



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
PRÓ-REITORIA DE PESQUISA E EXTENSÃO
COORDENAÇÃO DE PESQUISA**

RELATÓRIO DE ATIVIDADES DO ALUNO

Programa: PIBIC

Título do Projeto: Primeiras observações com o Radiotelescópio Uirapuru.

Aluno(a): Erik Henrique Limeira Guimaraes.

Orientador(a): Luciano Barosi de Lemos.

Campina Grande, 5 de março de 2024

1 Introdução:

A Universidade Federal de Campina Grande está em processo de comissionamento do Radio-telescópio UIRAPURU, que tem como objetivo o estudo de pulsares conhecidas e sinais transientes de alta intensidade, esse radiotelescópio funcionara de maneira isolada mas conjunta com o radiotelescópio BINGO o auxiliando a determinar a posição de fast radio burst (FRB), que deve ser detectada em grande quantidade pelo radiotelescópio BINGO e contribuirá para a caracterização e o estudo desse tipo de objeto e sua distribuição.

O projeto de pesquisa se insere no âmbito do projeto do radiotelescópio BINGO, e contempla também atividades de divulgação científica, portanto, com atividades também junto ao projeto de extensão radiotelescópio Bingo - Lutando pela ciência no sertão paraibano, sem prejuízo para o cronograma do projeto de IC. Nesse projeto de extensão buscamos levar o conhecimento científico para escolas e eventos, com apresentações informativas e/ou educativas além de demonstrações de experimentos como as minicornetas que desenvolvemos e confeccionamos no LABMET usando materiais de baixo custo, com elas podemos fazer medições reais e em tempo real de sinais gerados por nós ou não, levando assim uma abordagem mais visual e palpável do que seria o estudo de sinais em radiofrequência. As minicornetas estão passando também por um processo de calibração e posteriormente serão usadas para contribuições científicas.

Este radiotelescópio (UIRAPURU) encontra-se em fase de comissionamento e calibração, etapa fundamental para o seu funcionamento científico posterior. Nesta fase existem várias oportunidades de aprendizado, com impacto e relevância científica mas de simplicidade suficiente para serem abordados em diversos níveis de conhecimento.

O projeto visa realizar essa fase de comissionamento do UIRAPURU, realizando suas primeiras medições e observações, coletando esses dados, processando e analisando-os com ferramentas computacionais, assim buscando caracterizar as RFI e seus efeitos no equipamento. Estas interferências podem ter naturezas diversas, desde interferências em uma banda inteira de frequência, usualmente ocasionadas por radiofrequência produzida em atividades humanas, de caráter contínuo e de longa duração como interferências impulsivas, bem delimitadas tanto na frequência como no tempo, que podem ser causadas por efeitos terrestres, relacionados ou não com atividade humana, bem como com efeitos espaciais, ocasionados pela presença de satélites ou aviões no feixe.

Cada tipo de interferência deve ser tratada de forma adequada às suas características, e para isso é necessário e vital que sua natureza seja bem conhecida. Este é o principal objetivo deste projeto iniciar a caracterização destes sinais de natureza local e avaliar a performance de algumas técnicas de mitigação disponíveis na literatura.

Para tanto é necessário se deparar com conceitos básicos de eletromagnetismo, fenômenos ondulatórios, telecomunicações, transformadas de Fourier (FFT) e processamento digital de sinais, técnicas computacionais de processamento de sinais e de visualização de dados e elementos básicos de radioastronomia.

2 Objetivos:

2.1 Gerais:

1. Contribuir para a completa caracterização científica do radiotelescópio Uirapuru.
2. Compreender a influência de Interferência por Radiofrequência no ambiente do radiotelescópio Uirapuru.
3. Compreender o planejamento de observações radioastronômicas.
4. Contribuir para o desenvolvimento de técnicas de análise e mitigação de RFI impulsiva para o sistema de outriggers do Radiotelescópio BINGO.

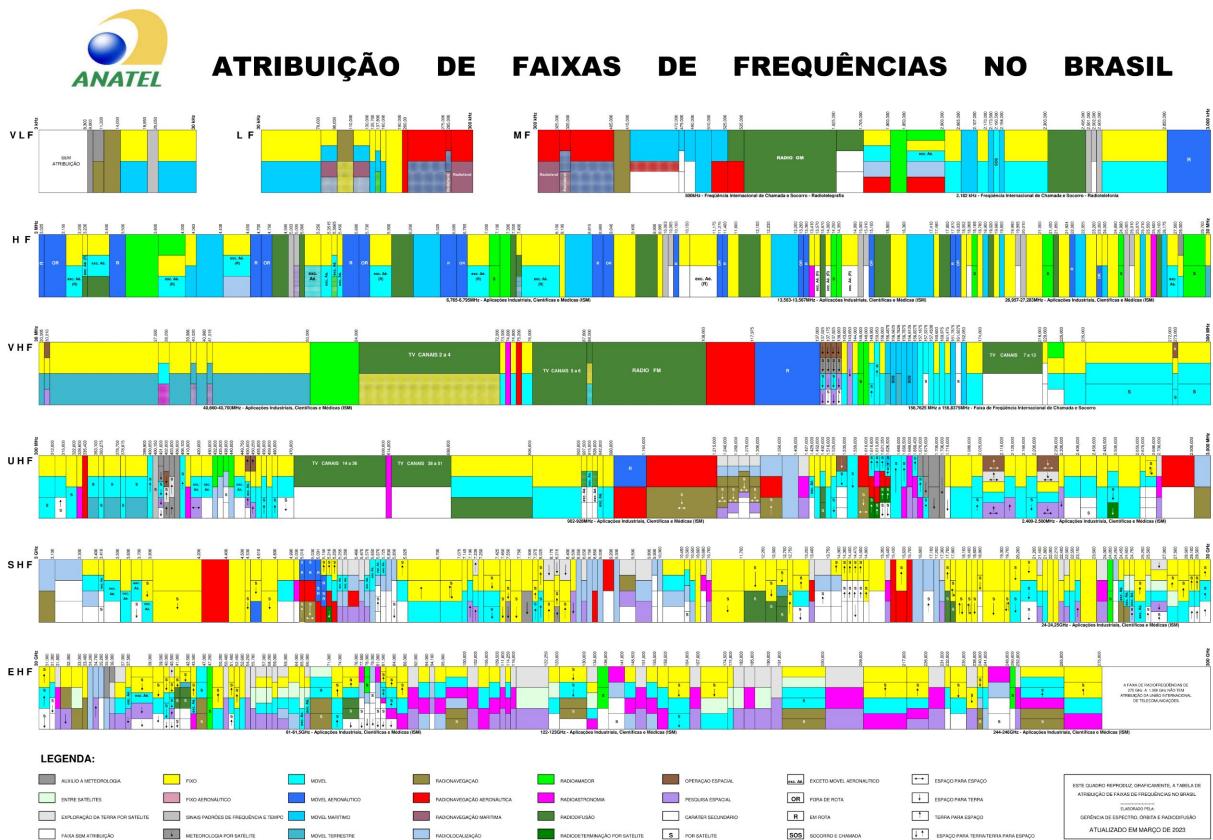
2.2 Específicos:

1. Compreender a alocação do espectro de radiofrequência brasileiro.
2. Representar matematicamente sinais eletromagnéticos de rádio frequência no domínio do tempo e no domínio da frequência.
3. Identificar sinais contaminantes impulsivos, com modulação na amplitude, modulação na frequência e com distribuições de ruído branco e rosa.
4. Compreender e executar o planejamento de uma observação radioastronômica.
5. Compreender os princípios de processamento de sinais, com ênfase nas técnicas de *baseline subtraction* e *MAD filtering*, utilizando o pacote de análise desenvolvido na UFCG radio telescope.
6. Analisar os fenômenos identificados em uma semana de dados do radiotelescópio Uirapuru conectado ao espectrômetro callisto.
7. Analisar os fenômenos identificados em dias selecionados de dados utilizando um RTLSDR como receptor e o radiotelescópio Uirapuru como frontend.
8. Realizar mapeamento do céu em 21cm com a minicorneta, utilizando o RTLSDR como backend e o software GNURADIO, integrado ao pacote de análise *radiotelescope*.

