

Implementação de um Demodulador Universal baseado em uma Plataforma Experimental para Rádio Reconfigurável

Angilberto Muniz Ferreira Sobrinho

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande – Campus de Campina Grande, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências do Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Orientadores

Prof. DSc. José Sérgio da Rocha Neto
Prof Dr. Antonio Marcus Nogueira Lima

Campina Grande, Paraíba, Brasil
© Angilberto Muniz Ferreira Sobrinho

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

F383i

2007

Ferreira Sobrinho, Angilberto Muniz

Implementação de um demodulador baseado em uma plataforma experimental para radio reconfiguravel / Angilberto Muniz Ferreira Sobrinho. — Campina Grande, 2007.

71p.: il.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Referências

Orientadores: Prof^o. DSc. Jose Sergio da Rocha Neto, Prof^o. Dr. Antonio Marcus Nogueira Lima.

1. Processamento da Informação – Engenharia Elétrica 2. Rádio Configurável – Demoludor Universal 3. Demoludor Universal – Rádio Reconfigurável 4. Rádio Reconfigurável – Plataforma Experimental I. Título.

CDU –621.391: 654.195

UFCG - BIBLIOTECA - CAMPUS I	
1342	11.09-07

**IMPLEMENTAÇÃO DE UM DEMODULADOR UNIVERSAL BASEADO EM UMA
PLATAFORMA EXPERIMENTAL PARA RÁDIO RECONFIGURÁVEL**

ANGILBERTO MUNIZ FERREIRA SOBRINHO

Dissertação Aprovada em 02.02.2007



JOSÉ SÉRGIO DA ROCHA NETO, D.Sc., UFCG
Orientador



ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFCG
Orientador



EDMAR CANDEIA GURJÃO, D.Sc., UFCG
Componente da Banca



MARIA DE FÁTIMA QUEIROZ VIEIRA TURNELL, Ph.D., UFCG
Componente da Banca

CAMPINA GRANDE - PB
FEVEREIRO - 2007

DEDICATÓRIA

Dedico este trabalho aos meus pais e irmãos, pelo carinho, respeito, dedicação e compreensão nos momentos de dificuldade.

Em especial, aos Srs. Chico Paulista (*In-Memorian*) e Barriga (*In-Memorian*).

*A minha esposa Marília
e meus filhos Anderson e Carolina*

AGRADECIMENTOS

- Ao Prof. Dr. José Sérgio da Rocha Netto, que não mediu esforços em oferecer todas as condições necessárias à realização do presente trabalho.
- Ao prof. Dr. Antonio Marcus Nogueira Lima, pelas observações, sugestões e tentativas de enriquecimento das possíveis aplicações dos resultados deste trabalho.
- À Universidade do Estado do Amazonas – UEA/EST, pelo incentivo ao desenvolvimento e capacitação de seus docentes.
- À Universidade Federal de Campina Grande – UFCG, pela oportunidade da participação no programa de Pós-graduação e acesso ao seu corpo docente e instalações.
- À empresa ELYTE Tecnologia, pelo patrocínio durante a primeira metade do programa e pelo desafio do trabalho que serviu de motivação inicial para este estudo.
- Ao Instituto Certi Amazônia – ICA, pelo compartilhamento de tempo na última etapa do programa, sem o qual este trabalho não seria concluído.
- Ao amigo e professor Roberto Higino pela compreensão, apoio e gerenciamento dos conflitos de horários.
- Ao meu velho amigo MSc. Ja-Ce-Ka-Son, pelo estímulo por seu exemplo.
- Aos colegas do curso pelo agradável relacionamento e, em especial, às amigas Carolina, Cláudia e Ângela, que tanto me ajudaram.
- À minha família que me apoiou incondicionalmente e se sacrificou durante todo o período em que estive envolvido neste trabalho.

RESUMO

Neste trabalho investigamos a tecnologia de Rádio Definido por Software – RDS (Software Defined Radio - SDR), como uma plataforma experimental para o estudo e desenvolvimento de sistemas de comunicação. Seu conceito está baseado na utilização de FPGA (Field Programmable Gate Array) como hardware para o processamento inicial de sinais e no software Matlab como ferramenta para o desenvolvimento de algoritmos. Esta plataforma serve de arcabouço para a implementação de um Demodulador Universal, capaz de reconhecer e identificar a modulação presente em um sinal desconhecido. Como prova de conceito, é apresentada uma aplicação básica que tem por finalidade monitorar a camada física de um sistema de comunicação.

ABSTRACT

Software Defined Radio (SDR) is investigated and used as a tool for experimentation and development of communication systems. Based on FPGA technology as a hardware front-end and Matlab as a back-end software tool, the platform is used as a framework for implementation of a Universal Demodulator. A proof-of-concept is demonstrated with an application that monitors the physical layer of a communication system.

ÍNDICE

Dedicatória.....	ii
Agradecimentos	iii
Resumo.....	iv
Abstract	v
Índice	vi
Lista de Figuras	vii
Lista de Tabelas	ix
Lista de Abreviaturas	x
Capítulo 1 - Introdução	1
Capítulo 2 – Comunicação Digital	5
Capítulo 3 - Plataforma Experimental	27
Capítulo 4 – Reconhecimento de Modulação... ..	39
Capítulo 5 - Aplicação – Demodulador Universal	48
Capítulo 6 – Resultados Experimentais	57
Capítulo 7 – Conclusões e Perspectivas Futuras	63
Referências Bibliográficas	66
Anexos	69
A.1 – Instalação do GNU-Radio.....	69
A.2 – Datasheets do USRP.....	71

LISTA DE FIGURAS

Figura 1.1 – Arquitetura UMA – Acesso transparente entre WiFi/BlueTooth/GSM.....	1
Figura 1.2 – Cenário Macro discutido pela iniciativa E2R.....	2
Figura 1.3 – Diagrama de blocos da aplicação Demodulador Universal.....	4
Figura 2.1 – Diagrama de blocos de um receptor digital típico.....	5
Figura 2.2 – Espectro real amostrado	6
Figura 2.3 – Espectro Amostrado após translação para DC.....	6
Figura 2.4 – Espectro complexo após o filtro passa-baixas.....	6
Figura 2.5 – Espectro do sinal após o processo de decimação	7
Figura 2.6 – Conceito simplificado de modulação	8
Figura 2.7 – Esquemas básicos de modulação	9
Figura 2.8 – Componentes <i>Em-Fase</i> e <i>Em-Quadratura</i>	10
Figura 2.9 – Envoltória complexa e sinal analítico.....	10
Figura 2.10 – Equivalência Polar-Retangular.....	11
Figura 2.11 – Componentes I e Q na transmissão.....	11
Figura 2.12 – Componentes I e Q na recepção.....	12
Figura 2.13 – Modulação AM no domínio do tempo.....	13
Figura 2.14 – Modulação AM (relação espectral).....	13
Figura 2.15 – Modulação FM.....	14
Figura 2.16 – Espectro resultante da modulação FM.....	14
Figura 2.17 – Processo de modulação BPSK.....	15
Figura 2.18 – Constelação mostrando as regiões válidas para sinais BPSK.....	16
Figura 2.19 – Demodulação conceitual BPSK.....	16
Figura 2.20 – Demodulação BPSK.....	17
Figura 2.21 – Diagrama de blocos de um Modulador QPSK	18
Figura 2.22 – Sinais gerados pelo modulador QPSK.....	19
Figura 2.23 – Constelação QPSK.....	19
Figura 2.24 – Diagrama de blocos de um Demodulador QPSK.....	20
Figura 2.25 – Demodulação QPSK.....	21
Figura 2.26 – Diagrama de Blocos de um RDS ideal.....	22
Figura 2.27 – Arquitetura prática para um RDS.....	22
Figura 2.28 – Conceito que deu origem ao GNU-Radio.....	23
Figura 2.29 – Abstração e Controle em GNU-Radio.....	23
Figura 2.30 – Fluxo de sinais em uma aplicação GNU-Radio/USRP.....	24
Figura 2.31 – Demodulação FM em GNU-Radio.....	25
Figura 3.1 – Kit USRP.....	27
Figura 3.2 – Diagrama de Blocos do USRP.....	28
Figura 3.3 – Conversores A/D e D/A MxSP (Mixed-Signal Processor).....	29
Figura 3.4 – Diagrama de Blocos detalhado do USRP.....	30

Figura 3.5 – Diagrama de blocos do DDC.....	31
Figura 3.6 – Rotação de coordenadas por um ângulo fixo.....	36
Figura 3.7 – Rotação de um vetor por um ângulo fixo.....	33
Figura 3.8 – DDC convencional usando NCO baseado em tabelas ROM.....	35
Figura 3.9 – DDC baseado no processador CORDIC no modo rotacional.....	36
Figura 3.10 – Blocos básicos de um filtro CIC.....	37
Figura 3.11 – Filtro decimador CIC de três estágios.....	37
Figura 3.12 – Filtro interpolador CIC de três estágios.....	38
Figura 4.1 – Abordagem baseada em hipóteses.....	39
Figura 4.2 – Abordagem baseada em reconhecimento de padrões.....	39
Figura 4.3 – Distribuição espectral existente em diferentes tipos de modulação.....	41
Figura 4.4 – Atributo $\gamma_{\max f}$ em função do tipo de modulação.....	42
Figura 4.5 – Atributo $\gamma_{\max a}$ em função do tipo de modulação analógica.....	43
Figura 4.6 – Relação entre variação de frequência e o atributo $\gamma_{\max a}$	44
Figura 4.7 – Relação entre variação de amplitude e o atributo $\gamma_{\max a}$	44
Figura 4.8 – Não-linearidade aplicada a um sinal BPSK.....	46
Figura 4.9 – Não-linearidade aplicada a um sinal QPSK.....	46
Figura 5.1 – Diagrama de blocos da aplicação Demodulador Universal.....	48
Figura 5.2 – Atributos considerados para determinação do tipo de Modulação.....	49
Figura 5.3 – Fluxograma de decisão do Demodulador Universal.....	50
Figura 5.4 – Classificação de modulação baseada no atributo <i>gama-max-f</i>	50
Figura 5.5 – Relação entre o atributo <i>gama-max-a</i> e modulação AM/FM.....	51
Figura 5.6 – Espectro resultante da operação $s^2(t)$ para modulação BPSK.....	51
Figura 5.7 – Espectro resultante da operação $s^2(t)$ para modulação QPSK.....	52
Figura 5.8 – Raias espectrais identificadas no espectro de $s^2(t)$	52
Figura 5.9 – Modelo Simulink para seleção de Demodulador.....	56
Figura 6.1 – Modelo Simulink do gerador de pseudo-voz.....	57
Figura 6.2 – Sinal pseudo-voz (a), portadora modulada em amplitude (b).....	57
Figura 6.3 – Sinal pseudo-voz (a), portadora modulada em frequência (b).....	58
Figura 6.4 – Sinal composto com 100 amostras de cada tipo de modulação.....	58
Figura 6.5 – Atributo $\gamma_{\max f}$ - valores $> 10e4$ indicam modulação analógica.....	59
Figura 6.6 – Atributo $\gamma_{\max a}$ - valores < 5 indicam modulação FM.....	59
Figura 6.7 – Arranjo utilizado para testes com modulação digital.....	60
Figura 6.8 – Arranjo utilizado para testes com modulação analógica.....	61
Figura 6.9 – Aplicação de demonstração do Demodulador Universal.....	62

LISTA DE TABELAS

Tabela 1 – Propriedades das modulações linear e exponencial	19
Tabela 2 – Correspondência entre <i>dibits</i> e ângulos de fase em QPSK.....	27
Tabela 3 – Resultados de simulação (50 testes cada).....	54
Tabela 4 – Resultados reais de classificação (50 testes cada).....	54

LISTA DE ABREVIATURAS

ADC	Analog to Digital Conversion/Converter
AM	Amplitude Modulation
AMR	Automatic Modulation Recognition
ASK	Amplitude Shift Keying
ASIC	Application Specific Integrated Circuit
BPSK	Binary Phase Shift
BSC	Base Station Controller
CIC	Cascaded Integrator Comb
CORDIC	Coordinate Rotation Digital Computer
DAC	Digital to Analog Conversion/Converter
DDC	Digital Down Converter
DFT	Discrete Fourier Transform
DUP	Digital Up Converter
DSP	Digital Signal Processor/Processing
FFT	Fast Fourier Transform
FI	Frequência Intermediária
FM	Frequency Modulation
FPGA	Field Programable Gate Array
FSK	Frequency Shift Keying
GPP	General Porpouse Processor
HDTV	High Definition Television
NCO	Numerical Controlled Oscilator
PM	Phase Modulation
QAM	Quadrature Amplitude Modulation
QPSK	Quaternary Phase Shift
RAN	Radio Access Network
RDS	Rádio Definido por Software
RF	Radio Frequency
ROM	Read Only Memory
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
UMA	Unlicensed Mobil Access
UMAN	Unlicensed Mobil Access Controller
UNC	UMA Network Controller
USRP	Universal Software Radio Peripheral
VCO	Voltage Controlled Oscilator

Capítulo 1

Introdução

Os sinais de comunicação que atravessam o espaço se apresentam com diferentes tipos de modulação e em diferentes frequências configurando um amplo e complexo espectro de desafios para os sistemas de comunicação, que devem identificar e monitorar esses sinais para aplicações diversas, tanto de uso militar quanto civil.

Aplicações de cunho civil incluem, por exemplo, monitoração de transmissores não-autorizados, enquanto que no escopo de aplicações militares, existe a necessidade, por exemplo, de interceptação e decodificação de mensagens em batalhas táticas [1].

O reconhecimento automático da modulação empregada em determinados sinais é de extrema importância em aplicações militares, em primeiro lugar porque o uso de um demodulador inapropriado pode mascarar parcial ou totalmente a mensagem contida no sinal. Em segundo lugar porque reconhecer o tipo correto de modulação pode ajudar na identificação de ameaças e, eventualmente, na determinação de formas de onda adequadas ao bloqueio das comunicações do inimigo.

No âmbito comercial, existe um crescente interesse por sistemas reconfiguráveis voltados para uma convergência de mídias, que tem como recente e atual exemplo, a arquitetura UMA – Unlicensed Mobile Access (Acesso Móvel Não-Licenciado) [2]. Sua arquitetura básica pode ser vista na figura 1.1.

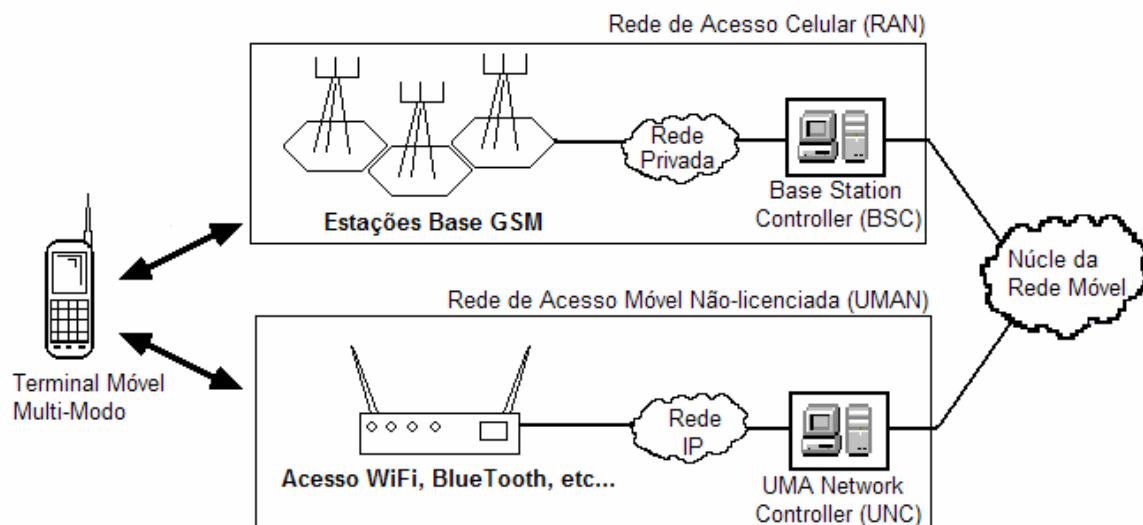


Figura 1.1: Arquitetura UMA – Acesso transparente entre WiFi/BlueTooth/GSM.

Este trabalho tem por objetivo investigar a aplicação da tecnologia RDS na implementação de um demodulador capaz de identificar um subconjunto de tipos de modulação.

1.1 Motivação – RDS

Reconfiguração, dinâmica e/ou estática, parcial e/ou completa é parte fundamental da tecnologia de RDS [3]. Graças a essa possibilidade, os sistemas são projetados e preparados para novas tecnologias de comunicação. Adaptação e mudança são palavras-chave no desenvolvimento dos novos sistemas de comunicação.

Os trabalhos de pesquisa e desenvolvimento mais recentes como TRUST e CAST (Europa) e SDR Fórum e WWRF, tem mostrado que o conceito de reconfiguração, especialmente no contexto das redes de comunicação celular/móvel, é um assunto complexo e com vários aspectos ainda obscuros quanto à arquitetura interna, tanto em relação ao dispositivo a ser reconfigurado quanto em relação à própria rede de comunicação [4].

Os inúmeros trabalhos de pesquisa e experimentação ocorridos na última década resultaram em uma sólida compreensão da tecnologia de Software Rádio e o potencial de sua aplicação em cenários práticos. Os esforços iniciais foram no sentido de aumentar ou maximizar a criação de módulos de comunicação em software. Esta abordagem traz inúmeras vantagens especialmente quando o equipamento de rádio é considerado como parte da rede de comunicação, como é o caso das redes celulares. O potencial dos Software Rádios pode ser explorado concretamente pela reconfiguração dos equipamentos sem necessidade de substituição de partes de hardware.

Evoluindo em direção aos sistemas de 4ª geração e além, essa flexibilidade potencial pode ser útil em diversos cenários de alta complexidade técnica, bem como em cenários comercialmente atrativos, como sugerem, por exemplo, as iniciativas denominadas MFC – Mobile Fixed Convergence (Convergência Fixo-Móvel) através de consórcios como E2R – End-to-End Reconfigurability (Reconfigurabilidade Fim-a-Fim) [5], figura 1.2.

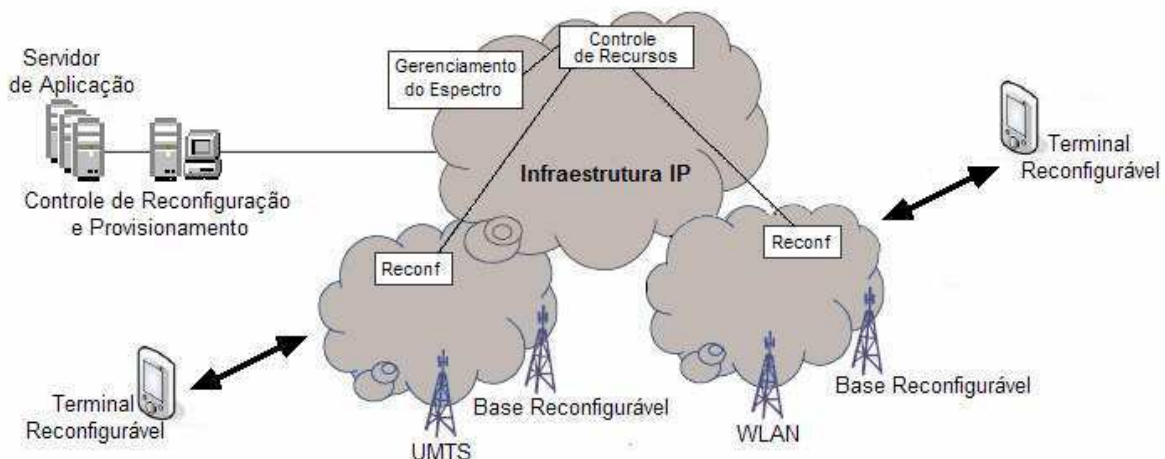


Figura 1.2: Cenário Macro discutido pela iniciativa E2R.

No futuro, o acesso transparente a redes de comunicação (ou redes de redes) para os usuários, a qualidade de serviço e a otimização do gerenciamento da(s) rede(s), exigirão que se leve em consideração a tecnologia de reconfiguração como elemento chave para operação dos sistemas, segundo a visão do movimento WWRF [6].

1.2 Motivação – Plataforma Experimental

O desenvolvimento de sistemas de comunicação sem fio apresenta desafios únicos por sua característica multidisciplinar onde as soluções são resultados de experiências nas áreas de eletrônica, processamento de sinais e computação. Inúmeros parâmetros precisam ser medidos e avaliados em função de compromissos entre desempenho, custo e flexibilidade. Em geral, os processos de simulação utilizam modelos ideais ou quasi-ideais do ambiente onde os sistemas serão utilizados, sendo necessária a utilização de mecanismos que permitam comparar os resultados teóricos previstos e a realização prática validando os conceitos iniciais. Uma plataforma experimental facilita o trabalho do pesquisador, quando permite uma observação real do resultado de práticas e técnicas teóricas. Ao mesmo tempo as plataformas experimentais desempenham um papel importante como instrumentos didáticos em cursos de graduação e pós-graduação, quando permitem aos alunos uma comprovação prática de conceitos abstratos e contribuem para a consolidação de conhecimentos adquiridos em sala de aula.

Em [7], é apresentado um levantamento de algumas plataformas experimentais existentes em algumas instituições como Virginia Tech e UCLA, com uma classificação básica com relação a arquitetura e estrutura. Em sua maioria, as arquiteturas são adequadas para estudos na forma de coleta de dados e análises “off-line” o que limita um pouco o cenário de experimentação. Aquelas que permitem observações em tempo real são relativamente complexas e de custo elevado e com arquiteturas que tem pouca ou nenhuma relação com a estrutura do projeto final a ser implementado.

O sistema apresentado neste trabalho, apesar de simples, tem capacidade para processamento em tempo real e uma arquitetura muito semelhante a um possível sistema final. Em geral os sistemas realizáveis tem capacidade limitada de processamento e devem utilizar algum tipo de dispositivo específico para as funções de processamento de sinal, provavelmente um ASIC (Application Specific Integrated Circuit). Nesta plataforma utilizamos um FPGA (Field Programable Gate Array) como elemento de processamento de sinais. Tipicamente, os FPGAs são utilizados como protótipos de ASIC em etapas intermediárias de projeto. Portanto, essa plataforma também tem importância na validação de conceitos para o desenvolvimento de ASICs.

1.3 Escopo do projeto

Este projeto investiga um hipotético sistema reconfigurável de comunicação cuja arquitetura pode ser dividida em tres partes: Aquisição de dados, Extração de atributos e Identificação de Modulação.

O objetivo principal é o desenvolvimento e a implementação de um método de identificação automática de modulação baseado em conjunto de atributos característicos dos sinais modulados e a aplicação deste método a um demodulador universal, conforme esquematizado na figura 1.3.

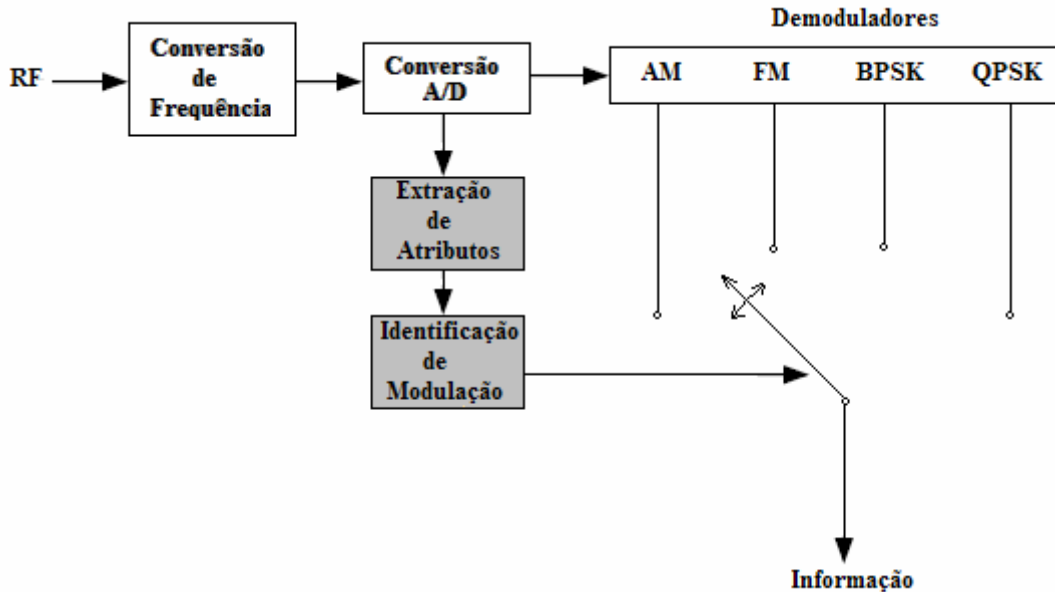


Figura 1.3: Diagrama de blocos da aplicação Demodulador Universal.

1.4 Estrutura deste documento

No capítulo 2 discorremos sobre alguns aspectos básicos da teoria de comunicações, apresentando, de forma simplificada, algumas técnicas de modulação e demodulação, e concluímos com uma introdução à tecnologia de rádios digitais implementados em software (RDS – Rádios Definidos em Software).

No capítulo 3 apresentamos a arquitetura da plataforma utilizada durante o desenvolvimento deste trabalho. São discutidos aspectos de hardware e algumas características de software são analisadas sob o contexto de software livre (Linux).

No capítulo 4 apresentamos um estudo sobre o estado da arte em técnicas de reconhecimento automático de modulação e concluímos apresentando nossa proposta para um método de reconhecimento automático de modulação em amplitude (AM), frequência (FM), fase binária (BPSK) e fase quaternária (QPSK).

No capítulo 5 apresentamos a realização de um sistema capaz de reconhecer, automaticamente, quatro tipos de modulação, aplicando o método proposto no capítulo 4.

Os resultados experimentais são comentados no capítulo 6, e nossas conclusões e recomendações para estudos futuros aparecem no capítulo 7.

Capítulo 2

Comunicação Digital

2.1 Introdução

Nos receptores digitais, as funções de demodulação e conversão de frequências são implementadas usando técnicas digitais. Um dispositivo conhecido como Digital Down Converter (DDC) é utilizado para converter o sinal de FI (Frequência Intermediária) para banda-base pela utilização de misturadores digitais e de osciladores controlados numericamente (NCO – Numerical Controlled Oscillator) além de filtros passa-baixas e um processo de decimação para redução da taxa de amostragem. A arquitetura básica pode ser vista na figura 2.1.

