0 \mu s$	$0.0 \text{ dB}$	$0.0 \text{ Hz}$
<b>Percurso 2</b>	$0.3 \mu s$	$12.0 \text{ dB}$	$0.0 \text{ Hz}$
<b>Percurso 3</b>	$3.5 \mu s$	$4.0 \text{ dB}$	$0.0 \text{ Hz}$
<b>Percurso 4</b>	$4.4 \mu s$	$7.0 \text{ dB}$	$0.0 \text{ Hz}$
<b>Percurso 5</b>	$9.5 \mu s$	$15.0 \text{ dB}$	$0.0 \text{ Hz}$
<b>Percurso 6</b>	$12.7 \mu s$	$22.0 \text{ dB}$	$0.0 \text{ Hz}$

Tabela 4.2 – Multipercursos do canal *Brazil B* [14]

A Figura 4.2 apresenta a magnitude da resposta ao impulso do canal *Brazil B* no domínio do tempo.

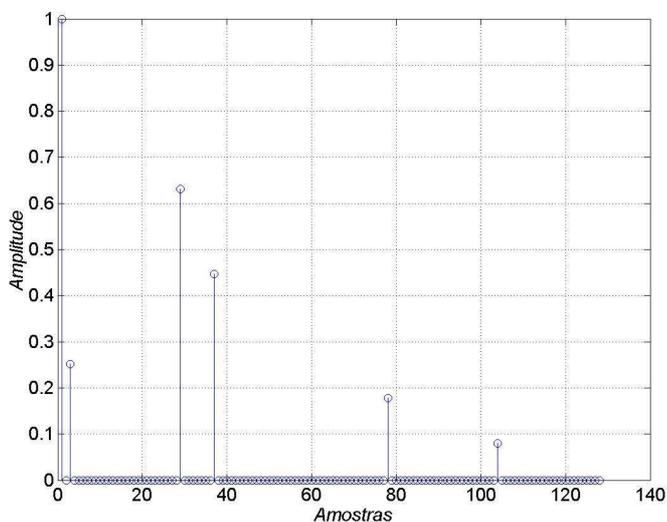


Figura 4.2 – Magnitude da resposta ao impulso do canal *Brazil B* no domínio do tempo.

### 4.1.3. Canal *Brazil C*

O canal *Brazil C* distingue dos dois primeiros principalmente por possuir atrasos menores com grandes intensidades, simulando regiões montanhosas, que podem até mesmo não possuir linhas de visada do transmissor com o receptor.

Seus seis multipercursos estão especificados na tabela 4.3.

<i>Brazil C</i>	Atraso	Atenuação	Espalhamento Doppler
<b>Percurso 1</b>	0.0 $\mu s$	2.8 dB	0.0 Hz
<b>Percurso 2</b>	0.089 $\mu s$	0.0 dB	0.0 Hz
<b>Percurso 3</b>	0.419 $\mu s$	3.8 dB	0.0 Hz
<b>Percurso 4</b>	1.506 $\mu s$	0.1 dB	0.0 Hz
<b>Percurso 5</b>	2.322 $\mu s$	2.5 dB	0.0 Hz
<b>Percurso 6</b>	2.799 $\mu s$	1.3 dB	0.0 Hz

Tabela 4.3 – Multipercursos do canal *Brazil C* [14]

A Figura 4.3 apresenta a magnitude da resposta ao impulso do canal *Brazil C* no domínio do tempo.

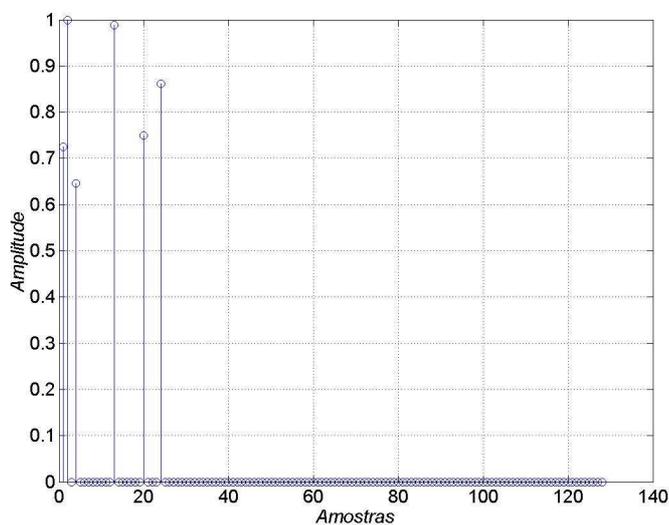


Figura 4.3 – Magnitude da resposta ao impulso do canal *Brazil C* no domínio do tempo.

#### 4.1.4. Canal *Brazil D*

O canal *Brazil D* é semelhante ao canal *Brazil C* em relação a seus ecos, porém deslocados no tempo, sendo assim denominados pré-ecos, simulando uma recepção em um ambiente dito *indoor*.

A tabela 4.4 apresenta seus multipercursos.

<i>Brazil D</i>	Atraso	Atenuação	Espalhamento Doppler
<b>Percurso 1</b>	0.15 $\mu s$	0.1 dB	0.0 Hz
<b>Percurso 2</b>	0.63 $\mu s$	3.8 dB	0.0 Hz
<b>Percurso 3</b>	2.22 $\mu s$	2.6 dB	0.0 Hz
<b>Percurso 4</b>	3.05 $\mu s$	1.3 dB	0.0 Hz
<b>Percurso 5</b>	5.86 $\mu s$	0.0 dB	0.0 Hz
<b>Percurso 6</b>	5.93 $\mu s$	2.8 dB	0.0 Hz

Tabela 4.4 – Multipercursos do canal *Brazil D* [14].

A Figura 4.4 apresenta a magnitude da resposta ao impulso do canal *Brazil D* no domínio do tempo.

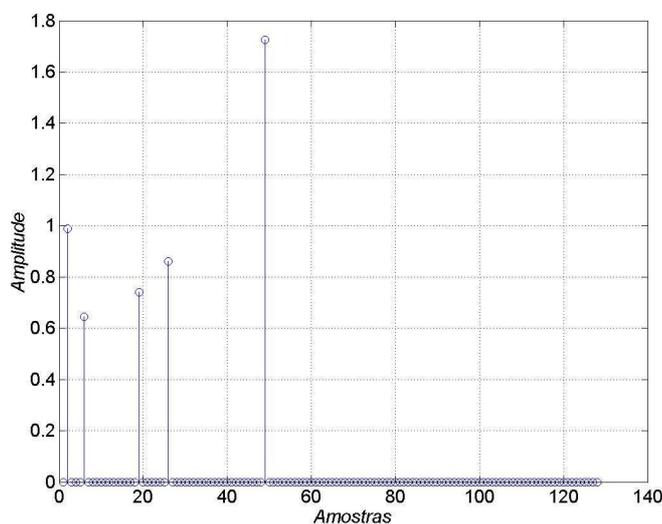


Figura 4.4 – Magnitude da resposta ao impulso do canal *Brazil D* no domínio do tempo.

#### 4.1.5. Canal *Brazil E*

Este canal possui apenas três multipercursos (tabela 4.5) que possibilitam a avaliação de um cenário com rede de frequência única, pois simulam no receptor a chegada de três sinais de mesma amplitude porém com atrasos diferentes, simulando assim um ponto com incidência de sinais de três transmissores distintos.

<i>Brazil E</i>	Atraso	Atenuação	Espalhamento Doppler
<b>Percorso 1</b>	0.0 $\mu$ s	0.0 dB	0.0 Hz
<b>Percorso 2</b>	1.0 $\mu$ s	0.0 dB	0.0 Hz
<b>Percorso 3</b>	2.0 $\mu$ s	0.0 dB	0.0 Hz

Tabela 4.5 – Multipercursos do canal *Brazil E* [14]

A Figura 4.5 apresenta a magnitude da resposta ao impulso do canal *Brazil E* no domínio do tempo.

