



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-13033-PRE/8310

**ESTUDO E DESENVOLVIMENTO DE UM RECEPTOR DIGITAL
PARA A ESTAÇÃO INPE-CRN**

Wallace Alves de Andrade Rocha*

*Bolsista UFRN

Relatório Final de Projeto de Iniciação Científica (PIBIC/CNPq/INPE), orientado pelo
Dr. Manoel Jozeane Mafra de Carvalho

INPE
São José dos Campos
2005



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

ESTUDO E DESENVOLVIMENTO DE UM RECEPTOR DIGITAL PARA A ESTAÇÃO INPE-CRN

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA (PIBIC/CNPq/INPE)

Wallace Alves de Andrade Rocha (UFRN, Bolsista PIBIC/CNPq)
E-mail: wallace@crn.inpe.br

Manoel Jozeane Mafra de Carvalho (INPE, Orientador)
E-mail: manoel@crn.inpe.br

Maio de 2004

RESUMO

Este projeto consiste em Pesquisa de um sistema para tratamento de sinais de rádio através de software, para ser empregado no desenvolvimento de uma estação de recepção de satélite, visando uma solução nacional e de baixo custo. A pesquisa fundamenta-se em revisão bibliográfica e análise de softwares similares para outras aplicações de livre distribuição e código aberto, basicamente o GNU Radio, projeto GNU para aplicações de rádio digital, e o PSpectra, conjunto de bibliotecas de tratamento de sinal. Os resultados já obtidos são: Conhecimento sobre tratamento de sinais digitais por software, sobre o funcionamento e estrutura do Gnuradio e PSpectra. Documentação do Gnuradio e de algumas de suas aplicações. Rotina de modulação para sinais analógicos, implementado em C++ para ser usado em testes. Rotina de modulação para sinais digitais, simulando e modulando dados de uma PCD – Plataforma de Coleta de Dados, em C++ e LabView. Desenvolvimento de aplicações de tratamento de sinais sobre o Gnuradio, inclusão de um bloco de Costas Loop na biblioteca Gnuradio e utilização do mesmo em aplicação de demodulação em fase. de equipamento (PC) capaz de adquirir dados através da placa MC4020 em sua plena capacidade (20 MS/s) e armazená-los em disco sem erros ou perda de dados. O próximo passo do projeto está centrado em otimizar e adaptar o sistema para demodulação e tratamento posterior em tempo real de sinais de satélite para a estação INPE/CRN de forma a possibilitar uma aplicação que atenda os requisitos de flexibilidade e robustez.

SUMÁRIO

1. Introdução	1
1.1. Sobre o Projeto	1
1.2. Estação de Recepção Atual	2
2. Fundamentação Teórica	4
2.1. O que é SDR	4
2.2. GNU Radio	6
2.3. Algoritmo Costas Loop	6
3. Desenvolvimento	8
3.1. Introdução a DSP baseado em software	8
3.2. Estudo de bibliotecas DSP	8
3.3. Configuração de placa de aquisição de dados e GNU Radio	8
3.4. Estudo do bloco Costas Loop do GNU Radio	9
3.5. Testes de desempenho do sistema de armazenamento de dados	9
3.6. Utilização de Serial ATA e RAID	10
3.7. Instalação GNU Radio	10
4. Resultados Obtidos	12
4.1 - Próximos Passos	15
5. Conclusão	16

1 - Introdução

O presente projeto consiste em Pesquisa e desenvolvimento de um sistema para demodulação e tratamento de sinais de rádio através de *software*, para ser utilizado em estações de recepção de satélite, visando barateamento, simplificação e flexibilização do equipamento utilizado nesse processo.

A pesquisa fundamenta-se em materiais presentes na Internet, e análise de *softwares* similares para outras aplicações de livre distribuição e código aberto, para embasamento teórico e possível aplicação de bibliotecas existentes no sistema a ser desenvolvido.

1.1 - Sobre o projeto

O projeto descrito neste relatório tem como objetivo estudo e desenvolvimento de um receptor digital realizado em *software* para a estação INPE-CRN.

Todo *software* desenvolvido deve atender ao licenciamento GNU-GPL e operar em ambiente Linux.

Ao final o *software* do receptor será constituído de dois grandes blocos independentes. O primeiro rodando em *hardware* externo (FPGA) e em tempo real, será responsável pela compatibilização da saída do conversor analógico/digital com a entrada do PC.

O segundo grande bloco de *software* deve rodar em uma plataforma PC, com Linux e utilizar as bibliotecas de *software* livre disponíveis através do projeto *GNU Radio* (descrito mais a frente). Esse por sua vez se subdivide em dois módulos principais: tempo real e processamento posterior, conforme indicado na figura 1 a seguir.

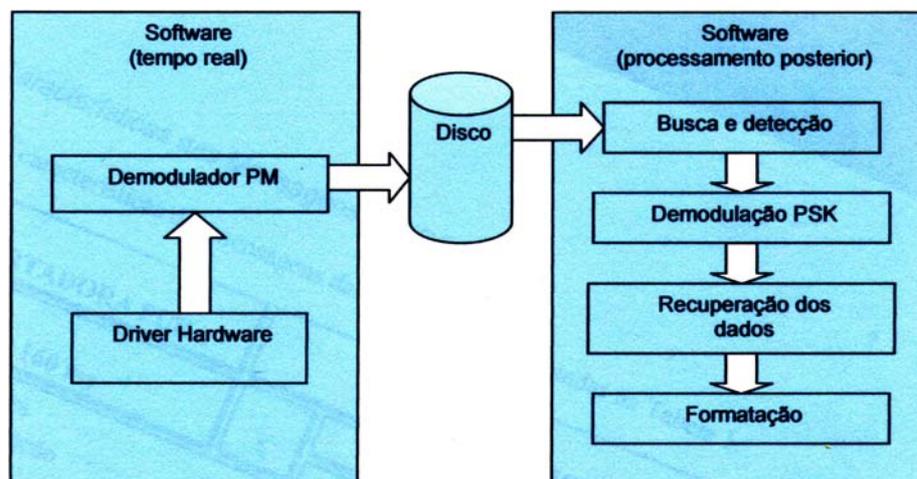


Figura 1: Diagrama para o receptor realizado em *software*

- *Software* Tempo Real

O *Software* realizado em tempo real é o foco do trabalho descrito nesse relatório, e tem três finalidades básicas:

1. Transferir o sinal analógico digitalizado pela placa ADC (nesse trabalho utilizada a PCI DAS-4020) para a memória do computador sem perda de dados;

2. Demodular o sinal. A demodulação em fase da portadora realizada em tempo real recupera o sinal em banda base que contém os eventuais sinais das PCDs;
3. Gravar em disco o sinal demodulado para processamento posterior.

- *Software* para processamento posterior

Ao final da passagem do satélite, efetua-se a busca e a detecção do sinal das PCDs as quais é associado um canal de processamento e recuperação dos dados. As funções de busca e detecção, bem como as de processamento do canal são implementadas em *software* rodando no mesmo PC. Uma vez recuperados, os dados são formatados e armazenados em um banco de dados instalado em um outro microcomputador via rede TCP/IP, que por sua vez disponibilizará aos usuários através de uma interface com uma rede local (LAN).

1.2 - Estação de Recepção Atual

No INPE-CRN existe atualmente uma Estação Solo inicialmente instalada para operar com os micro-satélites SACI-1 e SACI-2 (micro-satélite para aplicações científicas, desenvolvido pelo INPE). A Estação Solo encontra-se atualmente em regime inoperante, dado a perda dos satélites SACI-1 e SACI-2 em seu lançamento. Do ponto de vista físico, a Estação Solo de Natal consiste basicamente em dois blocos, conforme apresentado na figura 2.

O primeiro bloco está localizado em uma estrutura especial em cima do prédio da estação e é composto pelo SISTEMA DE ANTENA que inclui um refletor parabólico com 3m de diâmetro, a fonte, um duplexador localizado atrás do refletor e um posicionador mecânico. O segundo bloco está localizado em uma sala específica para a estação e é composto de um *rack*, que agrupa os subsistemas de TELECOMANDO, TELEMETRIA, ACIONAMENTO E CONTROLE DO POSICIONADOR e REFERÊNCIA DE TEMPO/FREQÜÊNCIA, e três microcomputadores compatíveis com IBM PC: um para o subsistema de TELECOMANDO, um para o subsistema de TELEMETRIA e outro para executar as funções de pilotagem da antena e controle da estação.

Em conjunto com o projeto descrito nesse relatório, está sendo desenvolvido um novo sistema de controle de posicionamento da antena. Os sistemas de TELEMETRIA e TELECOMANDO, específicos para o projeto SACI, foram desacoplados.

