



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

DESENVOLVIMENTO DE ALGORITMOS PARA DECODIFICAÇÃO DO SINAL DO  
SISTEMA BRASILEIRO DE COLETA DE DADOS

Eric Matheus Soares Macedo (Universidade, Bolsista PIBIC/CNPq)  
E-mail: [Eric.macedo@crn.inpe.br](mailto:Eric.macedo@crn.inpe.br)

José Marcelo Lima Duarte (INPE-CRN, Orientador)  
E-mail: [jmarcelo@crn.inpe.br](mailto:jmarcelo@crn.inpe.br)

Natal, Rio Grande do Norte  
Julho de 2015



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

DESENVOLVIMENTO DE ALGORITMOS PARA DECODIFICAÇÃO DO SINAL DO  
SISTEMA BRASILEIRO DE COLETA DE DADOS

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

Eric Matheus Soares Macedo (Universidade, Bolsista PIBIC/CNPq)  
E-mail: Eric.macedo@crn.inpe.br

José Marcelo Lima Duarte (INPE-CRN, Orientador)  
E-mail: jmarcelo@crn.inpe.br

Natal, Rio Grande do Norte  
Julho de 2015

## Resumo

Esse documento contém um relato do trabalho e estudos desenvolvidos durante o período da bolsa de iniciação científica no INPE-CRN (Centro Regional do Nordeste). O conteúdo tem como principal objetivo facilitar a continuação do trabalho por bolsistas futuros, através do detalhamento dos códigos desenvolvidos em Matlab. Outro objetivo consiste em documentar o sistema decodificador desenvolvido para o sistema brasileiro de coleta de dados ambientais.

# 1. Sumário

1. Introdução .....	5
2. Formato da mensagem .....	6
3. SbcdDecoder .....	7
3.1. Função decode.....	7
3.2. Função pcdDetection.....	7
3.3. Função pcdDecoder.....	8
4. DFT Leakage e Zero-Padding .....	8
4.1. Amenizando os efeitos do Leakage.....	11
4.2. DetectionSetup e a escolha do Limiar de Detecção .....	13
5. Conclusão.....	16
6. Bibliografia .....	17

## 2. Introdução

No ano de 1993, foi iniciada a operação do Sistema Brasileiro de Coleta de Dados Ambientais. Essa operação foi iniciada devido à grande necessidade de um sistema automático de aquisição de dados ambientais e do desenvolvimento de tecnologia espacial Nacional.

O sistema é atualmente composto por mais de 600 PCDs (Plataforma de coleta de dados) e MTRs (mini transmissores remotos), que coletam dados ambientais como mudanças climáticas, níveis de rios, localização de animais, pessoas ou embarcações, transmitindo os dados em UHF para os satélites do sistema.

O sistema satélite atual, funcionando apenas como espelho, recebe e retransmite a mensagem para as estações terrenas de recepção (ETRs) de sinais de satélite, atualmente localizada em Cuiabá e Alcântara. O sinal é então demodulado e a informação é extraída e enviada para o Centro de Missão, localizado em Cachoeira Paulista. O Centro de missão armazena os dados e distribui as informações aos usuários finais. Apesar de o sistema atual ser mais simples, a simples retransmissão requer uma visada direta entre a PCD, o satélite e o centro de recepção. Além disso, requer que o satélite transmita sinais constantemente para a estação base, já que não há armazenamento de dados no satélite e não se sabe quando existem dados sendo transmitidos. Isso acarreta num desperdício de energia quando não existem PCDs transmitindo informação, já que o transmissor do satélite permanece ligado. Além dos gastos com energia, o sinal é transmitido para a ETR com ruído acumulado no uplink.

O presente trabalho consiste em desenvolver um modelo de decodificador para o Sistema Brasileiro de Coleta de Dados Ambientais (SBCD) em uma linguagem de alto nível (MatLab). O objetivo consiste em utilizar o próprio satélite para decodificar o sinal no espaço, armazenar os dados decodificados e transmiti-los para a estação base apenas quando desejado. Essa funcionalidade eliminaria o acúmulo do ruído no uplink, restando apenas o ruído acrescentado no downlink, além de eliminar os gastos de transmissão quando não há informações sendo transmitidas pelas PCDs, diminuindo o gasto de energia do mesmo.