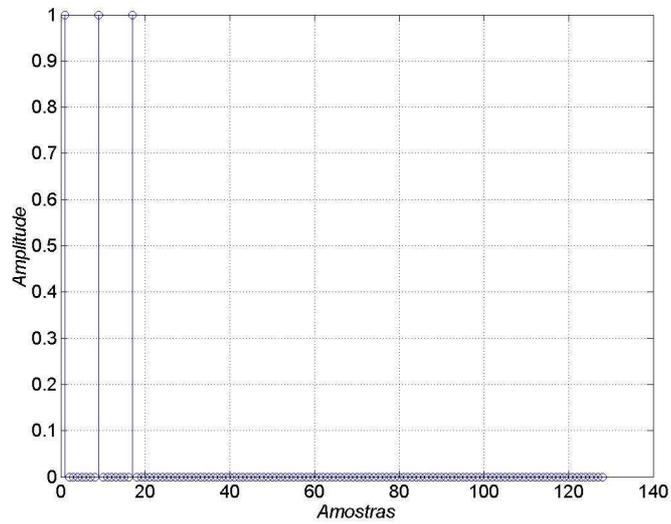


Figura 4.5 – Magnitude da resposta ao impulso do canal *Brazil E* no domínio do tempo.

## 5. Desenvolvimento

### 5.1. Introdução

Conforme apresentado na introdução geral deste trabalho, este consiste em identificar o início de um *frame* de dados de um sistema de comunicação digital *wireless* sem a necessidade de adição de portadoras pilotos ou qualquer outro dado extra distribuído dentro de um *frame*.

Nossa proposta é a substituição das todas portadoras pilotos distribuídas em um *frame* (ISDB possui 1/13 de informação de um *frame* gastos em portadoras pilotos) e outros dados para efeito de sincronismo de *frame* por apenas um preâmbulo – descrito na próxima seção – utilizando então a não linearidade de uma RNA no contexto de detecção de padrão para identificá-lo e apontar com precisão o início de um *frame*.

### 5.2. Preâmbulo

O preâmbulo é um símbolo OFDM que pode possuir informações de configuração do receptor. Ele é adicionado no início de um *frame*, como se fosse um cabeçalho, para localizarmos o início deste.

Este, foi criado baseado no método desenvolvido por Zhongshan Zhang et. al. [4], cujo método utiliza apenas as portadoras ímpares da IFFT para dados, zerando as portadoras pares, obtendo assim, um símbolo OFDM no tempo com duas metades iguais, conforme a Equação 5.1.

$$G_{(i)} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{\frac{j2\pi ik}{N}} \quad i = 0,1,2,\dots, N-1 \quad (5.1)$$

sendo obedecidos às condições das Equações 5.2 e 5.3:

$$G_k = s_{\left(\frac{N+k}{2}\right)} \text{ para } 0 \leq k \leq \frac{N}{2}-1 \quad (5.2)$$

$$G_k = s_{(N-k)} \text{ para } 1 \leq k \leq \frac{N}{2}-1 \quad (5.3)$$

onde  $G_{(i)}$  é o símbolo de saída,  $N$  é o número de pontos da IFFT e  $X_k$  é o símbolo IQ de entrada pertencente à constelação de referência da modulação digital adotada (QAM, PSK, etc...)

Utilizando esta técnica obtemos então um símbolo OFDM que possui duas metades iguais, a mostra a Figura 5.1 nos mostra um preâmbulo de no tempo de 64 pontos.

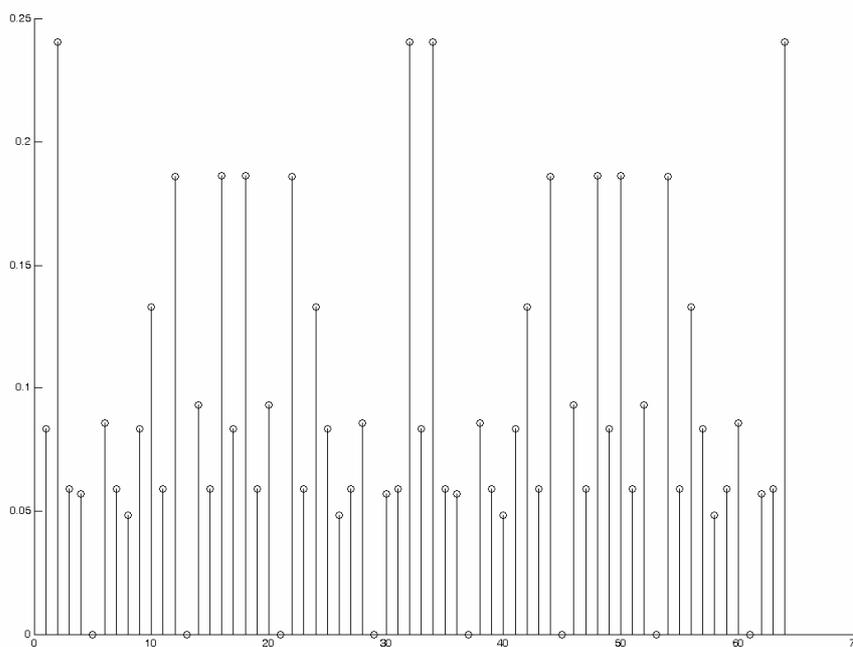


Figura 5.1 – Preâmbulo no Tempo com 64 pontos

Note que na Figura 5.1 se olharmos da esquerda para a direita ou vice versa a Figura possui a mesma forma.

A modulação do preâmbulo é BPSK (Binary Phase Shift Keying) pelo fato de ser uma modulação mais robusta em relação às demais em um meio com ruído e multipercurso. Já nos demais símbolos o tipo de modulação não importa, já que não são formados de maneira igual ao preâmbulo, não possuindo assim, duas metades iguais. Mas para o desenvolvimento e validação utilizamos todos demais símbolos após os preâmbulos com a mesma modulação BPSK, estressando assim o grau de incerteza na tomada de decisão da RNA.

### 5.3. Definição da RNA

Estando a parte do preâmbulo decidida e pronta, partimos para a definição da RNA que mais se adequava a este critério. Levando em consideração a complexidade dos códigos e tempo de processamento, optou-se por uma rede de detecção de padrão do tipo RBF.

A rede é composta por um vetor de entrada com 2048 valores IQ complexos (i.e.,  $I + jQ$ ) que formam um símbolo OFDM no modo 2K de transmissão dos dois sistemas multiportadoras sob estudo, e 8 funções de base radial, que neste caso são Gaussianas.

Para treinarmos a rede geramos um arquivo com 1000 símbolos OFDM do tipo preâmbulo e 1000 símbolos OFDM normais, onde, para cada símbolo OFDM composto por um preâmbulo a saída da rede foi treinada para apresentar resultado “1” e para cada símbolo OFDM de qualquer outra informação ou dados a rede foi treinada para apresentar na saída resultado “0”.

### 5.4. Método para Validação

O outro método que usaremos para validar a utilização da rede foi o da correlação, conforme Shinsuke Hara & Ranjee Prasad [1].

Neste, o símbolo OFDM de 2048 amostras é dividido em duas janelas de 1024 amostras, sendo feito o somatório do produto entre as amostras das duas janelas, como nos mostra a Equação 5.4.

$$S_{(d)} = \sum_{m=0}^{N/2-1} \left( d_{(m)} d_{\left(m+\frac{N}{2}\right)}^* \right) \quad (5.4)$$

onde  $S_{(d)}$  é à saída da correlação para cada deslocamento da janela sobre o vetor de entrada,  $d_{(m)}$  é a amostra da janela a ser multiplicada pela amostra  $d_{\left(m+\frac{N}{2}\right)}^*$  sendo  $N$  o tamanho da janela, que neste caso é de 2048 amostras.

## 5.5. Localização do Início do *Frame*

Como o intervalo de guarda (IG) é apenas uma cópia do final do símbolo (seção 2.3.3.2), utilizamos uma característica da correlação, demonstrada por Shinsuke Hara & Ranjee Prasad [1], que explica o fato de haver um platô na saída justamente onde deveria estar o IG, pois, ao correlacionar o símbolo, será destacado todo o IG, como mostra a Figura 5.2.

