

UNIVERSIDADE FEDERAL FLUMINENSE
CENTRO TECNOLÓGICO
MESTRADO EM ENGENHARIA DE TELECOMUNICAÇÕES

CAROLINA MEDEIROS CARVALHO

UM ESTUDO SOBRE RÁDIOS DE *SOFTWARE*

NITERÓI
2007

CAROLINA MEDEIROS CARVALHO

UM ESTUDO SOBRE RÁDIOS DE *SOFTWARE*

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre.

Área de concentração: Sistemas de Telecomunicações

Orientador: Prof. Dr. Luiz Pinto de Carvalho

NITERÓI
2007

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

C331 Carvalho, Carolina Medeiros.
Um estudo sobre rádios de software / Carolina Medeiros
Carvalho. – Niterói, RJ : [s.n.], 2007.
132 f.

Orientador: Luiz Pinto de Carvalho.
Dissertação (Mestrado em Engenharia de Telecomunicações) -
Universidade Federal Fluminense, 2007.

1. Engenharia de telecomunicações .2. Software. 3. Rádios de
software .4. Processamento de sinais-técnicas digitais. I. Título.

CDD 621.382

CAROLINA MEDEIROS CARVALHO

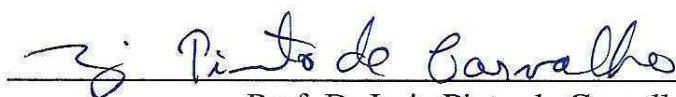
UM ESTUDO SOBRE RÁDIOS DE SOFTWARE

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre.

Área de concentração: Sistemas de Telecomunicações

Aprovada em 06 de novembro de 2007

BANCA EXAMINADORA



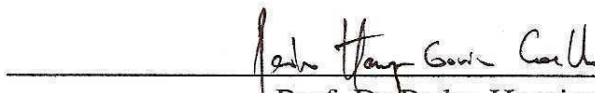
Prof. Dr Luiz Pinto de Carvalho - Orientador
Universidade Federal Fluminense



Prof. Dr. Murilo Bresciani de Carvalho
Universidade Federal Fluminense



Prof. Dr. Alexandre Santos de la Vega
Universidade Federal Fluminense



Prof. Dr Pedro Henrique Gouveia Coelho
Universidade do Estado do Rio de Janeiro

NITERÓI
2007

A Deus, Caminho, Verdade e Vida, por Seu amor, proteção e direção. A meus familiares, sobretudo a meus pais, Ronaldo e Rosane, e a meu marido Augusto, por todo carinho e apoio. Ao Professor Luiz Pinto de Carvalho pela disposição em ajudar sempre.

AGRADECIMENTOS

Agradeço ao Professor Luiz Pinto de Carvalho pelas muitas horas dispensadas a mim nas discussões da teoria e direcionamento para a elaboração deste trabalho. Sem ele, eu não teria conseguido.

Agradeço também a todos os professores da UFF que contribuíram de forma direta para o meu crescimento acadêmico e aprendizado.

Por fim, agradeço ao meu marido Augusto pela paciência e estímulo para seguir adiante.

RESUMO

Rádios definidos por *software*, também chamados de rádios de *software*, constituem uma tecnologia recém-desenvolvida em que *software* de processamento de sinal, executando em uma dada plataforma de processamento, realiza funções dos rádios que tradicionalmente são feitas através de *hardware*. Tais rádios assumem um papel importante no contexto atual das telecomunicações, pois contribuem para uma maior flexibilidade e rapidez no desenvolvimento das comunicações *wireless*, na medida em que estas incorporam novas funcionalidades, serviços e padrões. Este trabalho objetiva apresentar as características e princípios gerais dos rádios de *software*, bem como suas vantagens, aplicações e arquiteturas mais comumente utilizadas. A tecnologia de *hardware* subjacente que possibilita o projeto de tais rádios também é descrita. Há, ainda, a proposição de uma aplicação-exemplo simples, a recepção de sinais AM, incluindo dois diferentes tipos de demodulação, a fim de ilustrar como estas podem ser efetuadas através de um rádio de *software*, simplesmente programando de maneira apropriada um processador digital de sinais. Além disso, o trabalho descreve as simulações computacionais que foram realizadas para a comprovação da eficácia desta aplicação-exemplo. Os resultados das simulações são analisados levando-se em consideração tópicos importantes da teoria de processamento digital de sinais.

Palavras-chave: rádios de *software*, rádios definidos por *software*, comunicações *wireless*, processamento digital de sinais.

ABSTRACT

Software Defined Radios, also named Software Radios, are a new technology where signal-processing software, running over a certain platform, perform functions that are done commonly by hardware. These radios play an important role in the current telecommunications context because they help to provide more flexibility and speed for the wireless communications systems development as they incorporate new functions, services and standards. This paper intends to characterize the software radios and expose their general principles, advantages, applications and more common architectures. The hardware technologies that support software radios projects are also described. Besides, this paper proposes a simple application, the reception of AM signals, including two different types of demodulation, to illustrate how they can be performed by the use of software radio simply by programming a digital signal processor. This paper also describes the computational simulations that were performed to demonstrate the success of the reception process. The simulation results are analysed by taking into account important topics of the digital signal processing theory.

Key words: software radios, software defined radios, wireless communications, digital signal processing.

SUMÁRIO

1 Introdução.....	18
2 Rádios de <i>Software</i>.....	21
2.1 Definições.....	21
2.2 Princípios Gerais e Comparação com os Rádios Tradicionais.....	22
2.3 Aplicações.....	25
2.4 Arquiteturas Mais Comuns.....	27
2.4.1 Digitalização em RF.....	27
2.4.2 Digitalização em FI.....	29
2.4.3 Digitalização em Banda Base.....	31
3 Dispositivos Utilizados em Rádios de <i>Software</i>.....	34
3.1 Os Conversores Analógico-Digital e Digital-Analógico.....	34
3.1.1 A Conversão Analógico-Digital.....	35
3.1.2 A Conversão Digital-Analógica.....	47
3.1.3 Parâmetros dos Conversores.....	48
3.2 Processadores Digitais de Sinais.....	53
3.2.1 Porque usar DSPs nos Rádios de <i>Software</i>	53
3.2.2 Características Arquiteturais dos DSPs.....	60
3.2.3 DSPs Disponíveis no Mercado.....	63
4 Teoria de Processamento Digital de Sinais.....	66

4.1 Os Sinais Discretos no Tempo e o Processo de Amostragem.....	67
4.2 Sinais Discretos no Tempo: Representação no Domínio da Frequência.....	68
4.2.1 A Transformada de Fourier de Tempo Discreto.....	68
4.2.2 A Transformada Discreta de Fourier.....	70
4.2.3 A Transformada Rápida de Fourier.....	71
4.3 Sistemas de Tempo Discreto Lineares Invariantes no Tempo.....	74
4.3.1 Linearidade e Invariância ao Deslocamento no Tempo.....	74
4.3.2 Caracterização de Sistemas LIT no Domínio do Tempo: a Resposta ao Impulso.....	75
4.3.3 Representação de Sistemas LIT no Domínio da Frequência: a Resposta em Frequência.....	77
4.4 Filtros Digitais.....	79
4.4.1 Filtros FIR de Fase Linear.....	80
4.4.2 Projeto de Filtros FIR de Fase Linear.....	84
5 Aplicação-Exemplo: Recepção AM.....	92
5.1 A Transmissão e a Recepção AM Analógicas.....	93
5.1.1 A Modulação e a Demodulação AM-DSB com Portadora.....	94
5.1.2 A Modulação e a Demodulação AM-SSB.....	98
5.2 A Recepção AM Usando o Conceito de Rádio de <i>Software</i>	104
5.2.1 Demodulação AM-DSB com Portadora em um Rádio de <i>Software</i>	105
5.2.2 Demodulação AM-SSB em um Rádio de <i>Software</i>	106
5.3 Simulações Computacionais da Recepção AM Usando o Conceito de Rádio de <i>Software</i>	108
5.3.1 Simulação da Demodulação AM-DSB com Portadora: Resultados e Análise.....	108

5.3.2 Simulação da Demodulação AM-SSB: Resultados e Análise.....	116
6 Conclusões e Trabalhos Futuros.....	125
Referências Bibliográficas.....	127
Apêndice A - Programação no Scilab.....	129
A.1 Simulação da Demodulação AM-DSB com Portadora.....	129
A.2 Simulação da Demodulação AM-SSB-LSB.....	130

LISTA DE FIGURAS

Figura 2.1 - Equipamento de rádio: paradigma baseado em <i>hardware</i> (a) <i>versus</i> paradigma baseado em <i>software</i> (b).....	24
Figura 2.2 - Receptor TRF.....	28
Figura 2.3 - Receptor tradicional <i>versus</i> receptor rádio de <i>software</i> com digitalização em FI.....	29
Figura 2.4 - Receptor super-heterodino de conversão simples (a) e receptor super-heterodino de conversão dupla (b).....	30
Figura 2.5 - Receptor tradicional <i>versus</i> receptor com digitalização em banda base.....	32
Figura 2.6 - Receptor de conversão simples para sinais modulados em frequência e fase.....	32
Figura 2.7 - Evolução no ponto de digitalização dos rádios tradicionais em direção aos rádios de <i>software</i>	33
Figura 3.1 - Conversores de dados, transformando sinais entre os domínios analógico e digital.....	36
Figura 3.2 - Conversão analógico-digital.....	36
Figura 3.3 - O processo de amostragem, convertendo um sinal de tempo contínuo em um sinal de tempo discreto.....	37
Figura 3.4 - O processo de amostragem visto como a multiplicação do sinal a amostrar por um trem de impulsos.....	38
Figura 3.5 - (a) Espectro limitado em frequência do sinal a amostrar; (b) Espectro de um trem de pulsos periódico com período igual ao período de amostragem T_s ; (c) Espectro do sinal amostrado resultante da multiplicação no tempo do sinal do item a pelo sinal do item b destacando-se as Zonas de <i>Nyquist</i> ; (d) Espectro de um outro trem de pulsos periódico com período $T_{s'}$ maior que o do item b ($T_{s'} > T_s$ ou $f_{s'} < f_s$); (e) Espectro do sinal amostrado resultante da multiplicação no tempo do sinal do item a pelo sinal do item d, ilustrando-se o fenômeno de <i>aliasing</i> pela existência da condição $f_{s'} < 2f_m$	40
Figura 3.6 - Diagrama de blocos do conversor A/D.....	41
Figura 3.7 - Processo de amostragem destacando-se o resultado do circuito <i>sample-and-hold</i>	41

Figura 3.8 - Ilustração do mecanismo de quantização.....	42
Figura 3.9 - Diagrama de Blocos do conversor D/A.....	47
Figura 3.10 - Glitches no mapeamento de nível nas transições entre as palavras de código.....	48
Figura 3.11 - Ilustração do efeito causado pelo <i>jitter</i> em dois sinais de frequência distintas.....	51
Figura 3.12 - Degradação da razão sinal-ruído causada pelo <i>jitter</i>	52
Figura 3.13 - Consumo de potência mínima dissipada por um conversor em função da resolução para várias taxas de amostragem.....	52
Figura 3.14 - Limitações dos conversores.....	53
Figura 3.15 - Espaço de Solução Ideal.....	56
Figura 3.16 - Espaço de Solução para os ASICS.....	56
Figura 3.17 - Espaço de Solução para os Microprocessadores Genéricos.....	57
Figura 3.18 - Espaço de Solução para os DSPs.....	57
Figura 3.19 - Espaço de Solução para os FPGAs.....	58
Figura 3.20 - Exemplo de Arquitetura Básica presente nos DSPs.....	61
Figura 3.21 - <i>Market share</i> dos fabricantes de DSPs (ano de 2006, total de U\$ 8.3 bilhões).....	65
Figura 4.1 - Primeira etapa do algoritmo de FFT.....	73
Figura 4.2 - Estrutura completa utilizada para o cálculo da FFT de uma seqüência de 8 pontos.....	73
Figura 4.3 - Estrutura Radix-2 utilizada durante todas as etapas de cálculo da FFT.....	74
Figura 4.4 - Entrada e saída em um sistema de tempo discreto LIT.....	74
Figura 4.5 - Resposta ao impulso de um sistema LIT.....	75
Figura 4.6 - Quatro tipos de filtro ideal: (a) filtro passa-baixas, (b) filtro passa-altas, (c) filtro passa-faixa e (d) filtro corta-faixa.....	81

Figura 4.7 - Resposta em frequência de um filtro passa-baixas ideal com resposta em fase linear na banda passante.....	82
Figura 4.8 - Reposta em magnitude para o filtro de resposta ao impulso dada por (74) para quatro valores distintos de N.....	84
Figura 4.9 - Especificações de um filtro.....	86
Figura 4.10 - Efeitos do janelamento: resposta em frequência de um filtro passa-baixas ideal após ter sua resposta ao impulso truncada por uma janela retangular.....	88
Figura 4.11 - Janelas para projetos de filtros FIR.....	89
Figura 4.12 - Resposta em frequência de um mesmo filtro FIR após passar por diferentes tipos de janelas.....	89
Figura 5.1 - Receptor AM super-heterodino.....	94
Figura 5.2 - Modulação AM -DSB com portadora e a formação da envoltória.....	95
Figura 5.3 - Espectros na modulação AM-DSB com portadora : (a) sinal modulante, (b) portadora e (c) sinal modulado.....	96
Figura 5.4 - Detector retificador para AM.....	97
Figura 5.5 - Geração de um sinal SSB pelo método do deslocamento de fase.....	98
Figura 5.6 – Módulo e fase da resposta em frequência de uma rede defasadora de $\pi/2$	98
Figura 5.7 - (a) Espectro do sinal modulante; (b) Espectro do sinal SSB com banda lateral inferior; (c) Espectro do sinal SSB com banda lateral superior.....	101
Figura 5.8 - Demodulação SSB coerente por multiplicação pela portadora seguida de filtragem.....	102
Figura 5.9 - (a) Espectro do produto do sinal SSB-LSB pela portadora; (b) Espectro do produto do sinal SSB-USB pela portadora.....	102
Figura 5.10 - Método de demodulação de sinais AM-SSB.....	103
Figura 5.11 - Análise espectral da demodulação AM-SSB da Figura 5.10.....	104
Figura 5.12 - Proposta de recepção AM seguindo o conceito de rádio de <i>software</i>	105
Figura 5.13 - Blocos funcionais da demodulação AM-DSB com portadora em <i>software</i>	106

Figura 5.14 - Esquema da demodulação AM-SSB em <i>software</i>	108
Figura 5.15 - (a) Transformada de Fourier de $f(t)$; (b) Transformada de Fourier de $\Phi_{AM}(t)$; (c) Transformada de Fourier de $x[n]$; (d) DTFT de $x[n]$	109
Figura 5.16 - (a) Espectro de $x[n]$ no Scilab; (b) zoom em torno de 455kHz; (c) zoom em torno de 545 kHz.....	110
Figura 5.17 - (a) Espectro do sinal no ponto (B) da Figura 5.13; (b) zoom em torno de 0 Hz; (c) zoom em torno de 90 kHz.....	111
Figura 5.18 - Especificações do filtro FIR Passa-Baixas projetado para a simulação da detecção AM-DSB com portadora.....	112
Figura 5.19 - (a) Função Ganho do filtro FIR Passa-Baixas projetado para a simulação da detecção AM-DSB com portadora; (b) zoom em torno de $\omega_p = 0,005 (2\pi rad)$ (c) zoom em torno de $\omega_s = 0,085 (2\pi rad)$	113
Figura 5.20 - (a) Espectro do sinal no ponto (C) da Figura 5.13, ou seja, após passar pela filtragem; (b) zoom em torno de 2kHz.....	114
Figura 5.21 - Demodulação AM-DSB com portadora: sinal digital demodulado no tempo.....	115
Figura 5.22 - (a) Transformada de Fourier do sinal modulante; (b) Transformada de Fourier de $\Phi_{SSB-LSB}(t)$; (c) Transformada de Fourier de $s[n]$; (d) DTFT de $s[n]$	117
Figura 5.23 - (a) Espectro de $s[n]$ no Scilab; (b) zoom em torno de 453kHz; (c) zoom em torno de 547 kHz.....	118
Figura 5.24 - (a) Espectro em magnitude de $q[n]$ e $i[n]$ no Scilab; (b) zoom em torno de 2kHz; (c) zoom em torno de 92 kHz; (d) zoom em torno de 908 kHz; (e) zoom em torno de 998kHz.....	119
Figura 5.25 - Coeficientes do filtro de Hilbert utilizado na simulação da detecção AM-SSB.....	121
Figura 5.26 - (a) Função Ganho do filtro de Hilbert utilizado na simulação da detecção AM-SSB ; (b) zoom em torno de $\omega = 0$; (c) zoom para ver a partir de qual frequência a função ganho fica condicionada ao intervalo $[-1,+1]$ dB.....	121
Figura 5.27 - Espectro em magnitude de $i_h[n]$ no Scilab.....	122
Figura 5.28 - (a) Demodulação AM-SSB-LSB: Espectro do sinal demodulado digitalmente; (b) zoom em torno de 2 kHz.....	123
Figura 5.29 - Demodulação AM-SSB-LSB: sinal digital demodulado no tempo.....	124

LISTA DE SIGLAS

▪ A/D	Analógico-Digital
▪ ADC	Analog-to-Digital Converter
▪ AGC	Automatic Gain Control
▪ ALU	Arithmetic Logic Unit
▪ AM-DSB	Amplitude Modulation Double Sideband
▪ AM-SSB-SC	Amplitude Modulation Single Sideband Suppressed Carrier
▪ ASIC	Application-Specific Integrated Circuits
▪ BDTI	Berkeley Design Technology Inc.
▪ BIBO	Bounded-Input Bounded-Output
▪ BOPS	Billion Operations Per Second
▪ CLB	Configurable Logic Block
▪ CWTS	China Wireless Telecommunication Standard
▪ D/A	Digital-Analógico
▪ DAC	Digital-to-Analog Converter
▪ DAGENS	Data Address Generators
▪ DC	Direct Current
▪ DFT	Discrete Fourier Transform
▪ DSP	Digital Signal Processor
▪ DTFT	Discrete-Time Fourier Transform
▪ ENOB	Effective Numbers Of Bits
▪ FFT	Fast Fourier Transform

▪ FI	Frequência Intermediária
▪ FIR	Finite Impulse Response
▪ FLOPS	Floating Point Operations per Second
▪ FM	Frequency Modulation
▪ FPGA	Field Programmable Gate Arrays
▪ GHz	Gigahertz
▪ GPP	General Purpose Processor
▪ GPRS	General Packet Radio Service
▪ GPS	Global Position System
▪ IDTFT	Inverse Discrete-Time Fourier Transform
▪ IIR	Infinite Impulse Response
▪ kHz	Kilohertz
▪ LIT	Linear e Invariante no Tempo
▪ LNA	Low Noise Amplifier
▪ LSB	Lower Sideband
▪ MAC	Multiply-Accumulate
▪ MFLOPS	Million Floating Point Operations per Second
▪ MHz	Megahertz
▪ MIPS	Million Instructions per Second
▪ MMACS	Million Multiply-Accumulate Operations per Second
▪ RF	Radiofrequência
▪ SFDR	Spurious Free Dynamic Range
▪ SINAD	Signal-to-Noise-and-Distortion Ratio
▪ SQNR	Signal-to-Quantization-Noise Ratio

- TRF Tuned Radio Frequency
- UMTS Universal Mobile Telecommunication System
- USB Upper Sideband
- WLAN Wireless Local Area Network
- ZN Zona de Nyquist

1

INTRODUÇÃO

O presente trabalho propõe o estudo de vários aspectos relacionados à tecnologia dos rádios definidos por *software*. Como será visto, os rádios definidos por *software* utilizam a concepção de substituição de parte do *hardware* que executa suas funções básicas por *software* de processamento de sinal. Tal substituição confere a esses rádios características de maior flexibilidade, permitindo a operação em vários modos e padrões distintos, bem como facilitando a incorporação de novas funcionalidades e serviços com maior rapidez, através de simples adição/substituição de módulos de *software*.

Desta maneira, o assunto em questão assume uma grande importância no contexto dos sistemas de telecomunicações atuais, que têm passado por uma rápida evolução. É de conhecimento geral a grande expansão das tecnologias móveis e *wireless*, com uma gama crescente de dispositivos conectados a redes de comunicações comportando funções cada vez mais variadas. Dispositivos tais como *palmtops*, celulares e *smart-phones* vêm incorporando novas funcionalidades de tal forma que a antiga diferenciação entre o telefone, a TV e o PC (*Personal Computer*) tem sido dificultada. Além disso, as comunicações móveis de terceira geração (3G) têm fortemente incentivado a pesquisa em torno dos rádios de *software*. A presença de vários padrões e protocolos para as comunicações *wireless* e a grande gama de serviços oferecidos pelos provedores de comunicações móveis têm exigido sistemas que, com facilidade, se adaptem a novos requisitos e sejam reconfiguráveis.

Quanto aos trabalhos já realizados na área dos rádios de *software*, destacam-se aqueles que, ao explorarem os conceitos envolvidos, propõem aplicações práticas e prototipação. A referência [1], por exemplo, propõe um *hardware* de teste para implementar uma arquitetura de rádio de *software* multi-padrão englobando GPRS, UMTS e IEEE 802.11 (WLAN). A

referência [2], por sua vez, demonstra o conceito de rádio de *software* construindo um transmissor e um receptor com propósitos demonstrativos, emulando uma versão modificada do padrão de telefonia móvel CWTS (*China Wireless Telecommunication Standard*). Também merece destaque a referência [3], que descreve a implementação de uma plataforma de rádio de *software*, baseada em DSPs (*Digital Signal Processors* ou Processadores Digitais de Sinais), que suporta, com uma arquitetura de *hardware* comum, três padrões: Bluetooth, ETSI Hiperlan (*High-Performance European Radio LAN*) e IEEE 802.11.

Por outro lado, o presente trabalho objetiva apresentar de maneira detalhada as características e princípios gerais dos rádios de *software*, bem como as vantagens, aplicações e arquiteturas mais comumente utilizadas. Além disso, também é objeto de estudo o *hardware* e a tecnologia subjacentes que possibilitam o desenvolvimento de tais rádios. A dissertação inclui ainda a proposição de uma aplicação-exemplo simples, a recepção AM, incluindo dois diferentes tipos de demodulação, a fim de ilustrar como estas poderiam ser efetuadas através de um rádio de *software*, simplesmente programando de maneira apropriada um DSP. Para a confirmação da eficácia da aplicação-exemplo, foram realizadas simulações computacionais e, para a compreensão e análise dos resultados destas simulações, foi feita uma rápida abordagem de tópicos importantes da teoria de processamento digital de sinais.

No Capítulo 2, na Seção 2.1, são apresentadas as definições mais comuns encontradas na literatura para os rádios de *software*. Já a Seção 2.2 clarifica os princípios gerais dos rádios de *software* através da comparação com a arquitetura dos rádios tradicionais. Ela descreve, por exemplo, que o sinal a ser transmitido (rádio de *software* transmissor) ou recebido (rádio de *software* receptor) é digitalizado em algum estágio e é, então, processado usando-se técnicas de processamento digital de sinais residentes em *software*. Este define as características do rádio e é, intrinsecamente, flexível e reconfigurável. Além disso, a Seção 2.2 ressalta as vantagens dos rádios baseados em *software* sobre os rádios com paradigma baseado em *hardware*. Em seguida, na Seção 2.3, são ilustradas várias aplicações possíveis para os rádios de *software*. Já a Seção 2.4 explica, exemplifica e compara as principais arquiteturas utilizadas nos projetos de rádio de *software*. Estas se baseiam no estágio em que a digitalização ocorre: RF, FI ou banda base.

O objeto do Capítulo 3 é a tecnologia de *hardware* subjacente que dá suporte ao desenvolvimento dos rádios de *software*: os conversores A/D (analógico-digital) e D/A (digital-analógico) e a plataforma de processamento, com ênfase nos DSPs. A Seção 3.1 aborda os conversores A/D e D/A, com ênfase nos processos de conversão e nos principais

parâmetros que devem ser levados em consideração na escolha dos conversores a utilizar em um rádio de *software*. A Seção 3.2 aborda o *hardware* de processamento que pode ser utilizado em rádios de *software*, comparando variados tipos e motivando o uso dos DSPs. Além disso, ela descreve as principais características arquiteturais presentes nos DSPs e exemplifica alguns DSPs disponíveis no mercado, bem como os critérios utilizados na seleção deles.

O foco do Capítulo 4 é a abordagem de alguns tópicos da teoria de processamento digital de sinais relevantes para a compreensão e análise dos resultados das simulações computacionais descritas no Capítulo 5. As Seções 4.1 e 4.2 abordam os sinais de tempo discreto incluindo a representação destes sinais no domínio da frequência através de DTFT (*Discrete-Time Fourier Transform*), DFT (*Discrete Fourier Transform*) e FFT (*Fast Fourier Transform*). Já a Seção 4.3 dá ênfase aos sistemas de tempo discreto LIT (Lineares e Invariantes no Tempo) caracterizando-os nos domínios do tempo e da frequência através da resposta ao impulso e da resposta em frequência, respectivamente. Em seguida, a Seção 4.4 aborda parte da teoria de filtros digitais, concentrando-se nos conceitos básicos sobre os filtros FIR de fase linear e no projeto de tais filtros através de uma técnica conhecida como janelamento.

O Capítulo 5 propõe uma aplicação, exemplificando a recepção AM, usando o método de rádio de *software* para dois tipos distintos de demodulação: AM-DSB (*Double Sideband*, ou Banda Lateral Dupla) com portadora e AM-SSB-SC (*Single Sideband Suppressed Carrier*, ou Banda Lateral Única com Portadora Suprimida). A Seção 5.1 faz uma breve revisão de como usualmente são realizadas a transmissão e recepção AM analógicas nos rádios tradicionais, com ênfase na descrição dos processos de modulação e demodulação AM-DSB com portadora e AM-SSB-SC. Já a Seção 5.2 descreve a recepção AM com o uso das demodulações supracitadas, porém inserida no contexto alternativo de um rádio de *software*. Finalmente, na Seção 5.3, são descritas as simulações computacionais realizadas no *software Scilab*, gratuito e *open source*, cujos resultados comprovam a eficácia dos esquemas de recepção e demodulação apresentados na Seção 5.2. Também na Seção 5.3, os resultados das simulações são analisados usando-se a teoria de processamento de sinais.

2 RÁDIOS DE *SOFTWARE*

2.1 Definições

Uma exata definição de rádio de *software* é controversa e não existe consenso sobre o nível de reconfiguração exigida para classificar um rádio como rádio de *software*. Contudo, as seguintes definições têm sido utilizadas:

- a) Conforme citado em [4], os rádios de *software* são uma classe de rádios programáveis em que o mesmo *hardware* pode executar diferentes funções em diferentes momentos.
- b) Em [5], é citado que rádios de *software* são rádios reconfiguráveis nos quais uma grande parte da funcionalidade é implementada através de dispositivos programáveis de processamento de sinal, dando ao rádio a habilidade de mudar seus parâmetros de operação para acomodar novas características e capacidades. Pela substituição de componentes analógicos de rádios tradicionais por processamento de sinal digital, os rádios de *software* permitem uma maior flexibilidade.
- c) Segundo [5], também uma boa definição seria a de que rádios de *software* são rádios substancialmente definidos em *software* e cujo comportamento da camada física pode ser significativamente alterado através de mudanças neste *software*. Um rádio que inclui um DSP (Processador Digital de Sinais) não é

necessariamente classificado como rádio de *software*. Contudo, um rádio que define em *software* sua modulação, correção de erro, processos de criptografia, exibe algum controle sobre o *hardware* de RF e pode ser reprogramado é certamente um rádio de *software*.

- d) Em [1], diz-se que rádio de *software* é um sistema de rádio que tem parâmetros como largura de banda, nível de potência, tipo de modulação, codificação do canal, dentre outros, todos eles configurados por *software* sem a necessidade de se trocar o *hardware*.

Outro conceito que tem sido explorado por alguns autores é o conceito de rádio cognitivo. Trata-se de um rádio de *software* inteligente e adaptativo, que representa um avanço em relação ao conceito de rádio de *software* [6]. O rádio cognitivo é um rádio de *software* inteligente, que é capaz de se auto-adaptar, modificando seu modo de operação para alcançar maior desempenho e eficiência espectral. Ele requer inteligência artificial e poder computacional significativos para processar algoritmos adaptativos em tempo real. O rádio cognitivo deve ser capaz, por exemplo, de se adaptar ao ambiente de propagação modificando a forma de detecção dinamicamente para evitar que a recepção se deteriore. Ele também deve ser capaz de detectar informação de uma grande variedade de fontes, de usar protocolos distintos e de permitir aplicações diversas e requisitos mínimos de desempenho.

2.2 Princípios Gerais e Comparação com os Rádios Tradicionais

Para entender melhor os rádios de *software*, faz-se necessário compará-los com os rádios tradicionais, cuja concepção é basicamente orientada a *hardware*.

A Figura 2.1-a, presente em [7], exibe um modelo simplificado de um receptor tradicional. Para maior simplicidade, os estágios de amplificação do sinal foram omitidos. A antena recebe a RF (radiofrequência), mas a extração da informação diretamente do sinal de RF é complicada, pois existem muitos canais no sinal recebido pela antena. Primeiramente, um filtro de RF sintonizável seleciona o canal desejado, mas a filtragem não consegue eliminar completamente os canais adjacentes, pois geralmente tal filtro não possui fator de qualidade adequado para esta função. Para eliminar os canais adjacentes, o sinal é primeiramente convertido para uma frequência mais baixa, fixa, denominada FI (Frequência Intermediária). Após este ponto, o sinal é filtrado novamente, eliminando-se os componentes adjacentes. Como todos os canais são convertidos para a mesma frequência FI, este segundo

filtro não precisa ser sintonizável e pode ser construído com fator de qualidade adequado e a custos mais baixos, já que a filtragem se dá em frequências mais baixas. Posteriormente, a informação é extraída da FI através de técnicas de demodulação. Este esquema simplificado abrange as funções presentes em dispositivos como receptores tradicionais AM e FM. Transceptores modernos como os encontrados em Estações Rádio-Bases e telefones celulares requerem componentes de *hardware* adicionais para realizarem funções mais complexas como equalização e detecção de erro. A cadeia de transmissão é similar à cadeia de recepção, com a informação seguindo o caminho inverso: o sinal de informação é modulado em uma frequência intermediária e convertido para cima, ocupando a banda de RF a ser transmitida pela antena.

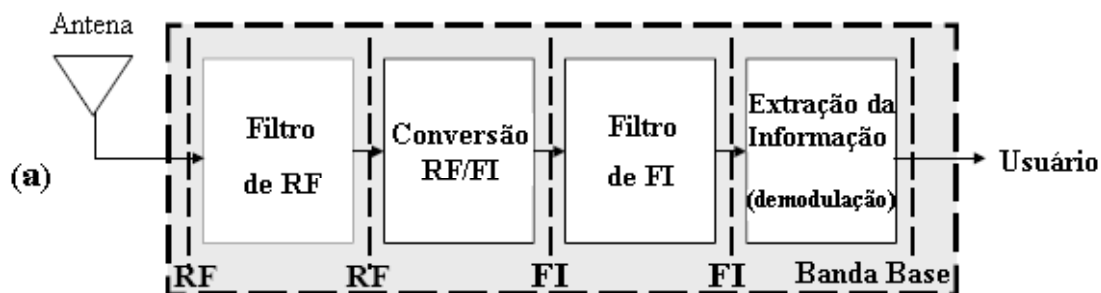
A concepção orientada a *hardware* dos rádios tradicionais impõe uma série de limitações. Primeiramente, rádios tradicionais apresentam baixa flexibilidade para se adaptarem a novos serviços e padrões. Cada elemento de *hardware* do rádio que executa uma determinada função é projetado para operar sobre uma banda de frequência de RF particular, levando-se em conta um certo padrão. Para que tais rádios estejam aptos para operar sob novas condições, eles devem ser reprojetados e seus módulos de *hardware* devem ser substituídos. Reprojetar, fabricação e substituição de componentes de *hardware* envolvem altos custos e longo tempo. Além disso, rádios tradicionais são também limitados no número de serviços que eles podem prover. Por exemplo, quando dois ou mais serviços precisam ser integrados no mesmo dispositivo, deve ser criada uma cadeia de RF para cada serviço.

A Figura 2.1-b ilustra um receptor usando a concepção de rádio de *software* ideal. Um rádio de *software* ideal é aquele em que toda a detecção do sinal é feita por *software*, substituindo-se praticamente todos os componentes analógicos. Observa-se, nesta figura, que um processador, executando um *software* especial, juntamente com um conversor analógico-digital (A/D) e um conversor digital-analógico (D/A), substituem a cadeia com componentes de *hardware* dos rádios tradicionais. O conversor A/D digitaliza o sinal de RF analógico e técnicas de processamento de sinal (*software* executado no processador) atuam sobre as amostras digitalizadas. Posteriormente, sobre o resultado deste processamento, é realizada a conversão digital-analógica, para que a informação seja apresentada em uma forma adequada para o usuário, por exemplo, áudio e/ou vídeo. Na prática, o esquema para rádios de *software* é um pouco mais complexo devido às limitações da tecnologia atual, mas isto será visto na Seção 2.4. O uso de processadores com *software* de processamento de sinal aumenta a flexibilidade do rádio de *software* que pode mais facilmente se adaptar a novos serviços e padrões: novo *software* é instalado sem que o *hardware* necessite ser substituído, desde que o *hardware* seja compatível com as características básicas do novo serviço. O desenvolvimento

do *software* requer menos tempo e é menos custoso em relação ao desenvolvimento de módulos de *hardware*.

Transmissores usando a concepção de rádio de *software* funcionam de forma análoga aos receptores. Um conversor A/D digitaliza a informação (analogica) a ser transmitida e envia as amostras digitais para um processador, que atua sobre tais amostras usando um *software* específico de processamento de sinal. A saída do processador é um sinal digital que representa números, que são enviados para um conversor D/A, gerando assim um sinal analógico a ser transmitido pela antena. Esta dissertação, entretanto, devido a esta simetria entre transmissores e receptores, focalizará, sobretudo, os receptores que usam a concepção de rádio de *software*.

Rádio Tradicional: paradigma baseado em hardware



Rádio de software: paradigma baseado em software



Figura 2.1 - Equipamento de rádio: paradigma baseado em *hardware* (a) versus paradigma baseado em *software* (b) .

Como visto anteriormente, projetos de rádios tradicionais (*hardware*) são realizados de forma específica para cada tipo de aplicação. Um rádio completamente baseado em *hardware* tem utilidade limitada, uma vez que os parâmetros da maioria dos módulos funcionais são fixos. Em rádios de *software*, funções tradicionalmente executadas por circuitos analógicos são realizadas por *software* que reside em processadores. Tais funções podem ser facilmente programadas para executar transmissão e recepção sobre uma larga gama de frequências, usando uma grande variedade de formatos, desde que os processadores e

os conversores tenham, respectivamente, poder computacional e taxas de amostragem suficientes. Portanto, os rádios de *software* apresentam uma série de vantagens em relação aos rádios convencionais. Estas vantagens advêm da substituição de parte dos componentes analógicos da cadeia de RF por módulos de *software* facilmente reconfiguráveis. Dentre as principais vantagens dos rádios de *software* sobre os rádios convencionais, destacam-se [5]:

- a) Operação multimodo. O explosivo crescimento da tecnologia *wireless* levou à proliferação de muitos padrões de transmissão, sendo desejável que um dispositivo possa operar de acordo com vários padrões. Um rádio de *software* pode mudar seu modo de operação simplesmente carregando o *software* apropriado em sua memória. Ele pode usar diferentes modulações, bandas de frequência, filtragem, técnicas de codificação e outros processamentos. Segundo [6], esta grande flexibilidade provê uma enorme oportunidade de resolver problemas de interoperabilidade entre diferentes padrões. Os rádios de *software*, por sua natureza, são independentes de padrões, protocolos e outras especificações, pois podem adaptar-se facilmente às especificações existentes e futuras.
- b) Flexibilidade para incorporar funcionalidade adicional e fazer atualizações. A arquitetura flexível dos rádios de *software* permite adicionar melhorias e funcionalidades sem a necessidade de substituição do *hardware*. Em [1], também se destaca que o rádio de *software* pode permitir que um produto único seja capaz de atender a uma grande gama de consumidores distintos que, através de atualizações de *software*, estariam sempre aptos a utilizar novos serviços proporcionados pelos provedores.
- c) Compactação e eficiência de potência. Isto ocorre porque um só processador de sinal digital pode implementar muitas funções, como demodulação, filtragem e amplificação, substituindo componentes da cadeia analógica, consumindo energia mais eficientemente, sobretudo quando o mesmo *hardware* é reutilizado para implementar múltiplos sistemas e interfaces.

2.3 Aplicações

As aplicações mais evidentes para os rádios de *software* encontram-se em infraestrutura *wireless* e em dispositivos de comunicação no mercado de massa. Em [1], menciona-se que os sistemas de rádios de *software* encontram grande aplicação nas comunicações móveis e *wireless*, pois ampliam a capacidade de reconfiguração de tais sistemas em resposta aos requisitos dos usuários. Através deles, a demanda por novas

aplicações pode ser economicamente suprida por meio de fácil reconfiguração do terminal. Um usuário poderia, por exemplo, escolher entre redes WLAN, GPRS ou UMTS, de acordo com sua preferência, usando o mesmo dispositivo físico. Com a crescente demanda por dispositivos conectados a redes de comunicação, sejam eles *notebooks*, celulares, *palmtops*, dentre outros, uma infinidade de padrões têm surgido e uma integração desses padrões em um terminal único tem se tornado desejável, o que é facilitado pelo uso de rádio de *software*.

Dentro da telefonia móvel, diversas gerações têm surgido (2G, 2.5G e 3G, por exemplo), levando não à substituição de uma geração por outra, mas à coexistência de diversos sistemas [2]. Além disso, embora existam esforços para reduzir o número de padrões de transmissão, existe uma infinidade de padrões que variam de um país para o outro, que usam técnicas de modulação, protocolos e bandas de frequência distintas. O excesso de padrões é ruim tanto para fabricantes como para consumidores. Um único padrão mundial traria benefícios, mas a competição entre os mercados americanos, europeus e asiáticos torna este padrão único difícil de se alcançar. Por este motivo, o rádio de *software* emerge como uma solução pragmática: o aparelho do usuário seria capaz de se adaptar dinamicamente ao ambiente no qual está localizado pela execução do *software* apropriado a cada situação, facilitando assim a questão do *roaming* global. Se um terminal é incompatível com uma dada tecnologia de rede em uma certa região, um módulo de *software* apropriado é instalado no aparelho (possivelmente pelo ar). Com isso, obtém-se o acesso à rede de forma transparente ao se cruzar várias regiões geográficas.

Além de ser menos dependente de padrões, o rádio de *software* traz inúmeras vantagens para usuários, provedores de serviço e fabricantes [6]. Usuários finais adquirem independência quanto ao seu aparelho, pois, com um simples *hardware* escalável e compatível com os sistemas em escala global, não têm necessidade de comprar novos aparelhos conforme novos padrões surjam. Provedores podem oferecer serviços diferenciados sem ter que dar suporte a uma grande gama de tipos de aparelhos. Estações rádio-bases e terminais usando arquiteturas baseadas em *software* podem permitir múltiplas interfaces aéreas, inclusive durante períodos de transição de uma geração de telefonia móvel para outra, e podem facilmente sofrer atualizações via *software*, ampliando, com rapidez, a diversidade de serviços. Além disso, fabricantes de equipamentos podem construir dispositivos mais genéricos, capazes de proporcionar múltiplas funcionalidades pela adição de módulos de *software*, dispondo, por isso, de um tempo de vida útil maior. Ao mesmo tempo, eles conseguem cumprir a exigência atual de curto tempo de desenvolvimento, pois o tempo necessário para se projetar e implementar um equipamento baseado principalmente em *software* é menor que o necessário para o desenvolvimento de um equipamento

predominantemente baseado em *hardware*.

2.4 Arquiteturas mais comuns

As principais diferenças entre os diversos projetos de rádio de *software* derivam, sobretudo, do estágio em que a digitalização ocorre (RF, FI ou banda base) e da plataforma de processamento de sinal digital utilizada. Ambos os fatores afetam também o *software* a ser desenvolvido. A plataforma de processamento de sinal digital pode ser baseada em dispositivos ASICs, FPGAs, DSPs, processadores de propósito geral ou uma combinação destes. A plataforma de processamento será explorada na Seção 3.2 e o estágio da digitalização na presente seção.

Na Seção 2.2, mencionou-se que, em um rádio de *software*, o sinal transmitido ou recebido é digitalizado em algum estágio e, então, processado usando técnicas de processamento de sinal digital residentes em *software* flexível e reconfigurável que define as características do rádio. Nesta seção, abordar-se-á o fato de que esta digitalização pode ocorrer basicamente em três estágios: RF, FI ou banda base. Nos receptores, foco deste trabalho, a digitalização feita em RF, logo após a antena, típica do modelo de rádio de *software* ideal (Figura 2.1-b), compõe o paradigma mais flexível, uma vez que todo o processamento do sinal é feito por *software*. Contudo, este tipo de digitalização é, na atualidade, difícil de ser realizado na prática, devido às limitações dos atuais conversores A/D e processadores, sobretudo quanto à velocidade de ambos e a capacidade computacional dos processadores. Portanto, a digitalização deve ocorrer em outros pontos da cadeia de rádio tradicional (ver Figura 2.1-a).

2.4.1 Digitalização em RF

A Figura 2.2 [5] mostra uma topologia de *front-end* conhecida como receptor TRF (*Tuned Radio Frequency*), a qual pode ser utilizada em projetos de rádio de *software* e que mais se aproxima do modelo de rádio de *software* ideal. Nesta arquitetura, a parte analógica está limitada à seção de RF, onde um filtro passa-faixa seleciona uma grande porção do espectro de frequências que contém os serviços desejados. Após a amplificação, possivelmente por um LNA (*Low Noise Amplifier*), um conversor analógico-digital, também chamado de ADC, digitaliza o sinal com uma dada precisão. O sinal digitalizado é, então, processado digitalmente, executando-se via *software* as funções no nível físico de :

- Centralização do espectro do sinal recebido para a banda do serviço de interesse

naquele momento;

- Filtragem necessária para rejeitar sinais adjacentes indesejáveis;
- Demodulação e decodificação do fluxo de símbolos, bem como adição de informações necessárias a processamentos subsequentes de camadas superiores, se existirem.