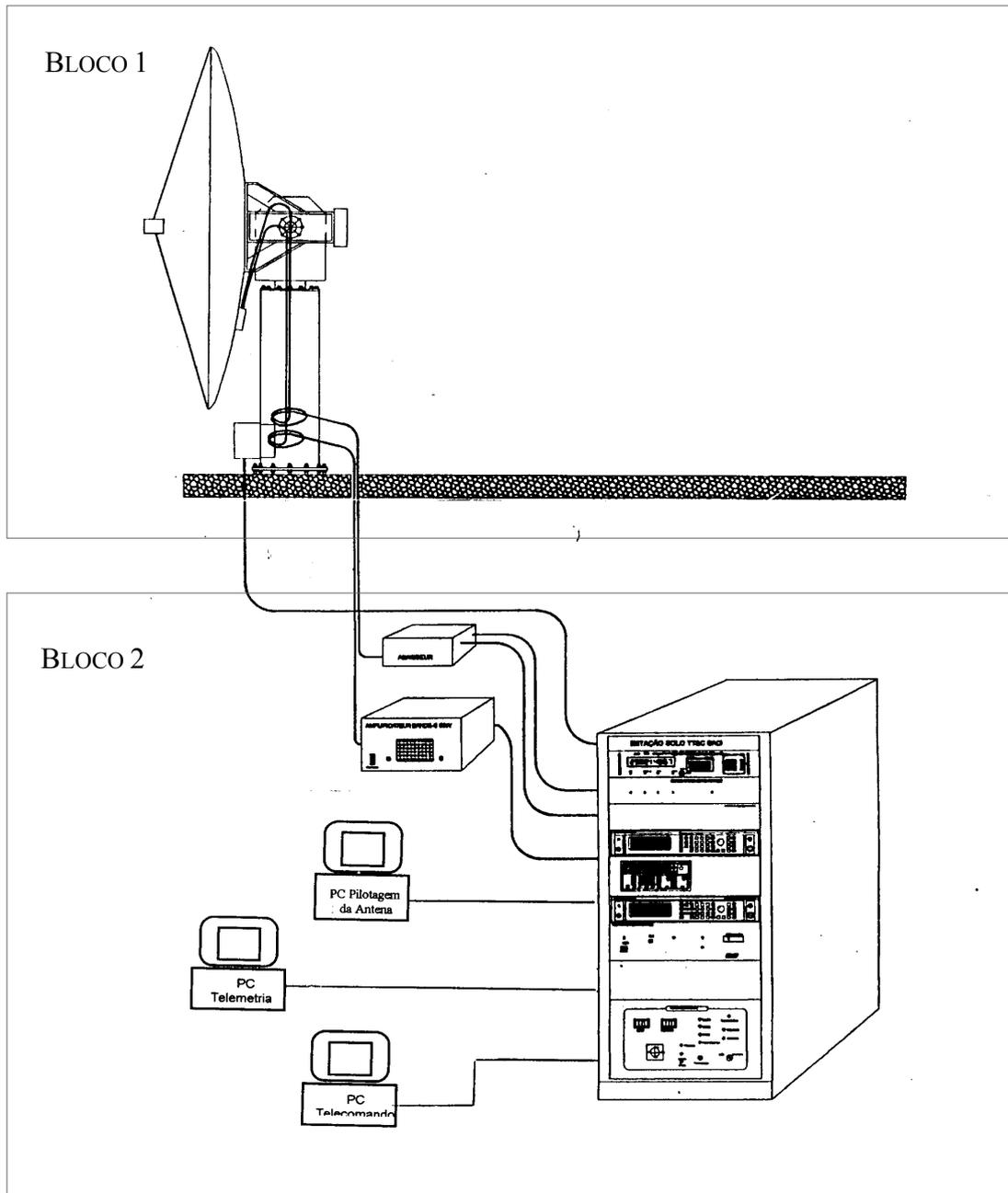


Figura 2: Visão geral da Estação Solo para o micro-satélite SACI-1

2 – Fundamentação Teórica

2.1 - O que é SDR

SDR, do inglês *Software Defined Radio* significa sistema de rádio baseado em *software*, basicamente, em sistemas desse tipo, substituem-se módulos de *hardware* convencionais, como filtros, demoduladores e componentes de interpretação de sinais e dados, por módulos em *software* sobre *hardware* não especializado capazes de realizar operações matemáticas sobre o sinal de rádio obtido através de placa de aquisição de dados simulando cada módulo de *hardware* normalmente necessário.

Sistemas SDR, possibilitam uma grande flexibilidade, permitindo grandes adaptações a novas tecnologias sem a necessidade de substituição de equipamentos, um sistema de televisão por exemplo, poderia se auto-adaptar a novos padrões, de forma que quando o equipamento receptor percebesse um sinal em um padrão diferente dos reconhecidos, tentaria fazer o download de um novo módulo que atendesse o padrão detectado para incorporar ao seu sistema. Esses sistemas também agregam conceitos como uso de múltiplos canais, e até múltiplos padrões demodulados simultaneamente.

Uma outra grande vantagem de sistemas SDR é a melhor utilização do espectro. Visto que a maioria dos sistemas atuais são analógicos, o espectro eletro-magnético está praticamente todo ocupado, sendo poucas as frequências ainda livres e que possam ser usadas para novas aplicações. Os mais modernos sistemas digitais, mesmo de “banda-larga” necessitam de uma banda bem mais estreita do espectro para uma melhor qualidade e oferta de serviço. Portanto, o futuro das telecomunicações é migrar para os novos sistemas digitais e reaproveitar o espectro. Como exemplo, vemos o sistema de DVT-T (Televisão Digital Terrestre) em que o espectro ocupado por centenas de canais digitais, até em alta-definição, é inferior ao usado atualmente para menos de uma dezena de canais analógicos. [2]

As pesquisas para desenvolver o SDR começaram na década de 90, quando o programa SPEAKeasy do governo americano procurava desenvolver aparelhos de rádio capazes de alternar entre sistemas incompatíveis usados por cada braço das forças armadas. Dez anos depois, o aumento do poder de processamento dos aparelhos móveis leva a indústria dos telefones celulares a se voltar ao SDR a fim de criar sistemas de telefonia mais flexíveis e lucrativos. [2]

Nas figuras abaixo é mostrado um exemplo para diferenciar um sistema receptor baseado em *hardware*, e um sistema baseado em *software*. Trata-se de um sistema de recepção *multi-modo*, ou seja um sistema para captação de padrões diferentes simultaneamente.