3 Material e Métodos/Metodologia:

1. Alocação de espectro de frequência brasileiro: A Lei Geral de Telecomunicações (9.472 de 16 de julho de 1997) atribui a ANATEL a responsabilidade por definir o plano de atribuição, destinação e distribuição de faixas de frequência do Brasil, conforme sua Resolução nº 759, de 19 de

janeiro de 2023. É fundamental conhecer o espectro na região de operação do radiotelescópio Uirapuru para compreender quais as possíveis interferências em seu sinal.



2. Representação Matemática de Sinais Eletromagnéticos: a maioria da análise de sinais será realizada utilizando a linguagem de computação Python. Para que se tenha sucesso nesta tarefa é importante compreender como sinais podem ser representados em seus harmônicos e representá-los no domínio do tempo e no domínio da frequência. A visualização destes dados pode ser feita utilizando-se o GNURadio e um dispositivo RTLSDR.
 3. Ruídos: compreender as propriedades matemáticas do ruído branco e do ruído rosa (também conhecido como ruído $1/f$) e aprender a simular estes ruídos com o apoio do python.
 4. Planejamento de Observação Astronômica: aprender a utilizar a ferramenta radiotelescope, desenvolvida para planejar as observações do Uirapuru, compreendendo quais são os objetos locais de interesse, como realizar filtros e como visualizar os objetos em gráficos em coordenadas equatoriais e horizontais.
 5. Estudar a literatura referente ao processamento de sinais com as técnicas de remoção e mitigação de RFI.

6. Aprender a implementar os processamentos de sinal com o uso dos pacotes desenvolvidos pela colaboração.
7. Analisar dados reais detalhadamente, fazendo o flagging manual de RFI no sinal medido e comparando os resultados com os resultados obtidos com a utilização dos procedimentos automáticos de filtragem, observando com cuidado a existência de correlações entre os objetos no feixe e os sinais observados.
8. Utilizar os pacote para computação científica MATPLOTLIB (HUNTER, 2007), NUMPY(OLIPHANT, 2006–), PANDAS (MCKINNEY et al., 2010) SCIPY (JONES et al., 2001–).
9. Utilizar os pacotes para astronomia ASTROPY (Price-Whelan et al., 2018), ASTROQUERY (Ginsburg et al., 2019) e SKYFIELD (Rhodes, 2019).

4 Resultados Parciais:

Como preparação e condicionamento para as atividades experimentais e computacionais foram realizados estudos teóricos e revisões bibliográficas, como:

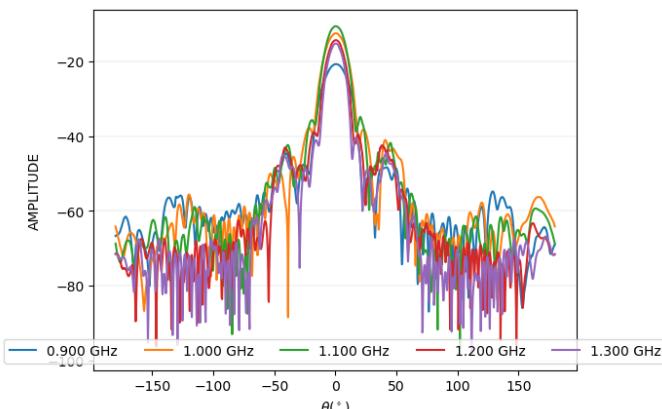
1. Estudo da bibliografia (KRAUS; MARHEFKA, 2002) para compreensão do fenômeno de interferência por dupla fenda, por fenda finita no limite de Fraunhofer e estudo da difração por um orifício circular.

O radiotelescópio Uirapuru é uma corneta com abertura circular e podemos modelar a forma de seu feixe se modelarmos a função de iluminação da abertura tal que o padrão do feixe pode ser obtido pela transformada de Haenkel da função de iluminação.

$$E(\rho, \omega) \propto 2\pi \int_0^{\infty} A(\rho) J_0 \left(\frac{2\pi\rho\rho'}{\lambda z} \right) \rho' d\rho' \quad (1)$$

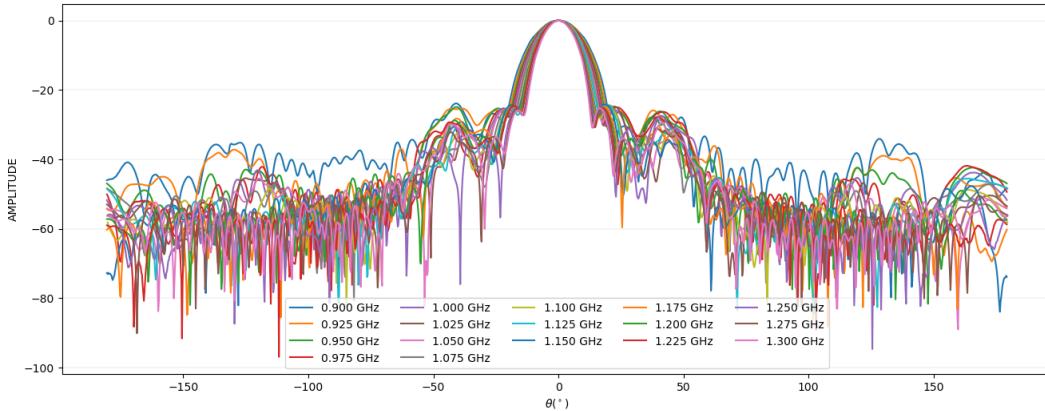
2. Apresentação ao conjunto de dados experimentais para a medida do feixe da corneta do radiotelescópio Bingo e Uirapuru;

O feixe da corneta foi medido no LIT - INPE (Laboratório de Testes Integrados do Instituto Nacional de Pesquisas Espaciais) e podemos visualizá-lo com a utilização do python.