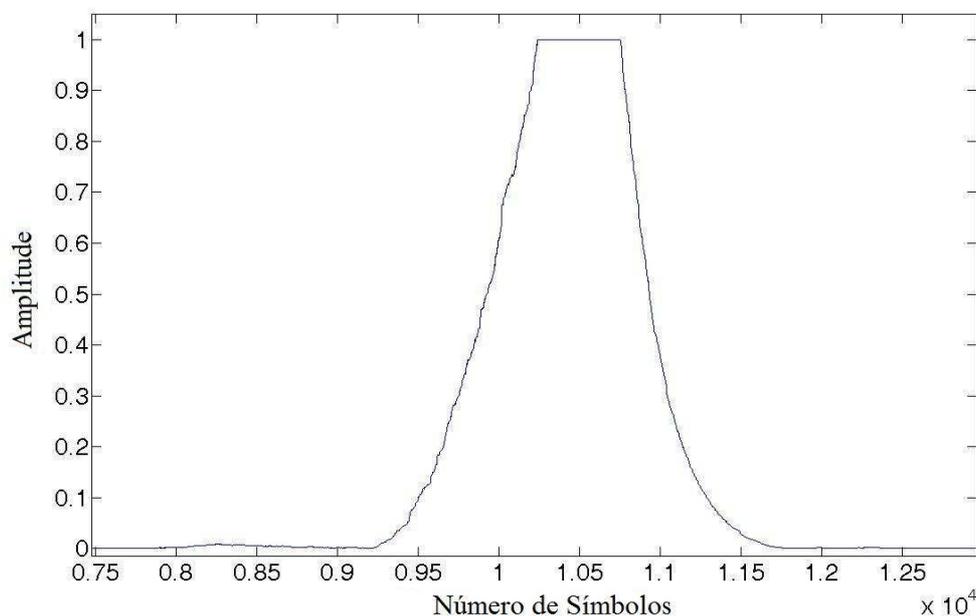


Figura 5.2 – Platô do Intervalo de Guarda do Símbolo de tamanho 1/4

Com isso acharemos um ponto dentro do IG de maior amplitude que não é necessariamente o início exato do símbolo, o que causará apenas um giro na constelação IQ após a FFT no receptor[1]. Este giro é corrigido posteriormente no sistema completo, pelo sincronismo de portadora [6], que não é do escopo deste trabalho.

## 5.6. Resultados Obtidos

Para efeito do desenvolvimento da nova técnica apresentada neste trabalho e para efeito de comparação com a técnica baseada em correlação, utilizamos arquivos de 10000 *frames* com IG de 1/4, 1/8, 1/16, 1/32 e 1/64 do tamanho do símbolo. Para ambas as técnicas, submetemos o sinal em banda-base que transporta o frame a um simulador de canal AWGN com uma Relação Sinal Ruído (SNR) de 0, 2, 4, 6, 8 e 10 dB's para todos os IGs, tendo assim uma seqüência de testes exaustivos para as duas técnicas.

Estes arquivos foram submetidos aos dois processos de identificação – o baseado em correlação e o processo baseado na RNA-RBF. Na Figura 5.3 podemos ver que as curvas (NET) que representam o erro da RNA-RBF possuem uma FSER (*Frame Synchronization Error Rate*) muito menor que nas curvas que representam o erro resultante do processo de identificação por correlação (COR).

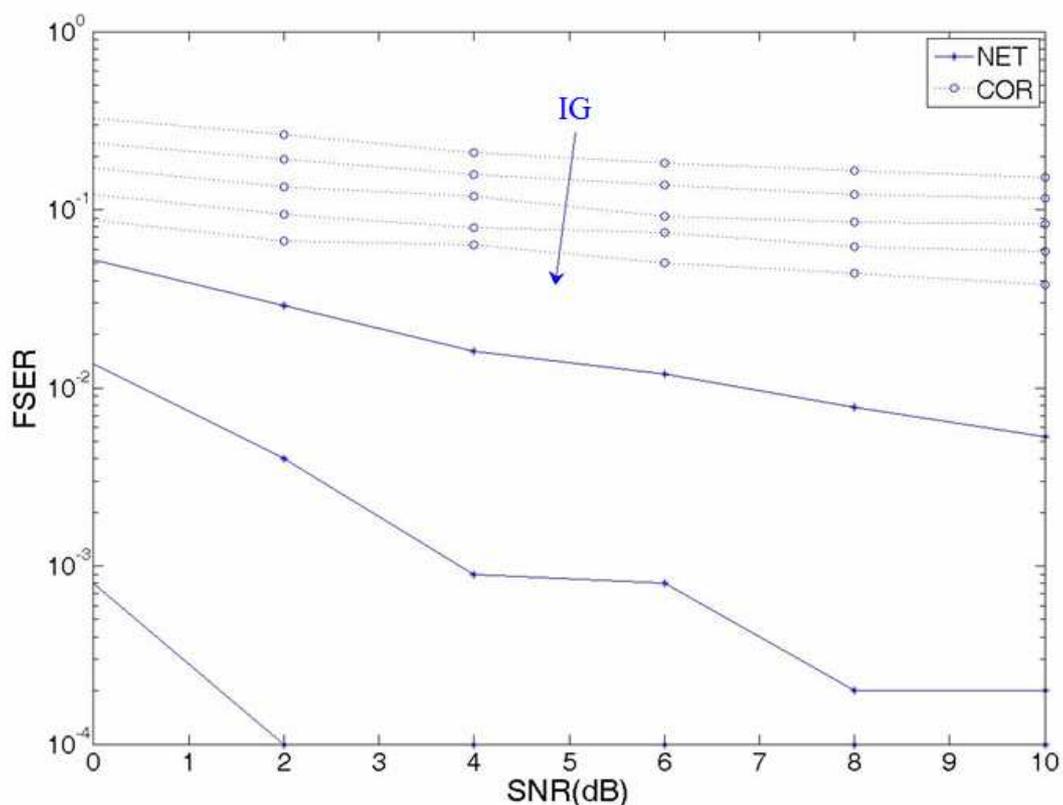


Figura 5.3 – Curva de FSER da identificação por RNA e por Correlação para IGs e SNR diferentes

A seta na Figura 5.3 representa o sentido em que aumenta o IG aplicado aos símbolos OFDM dos *frames* que foram submetidos ao processo de identificação. As linhas tracejadas representam os erros resultantes do processo de identificação por correlação e as outras representam os erros resultantes do processo de identificação por RNA.

Note também que para intervalos maiores como o de 1/4 (512 valores IQ) e 1/8 (256 valores IQ) não conseguimos encontrar quantidades de erros significativas para plotá-los na Figura 5.3.

Cabe lembrar aqui que a Figura 5.3 é resultante da identificação do preâmbulo no início de cada *frame*. Porém como sugerimos o novo método para qualquer sistema multiportadora OFDM, até mesmo nos já existentes, não poderíamos aplicá-lo em um receptor desta maneira, pois o transmissor não prevê o preâmbulo da maneira estudada neste trabalho.

Assim, partimos para o próximo passo que consiste em fazer a RNA identificar os IGs de cada símbolo OFDM, desta maneira garantimos que o método pode ser adicionado em qualquer receptor de sistemas já existentes.

Para desenvolvermos este outro teste, criamos arquivos com IG igual a 256 valores IQs e fomos diminuindo a janela de entrada da RNA até conseguirmos uma configuração com uma janela de 32 valores complexos  $I + jQ$ , ou seja, para cada símbolo OFDM a RNA utiliza para sua entrada os primeiros e os últimos 32 valores símbolos, que representariam o menor IG que um dos sistemas OFDM citados pode possuir. É importante lembrar aqui que a rede opera com valores de entrada complexos  $I + jQ$  gerados após a IFFT no transmissor. Portanto, embora estes valores estejam em quadratura, eles não constituem os símbolos IQ da constelação de referência da modulação digital. Os símbolos IQ da constelação de referência da modulação digital são aplicados na entrada da IFFT no transmissor [5][6].