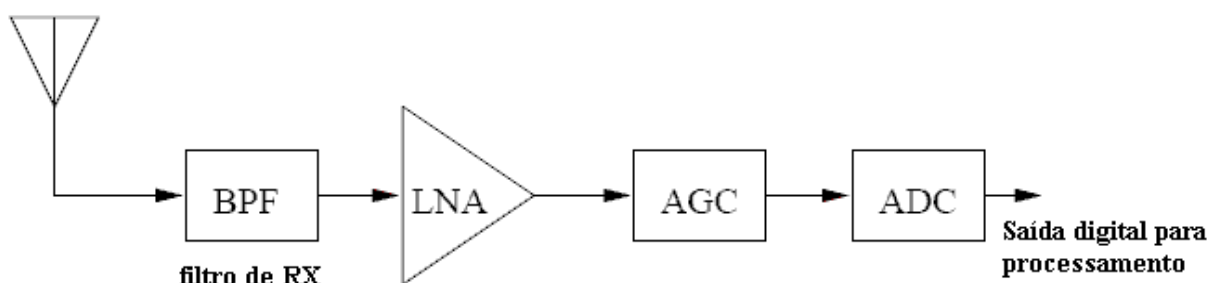


Figura 2.2 – Receptor TRF .

A dificuldade prática em se utilizar o TRF reside na limitação enfrentada pelo conversor analógico-digital, que deve lidar com sinais de alta frequência, devendo possuir alta taxa de amostragem. Ele também deve possuir uma faixa dinâmica grande. Este tipo de conversão é cara, difícil e consome muita energia. A vantagem desta topologia é o número mínimo de partes analógicas, conferindo ao receptor maior flexibilidade.

É necessário ressaltar que a capacidade de permitir distintos padrões, bem como a flexibilidade deste tipo de rádio de *software*, não é ilimitada, sendo restringida sobretudo pelos seguintes fatores:

- Banda de frequências e a largura de banda do estágio de RF;
- Frequência de amostragem do conversor analógico-digital ;
- Capacidade computacional do dispositivo responsável pelo processamento de sinal digital.

É importante lembrar que a digitalização em RF não é completamente explorada pelos sistemas comerciais atuais devido às limitações da tecnologia e considerações de custo. Primeiramente, a seção analógica de RF deve operar numa faixa muito larga, variando de MHz a dezenas de GHz. Além disso, a alta taxa de amostragem exigida pelas altas frequências impõe sobre os conversores severas exigências de operação. Por fim, a amostragem em estágios iniciais da cadeia de RF faz com que processadores mais rápidos e, por isso, mais caros sejam exigidos. Em virtude de tais limitações, a digitalização do sinal no

estágio de FI ainda é a mais utilizada para fins comerciais.

2.4.2 Digitalização em FI

Para superar os problemas atuais da digitalização em RF, os projetistas de rádio de *software* têm posicionado o conversor A/D após o estágio de FI. Esta arquitetura requer um *front-end*, que consiste basicamente de um filtro de RF, um conversor de RF para FI e um filtro de FI (veja Figura 2.3 [7]). O *front-end* seleciona e converte o sinal para FI como fazem os rádios tradicionais. Antes da demodulação, um conversor A/D digitaliza o sinal. Em seguida, o *software* de processamento de sinal, executado em um processador, extrai a informação contida na modulação.

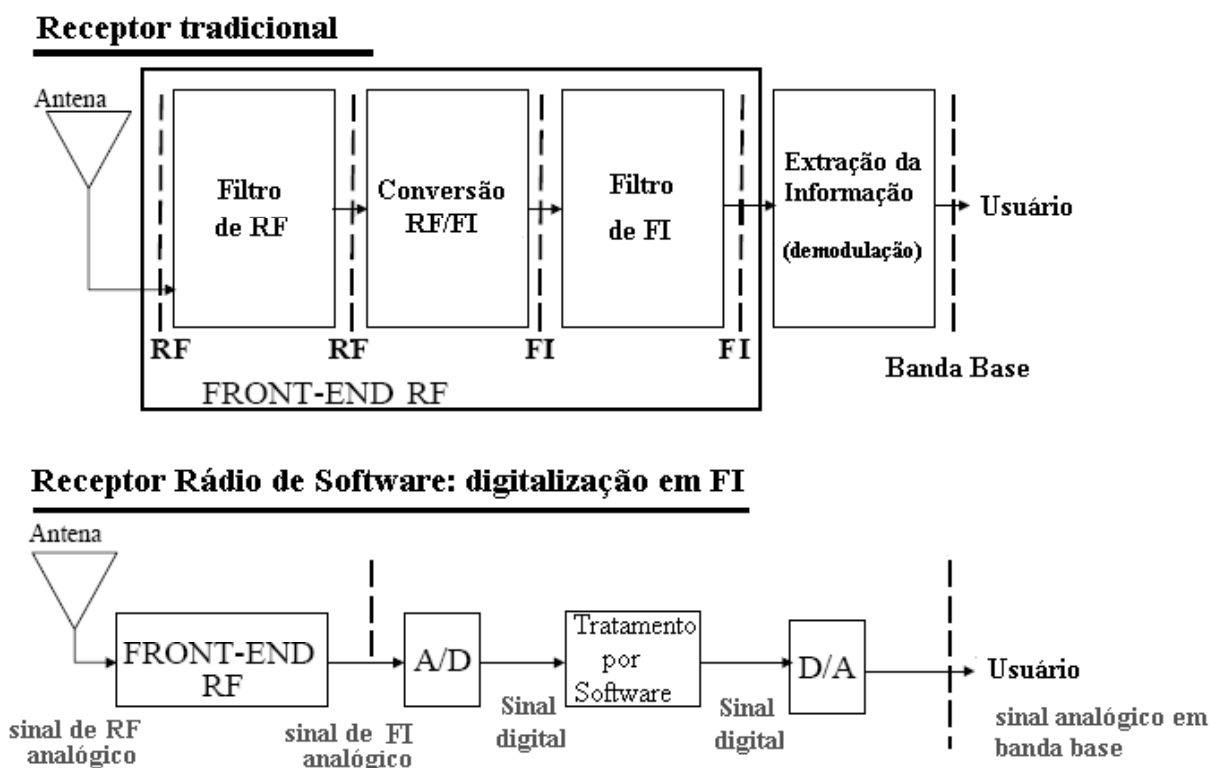


Figura 2.3 – Receptor tradicional versus receptor rádio de *software* com digitalização em FI .

Existem duas principais vantagens na digitalização em FI. Primeiramente, os conversores A/D atuais podem alcançar a velocidade e resolução exigidas na frequência de FI. Segundo, esta arquitetura exige menos recursos computacionais porque o filtro de RF sintonizável do *front-end* limita o número de canais recebidos, reduzindo o trabalho de seleção de canal feito pelo *software* (além de evitar a chamada frequência imagem). A frequência imagem é um componente de frequência indesejável que, após o processo de conversão de RF para FI, passa a ocupar a banda de frequências do canal desejado,

prejudicando a recepção. Isto pode ser evitado pela pré-filtragem realizada pelo filtro de RF.

O receptor super-heterodino mostrado na Figura 2.4-a [5] ilustra bem a digitalização em FI. Ele translada a frequência do sinal entrante para uma frequência intermediária (FI) mais baixa, que é fixa e independe da frequência central do canal desejado. Após a translação para FI, a conversão analógico-digital é feita. Comumente, esta topologia pode envolver dois estágios de conversão para baixo (Figura 2.4-b [5]), permitindo requisitos menos exigentes para os filtros e permitindo que o ganho seja alcançado gradativamente. Porém, esta concepção resulta em um circuito maior e em maior consumo de potência. A escolha pela conversão simples (Figura 2.4-a) ou dupla (Figura 2.4-b) depende de muitos fatores, dentre os quais o espaçamento entre os canais e o plano de frequências. Em geral, quanto menor o espaçamento entre os canais, mais atraente a conversão dupla se torna pela sua maior seletividade.

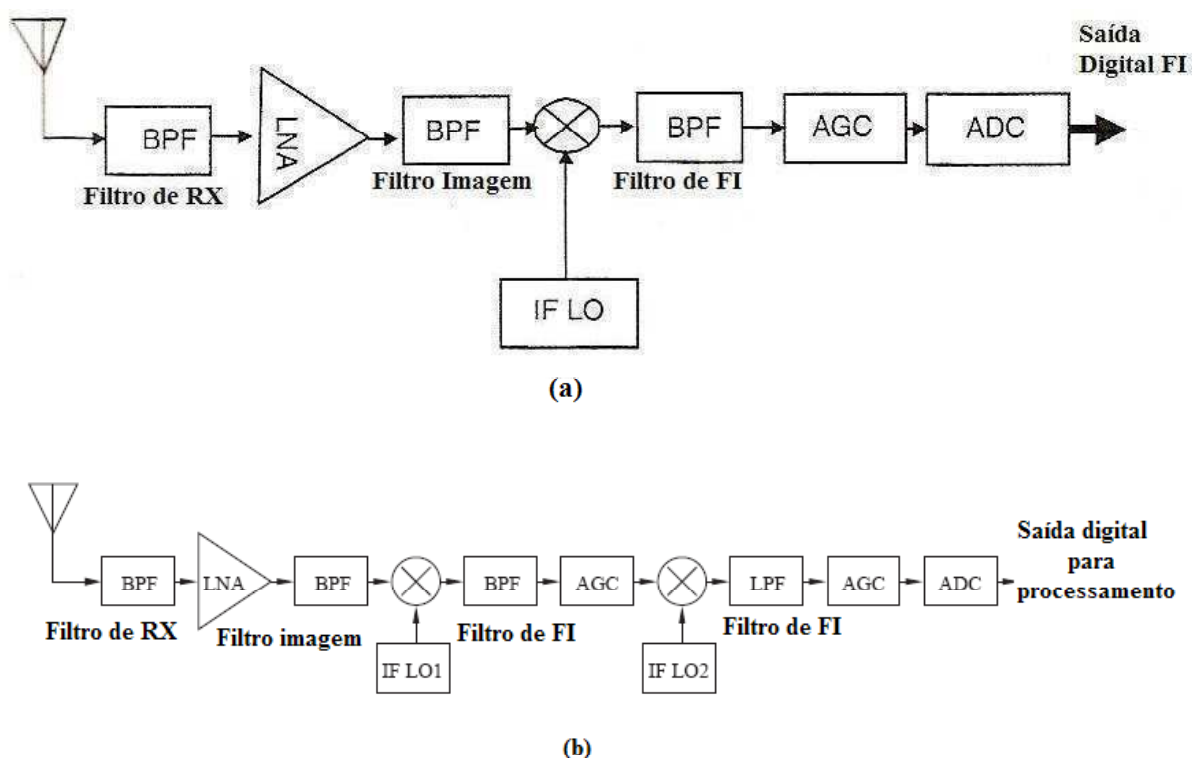


Figura 2.4- Receptor super-heterodino de conversão simples (a) e receptor super-heterodino de conversão dupla (b) .

Dentre os componentes do super-heterodino da Figura 2.4-a, destacam-se um filtro de RF sintonizável responsável pela pré-filtragem do canal selecionado, um LNA, que amplifica o sinal de RF filtrado sem aumentar o ruído, um filtro para retirar a frequência imagem antes da conversão para baixo, um oscilador e um misturador para realizar a conversão para baixo, transformando o sinal de RF em sinal de FI, um filtro centralizado em FI, eliminando-se por

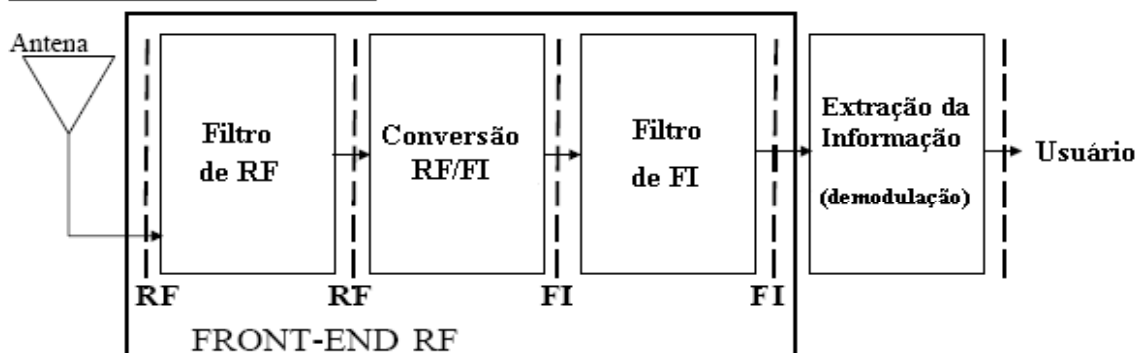
completo qualquer sinal adjacente, e um circuito de controle automático de ganho (AGC, *Automatic Gain Control*) para proporcionar um nível adequado de sinal para a posterior conversão analógico-digital. Já o super-heterodino da Figura 2.4-b mostra, além dos componentes presentes na Figura 2.4-a, um estágio de conversão para baixo adicional com um oscilador, um misturador e um filtro adicionais, porém todos os seus filtros possuem requisitos menos exigentes quanto ao fator de qualidade.

2.4.3 Digitalização em Banda Base

A digitalização em banda base é algo bem usual nos transceptores atuais. Em alguns casos, a informação é analogicamente extraída e a amostragem em banda base é usada nos estágios subseqüentes para prover técnicas de processamento de sinal, tais como equalização. Isto é comum em rádios de carros, por exemplo. Nestes casos, em virtude de nenhuma das funções de extração da informação ser realizada por *software*, estes rádios, mesmo usando digitalização em banda base, geralmente não são considerados rádios de *software*. A Figura 2.5 [7] mostra uma arquitetura com digitalização em banda base que adiciona à tradicional cadeia do rádio tradicional um *hardware* específico, usualmente circuitos integrados dedicados para realizar processamento sobre o sinal recuperado. Isto geralmente aumenta a qualidade da informação recebida, bem como o custo do rádio.

Outro exemplo de digitalização em banda base está ilustrado no receptor de conversão simples (também conhecido como homodino ou de conversão direta), ilustrado na Figura 2.6 [5], que usa um estágio simples para converter o sinal diretamente para banda base, a partir da qual a digitalização e processamento são feitos. A desvantagem deste tipo de arquitetura é a inflexibilidade do receptor, já que a digitalização ocorre muito tarde na cadeia de RF.

Receptor tradicional



Receptor com digitalização em banda base

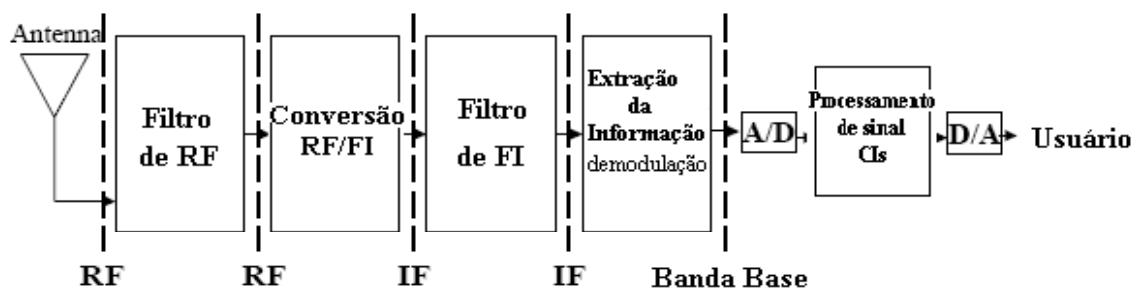


Figura 2.5 – Receptor tradicional versus receptor com digitalização em banda base.

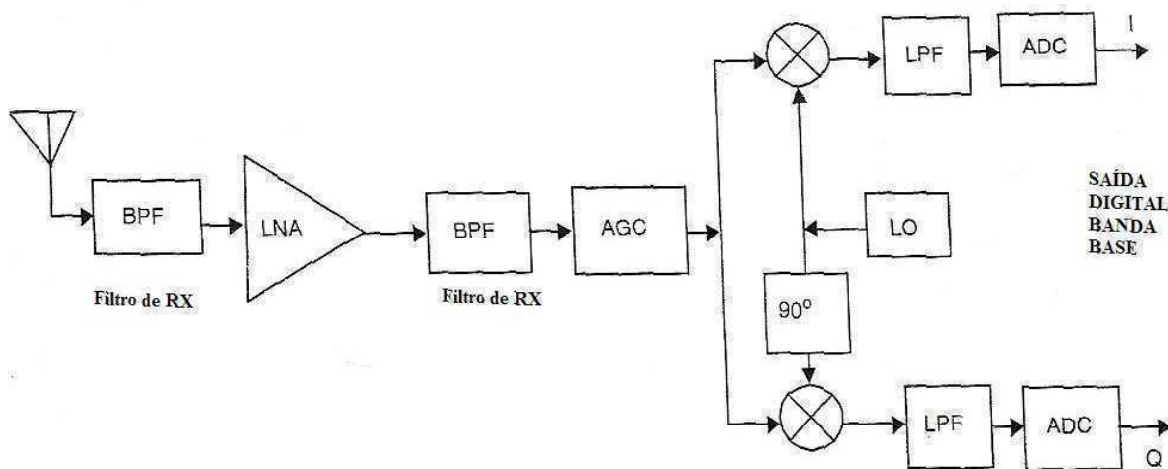


Figura 2.6- Receptor de conversão simples para sinais modulados em frequência e fase.

Comparando-se as diferentes arquiteturas, a digitalização em RF é a mais indicada para rádios de *software* que devem permitir múltiplos modos de interface aérea e múltiplas bandas, pois os requisitos de filtragem para os estágios de FI tornam difícil o suporte a múltiplas bandas no caso do receptor super-heterodino. A resintonização de um receptor pode resultar numa operação complexa, sobretudo com múltiplos componentes numa cadeia de RF. Em muitas situações, o receptor TRF, que usa digitalização em RF, é mais atraente pela maior flexibilidade que ele oferece. Estágios de conversão para baixo e filtragem podem ser

removidos, ao preço de um aumento no custo pelo uso de conversores e processadores mais avançados.

Na medida em que a tecnologia avança, maior parte da cadeia de RF poderá ser substituída por processamento de sinal digital. Isto permitirá maior flexibilidade, aumentando as capacidades do rádio e reduzindo as limitações da eletrônica de RF. A Figura 2.7 [7] mostra os passos da digitalização ao longo do tempo na evolução dos rádios tradicionais em direção aos rádios de *software*.

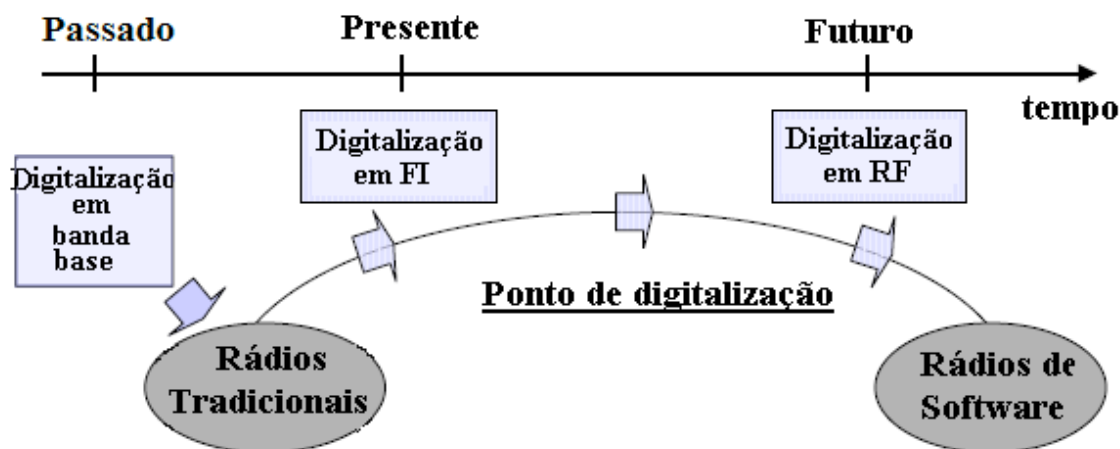


Figura 2.7 – Evolução no ponto de digitalização dos rádios tradicionais em direção aos rádios de *software* .

3

DISPOSITIVOS UTILIZADOS EM RÁDIOS DE *SOFTWARE*

No capítulo anterior, foi mencionado que cada rádio de *software*, seja ele transmissor ou receptor, inclui um processador, um conversor analógico-digital (A/D) e um conversor digital-analógico (D/A) em substituição a dispositivos de *hardware* tais como amplificadores, filtros e moduladores/demoduladores. O sinal analógico a ser processado é amostrado e codificado em um sinal digital binário através do conversor A/D. Então, sobre esse sinal digital binário, um processador realiza operações baseadas em algoritmos específicos de processamento de sinal implementados por módulos de *software*. A saída de tal processamento é um outro sinal digital binário, que é convertido novamente para a forma analógica através de um conversor D/A.

Neste capítulo, estudar-se-ão mais profundamente estes três importantes dispositivos presentes nos rádios de *software*. Na Seção 3.1, abordar-se-ão os conversores A/D e D/A e na Seção 3.2, os DSPs (Processadores Digitais de Sinal), que são tipos de processadores amplamente utilizados nos rádios de *software* e têm uma arquitetura diferente quando comparados aos conhecidos processadores de propósito geral.

3.1 Os Conversores Analógico-Digital e Digital-Analógico

Um conversor A/D transforma um sinal analógico, que é contínuo no tempo e em amplitude, ou seja, um sinal que pode assumir infinitos valores nos eixos do tempo e da

amplitude, em um sinal digital, ou seja, um sinal que só pode assumir valores discretos, representados de forma numérica, com relação aos eixos do tempo e da amplitude. Um conversor D/A, obviamente, opera de maneira inversa a um conversor A/D: transforma um sinal digital em um sinal analógico.

Todo conversor A/D executa um processo básico denominado conversão analógico-digital, que consiste em dois subprocessos básicos consecutivos: a amostragem seguida da quantização. O processo de amostragem transforma um sinal contínuo no tempo em um sinal que assume valores apenas em tempos discretos múltiplos do período de amostragem. A quantização, por fim, transforma este sinal já amostrado (discreto no tempo, mas que pode assumir infinitos valores de amplitude, ou seja, discreto no tempo e contínuo na amplitude) em um sinal discreto também na amplitude. A cada amplitude discreta do sinal quantizado é associada uma palavra de código com um certo número de *bits*. Este número de *bits*, bem como o período de amostragem, variam de conversor para conversor.

Para transformar um sinal digital em um sinal contínuo na amplitude e no tempo, todo conversor D/A efetua o processo chamado conversão digital-analógica, fazendo mapeamento de nível (por exemplo de tensão ou corrente) e reconstrução. O mapeamento de nível associa a cada palavra de código recebida na entrada do conversor D/A um certo nível de amplitude, gerando um sinal contínuo no tempo e discreto na amplitude. Por fim, a reconstrução suaviza as transições entre os níveis de amplitude, gerando um sinal analógico.

As transformações sofridas pelo sinal ao passar da forma analógica para a digital e vice-versa estão ilustradas na Figura 3.1, extraída de [5].

As Seções 3.1.1 e 3.1.2 exploram com maiores detalhes as conversões A/D e D/A e a Seção 3.1.3 aborda os principais parâmetros que devem ser levados em conta na escolha dos conversores A/D e D/A.

3.1.1 A Conversão Analógico-Digital

Conforme citado anteriormente, a conversão A/D pode ser vista como a combinação de dois processos: amostragem e quantização (veja Figura 3.2, extraída de [8]).

Primeiramente, o processo de amostragem será abordado e, posteriormente, o processo de quantização.

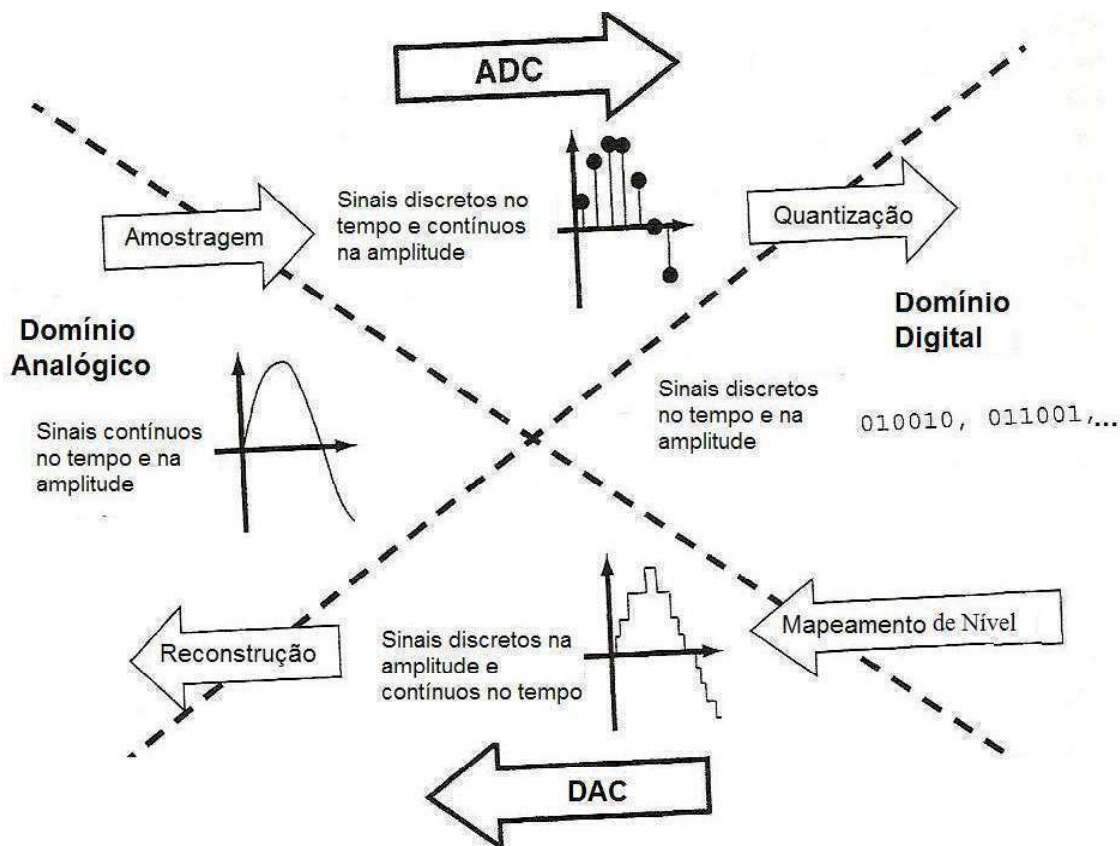


Figura 3.1- Conversores de dados, transformando sinais entre os domínios analógico e digital.

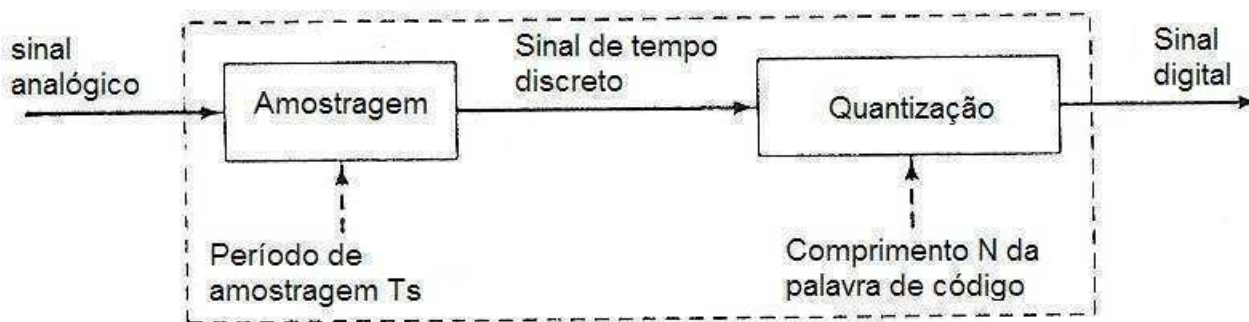


Figura 3.2 – Conversão analógico-digital.

Na amostragem, um sinal tem a sua amplitude medida periodicamente, em intervalos de tempos regulares, a cada período T_s . Isto resulta em uma seqüência de números que expressam a amplitude do sinal em tempos discretos. A Figura 3.3, presente em [5], ilustra a amostragem, que transforma sinais de tempo contínuo em sinais de tempo discreto.

Observe que os valores do sinal entre os períodos de amostragem são perdidos, porém, de acordo com o Teorema da Amostragem, um sinal analógico limitado em frequência com $f \leq f_m$ pode ser reconstruído com precisão a partir de suas amostras se a taxa de amostragem empregada, f_s , dada por $f_s = 1/T_s$, for maior ou igual ao dobro da largura de banda f_m do sinal amostrado, ou seja, se:

$$f_s \geq 2f_m \quad (3.1)$$

A seguir, será feita uma abordagem matemática do processo de amostragem, ilustrando-se o Teorema da Amostragem através de análise espectral. Se esse teorema for violado, ocorre o indesejável fenômeno de *aliasing*, que degrada o sinal e impede sua posterior recuperação.

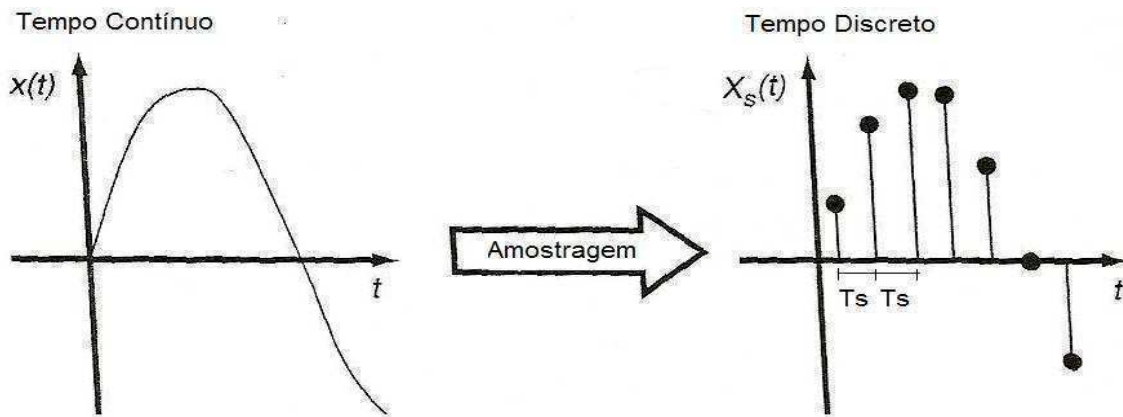


Figura 3.3 – O processo de amostragem, convertendo um sinal de tempo contínuo em um sinal de tempo discreto.

Seja $x(t)$ um sinal contínuo no tempo a ser amostrado e $x_s(t)$ a versão de tempo discreto do mesmo sinal. O sinal $x_s(t)$ pode ser visto como o resultado da multiplicação de um trem de impulsos $p(t)$ de período T_s pelo sinal $x(t)$, conforme ilustrado na Figura 3.4 [9].

Sendo assim, $x_s(t)$ é dado por:

$$x_s(t) = x(t)p(t), \quad (3.2)$$

onde:

$$p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s) \quad (3.3)$$

Então:

$$x_s(t) = \sum_{k=-\infty}^{\infty} x(t) \delta(t - kT_s) \quad (3.4)$$

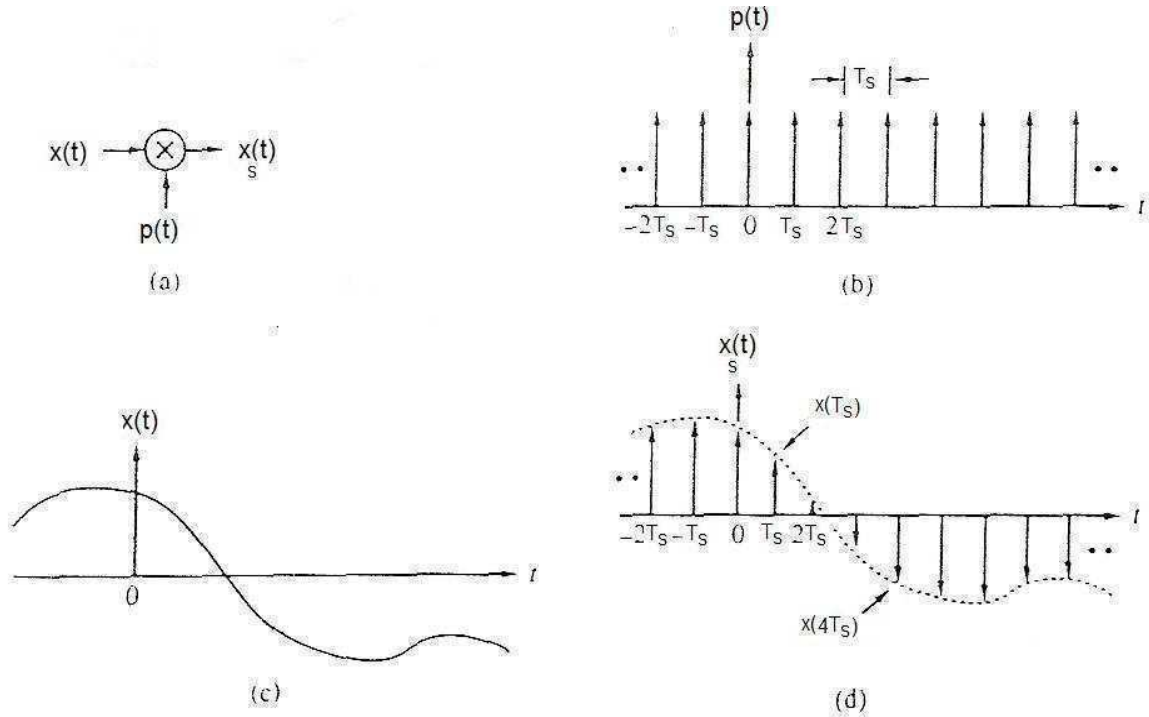


Figura 3.4 – O processo de amostragem visto como a multiplicação do sinal a amostrar por um trem de impulsos.

Para analisar como o processo de amostragem modifica o espectro de $x(t)$, é necessário obter a Transformada de Fourier de $x_s(t)$, $X_s(f)$. Sabe-se que a multiplicação no domínio do tempo equivale à convolução no domínio da frequência. Sendo assim:

$$X_s(f) = X(f) * P(f), \quad (3.5)$$

onde $X(f)$ é a Transformada de Fourier de $x(t)$ e $P(f)$ é a Transformada de Fourier de $p(t)$.

Sabe-se que $P(f)$ é dado por:

$$P(f) = f_s \sum_{k=-\infty}^{\infty} \delta(f - kf_s) \quad (3.6)$$

Substituindo-se (3.6) em (3.5), obtém-se:

$$X_s(f) = f_s \sum_{k=-\infty}^{\infty} X(f - kf_s) \quad (3.7)$$

Por (3.7), conclui-se que o espectro do sinal amostrado consiste de réplicas do espectro do sinal original em torno de múltiplos da frequência de amostragem f_s , ou seja, em torno de $k f_s$, sendo $k = \dots -3, -2, -1, 0, 1, 2, 3, \dots$. Isto pode ser visto na Figura 3.5-c, que ilustra graficamente o processo de amostragem. A Figura 3.5 foi extraída de [9] e sobre ela foram feitas algumas adaptações por questões de nomenclatura. Observe-se que o espectro do sinal amostrado contém o espectro do sinal original (em $k=0$). Note-se também que o processo de amostragem gera no espectro regiões similares espaçadas de $f_s/2$. Estas regiões são conhecidas como Zonas de *Nyquist* (ZNs). O espectro em cada ZN é o espectro espelhado das ZNs vizinhas revelando simetria em torno das frequências $k f_s/2$. Isto ocorre porque o sinal amostrado é um sinal real. As Figuras 3.5-d e 3.5-e mostram o uso de uma f_s menor, ilustrando o que acontece quando o espectro do sinal $x(t)$ se estende além da primeira ZN, ou seja, quando $f_m > f_s/2$ (ou $f_s < 2f_m$), originando o fenômeno de *aliasing*. Quando isto ocorre, o espectro do sinal original será degradado de tal forma que não poderá ser recuperado por filtragem. Por isso, é necessário obedecer ao teorema da amostragem expresso em (3.1).

No caso de um sistema de rádio de *software*, o sinal tipicamente recebido logo após a antena terá uma largura de banda maior que a largura de banda do sinal desejado, possivelmente contendo canais adjacentes e ruído. Por este motivo, em sistemas práticos, para que se possa fixar uma certa taxa de amostragem sem que haja *aliasing*, o sinal deve passar por um filtro *anti-aliasing* que o limite em banda antes de passar pelo conversor A/D. Este filtro deve estar presente na cadeia analógica do rádio.

Nos rádios de *software*, é desejável que o front-end de RF seja o máximo possível independente da frequência. Contudo, os componentes analógicos do front-end de RF de rádios de *software* práticos são dependentes da frequência. Como o filtro *anti-aliasing* posicionado na cadeia analógica deve ser sintonizável, o seu desempenho variará com a frequência central e com a largura de banda, de certa forma limitando a flexibilidade do rádio. O rádio de *software* ideal, entretanto, aposta no uso de conversores de alto desempenho, com alta taxa de amostragem e capazes de lidar com uma enorme largura de banda, fazendo a seleção do canal digitalmente, tornando o rádio menos dependente das limitações dos filtros analógicos.

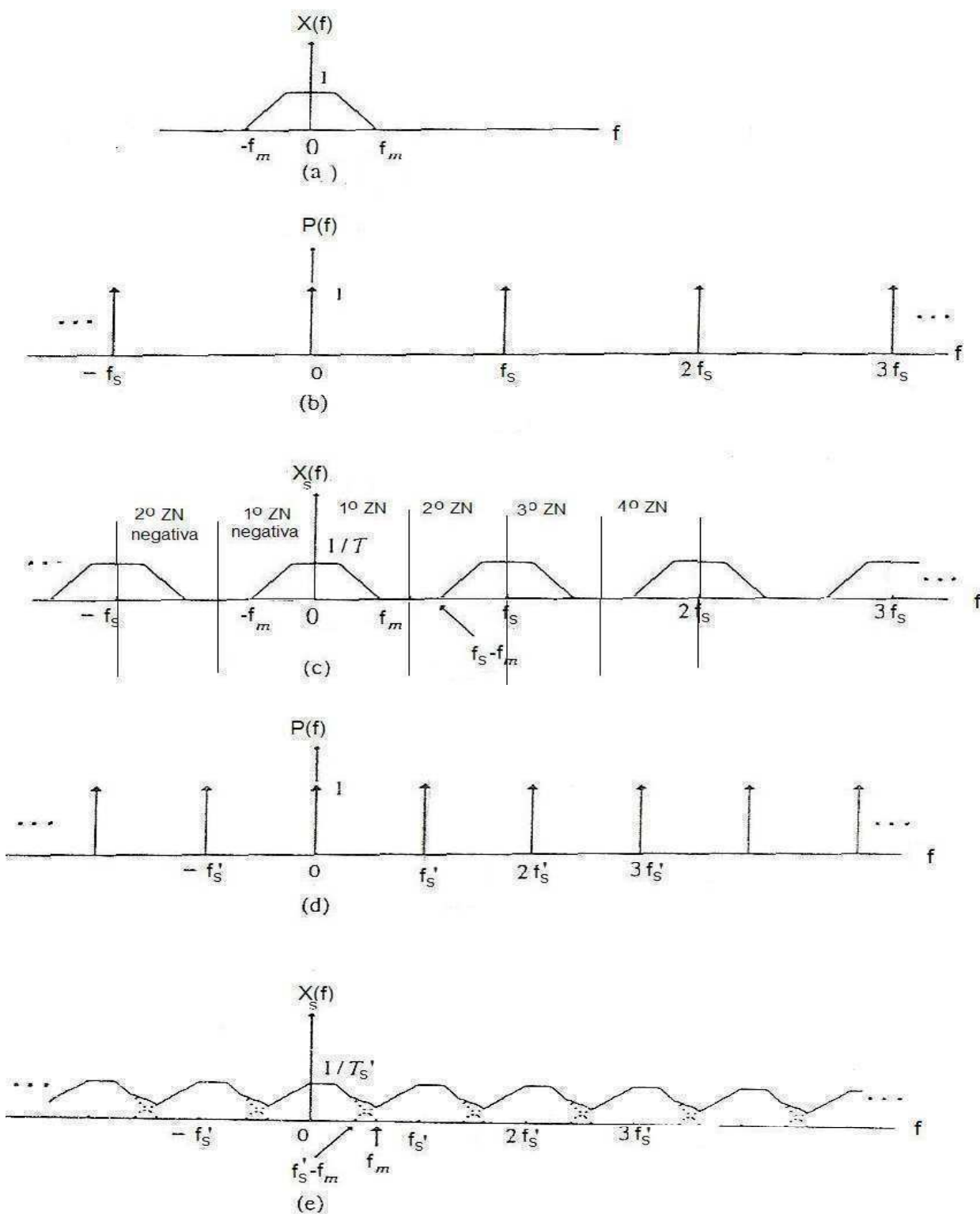


Figura 3.5 – (a) Espectro limitado em frequência do sinal a amostrar; (b) Espectro de um trem de pulsos periódico com período igual ao período de amostragem T_s ; (c) Espectro do sinal amostrado resultante da multiplicação no tempo do sinal do item a pelo sinal do item b, destacando-se as Zonas de Nyquist; (d) Espectro de um outro trem de pulsos periódico com período T_s' maior que o do item b ($T_s' > T_s$ ou $f_s' < f_s$); (e) Espectro do sinal amostrado resultante da multiplicação no tempo do sinal do item a pelo sinal do item d, ilustrando-se o fenômeno de *aliasing* pela existência da condição $f_s' < 2f_m$.

Para aliviar o fator de qualidade do filtro *anti-aliasing*, o sinal pode ser super-amostrado, ou seja, amostrado a uma taxa bem superior a duas vezes a largura de banda do sinal. Em seguida, ele é filtrado digitalmente para eliminar os sinais adjacentes indesejáveis. Além disso, sabe-se que, com a técnica da super-amostragem, aumenta a razão sinal-ruído de quantização, o que, conforme será visto, é vantajoso. Contudo, a super-amostragem requer o uso de conversores A/D de alta velocidade e resulta em taxas de processamento de dados maiores.