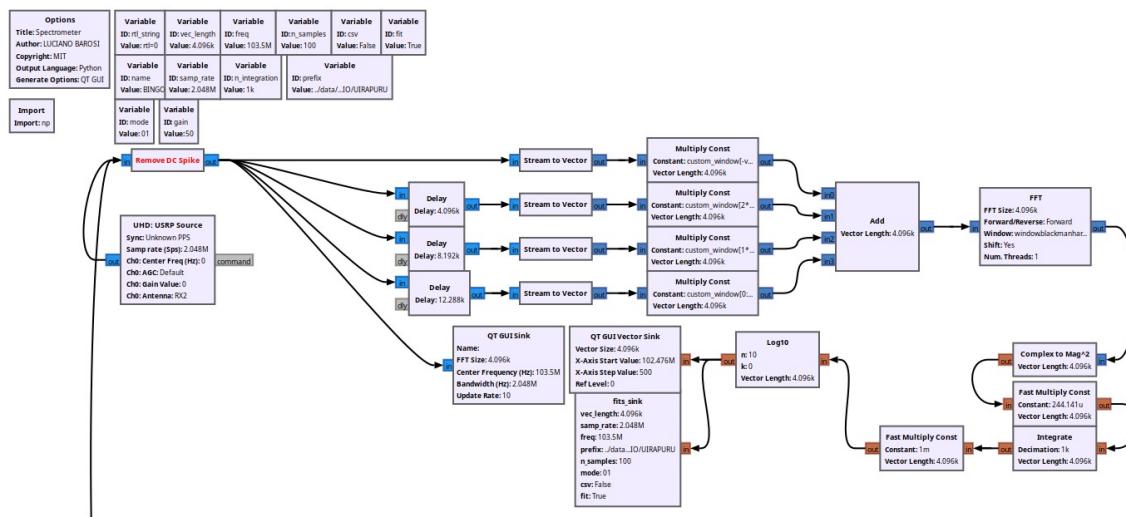


3. Normalização do padrão de antena - procedimentos matemáticos e computacionais;

Para estudarmos o padrão do feixe da corneta é necessário realizar uma normalização, de forma que o máximo em todas as frequências seja o mesmo valor de referência. Da mesma forma, é necessário que o máximo do padrão do feixe esteja localizado no ângulo nulo. Assim, o primeiro passo para a análise do feixe é efetuar esta manipulação nos dados, deslocando o máximo para a intensidade de $A = 0\text{dB}$ e o máximo para $\theta = 0^\circ$.



Já as atividades experimentais e computacionais realizadas até agora foram a coleta de dados através do radiotelescópio UIRAPURU, que coletou dados por cerca de 6 dias consecutivos utilizando um flow no software *GNURADIO* com o auxilio de um USRP ETTUS N210 (S/N F51535) que, por sua vez, está conectado a um computador por meio da porta ethernet com velocidade de 1000Mb/s.



O flow inicia removendo a frequência DC que é um sinal gerado pelo próprio USRP, passa por uma cadeia de processamento de sinais UOLA que separa o sinal recebido por quatro blocos *Stream to Vector* é geralmente utilizado como um pré-processamento dos dados antes de aplicar outros blocos de processamento de sinal. Ele permite a conversão do fluxo contínuo de dados em um formato mais conveniente para a análise, permitindo a aplicação de algoritmos de processamento que trabalham com vetores. Em seguida o bloco *multiply const* normaliza os dados antes de condicionar esse sinal a uma FFT (fast fourier transform) a FFT é uma forma muito mais rápida e eficiente que a DFT (discret

fourier transform). Ao final é realizada a integração nos dados e a gravação dos mesmos em formato FIT.

Os arquivos FITs estão no mesmo formato do *Callisto*, que é um pouco diferente do formato FITs padrão. Esses FITs (em formato float64) estão divididos em duas partições, a partição primária e outra partição (esse tipo de arquivo possui um funcionamento semelhante a um HD), na partição primária temos a matriz (1024x200) que nada mais é do que os dados armazenados no FITs em questão, mostrando o sinal em decibel capturado e quando ele foi capturado. e na outra partição temos uma tabela binária.

```
hdu = fits.open(filename[0])
hdu.info()
```

```
Filename: ../data/raw/GNURADIO/UIRAPURU_1000_20230301_234735_59.fit
No. Name Ver Type Cards Dimensions Format
 0 PRIMARY 1 PrimaryHDU 22 (1024, 1000) float64
 1 1 BinTableHDU 14 1R x 2C [1000D, 1024D]
```

Com essas duas partições, no fim, obtemos de forma organizada uma matriz que nos informa o tempo no qual a amostra foi coletada e as informações de frequência, assim podendo ser expressa graficamente em função de frequência (y) e tempo (x). A leitura desses arquivos é realizada através do método *fits* que está contido no conjunto *io* do *Astropy*.

Após a coleta e condicionamento desses dados, processamos separadamente cada frequência com o auxílio do pacote *radiotelescope* (que é um pacote desenvolvido dedicado ao projeto) onde cada imagem plotada é composta por um gráfico que mostra os objetos que cruzaram o feixe do uirapuru, informação obtida do pacote *skyfield* e está em função da declinação no eixo das ordenadas e tempo no eixo das abscissas. temos a escala de intensidade do sinal em decibel (dB) na parte inferior das imagens e na lateral direita um gráfico que mostra a mediana do sinal obtido em cada faixa de frequência. já o gráfico principal é justamente o conjunto de todos os arquivos FITs referentes a aquela frequência e o ao intervalo de tempo escolhido no formato *waterfall* do *matplotlib*.

A função responsável por gerar os três gráficos é:

```
def plot_timeseries(self, df_data, interval="1h"):
    """Plot waterfall and upper panel with celestials."""
    df_sky = self.get_all_beam()
    df_sky["TIME"] = pd.to_datetime(df_sky.TIME.values,
                                    unit="D", origin="julian")
    df_sky["TIME"] = df_sky["TIME"].dt.tz_localize("UTC").dt.tz_convert(self
                                                                      .instrument.timezone)
    df_fit = df_data.copy()
    df_fit = df_fit.reset_index()
    df_fit["index"] = df_fit["index"].dt.\
        tz_localize(self.instrument.timezone)
    df_fit = df_fit.set_index("index")
    # Set up the axes with gridspec
    freqs = df_fit.columns
```

```

begin = df_fit.index[0]
end = df_fit.index[-1]
ymin = self.pointings.dec.min().degree - self.instrument.fwhm
ymax = self.pointings.dec.max().degree + self.instrument.fwhm
fmt_major = mdates.MinuteLocator(interval=30)
fmt_minor = mdates.MinuteLocator(interval=15)
# Set up the axes with gridspec
mt = mdates.date2num((begin, end))
hfmt = mdates.DateFormatter('%Y-%m-%d\n%H:%M:%S', tz=self.instrument.
                           timezone)

# create grid format.
#fig = plt.figure(figsize=(16, 8))
fig = plt.figure(figsize=(16, 10))
#grid = plt.GridSpec(5, 8, hspace=0.0, wspace=0.1)
grid = plt.GridSpec(10, 9, hspace=0.0, wspace=0.1)
#spectrum_ax = fig.add_subplot(grid[1:, :-1])
spectrum_ax = fig.add_subplot(grid[2:-2, :-1])
sky_ax = fig.add_subplot(grid[0:2, :-1], xticklabels=[],
                        sharex=spectrum_ax)
ver_fig = fig.add_subplot(grid[2:-2, -1], sharey=spectrum_ax)
# waterfall.
main = spectrum_ax.imshow(df_fit.T, aspect='auto',
                           extent=[mt[0], mt[-1], freqs[-1], freqs[0]],
                           cmap=cm.inferno)
spectrum_ax.set_ylabel("Frequencies (MHz)")
spectrum_ax.set_xlabel("Time (local time)")
spectrum_ax.minorticks_on()
spectrum_ax.xaxis.set_major_formatter(hfmt)
spectrum_ax.xaxis.set_minor_locator(fmt_minor)
spectrum_ax.xaxis.set_tick_params(which='minor', bottom=True)
# colorbar
cbax = fig.add_subplot(grid[9, :-1])
Colorbar(ax=cbax, mappable=main, orientation="horizontal",
         ticklocation="bottom")

# SKY
if not df_sky.empty:
    mask = (df_sky["TIME"] > begin) & (df_sky["TIME"] < end)
    df_sky_filt = df_sky.loc[mask]
    for celeste in df_sky_filt.NAME.unique():
        sky = df_sky_filt[df_sky_filt.NAME == celeste]
        sky_ax.scatter(x=sky.TIME, y=sky.DEC, label=celest,
                       marker="*")
    sky_ax.axhline(ymin, color="gold", linewidth=2)
    sky_ax.axhline(ymax, color="gold", linewidth=2)
    sky_ax.set_ylabel("Declination")
    sky_ax.xaxis.tick_top()
    sky_ax.xaxis.set_minor_locator(fmt_minor)
    sky_ax.xaxis.set_major_formatter(hfmt)
    sky_ax.grid()

