

# INFORME

## INSTRUMENTACIÓN

## ASTRONÓMICA

Tarea 1

Estudiante: Camilo Núñez Barra  
 RUT: 20.533.326-6  
 Profesor: Leonardo Bronfman  
 Auxiliar: Paulina Palma  
 Curso: AS3201 Astronomía Experimental  
 Fecha: 4 de mayo de 2021

### 1. Introducción

Se utiliza el radiotelescopio MINI.

### 2. Hot–Cold Test

#### 2.1. Marco teórico

La señal de un objeto celeste recibida por un radiotelescopio debe pasar por la atmósfera antes de llegar al receptor de la antena. El receptor tiene corrientes internas y fenómenos de transporte de electrones y fotones con ondas estacionarias que aumentan la entropía y generan también un error sistemático correspondiente a ruido blanco.

El ruido blanco del receptor se puede disminuir al aumentar el tiempo de integración pues la señal del cielo permanece constante y la señal del receptor disminuye relativamente su tamaño, por lo que disminuir el ruido permite disminuir el tiempo invertido para detectar una señal astronómica y a la vez permite no empeorar la señal.

Una manera común de eliminar errores sistemáticos es mediante la calibración del instru-

mento de medición.

El método Hot–Cold Test permite calibrar el telescopio al caracterizar la temperatura de ruido del receptor mediante dos cargas cuyas temperaturas son conocidas y diferentes.

Una carga es un material absorbente adherido a un pedazo de madera con un mango. El material es un absorbente electromagnético que absorbe la radiación con muy poca reflexión por lo que se supone como cuerpo negro.

Sea  $T_{\text{rec}}$  la temperatura de ruido del receptor y  $G_{\text{rec}}$  la ganancia del receptor. Una carga a temperatura ambiente  $T_{\text{hot}}$  se pone enfrente de la bocina de la antena, permitiendo medir una potencia espectral  $W_{\text{hot}}$  dada por,

$$W_{\text{hot}} = G_{\text{rec}} k T_{\text{rec}} + G_{\text{rec}} k T_{\text{hot}}, \quad (2.1)$$

donde  $k$  es la constante de Boltzmann. Análogamente, para una carga fría a temperatura  $T_{\text{cold}}$ , la potencia espectral medida es,

$$W_{\text{cold}} = G_{\text{rec}} k T_{\text{rec}} + G_{\text{rec}} k T_{\text{cold}}. \quad (2.2)$$

Se define el factor  $Y$  como el cuociente entre la medición de la potencia espectral para la carga

caliente y para la fría,

$$Y = \frac{W_{\text{hot}}}{W_{\text{cold}}}, \quad (2.3)$$

que permite determinar la temperatura de ruido del receptor mediante variables medidas,

$$T_{\text{rec}} = \frac{T_{\text{hot}} - YT_{\text{cold}}}{Y - 1}. \quad (2.4)$$

En términos de operación del telescopio, la medición de  $T_{\text{rec}}$  se hace cada día que es observa, por lo que en una campaña de meses se tiene que hacer todos los días la medición. Esta constante medición corresponde a un chequeo del estado de la electrónica del telescopio pues si repentinamente difiere mucho la temperatura de ruido del receptor con respecto a la medición del día anterior es porque está funcionando mal y se debe arreglar. En realidad se usa la potencia por canales de frecuencia y no la potencia espectral de todo el ancho de banda, tal como se describe en la sección 2.4, por lo que la temperatura de ruido del receptor cambia con la frecuencia.

## 2.2. Datos y metodología

Esta calibración utiliza una carga caliente a temperatura ambiente en la cúpula del radiotelescopio MINI y una carga fría a temperatura de nitrógeno líquido, cuyas mediciones están en la tabla 1.

El MINI es pequeño, permitiendo acceder a la bocina por una escalera. Se sube y se pone la carga caliente enfrente de la bocina procurando apuntar el material absorbente a ella. Mediante un *powermeter* se mide la potencia integrada de la señal para todo el ancho de banda que tiene el receptor y se anota la lectura en la tabla 1.