O modelo em Matlab proposto é composto de vários blocos diferentes: O bloco detecção, sincronismo de frequência, e o bloco da decodificação. Nas seções a seguir, será apresentado cada um desses blocos.

### 3. Formato da mensagem

Antes de detalhar cada um dos blocos do sistema é importante entender mais sobre o formato das mensagens transmitidas por cada PCD/MTR.

As mensagens transmitidas possuem de 360 a 920 ms de duração. Os dados são codificados em bi-phase\_L (também conhecida como codificação Manchester) a 400 bps e modulados em fases de -60 a +60 graus. Nessa modulação são utilizados dois bits por símbolo, sendo a transição +60 a -60 graus utilizadas para representar o bit um, e -60 a +60 graus representando os bits zero. Por meio dessa representação obtemos a modulação da figura abaixo:

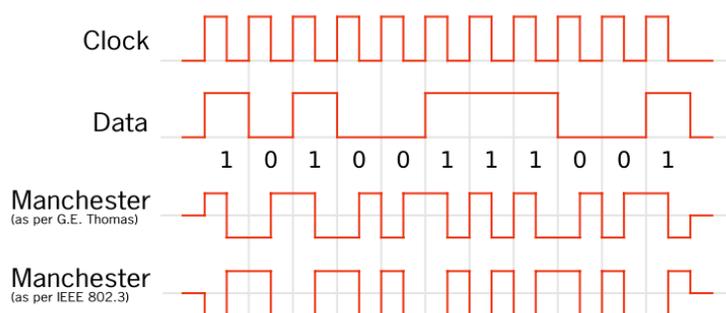


Figura 1: Modulação bi-phase\_L ou Manchester.

A mensagem é estruturada como mostra a Figura 2. Toda mensagem possui um período de Portadora pura de 160 ms, em que apenas a portadora é transmitida. Logo após o período de portadora, seguem 15 bits um para sincronização de Bits, e 8 bits para a sincronização de palavra. Após o período de sincronização, um bit um indica o início da mensagem, e precede quatro bits que representam o tamanho da mensagem.

Portadora Pura:	Portadora Modulada:						
160 [ms]	A	B	C	D	E	F	G
• Portadora Pura (Inicio):	160 ± 2.5 [ms]						
• Estrutura da Mensagem							
A – Sincronização de Bit:	15 bits “1” = 37,5[ms]						
B – Sincronização de Palavra:	8 bits ( “00010111”)						
C – Inicialização da mensagem:	1 bit “1”						
D – N°de Blocos de 32 bits (Nb):	4 bits ( Nb = 1,2,...8)						
E – Identificação da PCD:	20 bits						
F – Dados dos Sensores:	32*Nb						
G – CRC (opcional) (Fim):	8 bits						

Figura 2: Estrutura da Mensagem.

## 4. SbcdDecoder

Para favorecer a integração dos blocos do sistema citados na introdução, foi priorizado o uso de programação a objetos. A classe SbcdDecoder, a principal na integração dos sistemas, é composta por 3 métodos diferentes: a função **decoderAssigner**, a função **decode** e a função de inicialização. A função de inicialização é executada uma única vez, quando o objeto correspondente a classe SbcdDecoder é criado. Essa função tem como principal finalidade inicializar uma única vez e objetos de filtros e outras variáveis que serão utilizadas durante o processo de decodificação, evitando que essas variáveis sejam redeclaradas a cada iteração. Uma dessas variáveis consiste no vetor de 10 estruturas decoderData, que tem como função:

- Armazenar a memória dos decodificadores de PCDs.
- Integrar, num único ambiente, as variáveis de detecção, sincronismo de frequência e decodificação.

Ao ser criado, o objeto executa uma única vez uma função responsável por inicializar os objetos dos filtros, algumas variáveis e a memória dos 10 decodificadores. A memória inicializada é responsável por armazenar informações das PCDs detectadas, bem como o estado anterior das funções de sincronismo de frequência e decodificação, para que seja possível, na próxima iteração, recomençar os cálculos de onde cada função parou. Para fins de organização, temos como objetivos futuros tornar cada uma das funções as mais independentes possíveis, para tornar mais fácil a identificação de erros e a adaptação do código para outros sistemas, como o Argos III, por exemplo.