```

```

sky_ax.legend(loc='upper center', bbox_to_anchor=(0.5, 2.0),
              ncol=7, fancybox=True, shadow=True)

# Vertical Plot
spectrum = df_fit.median(axis=0)
# plot averaged spectrum in the vertical.
ver_fig.plot(spectrum, freqs, c='red')
ver_fig.grid()
ver_fig.yaxis.tick_right()
ver_fig.yaxis.set_label_position('right')
ver_fig.set_xlabel("digits")

return fig

```

As faixas de frequência que coletamos para processar visa estudar inicialmente as frequências mais comuns utilizadas por satélites GNSS e assim catalogar e mapear suas características e padrões de interferência.

satélites GPS:

Band	v (MHz)	PSD	γ_0 (MHz)	Power (dBm)
L5-I5	1176.45	BPSK (10)	1.023	-154.0
L2-L2C	1227.60	BPSK (1)	1.023	-158.5
L2-P(Y)	1227.60	BPSK (10)	1.023	-158.5
L2-M	1227.60	BOCs (10, 5)	1.023	-158.5

Satélites GLONASS:

Band	v (MHz)	PSD	γ_0 (MHz)	Power (dBm)
L5-I5	1176.45	BPSK (10)	1.023	-154.0
L2-L2C	1227.60	BPSK (1)	1.023	-158.5
L2-P(Y)	1227.60	BPSK (10)	1.023	-158.5
L2-M	1227.60	BOCs (10, 5)	1.023	-158.5

Satélites GALILEO:

Band	ν (MHz)	PSD	ν_0 (MHz)	Power (dBm)
E5a-1	1191.79	altBOC (15, 10)	1.023	-158
E6B	1278.75	BPSK (5)	1.023	-158
E6A	1278.75	BOCc (10.5)	1.023	-158

Satélites BEIDOU:

Band	v (MHz)	PSD	v_0 (MHz)	Power (dBm)
B2I	1191.795	ALTBLOC (15, 10)	1.023	-163
B3	1268.52	BPSK (10)	1.023	-163
B3-A	1268.52	BOC (15, 2.5)	1.023	-163

É possível comparar os sinais adquiridos e processados a fim de identificar o satélite responsável pela sua geração. Os sinais foram coletados em frequências predefinidas no flow do gnuradio, especificamente em 1000MHz, 1170MHz, 1200MHz, 1220MHz, 1240MHz e 1270MHz. A atividade subsequente consiste na correlação dos sinais com os satélites e na aplicação de técnicas adicionais de processamento de sinais para detectar sinais que possam estar ocultos nos dados.