A continuación y análogamente, se pone la carga fría enfrente de la bocina y se mide la potencia espectral con el *powermeter* pero esperando a que la lectura correspondiente converja tras disminuir la temperatura. Esta medición está en la tabla 1.

El *powermeter* entrega potencias en decibelio-milivatio (dBm), que es una escala logarítmica acorde a las eventuales amplificaciones y disminuciones de las señales. Una potencia  $W$  en escala logarítmica de dBm se convierte en una potencia  $P$  en escala lineal de  $W$  como se muestra a continuación,

$$P = 10^{\frac{W-3}{10}}. \quad (2.5)$$

## 2.3. Cálculo de $T_{\text{rec}}$

Se convierten las potencias de la tabla 1 a vatios según la ecuación 2.5 y se calcula  $Y$  según la ecuación 2.3, obteniendo  $Y = 2.2$ . Esto permite usar la ecuación 2.4 para obtener  $T_{\text{rec}} = 107.6\text{ K}$ . Este cálculo se desarrolla en el código 1.

## 2.4. Comparación con calibración del MINI

El software del MINI tiene el comando %hct para ingresar todo el sistema a una subrutina de Hot–Cold Test. Se usan cargas las mismas temperaturas de la tabla 1.

El sistema espera a que se ponga la carga caliente en la bocina del receptor y se marca la medición al presionar una botonera, permitiendo tener una potencia por cada canal de frecuencia de la señal, mostrando una variación en todo el espectro. Ahora se pone la carga fría y se apreta la botonera, midiendo una potencia por cada canal que también varía en todo el espectro pero que es menor.

Se usan los dos vectores de potencia para calcular el factor  $Y$  según la ecuación 2.3 y luego la temperatura de ruido del receptor por canal según la ecuación 2.4. Finalmente, se promedian las temperaturas de todos los canales, resultando una temperatura de ruido del receptor de  $T'_{\text{rec}} = (150.9 \pm 4.6)\text{ K}$ .

Se aprecia que  $|T'_{\text{rec}} - T_{\text{rec}}| = 43.3\text{ K}$ .

## 3. Antenna Dipping

### 3.1. Marco teórico

La turbulencias atmosféricas y el vapor de agua presente en la atmósfera distorsionan la señal de un objeto celeste dando lugar a un error sistemático que no depende del radiotelescopio.

La atmósfera en sus distintas capas varía la temperatura y la densidad pero en este experimento se hace una aproximación a primer orden para establecer que contribuye una temperatura de ruido  $T_{\text{atm}}$  fija. Además, se hace también una aproximación al igualar la temperatura ambiente del domo  $T_{\text{amb}}$  con  $T_{\text{atm}}$ .

La opacidad cenital  $\tau_w$  debido al contenido de agua en la atmósfera y la opacidad cenital total  $\tau_0$

de la atmósfera, que incluye al oxígeno, se aproximan también a primer orden para decir que son iguales.

El método Antenna Dipping permite calibrar el receptor del telescopio al determinar el estado y opacidad cenital de la atmósfera.

Se mide la potencia del cielo  $W_{\text{sky}}$  detectada por el telescopio a un determinado intervalo de frecuencias y elevación  $\varphi$ , que está dada por,

$$W_{\text{sky}} = c(T_{\text{rec}} + T_{\text{atm}}(1 - \exp(-\tau_0 / \sin \varphi))), \quad (3.1)$$

donde  $T_{\text{rec}}$  es la temperatura de ruido del receptor y  $c$  es una constante que, por ejemplo, puede depender de la ganancia del receptor.