A Figura 5.4 mostra os 2x32 pontos os quais são aplicados aos 64 nós de entrada da RNA-RBF.



Figura 5.4 – Montagem da Janela da RNA

Desta maneira conseguimos diminuir também a quantidade de neurônios necessários para esta aplicação, ficando apenas dois neurônios com 64 entradas e apenas uma saída. Torna-se assim uma RNA-RBF de fácil implementação em hardware para um trabalho futuro.

Note que os nós de entrada da RNA-RBF não utilizam o valor complexo  $I + jQ$  mas sim o seu módulo conforme a Equação 5.5. Isto reduz a sensibilidade do processo de identificação ao desvio Doppler gerado por multipercurso dinâmico.

$$e_n = \sqrt{(I_n^2 + Q_n^2)} \quad (5.5)$$

onde  $e_n$  é a entrada da rede conforme o valor  $I_n + jQ_n$  que chega ao receptor.

O treino da RNA foi feito de maneira que ela apresente em sua saída o valor “10” se identificar um intervalo de guarda e “0” quando não identificar, porém, devido às aproximações, já que esta RNA foi desenvolvida totalmente em aritmética de ponto fixo para facilitar uma futura implementação em hardware, ela apresenta resultados que variam entre “10” e 0, mas que permitem a identificação de um início de *frame*.

Primeiramente treinamos a RNA-RBF para localizar apenas os intervalos de guarda dos preâmbulos, assim obtemos os resultados das Figuras 5.5 á 5.9 dos canais *Brazil A* até *E* respectivamente.