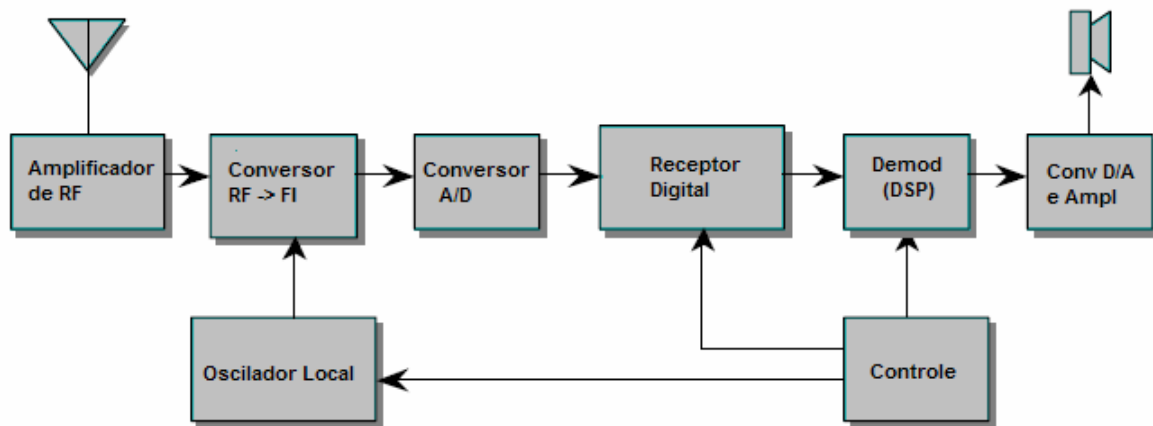


Figura 2.1: Diagrama de blocos de um receptor digital típico.

2.1.1 Conversão de Frequências

Nos receptores típicos, o sinal de FI é amostrado e convertido para digital. Algumas implementações amostram o sinal a uma taxa que satisfaz diretamente o critério de Nyquist. Outras implementações usam a técnica de sub-amostragem usando o conversor A/D como um segundo misturador convertendo o sinal para uma faixa que permita os processamentos digitais subsequentes. Essa técnica é utilizada quando a frequência intermediária é muito mais alta que a taxa de amostragem deseja considerando que a largura de banda do sinal atenda os critérios de Nyquist. O espectro do sinal amostrado está localizado em uma frequência intermediária real entre DC e $F_s/2$ e, tipicamente, contém a faixa de interesse bem como sinais não desejados. Uma vez que apenas uma porção do espectro é realmente desejada, a maior parte pode ser descartada para conservar largura de banda nos processos de demodulação subsequentes.

O receptor digital “foca” na faixa de interesse e descarta os sinais não desejados. Ver figura 2.2.

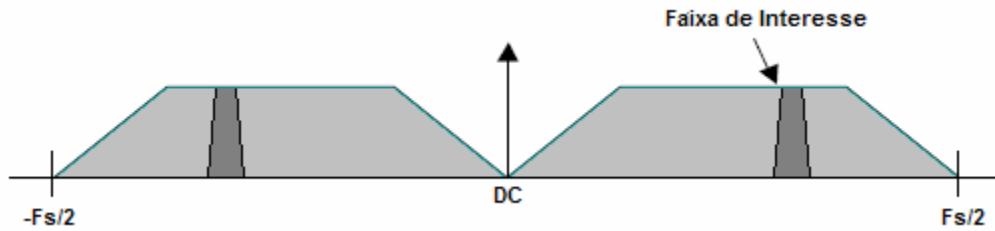


Figura 2.2: Espectro real amostrado.

O receptor digital efetua a operação de mixagem no sinal amostrado usando um Oscilador Local complexo. A frequência do OL é escolhida de forma a transladar a frequência central de interesse para DC, resultando em um sinal complexo, agora com um espectro com partes positivas e negativas. O Oscilador Local é implementado na forma de um Oscilador Digital Controlado Numericamente ou NCO, que tem a capacidade de mudar de frequência de forma instantânea em qualquer parte da faixa. Esta operação está representada na figura 2.3.

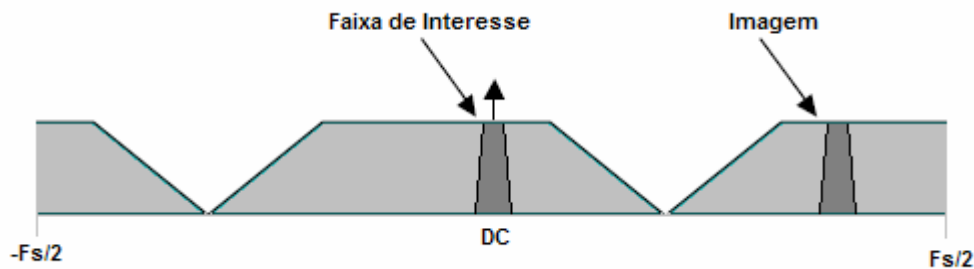


Figura 2.3: Espectro Amostrado após translação para DC.

Uma vez que o sinal tenha sido deslocado para DC, um filtro passa-baixas é utilizado para isolar a faixa de interesse. Este filtro, na verdade, é um duplo filtro que é aplicado às componentes ditas em-fase (I) e quadratura (Q), preservando as bandas laterais positivas e negativas e removendo a faixa não desejada, como podemos observar na figura 2.4.

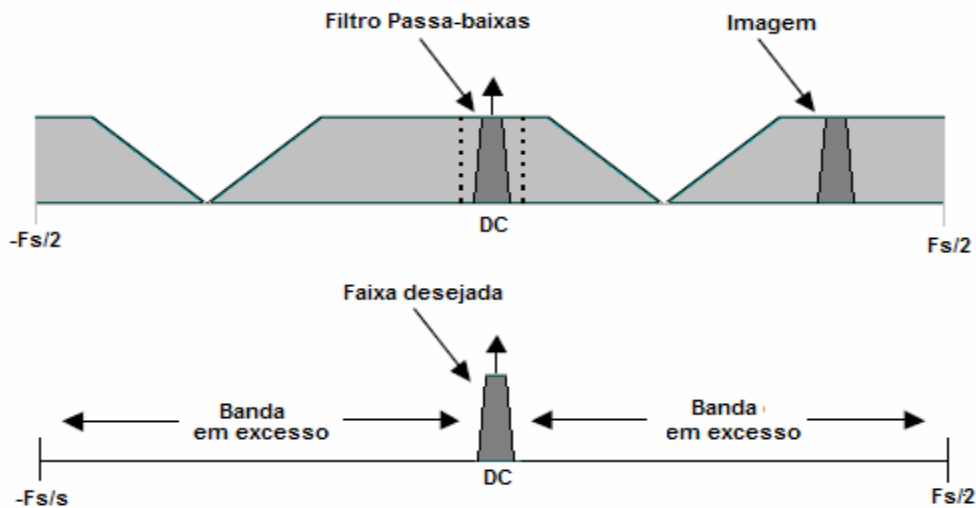


Figura 2.4: Espectro complexo após o filtro passa-baixas.

A figura 2.4 também nos mostra que o sinal filtrado está altamente sobre-amostrado (acima do critério de Nyquist). Essa característica não causa problemas mas exige uma taxa de processamento desnecessariamente alta. Portanto, para reduzir essa taxa de amostragem, o sinal deve ser sub-amostrado (“down-sampled”) para uma taxa mínima que satisfaça o critério de Nyquist sem perder informação. O resultado pode ser visto na figura 2.5. A “Banda em excesso” é simplesmente descartada no processo de decimação que ignora determinadas amostras reduzindo a taxa de amostragem identificada como F_s . O fator de decimação, N , é determinado pela relação entre a taxa de amostragem original e a nova taxa de amostragem, minimizando a carga de processamento necessária para os algoritmos subsequentes.

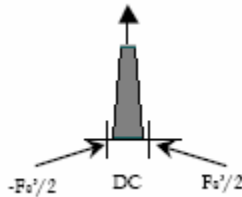


Figura 2.5: Espectro do sinal após o processo de decimação.

Uma vez que o sinal de interesse tenha sido convertido para um sinal complexo de faixa-estreita, centrado em DC (banda-base), ele pode ser demodulado. Algoritmos simples podem ser utilizados para processar formatos comuns de modulação como AM, ASK, FM, FSK e PM. Esses algoritmos podem ser implementados diretamente no Receptor Digital ou no DSP ou mesmo em um GPP - General Purpose Processor (Processador de Uso Geral, PC por exemplo).

A demodulação de AM (Amplitude Modulada) consiste simplesmente na extração do envelope do sinal complexo em banda-base. Um algoritmo de estimação de magnitude vetorial (CORDIC, por exemplo) [8] pode ser utilizado para calcular esse envelope.

Outros modos do algoritmo CORDIC ou uso de tabelas em ROM podem ser usados para calcular o arco da tangente do sinal complexo para implementar um demodulador de fase (PSK).

A diferença entre amostras adjacentes após a saída do demodulador de fase fornece uma função de demodulação FM.

Esquemas mais complexos de modulação, geralmente, fazem do uso do DSP para implementar algoritmos mais sofisticados como FFT, por exemplo.

2.2 Princípios de Modulação

Modulação é o processo de modificação ou alteração de um sinal padrão chamado de *portadora*, em função de outro sinal que contém a informação a ser transmitida [9].

Em sistemas de comunicação, os sinais são transmitidos através de um determinado meio físico denominado *canal* [10]. Os canais de comunicação têm larguras de banda limitadas, o que restringe as características de frequência do sinal a ser transmitido. O sinal de informação, chamado de *sinal em banda-base*, contém componentes de baixa frequência e grandes comprimentos de onda, impossibilitando sua transmissão direta e, por isso, dependendo do meio utilizado, pode ser necessário a utilização de um sinal senoidal de alta frequência como *transportador* (daí o termo *portadora*) da informação.

1. No transmissor, um sinal senoidal puro, chamado de portadora, é gerado;
2. A informação a ser transmitida é usada para modificar uma ou mais características da portadora. Qualquer modificação *detectável* pode *transportar* a informação;
3. No receptor, as modificações ou alterações da portadora são detectadas e a informação é recuperada.

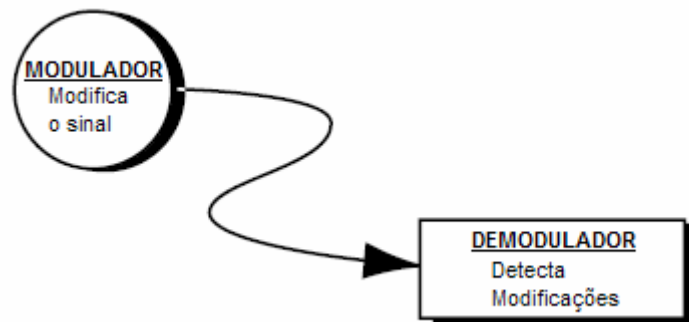


Figura 2.6: Conceito simplificado de modulação.

2.2.1 Características modificáveis da Portadora

Existem apenas três características da portadora que podem ser modificadas em função do sinal de informação a ser transmitido: Amplitude, Frequência e Fase. Podemos dividir os tipos de modulação em dois grandes grupos dependendo do sinal modulador: se o sinal modulador for contínuo no tempo, dizemos ser uma Modulação Analógica; se o sinal for composto de um conjunto discreto de símbolos, ela é dita Modulação Digital.

Em Modulação de Amplitude (AM) e suas derivadas (DSB-SC – Double Side Band – Supressed Carrier e VSB – Vestigial Side Band), a amplitude da portadora é variada proporcionalmente em relação à amplitude instantânea do sinal modulador.

Frequência Modulada (FM) é a técnica mais popular de modulação analógica por sua característica de boa imunidade a ruído. Em FM, a amplitude da portadora é mantida constante enquanto sua frequência é variada em função do sinal modulador. A Modulação em Fase é apenas outra forma de ver Modulação em Frequência uma vez que Fase e

Frequência estão relacionadas. Essas formas básicas de modulação estão esquematizadas na figura 2.7.

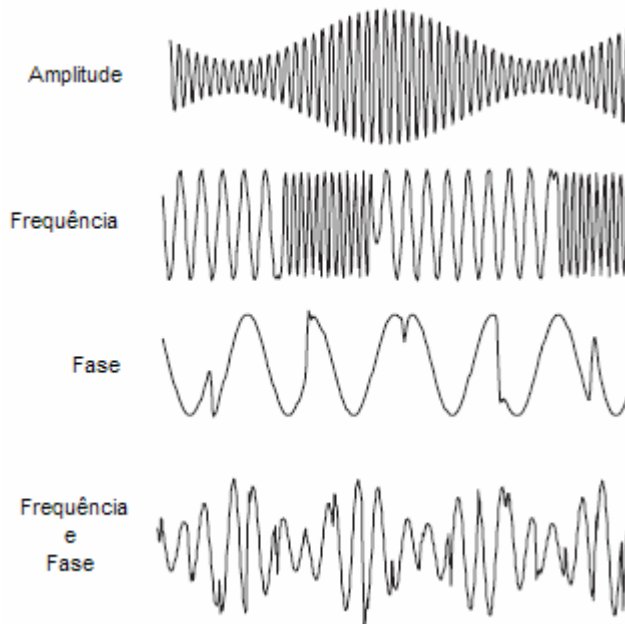


Figura 2.7: Esquemas básicos de modulação.

O grupo de modulação analógica pode, ainda, ser classificado como linear ou exponencial (também conhecida como Modulação angular) [11], e suas propriedades estão caracterizadas na tabela 1, onde *envelope* significa a envoltória do sinal, W representa a largura de banda do sinal e SNR representa a relação sinal-ruído.

Tabela 1: Propriedades das modulações linear e exponencial.

	LINEAR	EXPONENCIAL
Métodos	AM, DSB, SSB, VSB	FM, PM
Envelope	Depende do Sinal Modulador	Constante
Espectro	Deslocado em Frequência	Complexo
Largura de banda	$\leq 2W$	$> 2W$
SNR	Depende da Potência de transmissão	Depende da largura de banda

Na Tabela 1, observa-se que o índice SNR depende da potência de transmissão em AM e depende da largura de banda em FM. Como o controle da largura de banda é menos crítico que o controle da potência, é preferível o uso de modulação FM a modulação AM.

2.2.2 Representação de Sinais em Banda-Base

Seja um sinal real modulado, dito *em banda-passante*, $s(t) = a(t)\cos(2\pi f_c t + \theta(t))$, onde $a(t)$ representa a envoltória do sinal modulante, $\theta(t)$ representa fase do sinal e f_c representa a frequência da portadora.

Este sinal pode ser decomposto em suas componentes *em-fase* $s_I(t) = a(t)\cos(\theta(t))$ e *em-quadratura* $s_Q(t) = a(t)\sin(\theta(t))$,

$$s(t) = s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t) \quad (2.1)$$

$s_I(t)$ e $s_Q(t)$ são componentes em *banda-base* enquanto que $s(t)$ é um sinal em *banda-passante* com cujo espectro encontra-se nas vizinhanças de f_c . A envoltória complexa de $s(t)$, que é a representação em *banda-base* de $s(t)$, é o sinal complexo.

$$\tilde{s}(t) = s_I(t) + js_Q(t) \quad (2.2)$$

A partir de (2.2), definimos o sinal analítico $s_A(t)$, como

$$s_A(t) = \tilde{s}(t)e^{j2\pi f_c t} \quad (2.3)$$

cuja parte real é o sinal $s(t) = \Re[s_A(t)] = \Re[\tilde{s}(t)e^{j2\pi f_c t}]$. Este processo pode ser esquematizado conforme as figuras 2.8 e 2.9.

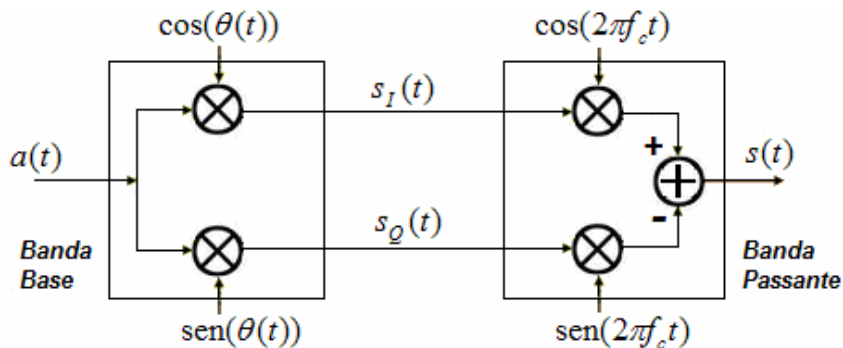


Figura 2.8: Componentes *Em-Fase* e *Em-Quadratura*.

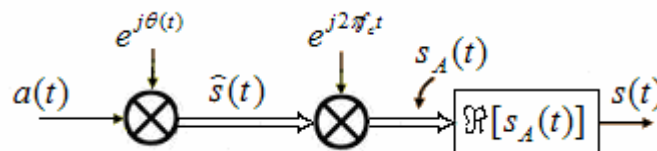


Figura 2.9: Envoltória complexa e sinal analítico.

2.2.3 Representação I/Q

Em comunicações Digitais é conveniente a representação de sinais em termos de suas componentes *Em-fase(I)* e *Em-Quadratura(Q)*. Esta é uma representação retangular de um diagrama polar, onde o eixo "I" situa-se em zero grau (referência) e o eixo "Q" encontra-se a 90 graus. A projeção do vetor do sinal no eixo "I" é sua componente *Em-fase (I)*, e a projeção do mesmo vetor no eixo "Q" é sua componente *Em-quadratura (Q)*, como pode ser visto na figura 2.10.

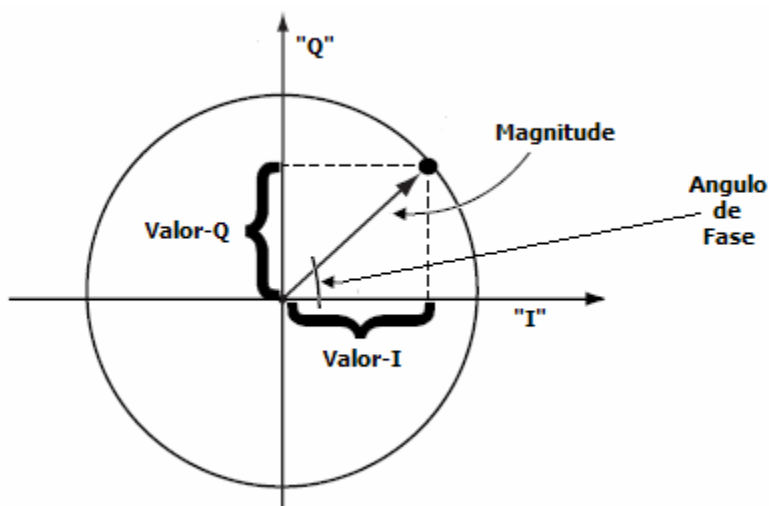


Figura 2.10: Equivalência Polar-Retangular.

Essa representação é particularmente interessante porque é facilmente relacionável com a forma como a maioria dos sinais são gerados em um modulador digital. Em um transmissor, sinais I e Q são *misturados* (produto entre dois sinais) com o mesmo oscilador local. Uma defasagem de 90° é forçada em um dos *braços* do oscilador, fazendo com que os produtos (I x LO) e (Q x LO) sejam ortogonais, ou estejam *em quadratura*. A principal característica desse arranjo, mostrado na figura 2.11, é que esses sinais não interferem um com outro – eles representam duas componentes independentes do sinal de informação.

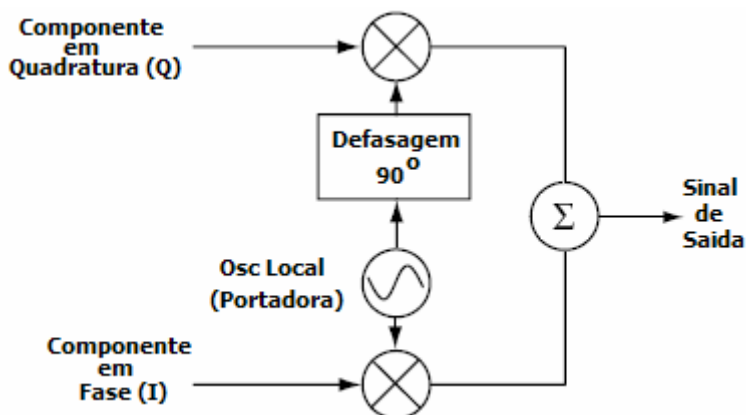


Figura 2.11: Componentes I e Q na transmissão.

Ao chegar no receptor, esse sinal é novamente *misturado* com um oscilador local (com frequência igual a da portadora). Esse oscilador local gera um sinal com fase arbitrária igual a zero e outro deslocado de 90 graus. Neste processo, o sinal composto de entrada (magnitude e fase) é decomposto em suas componentes *Em-fase (I)* e *Em-quadratura (Q)*. Esses componentes são independentes e ortogonais, onde alterações em um sinal são independentes de alterações no outro sinal. O processo esquematizado na figura 2.12 representa uma conversão polar-retangular, que é o processo utilizado nos receptores digitais.

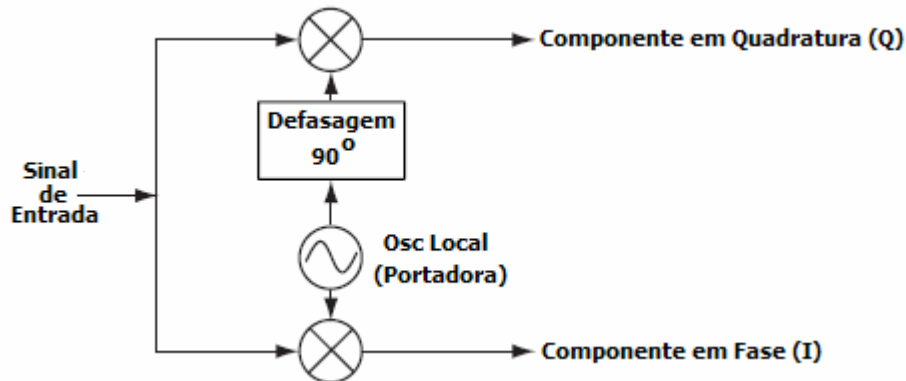


Figura 2.12: Componentes I e Q na recepção.

2.2.4 – Modulação Analógica

A alteração de parâmetros da portadora, de forma contínua no tempo, está presente em diversas formas de modulação analógica como, por exemplo, Amplitude Modulada (AM), Frequência Modulada (FM), Amplitude Modulada com Banda Lateral Superior (AM-USB), Amplitude Modulada com Banda Lateral Inferior (AM-LSB), Amplitude Modulada com portadora suprimida (DSB-SC) e outras. Por uma questão de espaço e tempo, neste projeto apenas as formas de modulação AM e FM são consideradas.

2.2.4.1 – Modulação em Amplitude (AM)

Utilizando-se de uma onda portadora representada por $A_c \cos(2\pi f_c t)$ e um sinal modulador representado por $A_m \cos(2\pi f_m t)$, o processo de modulação resulta em um sinal com amplitude igual a $A_c + A_m \cos(2\pi f_m t)$ e o sinal resultante, modulado em amplitude, pode ser representado por

$$(A_c + A_m \cos(2\pi f_m t)) \cos(2\pi f_c t) \quad (2.4)$$

reescrevendo, $S_{AM}(t) = A_c (1 + m \cos(2\pi f_m t)) \cos(2\pi f_c t)$ (2.5)

onde a constante 'm' é chamada de *índice de modulação*, e controla o nível de potência que estará presente nas bandas laterais espectrais. O processo pode ser ilustrado através da figura 2.13, onde LO representa a portadora com frequência angular $\omega_c = 2\pi f_0$.

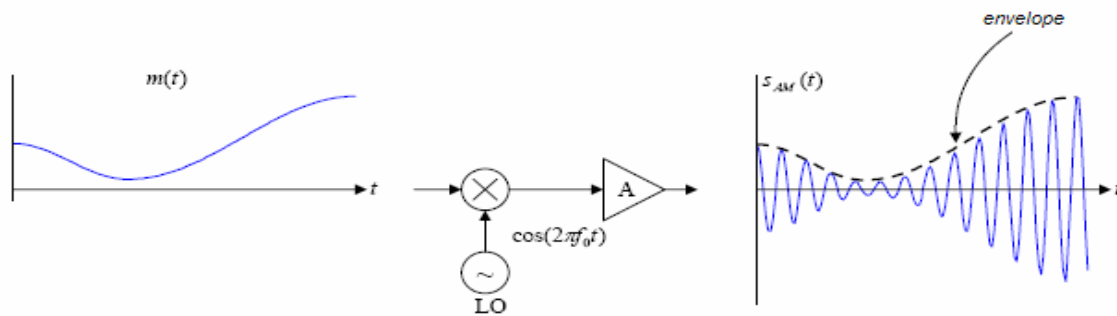


Figura 2.13: Modulação AM no domínio do tempo.

O espectro do sinal contendo a mensagem de informação, originalmente centrado na origem, é deslocado para $-f_0$ e f_0 , como poder ser visto na figura 2.14.

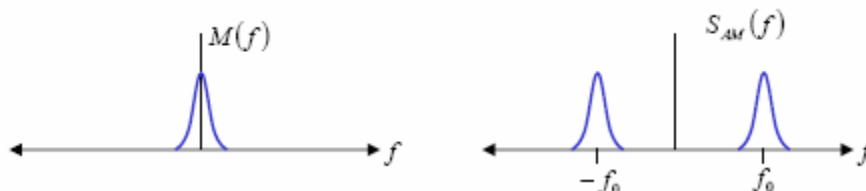


Figura 2.14: Modulação AM (relação espectral).

2.2.4.2 Modulação em Frequência (FM)

Tal qual no item 2.2.4.1, tomemos uma portadora igual a $A_c \cos(2\pi f_c t)$ e um sinal modulante (mensagem) igual $A_m \cos(2\pi f_m t)$. Se o sinal modulante for utilizado para variar a frequência da portadora, teremos um sinal em FM, cuja frequência instantânea é dada por

$$f = f_c + k_f A_m \cos(2\pi f_m t) \quad (2.6)$$

onde a quantidade $k_f A_m$ representa o desvio máximo de frequência e é denominado Δ_f .