Figura 1 – 1200MHz

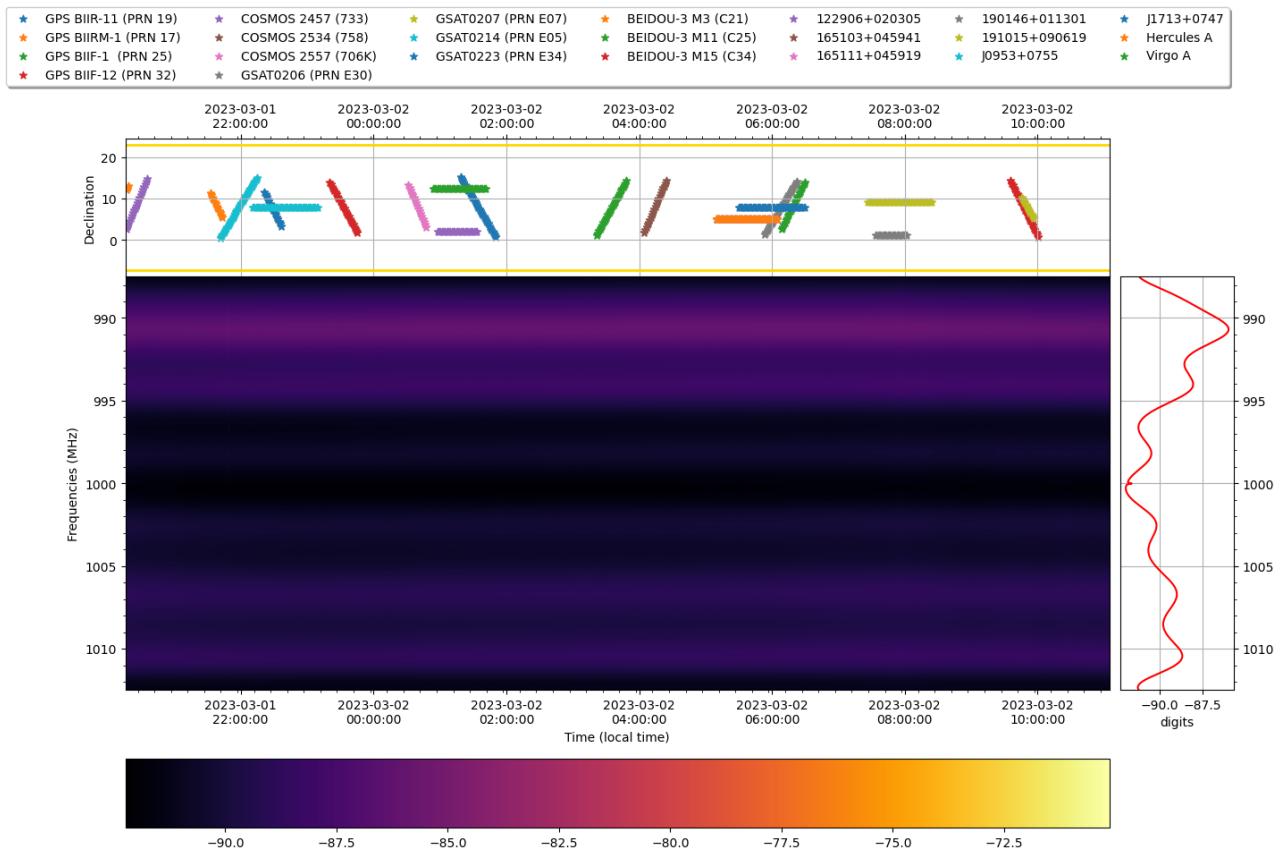


Figura 2 – 1170MHz

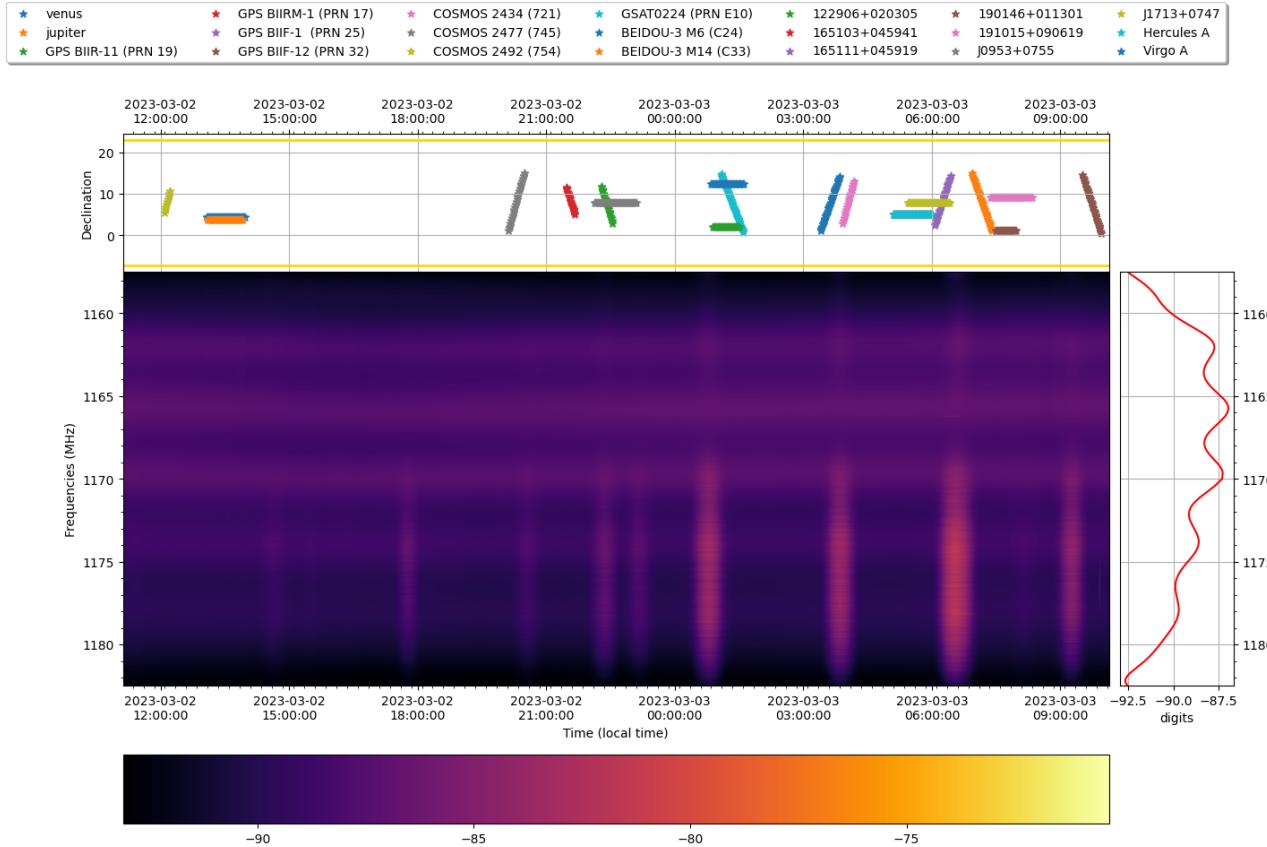


Figura 3 – 1200MHz

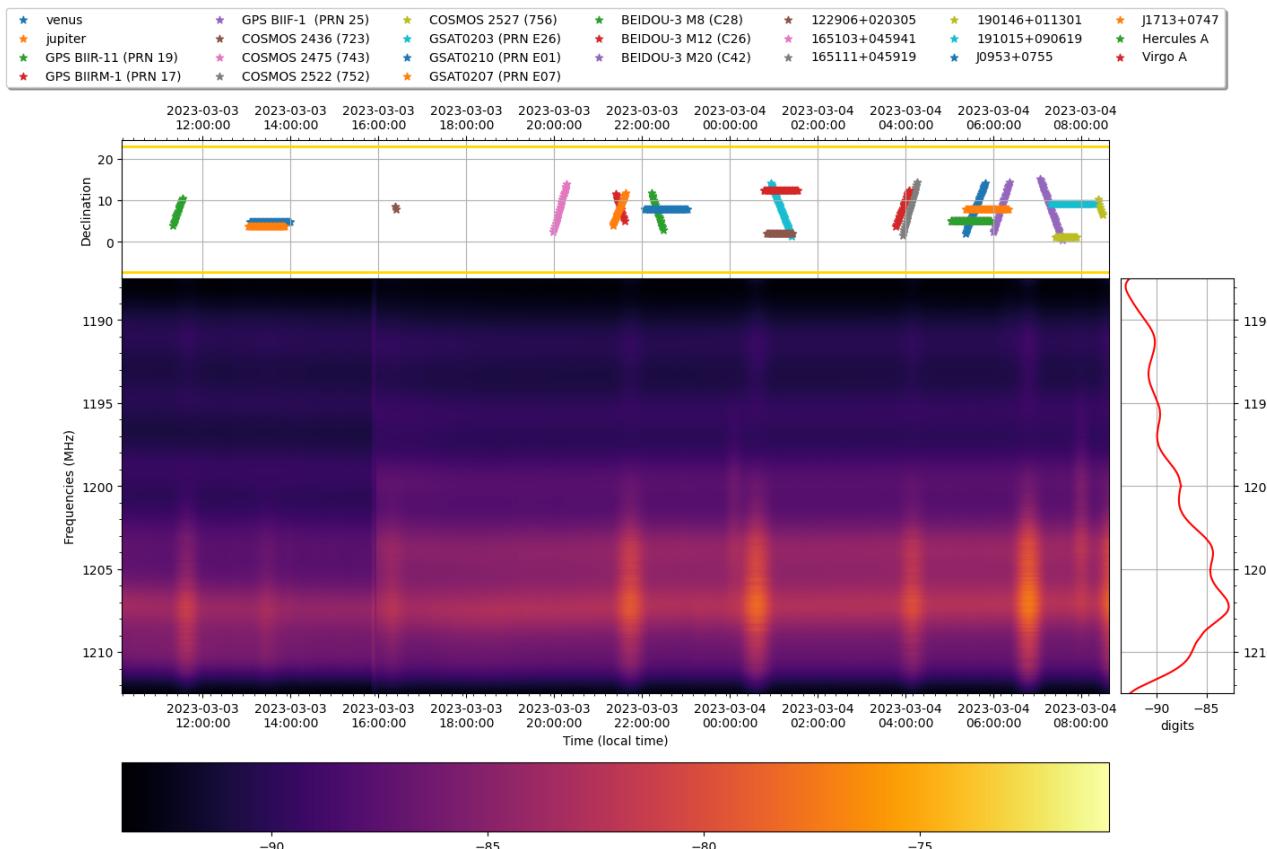