A Figura 3.6 [5] ilustra um modelo físico genérico para os conversores A/D. Note-se que ele inclui um filtro *anti-aliasing* e a amostragem é feita por um circuito “*sample-and-hold*” (amostra-e-retém). Tal circuito amostra o sinal de entrada, provendo ao quantizador um sinal que é constante sobre o período de amostragem, conforme ilustra a Figura 3.7.

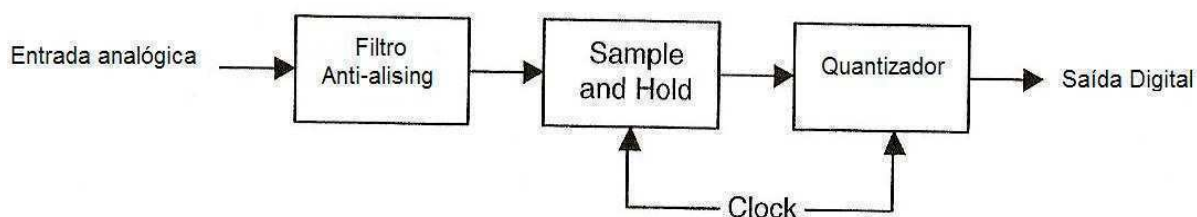


Figura 3.6 – Diagrama de blocos do conversor A/D.

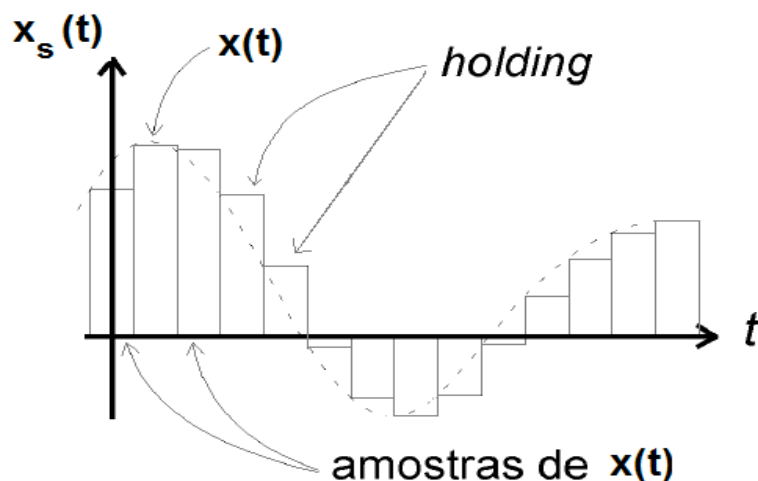


Figura 3.7 – Processo de amostragem, destacando-se o resultado do circuito *sample-and-hold*.

A saída do processo de amostragem é um sinal que pode assumir infinitos valores de amplitude. Para discretizar estes valores de amplitude de forma que um número finito de bits possa ser usado para representar a amplitude de cada amostra, é realizado o processo de quantização. Na quantização, cada amostra é representada pelo nível de quantização correspondente mais próximo a ela. Como existe uma diferença de valores entre a amplitude verdadeira e o nível de quantização utilizado para representar a amostra, o processo de quantização introduz um erro, chamado de erro de quantização. Tal erro pode ser visto como um sinal aleatório que se soma ao sinal original e freqüentemente é modelado como um ruído, dando origem ao termo ruído de quantização. A Figura 3.8 ilustra o processo de quantização. Repare-se que, nessa figura, existem apenas oito níveis de quantização possíveis: de x_0 a x_7 .

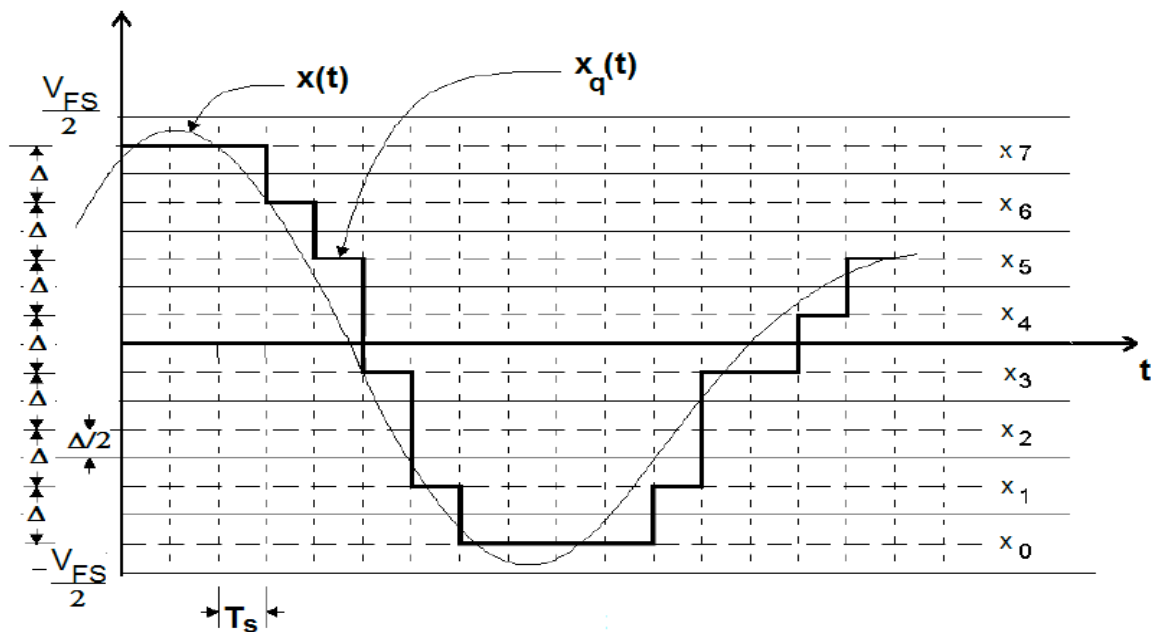


Figura 3.8 – Ilustração do mecanismo de quantização.

Intuitivamente, conclui-se que quanto maior o número de níveis disponíveis para se representar uma amostra, menor será o ruído de quantização pois a aproximação, típica da quantização, poderá ser mais bem feita. Portanto, a razão sinal-ruído de quantização SQNR (*Signal-to-Quantization-Noise Ratio*) depende do número de níveis de quantização. A seguir, isto é comprovado pela dedução da expressão da razão sinal-ruído de quantização, obtida de [1].

Seja N o número de níveis de quantização e B o número de *bits* utilizado para codificar cada amostra. Então:

$$N=2^B \quad (3.8)$$

Supondo que os níveis de quantização sejam uniformemente distribuídos, define-se o *step size* Δ como a distância entre os níveis de quantização (veja Figura 3.8). Os níveis de quantização estão distribuídos no intervalo $[-V_{FS}/2, V_{FS}/2]$, onde V_{FS} é a escala máxima de tensão “pico-a-pico” que o sinal na entrada do conversor pode assumir. Portanto:

$$\Delta = \frac{V_{FS}}{2^B} \quad (3.9)$$

Sendo x o sinal a ser quantizado, o erro de quantização é definido como a diferença entre o sinal quantizado x_q e o sinal original x . Então, o valor instantâneo do erro de quantização é dado por :

$$e(x) = x_q - x \quad (3.10)$$

Além disso, sabe-se que o valor máximo que o erro de quantização pode assumir é igual à metade da distância entre os níveis de quantização, ou seja:

$$\max(e(x)) = \pm \Delta / 2 \quad (3.11)$$

Para analisar os efeitos do ruído de quantização, supõe-se que as amostras do sinal de entrada compõem uma variável aleatória, com média nula, uniformemente distribuída ao longo dos níveis de quantização sobre o intervalo $[-V_{FS}/2, V_{FS}/2]$; supõe-se também que não haja *clipping* do sinal de entrada. Com estas suposições sobre as amostras do sinal de entrada e supondo-se ainda que ele seja espectralmente rico, o ruído de quantização (sinal de erro) pode ser modelado como um processo aleatório de média nula, de autocorrelação nula e que também não tem correlação com o sinal de entrada. Ele segue uma distribuição uniforme que se estende sobre o intervalo $[-\Delta/2, \Delta/2]$ (veja Equação 3.11). Logo, a função distribuição de probabilidade (fdp) do ruído de quantização é dada por:

$$f(e(x)) = \begin{cases} 1/\Delta, & -\Delta/2 \leq x \leq \Delta/2 \\ 0, & \text{caso contrário} \end{cases} \quad (3.12)$$

Sendo o ruído de quantização um processo ergódico, sua média no tempo é livremente intercambiável pela média estatística. Por isso, a potência média do ruído de quantização N_Q

pode ser calculada assim:

$$N_q = E[e^2(x)] \quad (3.13)$$

Sabe-se que, dada uma variável aleatória y , sua variância é igual ao valor esperado de y^2 menos a média ao quadrado de y , ou seja, $\sigma_y^2 = E[y^2] - E[y]^2$. Como no caso do ruído de quantização a média é nula, a Equação (3.13) pode ser reescrita como:

$$N_q = \sigma_e^2, \quad (3.14)$$

onde σ_e^2 é a variância do ruído de quantização.

Uma vez que o erro de quantização é uma variável aleatória uniformemente distribuída, como expressado na Equação (3.12), a Equação (3.14) pode ser reescrita da seguinte forma:

$$N_q = \Delta^2 / 12 \quad (3.15)$$

Substituindo-se (3.9) em (3.15), obtém-se:

$$N_q = \frac{V_{FS}^2}{12(2^{2B})} \quad (3.16)$$

Lembrando-se que a entrada do quantizador é uma variável aleatória uniformemente distribuída sobre o intervalo $[-V_{FS}/2, V_{FS}/2]$ com média nula, sua potência média P_x pode ser calculada de forma análoga à potência do ruído de quantização, resultando em :

$$P_x = E[x^2] = V_{FS}^2 / 12 \quad (3.17)$$

A razão sinal ruído de quantização SQNR é definida por:

$$SQNR = \frac{P_x}{N_q} \quad (3.18)$$

Substituindo-se (3.16) e (3.17) em (3.18), obtém-se:

$$SQNR = \frac{\frac{V_{FS}^2}{12}}{\frac{V_{FS}^2}{12(2^{2B})}} = 2^{2B} \quad (3.19 \text{ a})$$

Em decibéis:

$$SQNR \approx 6.02B \text{ dB} \quad (3.19 \text{ b})$$

Através de (3.19), é possível notar que cada *bit* adicional utilizado na quantização resulta numa melhora na razão sinal ruído de quantização de cerca de 6dB. Para fins ilustrativos, suponhamos que um certo conversor A/D possa aceitar um sinal que varie entre -1 e +1 V e que, após amostrado, origine uma variável aleatória uniformemente distribuída. Então, a SQNR do sinal quantizado para uma resolução de 8 e 16 *bits* será dada por:

Para B=8:

$$SQNR = 10\log_{10}(2^{2 \times 8}) \approx 48.2 \text{ dB}$$

Para B=16:

$$SQNR = 10\log_{10}(2^{2 \times 16}) \approx 96.3 \text{ dB}$$

Como a maioria dos sinais de entrada não serão uniformemente distribuídos ao longo de sua escala de tensão, faz-se necessário obter uma outra expressão da SQNR que leve tal fato em consideração. Seja $P_{x,pk}$ a potência instantânea de pico de um sinal de entrada e $P_{x,avg}$ sua potência média, este sinal pode ser caracterizado por uma razão η entre $P_{x,pk}$ e $P_{x,avg}$:

$$\eta = \frac{P_{x,pk}}{P_{x,avg}} \quad (3.20)$$

Supondo-se que a amplitude de pico do sinal seja $V_{FS}/2$, a potência de pico do sinal será dada por:

$$P_{x,pk} = \left(\frac{V_{FS}}{2} \right)^2 \quad (3.21)$$

Utilizando-se (3.9) em (3.21) e rearrumando:

$$P_{x,pk} = \left(2^B \frac{A}{2} \right)^2 = 2^{2(B-1)} A^2 = \frac{2^{2B} A^2}{4} \quad (3.22)$$

A potência média do sinal pode ser calculada usando-se (3.20) e (3.22):

$$P_{x,avg} = \frac{2^{2B} A^2}{4\eta} \quad (3.23)$$

Usando-se a potência média do sinal e a potência média do ruído, a razão sinal-ruído pode ser calculada, usando-se (3.15) e (3.23):

$$SQRN = \frac{P_{x,avg}}{\sigma_e^2} = \frac{2^{2B} A^2 / 4\eta}{A^2 / 12} = \frac{3(2^{2B})}{\eta} \quad (3.24 \text{ a})$$

Em decibéis:

$$SQRN = 6.02B + 4.77 - 10\log_{10} \eta \quad (3.24 \text{ b})$$

Usando a equação acima, pode-se calcular a SQNR para um sinal senoidal que ocupa a faixa total de tensão de um conversor sem *clipping*. A razão entre a potência de pico e a potência média no caso de uma senóide é dada por:

$$\eta = \frac{A^2}{(A/\sqrt{2})^2} = 2, \quad (3.25)$$

onde **A** é a amplitude da senóide, dada por: $A = V_{FS} / 2$

Substituindo-se (3.25) em (3.24), obtém-se:

$$SQRN_{avg} = 6.02B + 4.77 - 10\log_{10}(2) = 6.02B + 1.76 \text{ dB} \quad (3.26)$$

3.1.2 A Conversão Digital-Analógica

A estrutura de um conversor D/A é conceitualmente similar à estrutura de um conversor A/D na ordem inversa. Um diagrama de blocos funcionais de um conversor D/A é mostrado na Figura 3.9 [5]. Primeiramente, um circuito decodificador mapeia as palavras digitais em níveis discretos de tensão. Posteriormente, pode existir um circuito *sample-and-hold* para minimizar o efeito de *glitching*. Fisicamente, o circuito decodificador tenta simultaneamente ajustar vários comutadores para trocar o nível de tensão de saída que corresponda a palavra digital de entrada. Quando pequenas discrepâncias de tempo ocorrem entre tais comutadores, aparece o efeito de *glitching*, que pode ser melhor visualizado na Figura 3.10 [5]. *Glitches* são níveis de tensão transitórios incorretos que aparecem quando o DAC muda o nível de tensão. Eles são tipicamente atribuídos a *bits* incorretos que temporariamente excitam o decodificador enquanto este executa transições entre palavras de código ou a efeitos capacitivos que limitam o quão abruptamente o sinal de saída pode trocar de nível de tensão. A forma de atuação de um circuito *sample-and-hold* serve para anular o *glitching*: é o circuito de *deglitching*. Após cada transição, depois de um tempo suficiente para que a tensão se estabilize, o circuito de *deglitching* amostra a saída do decodificador. Este valor é mantido até que a próxima amostra seja obtida. Portanto, o circuito de *deglitching* pode remover os efeitos indesejáveis dos *glitches* desde que o tempo de amostragem seja bem escolhido. Ainda na Figura 3.9, a conversão D/A termina com um filtro de reconstrução passa-baixas (também chamado filtro de suavização), que pode ou não ser implementado no *chip* do DAC. Tal filtro é utilizado para transformar o sinal vindo do circuito de *deglitching* em um sinal contínuo na amplitude e para limitar a largura de banda do sinal analógico de saída.

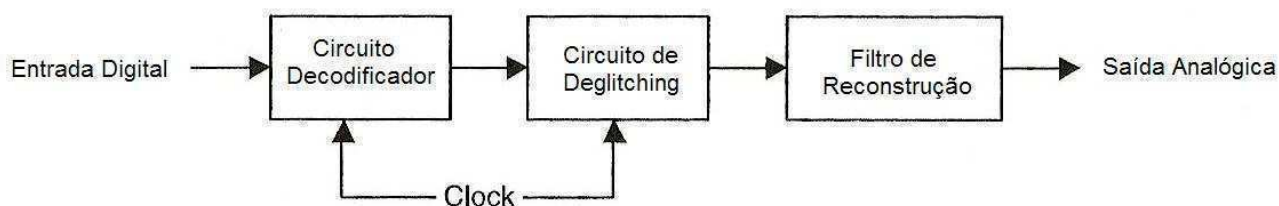


Figura 3.9 – Diagrama de Blocos do conversor D/A.

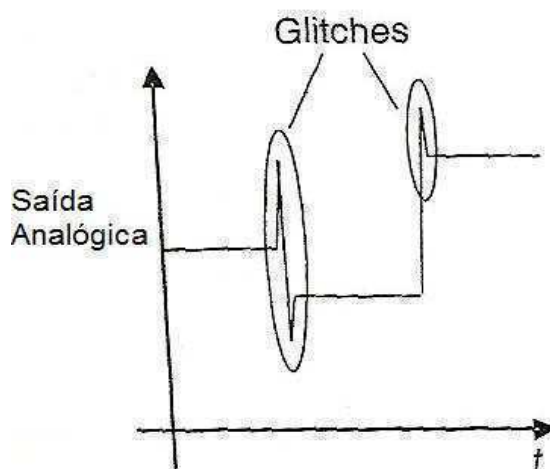


Figura 3.10 – *Glitches* no mapeamento de nível nas transições entre as palavras de código.

3.1.3 Parâmetros dos Conversores

Analizando-se o Capítulo 2, observa-se que as limitações do conversor A/D provavelmente muito influenciarão o projeto da arquitetura do rádio de *software*. A escolha dos conversores adequados deve levar em conta seus principais parâmetros e limitações reais, atendendo às necessidades do projeto sem perder de vista questões de custo e fazendo as devidas considerações. A escolha adequada dos conversores assume uma grande importância nos projetos de rádio de *software* pois afetam características importantes do rádio, incluindo sua flexibilidade, faixa dinâmica, largura de banda e consumo de potência.

Nas seções anteriores deste capítulo, através do estudo mais aprofundado dos processos de conversão A/D e D/A, já se pôde identificar dois dos principais parâmetros dos conversores: a taxa de amostragem ou frequência de amostragem (f_s , dada por amostras/segundo ou Hz) e a resolução (número de *bits* por amostra, o parâmetro B da Seção 3.1.1). Além disso, estudou-se a razão sinal ruído de quantização. Nesta seção, serão feitas considerações sobre faixa dinâmica e serão estudados outros importantes parâmetros dos conversores: SFDR (*Spurious Free Dynamic Range*), SINAD (*Signal-to-Noise-and-Distortion Ratio*), ENOB (*Effective Numbers Of Bits*), *jitter* e consumo de potência.

De uma maneira bem genérica, a faixa dinâmica é definida como a razão entre a maior potência de sinal que um dado sistema pode tolerar sem distorção e o menor valor de potência

do sinal que tal sistema consegue detectar. No caso de um conversor D/A, este valor mínimo detectável é influenciado por distorções de qualquer natureza e pelo ruído, incluindo-se o ruído de quantização e o ruído térmico (sempre presente onde existe movimento de elétrons em componentes resistivos). Como os conversores inerentemente realizam processos não-lineares, harmônicos são gerados gerando distorções harmônicas, também conhecidas como sinais espúrios.

O máximo teórico para a faixa dinâmica de um conversor é calculado considerando-se uma entrada senoidal e é dado pela Equação (3.26) [5]:

$$Faixa\ Dinâmica_{ideal} = SQRN = 6.02B + 1.76 \text{ dB} \quad (3.27)$$

Porém sua faixa dinâmica prática deve levar em conta, além do ruído de quantização, o ruído térmico e os sinais espúrios. Além disso, o percentual que o sinal na entrada do conversor realmente utiliza do intervalo total de tensão possível influencia na faixa dinâmica alcançada. Quanto menor este percentual, menor a faixa dinâmica alcançada e menos aproveitada é a resolução do conversor. Surge, então, a importância do circuito AGC de controle automático de ganho, presente na cadeia analógica, para ajustar a potência do sinal de entrada a um nível seguro que permita distinguir o sinal desejado dos sinais indesejados, ao mesmo tempo evitando-se o *clipping* do sinal.

A análise de como os espúrios afetam o desempenho do conversor pode ser simplificada considerando os efeitos do componente espúrio mais forte dentro da primeira Zona de *Nyquist*. A faixa dinâmica disponível na presença de espúrios, desconsiderando-se os efeitos do ruído, é o SFDR (ou faixa dinâmica livre de espúrios). O SFDR pode ser calculado assim:

$$SFDR = 10 \log_{10} \left(\frac{P_0}{\max(P_i)} \right) \text{ dB}, \quad (3.28)$$

onde: P_0 é a potência do sinal de entrada na primeira Zona de *Nyquist* e $\max(P_i)$ é a potência do maior componente espúrio na primeira Zona de *Nyquist*.

Uma vez que ambos, ruído e distorção, afetam a qualidade do sinal, uma outra medida comum para o desempenho de um conversor é o SINAD, ou razão entre o sinal e o ruído mais a distorção. Para os conversores, ele é definido como a razão entre a potência do sinal, P_0 , e a

soma de todas as fontes de ruído, N , somadas com a distorção de todos os harmônicos, P_i , dentro da primeira Zona de *Nyquist*:

$$SINAD = 10\log_{10} \left(\frac{P_0}{N + \sum_{i=1}^{\infty} P_i} \right) \quad (3.29)$$

O SINAD está fortemente correlacionado com a razão sinal ruído do sinal de entrada, mas à medida em que aumenta a frequência do sinal de entrada ou a taxa de amostragem, o termo advindo da distorção se torna dominante.

Conforme visto anteriormente, a faixa dinâmica de um conversor real é menor que a faixa dinâmica de um conversor ideal, que leva em consideração apenas o ruído de quantização conforme a Equação (3.27). Uma outra maneira de expressar a faixa dinâmica de um conversor real é através do número efetivo de *bits*, ou ENOB, que seria a resolução de um conversor ideal para alcançar a mesma faixa dinâmica de um conversor real. O ENOB pode ser calculado através do SINAD, pela seguinte relação, derivada de (3.27):

$$ENOB = (SINAD - 1.76) / 6.02 \quad (3.30)$$

Outro fator que influencia o desempenho dos conversores práticos é o *jitter*. Nos conversores, o *jitter* se expressa como variações aleatórias que o período de amostragem pode sofrer. Ele aparece devido a limitações do *clock* e do circuito de amostragem. Existe uma certa incerteza quanto ao espaçamento instantâneo entre as amostras. Isto pode resultar em erros significativos quanto ao valor amostrado, sobretudo se o sinal amostrado for de alta frequência. A Figura 3.11, retirada de [5], ilustra o efeito do *jitter* na amostragem de dois sinais de frequência diferentes. Observa-se que, para a mesma variação no período de amostragem, o erro no valor amostrado expressado por ΔV é maior para o sinal de maior frequência ($\Delta V_{\text{high}} > \Delta V_{\text{low}}$). Os efeitos do *jitter* podem ser grosseiramente modelados como um ruído aditivo que se soma ao sinal e, na ausência de outras fontes de ruído e distorção, produz no sistema uma razão sinal ruído dada por [5]:

$$SNR = -10\log_{10} \left(2^2 \pi^2 f^2 \overline{\Delta \sigma_a^2} \right), \quad (3.31)$$

onde f é a frequência do sinal de entrada e $\overline{\Delta \sigma_a^2}$ é a variância do *jitter*. Pela Equação (3.31) e

pela Figura 3.12, presente em [5], mostra-se que, quanto maior o *jitter* e maior a frequência, menor a razão sinal ruído, ou seja, maior é a degradação do sinal.

Considerando-se o efeito do *jitter* juntamente com o ruído de quantização, para conversores com baixa resolução, a razão sinal-ruído do sinal de saída é mais fortemente determinada pelo ruído de quantização. Contudo, quando o *jitter* é muito grande, o aumento no número de níveis de quantização não consegue melhorar significativamente a razão sinal-ruído.

Consumo de potência é outro importante fator a ser considerado ao se selecionar um conversor. A Figura 3.13, extraída de [5], mostra a relação entre a mínima potência dissipada por um conversor e a resolução deste para diferentes taxas de amostragem. Percebe-se que, quanto maior a taxa de amostragem e a resolução, maior é a potência consumida, o que é uma importante limitação, já que dispositivos móveis e portáteis constituem a principal aplicação dos rádios de *software*.

Em geral, existe uma escolha fundamental a ser feita entre a faixa dinâmica e a taxa de amostragem dos conversores ditado por sua tecnologia de fabricação. Se, em um dado projeto, for necessária uma altíssima taxa de amostragem, provavelmente a faixa dinâmica (ou número efetivo de *bits*) deverá ter que ser sacrificada, ou vice-versa. A Figura 3.14, presente em [5], exemplifica as limitações de desempenho dos conversores. Observa-se que trabalhar simultaneamente com uma grande largura de banda (que está ligada à taxa de amostragem) e uma grande faixa dinâmica resultará em maior consumo de potência e custo. Além disso, inevitavelmente a escolha será limitada pela tecnologia de fabricação.

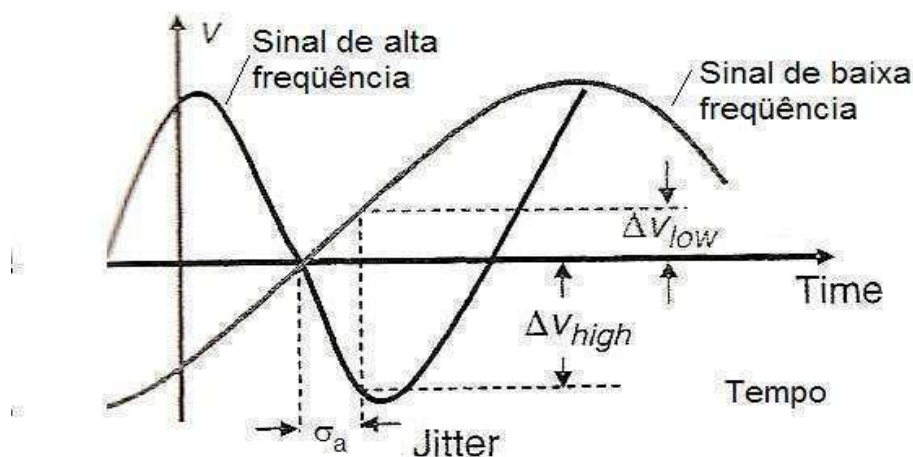


Figura 3.11 – Ilustração do efeito causado pelo *jitter* em dois sinais de frequência distintas.

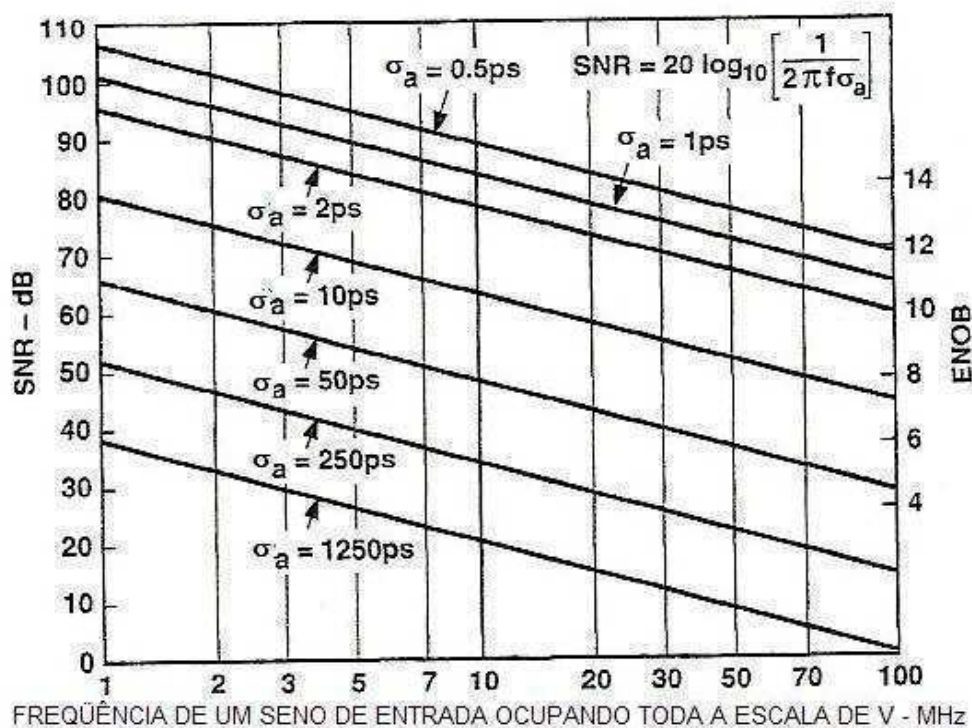


Figura 3.12 – Degradação da razão sinal-ruído causada pelo jitter.

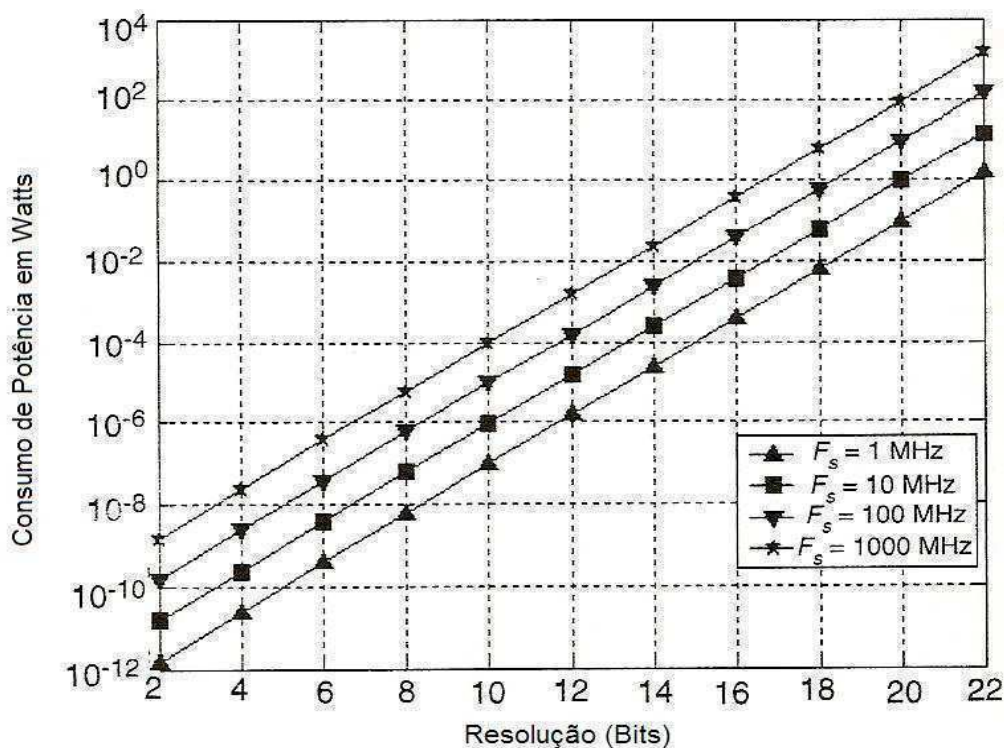


Figura 3.13 – Consumo de potência mínima dissipada por um conversor em função da resolução para várias taxas de amostragem.

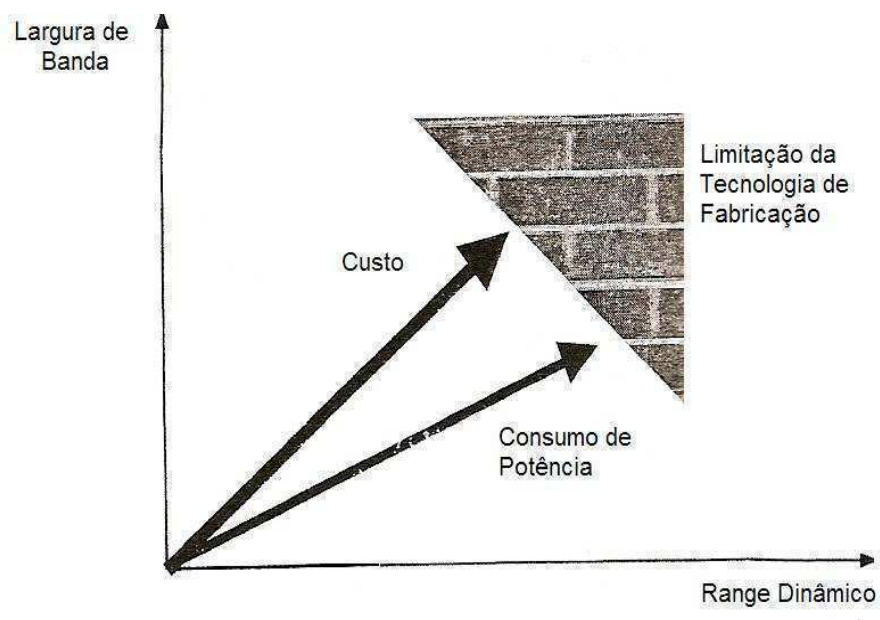


Figura 3.14 – Limitações dos conversores.

3.2 Processadores Digitais de Sinais

No Capítulo 2, mencionou-se que, em um rádio de *software*, um processador executa funções do rádio através de *software* de processamento de sinais, porém sem especificar qual o tipo de processador mais comumente utilizado para esta tarefa. Nesta seção, o foco será o estudo dos DSPs (Processadores Digitais de Sinais) que são amplamente utilizados nos rádios de *software*.

3.2.1 Porque Usar DSPs nos Rádios de *Software*

Genericamente, define-se o processamento digital de sinal como o conjunto de operações ou transformações executadas sobre sinais digitais ou digitalizados usando-se um computador ou outro *hardware* digital. Além de estar presente nos rádios de *software*, o processamento digital de sinal também está presente em muitas outras áreas de aplicação,

podendo-se citar, dentre elas: o processamento de voz (incluindo a compressão de voz, síntese de voz, reconhecimento de fala e interlocutor) e o processamento de imagem e vídeo (incluindo-se a compressão).

Os DSPs são microprocessadores especificamente projetados para realizar de forma otimizada as operações típicas do processamento digital de sinal, as quais serão abordadas mais detalhadamente no Capítulo 4. Por exemplo, os DSPs são otimizados para realizar operações do tipo MAC (*Multiply-Accumulate*, ou seja, multiplicação seguida de soma). Conforme será visto, as operações MAC estão sempre presentes na filtragem digital.

Segundo [10], os dispositivos de processamento tradicionalmente utilizados nos rádios de *software* podem incluir, além dos DSPs, outros *hardwares* de processamento, tais como microprocessadores de propósito geral (GPPs – *General Purpose Processors*) ou FPGAs (*Field Programmable Gate Arrays*). Combinações destes podem ainda ser utilizadas em conjunto com ASICs (*Application-Specific Integrated Circuits*). Na seleção de uma dada solução de *hardware* de processamento, diversos parâmetros devem ser considerados, pois influem diretamente no desempenho do rádio, a saber: poder computacional, densidade computacional, tempo de reconfiguração, eficiência de potência, flexibilidade da aplicação e flexibilidade do *hardware*.

O poder computacional está relacionado à velocidade do *clock* do processador e está inversamente relacionado ao tempo necessário para realizar alguma operação. Para arquiteturas de processadores idênticas, velocidades de *clock* maiores resultam diretamente em maior poder computacional. Contudo, uma vez que diferentes processadores têm arquiteturas distintas e suportam operações diversas, uma simples comparação de *clock*, até mesmo em termos de MIPS (*Million Instructions per Second*) ou FLOPS (*Floating Point Operations per Second*), não é capaz de refletir completamente a verdadeira capacidade computacional de um processador. Isto é um fato tão reconhecido na área de processamento de sinal que certas empresas como a BDTI (*Berkeley Design Technology Inc.*) desenvolveram testes baseados em algoritmos mais comumente implementados, por exemplo, filtragem FIR (*Finite Impulse Response*) e FFT (*Fast Fourier Transform*). Tais algoritmos foram implementados de forma otimizada em um certo número de processadores e, então, foi medido o tempo que estes processadores levaram para completar cada algoritmo, construindo-se uma classificação de desempenho, ou *benchmarking*. O poder computacional é um fator muito importante na construção de um rádio de *software* que atenda a padrões *wireless* 3G e uma grande variedade de padrões e modos de operação.

Outro fator que influi na escolha do *hardware* de processamento é a densidade computacional, definida como o poder computacional por unidade de volume. No domínio dos rádios de *software*, uma condição desejável é um processador com densidade computacional alta e alto poder computacional.

O fator tempo de reconfiguração é a quantidade de tempo necessária para se alterar o comportamento de um dado dispositivo. Rádios de *software* devem ser capazes de mudar completamente seu comportamento dentro de um curto período de tempo: quanto menor o tempo de reconfiguração da solução de *hardware* adotada, melhor.

A eficiência de potência está relacionada com a quantidade de potência consumida por um processador na realização de suas operações. Quanto menor a potência consumida na realização de uma mesma operação, maior é a eficiência de potência. Este fator é especialmente crítico no domínio dos *handhelds*, onde os usuários exigem longos tempos de bateria carregada. Portanto, projetos de rádios de *software* devem procurar maximizar a eficiência de potência.

A flexibilidade de aplicação é o nível em que podem ser realizadas mudanças significativas na aplicação executada pelo processador e o nível de granularidade disponível para tais mudanças. Por exemplo, um processador que facilite a inserção e remoção de módulos de *software* individuais, que controlam os parâmetros de operação do rádio, é mais flexível do que um processador em que a mudança de tais parâmetros possa implicar em alguma mudança no *hardware*. Para um rádio de *software*, um processador com grande flexibilidade de aplicação é sempre necessário.

Por fim, a flexibilidade de *hardware* também pode ser importante. Ela reflete o nível em que mudanças significativas possam ser feitas quanto ao *hardware* computacional e o nível de granularidade disponível para tais mudanças. Ela influencia na habilidade do processador de ajustar seu *hardware* computacional para a aplicação corrente, alternando-se entre funções realizadas em *hardware*.

Para fins de comparação entre GPPs, DSPs, FPGAs e ASICs, um espaço de solução, em que cada um dos seis parâmetros explicados anteriormente representa um diferente “vetor base”, é montado para cada um destes dispositivos. Isto está ilustrado nas Figuras de 3.15 a 3.19, retiradas de [10]. A Figura 3.15 representa um espaço de solução ideal hipotético onde todas as boas características são supridas. Note-se que um dos vetores base utilizados é o inverso do tempo de reconfiguração, pois o tempo de reconfiguração é o único fator cuja diminuição representa uma melhoria. Um aumento em cada vetor base representa um resultado mais desejável.

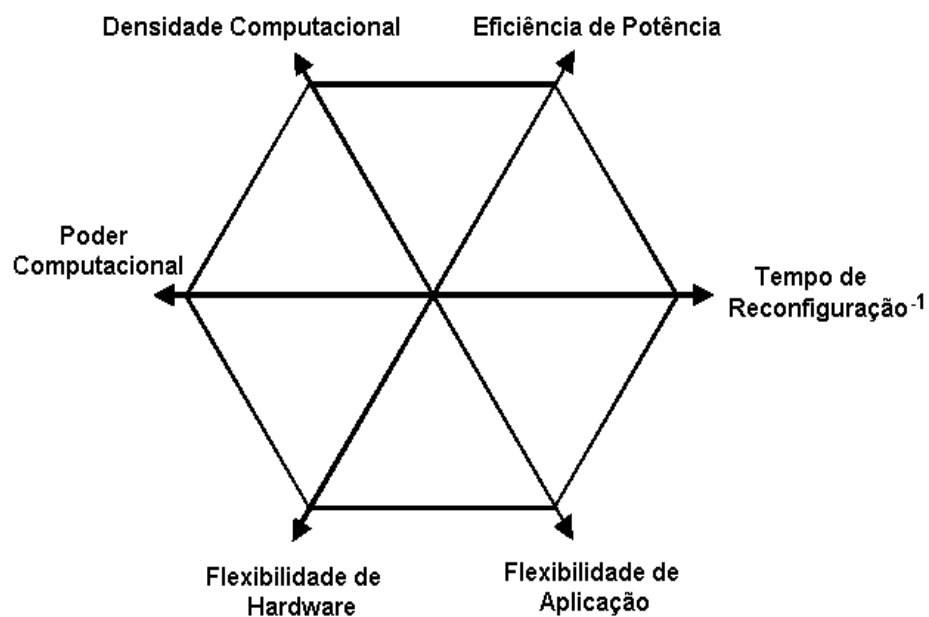


Figura 3.15- Espaço de Solução Ideal.

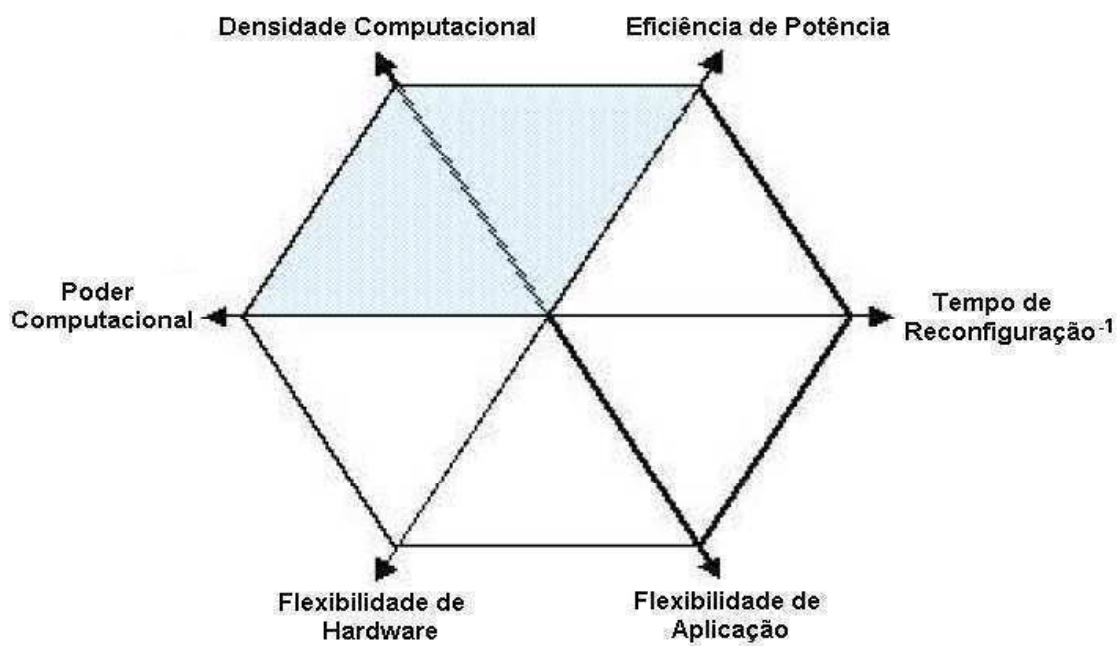


Figura 3.16 – Espaço de Solução para os ASICs.