Sabendo que $\omega = \frac{d\theta}{dt}$, podemos escrever $\frac{d\theta}{dt} = 2\pi f_c + 2\pi \cos(2\pi f_m t)$ (2.7)

e $\theta = 2\pi f_c t + \beta \sin(2\pi f_m t)$ (2.8)

onde o termo $\beta = \frac{\Delta_f}{f_m}$ é chamado de índice de modulação em FM.

O sinal resultante modulado em frequência pode ser representado por $s_{FM}(t) = A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t))$ (2.9)

O processo de modulação FM está mostrado, de forma simplificada, na figura 2.15, onde o VCO (Voltage Controlled Oscillator) simboliza um oscilador cuja frequência central f_0 varia conforme a tensão de entrada.

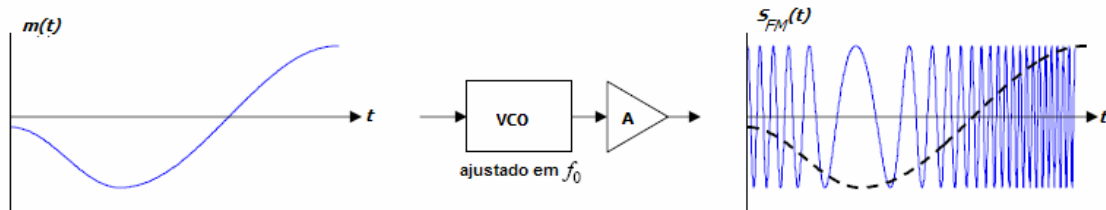


Figura 2.15: Modulação FM.

A distribuição espectral do sinal de FM está delineada na figura 2.16, onde podemos observar uma maior variação em torno de f_0 , quando comparamos com modulação AM.

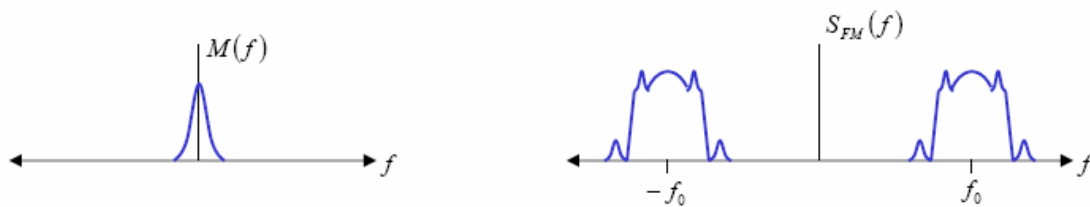


Figura 2.16: Espectro resultante da modulação FM

2.2.5 Modulação Digital

As principais formas de modulação digital são FSK (Frequency Shift Keying), onde a frequência da portadora tem seu valor alterado entre dois ou mais (M-ary FSK) valores discretos, PSK (Phase Shift Keying), onde o sinal modulante alterna a fase da portadora, e QAM (Quadrature Amplitude Modulation), onde a fase e a amplitude da portadora são alteradas simultaneamente.

Neste projeto são consideradas apenas as formas de modulação BPSK (M-Ary PSK com $M=2$) e QPSK (M-Ary PSK com $M=4$).

2.2.5.1 Modulação BPSK

Na modulação BPSK, a fase da portadora tem seu valor alternado entre 0 e π em função da informação binária (0 e 1) que tenha sido mapeada previamente para (-1 e +1).

Na figura 2.17 podemos observar que a fase da portadora se alterna de acordo com as variações da mensagem.

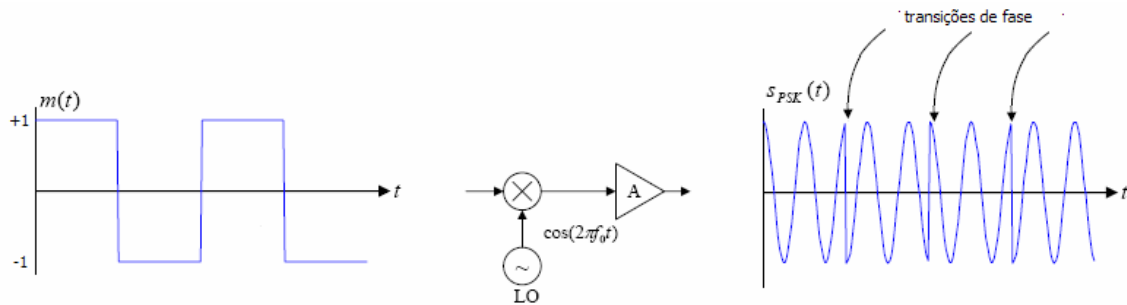


Figura 2.17: Processo de modulação BPSK.

Dada uma portadora senoidal com frequência f_c , um sinal $s(t)$ modulado digitalmente pode ser expresso por

$$s(t) = a(t) \cos(2\pi f_c t + \theta(t)) \quad (2.10)$$

onde $a(t)$ representa uma modulação em amplitude e $\theta(t)$ representa uma modulação em fase. Em modulação digital, a amplitude é mantida constante enquanto a fase é variada.

Portanto devemos escolher dois valores distintos de $\theta(t)$ para representar os símbolos 0 (zero) e 1 (hum). Escolhendo $\theta_1(t) = 0$ como o valor de $\theta(t)$ que representa 1 (hum) e $\theta_0(t) = \pi$ como o valor de $\theta(t)$ que representa 0 (zero), podemos escrever

$$s_0(t) = \sqrt{E_s} \cos(2\pi f_c t + \pi) \quad (2.11)$$

$$s_1(t) = \sqrt{E_s} \cos(2\pi f_c t) \quad (2.12)$$

ou

$$s_0(t) = -\sqrt{E_s} \cos(2\pi f_c t) \quad (2.13)$$

$$s_1(t) = \sqrt{E_s} \cos(2\pi f_c t) \quad (2.14)$$

onde $\sqrt{E_s}$ é o valor de pico da amplitude do sinal modulado e $s_0(t)$ e $s_1(t)$ são as representações BPSK dos símbolos digitais 0 (zero) e 1 (hum) respectivamente.

Na figura 2.18 está ilustrada a constelação BPSK, também conhecida como *espaço de sinal* [12], resultante do processo, onde se pode observar a delimitação da localização válida para os símbolos s_0 e s_1 (área achurada).

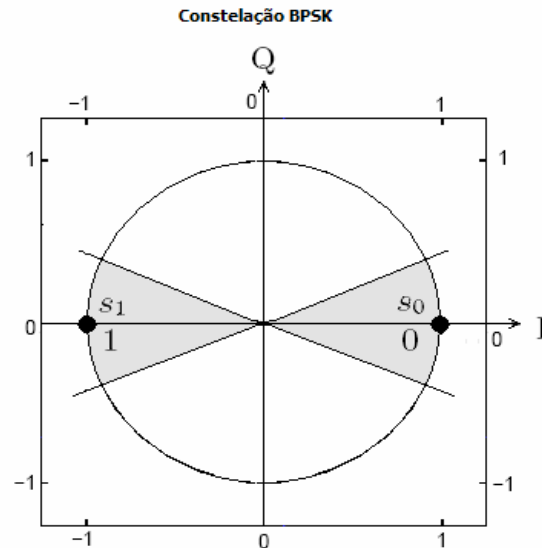


Figura 2.18: Constelação mostrando as regiões válidas para sinais BPSK.

O processo de recuperação da informação pode ser ilustrado conceitualmente de acordo com a figura 2.19, onde, no receptor, o sinal modulado BPSK é multiplicado por uma portadora com frequência e fase iguais às do transmissor. O resultado, após integrado discretamente, é utilizado para determinação do bit originalmente enviado.

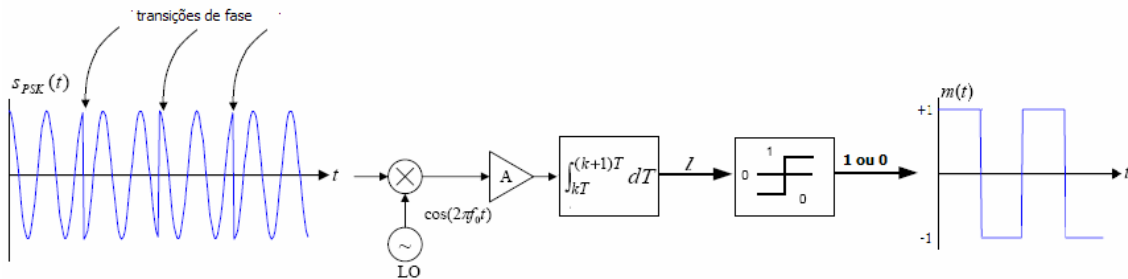


Figura 2.19: Demodulação conceitual BPSK.

$$\text{Analicamente, a partir do sinal BPSK } s_{BPSK}(t) = \sqrt{E_s} \cos(2\pi f_c t + \theta(t)) \quad (2.15)$$

$$\text{e considerando o sinal do oscilador local } LO(t) = \sqrt{E_s} \cos(2\pi f_c t) \quad (2.16)$$

temos, na entrada do integrador, $r(t) = LO(t) S_{BPSK}(t)$, expresso por

$$r(t) = \sqrt{E_s} \cos(2\pi f_c t + \theta(t)) \cdot \sqrt{E_s} \cos(2\pi f_c t) \quad (2.17)$$

que pode ser simplificado em $r_I(t)$ (parte real, em fase) com $r_Q(t)$ igual a zero.

$$r_i(t) = \frac{E_s}{2} [\cos(2\pi 2f_c t + \theta(t)) + \cos(\theta(t))] \quad (2.18)$$

a componente $2f_c$ é removida pelo filtro passa-baixa (Integrador), resultando apenas no sinal

$$r_i(t) = \frac{E_s}{2} \cos(\theta(t)) \quad (2.19)$$

que é positivo para $\theta(t) = 0$ e negativo para $\theta(t) = \pi$, como pode ser visto na figura 2.20.

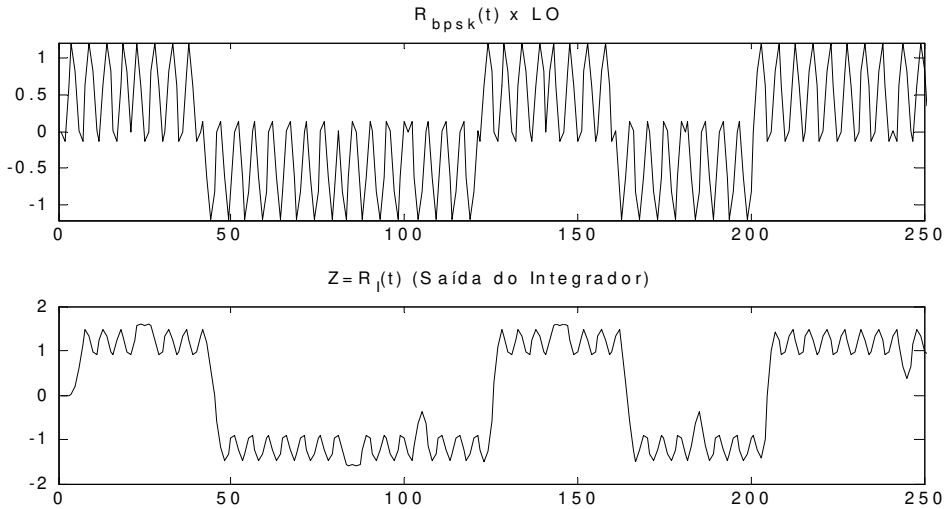


Figura 2.20: Demodulação BPSK.

2.2.5.2 Modulação QPSK

Se os bits forem agrupados dois a dois (agrupamentos conhecidos como *dibits*) e esses pares de bits forem utilizados para escolher um dentre um conjunto de fases discretas, teremos um sistema de modulação QPSK (*Quaternary Phase Shift Keying*).

Em QPSK temos 4 símbolos, onde cada símbolo representa um *dibit* em particular. Portanto, devemos selecionar 4 valores para $\theta(t)$, nosso parâmetro variável para modulação. A tabela 2 apresenta um possível conjunto de fases associados a cada um dos *dibits*.

Tabela 2: Correspondência entre *dibits* e ângulos de fase em QPSK.

<i>dibit</i>	$\theta(t)$
00	$+\pi/4$
01	$+3\pi/4$
10	$-\pi/4$
11	$-3\pi/4$

Esse mapeamento, resulta nos seguintes expressões para os sinais QPSK:

$$s_{00}(t) = \sqrt{E_s} \cos(2\pi f_c t + \frac{\pi}{4}) \quad (2.20)$$

Uma portadora em quadratura pode ser vista como uma exponencial complexa, $e^{jw_c t}$, bem como uma soma equivalente de senóides em fase e em quadratura, $\cos(w_c t) + j \text{sen}(w_c t)$. Portanto usando a identidade $\cos(a + b) = \cos(a) \cos(b) - \text{sen}(a) \text{sen}(b)$, podemos reescrever (2.20) como:

$$s_{00}(t) = \frac{\sqrt{2E_s}}{2} \cos(2\pi f_c t) + \frac{\sqrt{2E_s}}{2} \text{sen}(2\pi f_c t) \quad (2.21)$$

da mesma forma, expressamos $s_{01}(t)$, $s_{10}(t)$ e $s_{11}(t)$:

$$s_{01}(t) = -\frac{\sqrt{2E_s}}{2} \cos(2\pi f_c t) + \frac{\sqrt{2E_s}}{2} \text{sen}(2\pi f_c t) \quad (2.22)$$

$$s_{10}(t) = -\frac{\sqrt{2E_s}}{2} \cos(2\pi f_c t) - \frac{\sqrt{2E_s}}{2} \text{sen}(2\pi f_c t) \quad (2.23)$$

$$s_{11}(t) = \frac{\sqrt{2E_s}}{2} \cos(2\pi f_c t) - \frac{\sqrt{2E_s}}{2} \text{sen}(2\pi f_c t) \quad (2.24)$$

Um modelo conceitual de modulador QPSK pode ser visto na figura 2.21.

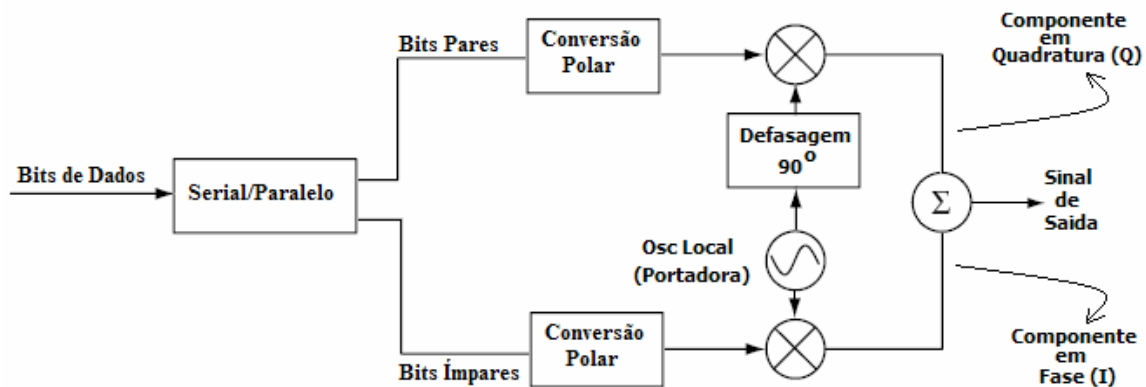


Figura 2.21: Diagrama de blocos de um Modulador QPSK.

Na figura 2.22 está ilustrado como o sinal QPSK é gerado como resultado da soma de duas senóides em quadratura, moduladas em fase pelos bits Ímpares (I) e Pares (Q).

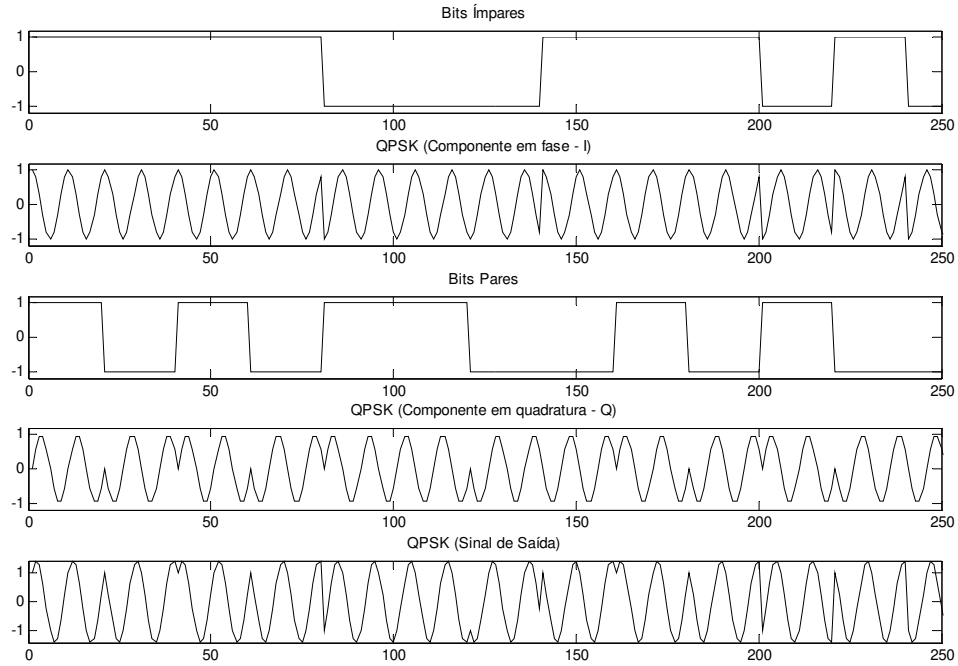


Figura 2.22: Sinais gerados pelo modulador QPSK.

A representação na forma de diagrama polar (ou diagrama de constelação) pode ser vista na figura 2.23, onde as áreas achuradas indicam as regiões válidas para os símbolos s_{iq} .

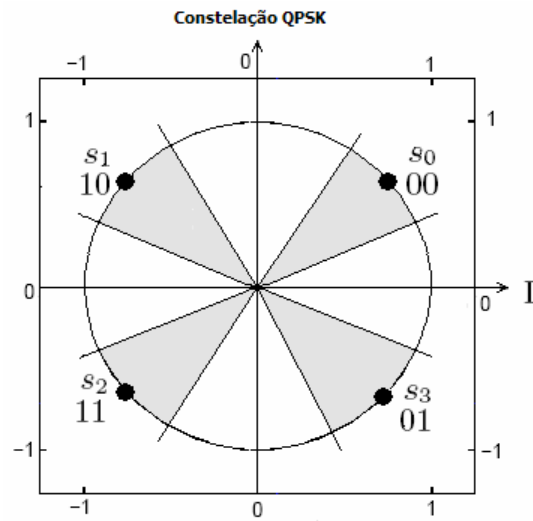


Figura 2.23: Constelação QPSK.

O processo de recuperação da informação QPSK é similar ao da modulação BPSK considerando, agora, que ambas as componentes, $r_I(t)$ e $r_Q(t)$, devem ser consideradas no processo. Um possível demodulador pode ser visto na figura 2.24.

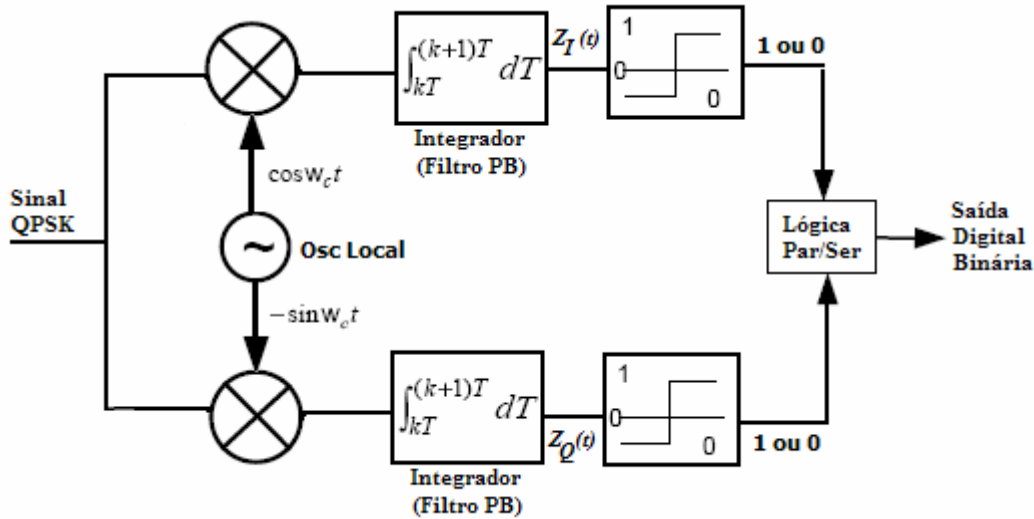


Figura 2.24: Diagrama de blocos de um Demodulador QPSK.

$$\text{A partir do sinal QPSK } s_{QPSK}(t) = \sqrt{E_s} \cos(2\pi f_c t + \theta(t)) \quad (2.25)$$

e considerando o sinal do oscilador local

$$LO(t) = \sqrt{E_s} e^{-j2\pi f_c t} = \sqrt{E_s} \cos(2\pi f_c t) - j\sqrt{E_s} \sin(2\pi f_c t) \quad (2.26)$$

temos, na entrada dos integradores, $r(t) = LO(t) S_{QPSK}(t)$, expresso por

$$r(t) = \sqrt{E_s} \cos(2\pi f_c t + \theta(t)) \sqrt{E_s} e^{-j2\pi f_c t} \quad (2.27)$$

que pode ser simplificado em $r_I(t)$ (parte real) e $r_Q(t)$ (parte imaginária),

$$r_I(t) = \frac{E_s}{2} [\cos(2\pi 2f_c t + \theta(t)) + \cos(\theta(t))] \quad (2.28)$$

$$r_Q(t) = \frac{E_s}{2} [-\sin(2\pi 2f_c t + \theta(t)) - \sin(-\theta(t))] \quad (2.29)$$

onde as componentes $2f_c$ são removidas pelos filtros passa-baixas (Integradores),

$$\text{resultando em } z_I(t) = \frac{E_s}{2} \cos(\theta(t)) \quad (2.30)$$

e

$$z_Q(t) = \frac{E_s}{2} \text{sen}(\theta(t)) \quad (2.31)$$

cujas formas de onda podem ser vistas na figura 2.25.

A informação de fase pode ser recuperada através da relação $\theta(t) = \arctan(z_Q(t)/z_I(t))$.

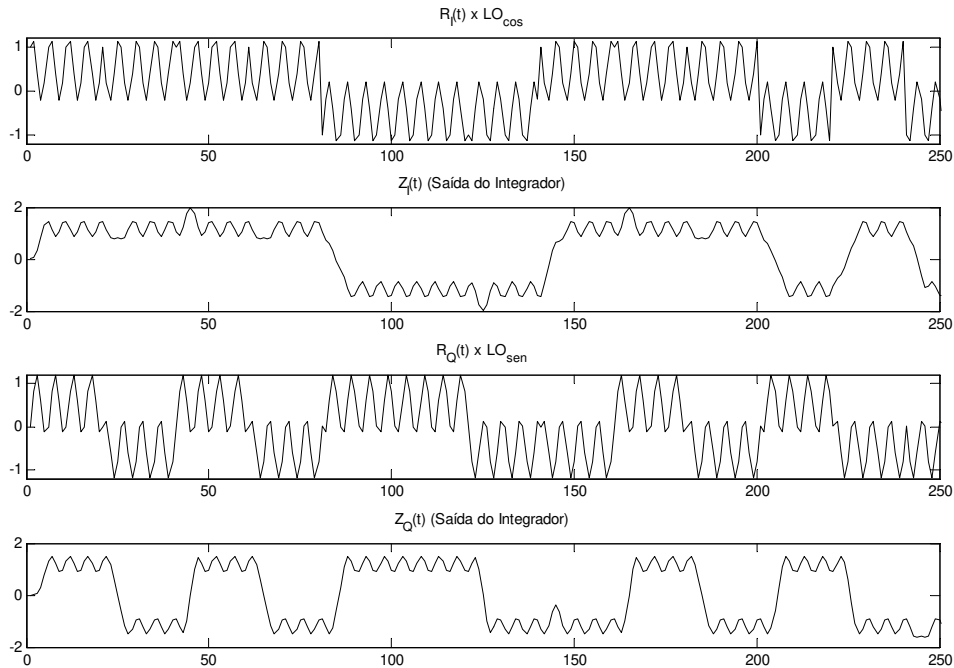


Figura 2.25: Demodulação QPSK.

2.3 Rádio Definido por Software

O termo *Rádio Definido por Software* ou *RDS* refere-se a uma classe de rádios reconfiguráveis onde o comportamento de sua camada física pode ser significativamente alterado sem modificações em seu hardware [13] e foi cunhado por Joseph Mitola em 1991. Os sistemas de Software-Rádio oferecem grande flexibilidade em termos de implementação e reconfigurabilidade uma vez que as funções básicas de transmissão e recepção (modulação e demodulação) são implementadas em software.

Idealmente, a digitalização do sinal recebido ocorre o mais próximo possível da antena e todo o processamento subsequente é efetuado por software, conforme pode ser visto na figura 2.26 que representa um diagrama de blocos de um RDS ideal.

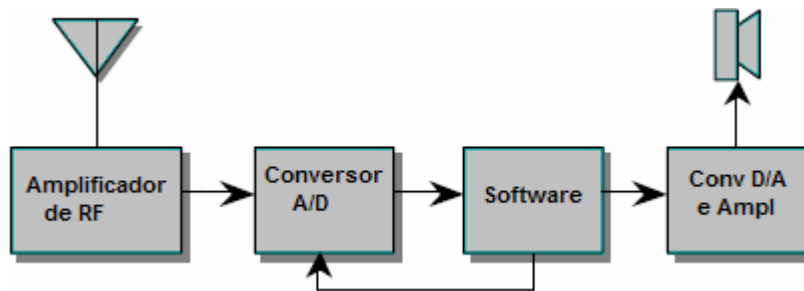


Figura 2.26: Diagrama de Blocos de um RDS ideal.