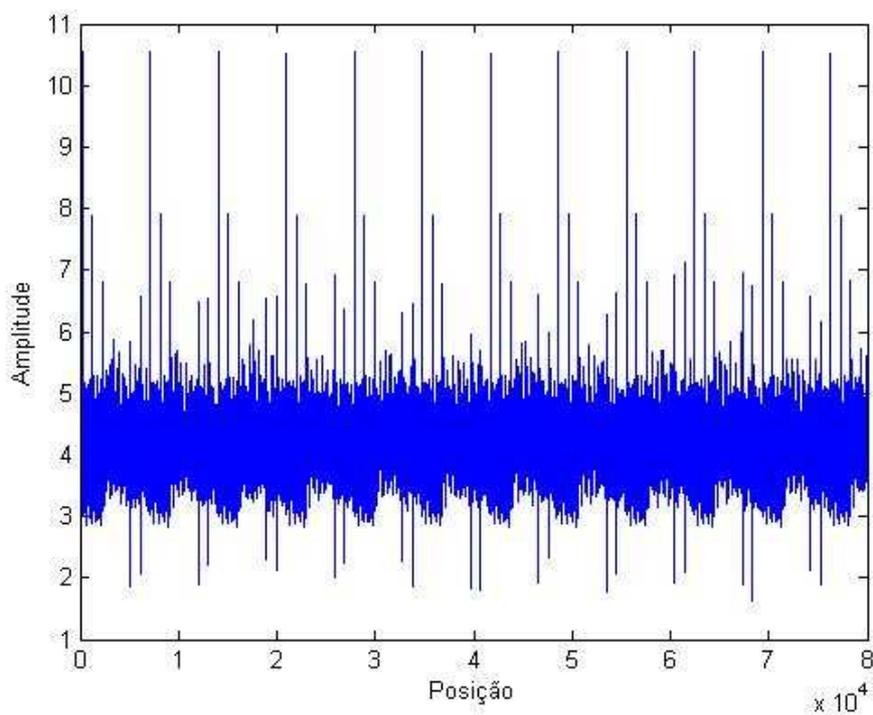


Figura 5.5 – Saída da RNA para o canal *Brazil A*

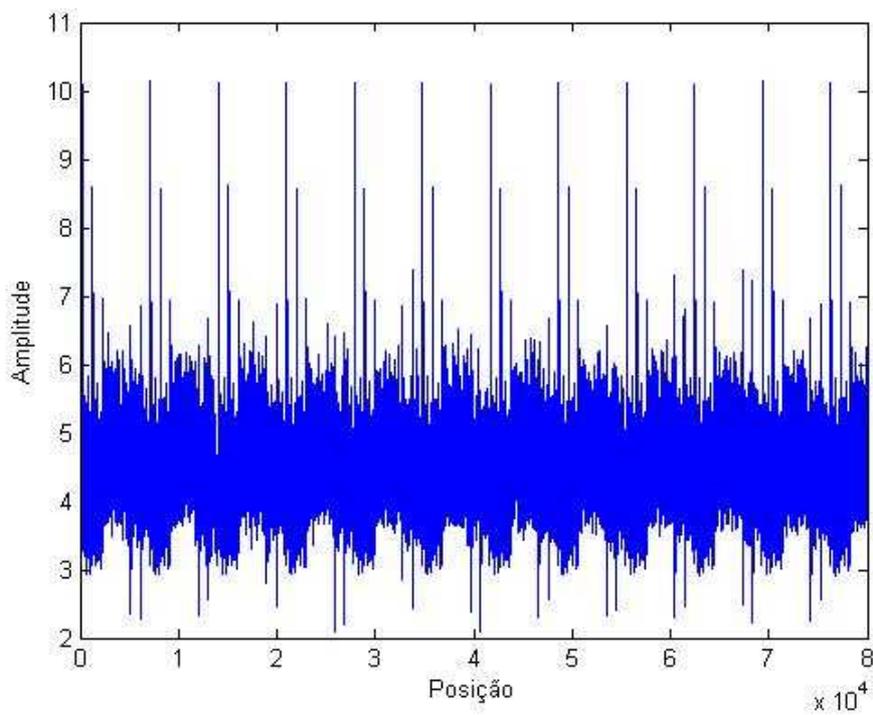


Figura 5.6 – Saída da RNA para o canal *Brazil B*

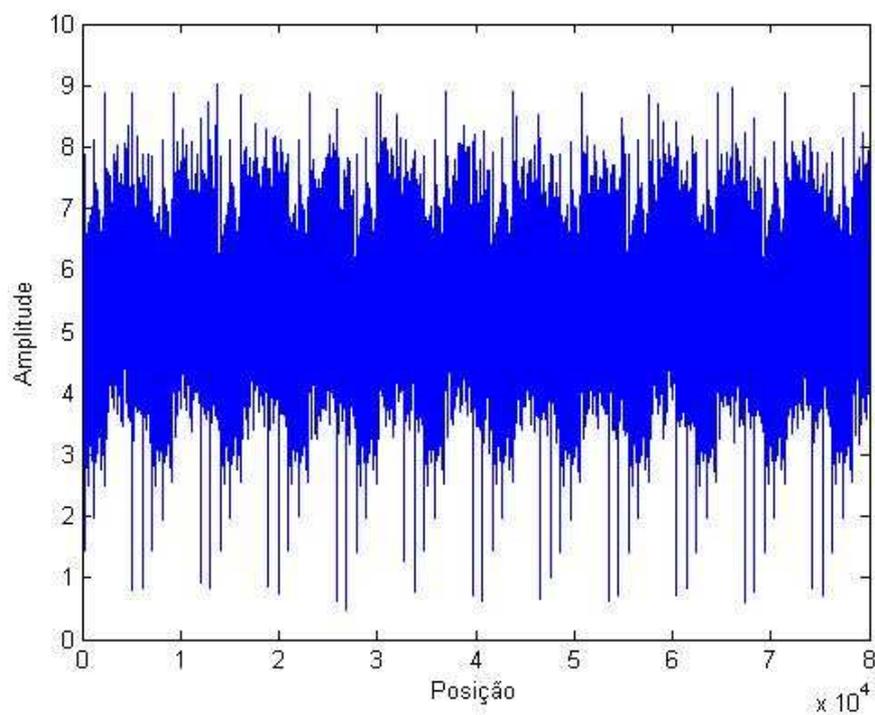


Figura 5.7 – Saída da RNA para o canal *Brazil C*

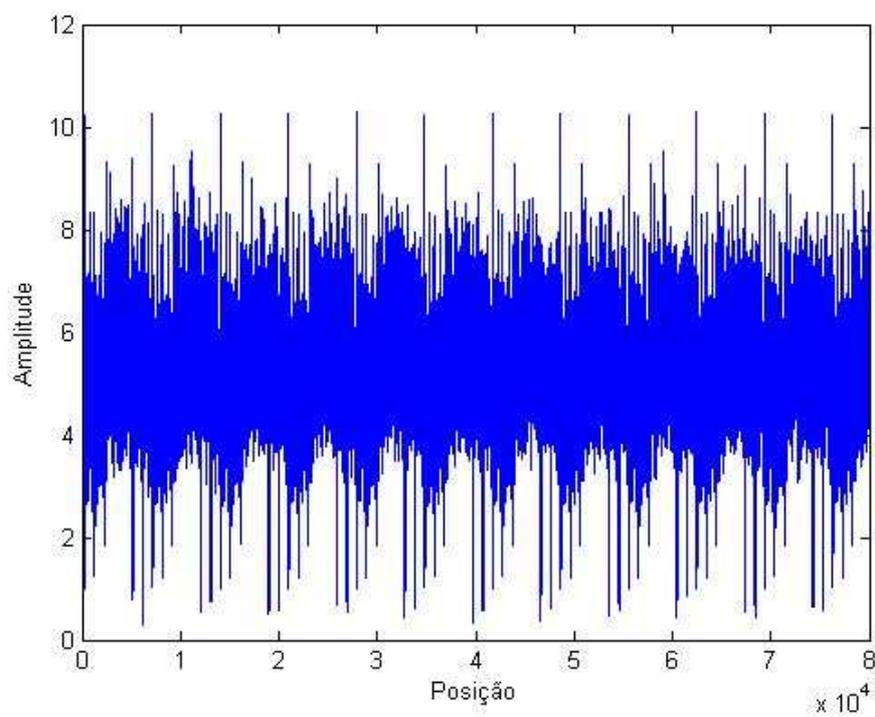


Figura 5.8 – Saída da RNA para o canal *Brazil D*

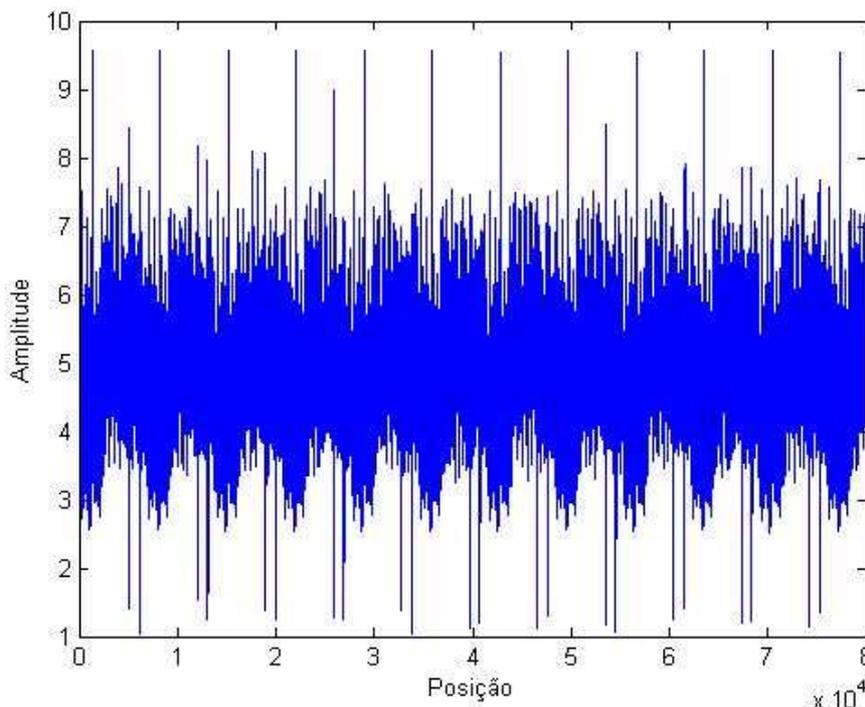


Figura 5.9 – Saída da RNA para o canal *Brazil E*

Note que entre as figuras acima a que representa o canal *Brazil C* é a que apresentou uma dificuldade de tomada de decisão por parte da rede, devido este ser um canal que simula regiões montanhosas que pode até mesmo não haver linha de visada entre o transmissor e o receptor, porém ainda podemos ver nitidamente o início de um *frame*.

Já nas figuras que representam os demais canais, podemos ver a distinção total e praticamente na posição ideal e correta os picos que representam o início de um *frame*.

Vejamos a Figura 5.5 que representa o canal *Brazil A*. Se dermos um zoom no primeiro pico desta obtendo a Figura 5.10, percebemos que este pico está exatamente na posição 257, que é o início do *frame* exato após o intervalo de guarda.

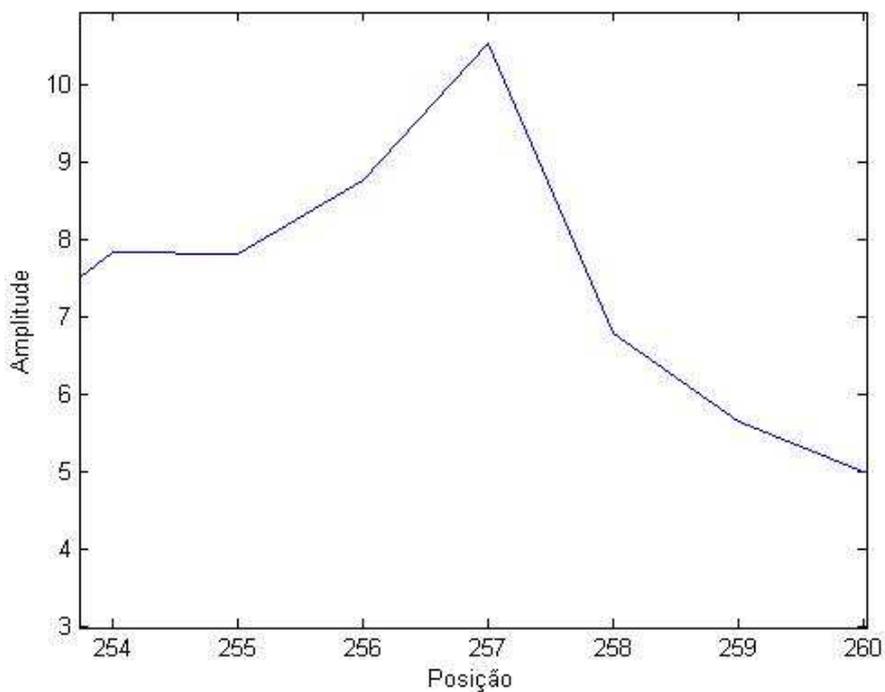


Figura 5.10 – Primeiro pico da Figura 5.5

Como utilizamos a configuração de um preâmbulo mais dois símbolos de dados, podemos fazer o seguinte cálculo para descobrir a posição  $S_n$  do segundo *frame*, como mostra a Equação 5.6.

$$S_n = N_{IQ} \times 3 \times n + 4 \times IG \times n + IG + 1 \quad (5.6)$$

onde  $S_n$  é a posição de saída,  $N_{IQ}$  é o número de símbolos IQs que formam um Símbolo OFDM, 3 é o número de símbolos OFDM em cada *frame*,  $n$  é o número do *frame*,  $IG$  é o tamanho do intervalo de guarda, 4 é o número de IG entre cada *frame* e 1 é a primeira posição do símbolo OFDM.