### 4.1. Função decode

A função decode funciona como uma Main, integrando todas as funções desenvolvidas no projeto. Esse método integra as funções responsáveis pela detecção de PCDs, sincronismo de frequência, de tempo e símbolo e decodificação das PCDs.

### 4.2. Função pcdDetection

A função pcdDetection utiliza a função DetectionSetup para obter a frequência das PCDs detectadas na amostra de sinal recebida. Após receber as frequências, a função verifica dentre os 10 decodificadores existentes, se há conflito entre as PCDs sendo decodificadas e a PCD detectada. O conflito é confirmado caso a nova PCD detectada esteja centralizada a menos de 1600 Hz de distância das PCDs já sendo decodificadas.

Essa distância é calculada a partir das características do sinal transmitido. Já que o sinal é transmitido a uma taxa de 400 bits/s, após aplicar a modulação Manchester, possuímos dois bits para cada bit modulado. Dessa forma, para que a modulação acompanhe a taxa de transmissão de bits é necessário que a taxa de transmissão após a modulação seja de 800 bits/s, o que acarreta numa banda de 1600 Hz. Normalmente, PCDs detectadas em zonas de conflito, são as PCDs previamente detectadas e sendo decodificadas.

Dessa forma, é necessário que o limiar detecte apenas a parte do sinal dentro da banda de 1600, senão os lóbulos laterais do sinal serão interpretados falsamente como uma nova PCD no espectro.

A função recebe como parâmetros de entrada 1280 amostras do sinal, e um objeto SbcdDecoder, contendo informações do sinal e dos 10 decodificadores existentes, retornando apenas o objeto SbcdDecoder com informações atualizadas das novas pcds detectadas nas amostras do sinal de entrada.

### 4.3. Função pcdDecoder

A função pcdDecoder recebe, como argumentos de entrada, um objeto SbcdDecoder e um bloco com 1280 amostras de um sinal coletado. A função é responsável pelo gerenciamento e processamento de cada um dos decodificadores. Os dez decodificadores são primeiramente analisados, verificando quais carregam consigo informações de PCDs a serem decodificadas. Os decodificadores contendo informações, são processados enquanto que os outros são ignorados.

O processamento ocorre da seguinte forma: Os dados da PCD detectada contidos no decodificador correspondente são enviados para o sistema de sincronismo de frequência.

## 5. DFT Leakage e Zero-Padding

A transformada discreta de Fourier opera sobre um número finito de N amostras de um sinal qualquer, amostrado a uma taxa de amostragem  $f_s$ . A transformada de Fourier, ao realizar a transformação em N amostras, retorna N amostras transformadas. Cada um desses N pontos transformados está diretamente associado a uma frequência. O vetor de N pontos, portanto, está associado a um vetor de frequências também com N pontos, dados por:

$$f_m(m) = \frac{mf_s}{N}$$

Em que  $m = 0, 1, 2, \dots, N-1$ .

Ao calcularmos a transformada discreta de Fourier de um sinal que contem a sua energia localizada precisamente nos pontos dados pela equação acima, obtemos o resultado aparentemente correto da transformada de Fourier de um seno, como saída. Se o sinal contiver, entretanto, energia entre os pontos dados pela equação acima (em  $\frac{1.5f_s}{N}$ , por exemplo), os resultados obtidos ao se calcular a transformada de Fourier apresentarão a energia do sinal “espalhada” por todos os pontos da frequência da DFT, ou seja, ocorre o Leakage.

Para demonstrar esse efeito, criou-se uma senoide com frequência de 294 Hz. A senoide foi amostrada a uma taxa de 44,1 kHz e foram coletadas 1500 amostras do sinal, o que resulta num vetor de frequências igual a:

$$f_m(m) = \frac{m \times 44100}{1500}$$

Em que  $m = 0, 1, 2, \dots, 1499$ .

Como a energia do sinal está contida em uma frequência múltipla de  $44100/1500$ , ( $294 = \frac{44100}{1500} \times 10$ ) o sinal será corretamente representado depois de realizada a sua DFT.