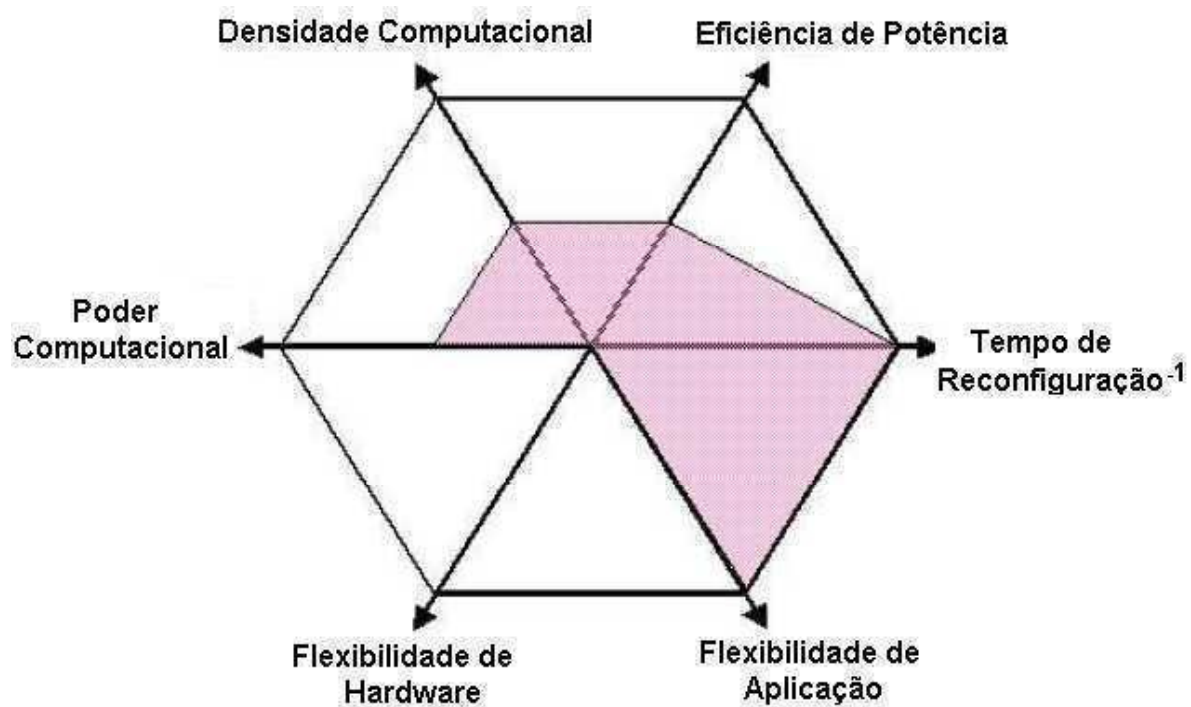


Figura 3.17 – Espaço de Solução para os Microprocessadores Genéricos.

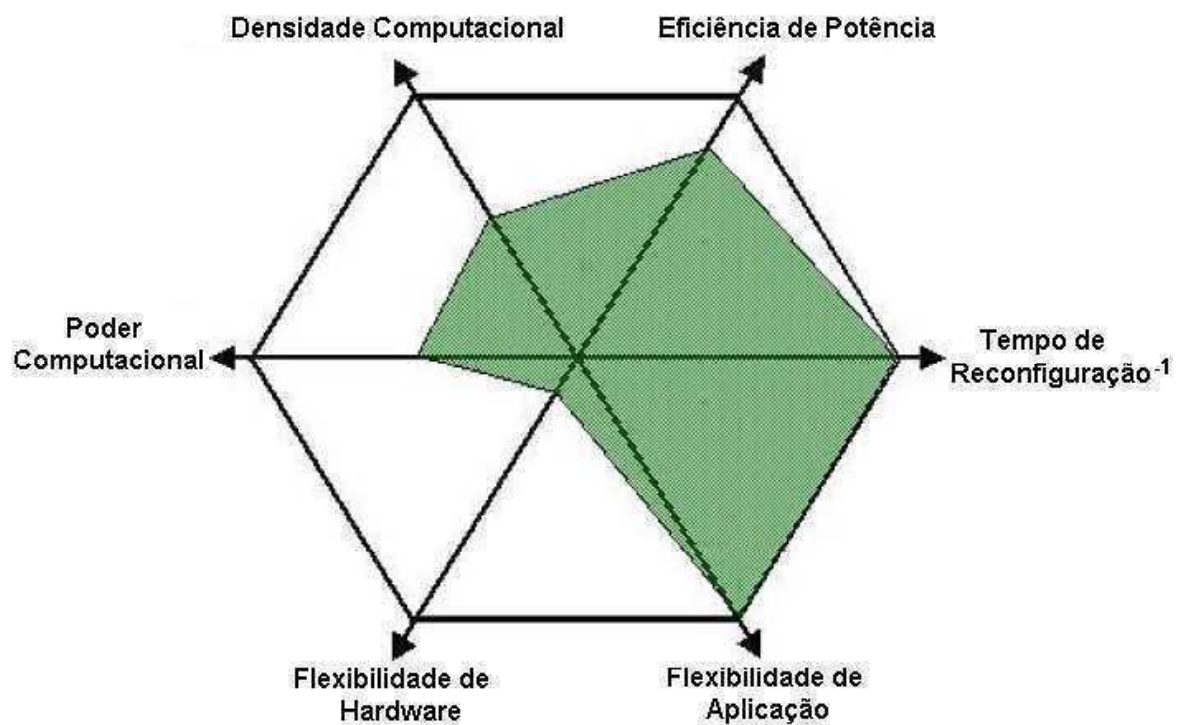


Figura 3.18 – Espaço de Solução para os DSPs.

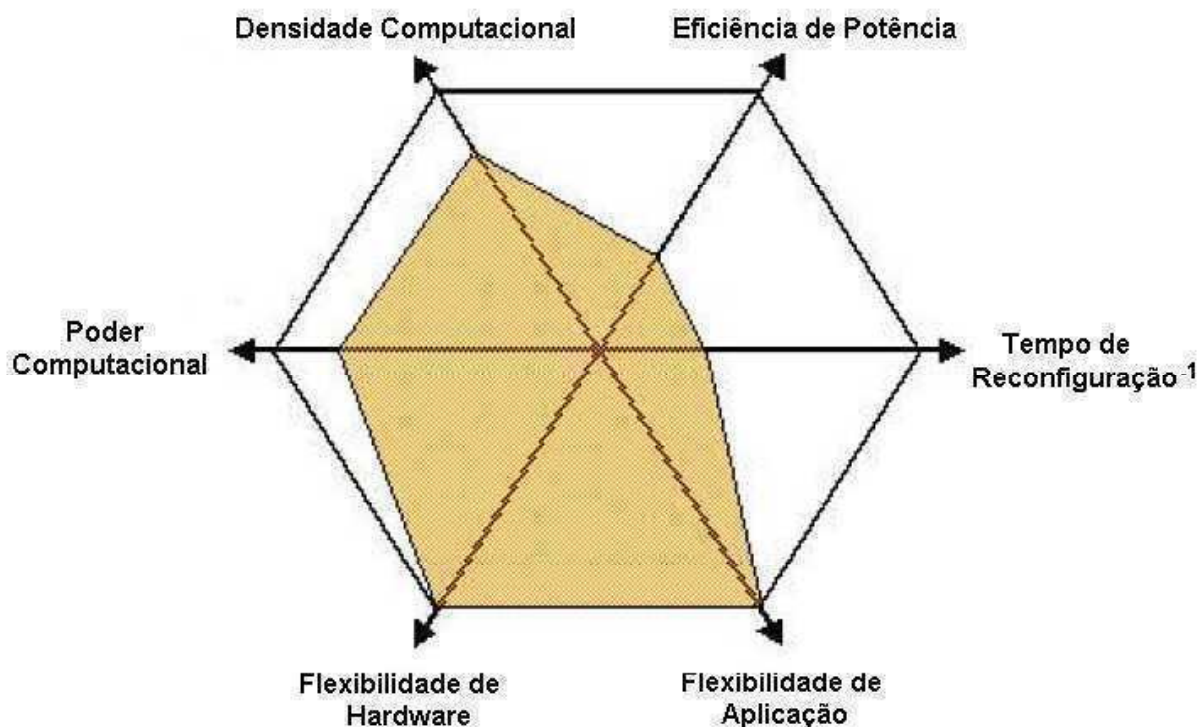


Figura 3.19 – Espaço de Solução para os FPGAs.

Como os ASICs são processadores projetados para implementar uma aplicação particular, é possível otimizá-los quanto aos parâmetros de poder computacional, eficiência de potência e densidade computacional; entretanto, uma vez fabricados, estes não podem ser reconfigurados para outras aplicações. Eles, portanto, são inflexíveis e têm tempo de reconfiguração infinito (veja Figura 3.16). Embora pareça que os ASICs encontrem pouca ou nenhuma aplicação nos rádios de *software*, eles são freqüentemente utilizados em conjunto com outros processadores para implementar algoritmos intensivos computacionalmente que se repetem em diversas aplicações.

Conforme pode ser observado na Figura 3.17, os microprocessadores genéricos assumem um paradigma oposto ao dos ASICs: maximizam a flexibilidade a custo da eficiência. Eles apresentam uma excelente flexibilidade de aplicação e tempo de reconfiguração muito baixo; contudo, seu poder computacional, quando comparado com os ASICs, costuma ser menor. Similarmente, a densidade computacional e a eficiência de potência dos GPPs são menores na comparação com os ASICs. A maioria dos microprocessadores de propósito geral possui baixa flexibilidade de *hardware*.

Os DSPs, conforme mencionado anteriormente, são microprocessadores otimizados para realizar operações de processamento digital de sinal. Maiores detalhes sobre sua

arquitetura serão vistos na próxima seção. Eles possuem densidade computacional e eficiência de potência maiores que as presentes nos GPPs e significativamente menores que as dos ASICs. Como mostrado na Figura 3.18, DSPs e GPPs alcançam níveis similares de poder computacional. Assim como os GPPs, os DSPs possuem baixíssimo tempo de reconfiguração e têm uma excelente flexibilidade de aplicação. Eles também apresentam baixa flexibilidade de *hardware*.

De uma maneira simplificada, os FPGAs podem ser vistos como uma rede de milhões de blocos lógicos configuráveis (os CLBs, *Configurable Logic Blocks*) interconectados entre si de uma maneira reconfigurável controlada por uma matriz de comutação. Cada CLB é projetado para permitir operações aritméticas em nível de bit, mas podem ser escalados para permitir operações que usam diversos bits. Apesar de serem logicamente independentes, os CLBs podem ser programados conjuntamente para realizar operações mais complexas. Um FPGA pode ser configurado para implementar quase todo tipo de aplicação, desde que seja ajustado para tal. Ele é capaz de alcançar capacidade computacional maior que um DSP, contudo ele não consegue alcançar a eficiência de potência e a densidade computacional de um ASIC. A eficiência de potência de um FPGA é frequentemente inferior a de um DSP, porém sua densidade computacional é quase sempre superior. Porém, o tempo de reconfiguração de um FPGA costuma ser maior em comparação com o DSP. O espaço de solução para os FPGAs pode ser visto na Figura 3.19.

Tendo em vista a aplicação dos dispositivos de *hardware* acima mencionados nos rádios de *software*, algumas conclusões e considerações adicionais devem ser feitas, englobando-se fatores além dos já mencionados. O uso de ASIC isoladamente já é descartado dada a baixa flexibilidade de tal dispositivo. O espaço de solução de DSPs e microprocessadores genéricos, embora similares, revelam alguma vantagem dos DSPs sobre os últimos. Embora os FPGAs, de uma maneira geral, apresentem um melhor desempenho em comparação com os DSPs, e também sejam flexíveis como estes, eles consomem potência de uma maneira menos eficiente e ainda são bem mais caros. Atualmente, eles apresentam um preço proibitivo que os impede de estar presentes em aparelhos terminais como celulares. Há de se ressaltar ainda que programar um FPGA é muito mais complexo do que programar um DSP. Além disso, sabe-se que algumas aplicações não conseguem ser adequadamente implementadas em FPGAs. Por todas as razões acima mencionadas, o presente trabalho pretende focar-se no uso de DSPs nos projetos de rádio de *software*.

3.2.2 Características Arquiteturais dos DSPs

Como processadores otimizados para realizar processamento digital de sinal, os diversos tipos de DSPs exibem certas características arquiteturais comuns. Esta seção explorará tais características.

As aplicações de processamento de sinal em geral incluem um grande número de amostras ou blocos de amostras alimentando continuamente o sistema. Sobre tais amostras são realizadas operações repetitivas: a mesma operação é muitas vezes aplicada a diferentes conjuntos de amostras. Em geral, requer-se um processamento paralelo para permitir operações em tempo real, sobre as quais incidem severos limitadores temporais. Além disso, em aplicações de processamento de sinal, a operação mais comum é a operação MAC (*Multiply-Accumulate*, ou seja, multiplicação seguida de soma), que é fundamental para a implementação de filtros digitais. Os DSPs são, portanto, projetados para realizarem repetidas operações MAC como as presentes na filtragem expressa por:

$$y(n) = \sum_{i=0}^{L-1} b_i x(n-i), \quad (3.32)$$

onde: $\{b_0, b_1, \dots, b_{L-1}\}$ são os L coeficientes dos filtros e $\{x(n), x(n-1), \dots, x(n-L+1)\}$ são as L amostras do sinal de entrada.

Com o objetivo de realizar uma operação MAC no menor período de tempo possível, dois dados precisam ser simultaneamente trazidos da memória, multiplicados e o resultado acumulado. Devido às suas características arquiteturais, um DSP é capaz de efetuar no mínimo uma operação MAC por ciclo de relógio. A Figura 3.20, retirada de [8], exibe o diagrama de blocos de uma arquitetura básica presente nos DSPs. Existem algumas diferenças entre os DSPs e os microprocessadores de propósito geral. A arquitetura de *Von Neumann*, típica dos microprocessadores genéricos, utiliza o mesmo espaço de memória para instruções e dados e apresenta um só barramento de acesso à memória, tanto para escrita como para leitura de instruções e dados. Em virtude disto, apenas um dado ou instrução pode ser lido (ou escrito) da memória por ciclo de relógio. Por outro lado, na arquitetura dos DSPs, chamada de arquitetura de *Harvard*, existem, para instruções e dados, espaços independentes de memória e barramentos separados. Por isso, nesta arquitetura, todos os acessos à memória podem ser

executados simultaneamente, aumentando a rapidez de processamento e o paralelismo. Dados e instruções podem ser trazidos simultaneamente da memória, assim como, ao mesmo tempo, resultados de processamentos anteriores podem ser armazenados em memória. Os espaços de memória e os barramentos independentes podem ser vistos na Figura 3.20. A ALU (*Arithmetic Logic Unit*), mostrada nesta figura, realiza operações lógicas e aritméticas. O multiplicador e o acumulador são usados para realizar as operações MAC. O deslocador (ou *shifter*) é usado para escalar os dados. A figura também mostra unidades geradoras de endereço (DAGENS- *Data Address Generators*) que efetuam o cálculo dos endereços dos operandos utilizados nas instruções de maneira paralela às operações efetuadas por outras unidades dentro do DSP. Os DAGENS provêm diversas formas de endereçamento. Com todos estes recursos disponíveis, o DSP proporciona um processamento rápido, fazendo operações dentro de suas unidades de forma concomitante.

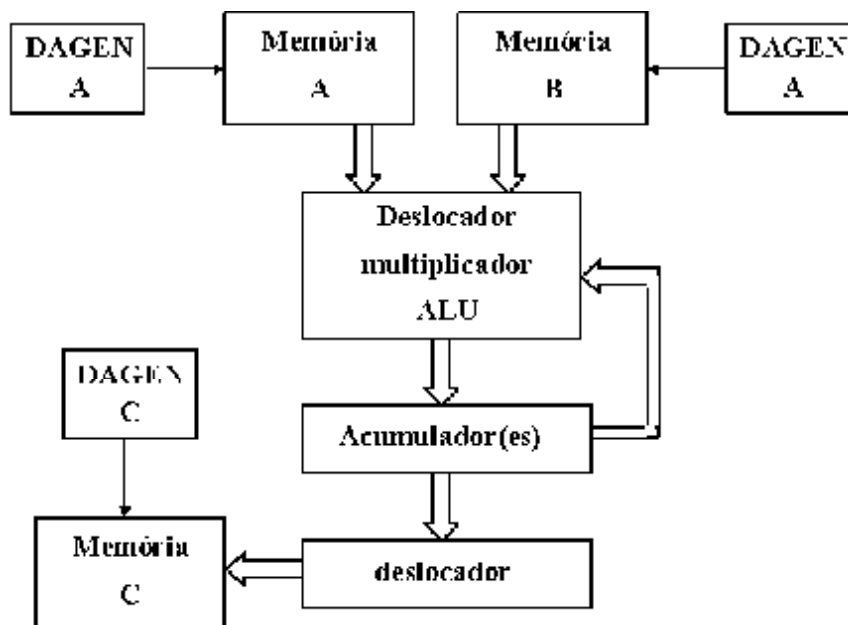


Figura 3.20- Exemplo de Arquitetura Básica presente nos DSPs.

É importante compreender melhor como os blocos funcionais de um DSP facilitam operações de filtragem dadas pela Equação (3.32), aumentando a velocidade de processamento pelo uso de paralelismo. Acessar os operandos, ou seja, os coeficientes do filtro $(b_0, b_1, \dots, b_{L-1})$ e as suas correspondentes amostras de sinal $\{x(n), x(n-1), \dots, x(n-L+1)\}$ é uma operação seqüencial repetitiva onde tais operandos estão em posições consecutivas na memória. Cada DAGEN simplesmente aumenta o ponteiro de endereço para apontar para o próximo item de dado. Isto não requer um ciclo de clock adicional, já que eles fazem isso ao

mesmo tempo em que outros blocos do DSP operam sobre dados anteriores, ou seja, sobre operandos que já foram carregados da memória. A maioria dos DSPs usa dois (um para dados e outro para instruções) ou três barramentos de memória (um para instruções e dois para dados), permitindo o acesso aos coeficientes do filtro e às amostras do sinal de entrada de forma simultânea. Ambos os operandos são carregados da memória simultaneamente. Como os coeficientes do filtro e as amostras do sinal ocupam espaços de memória separados, evita-se o conflito de acessar dois operandos da mesma memória ao mesmo tempo. Por exemplo, supondo-se $i=3$ na Equação (3.32), b_3 e $x(n-3)$ são carregados da memória em um único ciclo de relógio. Neste mesmo ciclo, b_2 e $x(n-2)$ são multiplicados no multiplicador, enquanto a parcela formada pela multiplicação no ciclo anterior, $b_1 x(n-1)$, é somada ao acumulador, que já contém $b_0 x_0$. Os DAGENs vão incrementando i na Equação (3.32) até que o último coeficiente b_{L-1} e a amostra $x(n-L+1)$ sejam obtidos. Depois disto, $y(n)$ é calculado pela acumulação da última parcela, $b_{L-1} x(n-L+1)$, e armazenado na memória. Nos DAGENs, o ponteiro para os coeficientes, de forma circular, aponta novamente para b_0 , preparando-se para a próxima iteração e sequência de amostras de entrada.

Recursos adicionais típicos dos DSPs incluem:

- a) memória cache;
- b) porta(s) serial(ais);
- c) interface para “*host*”;
- d) controlador de DMA;
- e) porta paralela;
- f) temporizador (contador, gerador de sinais controlado por *software*);
- g) gerador de relógio e PLL (*Phased Locked Loop*);
- h) administração de consumo de potência.

3.2.3 DSPs Disponíveis no Mercado

Atualmente, quando se considera os diversos DSPs oferecidos pelos fabricantes, verifica-se que estes têm tentado alcançar áreas de mercado específicas. A Tabela 3.1, retirada de [11], provê um resumo dos DSPs oferecidos pelos principais fabricantes, mostrando dados de desempenho e foco de aplicação.

Tabela 3.1- Exemplos de DSPs disponíveis no mercado.

DSP	Fabricante	Clock (MHz)	Desempenho	Precisão	Otimizado para...
DSP56800	Motorola	80	40 MIPS	Fixo 16 bits	Aplicações de controle
DSP56600	Motorola	60	60 MIPS	Fixo 16 bits	Telefone celular 2G
DSP56367	Motorola	150	150 MIPS	Fixo 24 bits	Processamento de áudio
MSC8102 (StarCore)	Motorola	300	4800MMAC	Fixo 16 bits	Alto desempenho de processamento
ADSP-2191	Analog Devices	160	160 MIPS	Fixo 16 bits	Processamento de áudio
SHARC	Analog Devices	100	600MFLOPS	Float 32/40 bits	Alta precisão
TigerSHARC	Analog Devices	150	1.2 Billion MACS	Fixo e Float 40 bits	Altíssimo desempenho de processamento
TMS320C24x	Texas Instruments	10	20-40 MIPS	Fixo 16 bits	Aplicações de controle
TMS320C54x	Texas Instruments	≤ 133	30-532 MIPS	Fixo ≤ 40 bits	Baixo consumo de potência 0.32mW/MIPS
TMS320C55x	Texas Instruments	200	400 MIPS	Fixo 16 bits	Baixo consumo de potência 0.05mW/MIPS
TMS320C62x	Texas Instruments	150-300	1200-2400 MIPS	Fixo	Poder de processamento de ponto fixo
TMS320C64x	Texas Instruments	400-600	3200-4800MIPS	Fixo	Poder de processamento de ponto fixo
TMS320C67x	Texas Instruments	100-167	600-1000 MFLOPS	Float	Poder de processamento de ponto flutuante
TMS320C8x	Texas Instruments	50	100 MFLOPS equiv. 2 BOPS (RISC)	Fixo 32 bits Float 32 bits	Telecomunicações e Imagem Processamento Paralelo 4 (C80) ou 2 (C82) processadores paralelos 1 processador RISC de 32 bits

Observa-se, na tabela acima, que o desempenho é dado por diferentes unidades:

- a) MIPS (*Million Instructions Per Second*). Reflete o número máximo de instruções executadas por segundo.
- b) MMACS (*Million MAC Operations per Second*). É o número máximo de instruções MAC executadas por segundo
- c) MFLOPS (*Million Floating Point Operations per Second*). Representa o número máximo de operações de ponto flutuante executadas por segundo. É especificado somente para DSPs de ponto-flutuante.
- d) BOPS(*Billion Operations Per Second*). É o número máximo de operações executadas por segundo.

A principal diferença entre operações e instruções depende da complexidade do conjunto de instruções e da capacidade do DSP. Por exemplo, um DSP particular pode ser capaz de executar mais de uma operação por instrução.

Como é difícil comparar o desempenho dos DSPs pelas unidades acima descritas, pois estes suportam instruções diferentes e possuem diferenças arquiteturais entre si, geralmente utiliza-se *benchmarking*, conforme mencionado anteriormente.

Geralmente, os DSPs são otimizados levando-se em conta três principais fatores:

- a) Capacidade de controle – DSPs que contam com a vantagem de possuírem facilidade para adicionar I/O e recursos de memória para uso em aplicações que requerem o controle de interfaces físicas no contexto do usuário.
- b) Consumo de Potência – DSPs para uso em uma grande variedade de dispositivos portáteis alimentados por baterias.
- c) Desempenho – DSPs que possibilitam a priorização do desempenho para diversas faixas de preço de acordo com requisitos específicos de processamento.

Quanto à escolha do DSP a utilizar, além de levar em conta a área de aplicação recomendada pelos fabricantes e os critérios de desempenho e consumo de potência, é interessante a análise da predominância dos fabricantes no mercado. A Figura 3.21, presente em [13], exibe a participação de diversos fabricantes no mercado dos DSPs (*market share*) no ano de 2006. Nota-se a predominância da TI (*Texas Instruments*) com 62% do mercado, seguida da Motorola (*FreeScale*) com 13%.

As principais famílias de DSPs da TI são [12]:

- a) TMS320C2000: ideal para aplicações industriais *embedded* como controle digital de motores e aplicações com sensores inteligentes.
- b) TMS320C5000 : engloba DSPs de ponto-fixa com alta eficiência de potência oferecendo gerenciamento de potência automático e avançado. DSPs

especialmente projetados para dispositivos portáteis pessoais.

- c) TMS320C6000: inclui DSPs de ponto flutuante de desempenho otimizado recomendados para aplicações de áudio de alto desempenho. Também engloba DSPs de alto desempenho otimizados para infra-estrutura *wireless* e de telecomunicações, assim como para aplicações de processamento de imagem. Além disso, conta com DSPs mais sofisticados e mais caros, de altíssimo desempenho, com clock em torno de GHz e recomendados para aplicações de vídeo.

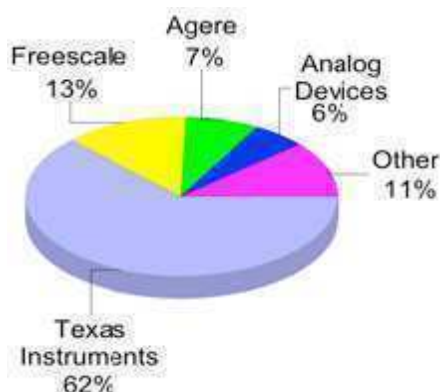


Figura 3.21 – Market share dos fabricantes de DSPs (ano de 2006, total de US\$ 8.3 bilhões).

4 **TEORIA DE PROCESSAMENTO DIGITAL DE SINAIS**

Nos capítulos anteriores, foram abordados o papel do processamento digital de sinais nos rádios de *software* e o *hardware* utilizado para realizar tal processamento. Neste capítulo, serão apresentados conceitos e operações presentes no processamento digital de sinais que servirão de arcabouço teórico para a compreensão e análise das simulações e resultados apresentados no Capítulo 5.

No presente capítulo, serão abordados os sinais discretos no tempo e os sistemas de tempo discreto. Conforme já visto, os sinais discretos no tempo surgem após o processo de amostragem na conversão A/D (antes da quantização) e por isso, neste capítulo, não se levará em conta o número limitado de bits para representar cada amostra. Os sistemas de tempo discreto operam sobre os sinais discretos no tempo para extrair deles determinada informação ou para produzir outros sinais discretos no tempo com características mais desejáveis. Note-se que os sistemas práticos utilizam sinais discretos já quantizados; no entanto, a análise matemática das transformações que podem ser impostas aos sinais não relacionadas com questões de precisão independe da quantização. Ênfase será dada aos sistemas de tempo discreto LIT (Lineares e Invariantes no Tempo), que exibem características específicas que serão apresentadas. Também serão estudados os filtros FIR de fase linear, que são sistemas LIT muito empregados em processamento de sinais.

4.1 Os Sinais Discretos no Tempo e o Processo de Amostragem

Como visto anteriormente, em processamento digital de sinais, os sinais são representados por seqüências de números chamadas de amostras. Em muitos casos, o sinal é originalmente um sinal analógico e um sinal de tempo discreto $x[n]$ é gerado pela amostragem periódica desse sinal analógico $x_a(t)$ usando-se intervalos de tempo uniformemente espaçados:

$$x[n] = x_a(t)_{t=nT_s} = x_a(nT_s), \quad n = \dots, -2, -1, 0, 1, 2, \dots \quad (4.1)$$

onde: T_s é o espaçamento entre duas amostras consecutivas e é chamado de período de amostragem.

A Equação (4.1) expressa a relação entre um sinal analógico e um sinal de tempo discreto gerado pela amostragem sobre o primeiro. A variável tempo t do sinal de tempo contínuo se relaciona com a variável n do sinal de tempo discreto somente nos instantes t_n dados por:

$$t_n = nT_s = \frac{n}{f_s} = \frac{2\pi n}{\Omega_s}, \quad (4.2)$$

onde f_s denota a frequência de amostragem e Ω_s denota a frequência angular de amostragem e é dada por:

$$\Omega_s = 2\pi f_s \quad (4.3)$$

Por exemplo, considere-se que o sinal $x_a(t)$ é dado por:

$$x_a(t) = A \cos(2\pi f_0 t + \Phi) = A \cos(\Omega_0 t + \Phi), \quad (4.4)$$

onde Ω_0 , definida por $\Omega_0 = 2\pi f_0$, é a frequência angular do sinal analógico dada em radianos por segundo.

Então, por (4.1) e (4.2), seu correspondente sinal de tempo discreto $x[n]$ será dado por:

$$x[n] = A \cos(\Omega_0 n T_s + \Phi) = A \cos\left(\frac{2\pi\Omega_0}{\Omega_s} n + \Phi\right) = A \cos(\omega_0 n + \Phi), \quad (4.5)$$

onde ω_0 é a frequência angular digital normalizada do sinal $x[n]$. Ela é dada em radianos e definida por:

$$\omega_0 = \frac{2\pi\Omega_0}{\Omega_s} = \Omega_0 T_s \quad (4.6)$$

Como visto no capítulo anterior, o teorema da amostragem requer que $f_s > 2f_0$, ou seja, que $\Omega_s > 2\Omega_0$. Isto para que se evite o fenômeno de *aliasing* e se possa, através de $x[n]$, univocamente determinar $x_a(t)$. Isto faz com que frequência angular digital sempre se situe no intervalo $[-\pi, \pi]$.

4.2 Sinais Discretos no Tempo: Representação no Domínio da Frequência

Sinais discretos no tempo podem ser expressos em termos de seqüências exponenciais complexas da forma $e^{-j\omega n}$. Estas seqüências discretas permitem expressar os sinais por meio de representações no domínio da frequência, que constituem uma importante ferramenta no processamento de sinal e são comumente utilizadas na prática. Nesta seção, serão abordadas três destas representações: a Transformada de Fourier de Tempo discreto, a Transformada Discreta de Fourier e a Transformada Rápida de Fourier.

4.2.1 A Transformada de Fourier de Tempo Discreto

Na Transformada de Fourier de Tempo Discreto ou DTFT (*Discrete-Time Fourier Transform*), uma seqüência discreta no domínio do tempo é mapeada em uma função contínua na frequência. O sinal de tempo discreto $x[n]$ é representado em termos de seqüências exponenciais complexas $e^{-j\omega n}$, onde ω é a variável real frequência angular. Se a DTFT de

uma dada seqüência existir, então ela é única e esta seqüência pode ser calculada através de sua DTFT pelo uso da operação de transformação inversa.

A DTFT $X(e^{j\omega})$ de $x[n]$ é definida por:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \quad (4.7)$$

Em geral, $X(e^{j\omega})$ é uma função complexa da variável real ω e pode ser expressa por seu módulo e fase:

$$X(e^{j\omega}) = |X(e^{j\omega})| e^{j\theta(\omega)}, \quad (4.8)$$

onde :

$$\theta(\omega) = \arg[X(e^{j\omega})] \quad (4.9)$$

Em muitas aplicações, a Transformada de Fourier é chamada de Espectro de Fourier e, por isso, $|X(e^{j\omega})|$ e $\theta(\omega)$ são chamados, respectivamente, de Espectro de Magnitude e Espectro de Fase.

Sabe-se também que $X(e^{j\omega})$ é uma função contínua e periódica de ω com período 2π . Sua periodicidade pode ser verificada abaixo, onde k é um inteiro:

$$X(e^{j(\omega_l + 2\pi k)}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j(\omega_l + 2\pi k)n} = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega_l n} e^{-j2\pi kn} = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega_l n} = X(e^{j\omega_l})$$

O sinal $x[n]$ pode ser calculado a partir de $X(e^{j\omega})$, usando-se a transformada inversa de Fourier, dada por:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega \quad (4.10)$$

A DTFT de um sinal de tempo discreto $x[n]$ derivado da amostragem de um sinal de tempo contínuo $x_a(t)$ pode ser também obtida a partir da Transformada de Fourier Contínua de $x_a(t)$. Assim, chamando-se a Transformada de Fourier Contínua de $x_a(t)$ de $X_a(j\Omega)$ e a

Transformada de Fourier Contínua de $x[n]$ de $X_s(j\Omega)$, sabe-se, pela Equação (3.7), que estes se relacionam da seguinte forma:

$$X_s(j\Omega) = \frac{1}{T_s} \sum_{-\infty}^{\infty} X_a(j(\Omega - k\Omega_s)) \quad (4.11)$$

Pela Equação (4.6), a DTFT de $x[n]$ pode ser facilmente obtida a partir de $X_s(j\Omega)$ por meio de um simples reescalonamento no eixo da frequência, fazendo-se:

$$X(e^{j\omega}) = X_s(j\Omega)_{\Omega=\omega/T_s} \quad (4.12)$$

Portanto, usando-se (4.12) em (4.11), obtém-se:

$$X(e^{j\omega}) = \frac{1}{T_s} \sum_{-\infty}^{\infty} X_a\left(j\left(\frac{\omega}{T_s} - k\Omega_s\right)\right) \quad (4.13)$$

4.2.2 A Transformada Discreta de Fourier

Sabe-se que, para uma seqüência de tempo discreto de comprimento N , apenas N amostras igualmente espaçadas da sua DTFT são suficientes para representar esta seqüência no domínio da frequência e que, a partir de tais amostras, é possível obter a seqüência de tempo discreto por uma simples operação de inversão.

Denotando-se uma seqüência de comprimento finito por $x[n]$, com $0 \leq n \leq N-1$, apenas N valores de $X(e^{j\omega})$, chamados amostras de frequência, situados em pontos de frequência $\omega = \omega_k$, com $k=0, 1, \dots, N-1$ são suficientes para univocamente determinar $X(e^{j\omega})$ e, logo, $x[n]$. Surge, então, o conceito da Transformada Discreta de Fourier ou DFT (*Discrete Fourier Transform*), que é aplicável somente a seqüências de comprimento finito.

Sendo $X[k]$ a DFT de $x[n]$, ela pode ser obtida a partir de $X(e^{j\omega})$ pela amostragem desta sobre o eixo ω com $0 \leq \omega \leq 2\pi$ nos pontos $\omega_k = 2\pi k / N$, $k=0, 1, \dots, N-1$. Da Equação

(4.7):

$$X[k] = X(e^{j\omega})_{\omega=2\pi k/N} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k=0,1,\dots,N-1 \quad (4.14)$$

Note-se que $X[k]$, assim como $x[n]$, também é uma seqüência de comprimento finito N , porém no domínio da freqüência.

Usando-se a notação comumente utilizada $W_N = e^{-j2\pi/N}$, (4.14) pode ser reescrita como:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k=0, 1, \dots, N-1 \quad (4.15)$$

A DFT inversa, também conhecida por IDFT (*Inverse Discrete Fourier Transform*), é dada por:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad n=0, 1, \dots, N-1 \quad (4.16)$$

4.2.3 A Transformada Rápida de Fourier

A FFT (*Fast Fourier Transform*), ou Transformada Rápida de Fourier, é um algoritmo que otimiza o cálculo da DFT. Ela reduz sensivelmente o tempo de processamento das rotinas que envolvem a análise de freqüência. De fato, existem vários algoritmos de FFT, porém todos eles utilizam uma mesma regra básica para eliminar operações redundantes, diminuindo o número de operações matemáticas e reaproveitando resultados previamente calculados.

Tradicionalmente, o cálculo da DFT poderia ser feito utilizando-se a Equação (4.15). Porém, para o uso da FFT, a seqüência $X[k]$ pode ser dividida em uma seqüência de amostras ímpares (com $n = 2r + 1$) e outra de amostras pares (com $n = 2r$), sendo r um inteiro:

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x[2r+1] W_N^{(2r+1)k} \quad (4.17)$$

Reescrevendo a equação acima:

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] (W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1] (W_N^2)^{rk} \quad (4.18)$$

Mas como $W_N^2 = e^{-2j(2\pi/N)} = W_{N/2}$, então:

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1] W_{N/2}^{rk} \quad (4.19)$$

A Equação (4.19) pode ser reescrita da seguinte forma:

$$X[k] = G[k] + W_N^k H[k], \quad (4.20)$$

onde $H[k]$ e $G[k]$ são duas seqüências de tempo discreto.

O algoritmo da FFT consiste em se aplicar a $G[K]$ e $H[K]$ o mesmo procedimento acima aplicado a $X[K]$. Este procedimento deve ser realizado repetidas vezes sobre as seqüências resultantes até que cada seqüência seja formada por apenas uma amostra. A Figura 4.1, extraída de [14], mostra uma seqüência de 8 pontos com os elementos pares e ímpares sendo agrupados separadamente para o cálculo de DFT de cada seqüência. Depois disto, com os quatro pontos de cada DFT, implementa-se o algoritmo mostrado anteriormente.

A DFT de 4 pontos deve ser expandida como mostrado na Figura 4.2 (retirada de [14]) até que só seja necessário o cálculo da DFT para conjuntos de 2 pontos. Nessa figura, percebe-se que uma estrutura básica de cálculo da DFT de dois pontos é utilizada como a primeira etapa do algoritmo de FFT. Observa-se que esta estrutura, ilustrada na Figura 4.3 (extraída de [14]) e denominada Radix-2, se repete por todo o algoritmo.

A divisão em estruturas Radix-2 é repetida 3 vezes: uma vez na entrada, outra vez numa fase intermediária e uma vez na saída, pois existem 2^3 pontos na seqüência de entrada.

Como pode ser percebido pelo gráfico da Figura 4.2, no algoritmo apresentado, resultados intermediários são utilizados para o cálculo de diversos pontos de saída. É isto que diminui o número de cálculos da DFT, pois as operações repetidas são realizadas uma única vez no algoritmo de FFT.

A redução no número de operações de multiplicação que precisam ser realizadas no cálculo da DFT através do algoritmo da FFT é o fator preponderante para o aumento de velocidade do processamento. Sem o uso da FFT, são realizadas aproximadamente N^2 operações de adição e multiplicação complexas para o cálculo da DFT. Por outro lado, com o uso da FFT este número cai para aproximadamente $N \log_2 N$ operações de adição e

multiplicação complexas. Por exemplo, para uma sequência de entrada com 1.024 pontos, serão realizadas 1.048.576 operações para o cálculo normal da DFT e 10.240 para o cálculo da FFT.

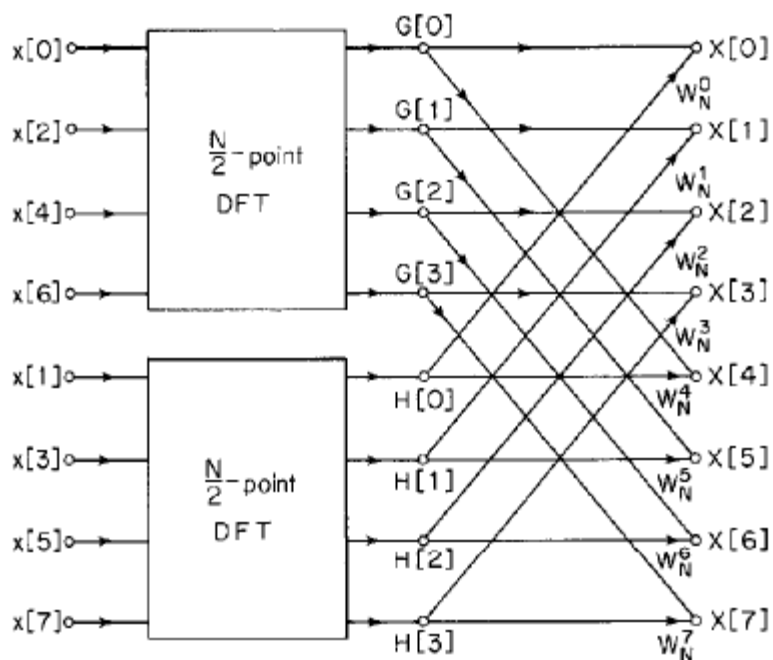


Figura 4.1: Primeira etapa do algoritmo de FFT.

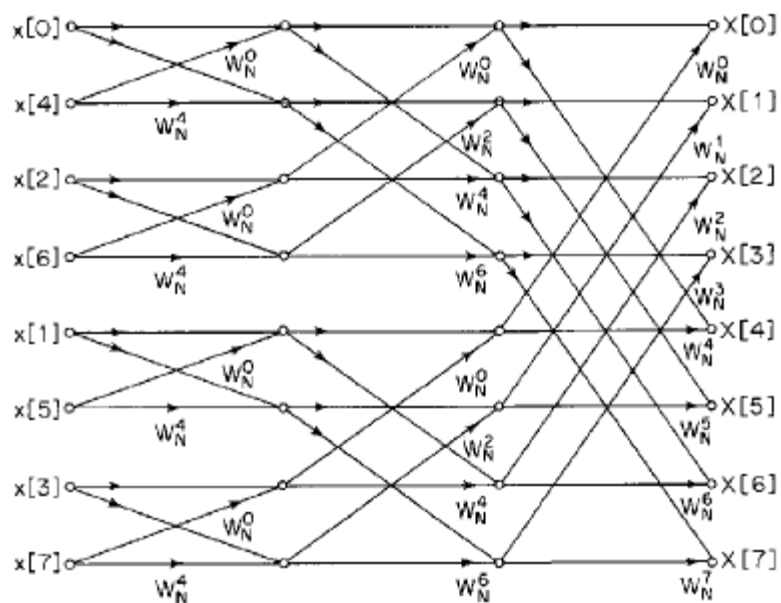


Figura 4.2: Estrutura completa utilizada para o cálculo da FFT de uma sequência de 8 pontos.

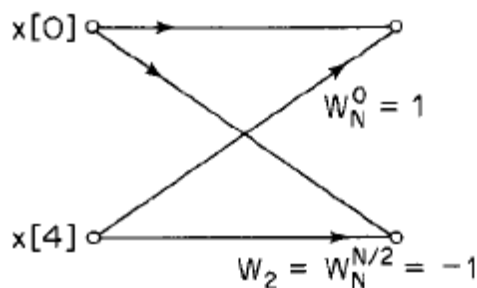


Figura 4.3: Estrutura Radix-2 utilizada durante todas as etapas de cálculo da FFT.