Figura 4 – 1220MHz

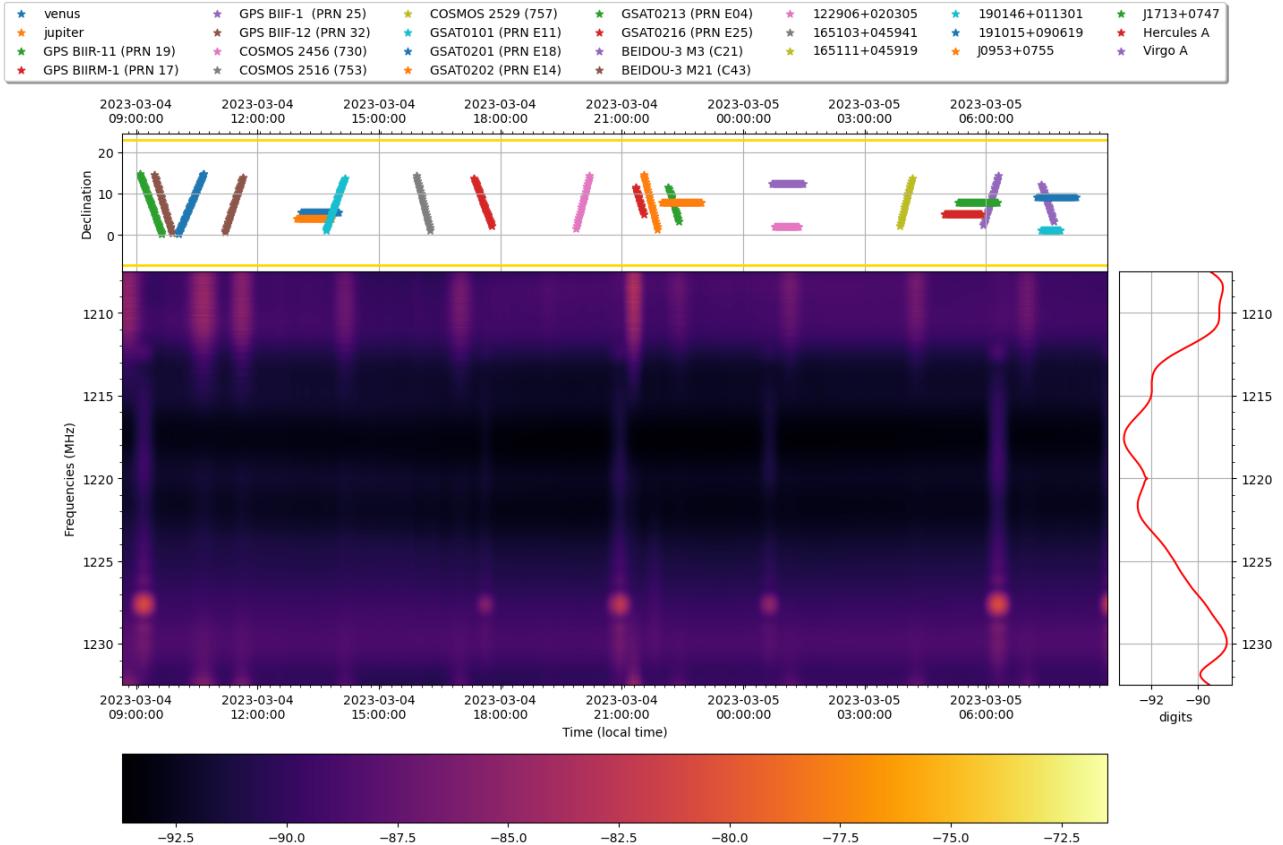


Figura 5 – 1240MHz

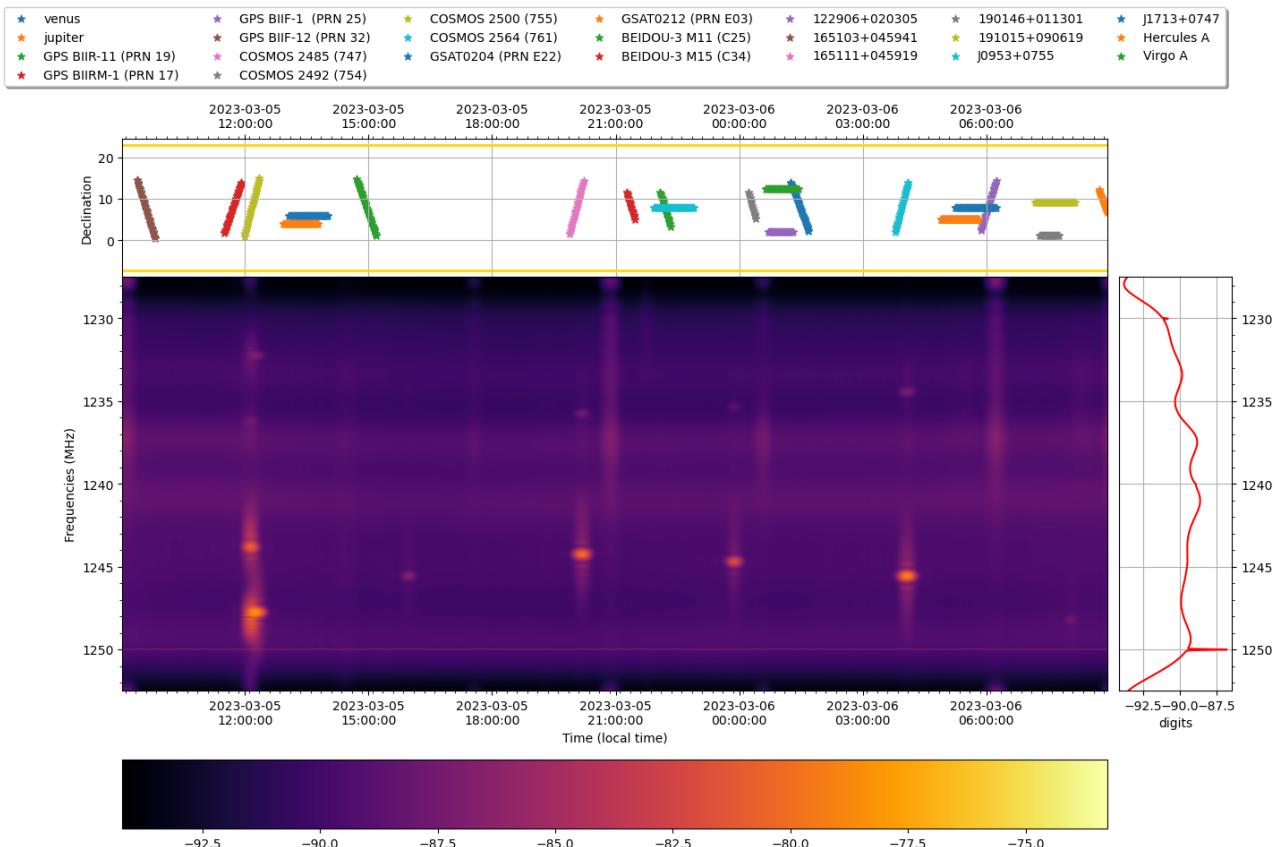
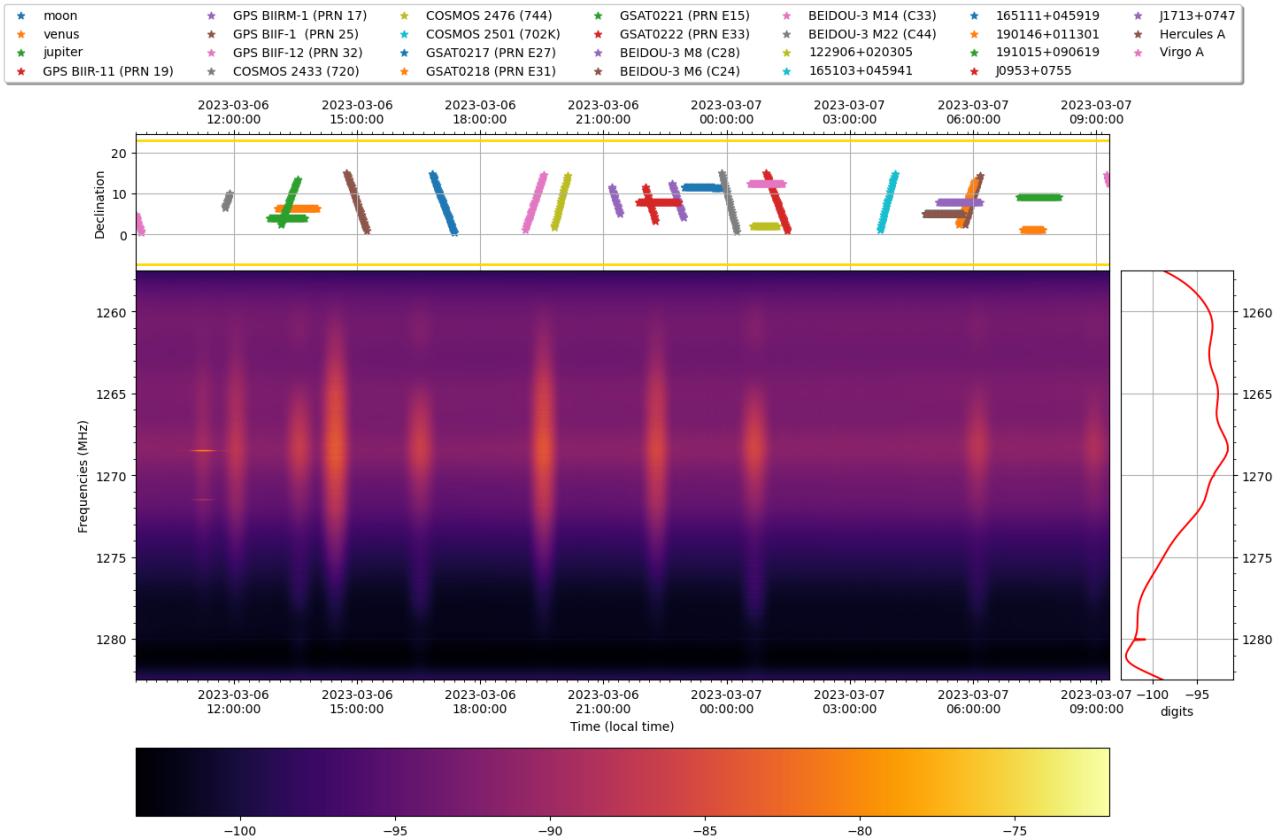