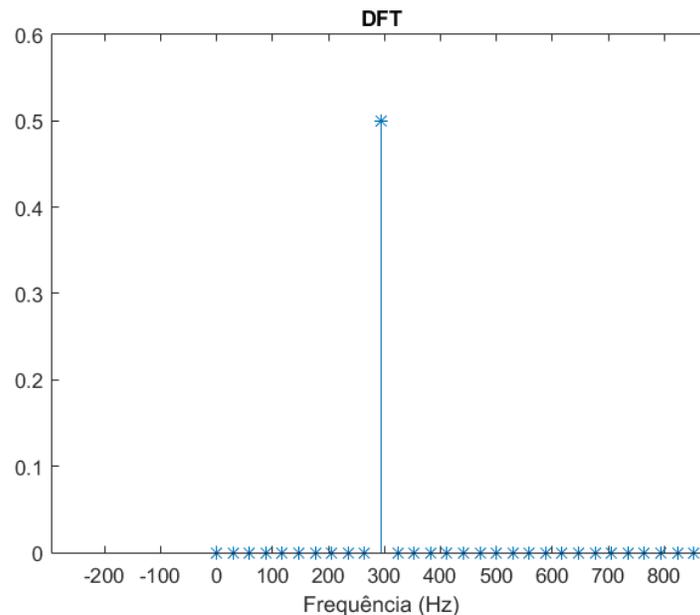


Figura 3: DFT de uma senoide.

Se utilizarmos um sinal cuja energia não está localizada dentro de um bin da DFT (dentro de um dos pontos do vetor de frequência calculado) como 307 Hz, por exemplo, o resultado obtido é o apresentado na figura abaixo:

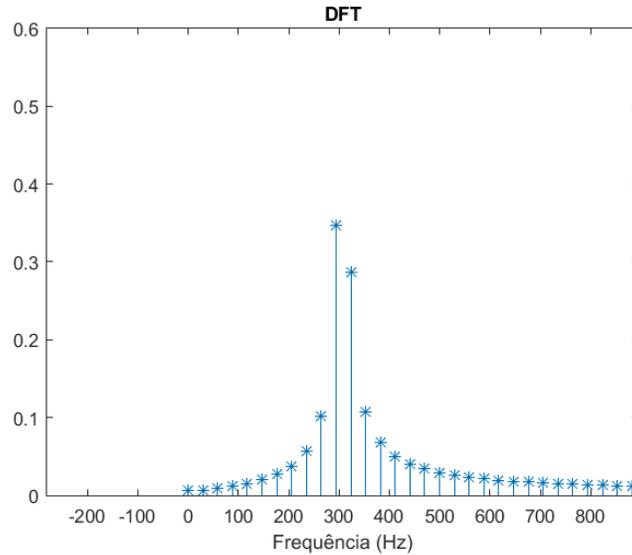


Figura 4: DFT de uma senoide com leakage.

Esse espalhamento ocorre pelo fato não estarmos usando uma senoide “real”, mas sim parte de uma senoide. A Transformada de Fourier de uma senoide não é igual a transformada de Fourier de uma senoide limitada no tempo. No nosso caso, por exemplo, a transformada de Fourier no tempo discreto (DTFT) de uma senoide contendo  $k$  ciclos, é dada aproximadamente por:

$$X(m) = \frac{A_o N}{2} \times \frac{\sin[\pi(k - m)]}{\pi(k - m)}$$

Em que  $k$  não necessariamente é um número inteiro. Essa equação é válida para os dois casos apresentados acima. Para obter uma melhor representação da DTFT, podemos aumentar o número de pontos da DFT pelo simples acréscimo de zeros ao sinal (Zero-Padding). Essa técnica nos permite obter uma melhor apresentação da DTFT de uma senoide. Aumentando o número de pontos da DFT por 10 através do Zero-Padding, obtemos o seguinte resultado para os dois casos anteriores.