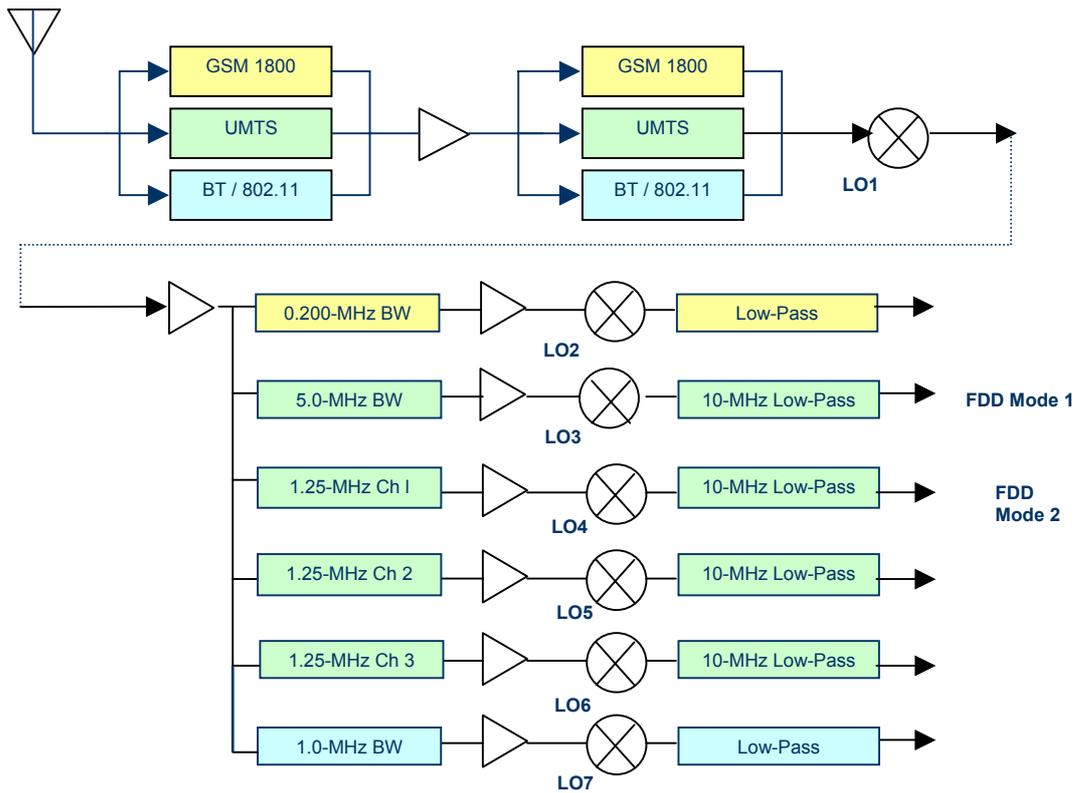


Figura 3: Sistema de recepção Multi-modo baseado em *hardware* [3]

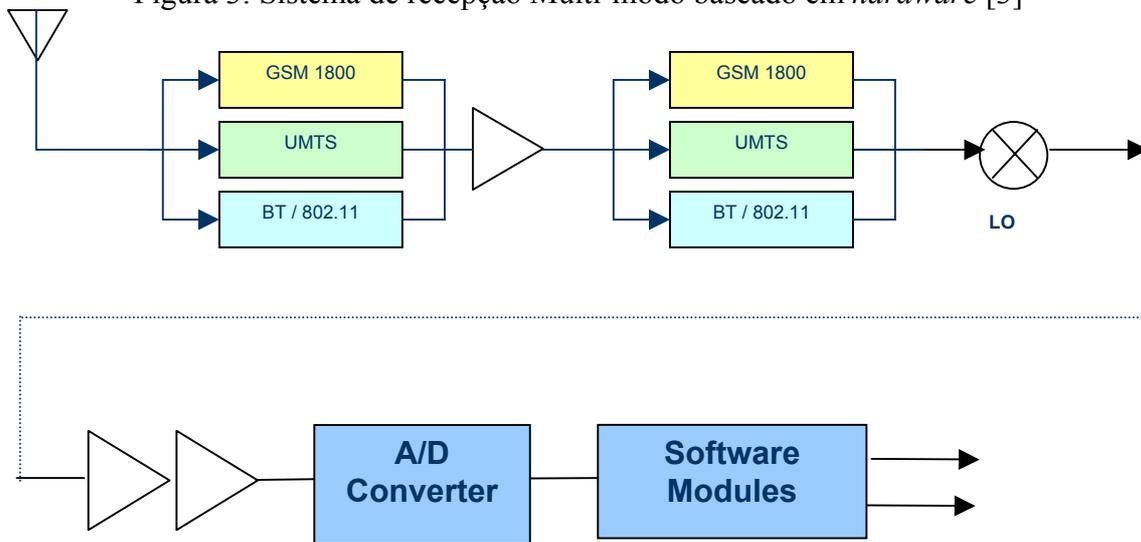


Figura 4: Sistema de recepção Multi-modo baseado em *software* [3]

No sistema baseado em *hardware* (Figura 3) existe bem mais *hardware* dedicado, para cada canal, um conjunto de *hardware* é necessário. Enquanto notamos no sistema por *software* (Figura 4), que o sinal após amplificado, é digitalizado, e todo o processamento dos padrões, filtragem de canais e demodulação passa a ser feita digitalmente em módulos de *software*, em um único *hardware* de processamento genérico, que pode ser por exemplo, um computador.

2.2 GNU Radio

GNU Radio é um projeto da GNU, de livre distribuição e código aberto. O *GNU Radio* fornece uma biblioteca de blocos de processamento de sinal e as ferramentas para ligá-las e utilizá-las. O programador constrói uma aplicação de rádio ou de processamento de sinal digital qualquer criando um grafo onde os vértices são blocos de processamento de sinal e as bordas representam o fluxo de dados entre eles. Os blocos de processamento de sinal são escritos e executados em C++. Os blocos normalmente processam *streams* de dados infinitos da sua entrada para sua saída. *Stream* é um fluxo de dados. Quando um arquivo é carregado para ser editado, esta carga ocorre em um fluxo "*stream*", ou seja, linha a linha até o carregamento total do arquivo. Esses *streams* de dados são geralmente tratados como vetores de tipos básicos de C++ (inteiros, pontos flutuantes, e complexos).

Alguns blocos possuem apenas portas de entrada ou portas de saída. Normalmente blocos de saídas de dados, como uma placa de som, arquivos ou visualizações possuem apenas portas de entrada, e blocos de entrada como aquisição de dados, geradores de sinal ou leitores de arquivo têm apenas portas de saída de dados.

Mais de 100 blocos já existem na biblioteca do *GNU Radio*, como filtros de sinal, demoduladores específicos, blocos de aquisição, exibição de sinais, entradas e saídas de arquivo, entre outros. Novos blocos também podem ser adicionados à biblioteca GNU.

Os grafos são construídos e executados em *Python*.

2.3 Algoritmo Costas Loop

Costas Loop é uma especialização de circuitos PLL (*phase-locked loop*), usado para recuperar a fase e frequência da portadora de um sinal modulado. Existem várias implementações de algoritmos Costas-Loop, mas todas baseiam-se em um oscilador controlado numericamente, e filtros passa-baixa. A mais clássica implementação de Costas Loop utilizada para demodulação BPSK (*Binary Phase Shift Keying*) é mostrada na figura 5.