4.3 Sistemas de Tempo Discreto Lineares Invariantes no Tempo

Conforme já mencionado, os sistemas de tempo discreto promovem transformações sobre sinais discretos no tempo, gerando outros sinais discretos no tempo. Supondo-se que uma sequência $x[n]$ seja aplicada à entrada de um sistema de tempo discreto, este produzirá uma sequência $y[n]$ em sua saída, conforme ilustrado na Figura 4.4. Nesta seção, serão explicadas as características comuns aos sistemas de tempo discreto classificados como LIT: a linearidade e a invariância no tempo (ou invariância ao deslocamento no tempo). Posteriormente, será abordada a relação existente entre a entrada e a saída nos sistemas LIT, caracterizando-os no domínio do tempo. Por fim, será estudada a resposta em frequência, importante representação dos sistemas LIT no domínio da frequência.

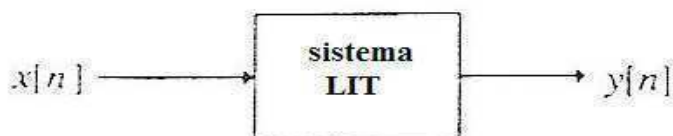


Figura 4.4- Entrada e saída em um sistema de tempo discreto LIT.

4.3.1 Linearidade e Invariância ao Deslocamento no Tempo

Um sistema é dito linear quando sempre vale o princípio da superposição. Mais precisamente, supondo-se que $x_1[n]$ e $x_2[n]$ sejam aplicados à entrada de um sistema, produzindo, respectivamente, $y_1[n]$ e $y_2[n]$ como saídas, este sistema é dito linear se e

somente se uma entrada $x[n]$ dada por $x[n] = \alpha x_1[n] + \beta x_2[n]$ produz uma saída $y[n]$ dada por $y[n] = \alpha y_1[n] + \beta y_2[n]$, onde α e β são constantes.

Supondo-se que uma sequência $x_1[n]$, quando aplicada à entrada de um sistema de tempo discreto, produza uma saída $y_1[n]$, este sistema será considerado invariante ao deslocamento no tempo ou, simplesmente, invariante no tempo, se uma entrada dada por $x[n] = x_1[n - n_o]$ produzir uma saída dada por $y[n] = y_1[n - n_o]$, onde n_o é qualquer inteiro. A propriedade de invariância no tempo assegura que, dada uma entrada específica, a saída é independente do tempo em que esta entrada está sendo aplicada.

Sistemas de tempo discreto LIT satisfazem ambas as propriedades descritas acima e, por isso, são sistemas mais fáceis de analisar e caracterizar e, por conseguinte, de projetar. Além disso, algoritmos de processamento de sinal muito úteis vêm sendo desenvolvidos usando sistemas deste tipo.

4.3.2 Caracterização de Sistemas LIT no Domínio do Tempo: a Resposta ao Impulso

Um sistema de tempo discreto LIT é completamente especificado por sua resposta ao impulso, ou seja, conhecida a resposta ao impulso, pode-se calcular a sua saída para uma entrada qualquer. A resposta ao impulso, denotada por $h[n]$, é a saída de um sistema LIT ao ser aplicado um impulso unitário $\delta[n]$ à sua entrada, conforme ilustrado na Figura 4.5. O impulso unitário é dado por:

$$\delta[n] = \begin{cases} 1, & n = 0, \\ 0, & n \neq 0 \end{cases} \quad (4.21)$$

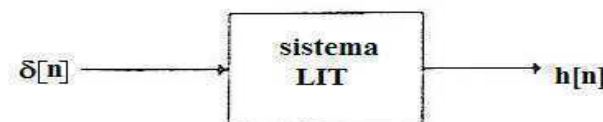


Figura 4.5- Resposta ao impulso de um sistema LIT.

Toda sequência pode ser representada no domínio do tempo por uma soma ponderada de impulsos unitários deslocados no tempo. A seguinte sequência de entrada $x[n]$ pode ser tomada como exemplo:

$$x[n] = 0.5\delta[n+2] + 1.5\delta[n-1] - \delta[n-2] + \delta[n-4] + 0.7\delta[n-6]$$

A saída produzida por um sistema LIT quando aplicada a entrada $x[n]$, expressa na equação acima, pode ser facilmente obtida conhecendo-se a resposta ao impulso $h[n]$ deste filtro. Como o sistema é invariante no tempo e a resposta à entrada $\delta[n]$ é $h[n]$, a resposta à entrada $\delta[n+2]$ será $h[n+2]$ e as respostas aos impulsos presentes nas demais parcelas de $x[n]$, $\delta[n-1]$, $\delta[n-2]$, $\delta[n-4]$, $\delta[n-6]$, serão dadas por $h[n-1]$, $h[n-2]$, $h[n-4]$ e $h[n-6]$ respectivamente. Portanto, como o sistema é linear, a resposta $y[n]$ à entrada $x[n]$ será dada por:

$$y[n] = 0.5 h[n+2] + 1.5h[n-1] - h[n-2] + h[n-4] + 0.7 h[n-6]$$

Uma maneira de representar uma sequência arbitrária $x[n]$ qualquer, levando-se em conta que ela pode ser representada no domínio do tempo por uma soma ponderada de impulsos unitários deslocados no tempo, é expressá-la por:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \quad (4.22)$$

Na expressão acima, $x[k]$ representa, especificamente, o valor da k -ésima amostra. A resposta de um sistema LIT para $x[k]\delta[n-k]$ será $x[k] h[n-k]$. Portanto, a saída $y[n]$ produzida por um sistema LIT para uma entrada qualquer $x[n]$, dada pela Equação (4.22), será:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \quad (4.23)$$

A equação acima pode ser alternativamente reescrita como:

$$y[n] = \sum_{k=-\infty}^{\infty} x[n-k]h[k] \quad (4.24)$$

Como expresso nas Equações (4.23) e (4.24), $y[n]$ pode ser calculado como a convolução das seqüências $x[n]$ e $h[n]$ e pode ser compactamente escrito como:

$$y[n] = x[n] * h[n], \quad (4.25)$$

onde a notação $*$ representa a operação de convolução, a qual apresenta a propriedade da comutatividade, ou seja, $x[n] * h[n] = h[n] * x[n]$.

4.3.3 Representação de Sistemas LIT no Domínio da Frequência: a Resposta em Frequência

A maioria dos sinais discretos no tempo encontrados na prática podem ser representados como uma combinação linear de um grande número, talvez infinito, de sinais discretos no tempo senoidais de diferentes frequências angulares. Portanto, através do conhecimento da resposta de um sistema LIT a uma simples entrada senoidal, pode-se determinar as respostas de tal sistema a entradas mais complicadas, pelo uso da propriedade da linearidade. Como um sinal senoidal pode ser expresso em termos de funções exponenciais, a resposta de um sistema LIT a uma entrada exponencial desperta um interesse prático. Isto está relacionado ao conceito de resposta em frequência, a representação de um sistema LIT no domínio da frequência.

Supondo-se que a entrada $x[n]$ é uma exponencial complexa da forma:

$$x[n] = e^{j\omega n}, -\infty < n < \infty \quad (4.26)$$

Substituindo-se (4.26) em (4.24), a saída $y[n]$ é dada por:

$$y[n] = \sum_{k=-\infty}^{\infty} e^{j\omega(n-k)} h[k] = \left(\sum_{k=-\infty}^{\infty} h[k] e^{-j\omega k} \right) e^{j\omega n} \quad (4.27)$$

A equação acima pode ser reescrita como:

$$y[n] = H(e^{j\omega}) e^{j\omega n}, \quad (4.28)$$

onde foi usada a seguinte definição:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n} \quad (4.29)$$

O termo $H(e^{j\omega})$ na equação acima é chamado de resposta em frequência e provê a descrição no domínio da frequência de um sistema LIT. Portanto, a resposta em frequência corresponde à DTFT da resposta ao impulso do sistema.

A Equação (4.28) implica que uma seqüência de entrada senoidal complexa de frequência angular ω produz uma saída também senoidal complexa de mesma frequência angular, porém ponderada por uma amplitude complexa $H(e^{j\omega})$, que é função da frequência de entrada ω e dos coeficientes da resposta ao impulso.

Como qualquer DTFT, a resposta em frequência é uma função complexa de ω com período 2π e pode ser expressa em termos de sua magnitude e fase:

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\theta(\omega)}, \quad (4.30)$$

onde $|H(e^{j\omega})|$ é chamada de resposta em magnitude e $\theta(\omega)$, dada por $\theta(\omega) = \arg[H(e^{j\omega})]$, é chamada de resposta em fase de um dado sistema.

Para sistemas LIT, a resposta em magnitude e a resposta em fase serão sempre funções reais de ω . Especificações de projetos para muitos sistemas discretos são feitas em termos das respostas em magnitude ou fase, ou ambas. Em alguns casos, a resposta em magnitude é especificada em decibéis, caso em que recebe o nome de função ganho, e é calculada por:

$$G(\omega) = 20\log_{10}|H(e^{j\omega})|dB \quad (4.31)$$

Para um sistema de tempo discreto com resposta ao impulso $h[n]$ real, a resposta em magnitude será sempre uma função par de ω e a resposta em fase será sempre uma função ímpar de ω .

A seguir será obtida a relação entrada-saída de um sistema LIT no domínio da frequência. Considerando-se que $Y(e^{j\omega})$ e $X(e^{j\omega})$ denotem respectivamente as DTFTs das

seqüências de saída $y[n]$ e entrada $x[n]$ e tomando a DTFT de ambos os lados na Equação (4.24), obtém-se:

$$Y(e^{j\omega}) = \sum_{n=-\infty}^{\infty} y[n] e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[n-k] h[k] \right) e^{-j\omega n} \quad (4.32)$$

Trocando os somatórios de lugar no lado direito da equação acima e rearrumando:

$$Y(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h[k] \left(\sum_{n=-\infty}^{\infty} x[n-k] e^{-j\omega n} \right) \quad (4.33)$$

Fazendo $l=n-k$ na equação acima:

$$Y(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h[k] \left(\sum_{l=-\infty}^{\infty} x[l] e^{-j\omega(l+k)} \right) = \sum_{k=-\infty}^{\infty} h[k] \left(\sum_{l=-\infty}^{\infty} x[l] e^{-j\omega l} \right) e^{-j\omega k} \quad (4.34)$$

O termo dentro dos parênteses no lado direito da Equação (4.34) pode ser identificado como $X(e^{j\omega})$, a DTFT de $x[n]$. Substituindo esta notação em (4.34) e rearrumando:

$$Y(e^{j\omega}) = \left(\sum_{k=-\infty}^{\infty} h[k] e^{-j\omega k} \right) X(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega}), \quad (4.35)$$

onde $H(e^{j\omega})$ é a resposta em frequência do sistema, dada por (4.29).

A Equação (4.35) finalmente relaciona a entrada e a saída de um sistema LIT no domínio da frequência. Portanto, a Transformada de Fourier da seqüência de saída é dada pelo produto da resposta em frequência do sistema LIT pela Transformada de Fourier da seqüência de entrada.

4.4 Filtros Digitais

Nesta seção, serão introduzidos alguns conceitos básicos sobre filtros digitais, focando-se principalmente sobre os filtros FIR de fase linear, sobretudo os filtros passa-

baixas. A partir do filtro passa-baixas, outros tipos de filtros podem ser facilmente derivados. Será estudado o projeto de filtros FIR de fase linear pela técnica conhecida como janelamento. Portanto, não se pretende abranger toda a teoria de filtros digitais. Os conceitos abordados nesta seção serão importantes para a compreensão das simulações descritas no Capítulo 5.

4.4.1 Filtros FIR de Fase Linear

Os sistemas LIT de tempo discreto podem ser classificados, de acordo com o comprimento de sua resposta ao impulso, em sistemas FIR (*Finite Impulse Response*) ou IIR (*Infinite Impulse Response*). O presente trabalho estudará apenas os sistemas FIR. Um sistema de tempo discreto é classificado como FIR se sua resposta ao impulso $h[n]$ tem comprimento finito, isto é, $h[n]=0$ para $n < N_1$ e $n > N_2$ com $N_1 < N_2$, onde N_1 e N_2 são dois inteiros. Para este tipo de sistema, a saída $y[n]$, de acordo com (4.24), pode ser expressa como:

$$y[n] = \sum_{k=N_1}^{N_2} h[k]x[n-k] \quad (4.36)$$

Além disso, sabe-se também que sistemas FIR são sempre estáveis no sentido BIBO (*Bounded-Input Bounded-Output*).

Uma das aplicações mais importantes dos sistemas LIT são os filtros digitais. Através deles, é possível deixar passar, preferencialmente sem distorção, certos componentes de frequência presentes na sequência de entrada e bloquear outros componentes de frequência. Conforme visto na fórmula da IDTFT, uma sequência de entrada arbitrária pode ser vista como uma soma ponderada de um infinito número de seqüências exponenciais, ou, equivalentemente, como uma soma ponderada de infinitas seqüências senoidais. Como consequência deste fato, através da escolha apropriada dos valores da resposta em magnitude de um filtro digital nas frequências correspondentes às frequências dos componentes senoidais de entrada, alguns destes componentes senoidais podem ser seletivamente atenuados ou filtrados com relação a outros componentes. A Figura 4.6 mostra a resposta em frequência de quatro tipos de filtros digitais ideais: filtro passa-baixas, filtro passa-altas, filtro passa-faixa e filtro corta-faixa. Como pode ser observado na Figura 4.6, presente em [9], os

filtros ideais exibem resposta em frequência plana na faixa de frequências que se quer preservar, também chamada de banda passante, e resposta em frequência nula nas demais frequências que compõem a chamada banda de rejeição. As frequências ω_c , ω_{c1} e ω_{c2} são denominadas frequências de corte dos seus respectivos filtros.

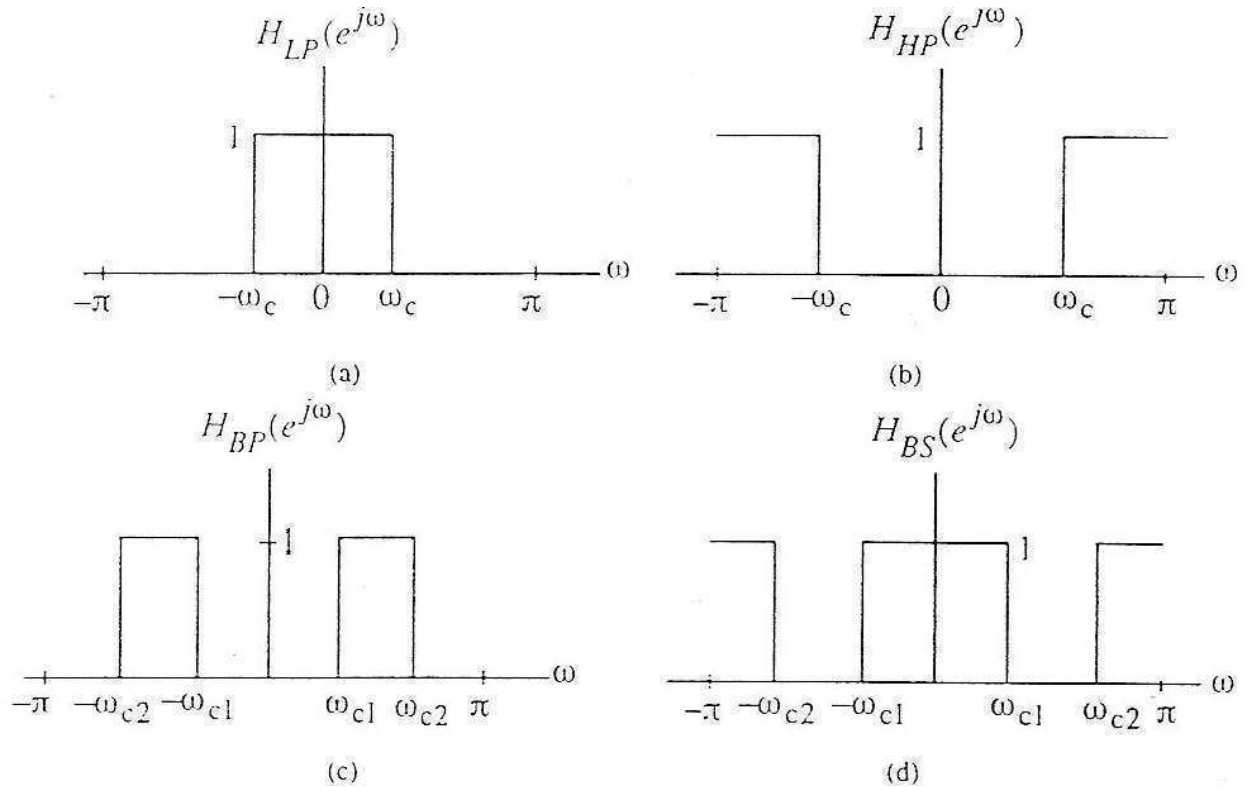


Figura 4.6 – Quatro tipos de filtro ideal: (a) filtro passa-baixas, (b) filtro passa-altas, (c) filtro passa-faixa e (d) filtro corta-faixa.

Sabe-se que, para se filtrar um sinal de entrada, obtendo-se os componentes de frequência da banda passante sem distorção em magnitude e fase, a resposta em frequência do filtro deve exibir, dentro da banda de interesse, uma resposta em magnitude unitária e uma resposta em fase linear. A Figura 4.7, extraída de [9], mostra a resposta em frequência de um filtro passa-baixas com estas características. Seja esta resposta em frequência dada por:

$$H_{LP}(e^{j\omega}) = \begin{cases} e^{-j\omega n_0}, & 0 < |\omega| < \omega_c, \\ 0, & \omega_c \leq |\omega| \leq \pi \end{cases} \quad (4.37)$$

O filtro acima, dado por (4.37), possui resposta em fase linear e $-n_0$ é a inclinação da resposta em fase do filtro, pois:

$$\frac{d \arg(H_{LP}(e^{j\omega}))}{d\omega} = \frac{d(-\omega n_0)}{d\omega} = -n_0 \quad (4.38)$$

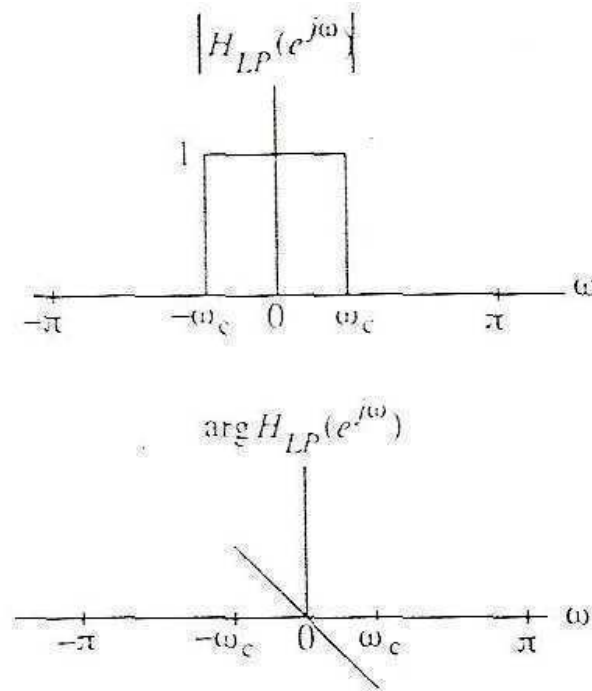


Figura 4.7 – Resposta em frequência de um filtro passa-baixas ideal com resposta em fase linear na banda passante.

Pretende-se obter a expressão da resposta ao impulso para este filtro. Para tanto, é necessário tomar a IDTFT de sua resposta em frequência. Primeiramente, uma resposta em frequência hipotética $H_{LP}'(e^{j\omega})$ é expressa da seguinte forma:

$$H_{LP}'(e^{j\omega}) = \begin{cases} 1, & 0 < |\omega| < \omega_c, \\ 0, & \omega_c \leq |\omega| \leq \pi \end{cases} \quad (4.39)$$

Calculando sua IDTFT:

$$\begin{aligned} h_{LP}'(n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{LP}'(e^{j\omega}) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega \\ &= \frac{1}{2\pi} \left(\frac{e^{j\omega_c n}}{jn} - \frac{e^{-j\omega_c n}}{jn} \right) = \frac{\text{sen}(\omega_c n)}{\pi n}, -\infty < n < \infty \end{aligned} \quad (4.40)$$

Aplicando-se a propriedade da Transformada de Fourier quanto ao deslocamento no tempo, dada por $DTFT(g[n-n_0]) = e^{-j\omega n_0} G(j\omega)$, onde $G(j\omega) = DTFT(g[n])$, calcula-se $h_{LP}(n)$, a resposta ao impulso do filtro da Figura 4.7, pois $H_{LP}(e^{j\omega}) = e^{-j\omega n_0} H_{LP}'(e^{j\omega})$:

$$h_{LP}[n] = h_{LP}'[n - n_0] = \frac{\text{sen}(\omega_c(n - n_0))}{\pi(n - n_0)}, -\infty < n < \infty \quad (4.41)$$

Pela equação acima, pode-se notar que, na prática, é impossível realizar um filtro com dimensão finita e resposta em frequência ideal conforme mostrado na Figura 4.7 pois sua resposta ao impulso é infinita. Além disso, $h_{LP}[n]$ é não causal, ou seja, $h_{LP}[n] \neq 0$ para $n < 0$. Contudo, truncando $h_{LP}[n]$, é possível realizar um filtro FIR causal que se aproxime do filtro passa-baixas ideal de fase linear. Sabe-se que, para que um filtro FIR de comprimento $N+1$ tenha resposta em fase linear, é necessário que sua resposta ao impulso apresente simetria em torno de $N/2$. Neste caso, a resposta ao impulso de um filtro FIR obtido a partir do truncamento de $h_{LP}[n]$, será dada por:

$$h_{LP}''[n] = \frac{\text{sen}(\omega_c(n - N/2))}{\pi(n - N/2)}, 0 \leq n \leq N \quad (4.42)$$

Por causa da simetria dos coeficientes de $h_{LP}''[n]$, a resposta em frequência deste filtro FIR será dada por:

$$H_{LP}''(e^{j\omega}) = e^{-j\omega N/2} |H_{LP}''(e^{j\omega})|, \quad (4.43)$$

onde $|H_{LP}''(e^{j\omega})|$ é a resposta em magnitude do filtro.

A Figura 4.8, retirada de [9], exibe esta resposta em magnitude para uma frequência de corte fixa e para diversos valores de N . É observável que, independente do valor de N , a resposta em magnitude do filtro FIR, obtido pelo truncamento do filtro ideal de comprimento infinito dado pela Equação (4.41), exibe um corte menos abrupto em comparação com o filtro ideal e o surgimento de *ripples* (oscilações). Além disso, um aumento em N resulta em um corte mais abrupto e no aumento do número de *ripples*.

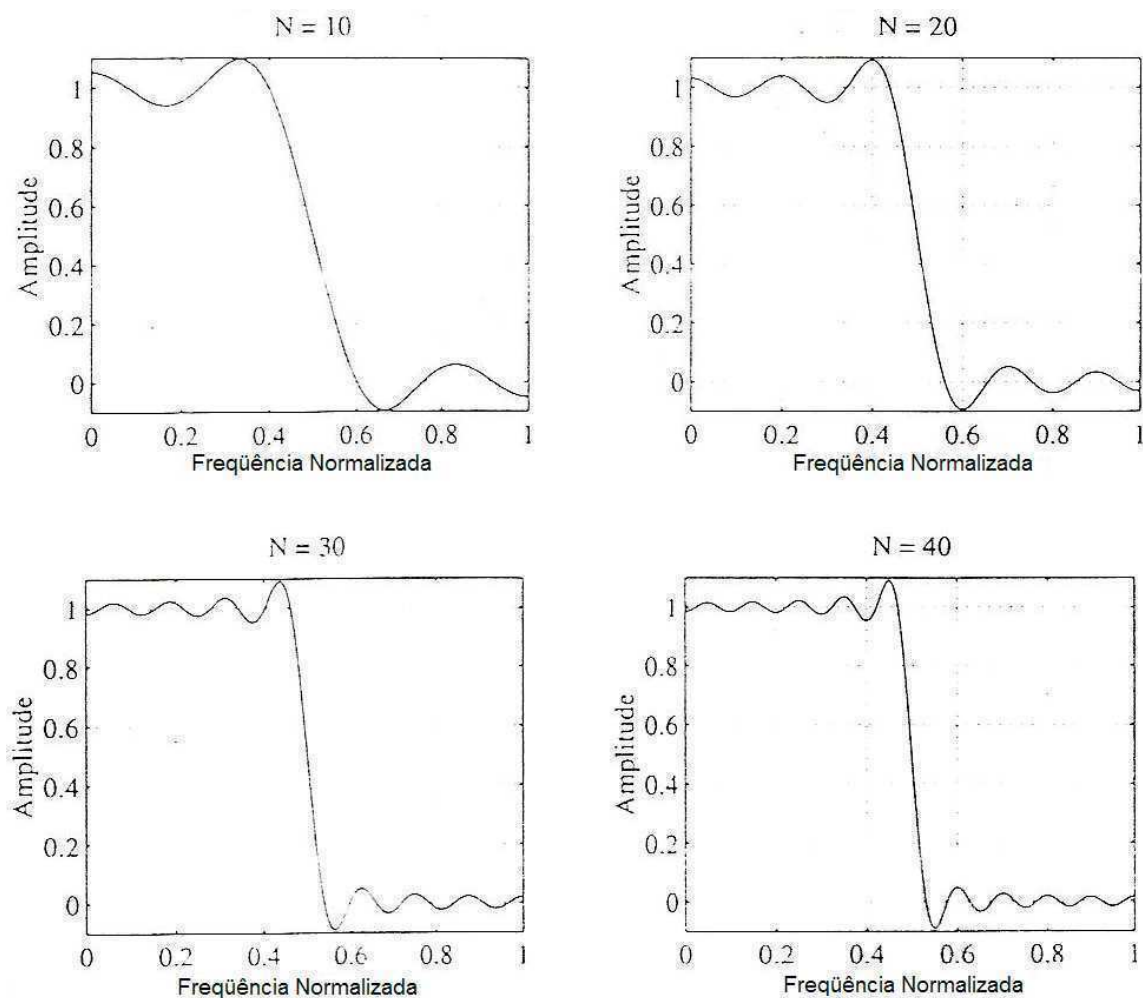


Figura 4.8 – Reposta em magnitude para o filtro de resposta ao impulso dada por (4.42) para quatro valores distintos de N .

4.4.2 Projeto de Filtros FIR de Fase Linear

Basicamente, o projeto de filtros digitais envolve três estágios principais:

- Especificação das propriedades do sistema desejado
- Aproximação do sistema usando um sistema discreto causal, o que inclui escolher o tipo de filtro a utilizar
- A realização do sistema.

A presente seção pretende focar-se no projeto de filtros digitais FIR pelo método de

janelamento.

Supondo-se que o sinal discreto a ser filtrado advinha de um sinal contínuo limitado em banda e amostrado usando-se uma frequência de amostragem suficiente para se evitar o aliasing, então o filtro passa-baixas contínuo ideal que se deseja projetar, denominado de filtro efetivo, se comportará como um sistema LIT contínuo no tempo com resposta em frequência dada por:

$$H_{eff}(j\Omega) = \begin{cases} H(e^{j\Omega T_s}), & |\Omega| < \pi / T_s \\ 0, & |\Omega| > \pi / T_s \end{cases} \quad (4.44)$$

Neste caso, é possível converter as especificações do filtro efetivo contínuo em especificações para o filtro discreto através da relação já discutida, $\omega = \Omega T_s$, onde T_s é o período de amostragem. Ou seja, $H(e^{j\omega})$ é especificado em um período por:

$$H(e^{j\omega}) = H_{eff}\left(\frac{j\omega}{T_s}\right), |\omega| < \pi \quad (4.45)$$

Tome-se como exemplo um sinal de tempo discreto $x[n]$ obtido a partir de um sinal contínuo $x_a(t)$ com período de amostragem $t=10^{-4}$ segundos. Suponha-se as seguintes especificações para um filtro passa-baixas contínuo que, se fosse utilizado para filtrar $x_a(t)$, produziria o mesmo efeito do filtro digital que se quer obter para filtrar $x[n]$:

- O ganho na faixa de passagem, dada por $0 \leq \Omega \leq 2\pi(2000)$, deve variar de no máximo ± 0.01
- O ganho na faixa de rejeição, dada por $\Omega \geq 2\pi(3000)$, não deve ser maior que 0.001

A Figura 4.9, presente em [14], ilustra as especificações do filtro acima onde Ω_p é o limite superior da banda passante e Ω_s o limite inferior da banda de rejeição. Nessa figura, δ_1 representa a máxima variação no ganho que a resposta em magnitude do filtro pode exibir dentro da banda passante e δ_2 representa a máxima variação no ganho dentro da banda de rejeição. Para o exemplo dado:

$$\Omega_p = 2\pi(2000) \text{ e } \Omega_s = 2\pi(3000)$$

$$\delta_1 = 0.01 \text{ e } \delta_2 = 0.001$$

As bandas passante e de rejeição podem ser, então vistas no domínio da frequência angular digital pela conversão dos valores de Ω_p e Ω_s usando-se a relação $\omega = \Omega T_s$:

$$\omega_p = \Omega_p T_s = 2\pi(2000)10^{-4} = 0.4\pi$$

$$\omega_s = \Omega_s T_s = 2\pi(3000)10^{-4} = 0.6\pi$$

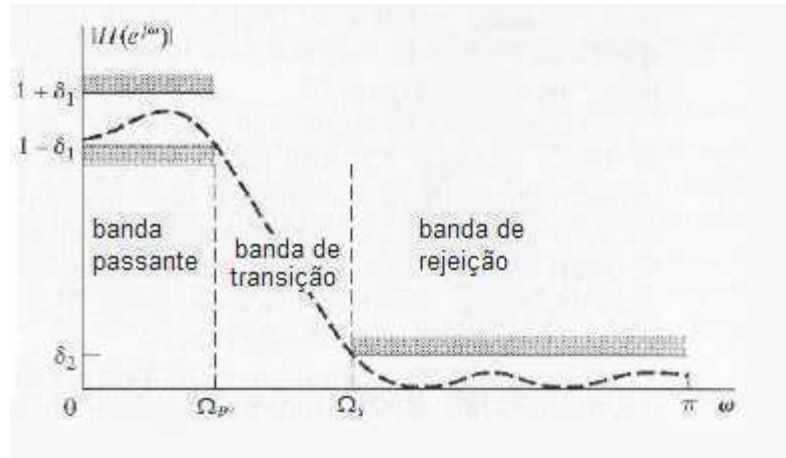


Figura 4.9- Especificações de um filtro.

O método mais simples de implementar projetos de filtros FIR causais é chamado de método de janelamento. Este método geralmente se inicia com a especificação do filtro seguindo uma estipulada resposta em frequência ideal, que pode ser representada como:

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n}, \quad (4.46)$$

onde $h_d[n]$ é a correspondente resposta ao impulso do filtro ideal.

Na Seção 4.4.1, foi discutida a necessidade de se truncar esta resposta ao impulso para realizar-se um filtro real FIR causal e foi realizado o truncamento da forma mais simples possível utilizando-se uma **janela retangular**, ou seja, definiu-se um novo sistema com resposta ao impulso $h[n]$ dada por:

$$h[n] = \begin{cases} h_d[n], & 0 \leq n \leq N \\ 0, & \text{caso contrário} \end{cases} \quad (4.47)$$

Mais genericamente, pode-se representar $h[n]$ como o produto da resposta ao impulso desejada por uma janela de duração finita $w[n]$, ou seja:

$$h[n] = h_d[n]w[n] \quad (4.48)$$

O truncamento simples da resposta ao impulso, expresso em (4.47), corresponde a janela retangular, que é dada por:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N \\ 0, & \text{caso contrário} \end{cases} \quad (4.49)$$

A resposta em frequência do filtro truncado será dada pela DTFT de $h[n]$. Como $h[n]$ é calculado genericamente por (4.48), sua DTFT será a convolução da resposta em frequência ideal pela DTFT da janela:

$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta, \quad (4.50)$$

onde $W(e^{j\omega})$ é a DTFT de $w[n]$.

Na medida em que se aumenta o comprimento da janela, ou seja, se aumenta N , $W(e^{j\omega})$ fica cada vez mais concentrada em torno de $\omega = 0$, aproximando-se cada vez mais do impulso, pois, no limite, quando N tende a infinito, $w[n]$ tende para a unidade e, conseqüentemente, $W(e^{j\omega})$ tende para $\delta(\omega)$, um impulso na origem das frequências. Portanto, quando N aumenta, $H(e^{j\omega})$ fica mais parecida com $H_d(e^{j\omega})$, a resposta ideal. Contudo, a opção por um N muito grande pode tornar o filtro muito complexo, aumentando o processamento requerido no cálculo para se obter a saída do filtro. A Figura 4.10, presente em [14], ilustra o efeito do janelamento para uma janela retangular.

Infelizmente, quando se utilizam janelas retangulares para o projeto de filtros, chega-se a um determinado ponto onde, por mais que se aumente o número de coeficientes dos filtros, não é possível continuar reduzindo a amplitude das oscilações presentes em $H(e^{j\omega})$. Para minimizar os efeitos das oscilações, utilizam-se diferentes tipos de janelas. Estas possuem a característica de amenizarem suavemente o fim da resposta ao impulso do filtro passa baixas ideal. Isto faz com que haja uma diminuição dos lóbulos laterais existentes na banda de rejeição, bem como a redução da amplitude das oscilações na banda de passagem.

Obviamente, a melhoria na resposta de amplitude tem um custo, que é o alargamento da banda de transição, mas que pode ser compensada com o uso de um maior número de coeficientes.

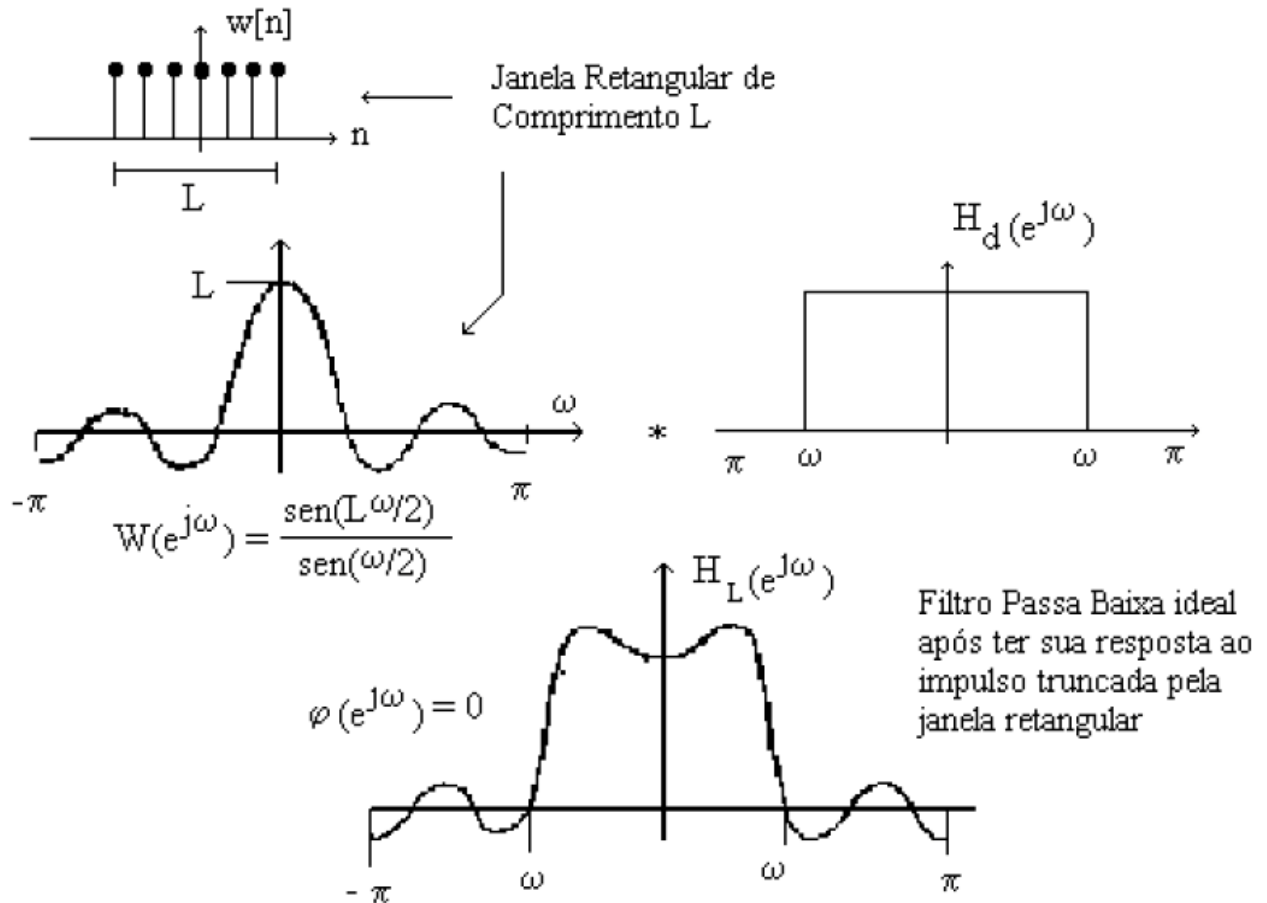


Figura 4.10 – Efeitos do janelamento: resposta em frequência de um filtro passa-baixas ideal após ter sua resposta ao impulso truncada por uma janela retangular.

As janelas mais comumente utilizadas no projeto de filtros FIR são:

a) Janela Retangular:

Sua resposta ao impulso é dada por (4.49).

b) Janela Triangular ou de *Bartlett*:

$$w[n] = \begin{cases} 2n/N, & 0 \leq n \leq N/2 \\ 2 - 2n/N, & N/2 < n \leq N \\ 0, & \text{caso contrário} \end{cases} \quad (4.51)$$

c) Janela de *Hamming*:

$$w[n] = \begin{cases} \alpha + (1 - \alpha) \cos\left(\frac{2\pi n}{N}\right), & 0 \leq n \leq N \\ 0, & \text{caso contrário} \end{cases} \quad (4.52)$$

onde o parâmetro α é geralmente 0.56, mas pode assumir outros valores. Quando $\alpha = 0,5$, a janela é chamada de janela de *Hanning*.

d) Janela de *Blackman*:

$$w[n] = \begin{cases} 0.42 - 0.5 \cos\left(\frac{2\pi n}{N}\right) + 0.08 \cos\left(\frac{4\pi n}{N}\right), & 0 \leq n \leq N \\ 0, & \text{caso contrário} \end{cases} \quad (4.53)$$

As formas das janelas acima podem ser vistas na Figura 4.11 (extraída de [14]), traçadas como uma função de uma variável contínua. Entretanto, as seqüências de janelamento são definidas apenas para valores inteiros de n .

A Figura 4.12, presente em [14], mostra a resposta em freqüência de um mesmo filtro FIR após passar por diferentes tipos de janelas com mesmo comprimento.

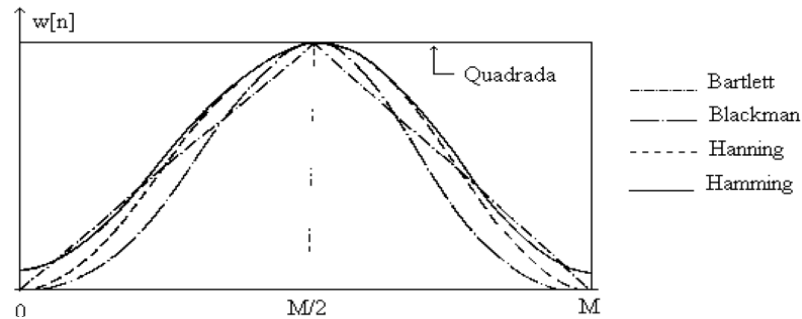


Figura 4.11-Janelas para projetos de filtros FIR.

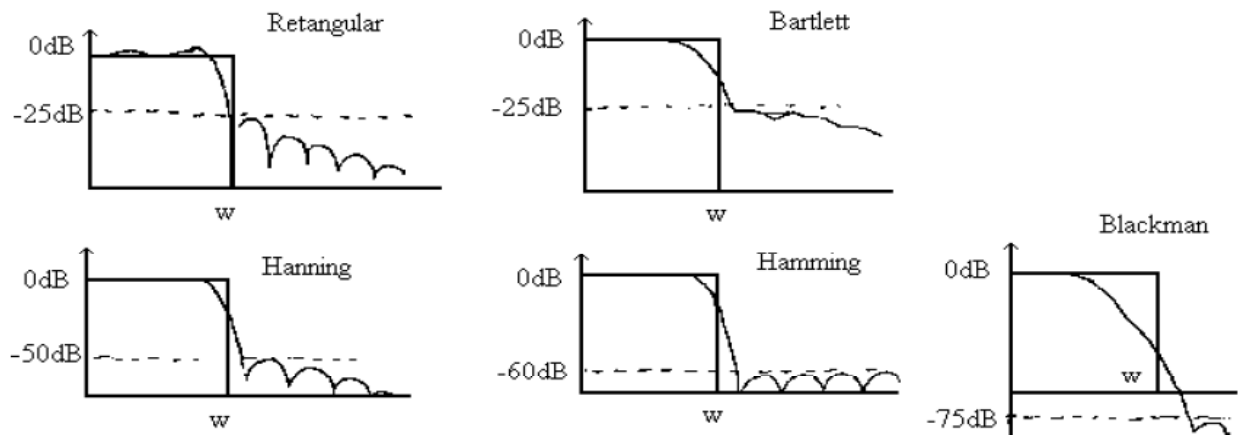


Figura 4.12- Resposta em freqüência de um mesmo filtro FIR após passar por diferentes tipos de janelas.

É importante ressaltar que existem duas características principais de $W(e^{j\omega})$ que influenciam a resposta em frequência presente no filtro resultante do processo de janelamento: a largura de seu lóbulo principal e sua razão de *ripple*. A largura do lóbulo principal de $W(e^{j\omega})$ influencia no tamanho da banda de transição do filtro resultante: quanto menor a largura do lóbulo principal, menor a banda de transição. A largura do lóbulo principal, por sua vez, é controlada pela escolha de N : quanto maior N , menor será a largura do lóbulo principal. A razão de *ripple* é a relação entre a amplitude do lóbulo principal e a amplitude do primeiro lóbulo secundário (ou lateral). Ela determina a mínima atenuação da banda de rejeição e o *ripple* da banda de passagem. A razão de *ripple*, por sua vez, é controlada pela escolha da janela, sendo que N praticamente não influencia esta característica.