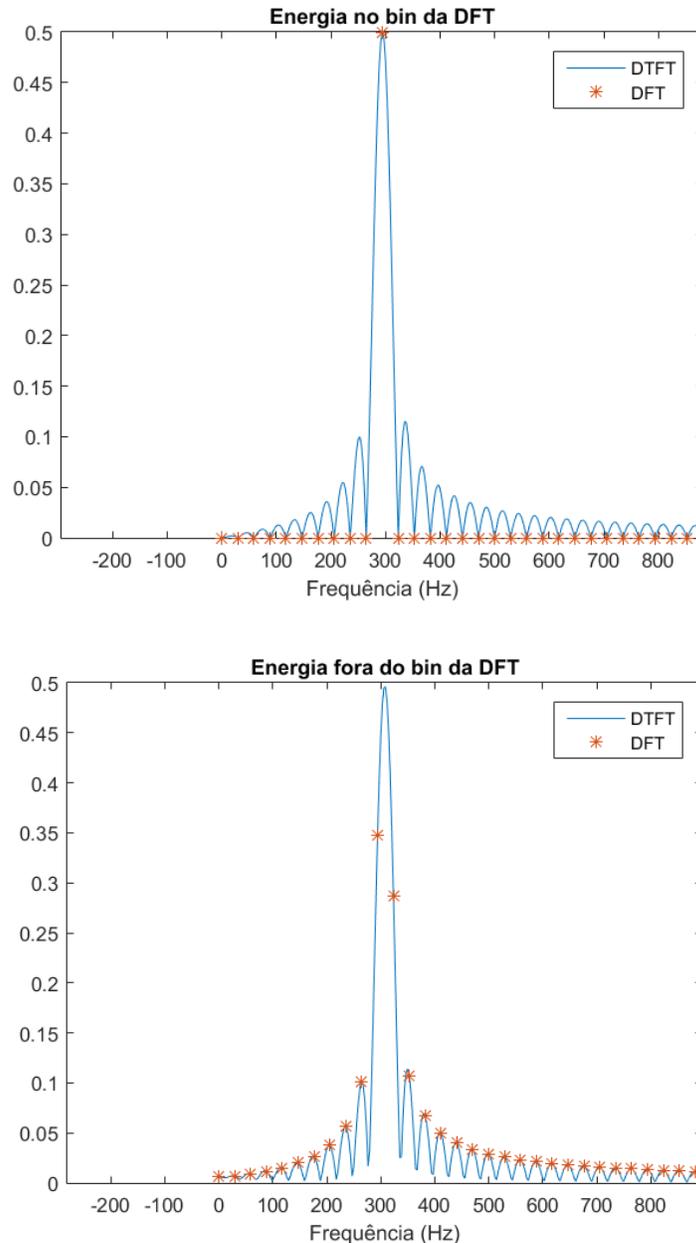


Figura 5: DTFT e DFT de uma senoide sem e com Leakage, respectivamente.

Esses resultados nos permitem ter uma ideia da verdadeira representação da DTFT de parte de uma senoide. É possível perceber também que o Zero-Padding contribui para que possamos obter a potência do sinal mais precisamente através de sua DFT.

### 5.1. Amenizando os efeitos do Leakage

Os efeitos do leakage podem ser amenizados diminuindo a amplitude dos lóbulos laterais da função sinc apresentada anteriormente. Isso é feito por meio de janelas de

suavização que fazem o sinal variar suavemente até seu valor de amplitude, ao invés de sofrer uma variação brusca, como ocorre quando usamos uma janela retangular. É importante ressaltar que quando extraímos parte de uma senoide, estamos na verdade multiplicando uma senoide por uma janela retangular de amplitude 1. Abaixo é possível ver uma janela Prolata de primeira ordem criada para fins de suavização.

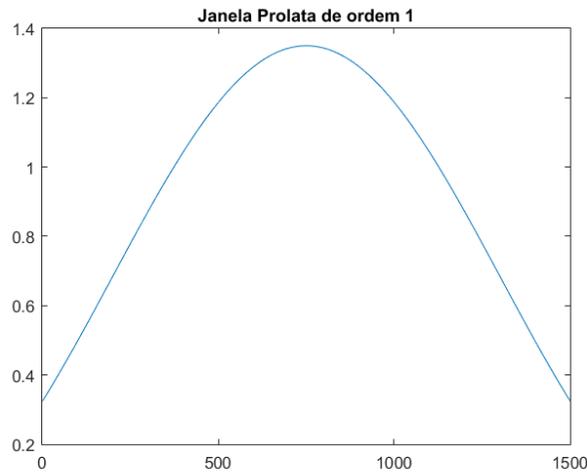


Figura 6: Janela Prolata de primeira ordem.