O sistema envolve dois laços simultâneos operando sobre o mesmo VCO (Voltage-Controlled Oscillator) ou NCO (Numerically-Controlled Oscillator). O primeiro laço está em fase com o sinal do oscilador (I), e o segundo está 90 graus defasado em relação ao oscilador (Q). Os sinais I e Q são mixados com o sinal de entrada e submetidos a um filtro passa-baixa, em seguida são multiplicados e mais uma vez filtrados para produzir um sinal de erro, que é realimentado no oscilador. Um sinal de erro negativo diminui a frequência do oscilador, enquanto um erro positivo incrementa a frequência do VCO/NCO. [4]

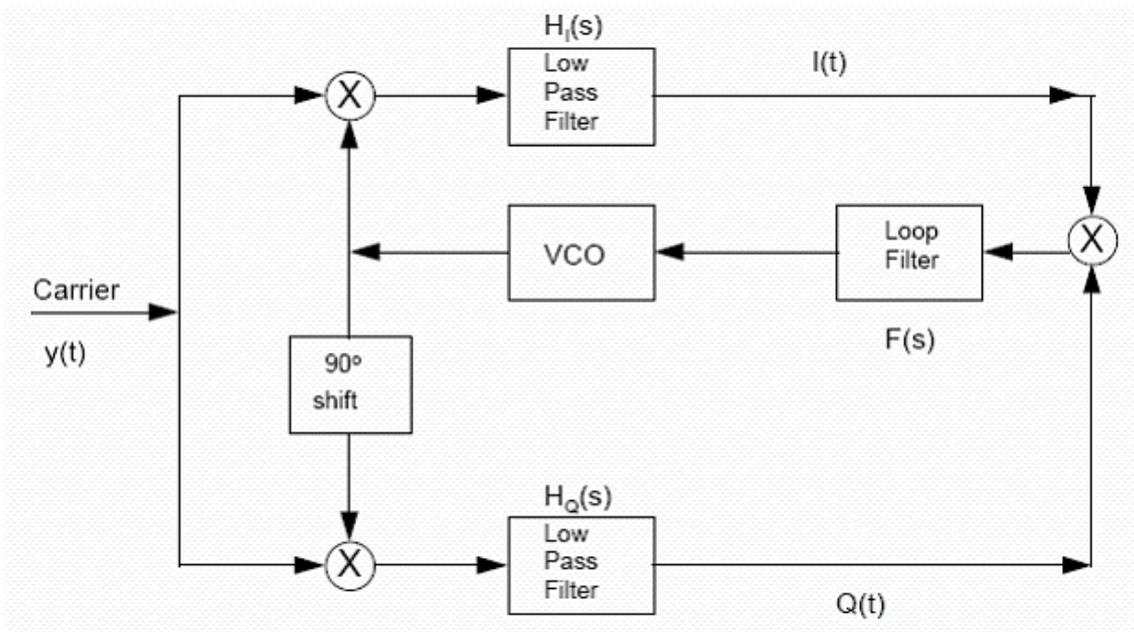


Figura 5: Diagrama de Blocos do Costas Loop [4]

Após o sincronismo do oscilador local com a portadora do sinal recebido, o sinal demodulado então pode ser percebido nas saídas dos filtros de cada laço. A modelagem matemática do Loop Costas não está no escopo desse trabalho.

3 - Desenvolvimento

3.1 – Introdução a DSP baseado em *software*

Os primeiros passos para o projeto foram o estudo de artigos sobre DSP (Processamento Digital de Sinais), envolvendo, modulação e demodulação em fase, quadratura, BPSK. O padrão utilizado no sistema ARGOS para PCDs (Plataforma de Coleta de Dados)[1], também foi estudado. Como resultado do entendimento desses assuntos foram desenvolvidos algumas rotinas em *software*:

- *Software* modulador para sinais analógicos, implementado em C++
 - Desenvolvido para simulação de modulação de fase, gera sinais aleatórios em formato de vetores, e em seguida realiza modulação de fase desses sinais, disponibilizando o resultado em arquivo para posterior teste com demoduladores.
- *Software* modulador para sinais digitais, simulando e modulando dados de uma PCD – Plataforma de Coleta de Dados, implementado em C++ e LabView
 - Desenvolvido para simular o sinal de uma PCD, de acordo com os padrões de modulação ARGOS, gera o conjunto de bits a ser enviado pela PCD, codifica, modula, e salva em arquivo para teste de *softwares* de demodulação digital, e interpretação de sinais digitais.

3.2 – Estudo de bibliotecas DSP

Em seguida iniciou-se um estudo com bibliotecas de processamento de sinal de código aberto existentes na internet, sempre em ambiente Linux, foram estudadas:

- SPUC – Bibliotecas em C++ contendo algoritmos DSP e funções de comunicação;
- *GNU Radio* – Como já abordado projeto GNU de SDR em *Hardware* genérico;
- Pspectra - Ambiente de *software* desenvolvido pelo MIT - Massachusetts Institute of Technology, contendo também bibliotecas em C++ para DSP.

O *GNU Radio*, ainda nas versões 0.x foi então tomado como foco do projeto. A partir daí toda a estrutura de classes que compõem a biblioteca GNU foi estudada e descrita.

3.3 – Configuração de placa de aquisição de dados e *GNU Radio*

Com a chegada da placa de aquisição de dados MC4020, fizemos então a integração da mesma com o *GNU Radio* utilizando o driver disponibilizado pela GNU no próprio projeto *GNU Radio*, a partir daí realizamos vários testes usando *GNU Radio* e a placa, dentre eles exibir um sinal digitalizado e sua FFT utilizando as interfaces gráficas do *GNU Radio*.