No atual estágio tecnológico os conversores analógico-digital ainda tem limitações de velocidade e não conseguem converter sinais na faixa de Gigahertz, por exemplo. Um RDS realizável está ilustrado na figura 2.27, onde as formas de onda específicas de acordo com a modulação pretendida são geradas em software na forma de amostras e enviadas a um conversor digital-analógico (DAC). O DAC converte a forma de onda em um sinal analógico que é convertido em frequência por uma etapa de RF (Rádio-Frequência). No receptor, o sinal de RF é convertido para FI (Frequência intermediária) por um módulo de RF. Este sinal de FI é, então, digitalizado por um conversor analógico-digital (ADC) e as amostras digitais passam a ser processados por software por algoritmos apropriados de demodulação.

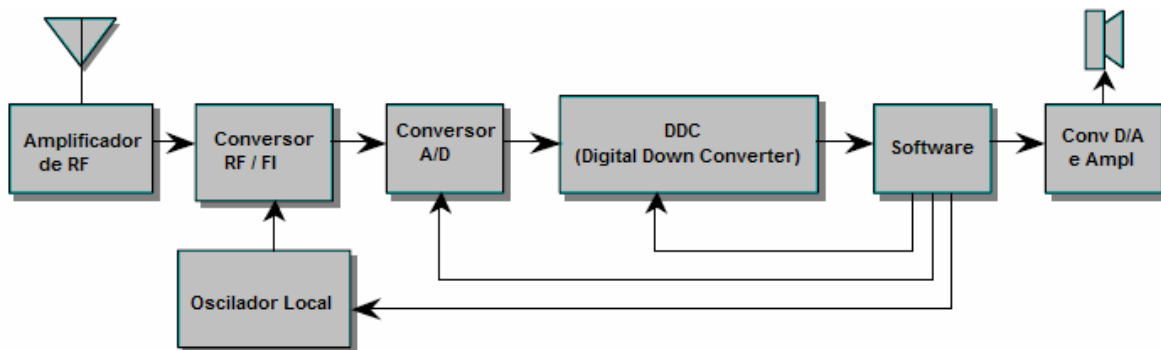


Figura 2.27: Arquitetura prática para um RDS.

2.4 GNU-Radio

A partir do projeto SpectrumWare do MIT, objeto de uma tese de doutorado [14] cujo conceito básico está representado na figura 2.28, nasceu o projeto de software livre GNU-Radio [15] que é mantido por Eric Blossom com contribuições de uma comunidade bastante ativa de desenvolvedores e interessados na tecnologia de RDS.

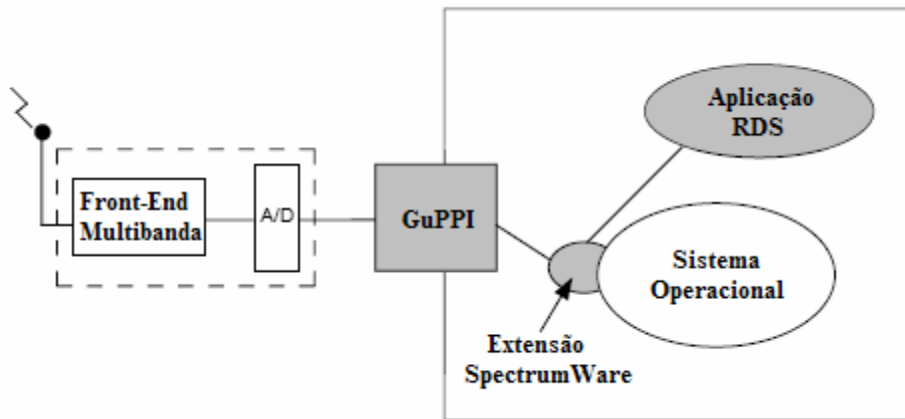


Figura 2.28: Conceito que deu origem ao GNU-Radio.

A plataforma GNU-Radio, a exemplo de outras plataformas para RDS, adota uma arquitetura baseada em camadas para o desenvolvimento de aplicações. No GNU-Radio o núcleo principal é implementado na forma de blocos primitivos de processamento de sinais e está escrito em C++ visando velocidade e robustez, enquanto que as interfaces e os controles de configuração e fluxo são implementadas em Python. O código Python, essencialmente cria grafos que controlam o fluxo de informações entre blocos de código que, por sua vez, implementam funções de processamento de sinais. Esses grafos conectam blocos produtores de informação (*sources*) a blocos consumidores (*sinks*) representando a dinâmica da aplicação. Os scripts Python criados pelo usuário são responsáveis por criar, executar e gerenciar os grafos. Esta abordagem está ilustrada, de forma simplificada, na figura 2.29.

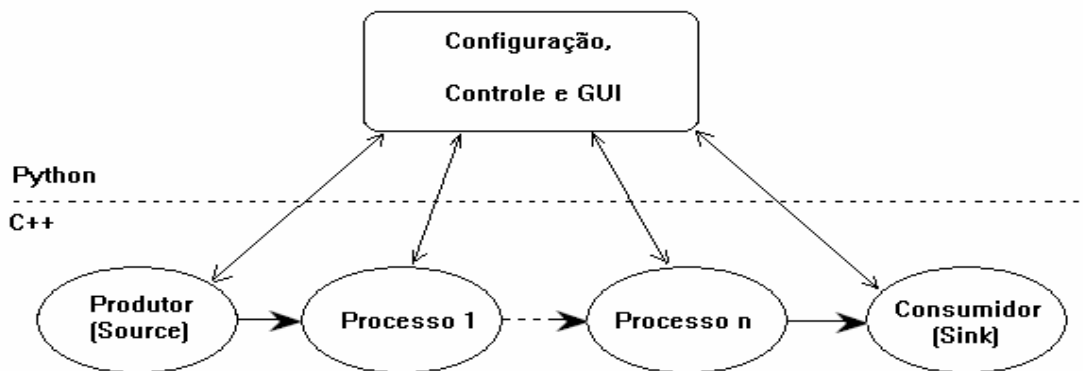


Figura 2.29: Abstração e Controle em GNU-Radio.

O sistema GNU-Radio pode ser complementado com *front-ends* como um sintonizador de TV a cabo [16], adaptadores para conversão analógico-digital como o PCI DAS card [17] e, principalmente, uma plataforma de hardware denominada Universal Software Radio Platform (USRP) [18]. Uma placa de som, utilizada para entrada e saída das aplicações também pode servir como um simples front-end analógico para recepção AM, por exemplo. Além dos exemplos que acompanham a plataforma, diversos outros projetos estão em

andamento como, por exemplo, HDTV, equivalente TiVo para rádio, rádio-astronomia, GPS em software, antenas inteligentes [19], etc.

2.5 Exemplo de aplicação GNU-Radio

Como exemplo de utilização da plataforma de hardware USRP e do kit de desenvolvimento GNU-Radio, apresentamos uma simples aplicação que implementa um receptor de rádio FM comercial.

Nesta aplicação, o sinal de FM é acoplado através de uma antena telescópica conectada à entrada do adaptador RX do USRP, conforme esquematizado na figura 2.30.

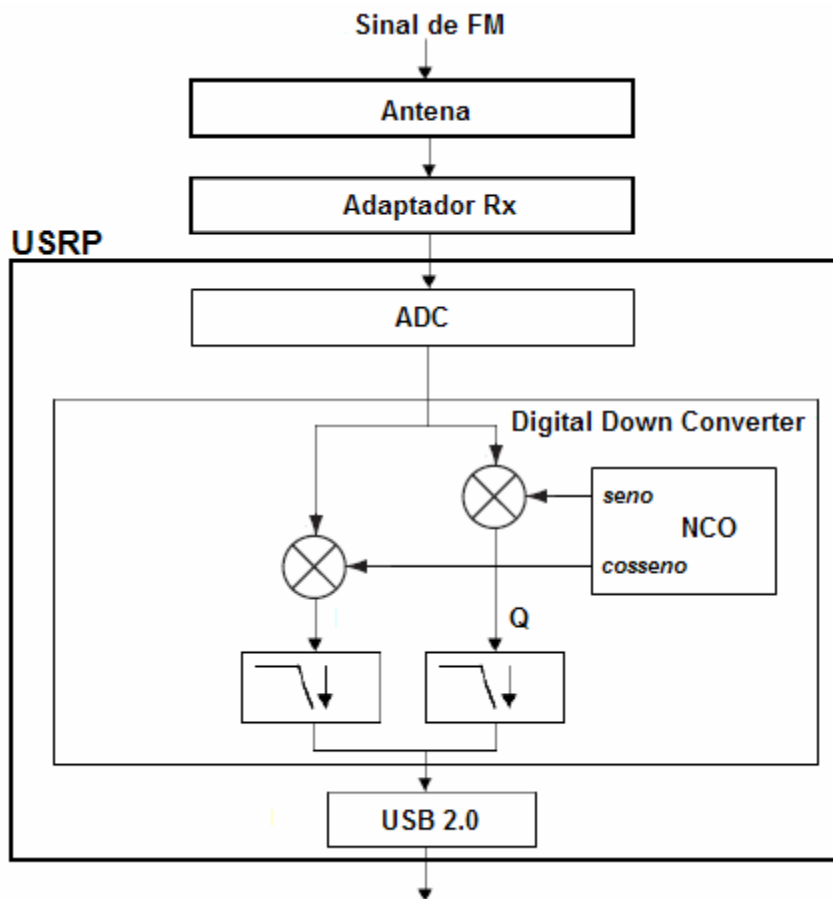


Figura 2.30: Fluxo de sinais em uma aplicação GNU-Radio/USRP.

Uma vez condicionado pelo adaptador RX, o sinal de RF é digitalizado pelo conversor A/D. Em seguida as amostras digitais são entregues ao bloco DDC, residente no FPGA, onde as amostras são multiplicadas por um sinal complexo (seno e co-seno) gerado pelo oscilador digital NCO (Numerically Controlled Oscillator) com frequência igual à FI do canal de FM desejado. O resultado é composto de uma componente de alta frequência centrada em 2 vezes a frequência de FI e uma componente em baixa frequência centrada em zero hertz. Os filtros passa-baixas eliminam a componente de alta frequência e,

simultaneamente efetuam a decimação, reduzindo a taxa de amostragem do sinal. O resultado desse processo é a conversão de frequências de RF/FI para banda-base. Esse sinal de baixa frequência é entregue, na forma complexa, à etapa de DSP para demodulação, implementada em software no PC.

O sinal de FM na entrada, representado pela equação 2.9, apresentada na seção 2.2.4.2, é transformado pelo DDC em um sinal em quadratura representado por

$$s_{FM_DDC}[n] = A_c \cos\left(\beta \operatorname{sen}\left(2\pi \frac{f_m}{f_s} n\right)\right) + j \left[A_c \operatorname{sen}\left(\beta \operatorname{sen}\left(2\pi \frac{f_m}{f_s} n\right)\right) \right] \quad (2.32)$$

$$s_{FM_DDC}[n] = A_c e^{j \left[\beta \operatorname{sen}\left(2\pi \frac{f_m}{f_s} n\right) \right]} \quad (2.33)$$

onde f_s é a frequência de amostragem do conversor A/D.

Considerando que podemos definir frequência instantânea como a derivada do ângulo de fase de um sinal e que, para sinais discretos, podemos simplificar a derivada como diferença do ângulo entre amostras sucessivas, o próximo passo no processamento do sinal de FM (demodulação) pode ser esquematizado conforme a figura 2.31 que implementa o algoritmo de demodulação em quadratura, que em sua forma mais simples consiste em multiplicar cada amostra pelo conjugado da amostra subsequente e calcular o arco da tangente do resultado, conforme as equações 2.34 e 2.35.

$$Audio_{FM} = \arg(Y) = \arg(s_{FM_DDC}[n] - s_{FM_DDC}[n-1]) \quad (2.34)$$

$$Audio_{FM} = \arg(s_{FM_DDC}[n] \cdot s_{FM_DDC}^*[n-1]) \quad (2.35)$$

onde $s_{FM_DDC}^*$ representa o complexo conjugado de s_{FM_DDC} .

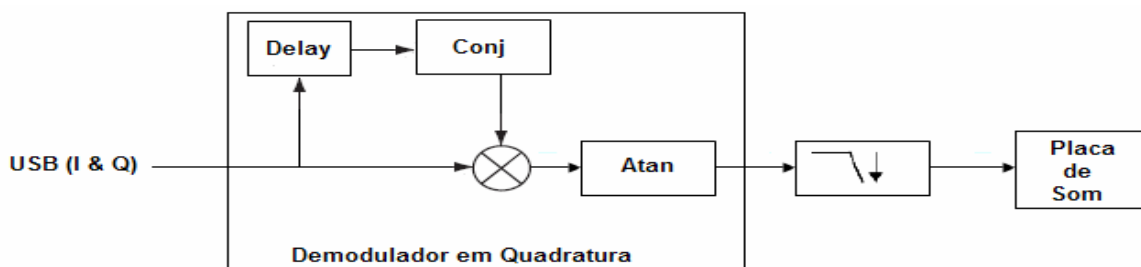


Figura 2.31: Demodulação FM em GNU-Radio.

A implementação dessa aplicação pode ser demonstrada pelo fragmento de código, em linguagem Python, apresentado a seguir.

```
#!/usr/bin/env python
from gnuradio import gr, gru, eng_notation, optfir
from gnuradio import audio
```

```

from gnuradio import usrp
from gnuradio.eng_option import eng_option
import sys
import math

class wfm_rx_graph (gr.flow_graph): # definição da aplicação

    def __init__(self):
        gr.flow_graph.__init__(self)

        self.vol = .1
        self.state = "FREQ"
        self.freq = 0

        # construção do grafo

        self.u = usrp.source_c() # os dados vem do USRP

        adc_rate = self.u.adc_rate() # 64 MS/s - taxa do ADC
        usrp_decim = 200
        self.u.set_decim_rate(usrp_decim)
        usrp_rate = adc_rate/usrp_decim # 320 kS/s - taxa no USB
        chanfilt_decim = 1
        demod_rate = usrp_rate / chanfilt_decim
        audio_decimation = 10
        audio_rate = demod_rate/audio_decimation # 32 kHz - amostragem

        filt_coeffs = optfir.low_pass (1, # ganho
                                      usrp_rate, # taxa de amostragem
                                      80e3, # passband cutoff
                                      115e3, # stopband cutoff
                                      0.1, # passband ripple
                                      60) # stopband attenuation

        chan_filt = gr.fir_filter_ccf (chanfilt_decim, filt_coeffs)

        self.guts = blks.wfm_rcv (self, demod_rate, audio_decimation)
        self.volume_control = gr.multiply_const_ff(self.vol)

        # placa de som com destino final
        audio_sink = audio.sink(int(audio_rate),
                                options.audio_output,
                                False)

        # Conectar os módulos
        self.connect (self.u, chan_filt, self.guts, self.volume_control,
                     audio_sink)
        ...

```

Capítulo 3

Plataforma Experimental

Neste capítulo é apresentada uma breve descrição do kit USRP (plataforma de hardware desenvolvida especificamente para o projeto GNU-Radio), mostrado na figura 3.1, onde podemos observar o conjunto de duas placas adaptadoras de recepção (*Rx daughterboard*) e duas placas adaptadoras de transmissão (*Tx daughterboard*). Também é apresentada uma sussinta discussão sobre alguns algoritmos utilizados no projeto GNU-Radio como, por exemplo, CORDIC (Coordinate Rotation Digital Computer), conversão de taxas de amostragem, filtros CIC (Cascaded Integrator Comb).

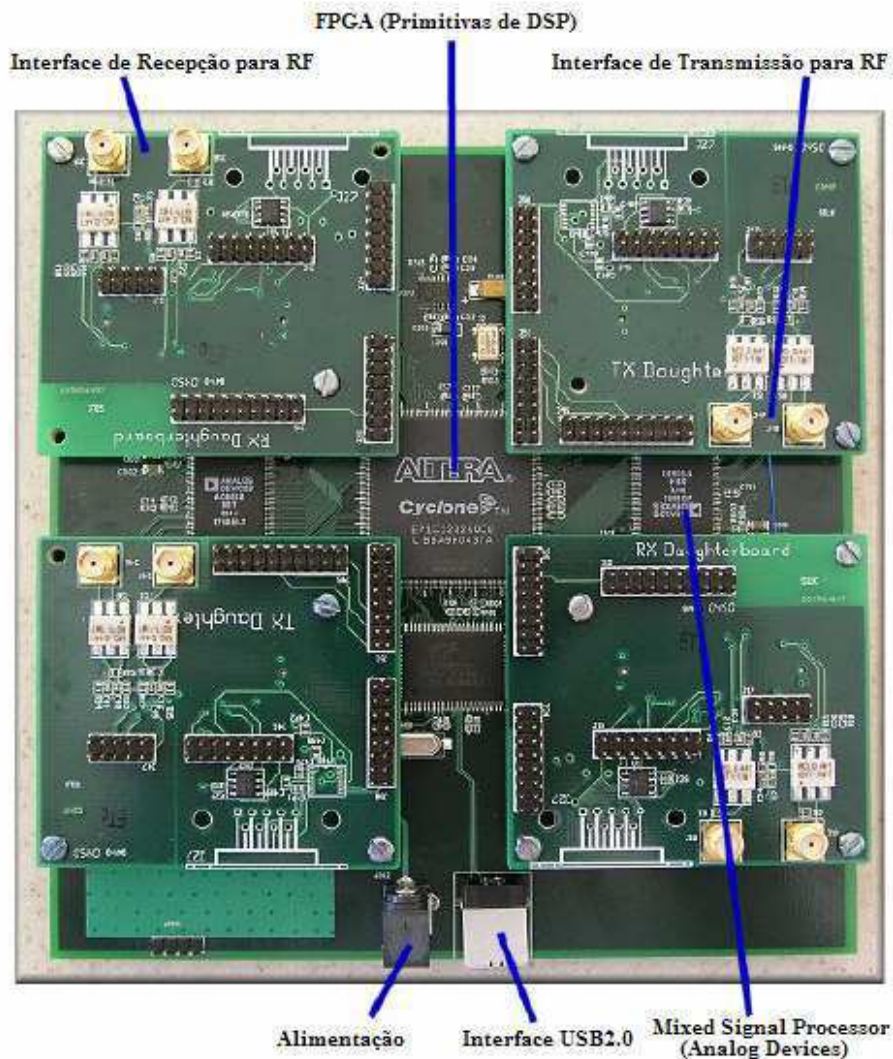


Figura 3.1: Kit USRP.

3.1 Universal Software Radio Peripheral (USRP)

O kit USRP é um sistema de hardware de alta velocidade e baixo custo, muito atraente como ferramenta de desenvolvimento e experimentação para aplicações de Radio Definidos em Software em tempo real. Esse hardware foi desenvolvido por um grupo de voluntários, liderados por Matt Ettus, para o grupo de usuários do projeto GNU-Radio. O kit é composto de uma placa (*motherboard*) que incorpora conversores A/D e D/A, interfaces de conexão externa e um FPGA como *front-end* para funções típicas de pre-processamento de sinais em alta velocidade como conversão de frequências (*Up Conversion* e *Down Conversion*), decimação e interpolação. Uma vez que esse kit foi criado, principalmente, visando o desenvolvimento e experimentação de Rádios Definidos em Software, o processamento específico de formas de ondas como modulação e demodulação, em geral é efetuado em um computador pessoal (PC), oferecendo uma grande flexibilidade para a comunidade investigadora da tecnologia de RDS.

Uma configuração típica é composta de uma placa principal (*motherboard*) e até quatro placas adaptadoras e depende de um computador pessoal (PC ou Mac) com interface USB2 para comunicação e gerenciamento. Na figura 3.2 pode ser visto um diagrama de blocos do kit mostrado na figura 3.1.

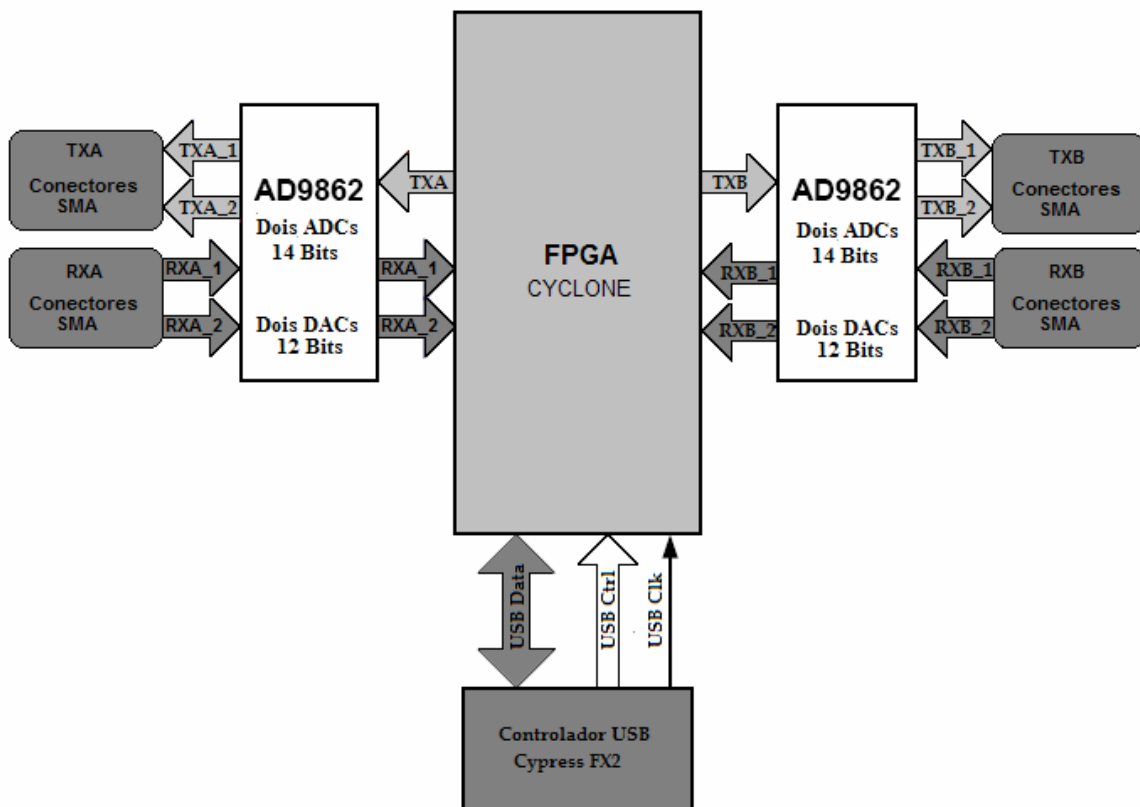


Figura 3.2: Diagrama de Blocos do USRP.

O sistema contém 4 conversores Analógico-Digital (ADC) de alta velocidade, cada um com 12 bits/amostra e taxa de amostragem de 64 Mamostras/segundo. Para transmissão existem 4 conversores Digital-Analógico (DAC) de alta velocidade, cada um com 14 bits/amostra e taxa de amostragem de 128 Mamostras/segundo. Esses 4 canais de recepção e 4 canais de transmissão operam em sintonia com um FPGA Cyclone EP1C12. Entre o FPGA e o PC do usuário existe uma interface USB2.0 controlada por microcontrolador Cypress FX2 que tem, como núcleo, um uC8051.

3.2 Conversores A/D e D/A (MxSP - Mixed Signal Processor)

Dois processadores de sinal da Analog Devices (AD9862) [21] são utilizados no USRP para as conversões de sinais de Analógico para Digital e de Digital para Analógico. O diagrama de blocos funcional desses processadores está mostrado na figura 3.3.

A princípio, com taxas de amostragem de 64 Ma/s seria possível digitalizar apenas sinais com largura de banda de até 32 MHz. Entretanto os ADCs podem efetuar conversões por sub-amostragem de sinais de até 150 MHz aproximadamente permitindo o mapeamento da banda do sinal de interesse para algum ponto entre -32 MHz e 32 MHz. Eventualmente isso pode ser utilizado em nosso favor, permitindo, por exemplo, a captura de sinais de estações comerciais de FM (faixa de 98 MHz – 108 MHz) sem nenhum hardware de RF adicional.

A faixa total de entrada dos ADCs é de 2 Vpp, e a entrada é de 200 Ohms diferencial, o que equivale a 40 mW ou 16 dBm. Amplificadores de ganho programável (PGA) com ganho de até 20 dB estão presente antes dos ADCs (especialmente importante para pequenos sinais).

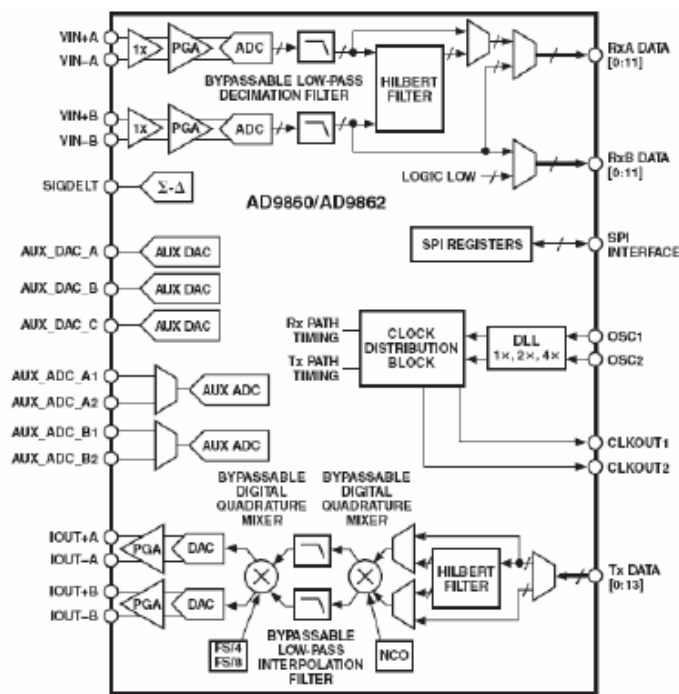


Figura 3.3: Conversores A/D e D/A MxSP (Mixed-Signal Processor).

3.3 Módulos Adaptadores (*Daughter Boards*)

Na *motherboard* existem quatro conectores (*slots*) para interconexão de até quatro *daughter boards*. Esses adaptadores são utilizados para conexão física com módulos de RF. Existem dois slots denominados TXA e TXB e dois slots denominados RXA e RXB. Cada *daughter board* tem acesso a dois dos quatro conversores A/D e D/A o que permite que cada adaptador controle dois canais independentes de RF. É possível observar, na figura 3.1, que cada adaptador tem dois conectores SMA.