Aplicando essa janela ao sinal do exemplo anterior, e plotando sua transformada de Fourier discreta, é possível perceber facilmente a atenuação dos lóbulos laterais na figura 7, abaixo.

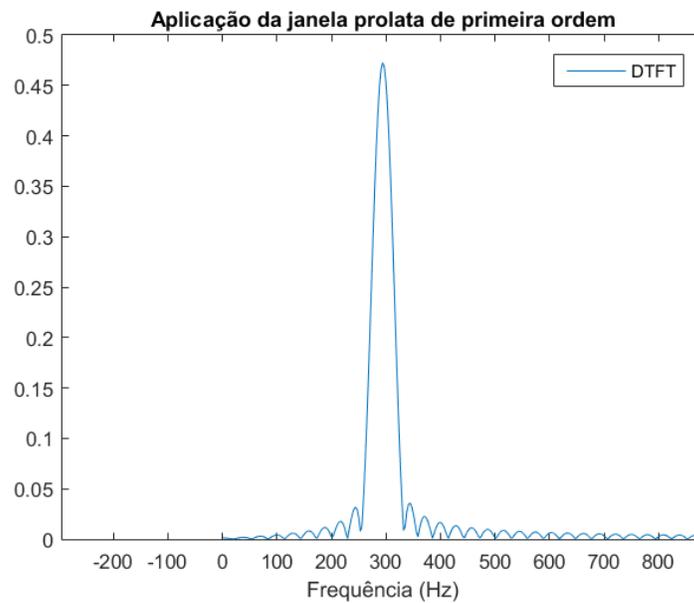


Figura 7: Resultado após suavização.

## 5.2. DetectionSetup e a escolha do Limiar de Detecção

A seção anterior sobre leakage e zero-padding foi introduzida com o objetivo de complementar essa seção. As 1280 amostras recebidas pela função DetectionSetup são suavizadas por meio de uma janela prolata de ordem 2, e o zero-padding é utilizado acrescentando 768 zeros ao sinal. A DFT utilizada, portanto, é de 2048 pontos. Depois de ser calculada a transformada discreta de Fourier do sinal, a transformada é enviada para a função peakDetection, responsável por detectar as PCD's presentes no espectro de frequência. São identificados como sinais de PCD os pontos que apresentarem um valor maior que certo limiar.

O limiar foi escolhido de modo que fosse possível detectar as PCD's de alta e de baixa potência, sem permitir que os lóbulos laterais fora da banda de 1,6 kHz das PCD's de alta potência atravessassem o limiar. Abaixo é possível ver uma figura na qual se compara a potência do lóbulo fora da banda de 1,6 kHz com a portadora de uma PCD mais fraca.

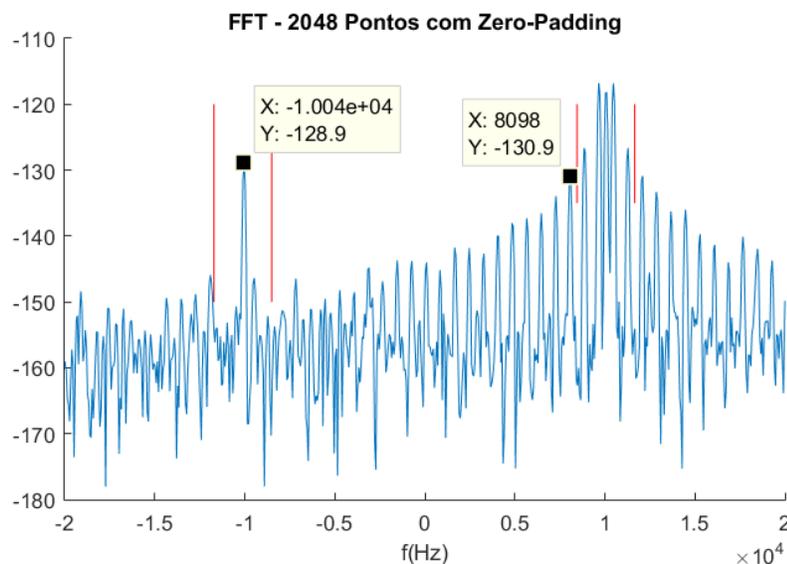


Figura 8: Comparação entre potência do lóbulo e PCD mais fraca.