Portanto, o projeto de um filtro FIR pode seguir o seguinte procedimento: a partir de uma especificação, escolher o tipo de janela que satisfaça a atenuação na banda de rejeição e, depois, escolher N para satisfazer a banda de transição desejada, usando-se tentativa erro.

Uma outra janela muito utilizada nos projetos de filtros FIR é a janela de Kaiser. Ela é definida por:

$$w[n] = \begin{cases} \frac{I_0\left[\beta\sqrt{1 - [(n - \alpha)/\alpha]^2}\right]}{I_0[\beta]}, & 0 \leq n \leq N \\ 0, & \text{caso contrário} \end{cases} \quad (4.54)$$

onde $\alpha = N/2$, $I_0(x)$ é a função de Bessel modificada de primeira espécie e ordem zero e β é uma constante que controla a razão de *ripple*.

A janela de Kaiser é uma janela ótima no sentido de que o *ripple* do lóbulo lateral é minimizado quanto aos mínimos quadrados para uma certa largura do lóbulo principal. Dada uma atenuação mínima exigida $A_{\min}$ para o lóbulo lateral, dada em dB, o parâmetro β é escolhido da seguinte forma:

$$\beta = \begin{cases} 0,1102(A_{\min} - 8,7), & A_{\min} > 50 \\ 0,5842(A_{\min} - 21)^{0,4} + (A_{\min} - 21), & 21 \leq A_{\min} \leq 50 \\ 0, & A_{\min} < 21 \end{cases} \quad (4.55)$$

A escolha do parâmetro N e, por conseguinte, de α , depende da especificação da largura da banda de transição exigida $\Delta\omega$ e de $A_{\min}$. N pode ser calculado por:

$$N = \frac{A_{min} - 8}{2.285 \Delta\omega} \quad (4.56)$$

Determinados β , N e α , pode-se facilmente calcular $w[n]$ pela Equação (4.54). Então, pode-se obter, através de $w[n]$, a resposta em frequência do filtro FIR resultante.

Conforme visto anteriormente, para se evitar distorção, é desejável que os filtros FIR tenham resposta em fase linear. O projeto de filtros FIR deve levar isto em consideração. É notório que todas as janelas definidas anteriormente possuem a propriedade de serem simétricas em relação ao ponto $N/2$, o que assegura a resposta em fase linear para os filtros delas originados.

5

APLICAÇÃO-EXEMPLO: RECEPÇÃO AM

Nos capítulos anteriores, foram estudados os principais tópicos para a compreensão dos rádios de *software*, passando pelos conceitos básicos, principais arquiteturas, *hardware* e teoria de processamento digital de sinais. Neste capítulo será explorada uma aplicação-exemplo simples: a recepção AM (*Amplitude Modulation* ou Modulação em Amplitude) para que se vislumbre como é utilizado o conceito de rádio de *software* em um dispositivo real.

Na Seção 5.1, encontra-se uma breve revisão de como comumente são realizadas a transmissão e recepção AM analógicas nos rádios tradicionais com ênfase na descrição dos processos de modulação e demodulação AM-DSB (*Double Sideband* ou Banda Lateral Dupla) com portadora e AM-SSB-SC (*Single Sideband Supressed Carrier*, ou Banda Lateral Única com portadora suprimida); por simplicidade, AM-SSB-SC será referida, daqui por diante, simplesmente como AM-SSB. Na Seção 5.2, é descrito como se daria a recepção AM usando o conceito de rádio de *software* para as demodulações AM-DSB com portadora e AM-SSB. Finalmente, na Seção 5.3 são descritas as simulações computacionais realizadas no *software Scilab*, cujos resultados comprovam a eficácia dos esquemas de recepção e demodulação apresentados na Seção 5.2. Também na Seção 5.3, os resultados das simulações são analisados usando-se a teoria de processamento de sinais.

O *Scilab* é uma plataforma *open-source* e gratuita para computação numérica. Para fazer *download* deste, veja [15]. Para maiores informações sobre o *Scilab* e sobre as funções matemáticas disponibilizadas por este, veja [16].

5.1 A Transmissão e a Recepção AM Analógicas

A radiodifusão AM utiliza a mais antiga de todas as formas de modulação de ondas de rádio: a modulação em amplitude. Ela consiste basicamente na utilização de um transdutor, por exemplo, um microfone, para transformar as ondas sonoras em sinais elétricos e no emprego desses sinais para modular a amplitude de uma onda de rádio que lhes serve de veículo portador: a onda portadora. Enquanto a portadora é gerada com amplitude e frequência constantes na estação emissora, as ondas de áudio apresentam-se como uma mistura irregular de frequências distintas. As ondas de áudio, após serem transformadas em sinais elétricos, imprimem suas variações na onda portadora, modulando sua amplitude. Na recepção, o demodulador detecta as variações de amplitude da onda portadora, produzindo um sinal de saída que é reproduzido pelos alto-falantes.

A Figura 5.1, extraída de [17], mostra um receptor AM super-heterodino que se enquadra no modelo de um receptor tradicional presente na Figura 2.1-a. A antena recebe RF na faixa de 535 a 1605 kHz, composta de diversos canais com largura de banda de 10kHz. Logo após a antena, um filtro de RF sintonizável, com fator de qualidade não muito bom, seleciona o canal desejado que, em seguida, é amplificado. Porém, para que se eliminem completamente os canais adjacentes, é feita, primeiramente, a conversão do sinal de RF para uma frequência mais baixa e fixa, dita frequência intermediária (FI) num processo conhecido como conversão para baixo. No caso da recepção AM, a FI é de 455 kHz. Esta conversão para baixo é feita por um misturador multiplicando-se o sinal de RF por um sinal de um oscilador local com frequência de oscilação f_{ol} dada por:

$$f_{ol}=FI+f_{ch} , \quad (5.1)$$

onde f_{ch} é a frequência central do canal sintonizado.

A frequência do oscilador sempre se adapta a f_{ch} de modo que, após a conversão para baixo, o canal sempre esteja centrado em FI, pois a multiplicação do canal em RF pelo sinal do oscilador faz com que o espectro do canal seja transladado para a frequência $f_{ol}-f_{ch}=FI$, em torno da qual ele é filtrado pelo filtro de FI (veja Figura 5.1).

Após esse estágio, os canais adjacentes já foram eliminados e o sinal é amplificado e em seguida demodulado. Após a demodulação, o sinal já se encontra na faixa de frequências audível e é filtrado e amplificado, gerando a saída de áudio que pode alimentar um alto-falante, por exemplo.

A cadeia de transmissão AM tem forma inversa à de recepção: o sinal de áudio gerado por um microfone é amplificado e filtrado e, em seguida, modula em amplitude uma portadora com frequência FI muito maior que as frequências presentes no sinal de áudio. Em seguida, o sinal em FI sofre um processo de conversão para cima e tem seu espectro transladado para uma frequência de RF ainda maior. Posteriormente, o sinal é filtrado e amplificado, sendo transmitido por uma antena.

As Seções 5.1.1 e 5.1.2 seguintes focalizarão algumas técnicas de modulação (empregadas na transmissão) e demodulação (empregadas na recepção) AM.

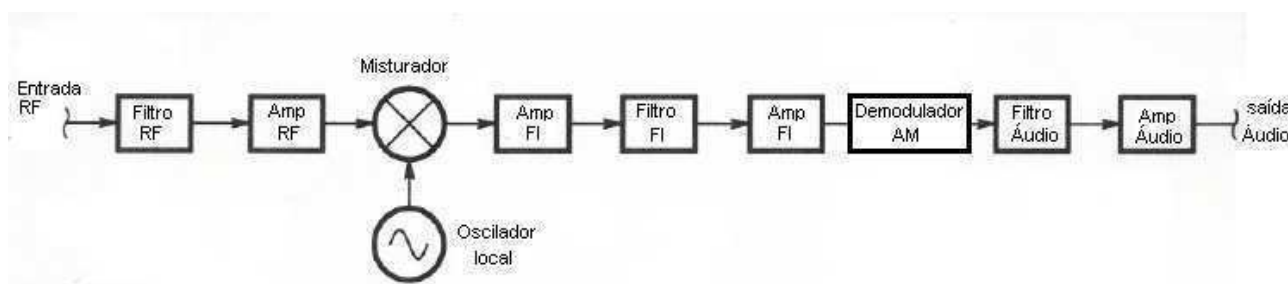


Figura 5.1- Receptor AM super-heterodino.

5.1.1 A Modulação e a Demodulação AM-DSB com Portadora

A forma mais simples de se implementar e mais convencional de modulação AM é a DSB com portadora. Ela pode ser descrita pela seguinte equação:

$$\Phi_{AM}(t) = A \cos \Omega_c t + f(t) \cos \Omega_c t = [A + f(t)] \cos \Omega_c t, \quad (5.2)$$

onde:

$\Phi_{AM}(t)$ é o sinal resultante da modulação, ou sinal modulado

$A \cos \Omega_c t$ é a portadora senoidal de amplitude A e frequência angular Ω_c .

$f(t)$ é o sinal modulante ou modulador, que contém a informação que se quer transmitir.

Na Figura 5.2, extraída de [17] e adaptada por questões de nomenclatura, são apresentadas as formas de onda de um sinal modulante (letra a) e do sinal modulado resultante para duas condições: quando $A + f(t)$ é sempre positivo para todo t (letra b) e quando isto não ocorre (letra c). Observe que quando $A + f(t)$ é sempre positivo, a envoltória (sinal tracejado na Figura 5.2-b, obtido da interpolação dos picos positivos do sinal

modulado) preserva a forma de $f(t)$ e tem valor $A+f(t)$. Porém, isto não ocorre quando esta condição não é verdadeira (caso da envoltória da Figura 5.2-c). Quando a envoltória do sinal modulado reproduz a forma de onda do sinal modulante, o processo de demodulação (processo de recuperação do sinal modulante a partir do sinal modulado) pode ser realizado de uma forma bem simples, extraindo-se do sinal modulado a sua envoltória, o que será visto posteriormente.

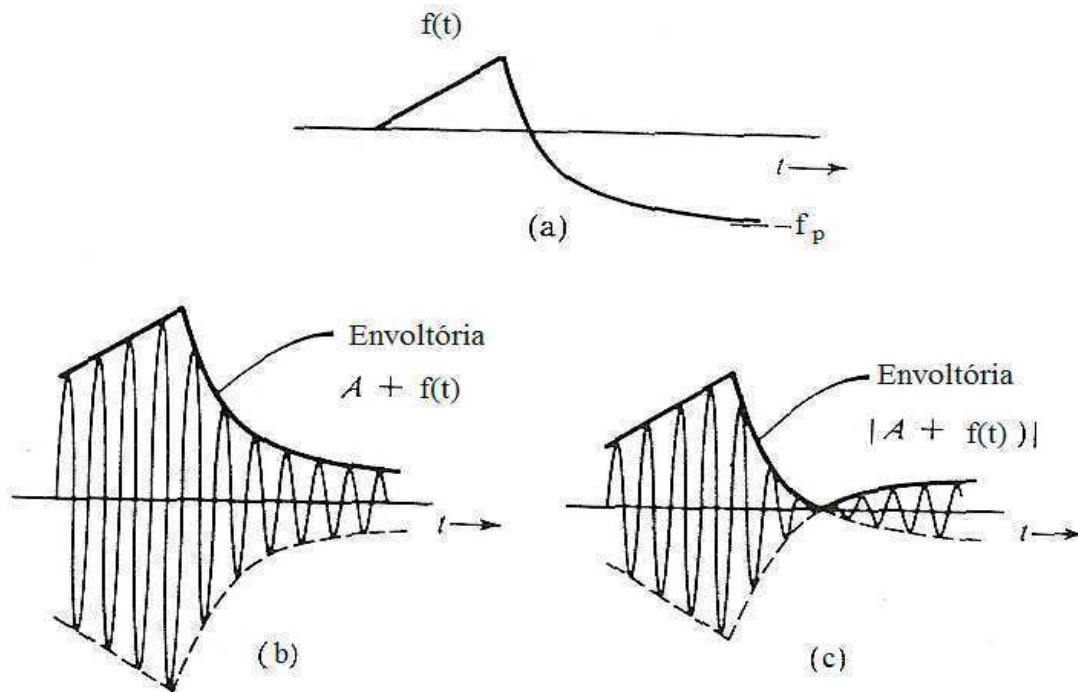


Figura 5.2- Modulação AM -DSB com portadora e a formação da envoltória.

Portanto, a condição necessária para o uso da detecção de envoltória é dada por:

$$A + f(t) \geq 0 \text{ para todo } t \quad (5.3)$$

Seja f_p a amplitude máxima negativa que $f(t)$ possa assumir (veja Figura 5.2-a), sabe-se que $f(t) \geq -f_p$. Portanto, a condição (5.3) é equivalente a:

$$A \geq f_p \quad (5.4)$$

A expressão (5.4) mostra que existe uma amplitude mínima para a portadora para que a detecção de envoltória seja viabilizada.

Define-se, então, o índice de modulação μ como a razão entre a amplitude máxima do sinal modulante e a amplitude da portadora:

$$\mu = \frac{f_p}{A} \quad (5.5)$$

Uma vez que f_p e A são constantes positivas, para que a Inequação (5.4) seja

verdadeira, o índice de modulação deve estar entre 0 e 1, ou seja:

$$0 \leq \mu \leq 1 \quad (5.6)$$

Quando, contrariamente a (5.4), $A < f_p$, ou seja $\mu > 1$, a detecção de envoltória não é mais possível por haver sobremodulação.

É também importante analisar-se a modulação DSB com portadora do ponto de vista da frequência. Tomando-se a Transformada da Fourier de (5.2), obtém-se:

$$\Phi_{AM}(\Omega) = \frac{1}{2} [F(\Omega + \Omega_c) + F(\Omega - \Omega_c)] + \pi A [\delta(\Omega + \Omega_c) + \delta(\Omega - \Omega_c)], \quad (5.7)$$

onde $F(f)$ é Transformada de Fourier do sinal $f(t)$, o sinal modulante.

A Figura 5.3 mostra os formatos dos espectros de frequência do sinal modulante, da portadora e do sinal modulado em amplitude AM-DSB com portadora. Pode-se observar que o processo de modulação em amplitude simplesmente translada (desloca) de Ω_c (frequência angular da portadora) o espectro do sinal modulante. Além disso, no espectro do sinal modulado, aparecem impulsos que correspondem à portadora senoidal e em torno dos quais é reproduzido o espectro do sinal modulante. Por isso, este tipo de modulação é denominado com portadora. No espectro do sinal modulado, existem duas bandas laterais: uma superior, dita USB (*Upper Sideband*), e outra inferior, dita LSB (*Lower Sideband*). Isto faz com que este tipo de modulação em amplitude seja conhecida como DSB (*Double Sideband* ou Banda Lateral Dupla). Note-se, também, que a largura de faixa do sinal modulado é o dobro da largura de faixa do sinal modulante.

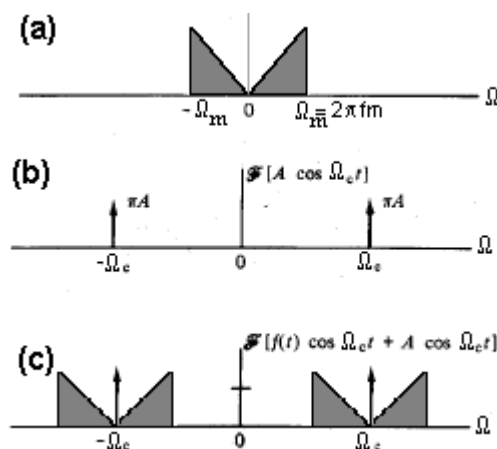


Figura 5.3 – Espectros na modulação AM-DSB com portadora : (a) sinal modulante, (b) portadora e (c) sinal modulado.

A principal vantagem do sinal modulado AM-DSB com portadora é a simplicidade do circuito demodulador que, como mencionado anteriormente, pode fazer a detecção de

envoltória. Tal circuito é utilizado largamente nos receptores de rádio AM. Com isso, pode-se eliminar a principal desvantagem da modulação AM-DSB sem portadora que é a exigência da detecção síncrona e a geração de uma portadora local no circuito receptor. O esquema básico de um detector retificador, usado para detectar a envoltória de um sinal AM-DSB com portadora, é mostrado na Figura 5.4 [17]. Quando um sinal AM é aplicado a um diodo e a um circuito resistivo conforme mostrado nesta figura, a parte negativa do sinal AM é suprimida. A saída sobre o resistor, $v_R(t)$, é o sinal AM após passar por uma retificação de meia onda. Segundo [17], $v_R(t)$ pode ser visto como a multiplicação do sinal AM por um sinal $w(t)$ conforme abaixo:

$$\begin{aligned}
 v_R &= \Phi_{AM}(t)w(t) \\
 &= [A + f(t)]\cos\Omega_c t \left[\frac{1}{2} + \frac{2}{\pi} \left(\cos\Omega_c t - \frac{1}{3}\cos3\Omega_c t + \frac{1}{5}\cos5\Omega_c t - \dots \right) \right] \\
 &= \frac{1}{\pi} [A + f(t)] + \text{outros termos de frequências superiores} \quad (5.8)
 \end{aligned}$$

Quando $v_R(t)$ é aplicada à entrada de um filtro passa-baixas com frequência de corte f_m , sendo f_m a máxima frequência presente no sinal modulante, a saída do filtro é dada por $\frac{1}{\pi} [A + f(t)]$, pois os termos com frequências superiores são suprimidos. Posteriormente, o termo DC, dado por A/π , é suprimido pelo capacitor (veja Figura 5.4) e o sinal modulante $f(t)$ é recuperado.

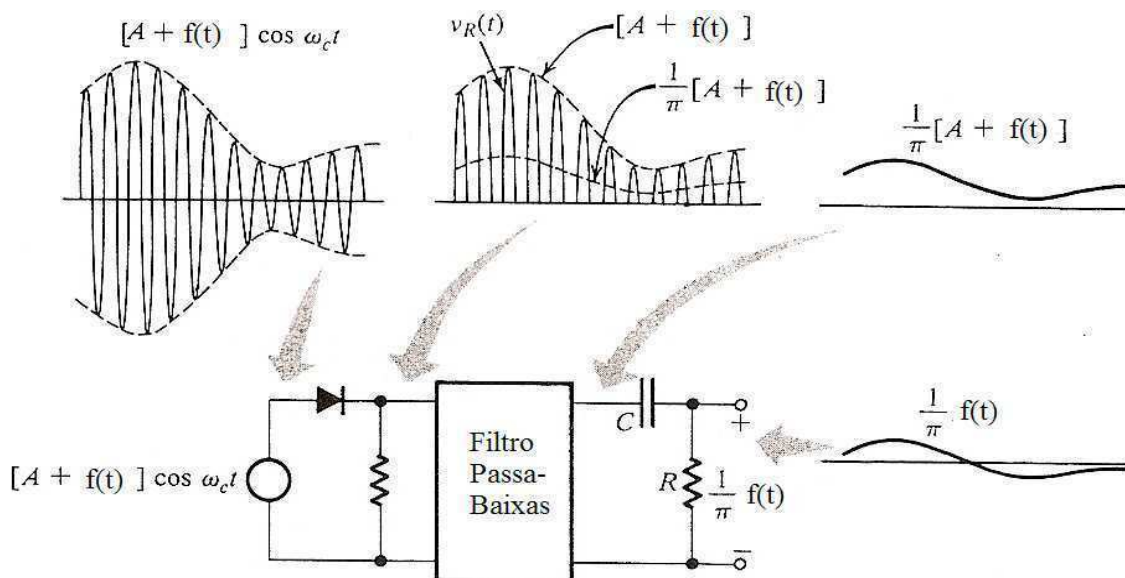


Figura 5.4 – Detector retificador para AM.

5.1.2 A Modulação e a Demodulação AM-SSB

Como observado na modulação AM com banda lateral dupla, o espectro do sinal modulado é composto de duas bandas laterais: a USB (superior) e a LSB (inferior). Como o sinal modulante é sempre real, o módulo de sua transformada será sempre uma função par, o que significa que ambas as faixas laterais possuem a mesma informação. Portanto, ao invés de se transmitir todo o espectro do sinal DSB, é suficiente transmitir apenas uma das faixas laterais (superior ou inferior) do espectro, sem perda de informação, permitindo a recuperação do sinal modulante na recepção. Este conceito deu origem à modulação AM-SSB (AM com banda lateral única).

A Figura 5.5 mostra o diagrama de blocos de um método de geração de sinais SSB, conhecido como método de deslocamento de fase. Nessa figura, $f(t)$ é o sinal modulante, Ω_c é a frequência angular da portadora e $\phi_{SSB}(t)$ é o sinal modulado resultante. Além disso, $H(\Omega)$ é a resposta em frequência de uma rede defasadora de $\pi/2$. Ela defasa de $\pi/2$ a fase do sinal entrante sem alterar o seu módulo. Módulo e fase de $H(\Omega)$ estão mostrados na Figura 5.6.

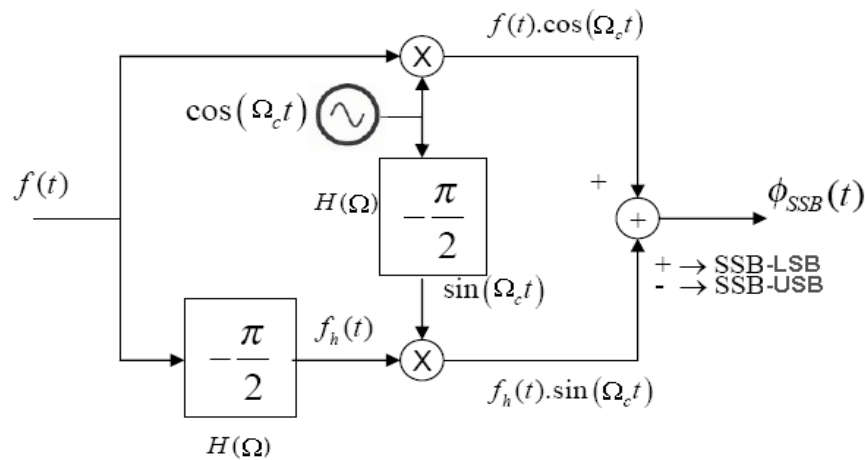


Figura 5.5 – Geração de um sinal SSB pelo método do deslocamento de fase.

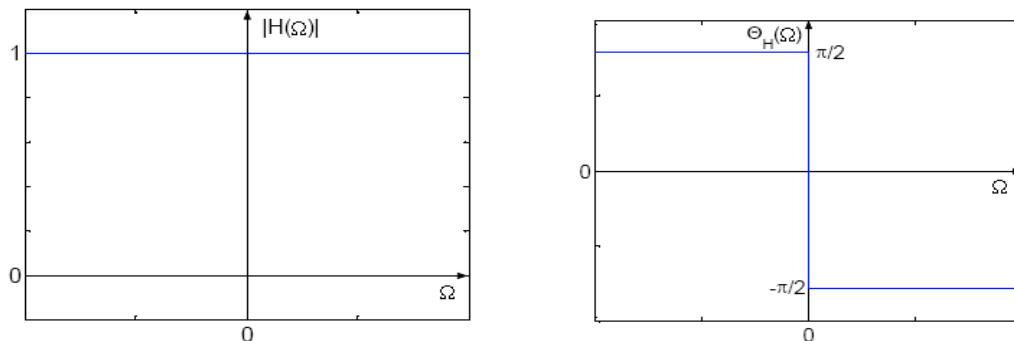


Figura 5.6 – Módulo e fase da resposta em frequência de uma rede defasadora de $\pi/2$.

Pela Figura 5.5, tem-se a seguinte expressão para o sinal modulado SSB:

$$\Phi_{SSB}(t) = f(t)\cos(\Omega_c t) \pm f_h(t)\sin(\Omega_c t), \quad (5.9)$$

onde $f_h(t)$ é o sinal modulante após passar por uma rede defasadora de $\pi/2$.

O sinal de soma na expressão (5.9) é utilizado para a modulação SSB-LSB. Quando, contrariamente, a subtração é utilizada, trata-se da modulação SSB-USB.

Pela Figura 5.6, observa-se que $H(\Omega)$ é definido por:

$$H(\Omega) = e^{-j\frac{\pi}{2}\text{sgn}(\Omega)}, \quad (5.10)$$

onde $\text{sgn}(\Omega)$ é a função *signum* (ou sinal) da frequência angular.

Como $e^{-j\frac{\pi}{2}} = -j$, a equação acima pode ser reescrita como:

$$H(\Omega) = -j\text{sgn}(\Omega) \quad (5.11)$$

A resposta ao impulso $h(t)$ da rede defasadora é dada pela Transformada de Fourier Inversa de $H(\Omega)$, ou seja:

$$h(t) = TF^{-1}\{H(\Omega)\} = TF^{-1}\{-j\text{sgn}(\Omega)\} = \frac{1}{\pi t} \quad (5.12)$$

Por (5.12), constata-se que a rede defasadora é um sistema não-causal já que $h(t)$ não é nulo para $t < 0$. Logo, na prática ela deve ser aproximada por um sistema fisicamente realizável, ou seja, causal. Isto modifica a sua resposta em frequência ideal (Figura 5.6), fazendo com que, por exemplo, seu módulo seja plano somente dentro de alguma faixa de frequências.

A rede defasadora é muitas vezes conhecida como Filtro de Hilbert ou Transformação de Hilbert. Sendo assim, $f_h(t)$ é chamada de Transformada de Hilbert de $f(t)$.

A seguir, será obtida a expressão da Transformada de Fourier de $\Phi_{SSB}(t)$, para a obtenção do espectro do sinal AM-SSB.

O espectro $F_h(\Omega)$ de $f_h(t)$ é dado por:

$$F_h(\Omega) = F(\Omega)H(\Omega) \quad (5.13)$$

Substituindo-se (5.11) em (5.13):

$$F_h(\Omega) = -j\text{sgn}(\Omega)F(\Omega) \quad (5.14)$$

Tomando-se a seta dupla como a representação da Transformada de Fourier, sabe-se que:

$$f(t)\cos(\Omega_c t) \Leftrightarrow \frac{1}{2}[F(\Omega - \Omega_c) + F(\Omega + \Omega_c)] \quad (5.15)$$

E que:

$$f(t)\text{sen}(\Omega_c t) \Leftrightarrow \frac{j}{2}[-F(\Omega - \Omega_c) + F(\Omega + \Omega_c)] \quad (5.16)$$

Logo:

$$f_h(t)\text{sen}(\Omega_c t) \Leftrightarrow \frac{j}{2}[-F_h(\Omega - \Omega_c) + F_h(\Omega + \Omega_c)] \quad (5.17)$$

Utilizando-se (5.14) em (5.17), obtém-se:

$$f_H(t)\text{sen}(\Omega_c t) \Leftrightarrow \frac{j}{2}[j \text{sgn}(\Omega - \Omega_c)F(\Omega - \Omega_c) - j \text{sgn}(\Omega + \Omega_c)F(\Omega + \Omega_c)] \quad (5.18)$$

Reescrevendo-se (5.18):

$$f_H(t)\text{sen}(\Omega_c t) \Leftrightarrow \frac{1}{2}[\text{sgn}(\Omega + \Omega_c)F(\Omega + \Omega_c) - \text{sgn}(\Omega - \Omega_c)F(\Omega - \Omega_c)] \quad (5.19)$$

Finalmente, a expressão da Transformada de Fourier de $\Phi_{SSB}(t)$ é obtida através de (5.9), (5.15) e (5.19) pelo uso da propriedade da linearidade da Transformada de Fourier. Para a banda lateral inferior, $\Phi_{SSB-LSB}(\Omega)$ é dada por:

$$\begin{aligned} \Phi_{SSB-LSB}(\Omega) &= \frac{1}{2}[F(\Omega - \Omega_c) + F(\Omega + \Omega_c)] + \\ &+ \frac{1}{2}[\text{sgn}(\Omega + \Omega_c)F(\Omega + \Omega_c) - \text{sgn}(\Omega - \Omega_c)F(\Omega - \Omega_c)] \end{aligned}$$

A equação acima pode ser reescrita como:

$$\Phi_{SSB-LSB}(\Omega) = \frac{1}{2}\{[1 + \text{sgn}(\Omega + \Omega_c)]F(\Omega + \Omega_c) + [1 - \text{sgn}(\Omega - \Omega_c)]F(\Omega - \Omega_c)\} \quad (5.20)$$

Nota-se que:

$$[1 + \text{sgn}(\Omega + \Omega_c)] = 2u(\Omega + \Omega_c) \quad (5.21)$$

E que:

$$[1 - \text{sgn}(\Omega - \Omega_c)] = 2u(-\Omega + \Omega_c), \quad (5.22)$$

onde $u(\Omega)$ é a função degrau unitário de Ω .

Substituindo (5.21) e (5.22) em (5.20), obtém-se:

$$\Phi_{SSB-LSB}(\Omega) = u(\Omega + \Omega_c)F(\Omega + \Omega_c) + u(-\Omega + \Omega_c)F(\Omega - \Omega_c) \quad (5.23)$$

A expressão acima representa o espectro do sinal SSB-LSB. Ele está ilustrado na Figura 5.7-b. A expressão do espectro do sinal SSB-USB pode ser obtida de uma forma análoga, resultando em:

$$\Phi_{SSB-USB}(\Omega) = u(-\Omega - \Omega_c)F(\Omega + \Omega_c) + u(\Omega - \Omega_c)F(\Omega - \Omega_c) \quad (5.24)$$

A Figura 5.7-c ilustra o espectro do sinal SSB-USB.

Quanto à demodulação de um sinal SSB, esta pode ser feita de maneira coerente (ou síncrona) conforme ilustrado na Figura 5.8 [17]. Isto pode ser verificado matematicamente. Tomando-se a expressão (5.9) do sinal SSB e multiplicando-se pela portadora, obtém-se:

$$\Phi_{SSB}(t)\cos(\Omega_c t) = \frac{1}{2}f(t)[1 + \cos(2\Omega_c t)] \pm \frac{1}{2}f_h(t)\sin(2\Omega_c t) \quad (5.25)$$

Rearrmando:

$$\Phi_{SSB}(t)\cos(\Omega_c t) = \frac{1}{2}f(t) + \frac{1}{2}[\cos(2\Omega_c t) \pm f_h(t)\sin(2\Omega_c t)] \quad (5.26)$$

Portanto, o produto do sinal SSB pela portadora produz o espectro do sinal modulante somado ao espectro de um outro sinal SSB com portadora $2\Omega_c$, conforme ilustrado na Figura 5.9. Após esta multiplicação, o sinal passa por um filtro passa-baixas (veja Figura 5.8) para obter de (5.26) apenas o termo desejado, ou seja, $\frac{1}{2}f(t)$.

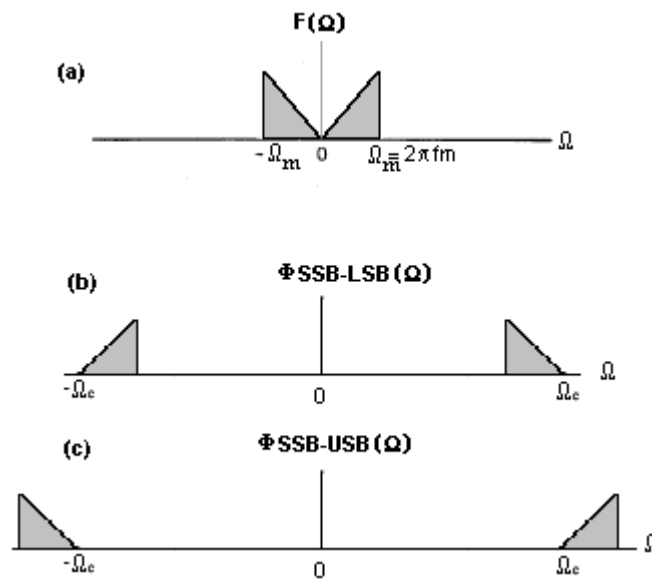


Figura 5.7- (a) Espectro do sinal modulante; (b) Espectro do sinal SSB com banda lateral inferior; (b) Espectro do sinal SSB com banda lateral superior.

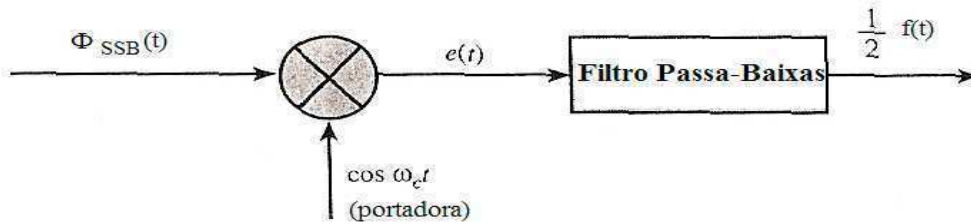


Figura 5.8 – Demodulação SSB coerente por multiplicação pela portadora seguida de filtragem.

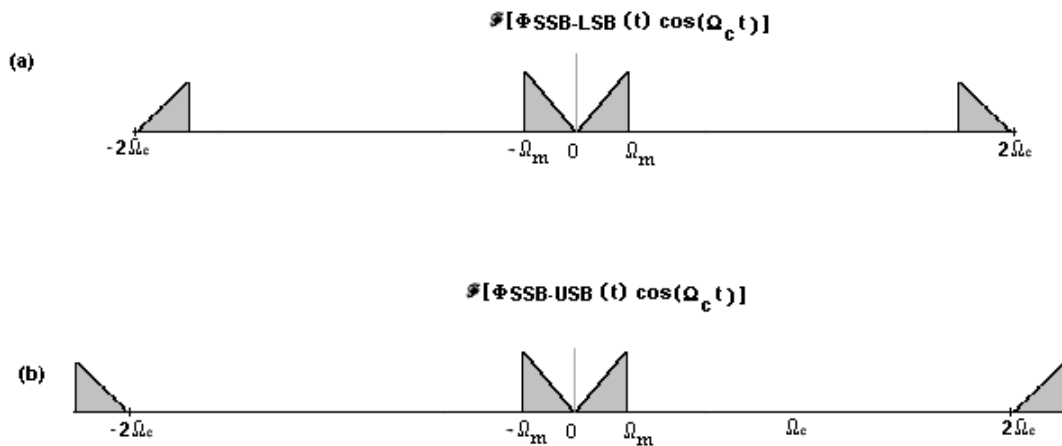


Figura 5.9 - (a) Espectro do produto do sinal SSB-LSB pela portadora; (b) Espectro do produto do sinal SSB-USB pela portadora.

Este tipo de demodulação síncrona ou coerente, contudo, exige a geração de uma portadora local na recepção em sincronismo de fase e frequência em relação à portadora usada na modulação. Entretanto, isto torna o receptor mais complexo e mais caro. Pequenas diferenças na fase e/ou na frequência podem introduzir distorções no sinal detectado. Segundo [17], os efeitos causados por tais diferenças são bem mais graves na demodulação AM-DSB-SC (AM com Banda Lateral Dupla sem Portadora) do que na demodulação SSB. Para o caso em que o sinal modulante é um sinal de áudio, esses efeitos não têm importância na demodulação SSB, uma vez que a inteligibilidade do áudio independe da fase das senóides componentes.

Outra maneira de se demodular sinais AM-SSB está ilustrada na Figura 5.10, que mostra um processo praticamente inverso ao ilustrado na Figura 5.5. Nessa figura, o sinal demodulado é obtido a partir da soma (para a demodulação SSB-USB) ou subtração (para a demodulação SSB-LSB) de dois sinais $q(t)$ e $i_h(t)$:

$$f(t) = q(t) \mp i_h(t), \quad (5.27)$$

onde $q(t)$ é dado por:

$$q(t) = \Phi_{SSB}(t) \cos(\Omega_c t), \quad (5.28)$$

e $i_h(t)$ é a Transformada de Hilbert do sinal $i(t)$ dado por:

$$i(t) = \Phi_{SSB}(t) \sin(\Omega_c t) \quad (5.29)$$

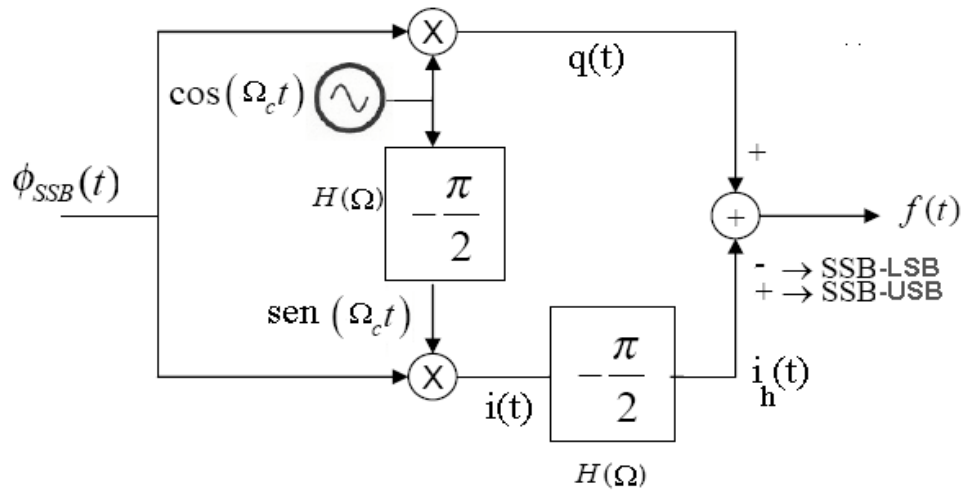


Figura 5.10 – Método de demodulação de sinais AM-SSB.

Esta maneira de se demodular sinais SSB pode ser comprovada por uma simples análise espectral que inclui módulo e fase, conforme ilustrado na Figura 5.11. Tal figura ilustra a demodulação de um sinal SSB-USB. Observa-se que, por simplicidade, a fase do sinal modulante não é considerada, o que não invalida a análise. A Figura 5.11-a mostra o espectro de um sinal SSB-USB. A Figura 5.11-b ilustra o espectro de $q(t)$, chamado de $Q(\Omega)$. Nota-se que $Q(\Omega)$ segue (5.15) com $f(t) = \Phi_{SSB-USB}(t)$. Já a Figura 5.11-c ilustra o espectro de $i(t)$, $I(\Omega)$. Observa-se que ele segue (5.16) com $f(t) = \Phi_{SSB-USB}(t)$. A Figura 5.11-d mostra o espectro de $i_h(t)$, ou seja, $I_h(\Omega)$, que nada mais é que $I(\Omega)$ defasado de $\pi/2$. Finalmente, a Figura 5.11-e mostra o espectro de $q(t) + i_h(t)$. Nota-se que a diferença de fase de π entre $Q(\Omega)$ e $I_h(\Omega)$ para $\Omega > 2\Omega_c$ e $\Omega < -2\Omega_c$, faz com que tais frequências desapareçam com a soma, restando apenas a faixa de frequências de interesse. Por isso, o espectro de $q(t) + i_h(t)$ é o espectro do sinal modulante, resultado da demodulação. Uma análise semelhante poderia ser feita para um sinal SSB-LSB, comprovando a eficácia deste método de demodulação.

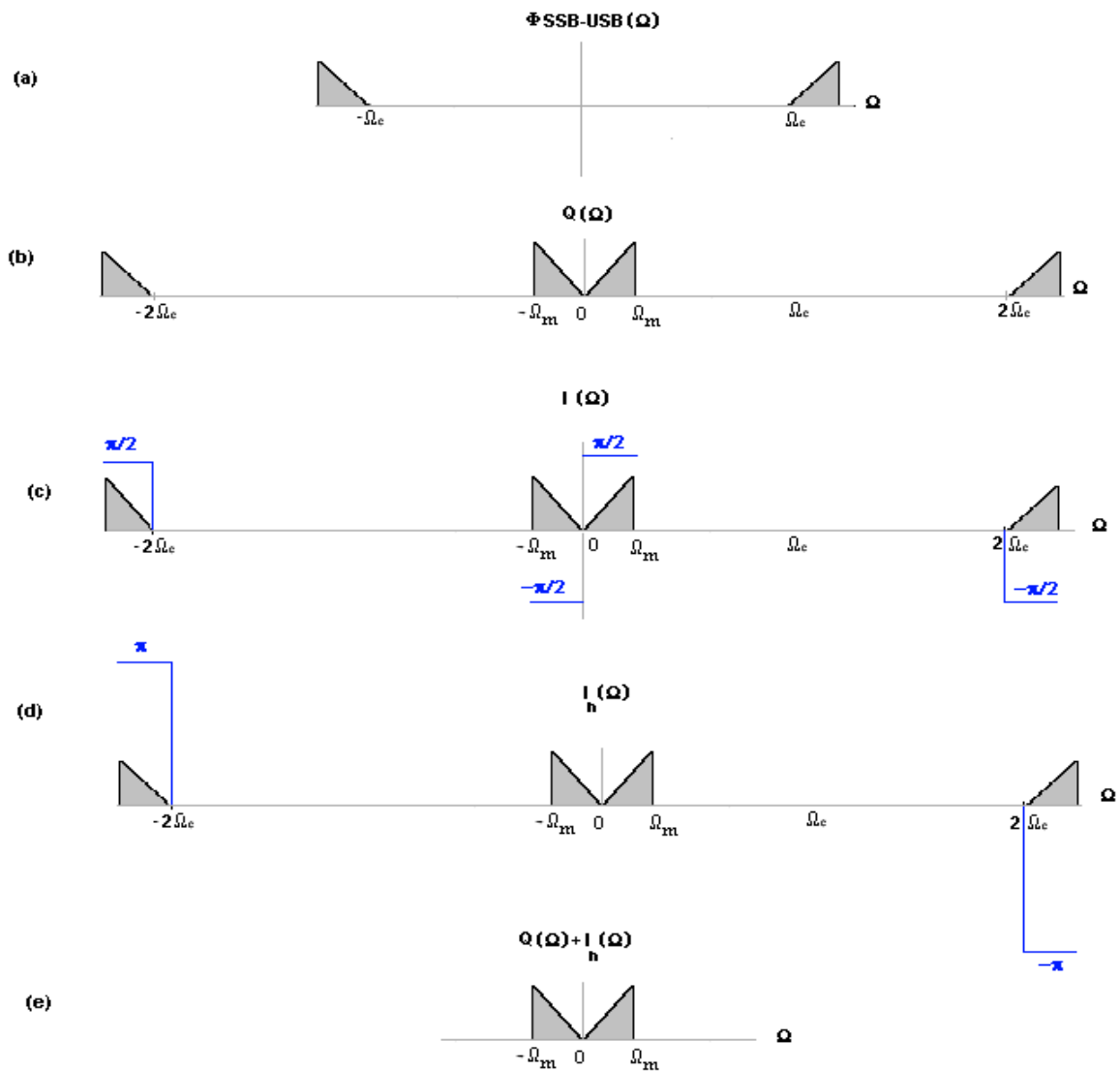


Figura 5.11 – Análise espectral da demodulação AM-SSB da Figura 5.10.