Algumas novas aplicações foram montadas incluindo blocos funcionais na biblioteca GNU, dentre elas estão, aplicações utilizando módulos do *GNU Radio* para geração de ruído branco ou gaussiano e para geração de onda quadrada.

A partir daí tínhamos dois objetivos, demodular o sinal recebido e armazená-lo em disco. Iniciamos então o estudo da classe GrCostasLoop, com a finalidade de usá-la pra demodulação PSK.

3.4 – Estudo do bloco Costas Loop do *GNU Radio*

A classe *GrCostasLoop* do *GNU Radio* utiliza 2 filtros passa baixa e um NCO como na figura abaixo.

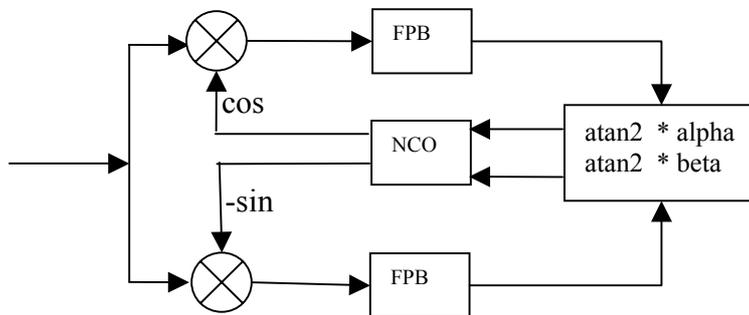


Figura 6: Diagrama de blocos da classe *GrCostasLoop* do *GNU Radio 0.x*. O trabalho do módulo é implementado na função *Work* e consiste em:

1. Combinar o sinal de entrada com \cos e $-\sin$ do NCO;
2. Filtra os sinais combinados, obtendo $q_filtered$ e $i_filtered$;
3. Aplica a função $\text{atan2}(q_filtered, i_filtered)$ que retorna o ângulo correspondente ao sinal filtrado e multiplica pela constante α , definido em 0.01 no código original e soma o resultado a fase do NCO corrigindo-a, e pela constante β , originalmente 0.01 e soma a frequência do NCO corrigindo-a.

O resultado do processamento realizado pelo módulo da classe *GrCostasLoop* seria extraído da saída de um dos filtros passa-baixa, como um sinal demodulado em fase.

A classe *GrCostasLoop* apresentou problemas de incompatibilidade com o módulo NCO (Oscilador controlado numericamente) que utilizava, atualizações de *software* realizadas no módulo NCO não foram acompanhadas pela classe.

3.5 – Testes de desempenho do sistema de armazenamento de dados

Com o objetivo de realizar testes de desempenho em aquisição de dados utilizando a placa conversora A/D MC-4020 para armazenamento em disco com diversas velocidades, para atender o segundo objetivo citado acima. Verificou-se a quantidade máxima de dados obtidos sem erro usando *GNU Radio*.

Inicialmente ativou-se a transferência de dados por UDMA 5. Utilizando-se o valor de buffer de saída como o padrão do *GNU Radio* (16 MB) efetuou-se testes verificando-se a quantidade de dados adquiridos sem erro. Em seguida alterou-se o tamanho dos dados para 32 MB. Obtemos a tabela:

Buffer 16 MB		
Taxa de amostragem (MS/s)	Velocidade de dados (MB/s)	Quantidade de dados adquiridos (GB)
7	14	7
8	16	4,15
10	20	3,9
12	24	1
15	30	0,9
Buffer 32 MB		

Taxa de amostragem (MS/s)	Velocidade de dados (MB/s)	Quantidade de dados adquiridos (GB)
7	14	6,6
8	16	7,2
10	20	4,5
12	24	1,5
15	30	1

Tabela 1: Resultados de teste de desempenho de discos

Dados:

- Tamanho da amostra: 2 bytes;
- Versão *GNU Radio*: 0.9;
- Versão driver MC4020: 0.7;
- Sistema: Conectiva Linux 9.0;
- *Hardware*: Athlon XP 2400, 1 GB Ram DDR 333 MHz, HD Seagate ATA 100 40 GB 7200 rpm

Neste teste quando ocorria um erro, alguma perda de dados, um caractere era impresso na tela pelo *GNU Radio*, então a aquisição era interrompida e verificava-se o quanto de dados tínhamos conseguido adquirir. Notamos que a uma taxa de aquisição de 15 MS/s o sistema já era bastante deficiente e erros ocorriam rapidamente, assim concluímos o teste mesmo sem a utilização da capacidade máxima de aquisição da placa AD.

Em testes de aquisição de dados que não envolviam transferência para Disco, como FFT e SimpleScope de sinal da placa MC4020 o sistema respondeu sem problemas mesmo quando a velocidade máxima da placa foi utilizada. Testes a 20MS/s foram realizados e nenhum erro reportado. Com isso foi verificada a deficiência do sistema no que se refere ao armazenamento de dados em disco, deixando claro a necessidade de tecnologias como RAID (Redundant Array of Independent Disks) e SATA (Serial ATA).

3.6 – Utilização de Serial ATA e RAID

A partir desse ponto, partimos para a especificação de uma nova arquitetura, baseada em RAID e SATA, que foi adquirida pelo INPE. Ao mesmo tempo, o *GNU Radio* foi atualizado para versões 2.x, tendo grandes modificações em sua estrutura, sendo agora dividido em blocos para download e utilizando *Python* para a montagem de suas aplicações.