Realizando a plotagem de todas as janelas do sinal no domínio da frequência, é possível perceber na figura abaixo que o limiar escolhido permite a detecção da PCDs mais fracas sem que existam falsas detecções.

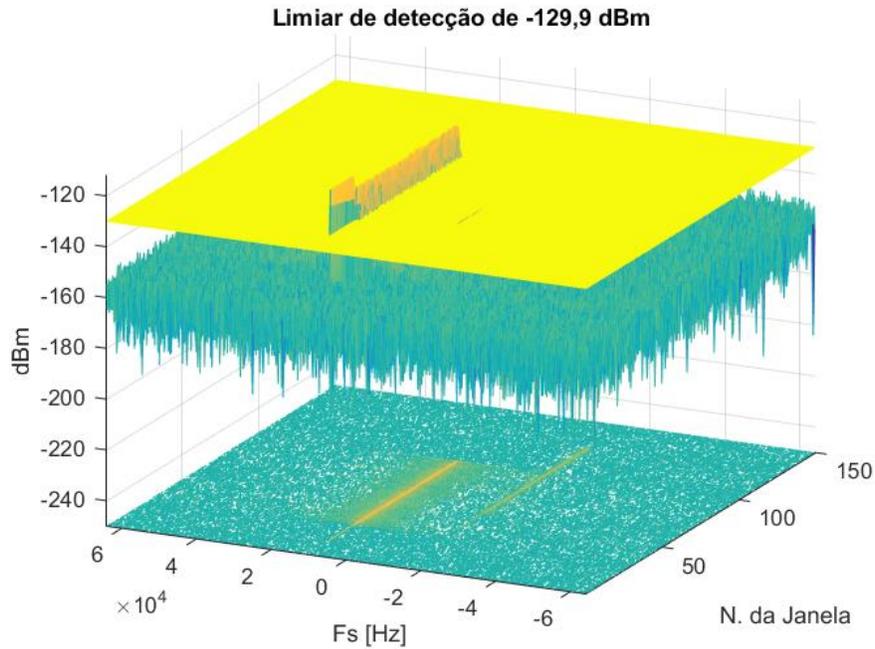


Figura 9: Plot 3D do limiar de detecção.

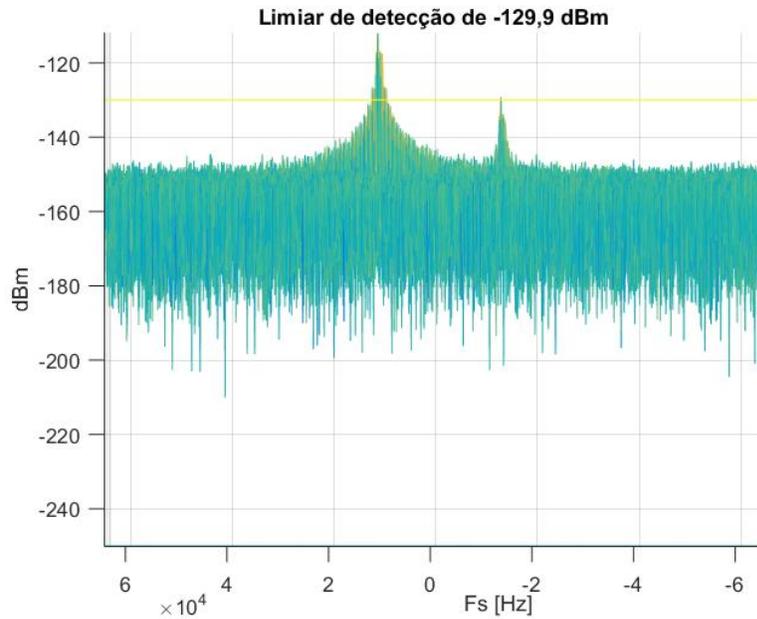


Figura 10: Plot 3D do limiar de detecção (outro ângulo).