Logo se substituirmos e calcularmos a posição inicial do segundo *frame*, obtemos a Equação 5.7.

$$S_1 = 2048 \times 3 \times 1 + 4 \times 256 \times 1 + 1 = 7169 \quad (5.7)$$

A Figura 5.11 nos mostra a posição inicial do segundo *frame* para o canal *Brazil A*, podemos assim, comparar com o resultado da equação (5.7) e validá-la.

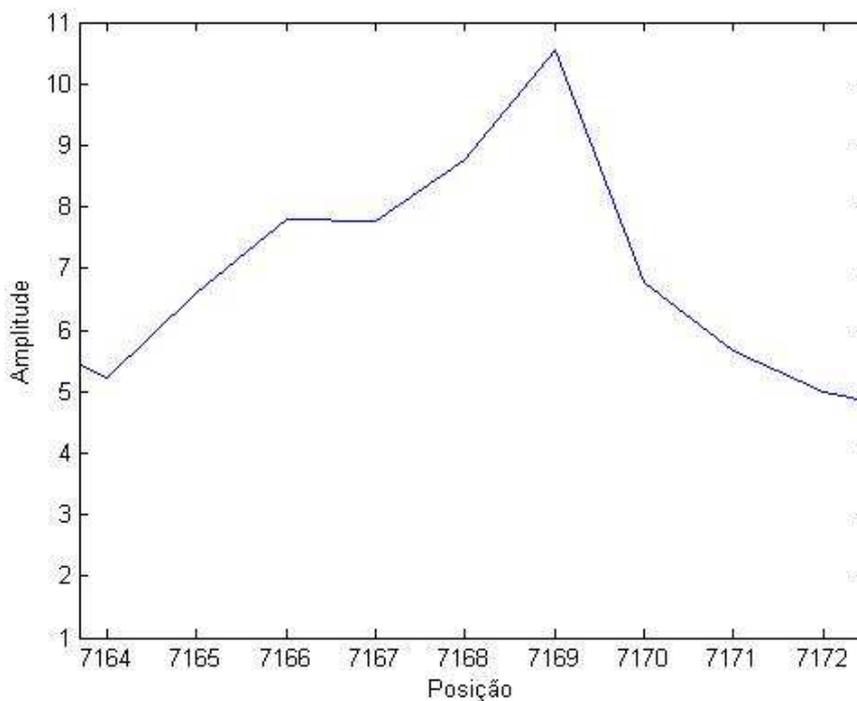


Figura 5.11 – Segundo pico da Figura 5.5

Assim, demonstramos o correto funcionamento do método proposto na identificação do IG de cada preâmbulo. Nos próximos passos, treinamos a RNA-RBF para encontrar os IGs de todos os símbolos OFDM contidos em um *frame*. Ou seja, treinaremos a RNA para encontrar os IGs de quaisquer símbolo OFDM, não necessitando mais de um preâmbulo, garantindo assim que o método funcione como o da correlação em sistemas já existentes, identificando todos os IGs de todos símbolos OFDM que chegam ao receptor, podendo ser aplicado nos receptores de sistemas já em uso como os citados anteriormente. Os resultados são mostrados nas Figuras 5.12 a 5.16.

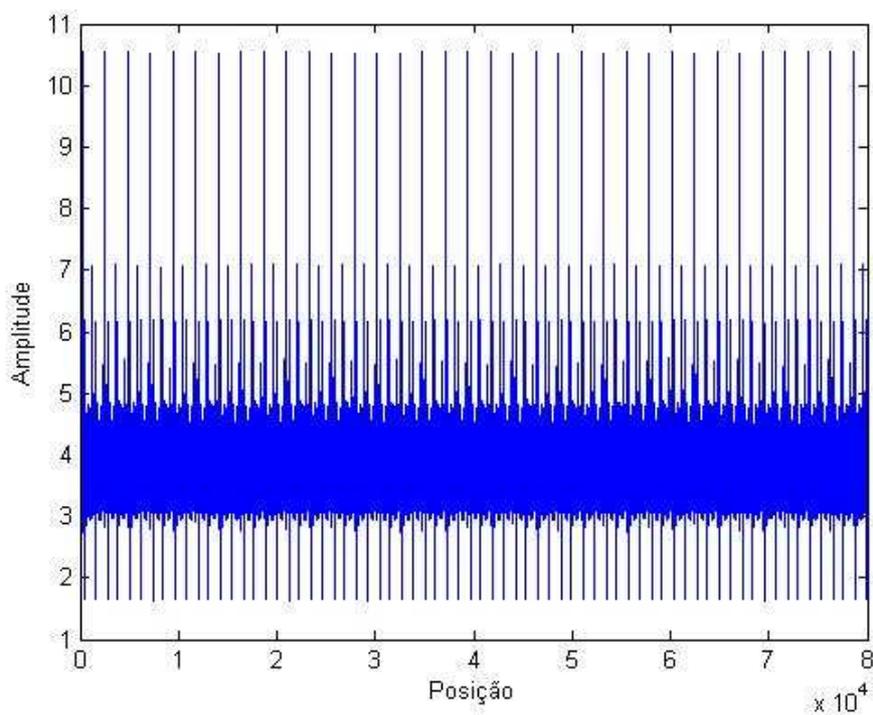


Figura 5.12 – Saída da RNA para o canal *Brazil A* identificando todos IGs.

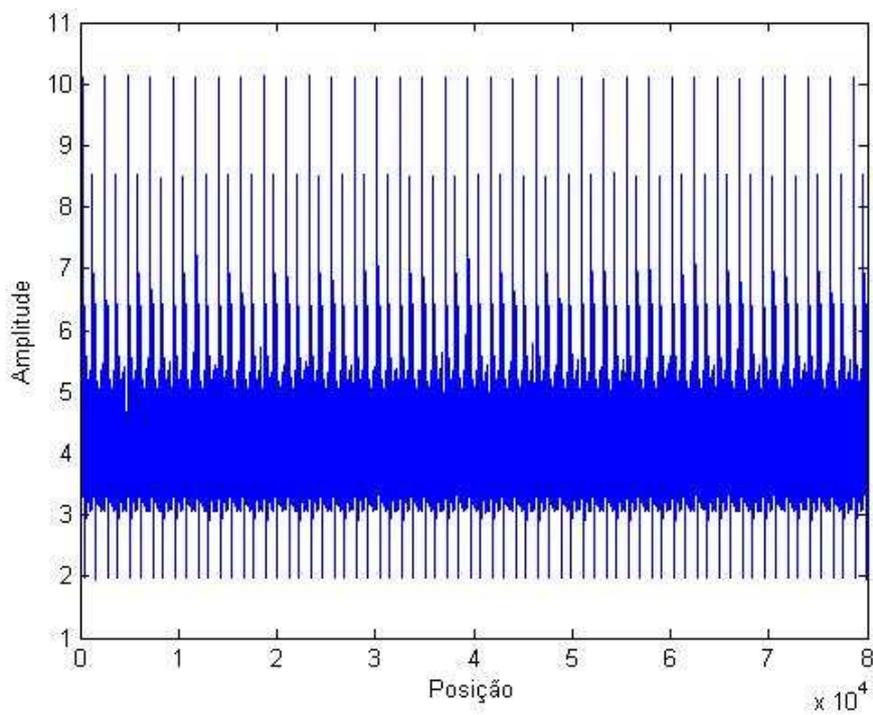


Figura 5.13 – Saída da RNA para o canal *Brazil B* identificando todos IGs.