3.4 FPGA (Field Programmable Gate Array)

O dispositivo FPGA é o coração do projeto USRP. Como mostrado na figura 3.2, todos os ADCs e DACs estão conectados no ou através do FPGA. Parte do processamento matemático de alta velocidade está sob responsabilidade deste dispositivo favorecendo uma redução da taxa de dados de modo que a comunicação possa ser mantida através da interface USB. Nossa configuração contempla blocos DDC (Digital Down Converter) implementados com filtros CIC (Cascaded Integrator-Comb). O FPGA suporta até quatro DDCs, permitindo a criação de 1, 2 ou 4 canais RX. No circuito de RX temos 4 ADCs e 4 DDCs. Cada DDC tem duas entradas (real e complexa) e cada um dos 4 ADCs pode ser remanejado para a entrada real ou complexa de qualquer dos 4 DDCs. Com arranjos desse tipo, é possível o estabelecimento de múltiplos canais a partir do mesmo ADC.

No circuito TX, os DUCs (Digital Up Converter) estão embutidos nos chips MxSPs AD9862 e não no FPGA, como pode ser visto na figura 3.4. Os únicos blocos de processamento existentes no caminho de TX, no FPGA, são os interpoladores.

Os blocos MUX funcionam com roteadores ou chaveadores determinando qual ADC (ou zero constante) é conectado a cada entrada de DDC ou DUP. Existem 4 DDCs e cada um tem duas entradas.

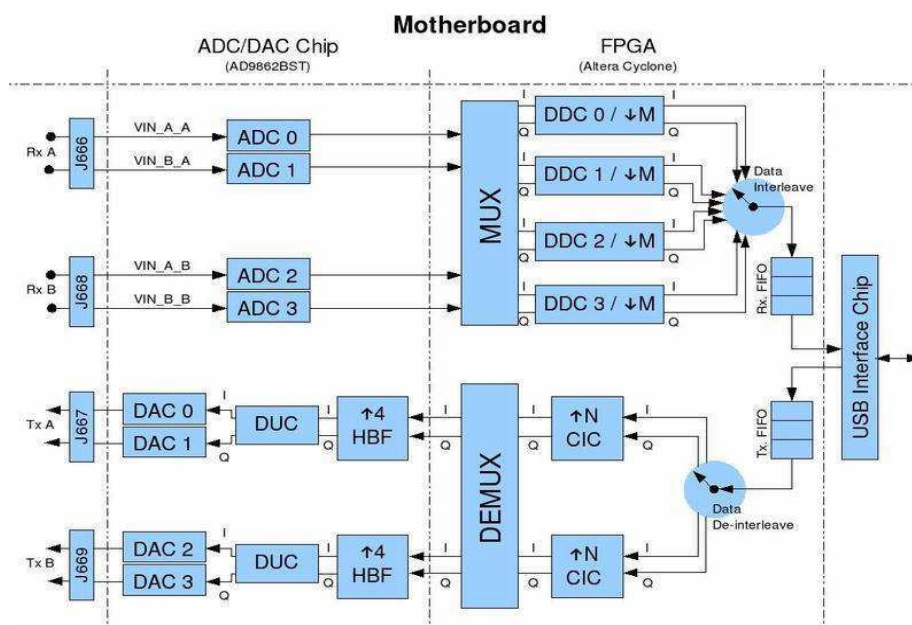


Figura 3.4: Diagrama de Blocos detalhado do USRP.

3.5 DDC (Digital Down Converter)

Os blocos DDCs convertem os sinais de FI (Frequência Intermediária) para banda-base. Em seguida efetuam a decimação no sinal de modo que a taxa de amostragem possa ser ajustada em relação à largura de banda da interface USB e se sustente em um valor razoável para o processamento pelo PC.

O diagrama de blocos, mostrado na figura 3.5, apresenta o DDC do ponto de vista funcional, onde podemos notar uma grande semelhança com as arquiteturas de receptores em quadratura vistas no capítulo 2.

O sinal complexo de entrada (I_1 e Q_1) é multiplicado pelo sinal exponencial de frequência fixa (em geral igual a FI). O sinal resultante também é complexo e está centrado em zero Hz. Na sequência ocorre a decimação por um fator M . O decimador pode ser visto como um filtro passa-baixas seguido de um redutor de amostragem. A taxa de decimação deve estar entre 4 e 256. Finalmente os sinais complexos I_2 e Q_2 chegam ao PC, via interface USB, para processamentos subsequentes.

Esta descrição está associada à funcionalidade de um DDC convencional. Entretanto sua implementação no USRP faz uso do algoritmo CORDIC conforme descrito nas sessões 3.6 e 3.7.

No caminho de TX, ocorre o inverso. Enviamos um sinal complexo I e Q em banda-base ao USRP. O DUC (Digital Up Converter) interpola o sinal, compatibilizando a taxa de amostragem com o MxSP, convertendo-o para banda FI e, finalmente, o envia através do DAC.

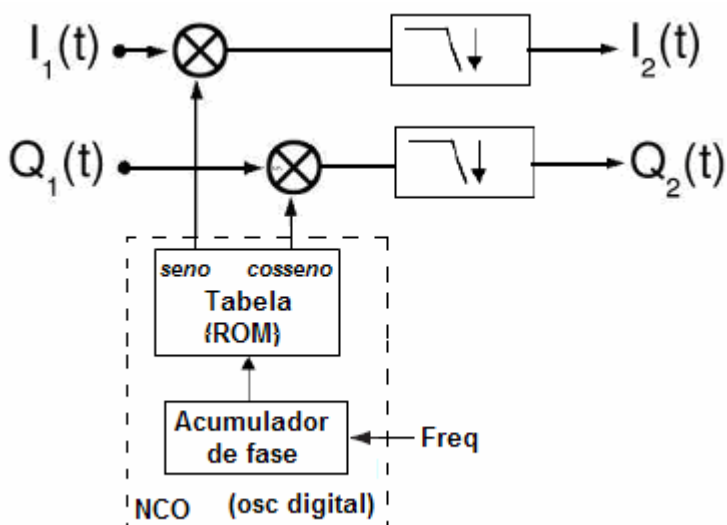


Figura 3.5: Diagrama de blocos do DDC.

3.6 O Algoritmo CORDIC (Coordinate Rotation Digital Calculation)

CORDIC é um algoritmo especial usado para o cálculo de funções trigonométricas como seno, cosseno, magnitude e fase (arco-tangente) para qualquer nível desejado de precisão. Também pode ser utilizado para calcular funções hiperbólicas, lineares e logarítmicas. O mais interessante desse algoritmo é a não utilização de multiplicadores. Apenas simples operações matemáticas como soma, subtração, comparação, deslocamentos e tabelas são necessárias para a implementação dessas funções. A teoria CORDIC é baseada no algoritmo de rotação vetorial que permite a derivação de todas as funções trigonométricas. Rotações vetoriais também são úteis para conversões como polar-retangular e magnitudes vetoriais. O algoritmo CORDIC, publicado pela primeira vez por Volder [22] em 1959, faz uso de um método iterativo para realização de rotações vetoriais em torno de ângulos arbitrários usando apenas deslocamentos e somas.

Na figura 3.6(a), as coordenadas do ponto p são (x,y) no sistema original de coordenadas e (x',y') no novo sistema de coordenadas. O sistema de coordenadas original está rotacionado de um ângulo ϕ . Na figura 3.6(b) está representada a relação entre (x',y') e (x,y) . As equações ligadas às novas coordenadas são:

$$x' = x \cos \phi + y \sin \phi \quad (3.1)$$

$$y' = y \cos \phi - x \sin \phi \quad (3.2)$$

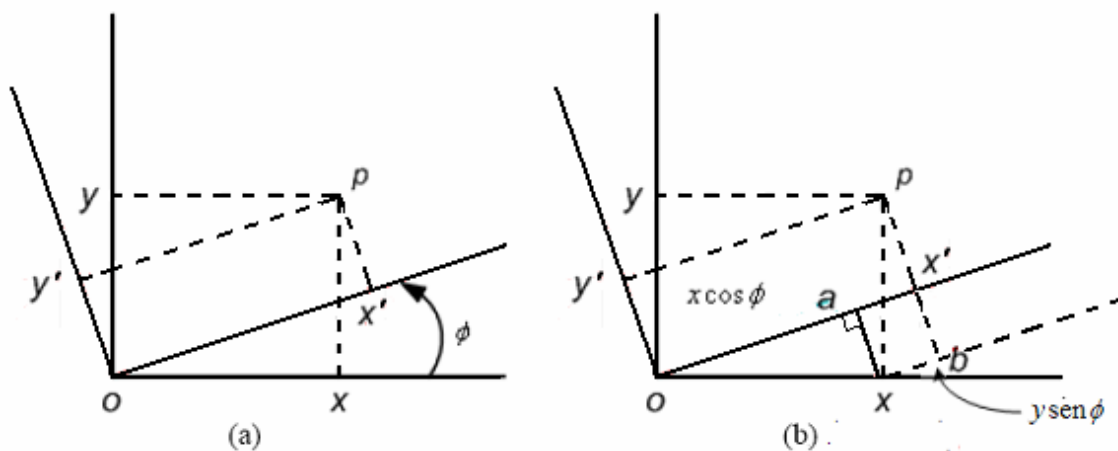


Figura 3.6: Rotação de coordenadas por um ângulo fixo.

Na figura 3.7 está ilustrado como a rotação de coordenadas está relacionada com rotação de vetores. O vetor original com origem em O' e coordenadas em (x,y) é rotacionado de um ângulo $-\phi$ para a nova coordenada (x',y') . Dessa vez estamos movendo o ponto em vez da base de coordenadas. A transformação de coordenadas é igual ao caso anterior só que o ângulo agora é $-\phi$ em vez de $+\phi$.

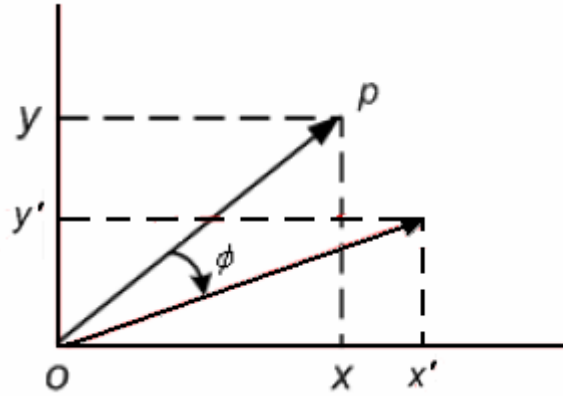


Figura 3.7: Rotação de um vetor por um ângulo fixo.

E as novas coordenadas são dadas por:

$$x' = x \cos \phi - y \sin \phi \quad (3.3)$$

$$y' = y \cos \phi + x \sin \phi \quad (3.4)$$

rearranjando, temos:

$$x' = x \cos \phi [x - y \cdot \tan \phi] \quad (3.5)$$

$$y' = y \cos \phi [y + x \cdot \tan \phi] \quad (3.6)$$

Se assumirmos que $\tan \phi = \pm 2^{-i}$, então a multiplicação pelo termo tangente reduz-se a um simples deslocamento. Ângulos arbitrários de rotação são obtidos através de pequenas rotações elementares sucessivas. Se a decisão a cada passo iterativo, i , for para qual direção rotacionar em vez de se deve rotacionar ou não rotacionar, então o termo $\cos(\delta_i)$ transforma-se em uma constante, uma vez que $\cos(\delta_i) = \cos(-\delta_i)$. A rotação iterativa pode, então ser expressa por:

$$x_{i+1} = K_i [x_i - y_i d_i 2^{-i}] \quad (3.7)$$

$$y_{i+1} = K_i [y_i + x_i d_i 2^{-i}] \quad (3.8)$$

onde $K_i = \cos(\tan^{-1} 2^{-i}) = \frac{1}{\sqrt{1+2^{-2i}}}$ e $d_i = \pm 1$

Essas equações iterativas se transformam em um algoritmo simples de deslocamentos e somas se removermos o termo constante K_i . O produto do termo K_i pode ser calculado separadamente e tende a 0,6073 quando o número de interações tende ao infinito. Portanto esse algoritmo tem um ganho, A_n , de aproximadamente 1,647 e depende da seguinte expressão:

$$A_n = \prod_n \sqrt{1+2^{-2i}} \quad (3.9)$$

O ângulo de uma rotação composta é definido pela seqüência de direções das rotações elementares. Essa seqüência pode ser representada por um vetor de decisão e o conjunto de

todos os possíveis vetores de decisão formam um sistema de medida angular baseado em arco-tangentes binários. Conversões entre esse sistema angular e qualquer outro podem ser implementadas pela utilização de tabelas. Um método melhor utiliza um somador-subtrator adicional que acumula os ângulos de rotação elementar a cada interação. Os ângulos elementares podem ser expressos em qualquer unidade angular conveniente e podem ser fornecidos por uma pequena tabela (um valor para cada interação) ou podem ser fixados em hardware, dependendo da implementação. O acumulador de ângulos adiciona uma terceira equação ao algoritmo CORDIC:

$$z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \quad (3.10)$$

Existem dois modos de operação para o rotacionador CORDIC. O primeiro, rotacional, rotaciona um vetor de entrada por um ângulo especificado (como um argumento). O segundo modo, dito vetorial, rotaciona o vetor de entrada em torno do eixo x enquanto registra o ângulo necessário para tal rotação. As equações para o modo rotacional são:

$$x_{i+1} = x_i - y_i d_i 2^{-i} \quad (3.11)$$

$$y_{i+1} = y_i + x_i d_i 2^{-i} \quad (3.12)$$

$$z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \quad (3.13)$$

onde $d_i = -1$ se $z_i < 0$, caso contrário $d_i = +1$. Isso nos leva aos seguintes resultados:

$$x_n = A_n [x_0 \cos z_0 - y_0 \sin z_0] \quad (3.14)$$

$$y_n = A_n [y_0 \cos z_0 + x_0 \sin z_0] \quad (3.15)$$

$$z_n = 0 \quad (3.16)$$

$$A_n = \prod_n \sqrt{(1 + 2^{-2i})} \quad (3.17)$$

No modo vetorial as equações são:

$$x_{i+1} = x_i - y_i d_i 2^{-i} \quad (3.18)$$

$$y_{i+1} = y_i + x_i d_i 2^{-i} \quad (3.19)$$

$$z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \quad (3.20)$$

onde $d_i = +1$ se $z_i < 0$, caso contrário $d_i = -1$, com o seguinte desdobramento:

$$x_n = A_n \sqrt{x_0^2 - y_0^2} \quad (3.21)$$

$$y_n = 0 \quad (3.22)$$

$$z_n = z_0 + \tan^{-1}(y_0/x_0) \quad (3.23)$$

$$A_n = \prod_n \sqrt{(1 + 2^{-2i})} \quad (3.24)$$

Esse algoritmo pode ser utilizado para computar diversas funções diretamente e outras indiretamente. A escolha adequada de valores iniciais e modos de operação permitem o

cálculo direto de seno, co-seno, arco-tangente, magnitude vetorial e conversões de coordenadas polar-cartesiano e cartesiano-polar. O modo rotacional permite ao CORDIC o cômputo simultâneo de seno e co-seno de um dado ângulo. Fazendo o componente y do vetor de entrada igual a zero, o resultado pode ser reduzido a:

$$x_n = A_n x_0 \cos z_0 \quad (3.25)$$

$$y_n = A_n x_0 \sen z_0 \quad (3.26)$$

Fazendo $x_0 = 1/A_n$, a rotação produz o seno e cosseno do ângulo-argumento z_0 , o que é muito interessante para a determinação dos componentes I e Q de um sistema de comunicação sem o uso de multiplicadores.

3.7 CORDIC no processo de conversão de freqüências (DDC)

Os demoduladores em quadratura são utilizados como front-end digitais para uma grande variedade de esquemas de modulação como BPSK, QPSK e QAM. No receptor, o chamado misturador em quadratura deve transladar a freqüência do sinal de entrada de RF para FI ou de FI para banda-base pela multiplicação deste sinal por um sinal exponencial complexo [23], com freqüência igual a F_{rf} ou F_{fi} dependendo do caso. Em geral esse processo é implementado com o uso de osciladores digitais baseados em tabelas ROM combinados com multiplicadores complexos, conforme pode ser visto nas figuras 3.5 e 3.8.

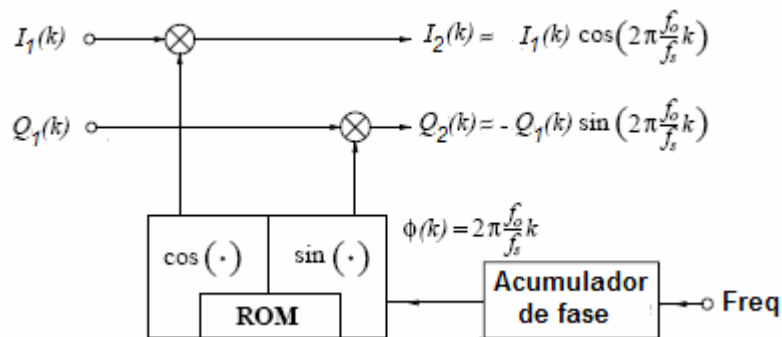


Figura 3.8: DDC convencional usando NCO baseado em tabelas ROM.

Esta arquitetura tem o inconveniente de necessitar uma grande tabela em ROM para conseguir a resolução necessária ao processo de geração das amostras de seno e co-seno, consumindo uma área relativamente grande do chip além de um elevado consumo de energia.

Uma alternativa para evitar esse inconveniente é a utilização do algoritmo CORDIC para a geração das amostras de seno e co-seno. A desvantagem da utilização de um processador CORDIC neste tipo de aplicação é o aumento da complexidade do circuito; desvantagem essa compensada pela dispensa dos multiplicadores e do oscilador digital explícito.

Se o processador CORDIC for operado no modo de rotação conforme visto na sessão 3.6, caracterizado pelas equações 3.14, 3.15, 3.16 e 3.17, aqui repetidas por conveniência,

$$x_n = A_n [x_0 \cos z_0 - y_0 \sin z_0] \quad (3.14)$$

$$y_n = A_n [y_0 \cos z_0 + x_0 \sin z_0] \quad (3.15)$$

$$z_n = 0 \quad (3.16)$$

$$A_n = \prod_n \sqrt{1 + 2^{-2i}} \quad (3.17)$$

o DDC pode ser implementado de forma eficiente para aplicações de conversão de frequências, sincronismo e correções de fase [24].

Substituindo a saída do DDC convencional da figura 3.8 nas equações 3.14 e 3.15 e representando o sinal de entrada em banda-passante ou FI como $S_{BP}(k)$, podemos reformular a arquitetura do DDC conforme mostrado na figura 3.9.

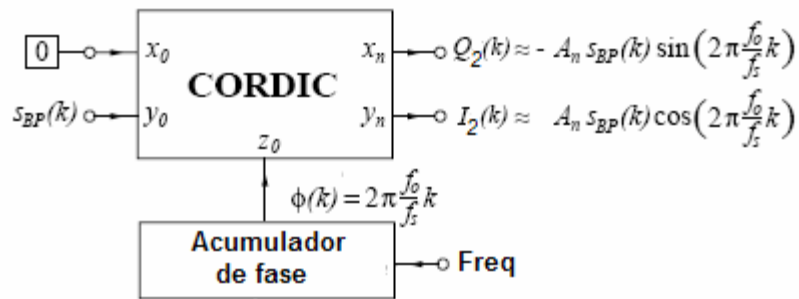


Figura 3.9: DDC baseado no processador CORDIC no modo rotacional.

Nesta arquitetura, a cada intervalo de clock a entrada y_0 é carregada com a amostra corrente do sinal de entrada $S_{BP}(k)$ e a entrada z_0 é carregada com a amostra corrente do sinal de fase $\phi(k)$ que é fornecida pelo acumulador de fase e corresponde à frequência do oscilador local $f_0 = \text{Freq}$.

Conforme descrito na sessão 3.6, após $n+1$ interações, o processador CORDIC fornece em sua saída as amostras $I_2(k)$ e $Q_2(k)$ com uma resolução de n bits.

Considerando que o FPGA utilizado no USRP (Altera Cyclone EP1C12) não dispõe de grande quantidade de memória e nem de multiplicadores dedicados, essa arquitetura se apresenta como uma alternativa natural para a implementação do módulo DDC no projeto GNU-Radio.

3.8 Filtros CIC (Conversão de taxa de amostragem)

Na medida em que os conversores de dados tornam-se mais e mais rápidos, aplicações de extração de sinais de faixa-estreita a partir de sinais de faixa-larga e a construção de sinais de faixa-larga a partir de sinais de faixa-estreita passam a ser cada vez mais importantes. Essas funções necessitam de dois procedimentos básicos em processamento de sinais: decimação e interpolação. Em 1981, uma forma eficiente de implementá-las foi apresentada por Hogenauer [25]. Sua proposta era baseada em um filtro flexível, sem multiplicadores, adequado a implementação em hardware e que podia processar conversões arbitrárias e amplas. Esse processo ficou conhecido como *Cascaded Integrator Comb Filters*, ou apenas filtros CIC.

Os blocos básicos componentes desse tipo de filtros estão esquematizados na figura 3.10.

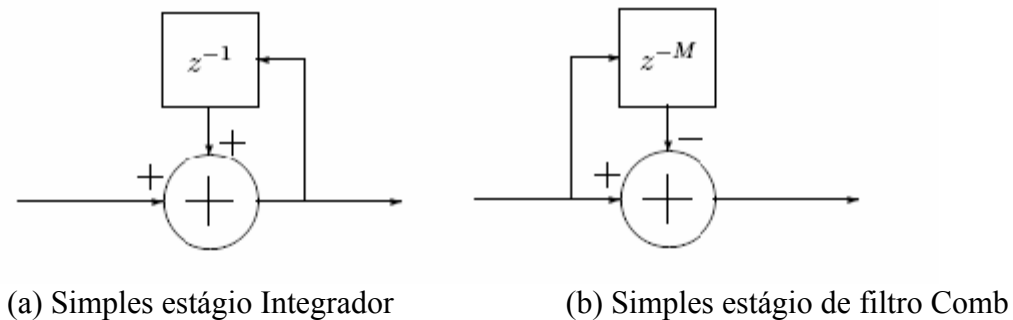


Figura 3.10: Blocos básicos de um filtro CIC.

A implementação de um filtro CIC é composta do encadeamento de N módulos integradores e N módulos comb.

A esse conjunto é adicionado um módulo de conversão de taxa de dados. Nas figuras 3.11 e 3.12 estão exemplificados um filtro decimador de três estágios e um filtro interpolador de três estágios, respectivamente.

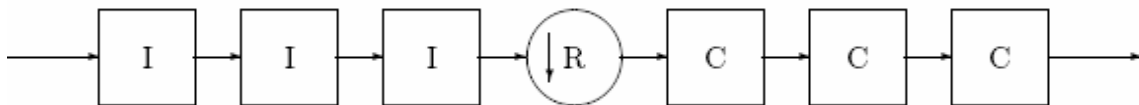


Figura 3.11: Filtro decimador CIC de três estágios.

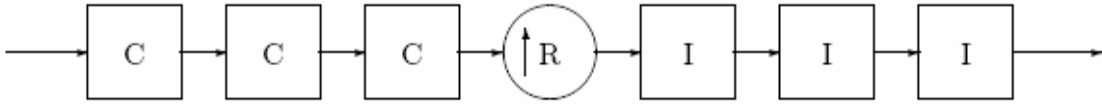


Figura 3.12: Filtro interpolador CIC de três estágios.

Capítulo 4

Identificação de Modulação

O reconhecimento automático da modulação empregada em determinados sinais é de extrema importância em aplicações militares, em primeiro lugar porque o uso de um demodulador inapropriado pode mascarar parcial ou totalmente a mensagem contida no sinal. Em segundo lugar porque reconhecer o tipo correto de modulação pode ajudar na identificação de ameaças e, eventualmente, na determinação de formas de onda adequadas ao bloqueio das comunicações do inimigo.

Vários trabalhos são encontrados na literatura especializada relacionados ao tema reconfigurabilidade, cobrindo desde técnicas de reconfiguração de hardware, passando por arquiteturas modulares de software, até estratégias de acesso.

4.1 Estado da Arte (Literatura)

A literatura pesquisada indica que existem dois grandes grupos de abordagens relacionadas ao reconhecimento, identificação e classificação automática de modulação em sistemas inteligentes de comunicação: decisão baseada em hipóteses e decisão baseada em reconhecimento de padrões.

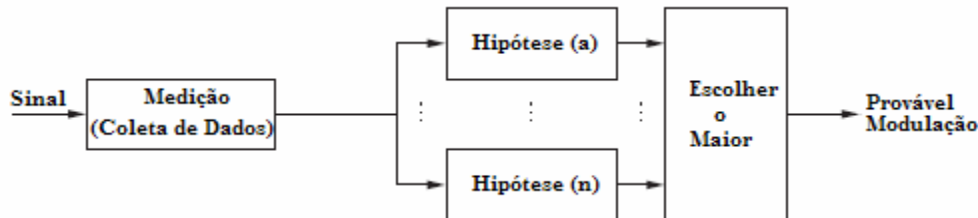


Figura 4.1: Abordagem baseada em hipóteses.

Já nas abordagens baseadas em reconhecimento de padrões, o sistema de classificação é dividido em dois sub-sistemas. O primeiro é um sub-sistema de extração de características que identifica e extrai um conjunto pré-definido de características do sinal. O segundo sub-sistema tem por finalidade determinar o tipo de modulação presente no sinal. Esse segundo sub-sistema funciona em duas etapas: etapa de treinamento para configurar o sistema classificador seguido de uma fase de testes que fornece a decisão de classificação. Esse mecanismo é mais apropriado para análises *off-line* do que para processos em tempo real devido ao alto custo de processamento envolvido.

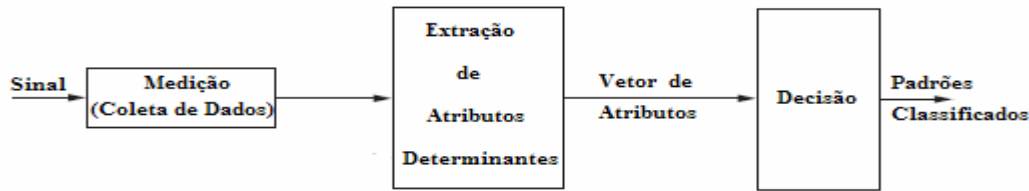


Figura 4.2: Abordagem baseada em reconhecimento de padrões.