Nos plots apresentados, é possível identificar claramente o sinal da PCD's de maior e de menor potência. Plotando uma vista superior dos pontos que atravessam o limiar, obtemos a figura abaixo:

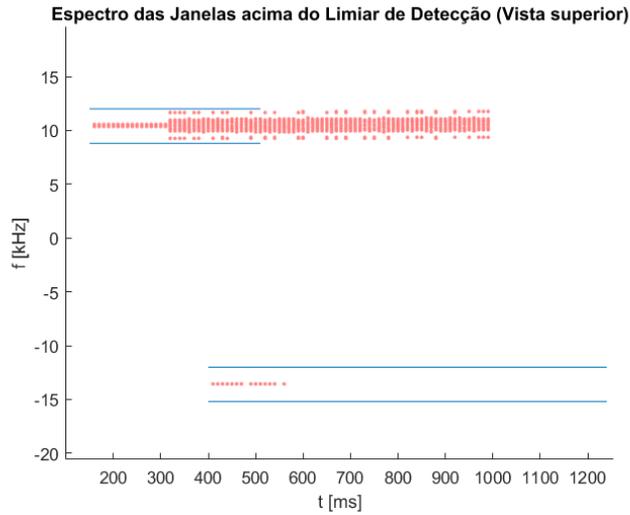


Figura 11: Vista superior da figura 6.

As linhas azuis foram acrescentadas como referência, para identificar se os pontos acima do limiar estão dentro da banda de 1600 kHz. Se utilizarmos um limiar muito baixo, os lóbulos fora da banda atravessarão o limiar e serão falsamente identificados como PCD's, como mostra a figura abaixo:

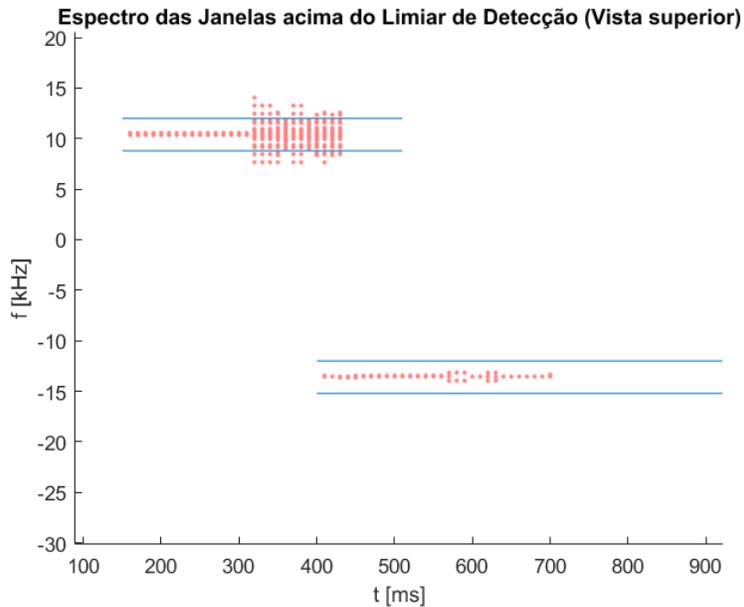


Figura 12: Vista superior da figura 6 com limiar muito baixo.

## **6. Conclusão**

O trabalho desenvolvido até o presente momento mostra que é possível a decodificação do sinal das PCDs no próprio satélite. Os sistemas citados já foram finalizados, restando apenas finalizar a integração entre os sistemas e a elaboração de um plano de testes, para que seja possível avaliar a eficácia do decodificador proposto. O sistema de detecção já foi devidamente integrado com o sistema do sincronismo de frequência, restando apenas à integração com o decodificador.

Para facilitar a integração, foi proposta a elaboração de cada um desses sistemas em programação orientada a objeto, o que também facilitará a adaptação do sistema para o modelo ARGOS III, um dos nossos objetivos nos trabalhos futuros.

## 7. Bibliografia

Eddins, S. (23 de Setembro de 2014). *Sinusoids and FFT frequency bins*. Acesso em 1 de Março de 2015, disponível em Matlab Central: <http://blogs.mathworks.com/steve/2014/09/23/sinusoids-and-fft-frequency-bins/>

Lyons, R. G. (2011). *Understanding Digital Signal Processing* (3ª Edição ed.). Prentice Hall.

Rae, J. C. (Junho de 2005). *Detector de sinais para os satélites do Sistema Brasileiro de Coleta de Dados usando análise espectral digital*. Brasil.