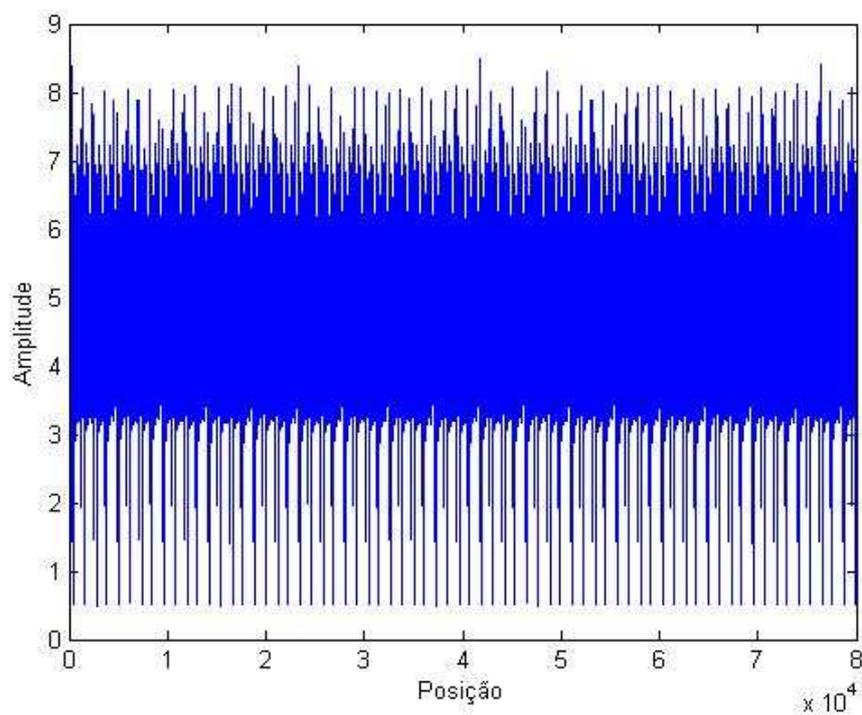


Figura 5.14 – Saída da RNA para o canal *Brazil C* identificando todos IGs.

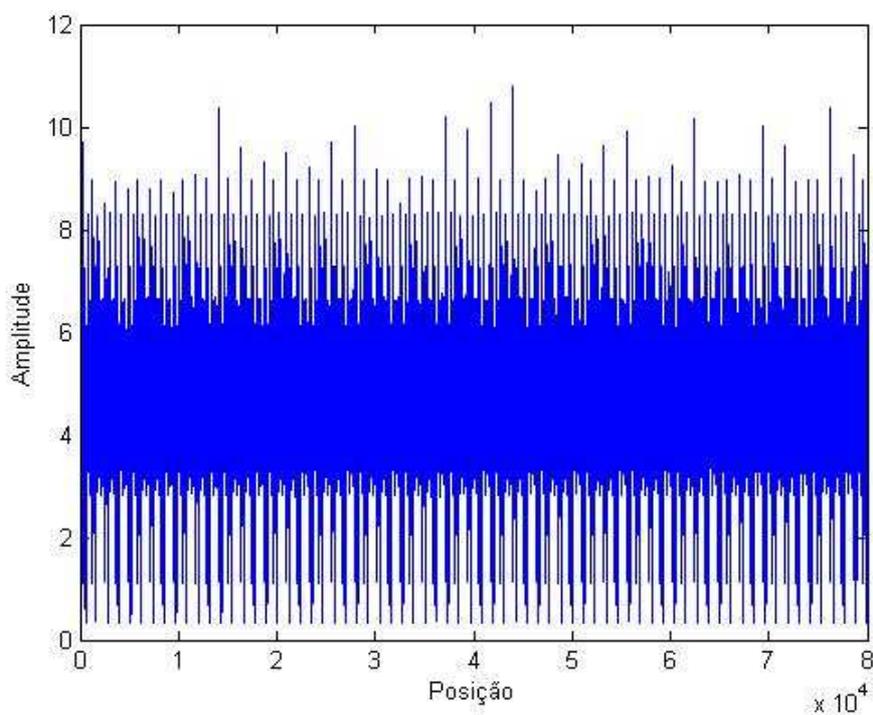


Figura 5.15 – Saída da RNA para o canal *Brazil D* identificando todos IGs.

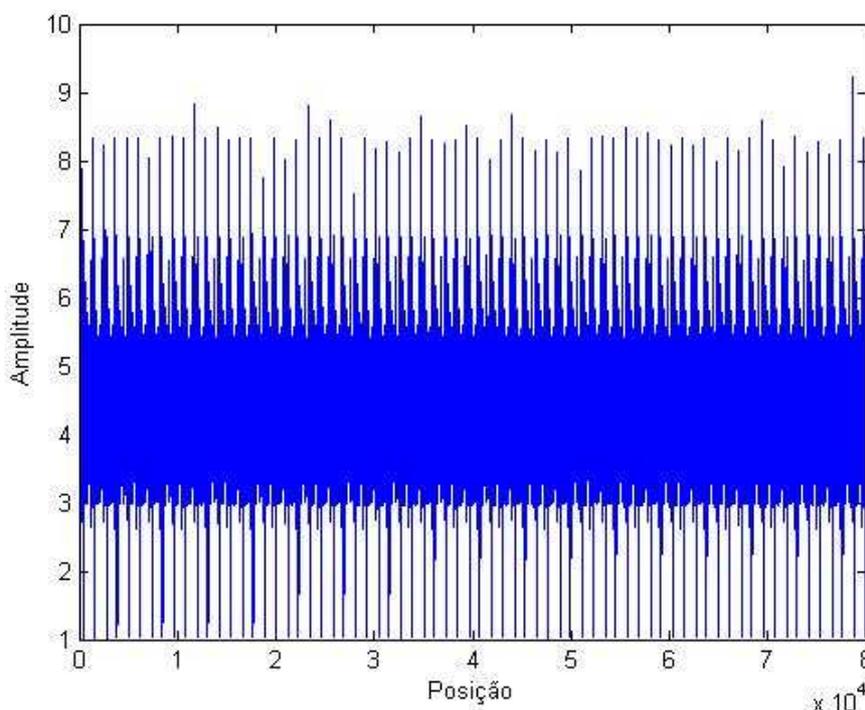


Figura 5.16 – Saída da RNA para o canal *Brazil E* identificando todos IGs.

Agora nossa equação para localização de cada símbolo OFDM é a seguinte:

$$S_n = N_{IQ} \times n + (n+1) \times IG + 1 \quad (5.8)$$

onde  $S_n$  é a posição inicial do símbolo OFDM,  $N_{IQ}$  é a quantidade de símbolos IQs em um símbolo OFDM,  $n$  é o número do símbolo OFDM e IG é o tamanho do intervalo de guarda.

Ampliando a Figura 5.12 podemos ver a posição do primeiro símbolo OFDM que será quando  $n=0$ , como mostra a equação (5.9) e para o segundo símbolo OFDM utilizamos  $n=1$  e obtemos a Equação 5.10. As Figuras 5.17 e 5.18 mostram estas duas posições iniciais dos dois primeiros símbolos OFDM para o canal *Brazil A*.