Figura 6 – 1270MHz



5 Resultados e discussões.

Após a analise dos resultados parciais fez-se necessária uma nova calibração a respeito da própria captação de sinais. Na analise desses dados nós fizemos uma simulação com base nos bancos de dados de satélites de como deveríamos estar observando os satélites com o *Uirapuru*. Essa segunda parte do projeto se deu quase que exclusivamente no laboratório onde foram realizadas varias medições utilizando o instrumento, não só a antena como também o receptor que fica localizado numa caixa isolada ao lado da antena. Essas medidas foram e estão sendo fundamentais para a calibração do instrumento e já geraram resultados impressionantes na própria observação dos satélites. Para a calibração seguimos um roteiro de coleta de medidas que consistia em:

Configuração do Analisador de Espetro:

- tempo de captura: 15 min
- Freqs: 900 - 1500MHz
- RBW 20Khz.

E com essas configurações foram feitas cerca de 20 medidas dos seguintes sistemas:

1. Cadeia 1 com terminação de 50 Ohms ligada diretamente no Analisador.
2. - Medida 2: Cadeia 2 com terminação de 50Ω ligada diretamente no Analisador.
3. Com o FieldFox (Analizador de espectro portátil), salvar o espectro em MAX HOLD com uma integração de 1min, por cinco vezes consecutivas, com o fieldfox ligado diretamente no cabo do Uirapuru que vem de uma das polarizações.
4. mesma coisa com a outra polarização.
5. Com o analisador, receptor com uma terminação e ligada pelo cabo longo.
6. Mesma coisa com a outra polarização.
7. Com o analisador, medir uma das polarizações passando pelo receptor e pelo cabo.
8. Mesma coisa com a outra polarização.

Apos a coleta dessas medidas inicia-se o roteiro para a calibração do instrumento que consiste nos seguintes passos:

1. Determinar o ganho integrando o padrão de feixe, com a seguinte expressão:

$$G_\nu = \frac{4\pi}{\iint |B(\theta)|^2 \sin \theta d\theta d\phi}$$

Onde:

- $B(\theta)$ representa o padrão de radiação ou o diagrama de irradiação da antena. Esse padrão descreve como a antena direciona ou irradia energia em diferentes direções em torno dela.

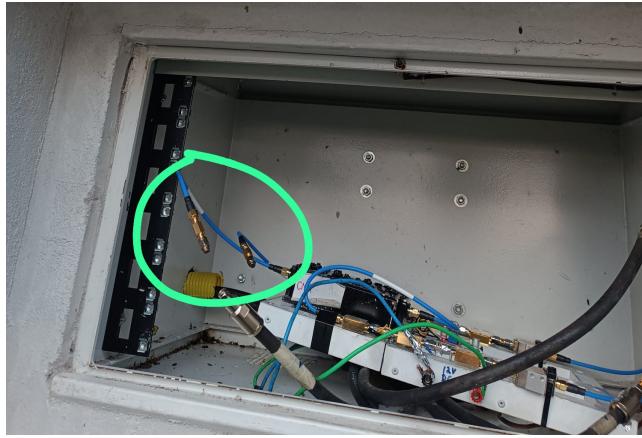
$-G_\nu$ É o ganho da antena na frequência . (O ganho de uma antena é uma medida da direcionalidade com a qual ela irradia energia e recebe sinais de uma determinada direção em comparação com uma antena de referência.)

É importante pontuar que como queremos calcular o ganho da antena em relação ao que seria a antena ideal já se pode notar a importância de se comparar uma simulação do que deveria ser observado com o Uirapuru e o que de fato foi captado pelo instrumento.

2. Medições quando uma terminação de 50Ω é conectada a entrada de sinais do receptor do Uirapuru no lugar da própria antena. Esse passo é crucial para a calibração do receptor do uirapuru pois com ele pode-se medir o ruido gerado pelo receptor já que a carga recebida pelo mesmo é constante vinda da terminação e não varia como seria o caso de observar o céu com o uirapuru.



(a) Receptor do Uirapuru.



(b) Receptor com terminações

A potencia media do sinal captado com o receptor nas terminações é denotada por P_{load} e é usada para medir o fator de ruído (Noise Figure) $NF[dB]$ do sistema, a partir da expressão:

$$NF[dB] = M(P_{\text{load}}(dB)) - 10 \log \left(\frac{k_B T_{\text{cal}} BW}{0.001} \right)$$

Onde:

- $M(P_{\text{load}}(dB))$ representa a medição do fator de potência realizada anteriormente com a carga de 50Ω das terminações.

- k_B é a constante de Boltzmann.

- T_{cal} é a temperatura de calibração.

- $\frac{BW}{0.001}$ é a largura de banda eficaz (bandwidth) do sistema.

3. Obter o valor da potencia calibrada $P_{\text{cal}}(d_Bm)$ que é obtido através da expressão:

$$P_{\text{cal}}(d_Bm) = P_{\text{meas}}(dB) - NF$$

onde:

- $P_{\text{cal}}(d_Bm)$ é a potencia calibrada em um ponto d_Bm .

- $P_{\text{meas}}(dB)$ é a potência medida nesse ponto.

-NF (fator de ruido) é subtraído da medição $P_{\text{meas}}(dB)$ para obter o valor calibrado.

4. Baixando os dados do Fluxo Solar do Observatório Learmonth, interpolando e calculando a média em 1 hora e obtendo a densidade espectral de fluxo solar com a seguinte expressão:

$$S_{\nu,0} = \frac{4\pi\nu^2}{G_v c^2} \frac{10^{\frac{P_{\text{cal,sun}}}{10}}}{BW} \frac{10^{-3}W}{mW} \frac{10^{22}SFU}{Wm^{-2}Hz^{-1}}$$

Onde:

- $S_{\nu,0}$ é a densidade espectral de fluxo solar.

- ν é a frequencia.

- G_v é o ganho da antena.

- c é a velocidade da luz.

- $P_{\text{cal,sun}}$ é a potencia calibrada do sol.

- BW é a largura de banda eficaz (bandwidth).