5.2 A Recepção AM Usando o Conceito de Rádio de Software

Nesta seção, será apresentado o esquema da recepção AM para os dois tipos de demodulação descritos na seção anterior (AM-DSB com portadora e AM-SSB), porém com o uso do conceito de rádio de *software*.

A recepção AM, seguindo o conceito de rádio de *software*, é mostrada na Figura 5.12.

Na cadeia de recepção, o sinal é recebido pela antena, filtrado e amplificado em RF, sofre o processo de *down-conversion* e é digitalizado em FI (ou seja, sofre a conversão A/D). Em seguida, o sinal é demodulado por um DSP programado com instruções específicas necessárias à demodulação utilizada. A partir de então, o sinal sofre a conversão D/A, é filtrado e amplificado para, depois, alimentar um transdutor, como um alto-falante. Portanto, a aplicação-exemplo apresentada neste trabalho segue a arquitetura de rádio de *software* com digitalização em FI, sendo o bloco da demodulação implementado em *software* a ser executado em um DSP. Ela permite, então, flexibilidade quanto à demodulação a se utilizar na recepção, bastando que o *software* apropriado seja carregado no DSP. A Seção 5.2.1 descreve como pode ser implementada digitalmente a demodulação AM-DSB com portadora e a Seção 5.2.2 como pode ser implementada digitalmente a demodulação AM-SSB.

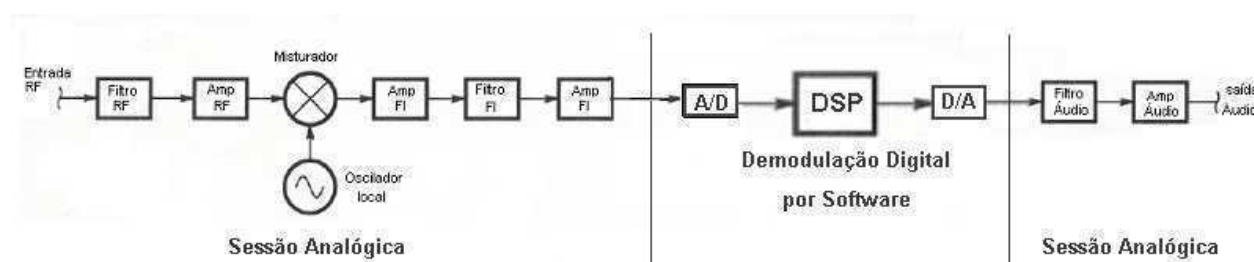


Figura 5.12 – Proposta de recepção AM seguindo o conceito de rádio de *software*.

5.2.1 Demodulação AM-DSB com Portadora em um Rádio de *Software*

A presente seção propõe que a implementação em *software* da demodulação AM-DSB com portadora siga o esquema da Figura 5.13, onde cada bloco não representa um bloco físico, mas um bloco funcional que deve ser implementado em *software* e executado em um DSP.

Antes da conversão A/D, está presente um canal do sinal AM-DSB com portadora, cujo espectro está centrado em FI, uma vez que ele já foi translado pelo processo de conversão para baixo. No ponto (A) da Figura 5.13, este sinal já foi digitalizado e é chamado de $x[n]$. A partir de então, blocos de amostras de $x[n]$ de tamanho M passam por todos os blocos funcionais responsáveis pela demodulação. A implementação da demodulação em *software* deve ser suficientemente eficaz e o DSP utilizado suficientemente rápido para que as M amostras de $x[n]$ já processadas formem uma entrada para a conversão D/A antes que as

próximas M amostras cheguem para serem processadas. O primeiro bloco funcional pelo qual passam as M amostras de $x[n]$ é o bloco que calcula o módulo de $x[n]$. Neste bloco funcional, cada amostra tem seu valor absoluto calculado. Fazendo-se a analogia com o circuito da Figura 5.4, ou seja, com o circuito analógico normalmente utilizado para demodular sinais AM-DSB com portadora, este módulo está realizando a função do diodo retificador. Posteriormente, o sinal digital no ponto (B) da Figura 5.13 passa por um filtro FIR passa-baixas para que se selecionem as componentes de frequência em torno de $\omega=0$ e se recupere o espectro do sinal modulante a menos da componente DC introduzida, que depende do índice de modulação. O projeto desse filtro digital passa-baixas, ou seja, a determinação de sua resposta ao impulso $h[n]$, segue o processo de janelamento explicado no Capítulo 4. O processo de filtragem envolve a convolução de $\text{abs}\{x[n]\}$ com $h[n]$, compreendendo operações MAC típicas da Equação (4.36). Após a filtragem, o sinal no ponto (C) da Figura 5.13, já demodulado digitalmente, passa pela conversão D/A e segue pela cadeia analógica restante (vide Figura 5.12). Maiores detalhes sobre esta demodulação em *software* e o filtro FIR passa-baixas utilizado, serão vistos na Seção 5.3.1 que abordará uma simulação computacional no *Scilab* desta demodulação AM-DSB com portadora incluindo a análise espectral em diversos pontos.

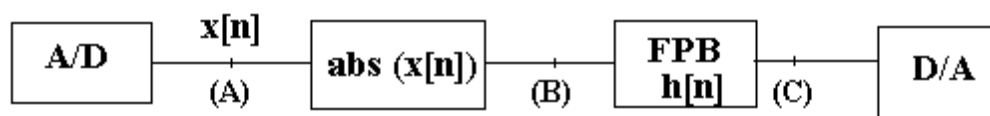


Figura 5.13 – Blocos funcionais da demodulação AM-DSB com portadora em *software*.

5.2.2 Demodulação AM-SSB em um Rádio de *Software*

A presente seção pretende apresentar um esquema de demodulação SSB que pode ser utilizado em um rádio definido por *software*. A simulação computacional desta demodulação é apresentada na Seção 5.3.2 e comprova a eficácia deste esquema. A Figura 5.14 mostra o diagrama do esquema de demodulação e contém as funções que devem ser executadas digitalmente no DSP da Figura 5.12.

No ponto (X) da Figura 5.14 está presente um canal do sinal AM-SSB, cuja expressão

é dada por (5.9), com $\Omega_c = 2\pi FI$, uma vez que o espectro já foi transladado pelo processo de conversão para baixo. Na Figura 5.14, este sinal, após ser digitalizado pelo conversor A/D, é chamado de $s[n]$. Então, é tomada uma certa quantidade de amostras de $s[n]$. Cada uma destas amostras é multiplicada pelas amostras correspondentes de um sinal cossenoidal discreto de frequência angular ω_c que, assim como $s[n]$, só possui valores não-nulos em múltiplos do período de amostragem. Esta multiplicação origina o sinal $q[n]$. Observe-se que, ao contrário do sinal cossenoidal empregado na demodulação analógica (veja Figura 5.10), gerado por um oscilador, tal sinal cossenoidal discreto é gerado matematicamente no DSP. Além disso, ele não advém de um sinal analógico amostrado. Ao mesmo tempo em que o sinal $q[n]$ é calculado, também é calculado o sinal $i[n]$ pela multiplicação de cada uma das amostras tomadas de $s[n]$ pelas amostras correspondentes de um sinal senoidal discreto de frequência angular ω_c que só possui valores não-nulos em múltiplos do período de amostragem. Este sinal senoidal discreto também é gerado por *software* no DSP e não provém de um oscilador real. Posteriormente, o sinal $i[n]$ passa pela Transformação de Hilbert, tendo seu espectro em fase defasado de $\pi/2$. Conforme visto anteriormente, o filtro ideal de Hilbert não é causal e aproximações devem ser utilizadas para a implementação do mesmo. Normalmente, a implementação digital desse filtro prevê a aproximação por um filtro FIR linear. No DSP, os coeficientes deste filtro, já previamente calculados, ficam armazenados. Ao passar pelo filtro de Hilbert, as amostras de $i[n]$ e os coeficientes do filtro passam pela operação de convolução, realizando-se operações MAC típicas da Equação (4.36), o que origina o sinal $i_h[n]$. Este sinal é, então, somado a $q[n]$ (se a demodulação é SSB-USB) ou subtraído de $q[n]$ (se a demodulação é SSB-LSB), obtendo-se o sinal demodulado, ainda que discreto, no ponto (M) da Figura 5.14. O sinal no ponto (M) passa, então, pela conversão D/A e continua passando pela cadeia analógica restante (veja Figura 5.12). Maiores detalhes sobre esta demodulação em *software* e o filtro de Hilbert utilizado serão vistos na Seção 5.3.2, que abordará uma simulação computacional desta demodulação SSB no *Scilab*, incluindo a análise espectral em diversos pontos.

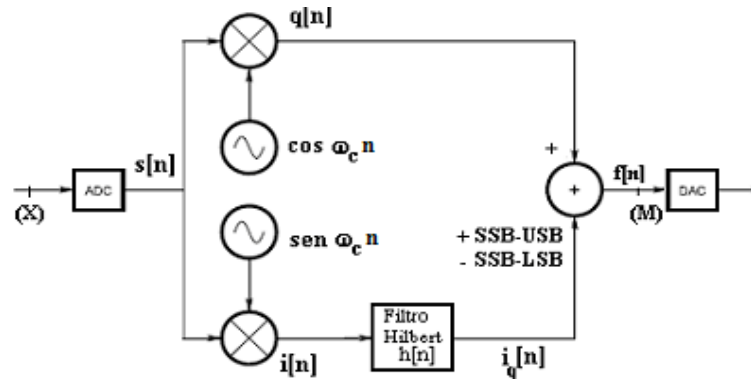


Figura 5.14 – Esquema da demodulação AM-SSB em *software*.

5.3 Simulações Computacionais da Recepção AM Usando o Conceito de Rádio de Software

5.3.1 Simulação da Demodulação AM-DSB com Portadora: Resultados e Análise

Para a simulação no *Scilab* da demodulação AM-DSB com portadora (esquema da Figura 5.13), foi suposto o uso de um sinal modulante de teste de 2kHz, o qual se encontra dentro da faixa do áudio geralmente utilizada para modular sinais AM comerciais de radiodifusão. Esta faixa se estende de 0 a 5 kHz (sendo as componentes em 0 e 5kHz praticamente desprezíveis) gerando uma faixa de sinal transmitido de 10 kHz em torno da portadora de cada canal. No momento anterior ao da demodulação, independente do canal selecionado, seu espectro fica centralizado em torno da Frequência Intermediária, que tem valor de 455 kHz. A Figura 5.15-a ilustra a Transformada de Fourier do sinal modulante de 2kHz, ou seja, seu espectro. Já na Figura 5.15-b, está ilustrado o formato do espectro esperado para $\Phi_{AM}(t)$ (sinal recebido antes de ser digitalizado e demodulado). Observa-se que $\Phi_{AM}(\Omega)$ segue a Equação (5.7), com $\Omega_c = 2\pi FI$, já que o espectro já foi transladado para FI.

Na simulação, $\Phi_{AM}(t)$ segue a fórmula (5.2), com $A=1,2$ (índice de modulação de cerca de 83,33%), $f_c=FI=455000$ e $f(t)=\cos(2\pi 2000t)$:

$$\Phi_{AM}(t) = 1,2\cos(2\pi 455000t) + \cos(2\pi 2000t)(2\pi 455000t)$$

$$= 1,2\cos(2\pi 455000t) + 0,5[\cos(2\pi 453000t) + \cos(2\pi 457000t)]$$

A frequência de amostragem escolhida, $f_s=1\text{MHz}$, obedece ao Teorema da Amostragem e é mais que duas vezes maior que a maior componente de frequência possível de estar presente no canal selecionado, ou seja, $f_s > 2 \times (455 \text{ kHz} + 5\text{kHz})$.

O sinal a demodular já digitalizado, ou seja, o sinal $x[n]$ da Figura 5.13, deve possuir um espectro totalmente compatível com sua DTFT, cuja teoria foi apresentada na Seção 4.2.1. Neste caso, a Transformada de Fourier de $x[n]$, $X_s(j\Omega)$, é formada por réplicas da Transformada de Fourier de $\Phi_{AM}(t)$ em torno de múltiplos da frequência de amostragem, kf_s , com k inteiro (veja Figura 5.15-c). Além disso, a DTFT de $x[n]$ pode ser facilmente obtida a partir de $X_s(j\Omega)$ através de um simples reescalonamento no eixo da frequência fazendo-se $\omega = \Omega T_s$, onde $T_s = 1/f_s = 10^{-6}$. Sendo assim, a DTFT de $x[n]$ deve ter o aspecto mostrado na Figura 5.15-d.

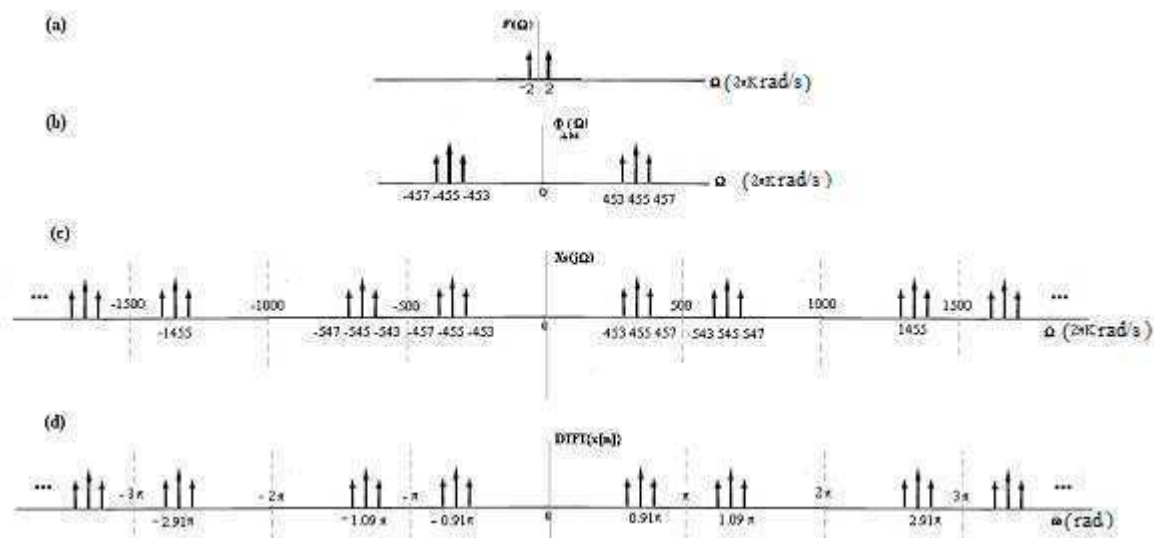


Figura 5.15 – (a) Transformada de Fourier de $f(t)$; (b) Transformada de Fourier de $\Phi_{AM}(t)$; (c) Transformada de Fourier de $x[n]$; (d) DTFT de $x[n]$.

O sinal $x[n]$ foi produzido matematicamente no *Scilab* (vide Apêndice A.1 - Simulação da Demodulação AM-DSB com Portadora). Em seguida, a simulação incluiu o cálculo da FFT de $x[n]$ e a plotagem de seu Espectro de Magnitude (veja Figura 5.16). Essa plotagem foi feita para ω no intervalo de $[0, 2\pi]$. Para uma melhor visualização em termos de frequência, o eixo ω foi multiplicado pelo fator $10^6/(2\pi)$. Observe a total compatibilidade entre o espectro previsto pela teoria (Figura 5.15) e o mostrado na Figura 5.16, já que a FFT é

apenas um método mais eficaz de se terminar a DFT e a DFT é composta por amostras da DTFT. Repare que a assimetria dos impulsos vista na Figura 5.16-b e c está relacionada ao número finito de pontos do eixo horizontal. Além disso, nas simulações, todos os gráficos que incluem a plotagem da FFT de algum sinal, aparecem, para uma melhor visualização, como funções contínuas (os pontos discretos são ligados), apesar de serem discretos na frequência.

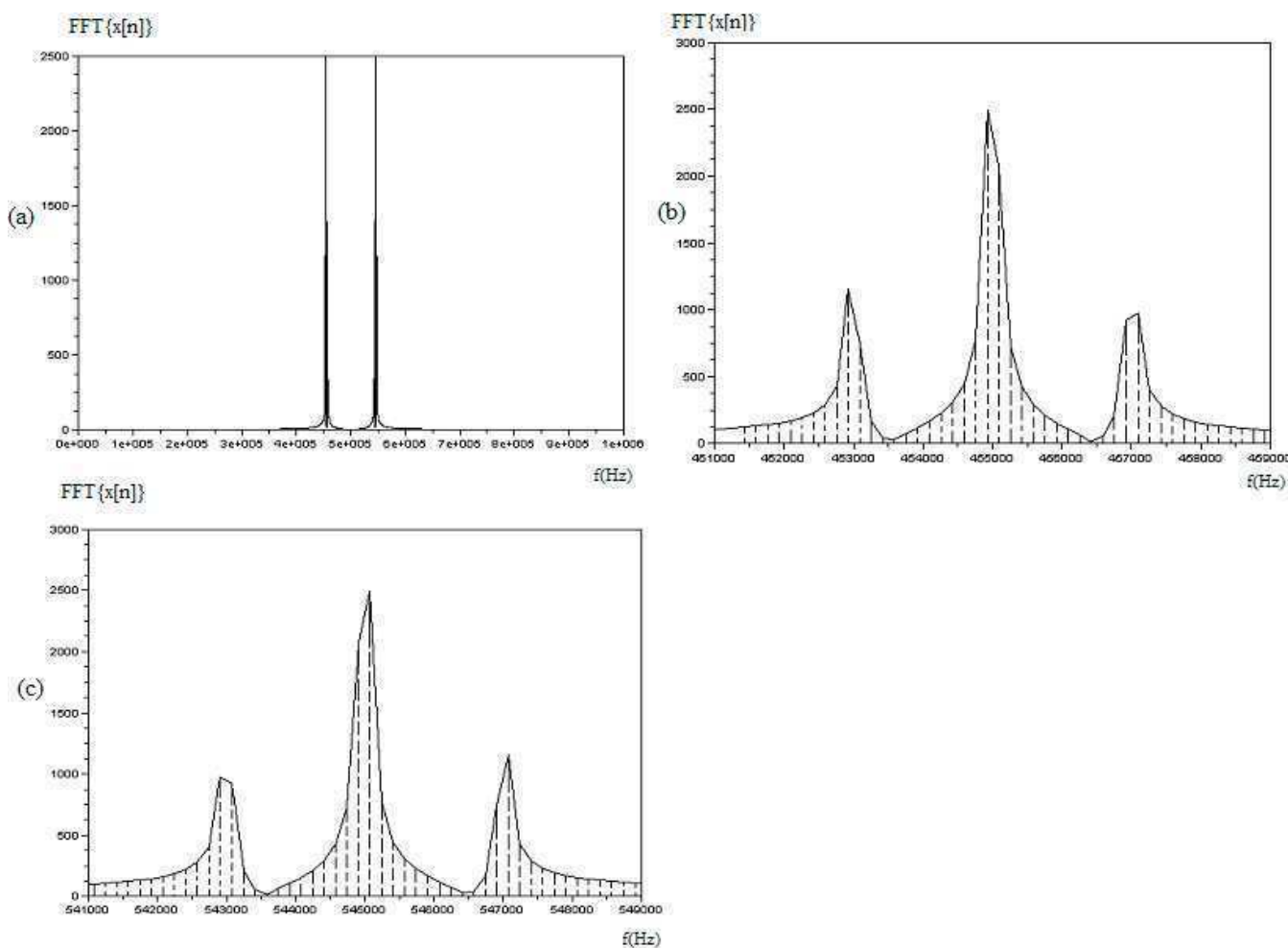


Figura 5.16 – (a) Espectro de $x[n]$ no Scilab; (b) zoom em torno de 455kHz; (c) zoom em torno de 545 kHz.

A primeira etapa da demodulação de $x[n]$ simulada no *Scilab* é o cálculo de seu módulo, conforme ilustrado na Figura 5.13. Após efetuar-se este cálculo, o sinal digital está no ponto (B) da Figura 5.13 e tem seu espectro plotado, de 0 a 1MHz, pelo *Scilab*, através do uso da FFT (vide Apêndice A.1 - Simulação da Demodulação AM-DSB com Portadora). Este espectro está mostrado na Figura 5.17. Observa-se a presença de um componente DC de grande amplitude, de um componente em 2kHz e o surgimento de vários componentes espúrios que devem ser removidos pelos passos seguintes da demodulação. Por razões de escala, não se distingue o componente em 2kHz em (a), mas é claramente visível, junto com o

componente DC, em (b); em (c), são visualizados componentes em 88, 90 e 92 kHz que são os espúrios de mais baixa frequência que estão acima de 5kHz.

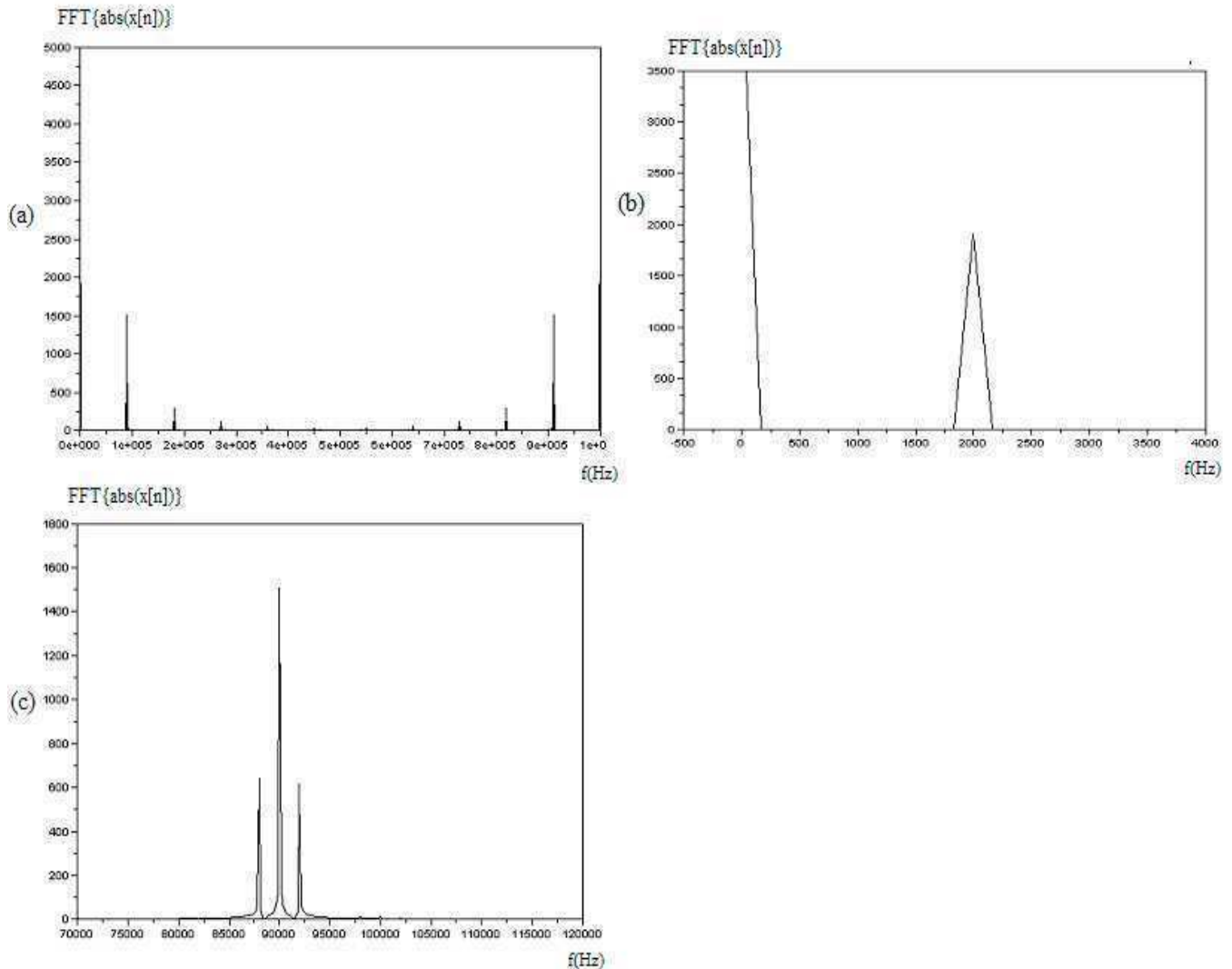


Figura 5.17 – (a) Espectro do sinal no ponto (B) da Figura 5.13; (b) zoom em torno de 0 Hz; (c) zoom em torno de 90 kHz.

A etapa seguinte da demodulação é a filtragem passa-baixas, para eliminar frequências altas, deixando passar todas as componentes de frequência até 5kHz (a maior frequência possível do sinal AM a detectar). Neste caso, após ser filtrado, o sinal deverá estar bastante atenuado nas frequências acima de 85kHz (veja Figura 5.17-c). A Figura 5.18 mostra as especificações do filtro projetado com suas bandas passantes, de transição e rejeição. Observe que a maior frequência angular digital da banda passante é dada por $\omega_p = 2\pi 5000 \times 10^{-6} = 0,01\pi$ e a banda de rejeição começa em $\omega_s = 2\pi 85000 \times 10^{-6} = 0,17\pi$. Sendo assim, o tamanho da banda de transição é dado por:

$\Delta\omega = \omega_s - \omega_p = 0,17\pi - 0,01\pi = 0,16\pi$. Também foi estabelecido que a atenuação mínima na banda de rejeição seria de 60 dB. A janela escolhida para o cálculo dos coeficientes deste filtro foi a janela de *Kaiser*. Sendo assim, a escolha dos parâmetros N (ordem do filtro que influi na banda de transição) e β (parâmetro que regula a atenuação na banda de rejeição) seguem as fórmulas (89) e (88). Adotando-se $\Delta\omega = 0,16\pi$ e $A_{\min}=60$ dB em (89) obtém-se $N \approx 45,3$. Este valor é arredondado para $N=46$. Assumindo-se $A_{\min}=60$ em (88) obtém-se $\beta \approx 5,65$

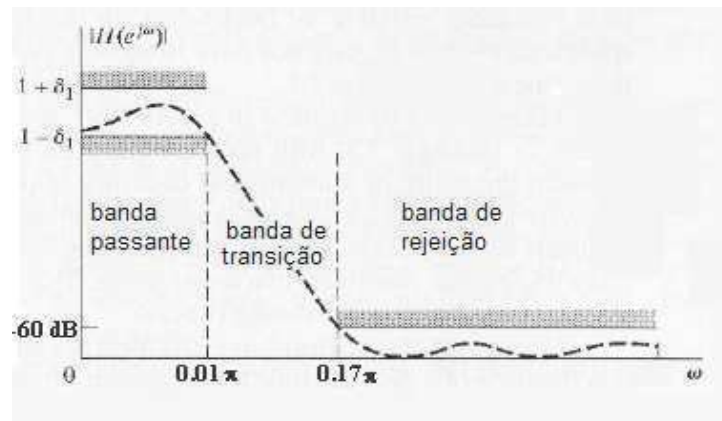


Figura 5.18 – Especificações do filtro FIR Passa-Baixas projetado para a simulação da detecção AM-DSB com portadora.

O *Scilab* calculou os coeficientes deste filtro com base nos seguintes parâmetros fornecidos:

- A ordem do filtro $N=46$
- O parâmetro $\beta = 5,65$
- A frequência de corte $\omega_c = 2\pi \cdot 40000 \times 10^{-6} = 0,08\pi$ que foi escolhida entre ω_p e ω_s , mais ou menos no meio da banda de transição.

No *Scilab*, também foi plotada a função ganho deste filtro, ou seja, o módulo de sua resposta em frequência dado em decibéis (veja Figura 5.19). Observe que o eixo da frequência angular digital foi normalizado para 2π radianos. Observe também que a resposta é bem plana na faixa passante (de 0 a 0,005, o que corresponde à faixa de 0 a 5000Hz) e que a banda de rejeição (a partir de 0,085, ou seja, a partir de 85kHz) tem pelo menos 60 dB de atenuação.

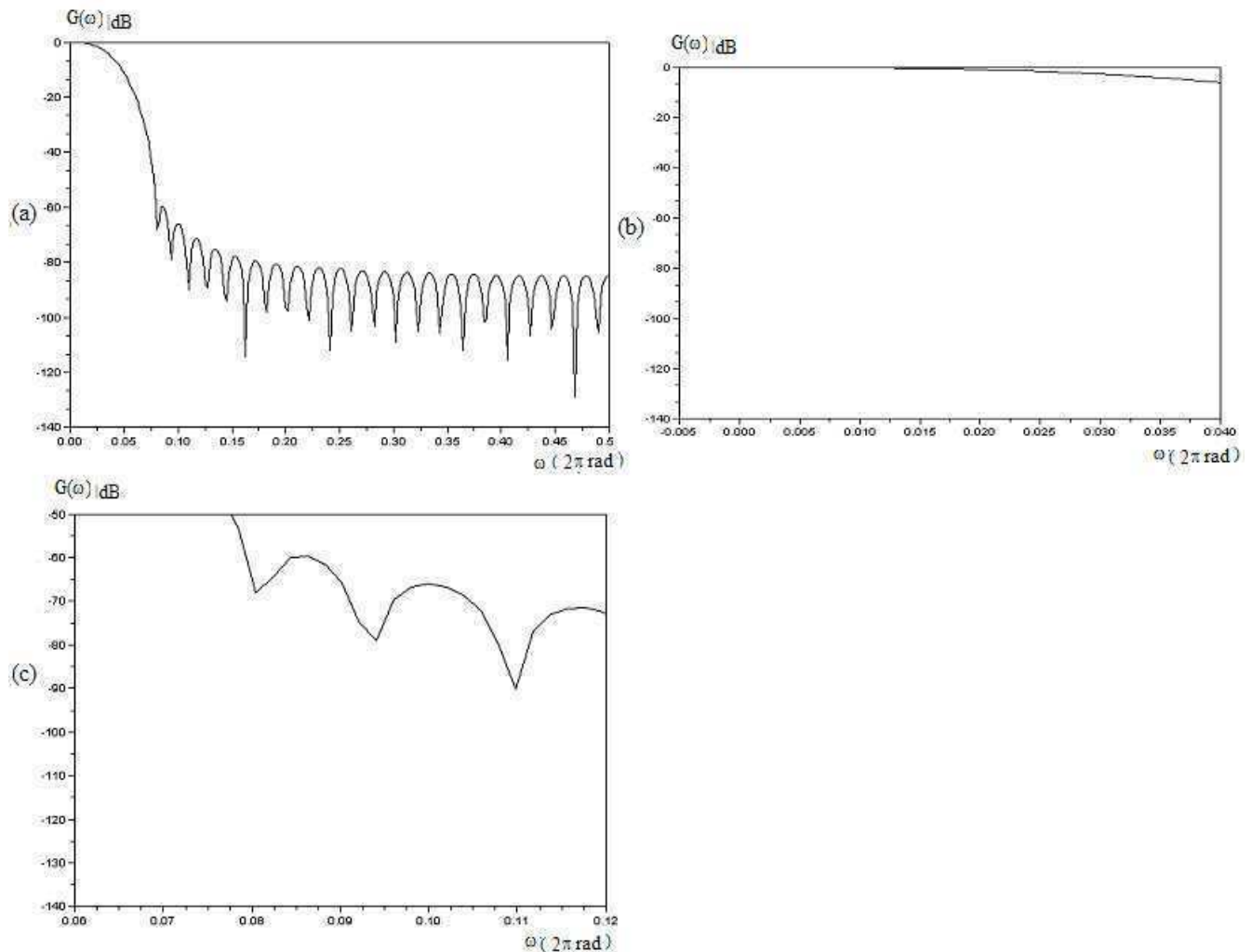


Figura 5.19 – (a) Função Ganho do filtro FIR Passa-Baixas projetado para a simulação da detecção AM-DSB com portadora; (b) zoom em torno de $\omega_p = 0,005 (2\pi \text{ rad})$; (c) zoom em torno de $\omega_s = 0,085 (2\pi \text{ rad})$.

Após o projeto do filtro no *Scilab*, foi feita a filtragem propriamente dita, ou seja, foi calculada a convolução do sinal no ponto (B) da Figura 5.13 com os coeficientes $h[n]$ do filtro passa-baixas (vide Apêndice A.1 - Simulação da Demodulação AM-DSB com Portadora) e o espectro do sinal resultante foi plotado (espectro do sinal no ponto C da Figura 5.13). O resultado está ilustrado na Figura 5.20. Observa-se que, além da componente de frequência desejada de 2kHz, aparece ainda uma grande componente DC advinda da inserção da portadora na modulação, porém as demais componentes de frequência já foram atenuadas.

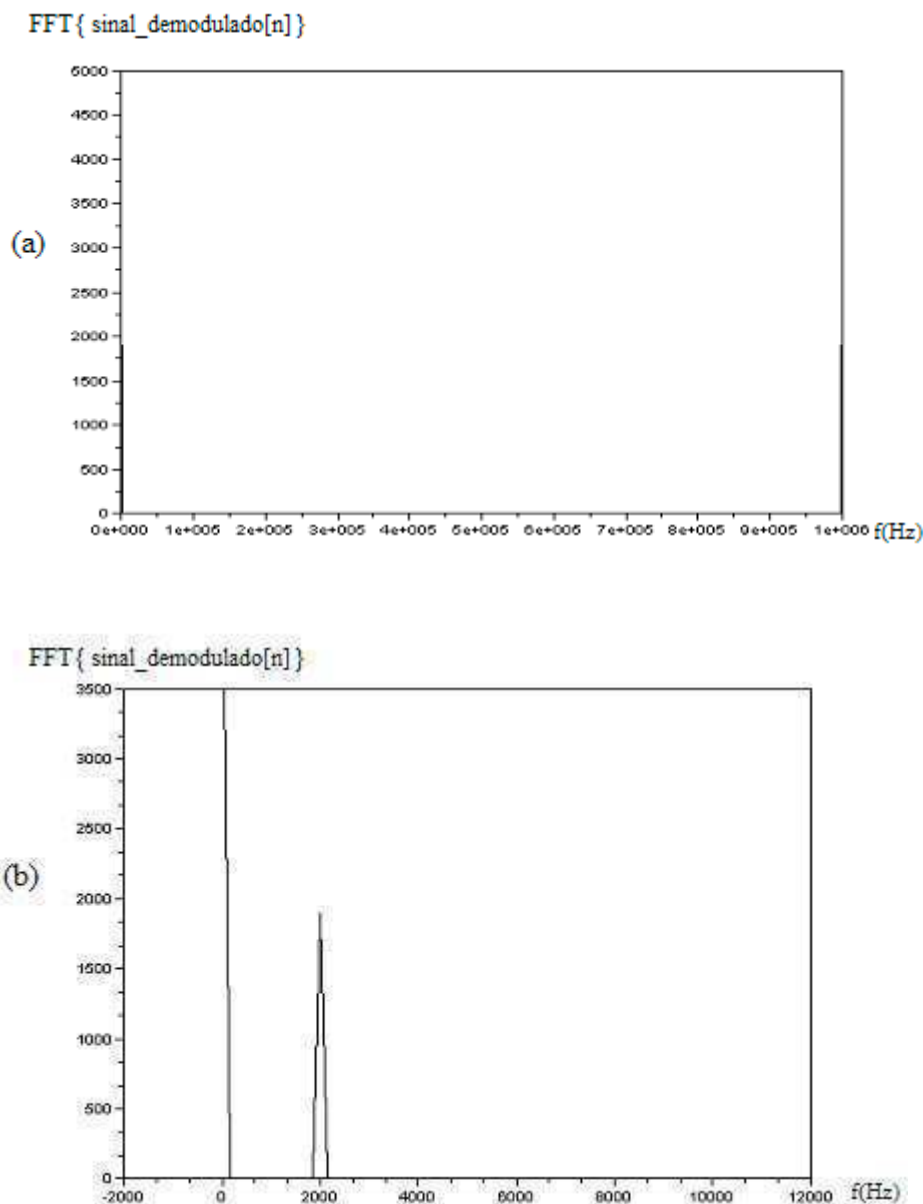


Figura 5.20 – (a) Espectro do sinal no ponto (C) da Figura 5.13, ou seja, após passar pela filtragem; (b) zoom em torno de 2kHz.

Ao alcançar o ponto (C) da Figura 5.13, o sinal já se encontra demodulado digitalmente a menos de uma constante (nível DC introduzido pela inserção da portadora na demodulação). Como ele ainda é um sinal digital, na cadeia de recepção AM ele deve ainda passar pela conversão D/A antes de ter seu nível DC removido e alimentar um transdutor analógico (veja Figura 5.12). No domínio do tempo, este sinal demodulado digitalmente pode ser visto na Figura 5.21. Repare que, no gráfico, ele aparece, para uma melhor visualização, como uma função contínua, tendo seus pontos discretos no tempo ligados. Ele possui as seguintes características: forma cossenoidal com amplitude pico a pico $A_{pp}=1,275$, valor

médio=0,7625 e período=0,0005 (frequência de 2 kHz). Tais características confirmam o sucesso da demodulação digital na comparação com o sinal modulante (original): cosseno com frequência de 2 kHz, valor médio de 0 V e $A_{pp}=2V$. Apesar da diferença encontrada na amplitude pico a pico, isto não tem muita importância já que simulações com sinais modulantes cossenoidais de outras frequências revelaram atenuação semelhante, ou seja, a demodulação não está introduzindo distorção dentro da banda do canal recebido. Atenuação mais acentuada só pode ser sentida para sinais modulantes com frequência bem próxima a 0 Hz, porém, como anteriormente visto, estas frequências geralmente não estão presentes.

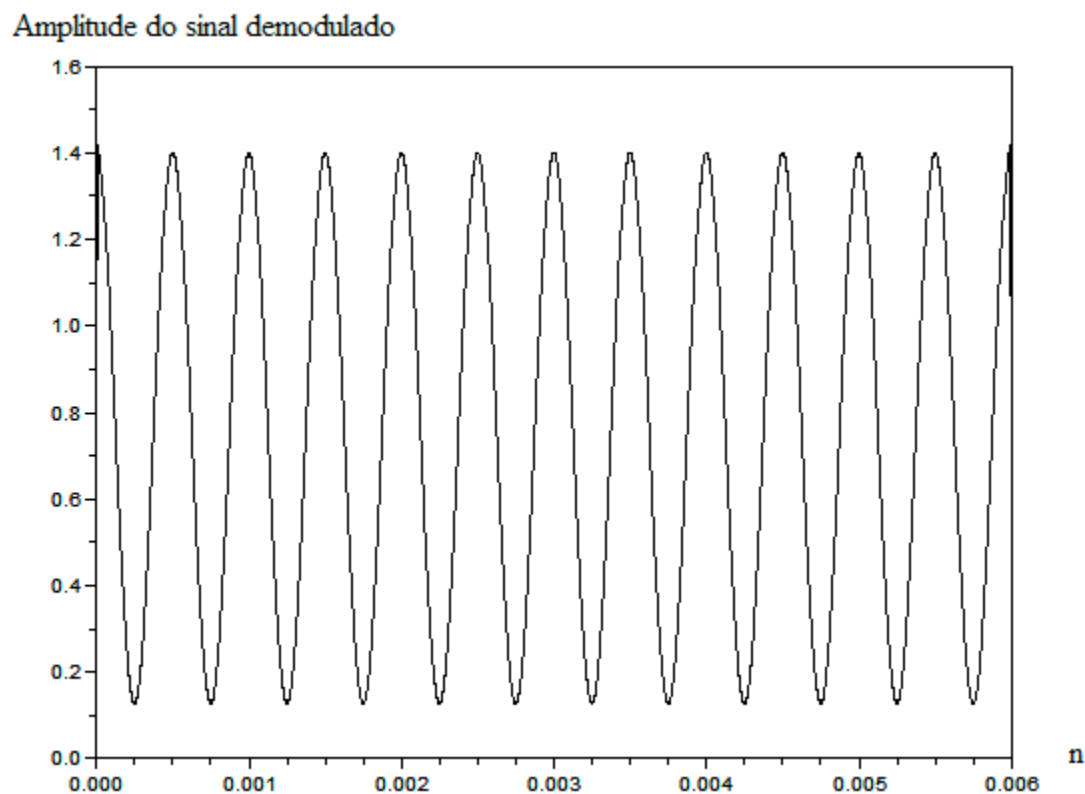


Figura 5.21 - Demodulação AM-DSB com portadora: sinal digital demodulado no tempo.

5.3.2 Simulação da Demodulação AM-SSB: Resultados e Análise

Para a simulação da demodulação AM-SSB no *Scilab* foi adotada a premissa de que o sinal a ser detectado possa ocupar a faixa de 0 a 5 kHz com nenhum componente espectral nos extremos da faixa. Para tal, foi simulado o uso de um sinal modulante cossenoidal de teste com frequência de 2kHz e o uso da faixa lateral inferior. Uma simulação semelhante poderia ter sido feita para um sinal SSB-USB, alcançando resultados análogos.

Supondo-se uma frequência intermediária de 455 kHz e usando-se a teoria da Seção 5.1.2, sabe-se que o espectro esperado para o sinal analógico a ser demodulado, ou seja, $\Phi_{SSB-LSB}(\Omega)$, segue a Figura 5.22-b. Observa-se a presença de um impulso em $\Omega = \pm(455 - 2) 2\pi k = \pm 453 2\pi k \text{ rad/s}$. A expressão do sinal a demodular no domínio do tempo segue a Equação (5.9) com $f(t) = \cos(2\pi 2000t)$, $\Omega_c = 2\pi FI$ e $f_h(t) = \sin(2\pi 2000t)$ (já que o seno é o cosseno defasado de $\pi/2$). Sendo assim:

$$\Phi_{SSB-LSB}(t) = \cos(2\pi 2000t) \cos(2\pi 455000t) + \sin(2\pi 2000t) \sin(2\pi 455000t)$$

A frequência de amostragem escolhida $f_s = 1\text{MHz}$ obedece ao Teorema da Amostragem e é maior que o dobro da maior frequência estar presente no canal selecionado, ou seja, $f_s > 2 \times (455 \text{ kHz} + 5\text{kHz})$ (sinal modulante de 5kHz e uso de banda lateral superior).