Migrando então nosso Sistema Operacional para o Conectiva Linux 10, por ter Kernel 2.6, que tinha suporte nativo ao Serial ATA, ainda não suportado por sistemas com núcleo 2.4, tivemos então vários problemas, primeiro relacionados ao driver da placa MC4020, disponibilizado para kernel 2.6 somente algum tempo depois ainda em versões experimentais, pelo projeto GNU. Durante esse tempo, não conseguimos fazer funcionar de maneira satisfatória a placa AD com o sistema de armazenamento SATA.

3.7 – Instalação *GNU Radio*

Vencidos os problemas com drivers tanto da placa AD quanto da controladora SATA partimos então para instalação do *GNU Radio*, agora na versão 2.4, tivemos vários problemas pois a biblioteca da GNU tinha requisitos de bibliotecas mais novas

que as instaladas com a distribuição da Conectiva, e instalações das versões novas acabavam gerando incompatibilidades com versões antigas.

Então partimos para a instalação do Slackware 10.1, recém lançada, que atendia a maior parte dos requisitos do *GNU Radio*. Os demais requisitos puderam ser instalados sem gerar conflitos. O Slackware dá suporte a controladora SATA tanto para utilização com kernel 2.4 como 2.6, eliminando também vários de nossos problemas.

4 - Resultados Obtidos

Especificação, instalação e configuração de sistema Linux funcionando sobre arquitetura RAID nível 0:

- Athlon XP 2800+
- 1 GB Memória RAM
- 2 Discos Serial ATA 80 GB cada em configuração RAID 0 controladora VIA 8237 [6]
- Slackware 10.1 (kernel 2.4)

Instalação do *GNU Radio* atualmente na versão 2.5, com os módulos core, audio-oss, wxgui, examples, qr-mc4020 e mc4020.

Com o *GNU Radio* e o driver da placa MC4020 instalados, montamos uma aplicação usando os blocos do *GNU Radio* para realizar aquisição de dados através da placa ADC e armazenar diretamente em disco. Com o sistema RAID 0, conseguimos realizar a aquisição na máxima capacidade da placa ADC, 20 MS/s, enquanto havia espaço em disco sem nenhum erro ou perda de dados, o que atestou a eficiência do sistema de armazenamento de dados em disco especificado.

Algumas outras aplicações foram montadas com os módulos do *GNU Radio*, dentre elas, exibição de um sinal adquirido da placa MC4020 (figura 7) e sua FFT (figura 8).

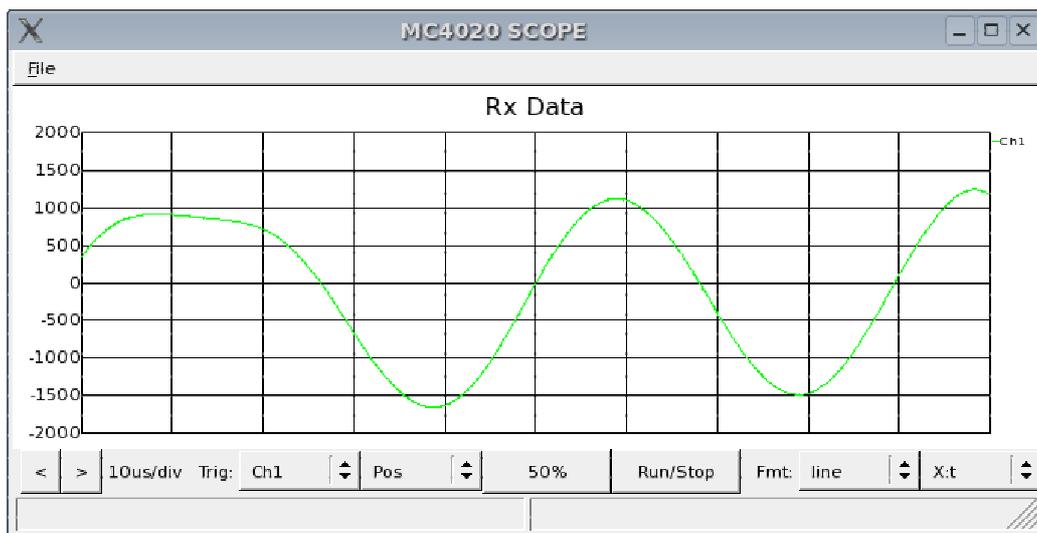


Figura 7: Um sinal modulado em fase captado pela placa MC4020 usando *GNU Radio*

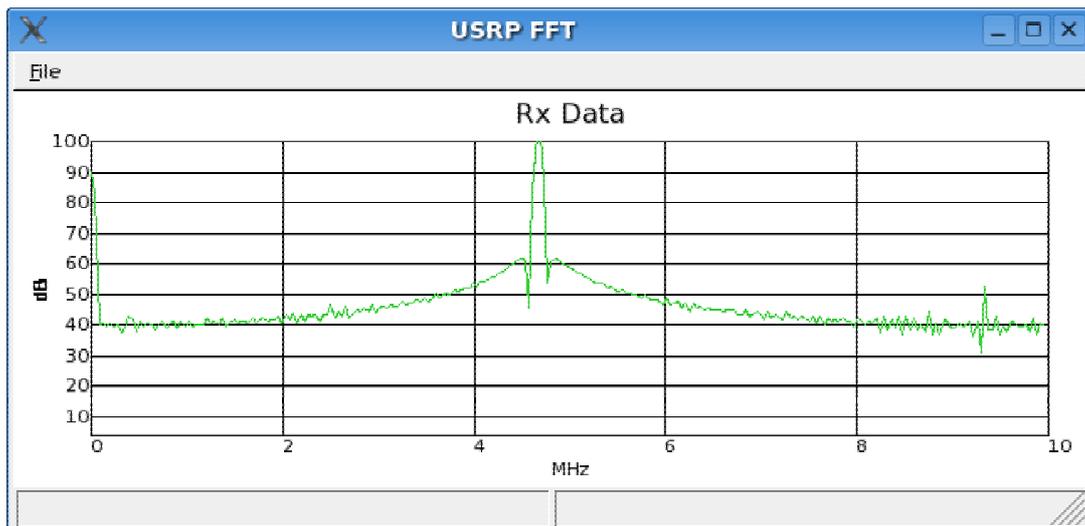


Figura 8: FFT de um sinal modulado em fase captado pela placa MC4020 usando *GNU Radio*