Os novos gráficos waterfall dos mesmos dados apos a calibração se encontram a seguir e podem ser comparados com os gráficos do resultado parcial.

Figura 8 – 1170MHz calibrado.

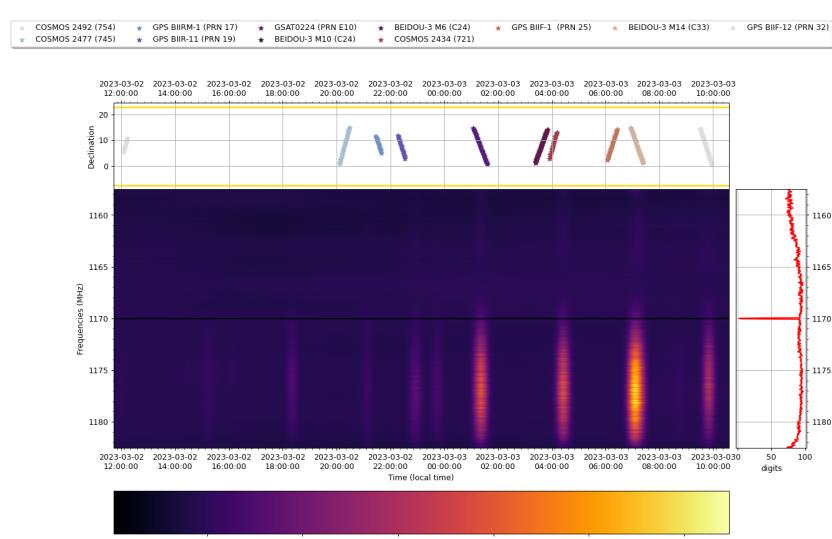


Figura 9 – 1200MHz calibrado.

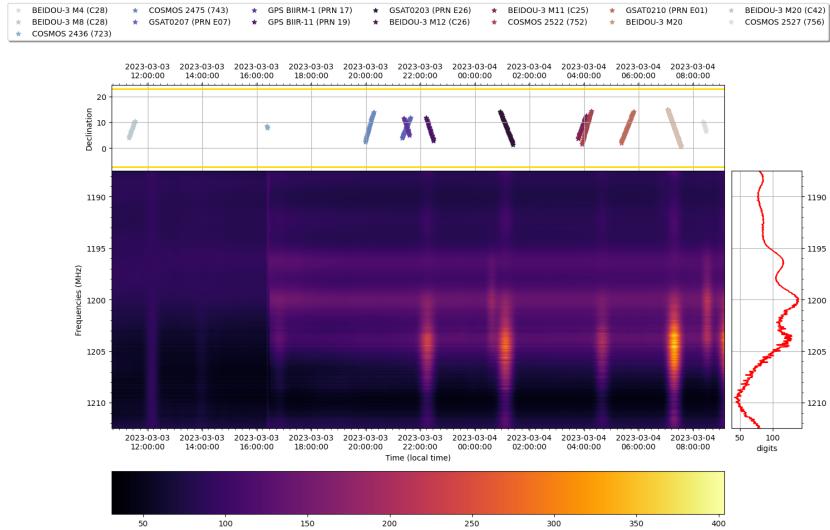


Figura 10 – 1170MHz calibrado.

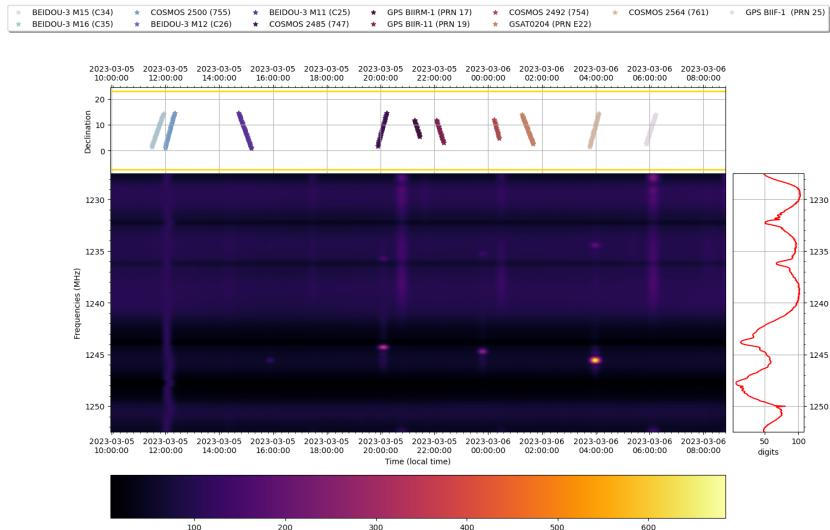
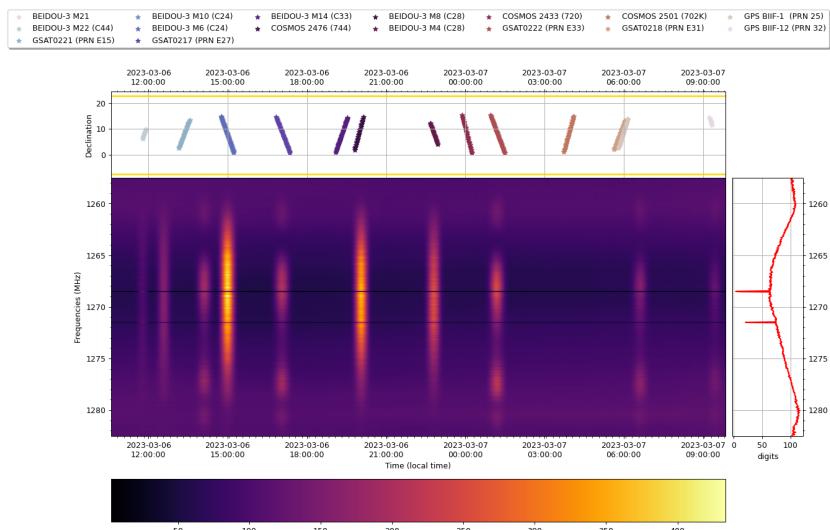


Figura 11 – 1170MHz calibrado.



6 Conclusão:

Após obter todas essas informações a partir dos dados foi possível aprimorar muito a precisão do Uirapuru e assim concluir o objetivo do projeto que seria justamente o comissionamento e calibração do instrumento para fazer ciência e servir seus propósitos para com o radiotelescópio BINGO e suas futuras pesquisas.

7 Agradecimentos:

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil.

Referências

Ginsburg, A. et al. astroquery: An Astronomical Web-querying Package in Python. , v. 157, p. 98, mar. 2019.

HUNTER, J. D. Matplotlib: A 2d graphics environment. Computing in Science & Engineering, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

JONES, E. et al. SciPy: Open source scientific tools for Python. 2001–. [Online; accessed]. Disponível em: <<http://www.scipy.org/>>.

KRAUS, J. D.; MARHEFKA, R. J. Antenna for all applications. [S.l.]: McGraw-Hill, 2002.

MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. Proceedings of the 9th Python in Science Conference. [S.l.], 2010. v. 445, p. 51–56.

OLIPHANT, T. NumPy: A guide to NumPy. 2006–. USA: Trelgol Publishing. [Online; accessed <today>]. Disponível em: <<http://www.numpy.org/>>.

Price-Whelan, A. M. et al. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. , v. 156, p. 123, set. 2018.

Rhodes, B. Skyfield: High precision research-grade positions for planets and Earth satellites generator. 2019. ascl:1907.024 p. Astrophysics Source Code Library, record ascl:1907.024.