As técnicas empregadas para resolver o problema de reconhecimento de modulação podem ser agrupadas em 4 classes:

- 1 - Processamento espectral,
- 2 - Amplitude, Freqüência e Fase instantâneas,
- 3 - Histogramas de amplitude, freqüência e fase e
- 4 - Combinações dessas três técnicas.

4.1.1 Abordagem por funções *likelihood* (Verossimilhança)

O conceito central dessa abordagem baseia-se na premissa de que as funções de verossimilhança (*likelihood function* - *LF*) ou verossimilhança logarítmicas (*log-likelihood function* - *LLF*) do sinal observado contêm todas as informações necessárias para uma grande variedade de inferências (detecção do sinal, classificação e estimação de parâmetros). Em [26] Huang e Polydoros derivam um método generalizado para classificar sinais M-PSK em um canal perturbado por ruído do tipo AWGN. Soliman [27] se utiliza de momentos estatísticos da fase do sinal para classificar modulação M-PSK. Callagan [28] utiliza técnica de cruzamento em zero para classificar modulações como AM, FM e FSK.

4.1.2 Abordagem por reconhecimento de padrões

Um método popular nesta abordagem é a utilização de redes neuronais artificiais (ANN) [29] e lógica Fuzzi [30]. Aparentemente a vantagem básica dessa abordagem sobre a de decisão teórica baseada em hipóteses é que os parâmetros de limiar de decisão para os perceptrons são escolhidos (encontrados) automática e adaptativamente [31]. Os reconhecedores baseados em ANN são compostos de três estágios [1]:

- 1- Pré-processamento, onde características de qualquer sinal são extraídas;
- 2- Treinamento e aprendizagem para configurar a rede;
- 3- Fase de testes para a efetiva classificação do tipo de modulação.

Extração de característica – Um sinal de comunicação pode ser completamente caracterizado por seus parâmetros de Amplitude, Freqüência e Fase, cujas estatísticas podem ser utilizadas para diferenciá-lo de outros sinais. A idéia é criar um mapeamento ou relacionamento entre essas características e o esquema de modulação. As características

escolhidas e a arquitetura da rede definem a velocidade de convergência e precisão dos resultados.

Fase de Treinamento – Uma vez que as características tenham sido extraídas, a rede é treinada com pares de entrada-saída. Isso é feito interativamente até que a rede estabilize em estado de mínimo erro. Quando o erro atinge um patamar mínimo a rede terá convergido.

Fase de Testes – Nesta fase, a rede é ativada com os parâmetros pré-estabelecidos na fase de treinamento e é utilizada para a efetiva identificação da modulação presente no sinal.

4.2 Abordagem Investigada

A abordagem *ad hoc* aqui proposta é um método simples de decisão binária baseado em atributos extraídos do conteúdo espectral do sinal sob teste.

4.2.1 Classe de modulação (Análogica / Digital)

Em geral, o conteúdo espectral de um sinal modulado digitalmente apresenta picos característicos devido a variações periódicas resultante de mudanças constantes na fase da portadora. Um sinal com modulação analógica deve apresentar um conteúdo espectral com uma distribuição mais uniforme [32] conforme está mostrado na figura 4.3.

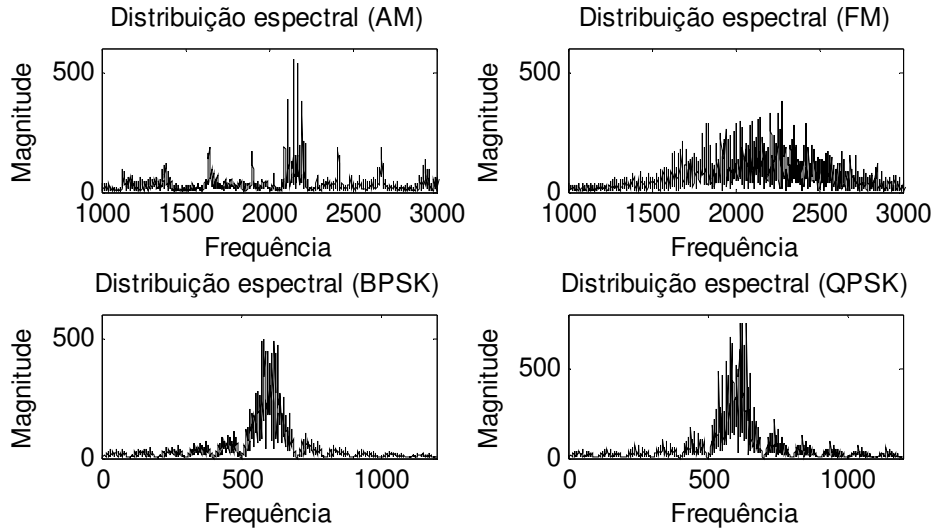


Figura 4.3: Distribuição espectral existente em diferentes tipos de modulação.

A decisão se baseia na seguinte característica (atributo) extraída do conteúdo espectral do sinal sob teste $x(n)$ [1]:

$$\gamma_{\max f} = \max[DFT(f_{cn}(n))] \quad (4.1)$$

onde $DFT(x)$ representa a transformada discreta de Fourier definida como

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi kn}{N}} \quad (4.2)$$

e $f_{cn}(n)$ representa a frequência instantânea normalizada do sinal, definida como

$$f_{cn}(n) = f_n(n) - m_a \quad (4.3)$$

onde m_a representa a média da frequência instantânea e a frequência normalizada é definida por

$$f_n(n) = \frac{f(n)}{f_{\max}} \quad (4.4)$$

com f_{\max} igual a frequência instantânea máxima e $f(n)$ é a frequência instantânea, definida por

$$f(n) = \frac{\phi(n+T) - \phi(n)}{2\pi T} \quad (4.5)$$

onde $\phi(n)$ representa a fase instantânea do sinal, que pode ser encontrada como

$$\phi(n) = \tan^{-1} (\Im[x(n)] / \Re[x(n)]) \quad (4.6)$$

$\gamma_{\max f}$ representa o valor máximo da densidade espectral de potência da frequência instantânea normalizada do sinal.

Na figura 4.4 podemos ver um exemplo do comportamento deste atributo para os diferentes tipos de modulação. Esta figura foi gerada a partir de quatro sinais com 8000 amostras cada. Baseado neste comportamento, podemos concluir que o atributo $\gamma_{\max f}$ pode ser útil na diferenciação entre modulação analógica e modulação digital, pois se $\gamma_{\max f}$ for menor que um patamar de referência significa que a modulação presente é BPSK ou QPSK. Por exclusão, se $\gamma_{\max f}$ for maior que o patamar, deduzimos que se trata de modulação analógica (AM ou FM).

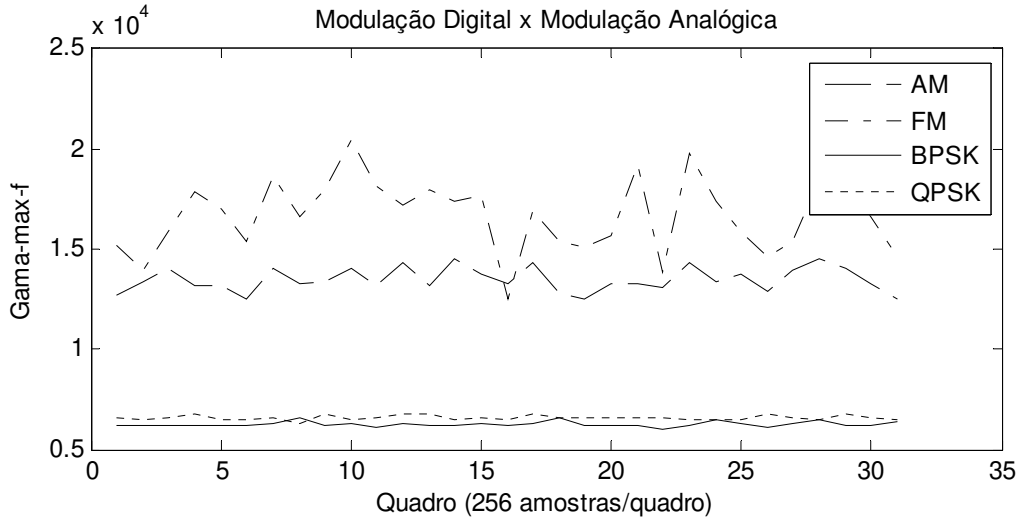


Figura 4.4: Atributo $\gamma_{\max f}$ em função do tipo de modulação.

4.2.2 Modulação analógica (AM / FM)

Se o sinal for classificado como modulado analogicamente, a decisão quanto ao tipo (Amplitude Modulada ou Frequência Modulada) é tomada com base na análise de outra característica extraída do conteúdo espectral do sinal sob teste $x(n)$ [1]:

$$\gamma_{\max a} = \max[[DFT(a_{cn}(n))]^2] / N \quad (4.7)$$

onde N representa o número de amostras analisadas, $a_{cn}(n)$ representa a amplitude instantânea normalizada pela média e centralizada do sinal, definida como

$$a_{cn}(n) = \frac{|x(n)|}{E[x(n)]} - \bar{1} \quad (4.8)$$

$$|x(n)| = \sqrt{\Im[x(n)]^2 + \Re[x(n)]^2} \quad (4.9)$$

$$E[x(n)] = \frac{1}{N} \sum |x(n)| \quad (4.10)$$

onde N representa o número de amostras, $\bar{1}$ representa um vetor unitário e $\gamma_{\max a}$ representa o valor máximo da densidade espectral de potência da amplitude instantânea normalizada do sinal. Como esse atributo está relacionado com a variação de amplitude do envelope (envoltória) do sinal, é de se esperar que esse valor seja maior para sinais modulados em amplitude do que para sinais modulados em frequência, conforme podemos observar na figura 4.5.

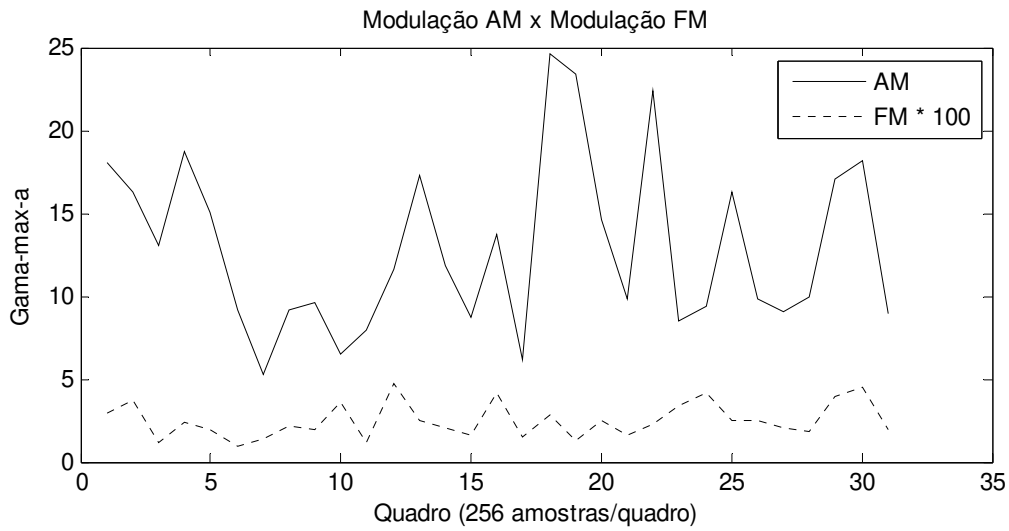


Figura 4.5: Atributo $\gamma_{\max a}$ em função do tipo de modulação analógica.

Se $\gamma_{\max a}$ for maior que um determinado patamar de referência a modulação presente é AM, do contrário nos decidimos por FM.

A variação do atributo $\gamma_{\max a}$, em função de variações nos índices de modulação, tanto em AM como em FM, está ilustrada nas figuras 4.6 e 4.7.

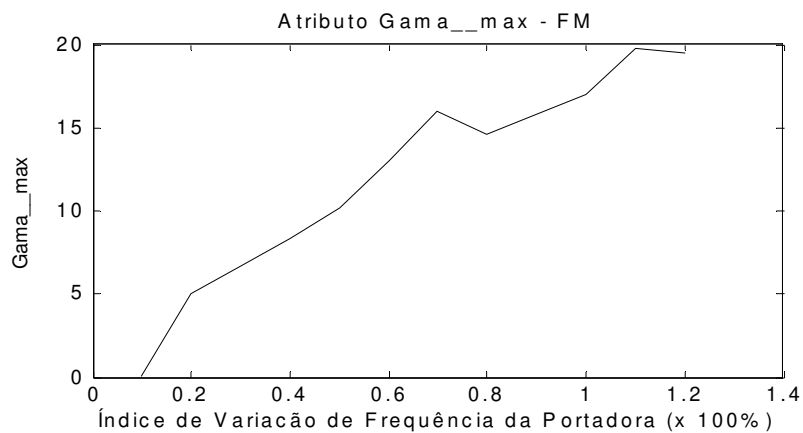


Figura 4.6: Relação entre variação de frequência e o atributo $\gamma_{\max a}$.

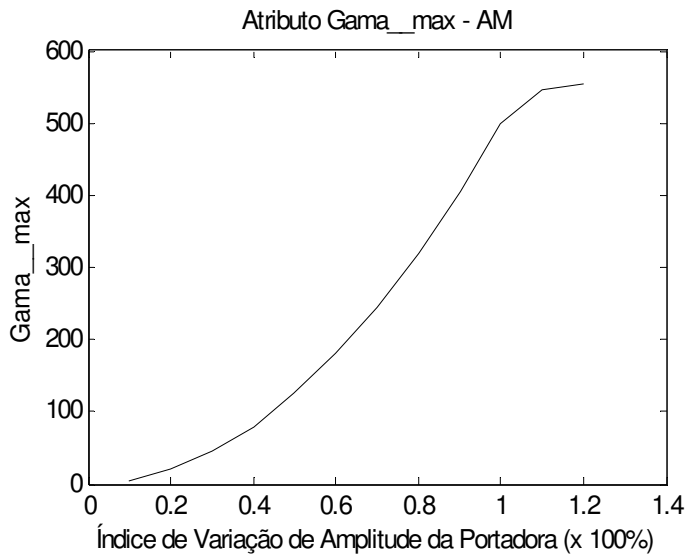


Figura 4.7: Relação entre variação de amplitude e o atributo $\gamma_{max a}$.

As figuras 4.6 e 4.7 foram geradas a partir de simulação com sinais gerados no Matlab com os seguintes parâmetros:

Portadora: Frequência de 2KHz, amplitude de 1Vpp e taxa de amostragem de 8KHz.
 Mensagem: Frequência de 100Hz, amplitude de 1Vpp e taxa de amostragem de 8 KHz.
 Número de amostras: 1000

Na figura 4.6 está representada uma portadora cuja frequência varia entre 0 e 100% de seu valor nominal (1KHz).

Na figura 4.7 está representada uma portadora cuja amplitude varia entre 0 e 100% do seu valor nominal.

4.2.3 Modulação digital (BPSK / QPSK)

Se o sinal for classificado como modulado digitalmente, então podemos explorar o fato da modulação digital provocar transientes na forma de onda nos instantes em que ocorrem as transições dos símbolos.

Pode-se explorar esta característica aplicando-se uma não-linearidade ao sinal e observando seu espectro. Uma característica da modulação BPSK é que surge um forte pico no espectro caso a função seja elevada ao quadrado (não-linearidade).

Escrevendo (2.10) de uma forma ligeiramente diferente temos:

$$s(t) = A \cos(2\pi f_c t + \pi p(t) + \theta_0) \quad (4.11)$$

onde $p(t)$ representa uma função binária de comutação cujos possíveis estados são 0 e 1 representando a mensagem transmitida.

Calculando o quadrado de (4.11) obtemos:

$$s^2(t) = \frac{A}{2} + \frac{A}{2} \cos(4\pi f_c t + 2\pi p(t) + 2\theta_0) \quad (4.12)$$

A função de comutação agora está multiplicada por 2π e, portanto, não provoca nenhuma mudança de fase no sinal resultante.

A simplificação de (4.12) resulta em

$$s^2(t) = \frac{A}{2} + \frac{A}{2} \cos(4\pi f_c t + 2\theta_0) \quad (4.13)$$

Isso pode ser interpretado como uma forma de eliminar a modulação BPSK em troca de um pico facilmente identificável no espectro do sinal.

Se compararmos os diagramas espectrais das duas funções (4.11) e (4.13), podemos notar claramente que a informação contida na modulação desaparece e surge um pico equivalente a 2 vezes a frequência da portadora.

Essa característica da modulação BPSK pode ser avaliada observando as figuras 4.8 e 4.9, geradas a partir de simulação com sinais gerados no Matlab com os seguintes parâmetros:

Portadora: Frequência de 100 Hz, amplitude de 1Vpp e taxa de amostragem de 1 KHz.

Símbolos: Frequência de 10 Hz, amplitude de 1Vpp e taxa de amostragem de 1 KHz.

Deslocamento de Fase: π (BPSK) e $\pi/4$ (QPSK)

Número de amostras: 1000

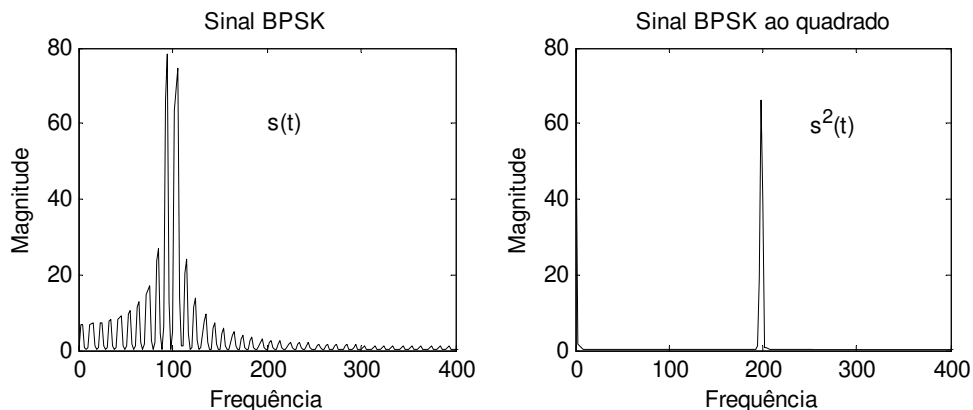


Figura 4.8: Não-linearidade aplicada a um sinal BPSK.

Aplicando um filtro passa-baixas ao sinal (4.13) teremos como resultado um nível DC caso a modulação presente seja BPSK. Se, ao contrário, a modulação for QPSK teremos picos no espectro devido a presença da informação de modulação, conforme pode ser visto na figura 4.9.

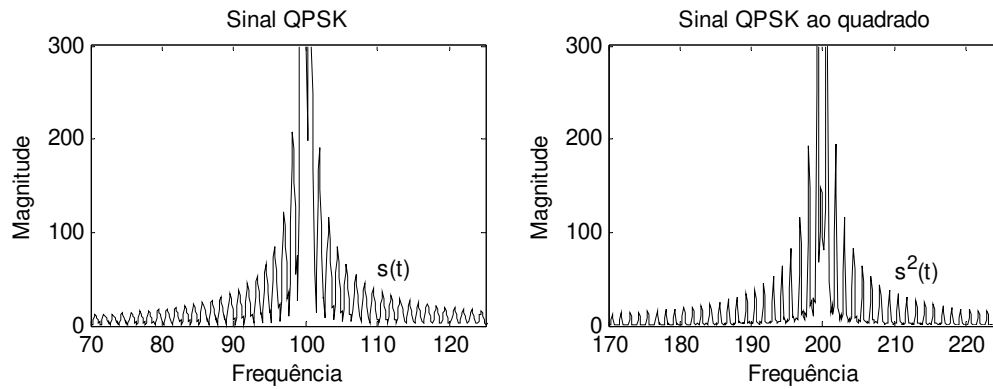


Figura 4.9: Não-linearidade aplicada a um sinal QPSK.

A diferenciação do sinal após o filtro passa-baixas permite capturar as descontinuidades do sinal QPSK

Portanto, essa característica pode ser utilizada para diferenciação entre modulação BPSK e QPSK com base na quantidade picos presente no sinal resultante.

Capítulo 5

Aplicação – Demodulador Universal

Visando validar a abordagem apresentada no capítulo 4, uma aplicação foi desenvolvida na linguagem Python sob o sistema operacional Linux Ubuntu 6.0.

Esta aplicação é composta de dois módulos independentes que se comunicam: Módulo Monitor de Camada Física cuja função é extrair os atributos do sinal que servirão de apoio para a análise de modulação e um Módulo Identificador de Modulação que se baseia nestes atributos para determinação do tipo de modulação presente no sinal.

A implementação do demodulador universal está representada esquematicamente na forma de um diagrama de blocos que pode ser visto na figura 5.1.

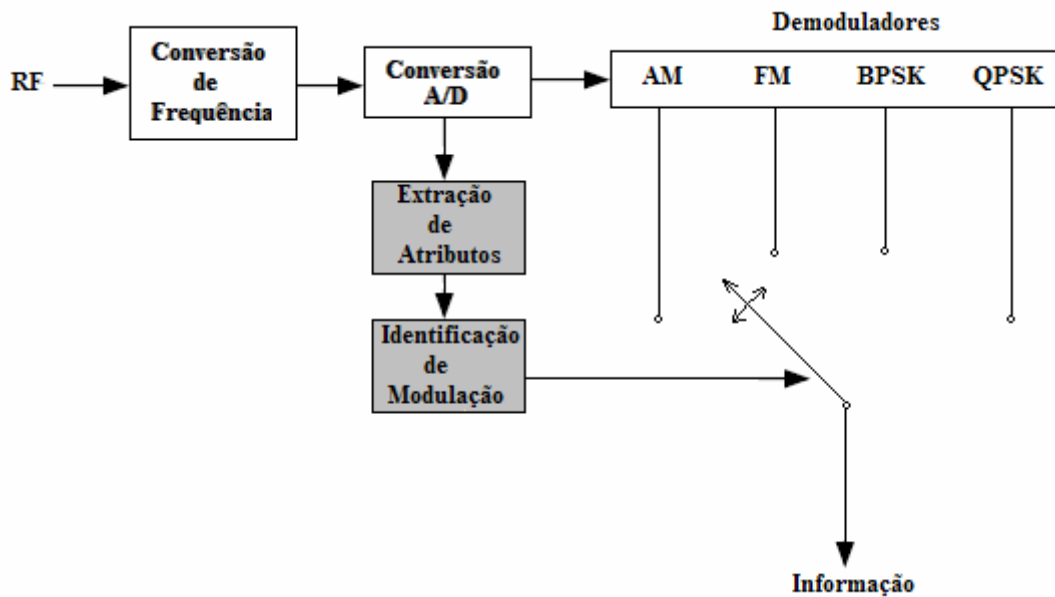


Figura 5.1: Diagrama de blocos da aplicação Demodulador Universal.

A aplicação descrita neste capítulo refere-se aos blocos achurados e identificados na figura 5.1 como “Extração de Atributos” e “Identificação de Modulação”. O bloco “Conversão de Frequência” realiza a translação de RF para FI conforme Capítulo 2. A denominação “Conversão A/D” é uma simplificação deste bloco, uma vez que o mesmo inclui, por exemplo, o processador CORDIC e os filtros de decimação visando posicionar o sinal em banda-base também de acordo com o Capítulo 2. Os demoduladores AM e FM estão presentes na biblioteca GNU-Radio, mas decidiu-se por outra implementação mais simples, baseada na resolução direta das equações do capítulo 2. Os demoduladores BPSK e QPSK

existem apenas na forma diferencial na biblioteca GNU-Radio e foram também re-implementados.

5.1 Monitor de Camada Física

Este módulo de software, atualmente implementado em Python, faz a interface direta com o hardware, recebendo as amostras complexas do USRP. Depois de configurar o bloco CORDIC, residente no FPGA, estabelecendo a frequência base do oscilador local para a tarefa de translação da frequência intermediária para banda-base, o módulo entra em um loop contínuo, coletando e analisando as amostras que vem do USRP.

O Processo de extração de atributos, representado na figura 5.2, na verdade, se resume a resolver as equações apresentadas no capítulo 4 e repetidas aqui por conveniência:

$$\gamma_{\max f} = \max[DFT(f_{cn}(n))] \quad (4.1)$$

$$\gamma_{\max a} = \max[[DFT(a_{cn}(n))]^2] / N \quad (4.7)$$

$$s^2(t) = \frac{A}{2} + \frac{A}{2} \cos(4\pi f_c t + 2\theta_0) \quad (4.13)$$

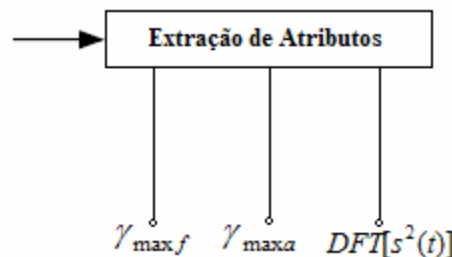


Figura 5.2: Atributos considerados para determinação do tipo de Modulação.

As amostras são recebidas do USRP pelo Monitor, numa representação complexa na forma de dois fluxos de dados (I – componente em fase e Q – componente em quadratura). Esse fluxo é dividido em blocos (quadros) e a cada bloco é associado um conjunto de atributos que são enviados para o módulo identificador de modulação.

5.2 Identificação de Modulação

Este módulo, também escrito em Python, recebe, do módulo monitor, um fluxo contínuo de dados representando os atributos considerados no capítulo 4. Sequencialmente, o módulo compara os valores dos atributos com valores de referências e indica qual o tipo provável de modulação existente no sinal.

Na figura 5.3 está representada a árvore binária, onde os atributos γ_{max-f} , γ_{max-a} e $DFT[s^2(t)]$ são testados sequencialmente e dirigem o fluxo de decisão.