O sinal a demodular já digitalizado, ou seja, o sinal $s[n]$ da Figura 5.14, tem seu espectro totalmente compatível com sua DTFT, cuja teoria foi apresentada na Seção 4.2.1. Neste caso, a Transformada de Fourier Contínua de $s[n]$, $S_s(j\Omega)$, é formada por réplicas de $\Phi_{SSB-LSB}(\Omega)$ em torno da frequência de amostragem (veja Figura 5.22-c). Além disso, a DTFT de $x[n]$ pode ser facilmente obtida a partir de $S_s(j\Omega)$ através de um simples reescalonamento no eixo da frequência fazendo-se $\omega = \Omega T_s$, onde $T_s = 1/f_s = 10^{-6}$. Sendo assim, a DTFT de $s[n]$ deve ter o aspecto ilustrado na Figura 5.22-d.

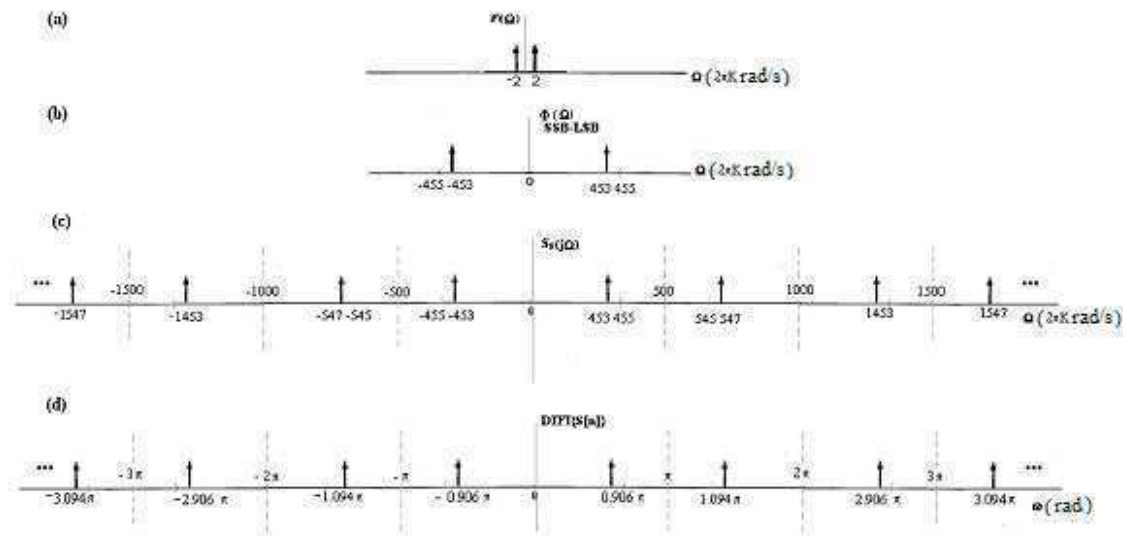


Figura 5.22 – (a) Transformada de Fourier do sinal modulante; (b) Transformada de Fourier de $\Phi_{\text{SSB-LSB}}(t)$; (c) Transformada de Fourier de $s[n]$; (d) DTFT de $s[n]$.

O sinal $s[n]$ é produzido matematicamente no *Scilab* (vide Apêndice A.2 - Simulação da Demodulação AM-SSB-LSB) e contém amostras espaçadas de 10^{-6} . Em seguida, a simulação inclui o cálculo da FFT de $s[n]$ e a plotagem de seu Espectro de Magnitude (veja Figura 5.23). Esta plotagem foi feita para ω no intervalo $[0, 2\pi]$. Para uma melhor visualização em termos de frequência, o eixo ω foi multiplicado pelo fator $10^6/(2\pi)$. Há a presença de duas componentes espectrais (ou impulsos): em 453 kHz e em 547 kHz. Observe a total compatibilidade entre o espectro previsto pela teoria (Figura 5.22) e o mostrado na Figura 5.23, já que a FFT é apenas um método mais eficaz de se determinar a DFT e a DFT é composta por amostras da DTFT. Observe também a simetria em torno de 500MHz, ou π . Como visto anteriormente, a simetria em torno de $kf_s/2$ (sendo k um inteiro) sempre estará presente em todos os espectros plotados e é decorrente do processo de digitalização.

A primeira etapa da demodulação de $s[n]$ simulada no *Scilab* é o cálculo de $q[n]$ conforme ilustrado na Figura 5.14, multiplicando-se $s[n]$ por um sinal cossenoidal discreto de frequência 455 kHz, com amostras também espaçadas de 10^{-6} (veja Anexo – Simulação Demodulação AM-SSB-LSB). Após efetuar-se este cálculo, $q[n]$ tem seu espectro em módulo plotado pelo *Scilab*, entre as frequências de 0 e 1MHz, através do uso da FFT (vide Apêndice A.2 - Simulação da Demodulação AM-SSB-LSB). Este espectro está mostrado na Figura 5.24. Observa-se a presença de quatro componentes espectrais, ou impulsos, situados nas frequências 2 kHz, 92 kHz, 908 kHz e 998 kHz. Estas componentes espectrais podem ser explicadas pela convolução entre o espectro de $s[n]$ (Figura 5.23) e o espectro do cosseno de 455 kHz (dois impulsos em ± 455 kHz) e pela propriedade da simetria em torno de $k f_s/2$,

sendo k um inteiro. Esta convolução produz componentes em:

- $455+453 = +908$ kHz
- $455-453 = +2$ kHz
- $455+547 = +1002$ kHz
- $455-547 = -92$ kHz
- $-455+453 = -2$ kHz
- $-455-453 = -908$ kHz
- $-455+547 = +92$ kHz
- $-455-547 = -1002$ kHz

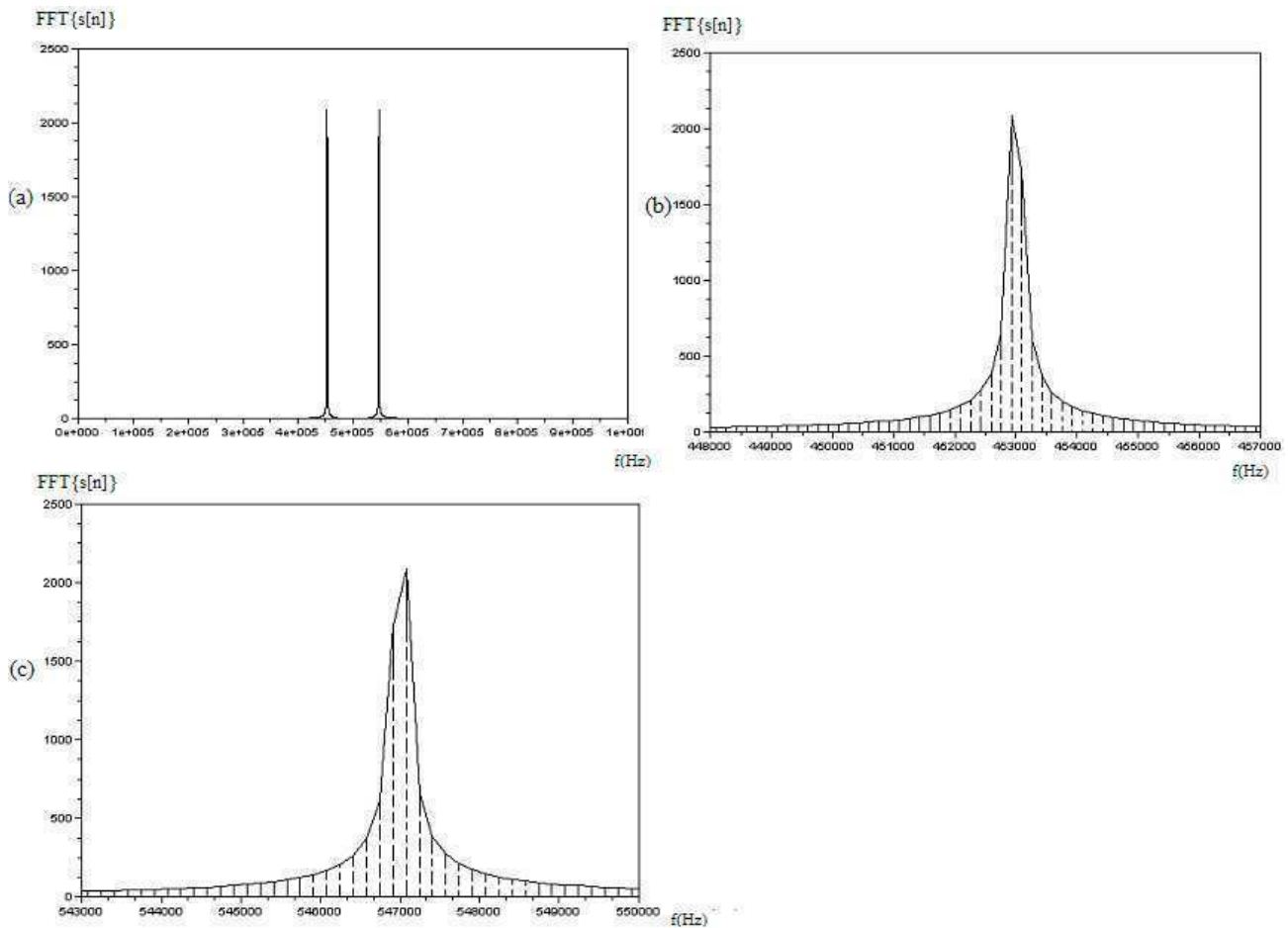


Figura 5.23– (a) Espectro de $s[n]$ no *Scilab*; (b) zoom em torno de 453kHz; (c) zoom em torno de 547 kHz.

No espectro de $q[n]$, a componente em +1002 kHz revela que existe uma componente em +998 kHz. Isto ocorre por causa da simetria em torno de 1 MHz. Como foram plotadas somente as componentes presentes nas frequências de 0 a 1MHz, aparecem as componentes em: +2 kHz, +92 kHz, +908 kHz e +998 kHz (veja Figura 5.24).

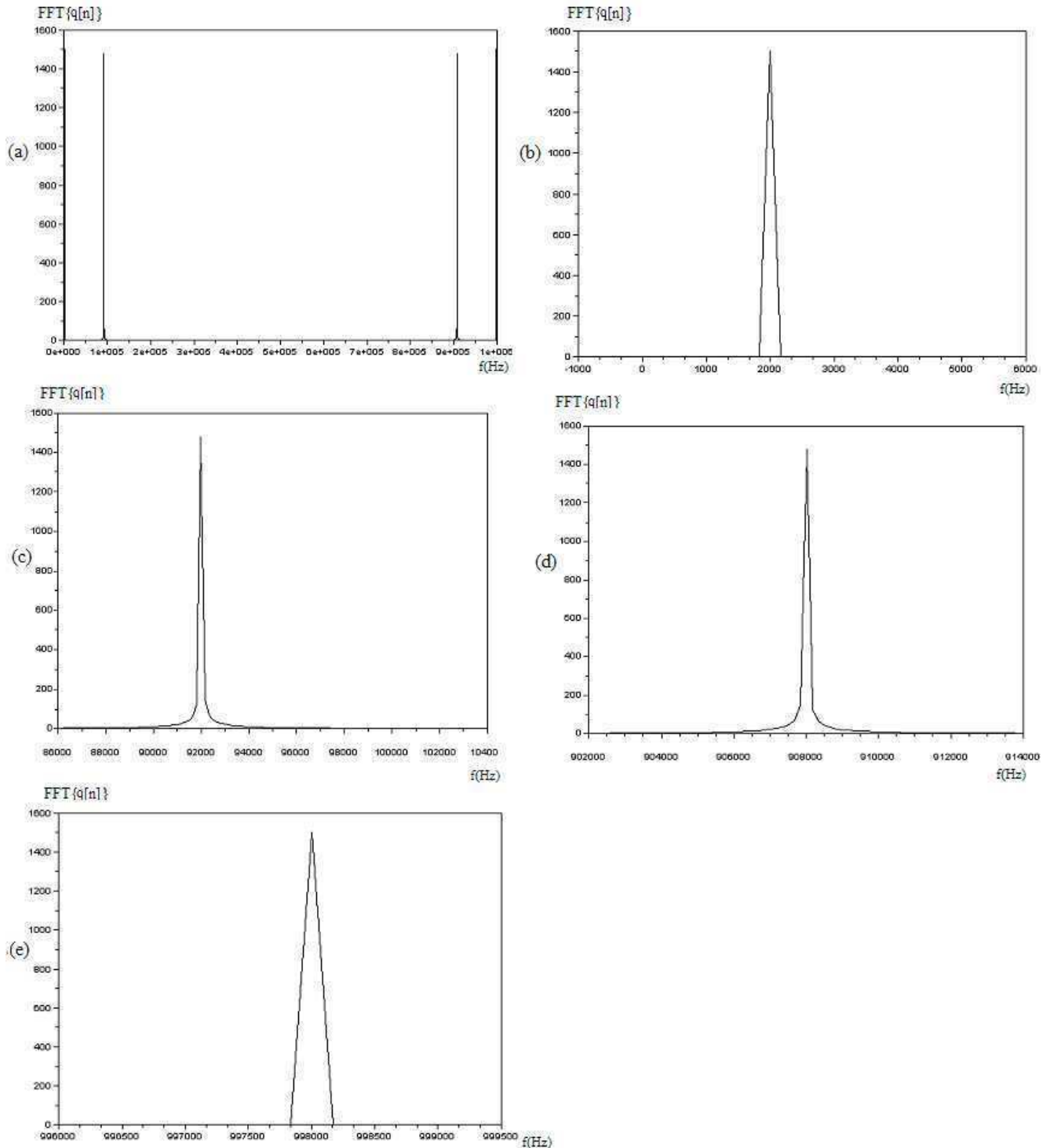


Figura 5.24– (a) Espectro em magnitude de $q[n]$ e $i[n]$ no *Scilab*; (b) zoom em torno de 2kHz; (c) zoom em torno de 92 kHz; (d) zoom em torno de 908 kHz; (e) zoom em torno de 998kHz.

A seguir, no *Scilab*, foi efetuado o cálculo de $i[n]$ pela multiplicação das amostras $s[n]$ por um sinal senoidal discreto de 455 kHz, com amostras também espaçadas de 10^{-6} (veja Apêndice A.2 - Simulação da Demodulação AM-SSB-LSB). O espectro em módulo de $i[n]$ foi então plotado e observou-se que ele era igual ao espectro em módulo de $q[n]$ (Figura 5.24), o que é totalmente compatível com a teoria apresentada na Seção 5.12 (veja Figura 5.11-b e c).

Posteriormente, os coeficientes de um filtro de Hilbert, de ordem $N=200$, foram calculados, usando-se uma função própria do *Scilab* (veja Apêndice A.2 - Simulação da Demodulação AM-SSB-LSB). Este número N foi fixado após sucessivas tentativas de se obter, com N mínimo, uma demodulação mais limpa, ou seja, um espectro resultante (no ponto M da Figura 5.14) com apenas um componente em 2kHz ou com amplitudes desprezíveis para componentes em frequências maiores que esta. Após serem calculados, os coeficientes do filtro foram plotados conforme mostrado na Figura 5.25, que exibe $h[0]$ a $h[200]$. Tais coeficientes exibem uma característica interessante: todos os coeficientes de índice par são nulos. Isso reduz pela metade o número de operações MAC que devem ser realizadas na filtragem. O *Scilab* também calculou e plotou a função ganho deste filtro, ou seja, o módulo de sua resposta em frequência, dado em decibéis. Esta função ganho está mostrada na Figura 5.26. O eixo horizontal representa a frequência angular digital que está normalizada para 2π radianos. É notório que a função ganho é crítica perto das frequências próximas de $\omega=0$, apresentando *ripple*, ou seja, a função ganho do filtro não é totalmente plana, como deveria ser idealmente. Isto se deve à aproximação do Filtro de Hilbert ideal por um filtro FIR linear causal pelo truncamento da resposta ao impulso ideal (infinita). Quanto menor a ordem do filtro, maior o *ripple* apresentado próximo a $\omega=0$. Pela Figura 5.26-c, verifica-se que, para $N=200$, só a partir de $\omega = 2\pi 0,0065$, o *ripple* da função ganho fica razoável, ou seja, menor que 1dB.

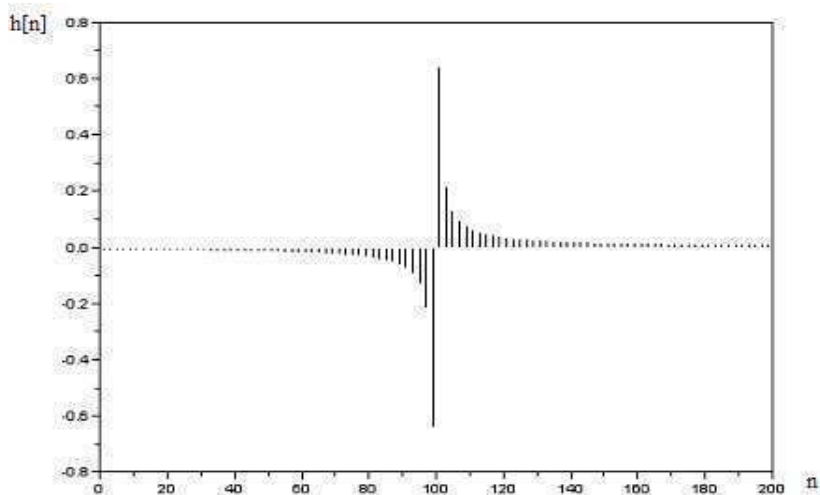


Figura 5.25 – Coeficientes do filtro de Hilbert utilizado na simulação da detecção AM-SSB.

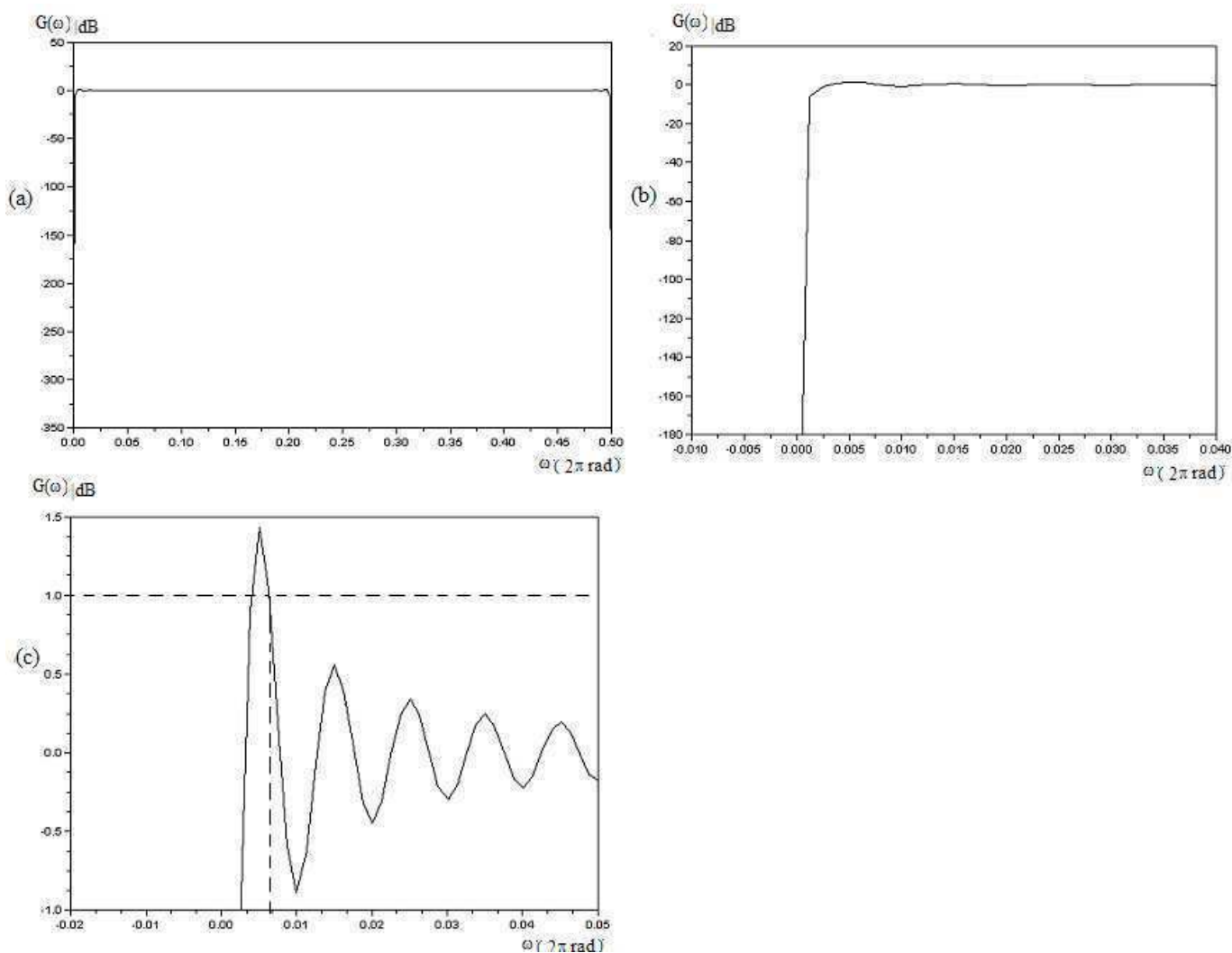


Figura 5.26 – (a) Função Ganho do filtro de Hilbert utilizado na simulação da detecção AM-SSB ; (b) zoom em torno de $\omega = 0$; (c) zoom para ver a partir de qual freqüência a função ganho fica condicionada ao intervalo $[-1, +1]$ dB.

A etapa seguinte da simulação da demodulação SSB-LSB foi a filtragem propriamente dita, ou seja, foi calculada a convolução de $i[n]$ com os coeficientes do filtro de Hilbert (vide Apêndice A.2 - Simulação da Demodulação AM-SSB-LSB) e o espectro em módulo do sinal resultante $i_h[n]$ foi plotado. O resultado está ilustrado na Figura 5.27. Observa-se a semelhança desta figura com relação à Figura 5.24-a, a menos de uma pequena atenuação sofrida pela componente de frequência em 2kHz. Tal atenuação é decorrente do *ripple* presente no filtro de Hilbert, mas, conforme será visto, isto não prejudicou a demodulação.

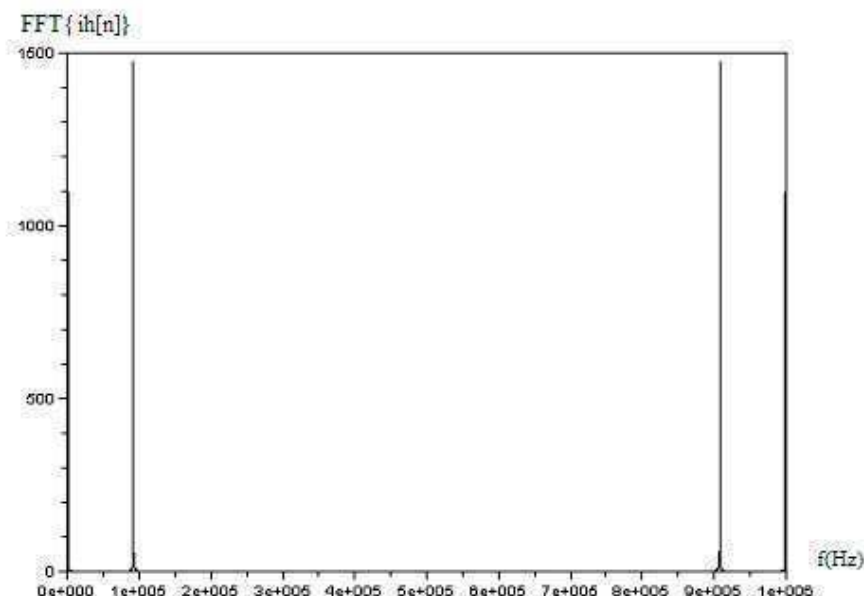


Figura 5.27– Espectro em magnitude de $i_h[n]$ no *Scilab*.

A seguir, conforme mostrado na Figura 5.14, o sinal demodulado foi obtido no *Scilab* pela subtração de $q[n]$ por $i_h[n]$, obtendo-se $f[n]$. Como o sinal demodulado ainda é um sinal digital, na cadeia de recepção AM ele deve ainda passar pela conversão D/A antes de alimentar um transdutor analógico (veja Figura 5.12). O espectro de $f[n]$ é calculado através de sua FFT e, a seguir, plotado, conforme Figura 5.28. No domínio do tempo, $f[n]$ pode ser visto na Figura 5.29. Nota-se que, no gráfico, ele aparece, para uma melhor visualização, como uma função contínua, tendo seus pontos discretos no tempo ligados. Além disso, o sinal demodulado $f[n]$ tem as seguintes características: forma cossenoidal com amplitude pico a pico $A_{pp}=1,73$, valor médio=0 e período=0,0005 (frequência de 2 kHz). Tais características confirmam o sucesso da demodulação digital na comparação com o sinal modulante (original): cosseno com frequência de 2 kHz, valor médio de 0 V e $A_{pp}=2V$. Apesar da diferença encontrada na amplitude pico a pico, isto não tem muita importância, já que simulações com sinais modulantes cossenoidais de outras frequências revelaram atenuação

semelhante, ou seja, a demodulação não está introduzindo distorção dentro da banda do canal recebido. Atenuação mais acentuada só pôde ser sentida para sinais bem próximos de 0Hz, porém, como anteriormente visto, estas frequências geralmente não estão presentes.

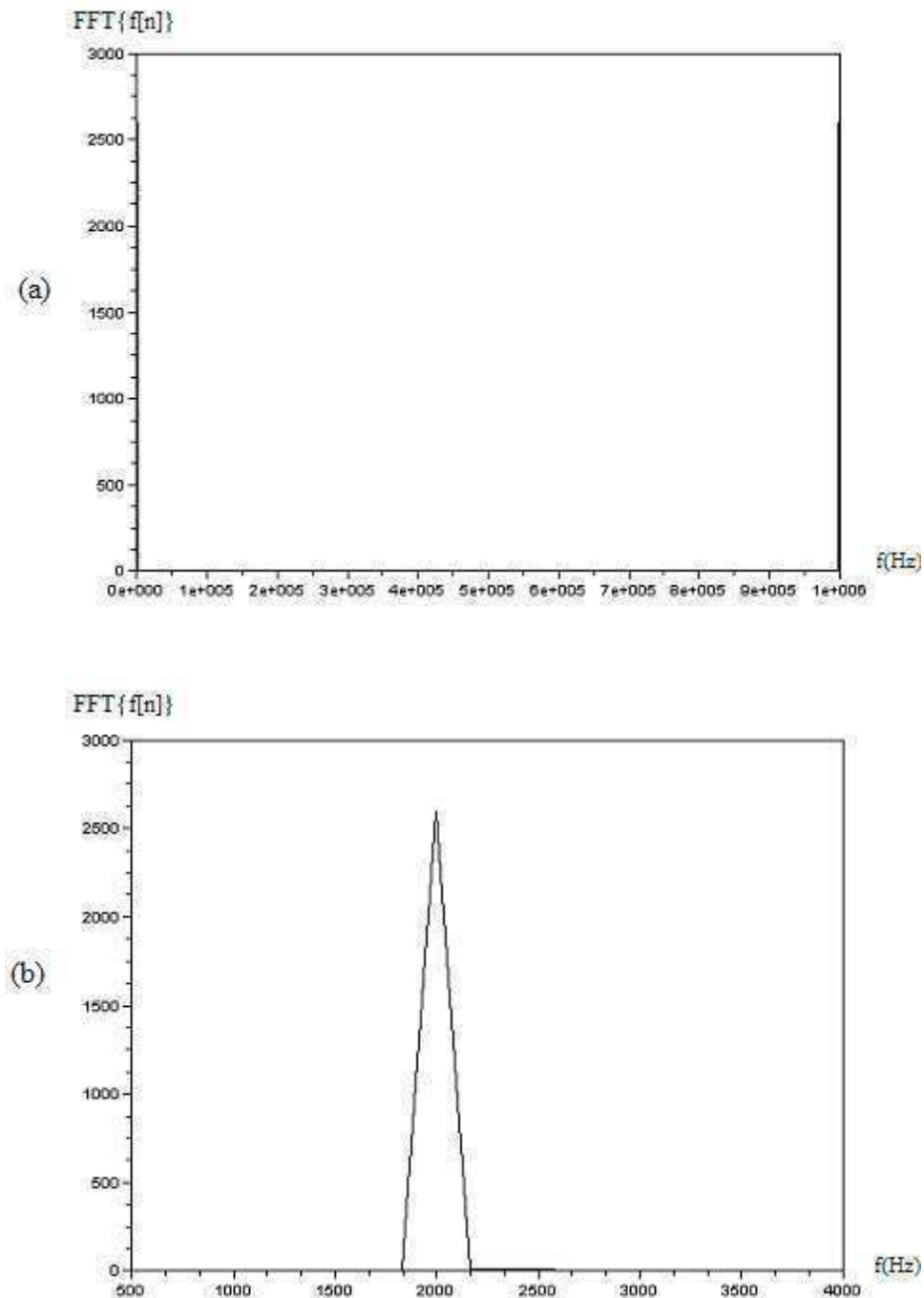


Figura 5.28 - (a) Demodulação AM-SSB-LSB: Espectro do sinal demodulado digitalmente; (b) zoom em torno de 2 kHz.

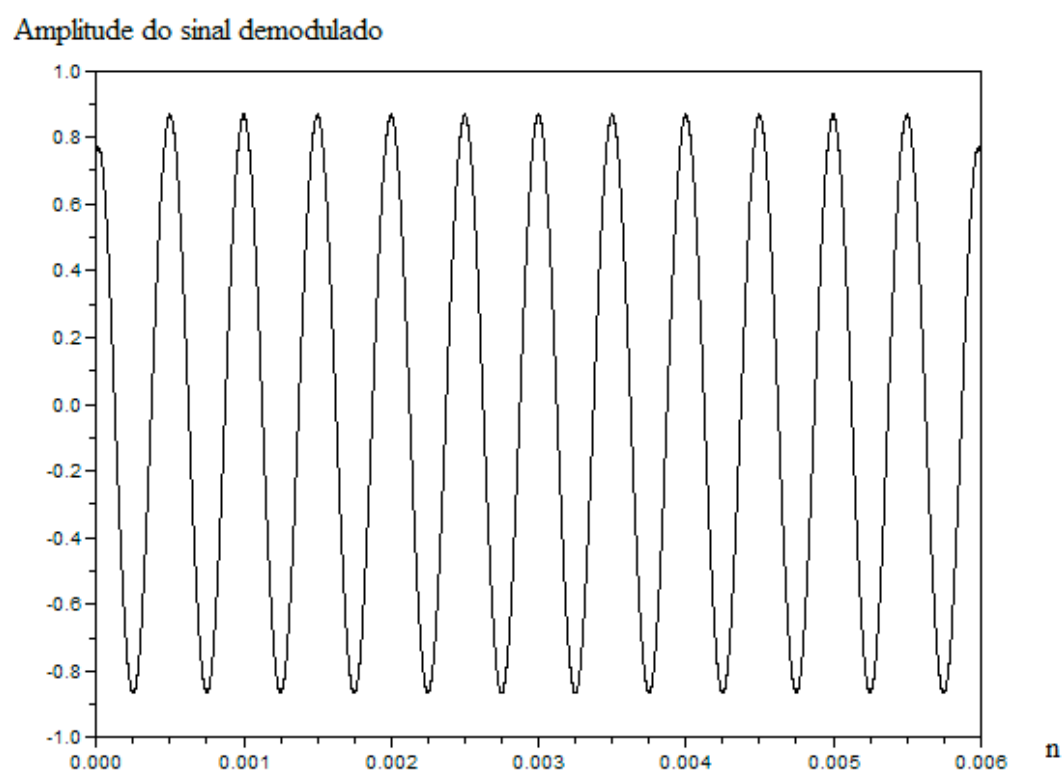


Figura 5.29 - Demodulação AM-SSB-LSB: sinal digital demodulado no tempo.

6

CONCLUSÕES E TRABALHOS FUTUROS

No presente trabalho, exploraram-se diversos aspectos técnicos relacionados aos rádios definidos por *software*, uma tecnologia recém-desenvolvida, em que um *software* de processamento de sinal, executando em uma dada plataforma de processamento, realiza as funções dos rádios que tradicionalmente eram feitas através de *hardware*. As definições, princípios gerais, aplicações e principais arquiteturas dos rádios de *software* foram descritos, assim como seu *hardware* subjacente.

Ao propor, como exemplo, um esquema de rádio de *software* para a recepção de sinais AM, incluindo dois diferentes tipos de demodulação, este trabalho ilustrou como utilizar a teoria apresentada em uma aplicação prática. Através da realização de simulações computacionais e da obtenção de resultados devidamente analisados à luz da teoria de processamento digital de sinais, a eficácia do esquema proposto foi comprovada.

Como possível trabalho a ser realizado, tomando por base o presente trabalho, propõe-se a montagem, em *hardware*, do esquema de recepção supracitado, para verificação experimental dos resultados apresentados. Com o uso das mesmas faixas de frequências e demodulações, é possível, a partir dos procedimentos realizados em cada bloco funcional presente nas simulações, escrever componentes de *software* em uma linguagem específica para o DSP a utilizar.

Outra opção de trabalho a realizar seria a proposição e a simulação computacional de esquemas de rádio de *software* para a recepção de sinais de outro tipo, que utilizem outras demodulações e faixas de frequência. Por exemplo, uma arquitetura de rádio de *software* com digitalização em FI para a recepção de sinais FM poderia ser proposta e simulada computacionalmente. Alternativamente, poderia se pensar na recepção de sinais de GPS ou

aplicações úteis em telefonia celular e em outros sistemas de telecomunicações. Complementando esses trabalhos, pode-se também fazer a verificação experimental de tais receptores por meio da montagem de *hardware* baseado em um DSP apropriado (a depender da faixa de frequências a ser utilizada).

Adicionalmente, pode-se simular e implementar, em *hardware*, transmissores, tanto para a faixa de frequências e tipos de modulação exploradas neste trabalho como para outras faixas e outros tipos de modulação.

Este trabalho e outros que sejam feitos em continuação a ele podem servir como base para a produção industrial de equipamentos de telecomunicações, possivelmente no Brasil.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] E. Sereni et al. **A Software Re-Configurable Architecture for 3G and Wireless Systems**. University of Perugia, Perugia, Italy, 2002.
Disponível em: <<http://www.digilab2000.it/file/paper92.pdf>>
- [2] Schiphorst, R. **Demonstration of the Software-Radio Concept**. M.Sc. Thesis - University of Twente, The Netherlands, 2000.
Disponível em: <<http://doc.utwente.nl/56278/1/schiphorst00demonstration.pdf>>
- [3] E. Sereni et al. **A Software Radio OFDM Transceiver for WLAN Applications**. University of Perugia, Perugia, Italy, 2001.
Disponível em: <<http://speech.di.uoa.gr/dsp/X/PERUGI.PDF>>
- [4] Joseph Mitola, III. **Software Radio Architecture: Object Oriented Approaches to Wireless Systems Engineering**. John Wiley and Sons, 2000.
- [5] Reed J.H. **Software Radio: A Modern Approach to Radio Engineering**. Prentice Hall PTR, 2002.
- [6] Broadcast & Communications Group, Centre for Development of Advanced Computing (Dept. of Information Technology, Govt. of India). **Study Report of SDR Technology & System Design**. India, [ca. 2003]. 4p.
- [7] Merino, M.F.A. **Market Impact of Software Radio: Benefits and Barriers**. M.Sc. Thesis- Massachusetts Institute of Technology, Massachusetts, USA, 2002.
Disponível em: <http://itc.mit.edu/itel/students/papers/merino_thesis.pdf>
- [8] Sen M. Kuo; Woon-Seng Gan. **Digital Signal Processors: Architectures, Implementations, and Applications**. Pearson Prentice Hall, 2005.
- [9] Mitra, Sanjit Kumar. **Digital Signal Processing: a Computer-Based Approach**. McGraw-Hill, 1998.

- [10] Nell, James. **Simulation of an Implementation and Evaluation of the Layered Radio Architecture**. M.Sc. Thesis - Virginia Polytechnic Institute and State University, Virginia, USA, 2002

- [11] CAST Consortium. **Configurable radio with Advanced Software Technology (Report)**.2002.14p.
Disponível em: <<http://klynn.tripod.com/store/CastReportD424.pdf>>

- [12] TEXAS INSTRUMENTS. **Selecting your TI DSP**. Disponível em:
<<http://focus.ti.com/dsp/docs/dspsupportat.tsp?sectionId=4&tabId=1447&familyId=44>>

- [13] Strauss, W. **DSP/Wireless Market Analysis**.
Disponível em: <http://www.dsp-fpga.com/columns/Forward_Thinking/2007/02/>

- [14] Pino, A.V. **Apostila de Processamento Digital de Sinais**. Universidade Católica de Pelotas, RS, Brasil,1998.
Disponível em: <<http://www.fag.edu.br/professores/stenio/dsp.pdf>>

- [15] SITE DO SCILAB. Disponível em: <<http://www.scilab.org/>>

- [16] Scilab Group, INRIA Meta2 Project/ENPC Cergrene. **Signal Processing With Scilab**. Disponível em:
<<http://pauillac.inria.fr/cdrom/ftp/scilab/documentation/pdf/signal.pdf>>

- [17] Lathi, B. P. **Modern Digital and Analog Communication Systems**. Oxford University Press, 1998.

Apêndice A

Programação no Scilab

A.1 Simulação da Demodulação AM-DSB com Portadora

```
f_sinal_modulante=2000;//sinal modulante de 2KHz

FI=455000; //Frequência Intermediária de 455KHz

freq_amostragem=1000000;// Frequência de amostragem de 1MHz

t = 0:1/freq_amostragem:0.006; //período temporal analisado de
0 a 0.006s espaçados de Ts=10-6 s

x=(1.2*cos(2*pi*FI*t))+(0.5*cos(2*pi*(FI+f_sinal_modulante)*
t))+ (0.5*cos(2*pi*(455000-f_sinal_modulante)*t)); // x:
sinal modulado digitalizado

N=size(t,'*');// tamanho do vetor de tempo
f=freq_amostragem*(0:(N-1))/N; // vetor frequência associado
n=size(f,'*');

Xw=fft(x); // Xw: espectro do sinal modulado amostrado

clf
plot2d(f,abs(Xw(1:n)))

//DEMODULANDO

modx=abs(x);// obtendo o módulo de x

MODXw=fft(modx);// MODXw: espectro de módulo de x

clf
plot2d(f,abs(MODXw(1:n))) //plotando MODXw

// Filtro Passa-Baixas usando a janela de Kaiser

[wft,wfm,fr]=wfir();

//parâmetros do filtro: tipo 'lp' (low pass); N=46; beta=5.65;
fh=0.04 (0.04 * 2pi, frequência de corte=40kHz)

clf
plot2d(fr,(20 * log10(wfm)))// módulo da resposta em
```

```

frequência do filtro em dB

y=convol(wft,modx); // y:modx já filtrado

size(y)

y=y(24:(6047-23)); //compatibilizando a dimensão de t e y

Yw=fft(y); //Yw: espectro de y

clf
plot2d(f,abs(Yw(1:n))) // plotando Yw
clf
plot2d(t,y) // sinal demodulado no tempo

```

A.2 Simulação da Demodulação AM-SSB-LSB

```

f_sinal_modulante=2000; // sinal modulante de 2KHz

FI=455000; //Frequência Intermediária de 455KHz

freq_amostragem=1000000; // Frequência de amostragem de 1MHz

t = 0:1/freq_amostragem:0.006;

//calculo de s

deff(' [x1]=f(t) ', 'x1=cos(2*pi*FI*t)*cos(2*pi*f_sinal_modulan
te*t) ');

y=feval(t,f);

deff(' [x2]=g(t) ', 'x2=sin(2*pi*FI*t)*sin(2*pi*f_sinal_modulan
te*t) ');

y2=feval(t,g);

s=y+y2; // s: sinal SSB (banda lateral inferior) amostrado a
1MHz

N=size(t, '*'); //número de amostras
f=freq_amostragem*(0:(N-1))/N; //vetor frequência associado
n=size(f, '*') // n: tamanho do vetor f

espectro_s=fft(s); // espectro de s ; fft: Transformada
Rápida de Fourier

clf

```

```

plot2d(f,abs(espectro_s(1:n)))// plotando o espectro de s

//-----
//DEMODULAÇÃO

portcos=cos(2*%pi*FI*t);// cosseno de tempo discreto e freq.
FI

for j = 1:N, q (j)= s(j)*portcos(j);end // calculando q: sinal
resultante da multiplicação amostra a amostra entre portcos e
s

espq=fft(q);//calculando o espectro de q

clf
plot2d(f,abs(espq(1:n)))//plotando o espectro de q

portseno=sin(2*%pi*FI*t);// seno de tempo discreto e freq. FI

for j = 1:N, i (j)= s(j)*portseno(j);end//calculando i: sinal
resultante da multiplicação amostra a amostra entre portseno e
s

espi=fft(i);//calculando o espectro de i
clf
plot2d(f,abs(espi(1:n)))//plotando o espectro de i

//filtro de Hilbert
xh=hilb(201);// cálculo dos coeficientes de um filtro de
Hilbert com 201 coeficientes (100 deles resultam em
coeficientes nulos)

clf
plot2d3(0:200,xh)// plotagem dos coeficientes do filtro de
Hilbert {h[n]}

[xm,fr]=frmag(xh,400); //calculando o módulo da resposta em
frequência do filtro para 400 pontos

clf
plot2d(fr,20*log10(xm))// plotagem do módulo da resposta em
frequência do filtro de Hilbert em dB

ih=convol(xh,i);// cálculo de ih: i após ser filtrado pelo
Filtro de Hilbert

size(ih)

ih=ih(101:6101);// ajustando o tamanho de ih para se adequar
ao tamanho do vetor t

```

```
size(ih)

espih=fft(ih); // calculando o espectro de ih

clf
plot2d(f,abs(espih(1:n))) // plotando o espectro de ih


sinaldem=q'- ih; // obtendo o sinal demodulado

espsinaldem=fft(sinaldem); //calculando o espectro do sinal
demodulado

clf
plot2d(f,abs(espsinaldem(1:n))) // plotagem do espectro do
sinal demodulado

clf
plot2d(t,sinaldem) // plotagem do sinal demodulado no tempo
```