$$S_0 = 2048 \times 0 + (0+1) \times 256 + 1 = 257 \quad (5.9)$$

$$S_1 = 2048 \times 1 + (1 + 1) \times 256 + 1 = 2561 \quad (5.10)$$

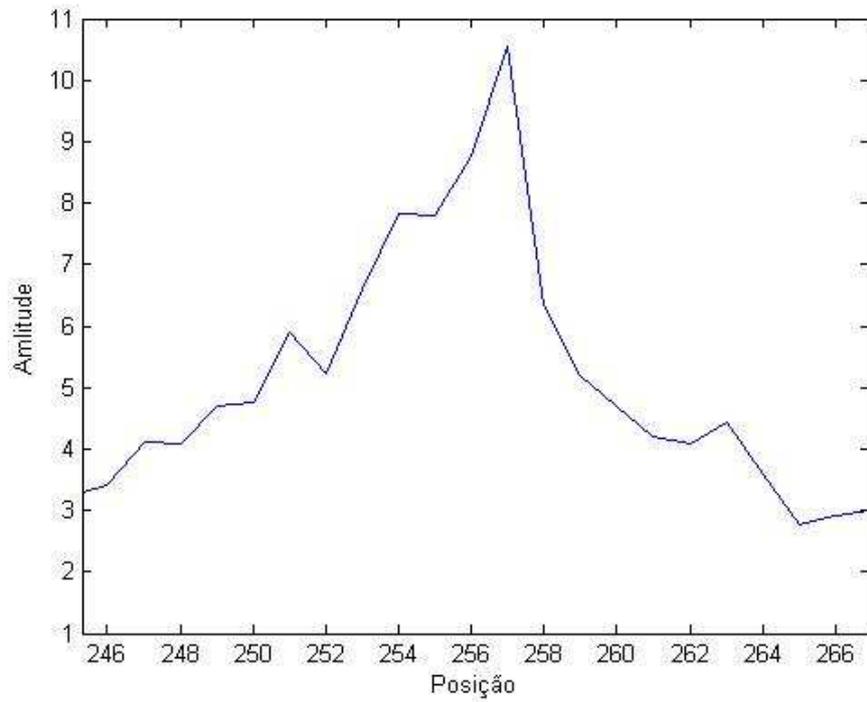


Figura 5.17 – Posição Inicial do Primeiro Símbolo OFDM do Canal *Brazil A*

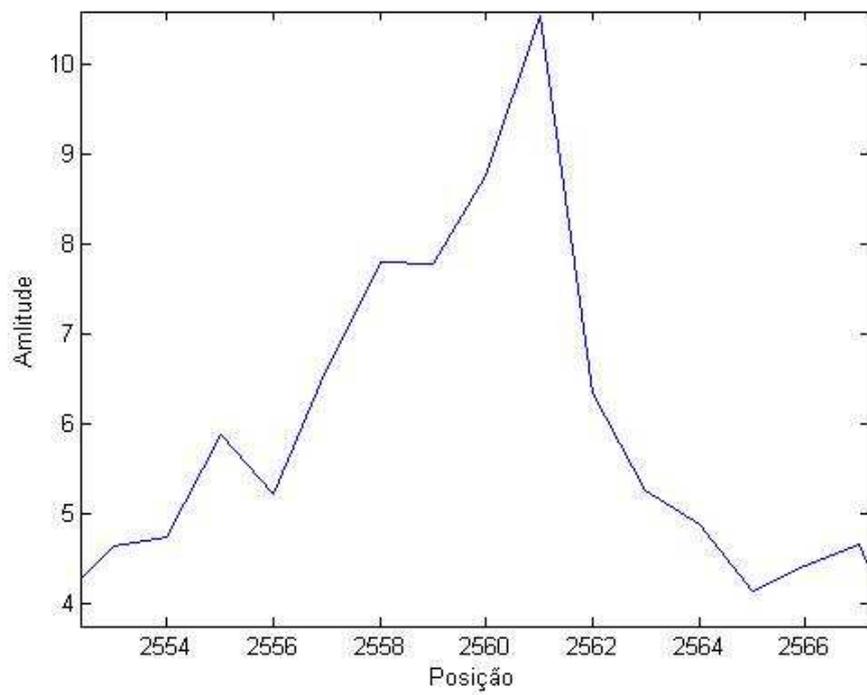


Figura 5.18 – Posição Inicial do Segundo Símbolo OFDM do Canal *Brazil A*

Novamente podemos perceber que a RNA identificou o IG de cada símbolo OFDM, validando o método proposto e garantindo seu funcionamento para qualquer sistema já existente ou que venham a serem desenvolvidos.

Nestes novos arquivos simulados obtivemos valores de erro baixo as curvas de FSER, já que o método identifica precisamente o início de um símbolo OFDM e quando erra a posição, o erro ocorre para dentro do IG, o que não é considerado um erro conforme já comentamos na seção 5.5.

## 6. Conclusão

O sincronismo é talvez a etapa mais crucial de um sistema de comunicação digital, pois sem ele não é possível haver demodulação ou decodificação. O sincronismo de *frame* em sistemas multiportadoras é usualmente efetuado mediante detecção por correlação, que é um processo baseado em estatística de segunda ordem. Mostramos neste trabalho que é possível utilizar estatísticas de ordem superior para efetuar tal processo de detecção – especificamente, uma rede neural do tipo RBF foi utilizada como detetor. Uma rede neural RBF, devido à função de ativação exponencial quadrática (gaussiana) dos neurônios da camada escondida, implicitamente efetua operações utilizando estatística de ordem superior. Os resultados mostraram uma melhora significativa na performance do processo de sincronismo em sistemas *wireless* multiportadora, obtendo-se um erro de detecção menor do que o consagrado método que utiliza correlação.

Foi observado que a detecção com RNA-RBF obteve um menor erro do que a detecção por correlação para todos os intervalos de guardas simulados em um canal. A detecção por correlação, por sua vez, não é afetada para diferentes intervalos de guarda, mantendo seu erro praticamente igual (figura 5.3), não apresentando melhora de desempenho considerável em função de um intervalo de guarda maior. Note que o aumento do intervalo de guarda diminui a taxa de transmissão do sistema.

Nota-se também que na detecção por RBF não ocorre o platô característico da detecção por correlação (seção 5.5), não necessitando assim uma de uma etapa para efeito da decisão da posição inicial, já que, sempre haverá um ponto de amplitude mais elevada para identificar esta posição.

Para trabalhos futuros pretende-se implementar em hardware o método de detecção aqui proposto aplicando técnicas de desenvolvimento em linguagem VHDL ou para processadores de sinais digitais (DSP's), garantindo, assim, a exequibilidade do método.

## 7. Bibliografia

- [1] Shinsuke Hara & Ranjee Prasad, “Multicarrier Techniques for 4G Mobile Communications”, Boston, Artech House, 2003.
- [2] Van Nee, R., and R. Prasad, OFDM for Wireless Multimedia Communications, Norwood, MA: Artech House, 2000.
- [3] Proakis, J. G., *Digital Communications, Fourth Edition*, New York: McGraw-Hill, 2001.
- [4] ATSC. (2007). *ATSC Digital Television Standard*. Advanced Television Systems Committee, Washington.
- [5] ETSI. (2004). *Digital Video Broadcasting*. European Telecommunications Standards Institute, Sophia Antipolis - França.
- [6] ARIB. (2005). *Transmission System for Digital Terrestrial Television Broadcasting*. Association of Radio Industries and businesses.
- [7] M.C.F De Castro, “*Predição Não-Linear de Séries Temporais Usando Redes Neurais RBF por Decomposição em Componentes Principais*”, UNICAMP, Campinas - SP, Março de 2001
- [8] S. Haykin, *Adaptive Filter Theory*, 3<sup>rd</sup> ed., Prentice Hall, Upper Saddle River, New Jersey, 1996.
- [9] Zhongshan Zhang, Keping Long, Ming Zhao, and Yuanan Liu, “*Joint Frame Synchronization and Frequency Offset Estimation in OFDM Systems*”, *IEEE Trans. On Broadcasting*, vol. 51, NO.3, September 2005
- [10] Prasad, R., *Universal Wireless Personal Communications*, Norwood, MA: Artech House, 1998.
- [11] D. Lee and K. Cheun, “*A new symbol timing recovery algorithm for OFDM systems*,” *IEEE Trans. Cons. Electron.*, vol. 43, no. 3, Aug. 1997

- [12] Ureten, O.; Tascioglu, S.; Serinken, N.; Yilmaz, M. “*Search for OFDM synchronization waveforms with good aperiodic autocorrelations,*” *Electrical and Computer Engineering*, 2004. Canadian Conference on Volume 1, 2-5 May 2004 Page(s):13 - 18 Vol.1
- [13] Santella, G., “*A frequency and symbol synchronization system for OFDM signals: architecture and simulation results*”, *Vehicular Technology, IEEE Transactions on* Volume 49, Issue 1, Jan. 2000 Page(s):254 – 275
- [14] Communications Research Centre Canada, “*Results of the Laboratory Evaluation of an 8 MHz ADTB-T Television System for Terrestrial Broadcasting*”, *Final Report (Version 3.0)*, January 15, 2003.
- [15] S. Lin and D. Costelloz, “*Error Control Coding: Fundamentals and Applications*”, Prentice-Hall, Englewood Cliffs, NJ, 2004.
- [16] S. Wicker, “*Error Control Systems for Digital Communication and Storage*”, Prentice Hall , 1995.