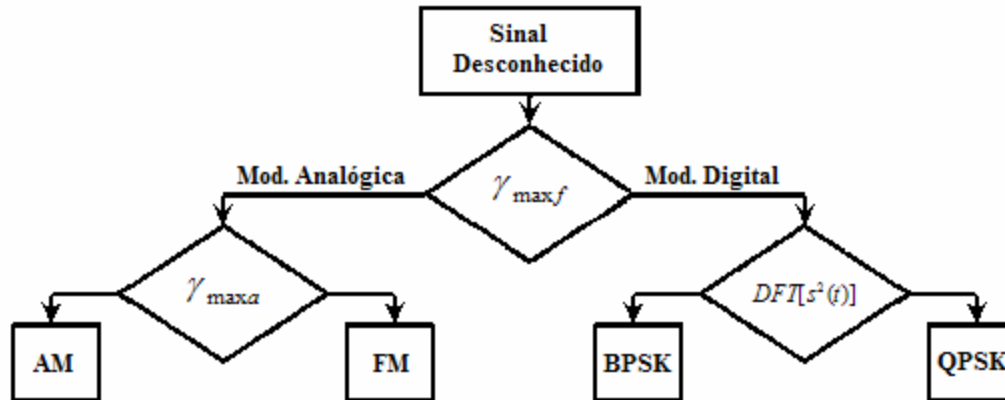


Figura 5.3: Fluxograma de decisão do Demodulador Universal.

As decisões são tomadas baseadas em valores médios de referência, observados experimentalmente através de simulações em Matlab/Simulink.

Como é possível observar na figura 5.4, os valores médios do atributo *gama-max-f*, permitem diferenciar claramente as classes de modulação, uma vez que para modulação analógica (AM e FM), o atributo situa-se em uma faixa de valores acima de 10.000, enquanto que para modulação digital (BPSK e QPSK), os valores situam-se em uma faixa abaixo de 10.000.

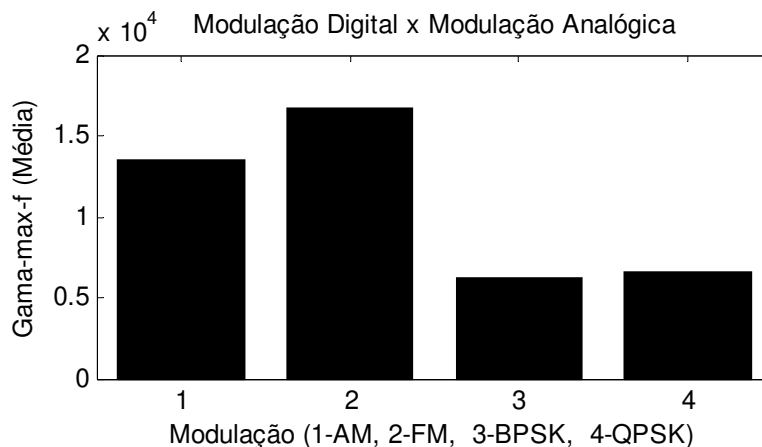


Figura 5.4: Classificação de modulação baseada no atributo *gama-max-f*.

A diferenciação entre modulação AM e FM é baseada no atributo *gama-max-a*, que está associado ao grau de variação de amplitude do sinal, conforme apresentado na figura 5.5,

onde o valor de referência situa-se em torno de 5. Valores maiores que 5 indicam modulação AM e valores abaixo de 5 indicam modulação FM.

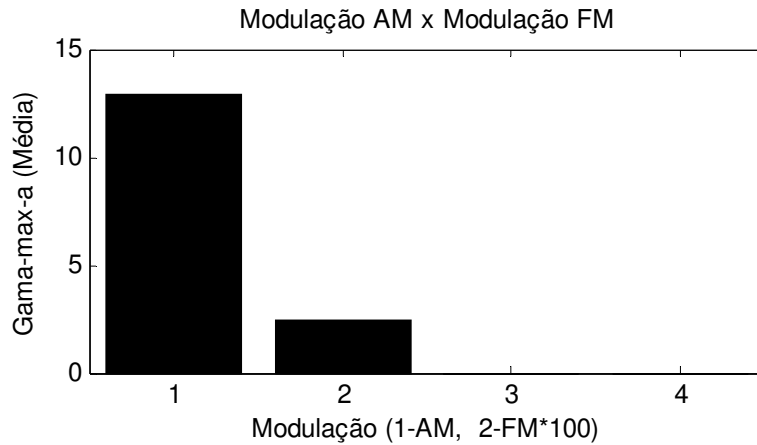


Figura 5.5: Relação entre o atributo *gama-max-a* e modulação AM/FM.

Quando o atributo *gama-max-f* indicar modulação digital, a diferenciação entre BPSK e QPSK é feita com base no conteúdo espectral do sinal elevado ao quadrado, o que permite eliminar o efeito da modulação deixando apenas um sinal com frequência igual a duas vezes a frequência da portadora caso o esquema empregado seja BPSK, conforme pode ser visto na figura 5.6.

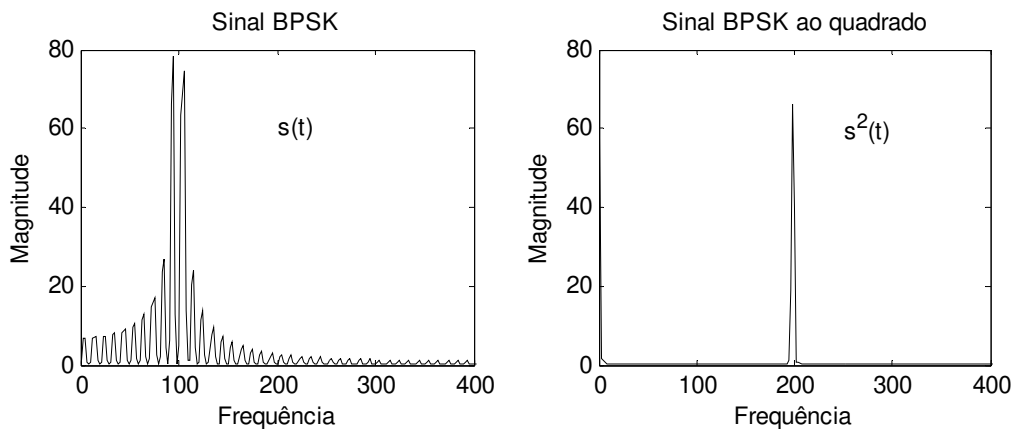


Figura 5.6: Espectro resultante da operação $s^2(t)$ para modulação BPSK.

Entretanto, se o esquema empregado na modulação for QPSK, a operação $s^2(t)$ resulta em um espectro com diversas raias espectrais, que pode ser visto na figura 5.7.

A quantidade de raias espectrais é utilizada como elemento de decisão entre modulação BPSK e QPSK, após sua localização e identificação.

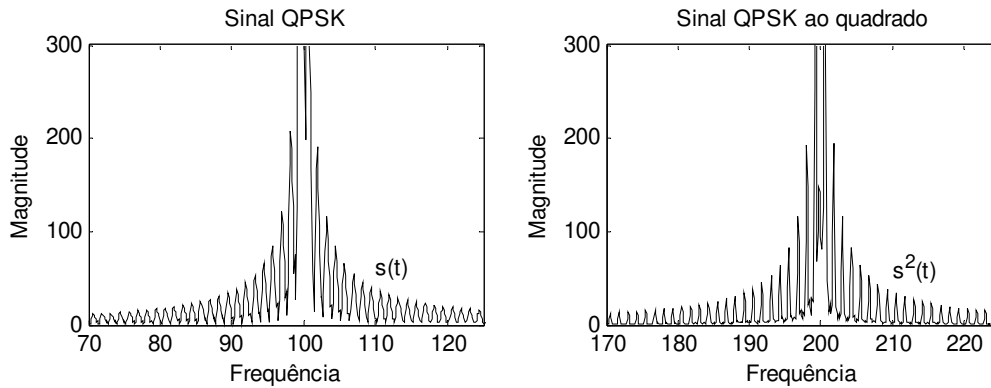


Figura 5.7: Espectro resultante da operação $s^2(t)$ para modulação QPSK.

A identificação e localização das raiais é efetuada por um algoritmo de busca de máximos (*peak-search*) adaptado de [33], cujo resultado pode ser exemplificado pela figura 5.8, onde as marcas triangulares indicam as raiais (picos) localizadas e identificadas.

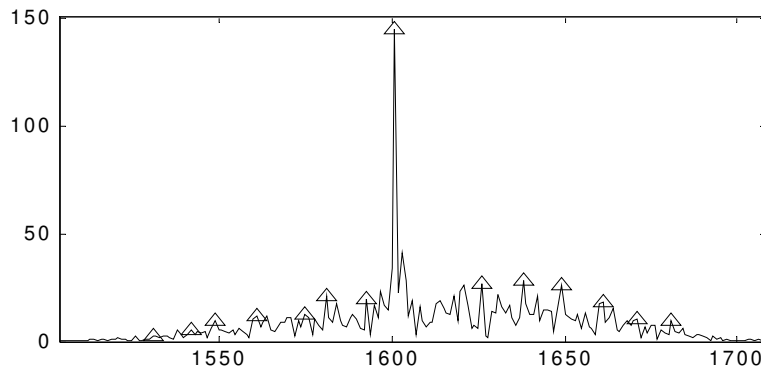


Figura 5.8: Raias espectrais identificadas no espectro de $s^2(t)$.

Um número de raiais abaixo de um determinado valor (encontrado experimentalmente), indica modulação BPSK. Se a quantidade de raiais for maior que este valor de referência, então a modulação é QPSK.

Este módulo fornece em sua saída um conjunto de identificadores de modulação (AM_Id, FM_Id, BPSK_Id e QPSK_Id) onde apenas um deles tem valor igual a 1 (hum). Os demais tem valor igual a 0 (zero).

5.3 Implementação

O Processo de desenvolvimento obedeceu o seguinte ciclo:

- 1 – Teste dos algoritmos e simulação em Matlab/Simulink;

- 2 – Porte para linguagem Python;
- 3 – Alimentação da aplicação com dados simulados (gerados em Matlab);
- 4 – Alimentação da aplicação com dados reais coletados pelo USRP.

5.3.1 Inicialização do USRP

```
#!/usr/bin/env python

# Processa sinais vindos do USRP ou de um arquivo (256ksps/complexo).

from gnuradio import gr, eng_notation
...
import math

def amr_build_graph (file_name, freq_khz): # define a aplicação
    # Frequência do Oscilador Local.
    IF_freq = 10.7e6

    # Taxa de amostragem do ADC (USRP)
    adc_rate = 64e6

    # Fator redutor de amostragem.
    usrp_decim = 250

    # Taxa de amostragem de IF.
    if_rate = adc_rate // usrp_decim

    # Reduzir a taxa de amostragem para 64 kbps
    if_decim = 4
    demod_rate = if_rate / if_decim

    audio_decimation = 2
    audio_rate = demod_rate / audio_decimation

    block_size = 256

    # Este grafo tem uma entrada (arquivo) e uma saída (audio).
    fg = gr.flow_graph ()

    # Assume dados no formato 256 kspb / complexo.
    if (file_name == "usrp"): # Dados reais (coletados pelo USRP)
        mux = 0xf0f0f0f0 # USRP Canal 0
        src = usrp.source_c (0, usrp_decim, 1, mux)
        src.set_rx_freq (0, -IF_freq)
        actual_IF_freq = src.rx_freq(0)
        src.set_pga (0, 20)
    else: # Dados simulados gerados pelo Matlab
        src = gr.file_source (gr.sizeof_gr_complex, file_name)

    ...
```

5.3.2 Monitor (Extração de Atributos)

Devido a variações de amplitude mesmo em sinais modulados em frequência (causadas por reflexões de multi-percurso), e pequenas variações de frequência em sinais modulados em amplitude (causadas pela movimentação do receptor em relação ao transmissor, por

exemplo), observadas no capítulo 4, figuras 4.4 e 4.5, as amostras oriundas do USRP são agrupadas em quadros e processadas em bloco.

Para cada bloco de 256 amostras é gerado um conjunto de atributos.

Este módulo implementa diretamente as equações definidas no capítulo 4, como pode ser visto no trecho de código a seguir que é parte da rotina produtora do atributo *gama-max-f*:

$$\gamma_{\max f} = \max[DFT(f_{cn}(n))] \quad (4.1)$$

```
s2v = gr.stream_to_vector(gr.sizeof_gr_complex, self.block_size)

phi=gr.complex_to_arg()          # fase instantânea
dphi= gr.diff_cc()              # freq_inst = derivada da fase
ma = gr.single_pole_iir_filter_ff(1.0, block_size) # freq inst média
div = gr.divide_cc()            # freq normalizada

connect (src, ddc, phi, s2v)
connect (s2v, dphi)
fmax = max(1, block_size, dphi)  # freq inst máxima

connect (dphi, (div,0))
connect (dphi,max, (div,1))
connect (dphi, ma)

sub = gr.sub_cc()

connect (div, (sub,0))
connect (ma, (sub,1))

mywindow = window.blackmanharris(self.block_size)

dft = gr.fft_vcc(self.block_size, True, mywindow)

power = 0
for tap in mywindow:
    power += tap*tap

c2mag = gr.complex_to_mag_squared(self.block_size)

connect (sub, fft)
connect (fft, c2mag)                # [gama-max-f]

. . .
```

5.3.3 Identificador de Modulação

Os atributos fornecidos pelo módulo Monitor são comparados com valores referência e, em função do resultado, os indicadores AM_Id, FM_Id, BPSK_Id e QPSK_Id recebem valores que são utilizados para selecionar o demodulador apropriado.

Os valores de referência aparecem como constantes no trecho de código a seguir:

```

Falso = 0          # zero
Verd = 1          # hum

AM_Id = Falso
FM_Id = Falso
BPSK_Id = Falso
QPSK_Id = Falso

if gama_max_f > CLASS_MOD_REF
    # modulação analógica
    if gama_max_a > AM_FM_REF
        # modulação AM
        AM_Id = Verd
    else
        # modulação FM
        FM_Id = Verd
else
    # modulação digital
    if maximos_psk > PSK_REF
        # constelação QPSK
        QPSK_Id = Verd
    else
        # constelação BPSK
        BPSK_Id = Verd
....

```

5.3.4 Demodulação AM

```

# Detetor clássico de envoltória sqrt (I*I + Q*Q)
am_det = gr.complex_to_mag()      # biblioteca GNU-Radio
fg.connect (src, ddc)            # conversão de freq
fg.connect (ddc, am_det)        # demod AM

```

5.3.5 Demodulação FM

```

# Demodulador FM em quadratura (biblioteca GNU-Radio)
fm_demod = blks.wfm_rcv (self, demod_rate, audio_decimation)
fg.connect (src, ddc)            # conversão de freq
fg.connect (ddc, fm_demod)      # demod FM
...

```

5.3.6 Seleção dos Demoduladores

Uma vez que a biblioteca GNU-Radio não dispõe, ainda, de mecanismos para carregamento dinâmico de módulos, foi necessário encontrar um mecanismo alternativo para a ativação do demodulador apropriado. Foi utilizado um artifício simples com resultado funcionalmente equivalente à carga dinâmica de módulos. O artifício se baseia em acrescentar um bloco multiplicador após cada demodulador onde a variável multiplicadora é, na verdade, o indicador correspondente àquela modulação, conforme pode ser visto no modelo Simulink apresentado na figura 5.9.

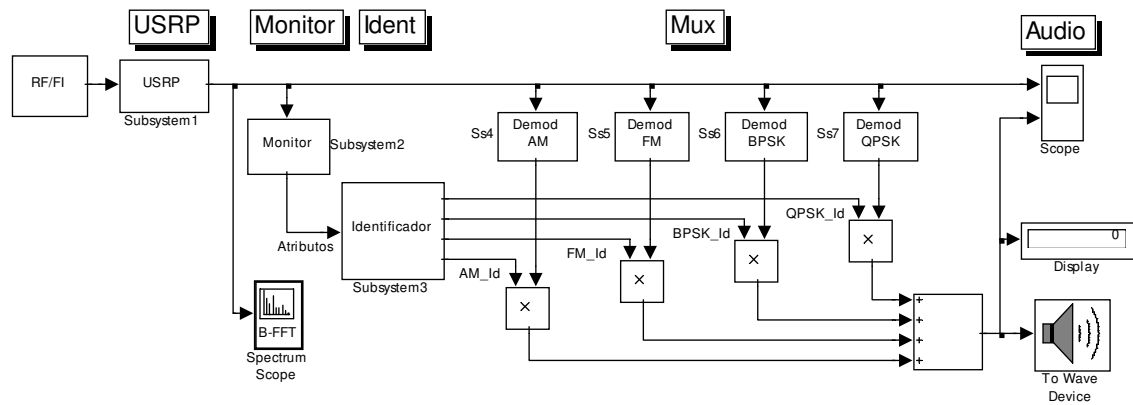


Figura 5.9: Modelo para seleção de Demodulador.

O trecho de código a seguir ilustra o mecanismo utilizado. para a escolha do demodulador.

```

...
am_sel = gr.multiply_ff()           # define os multiplicadores
fm_sel = gr.multiply_ff()
...
connect (am_demod, (am_sel,0));     # conecta a saída dos demods aos multiplicadores
connect (fm_demod, (fm_sel,0));
...
connect (AM_IDENT, (am_sel,1));     # controla a saída do multiplicador
connect (FM_IDENT, (fm_sel,1));     # controla a saída do multiplicador
connect (am_sel, fm_sel, mux);     # combina os demoduladores
...
connect (mux, audio);              # apresenta o resultado para o usuário
...

```

Capítulo 6

Resultados Experimentais

Durante a fase de testes, dois cenários foram investigados: ambiente simulado com Matlab/Simulink e ambiente real com a utilização do USRP.

6.1 Ambiente de simulação

As formas de onda com modulação analógica, foram inicialmente geradas com sinais ideais senoidais. Posteriormente o sinal modulante foi substituído por um gerador auto-regressivo de primeira ordem, recomendado na literatura [34, 35] como fonte de sinal com características similares a de sinais de áudio (voz). Esse gerador pode ser descrito pela equação 6.1,

$$y[n] = x[n] + k y[n-1] \quad (6.1)$$

onde k é uma constante igual a 0,95 limitando a largura de banda do sinal em 4000 Hz e $x[n]$ é um sinal aleatório com módulo de amplitude igual a 1. Os sinais foram gerados segundo o modelo Simulink mostrado na figura 6.1.

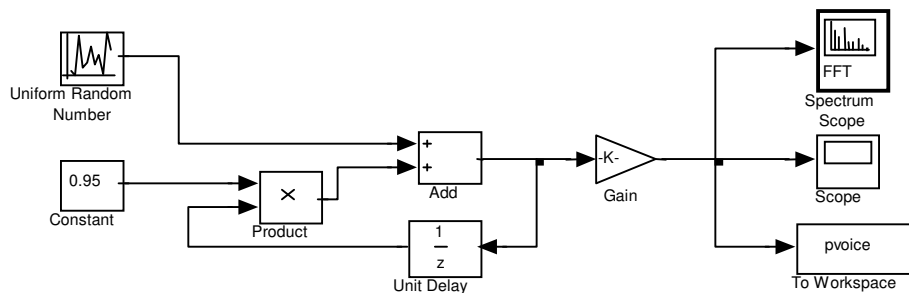


Figura 6.1: Modelo Simulink do gerador de pseudo-voz.

O sinal resultante desse gerador, pode ser visto isoladamente na figura 6.2-a e modulando uma portadora em amplitude na figura 6.2-b.

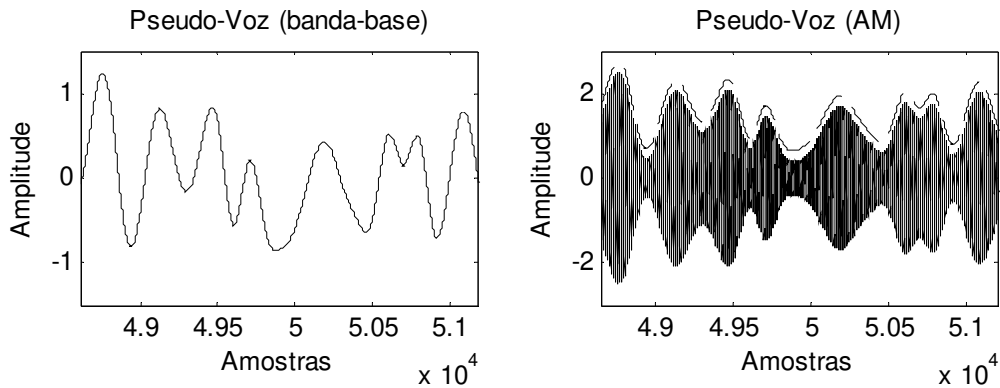


Figura 6.2: Sinal pseudo-voz (a), portadora modulada em amplitude (b).

O mesmo sinal de voz simulada modula, em frequência, a mesma portadora cujo resultado aparece nas figuras 6.3 (a) e 6.3 (b).

Os sinais modulados foram gerados de acordo com as equações apresentadas no capítulo 2, e aqui repetidas por conveniência,

$$S_{AM}(t) = A_c (1 + m \cos(2\pi f_m t)) \cos(2\pi f_c t) \quad (2.5)$$

$$s_{FM}(t) = A_c \cos(2\pi f_c t + \beta \text{sen}(2\pi f_m t)) \quad (2.9)$$

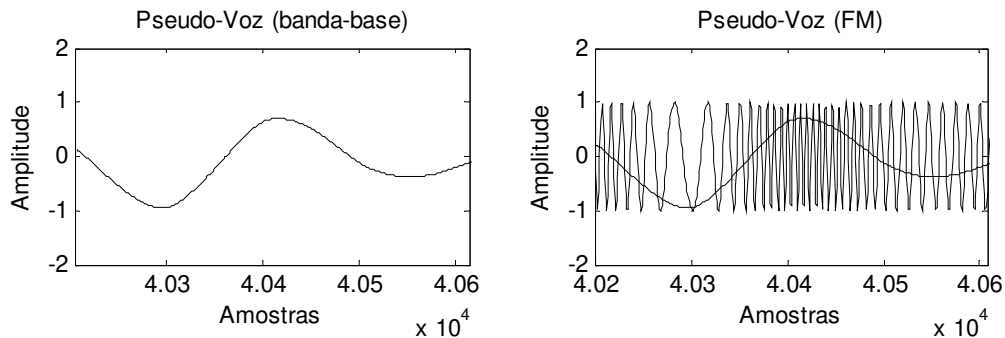


Figura 6.3: Sinal pseudo-voz (a), portadora modulada em frequência (b).

Os parâmetros de simulação foram ajustados com os seguintes valores:

Portadora: Frequência de 100 KHz, amplitude de 1Vpp e taxa de amostragem de 1 Msps.

Áudio (pseudo-voz): 0-4Khz, amplitude de 1Vpp e taxa de amostragem de 8 Ksps, com fator de interpolação 125. (taxa equivalente de amostragem igual a 1 Msps).

Símbolos: Frequência de 4 KHz, amplitude de 1Vpp e fator de interpolação 125 (taxa equivalente de amostragem igual a 1 Msps).

Deslocamento de Fase: PI (BPSK) e PI/4 (QPSK)

As formas de onda com modulação digital (BPSK e QPSK) foram geradas diretamente a partir de blocos padrão do Simulink e exportadas para o ambiente Matlab.

Os diferentes sinais modulados em amplitude, frequência e fase, foram combinados de forma aleatória, gerando o sinal de teste mostrado na figura 6.4.

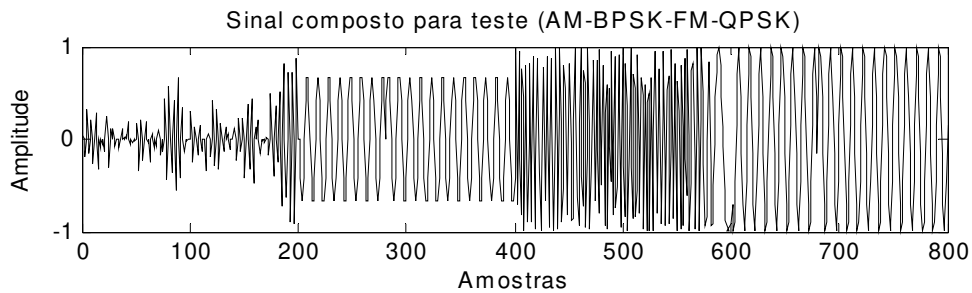


Figura 6.4: Sinal composto com 100 amostras de cada tipo – AM, BPSK, FM e QPSK.

6.1.1 Resultados de simulação

O sinal mostrado na figura 6.4 (composto de 50 segmentos com 2000 amostras de cada tipo de modulação em cada segmento) foi aplicado ao sistema sob teste e o comportamento dos atributos podem ser observados nos gráficos das figuras 6.5 ($\gamma_{\max f}$) e figura 6.6 ($\gamma_{\max a}$).

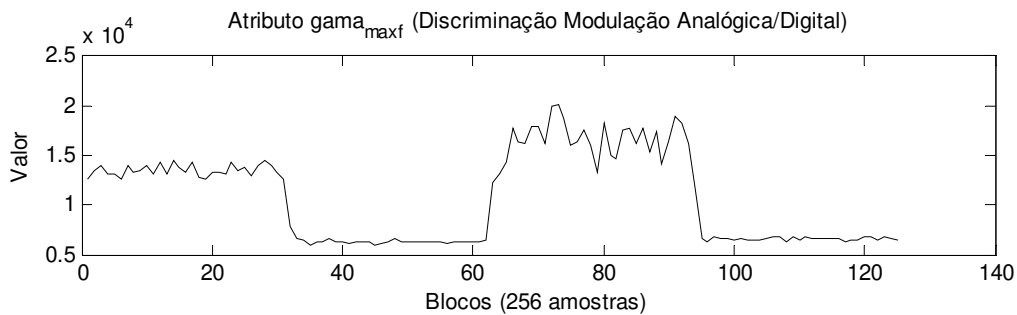


Figura 6.5: Atributo $\gamma_{\max f}$ - valores $> 10e4$ indicam modulação analógica.

A resposta do monitor de atributos condiz com a expectativa, como é possível comprovar pela observação da figura 6.5, onde o atributo $\gamma_{\max f}$ assume valores maiores que 10000 para os blocos com modulação analógica (amostras de 0 a 35 e amostras de 62 a 95) e assume valores < 10000 para modulação digital (amostras de 36 a 61 e amostras de 96 a 125).