Foi incluído às bibliotecas do *GNU Radio*, um algoritmo desenvolvido por pesquisadores do MIT - Massachusetts Institute of Technology, disponibilizado em <http://web.mit.edu/imirkin/www/gnuradio/bpsk.html> para realização de demodulação BPSK através de Costas Loop. Após recompilação do SDR GNU com o módulo BPSK incluso, montamos então uma aplicação para utilizá-lo sobre um sinal adquirido da placa MC4020 e mostrar as suas duas saídas em gráficos de osciloscópio. Utilizamos como fonte de sinal, um gerador de funções programado para gerar um sinal com modulação BPSK, o mesmo sinal mostrado nas figuras 7 e 8, o resultado do processamento é mostrado na figura 9 a seguir.

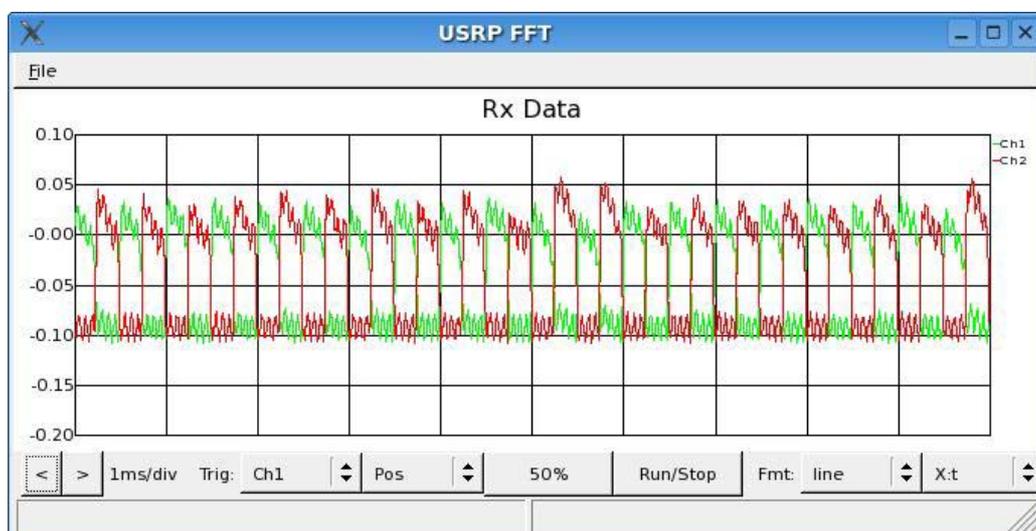


Figura 9: Saída do algoritmo Costas Loop incorporado ao *GNU Radio* sobre um sinal BPSK

Porém o algoritmo incorporado teve baixo desempenho de processamento em tempo real, reportando perdas de dados a frequências de amostragem maiores que 1 MS/s. Chamadas a funções trigonométricas são usadas no algoritmo, essas funções, padrões de C++ ou da biblioteca GNU, utilizam séries de Taylor, o que degrada o desempenho com excesso de cálculos. Atualmente estamos substituindo essas funções

por consultas a tabelas pré-calculadas, além de algumas otimizações no código do algoritmo para melhorar o seu desempenho.

4.1 - Próximos Passos

Atualmente temos o sistema funcionando com *GNU Radio* e o módulo *Costas Loop* incorporado. Temos também o sistema de aquisição e armazenamento e recuperação de dados em disco configurados e integrados ao *GNU Radio*.

Os próximos passos do projeto estarão centrados em aperfeiçoar o algoritmo de *Costas Loop*, de forma a possibilitar seu uso em tempo real, de acordo com os requisitos descritos na seção 1.2. A partir daí teremos o primeiro bloco do *Software* demodulador, o *Software* de Tempo Real funcional, e partiremos para o *Software* de tratamento posterior, que deverá realizar a busca, detecção e processamento de sinais das PCDs no sinal recuperado pelo *Software* de Tempo Real. Em seguida, deveremos trabalhar na interface com o usuário do sistema de demodulação e tratamento, e na integração com demais partes do sistema de recepção, controle de posicionamento da antena, e bancos de dados.

5 – Conclusão

O aproveitamento da estrutura já existente da antiga estação SACI para implantação de uma Estação Multi-Missão utilizando SDR é um projeto inovador, de baixo custo e muito importante para a tecnologia nacional. Surgindo como um sistema desenvolvido nacionalmente, aberto a alterações e adaptações, um sistema de recepção de satélite totalmente baseado em *software* livre poderia aparecer como uma alternativa mais barata, eficaz e principalmente flexível às tecnologias atualmente empregadas, geralmente proprietárias e conseqüentemente fechadas e de difícil adaptação. O projeto do receptor digital ainda está em andamento, mas alguns caminhos já estão sendo traçados, como o uso do projeto da GNU, *GNU Radio* como base no desenvolvimento do sistema de tratamento de sinal, suas ferramentas, bibliotecas e estrutura facilitam a criação de tal sistema.