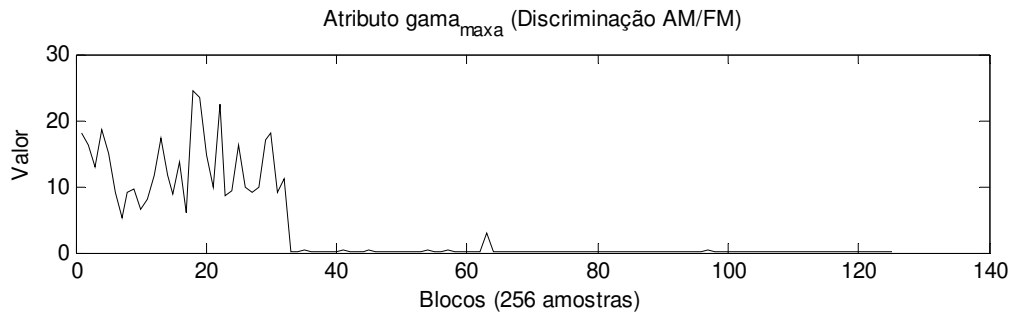


Figura 6.6: Atributo $\gamma_{\max a}$ - valores < 5 indicam amplitude constante (FM ou PSK).

A resposta do monitor com relação ao atributo $\gamma_{\max a}$ também condiz com a expectativa mas a figura 6.6 pode parecer ambígua pois como este atributo está relacionado com a variação de amplitude do sinal e os sinais com modulação digital (BPSK e QPSK) são considerados, neste estudo, sinais com amplitudes constantes, o atributo $\gamma_{\max a}$ sugere que a modulação presente nestes sinais seja FM. Este detalhe é observado pelo módulo identificador de modulação que só considera este atributo caso $\gamma_{\max f}$ indique modulação analógica.

Os resultados dos testes com sinais simulados estão apresentados na tabela 3, onde a taxa de 100% deve ser atribuída ao fato dos sinais terem características ideais sem influência de ruídos nem de reflexões.

Tabela 3: Resultados de simulação (50 testes cada).

Modulação	Classificação			
	AM	FM	BPSK	QPSK
AM	100 %			
FM		100 %		
BPSK			100 %	
QPSK				100 %

6.2 Resultados com sinais reais capturados pelo USRP

Em uma seqüência de experimentos composta de 50 testes para cada tipo de modulação, o identificador teve um desempenho médio de 88% de acertos.

Os testes foram efetuados a partir de dois arranjos diferentes: um para modulação digital e outro para modulação analógica.

Para testes de modulação digital foi utilizado o mesmo conjunto de sinais gerados pelo Matlab que foram transmitidos por um USRP e recebidos por outro USRP via cabo coaxial, conforme está mostrado esquematicamente na figura 6.7.

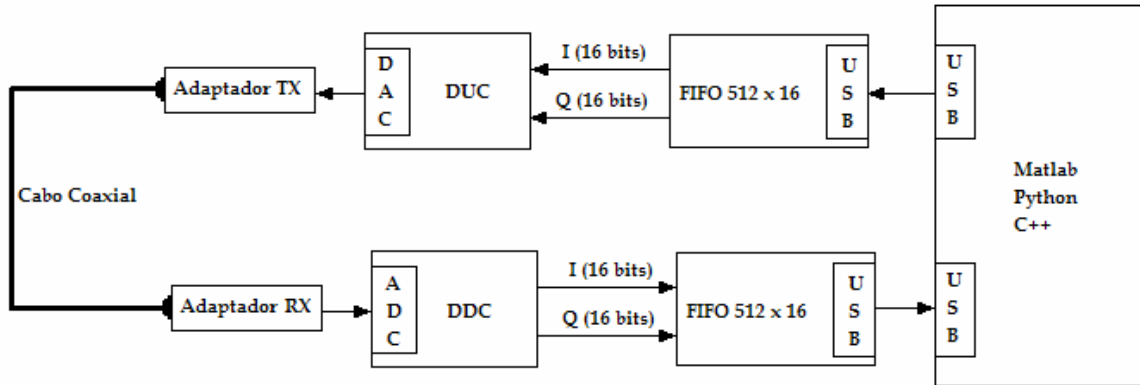


Figura 6.7: Arranjo utilizado para testes com modulação digital.

Para os testes de modulação analógica, foi utilizado um arranjo esquematizado conforme a figura 6.8, onde uma simples antena telescópica, conectada ao terminal de entrada da placa adaptadora RX, permitiu sintonizar o sistema em frequências nas faixas de estações comerciais de rádios AM e FM. Para a captura de sinais de FM foi utilizada a técnica de sub-amostragem conforme mencionada no capítulo 3.

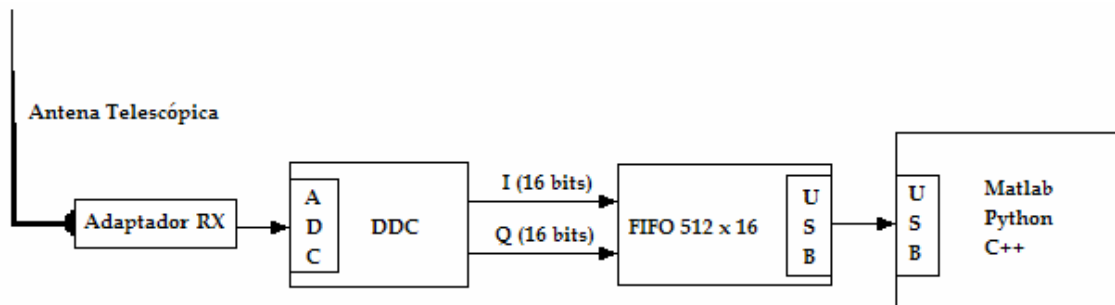


Figura 6.8: Arranjo utilizado para testes com modulação analógica.

A síntese geral dos resultados está apresentada na tabela 4, onde os erros de classificação de modulação AM como FM (6%) pode ser atribuído a um baixo índice de modulação cuja correspondente envoltória do sinal é incapaz de sensibilizar o atributo $\gamma_{\max a}$. Já os erros de classificação de FM como AM (14%) pode ser atribuído a variações indesejadas na amplitude do sinal de FM causada por efeitos do meio de transmissão. Tais variações terminam por elevar o valor do atributo $\gamma_{\max a}$.

Como os sinais com modulação digital (BPSK e QPSK) têm como característica a amplitude constante, o atributo $\gamma_{\max a}$ indica modulação analógica. É provável que os erros de classificação das modulações BPSK e QPSK como FM, sejam conseqüências da influência de ruídos e outros efeitos do meio de transmissão confundindo o cálculo do atributo $\gamma_{\max f}$.

Tabela 4 – Resultados reais de classificação (50 testes cada).

	Classificação			
Modulação	AM	FM	BPSK	QPSK
AM	94 %	6%		
FM	14%	86 %		
BPSK		10%	90 %	
QPSK		18%		82 %

6.2 Aplicação de Demonstração

Os módulos Monitor e atributos e Classificador de Modulação foram inseridos em uma interface gráfica utilizado por um dos exemplos da biblioteca GNU-Radio, o que permitiu, após pequenas adaptações na interface, à implementação de um demonstrador simples, cuja aparência pode ser vista na figura 6.9.

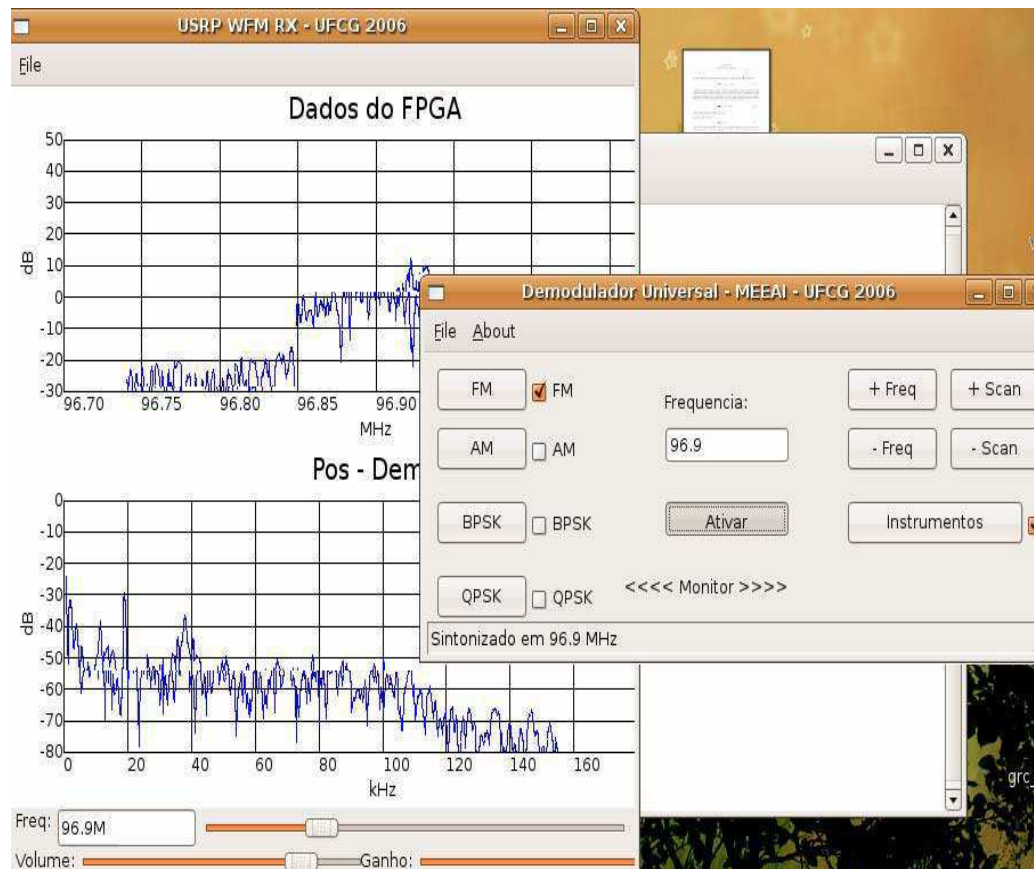


Figura 6.9 – Aplicação de demonstração do Demodulador Universal.

Capítulo 7

Conclusões e Perspectivas Futuras

Neste capítulo resumimos o trabalho executado durante o curso desse projeto de investigação da tecnologia RDS aplicada ao reconhecimento de sinais modulados.

O projeto iniciou-se com um entendimento básico da tecnologia RDS e, mais particularmente, da arquitetura de hardware experimental e software livre do projeto GNU-Radio.

Em seguida foi realizado um estudo sobre as técnicas de identificação e reconhecimento de sinais modulados.

Visando avaliar o potencial da arquitetura GNU-Radio, foi idealizado um cenário de interceptação de comunicação para o qual foi desenvolvida uma aplicação simples denominada “Demodulador Universal”.

7.1 Conclusões

O ambiente GNU-Radio revelou-se como uma ferramenta de bom nível técnico, adequada para o desenvolvimento de aplicações robustas e flexíveis no contexto de sistemas de comunicação digital, ainda que na versão corrente seja carente de métodos mais claros e consistentes de interfaceamento entre módulos e de documentação adequada para um melhor aproveitamento da ferramenta.

Após um período de aprendizagem envolvendo as técnicas de processamento digital de sinais (DSP), diversos modelos simplificados de comunicação foram implementados e testados no ambiente Matlab/Simulink. Em seguida esses modelos foram reproduzidos no ambiente de desenvolvimento apoiado pelo GNU-Radio e foi possível intercambiar dados entre os dois sistemas.

Foi investigado o problema de detecção, identificação e classificação da modulação presente em um sinal desconhecido como parte dos procedimentos para a implementação de um demodulador capaz de se ajustar e reconhecer um sub-conjunto de esquemas de modulação.

A utilização da ferramenta de software Matlab/Simulink tornou possível o teste simulado dos algoritmos sem a necessidade, em uma primeira fase, do uso de hardware nem da aplicação de codificação em Verilog (linguagem de descrição de hardware) para a implementação de módulos de processamento de sinais.

A fase de porte dos algoritmos para o ambiente real baseado em linguagem Python revelou problemas inerentes ao aprendizado de uma nova linguagem e adaptação de técnicas de

programação de uma linguagem fundamentalmente procedural para um ambiente orientado a objetos.

Como principal deficiência do ambiente GNU-Radio é digno de nota a falta de suporte ao carregamento dinâmico e execução dessincronizada de processos opcionais (idéia central do demodulador universal), forçando a criação de um mecanismo alternativo para a implementação da multiplexação dos diferentes demoduladores. Esse mecanismo peca pelo baixo desempenho uma vez que, na verdade, todos os demoduladores estão ativos ao mesmo tempo consumindo recursos de processador e memória.

Ainda assim, o desempenho geral do processador envolvido (Computador Pessoal) foi considerado aceitável, tornando desnecessária a implementação do algoritmo ou de partes do algoritmo em hardware (FPGA) como a princípio imaginado. Todo o processamento, desde a extração dos atributos dos sinais até os procedimentos de modulação propriamente ditos, foram implementados em Python no Computador Pessoal.

Como parte do processo de aprendizagem e visando facilitar a criação de sinais de estímulos para testes, foram implementados em FPGA e integrados ao ambiente GNU-Radio, um gerador de sinais e moduladores digitais BPSK e QPSK. Apesar de não serem absolutamente necessários ao projeto, tais módulos contribuíram para o descongestionamento do canal de comunicação entre o Computador Pessoal e o USRP uma vez que a interface USB dos dois sistemas tem limitação de largura de banda. A geração de sinais de testes localmente no FPGA libera totalmente o canal USB para o fluxo de dados no sentido USRP -> PC permitindo uma maior taxa de transferência de dados no contexto desse projeto.

O demodulador teve um desempenho razoável para uma primeira experimentação com a plataforma e os resultados foram encorajadores. A taxa de acerto do demodulador apresentou uma taxa de acerto em torno de 88 %, o que pode ser considerado razoável uma vez que os relatos na literatura indicam taxas que variam entre 60 % e 98 %.

7.2 Perspectivas Futuras

As possibilidades são amplas, cobrindo um espectro que vai do estudo de metodologias para desenvolvimento de software, passando por algoritmos específicos de DSP e chegando a eventuais adaptações do hardware para condições particulares voltadas para comunicações, instrumentação, experimentação e ferramenta didática para cursos de graduação e pós-graduação.

Alguns tópicos específicos que chamaram a atenção do autor:

- A falta de uma ferramenta para desenvolvimento estruturado de aplicações em RDS indica que existem oportunidades para o desenvolvimento de metodologias e sistemas que permitam, por exemplo, a integração de editores baseados em UML que possam gerar descrição de alto nível em XML para serem, então, executadas pelo interpretador Python sob ambiente Linux.

- A total falta de suporte em ambiente Windows. Seria de grande valor a criação de uma sistema equivalente ao GNU-Radio para o ambiente Windows. Este desafio incorpora atividades relacionadas com desenvolvimento de drivers USB, criação de bibliotecas de processamento de sinais de alta velocidade e, eventualmente, permitiria a integração direta com ferramentas como Matlab/Simulink e LABView.
- Apesar da plataforma experimental contemplar os blocos básicos para comunicações como Digital Down Converter, Digital Up Converter, Filtros Interpoladores e Filtros Decimadores, observa-se a ausência de módulos sincronizadores que poderiam ser implementados na forma de PLL digitais e que serviriam para a implementação de demoduladores síncronos, por exemplo.
- A implementação, em hardware, do extrator de atributos e, eventualmente, do classificador de modulação permitiria a liberação do processador do Computador Pessoal para a execução de aplicações mais complexas.
- Os efeitos de ruídos e efeitos do meio de transmissão devem ser considerados e métodos para mitigá-los devem ser estudados, aproximando o demodulador universal de uma realidade mais prática.
- O desenvolvimento de algoritmos inovadores que explorem e que possam ser integrados a processadores CORDIC poderia trazer benefícios pela otimização de área utilizada no chip (FPGA) e, eventualmente, pela redução do consumo de energia.
- Exploração de tecnologias de Antenas Inteligentes visando complementar o sistema de identificação de sinais com informações de direção de origem dos sinais sob investigação.
- Aplicações de telemetria baseadas em redes de sensores inteligentes poderiam se beneficiar dos recursos da tecnologia RDS aproveitando sua flexibilidade para a implementação de redes *Ad-Hoc* com protocolos de auto-roteamento, por exemplo.

Referências Bibliográficas

- [1] Elsayed Elsayed Azzouz and Asoke Kumar Nandi. *Automatic Modulation Recognition of Communication Signals*. Kluwer Academic Publishers, 1996.
- [2] Lainé, Ph., Drevon, N., Cannet, J-M., *Networks Models for Converged Fixed and Mobile Telephony*, Alcatel Telecomm Review, 1st Quarter 2005.
- [3] W. Tuttlebee, *Software Radio Technology: A European Perspective*, IEEE Comm. Mag., Feb. 1999.
- [4] J. Pereira, *Re-Defining Software (Defined) Radio: Re-Configurable Radio Systems and Networks*, IEICE Trans. on Comm., vol. E83-B, no. 6 pp. 1174, 2000.
- [5] J. Mitola III, G.Q. Maguire Jr., *Cognitive Radio: Making Software Radios more Personal*, IEEE Pers. Communications , vol. 6, no. 4, 1999.
- [6] WP5 Deliverable 5.1, *Requirements, Interfaces and Specification of the Proof of Concept*, End-to-End Reconfigurability II (E2ER-II), Aug. 2006.
- [7] Rao, R., Zhu, W., et-al, *Multi-antenna Testbeds For Research and Education in Wireless Communications*, IEEE Communications Magazine, 2004.
- [8] Ray Andraka, *A survey of CORDIC algorithms for FPGA based computers.* , FPGA '98. Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, 1998.
- [9] Agilent Technologies, *Digital Modulation in Communications Systems – An Introduction*, Application Note 1298, May 2002.
- [10] Simon Haykin, *An Introduction to Analog and Digital Communications*, John Wiley & Sons, New York, 1989.
- [11] A. B. Carlson, *Communicatin Systems*, 3rd Edition, McGraw Hills, 1986.
- [12] Nolan, K.E., Doyle, L., O'Mahony, D., Mackenzie, P., *Signal space based adaptive modulation for software radio*, Wireless Communications and Networking Conference, WCNC2002, 2002.
- [13] M. K. Nezami, *RF Architectures & DSP aspects of Digital Wireless Transceivers, (Draft)*, Raython, Inc., 2003.
- [14] J. H. Reed, *Software Radio: A Modern Approach to Radio Engineering*, Prentice Hall, Upper Saddle River, NJ, 2002.

- [15] Vanu, Inc. *Homepage*, <http://www.vanu.com>, Jul-2006.
- [16] *GNU Radio Project Homepage*, <http://www.gnu.org/software/gnuradio>, Nov-2005.
- [17] *Microtune 49xx*, [http://www.microtune.com/products/49xx Series.html](http://www.microtune.com/products/49xx_Series.html), 2004.
- [18] *Measurement Computing Website*, <http://www.measurementcomputing.com>, Ago-2005.
- [19] M. Ettus, *Ettus Research LLC Homepage*, <http://www.ettus.com>, Jan-2007.
- [20] Ferreira-Sobrinho, A. M., Rocha-Neto, J. S., *Plataforma Experimental FPGA-Matlab para Antenas Inteligentes*, Congresso Nacional de TI e Comunicações, Sucesu@2006, Jul-2006 (adiado para 2007).
- [21] AD9862 Datasheet, Analog Devices, <http://www.analog.com>.
- [22] Jack E. Volder, *The CORDIC Trigonometric Computing Technique*, IRE Transactions on Electronic Computers, September 1959.
- [23] S. Nahm, K. Hun, W. Sung, *A CORDIC-Based Digital Quadrature Mixer : Comparison With a ROM-Based Architecture*, School of Electrical Engineering, Seoul National University, 1998
- [24] NCO Compiler, MegaCore Function User Guide, Altera, Inc., June 2004.
- [25] E. B. Hogenauer, 'An economical class of digital filters for decimation and interpolation'. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol ASSP-29, 1981.
- [26] Chung-Yu Huang and Andreas Polydros. *Likelihood methods for mpsk modulation classification*. IEEE Transactions on Communications, 43(2/3/4):1493–1504, 1995.
- [27] Samir S. Soliman and Shue-Zen Hsue. *Signal classification using statistical moments*. IEEE Transactions on Communications, 40(5):908–916, 1992.
- [28] J. L. Perry T. G. Callaghan and J. K. Tjho. *Sampling and algorithms aid modulation recognition*. Microwaves RF, 24(9):117–119, 121, 1985.
- [29] Le, B., Rondeau, T. W., Maldonado, D., Bostian, C. W., *Modulation Identification Using Neural Networks for Cognitive Radios*, Software Defined Radio Technical Conference (SDR05), USA, 2005.
- [30] Abdulkadir, E., Hanif, G., *An Education Interface for Automatic Recognition of Analog Modulation Signals*, Journal of Applied Sciences 5 (3): 513-516, 2005.

- [31] Iversen, A., *Classification of Digital Modulation Schemes using Multi-layered Perceptron*, Herriot-Watt University, 2004.
- [32] Keyth E. Nolan, Linda Doyle, *Modulation Scheme Recognition Techniques for Software Radio on General Purpose Processors*, Proceedings of IEI/IEE Symposium on Telecommunications Systems Research, Dublin, 2001.
- [33] S. Fischer, R. Klinkenberg, I. Mierswa, *YALE - Yet Another Learning Environment*, Computer Science Department, University of Dortmund, Germany
<http://yale.cs.uni-dortmund.de/java>
- [34] Dubuc, C, Boudreau, D., Patenaude F., Inkol R., *An Automatic Recognition Algorithms for Spectrum Monitoring Applications*, Communication Research Center, Canada, 1995.
- [35] Azzouz, E. Nandi, A., *Automatic Identification of Digital Modulations Types*, IEEE-Signal Processing, 1995.

A – Anexos

A.1 Instalação do kit GNU-Radio (Software)

Este apêndice descreve os passos necessários para instalar o pacote GNU-Radio em um sistema baseado no sistema operacional Linux, distribuição Ubuntu 6.06.

1 – Instalação do sistema operacional Linux Ubuntu 6.06, atualizado.

2 – Instalação/Atualização de pacotes auxiliares:

Os seguintes módulos ou utilitários são necessários para a correta construção de um sistema GNU-Radio. Todos esses programas estão incluídos na distribuição Linux Ubuntu 6.06. Entretanto, como existe uma grande interdependência de bibliotecas, configurações e drivers, é recomendado que se proceda uma atualização de todos esses utilitários via aplicativo *apt-get*.

- SWIG

Utilitário para a criação de encapsulamento de funções C++, integrando-as ao sistema Python.

- Python (incluindo python-dev e bibliotecas *numpy* e *numarray*)

Parte do sistema GNU-Radio está escrito em Python.

- GCC, G++ (versão 3.2, 3.4 ou maior. Versão 3.3 conflita com GnuRadio)

Compiladores necessários para compilar GNU-Radio a partir do código fonte.

- Make, Autoconf, Libtools

Utilitários necessários para reconstruir GNU-Radio a partir o código fonte.

- FFTW

Biblioteca que implementa a transformada rápida de Fourier.

- SDCC

Montador utilizado para compilar o firmware do FX2 em código nativo 8051.

- FXLOAD

Não é parte do GNU-Radio mas é útil para a manipulação do FX2 diretamente da linha de comando.

- USBVIEW

Também não é parte do GNU-Radio mas é útil para depurar interação com dispositivos USB.

3 – Instalação dos módulos GNU-Radio.

O sistema GNU-Radio é composto de diversos módulos baseados nas linguagens C , C++ e Python. Estes módulos são encontrados no repositório de códigos e carregados, via Internet, a partir do URL <http://gnuradio.org/trac/wiki>. O sistema é disponibilizado de duas formas : Tarball e SVN.

Tarball: Disponibilizada na forma de arquivos compactados, um para cada parte do GNU-Radio. Estes arquivos devem ser descompactados e instalados de acordo com instruções específicas que os acompanham.

SVN: Acesso direto a um sistema flexível para controle de versões e utilizado por usuários mais avançados e que precisam dos recursos mais novos ou de versões mais atualizadas.

Neste projeto consideramos apenas a forma *Tarball*.

Módulos GNU-Radio:

- gnuradio-core – Código do núcleo do GNU-Radio
- gr-usrp – Interfaces para acesso e controle do USRP
- gr-audio-alsa-0.5.orig – Interface ALSA para adaptador de áudio do PC
- gr-wxgui – Interface para desenvolvimento de aplicações gráficas
- gr-audio-oss – Interface Open-Sound System para adaptador de áudio do PC
- usrp-0.12 – Acionadores de baixo nível para controle do USRP
- gr-howto-write-a-block - Exemplos e tutoriais

4 – Configuração e Compilação do sistema GNU-Radio.

Um conjunto de *scripts* se utilizam dos utilitários Make, Autoconf e libtools para automatizar o processo de adequação ao hardware e compilação das diferentes partes do sistema. Uma vez que todos os módulos/pacotes tenham sido carregados pode-se disparar o processo de compilação e geração do sistema GNU-Radio:

```
./configure; # interroga o ambiente e cria arquivos temporários de configuração
$make check; # garante que todos os pré-requisitos estejam satisfeitos
$make;       # compilar o sistema (espere... espere... espere...)
$make install; # instalar os arquivos binários em seus respectivos diretórios.
```

5 – Experimentand GNU-Radio

Durante o processo de compilação e instalação, diversas pastas são criadas para o correto funcionamento do sistema. A pasta mais importante é *gnuradio/examples*, onde diversas aplicações exemplo estão residentes e servem como tutorial para um primeiro contato com o GNU-Radio.

A.2 Datasheets da plataforma USRP (Hardware)

Os diagramas esquemáticos do USRP estão em anexo nos seus respectivos arquivos em formato *.pdf* cujas versões mais atualizadas podem ser encontradas no repositório do projeto GNU-Radio (<http://gnuradio.org/trac/wiki>).

fpga.pdf – Circuito principal mostrando o FPGA e suas conexões.

rxmod.pdf – Adaptador de interface analógica de entrada (RX Adapter).

txmod.pdf – Adaptador de interface analógica de saída (TX Adapter).

clock.pdf – Oscilador a cristal e gerador de temporização.

power.pdf – Circuito de alimentação de energia.

inteface.pdf – Conectores e conexões.

ad9860_62_0.pdf – Processador misto de sinais (ADC, DAC e DUC).