Análogamente, la potencia para una carga en frente de la bocina a temperatura ambiente  $T_{\text{amb}}$  (la misma carga Hot del Hot–Cold Test), la potencia es,

$$W_{\text{amb}} = c(T_{\text{rec}} + T_{\text{amb}}). \quad (3.2)$$

Aplicando las aproximaciones, se define,

$$\Delta W \equiv W_{\text{amb}} - W_{\text{sky}} = cT_{\text{amb}} \exp(-\tau_w / \sin \varphi), \quad (3.3)$$

y  $z = \pi/2 - \varphi$ , permitiendo obtener la relación,

$$\ln(\Delta W) = -\sec(z) \tau_w + \ln(cT_{\text{amb}}), \quad (3.4)$$

que es una relación lineal de  $\ln(\Delta W)$  con respecto a  $-\sec(z)$ , donde  $\tau_w$  es la pendiente de la recta.

### 3.2. Datos y metodología

Se utiliza una carga caliente a temperatura ambiente en la cúpula del radiotelescopio MINI, midiendo la potencia que muestra Domo en la tabla 2.

Se mide con un *powermeter* la potencia espectral de diez puntos a distintas elevaciones y azimut fijo, entregando los resultados de la tabla 2.

### 3.3. Cálculo de $\tau_w$

El siguiente procedimiento corresponde al código 2. Se lee el archivo `antdip_AE2021A` provisto por el equipo docente con los datos necesarios para realizar el ajuste lineal según la ecuación 3.4. El gráfico de este se muestra en la figura 1. La pendiente de la recta establece un valor de 0.257 para la opacidad cenital debido al contenido de agua en la atmósfera.

Carga	Temperatura K	Potencia dBm
Hot	300	-44.50
Cold	77	-47.94

Cuadro 1: Temperatura y potencia medidas para las cargas del Hot–Cold Test

Punto	Elevación °	Potencia dBm
1	23.50	-45.56
2	16.60	-45.22
3	12.84	-45.01
4	10.48	-44.88
5	8.85	-44.80
6	7.66	-44.73
7	6.76	-44.71
8	6.04	-44.67
9	5.47	-44.65
10	4.99	-44.63
Domo		-44.54

Cuadro 2: Elevación y potencia para los distintos puntos a azimut fijo. Se incluye domo con carga caliente

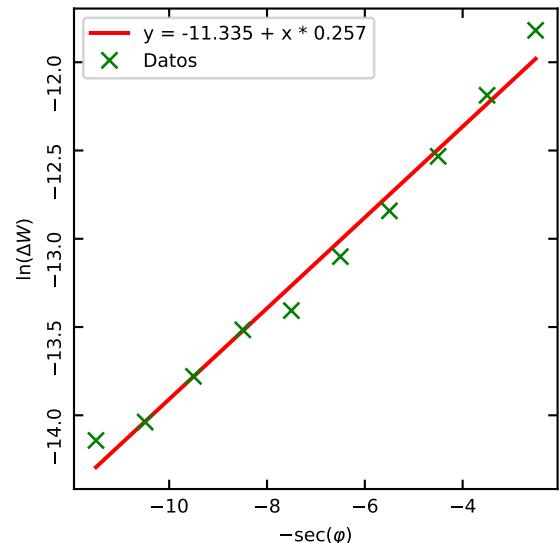


Figura 1: Equis roja es medición del telescopio. Línea azul es ajuste lineal. La pendiente es  $\tau_w$

### 3.4. Comparación con calibración del MINI

El software del MINI tiene el comando `%antdip` para ingresar todo el sistema a una subrutina de Antenna Dipping.

Primeramente se debe ingresar la cantidad de masas de aire por punto y se usa 1, que es el valor típico.

Luego el sistema recolecta automáticamente información para diez puntos separados por una masa de aire a través de la elevación a azimut fijo, midiendo la mayor elevación primero y luego disminuye progresivamente. El telescopio constantemente evalúa su posición y solo toma datos cuando está apuntando con cierta tolerancia a la coordenada determinada por el sistema de control.

La información recolectada corresponde a la diferencia de potencia de la ecuación 3.3, donde la potencia para el cielo es la que apunta según la elevación correspondiente y la potencia para la carga caliente es según el *chopper*, una carga absorbente electromagnética, que está dentro de la bocina de la antena y automática y periódicamente obstruye la visión del telescopio gracias a un motor.

Tras medir los diez puntos, automáticamente el telescopio apunta al domo y toma una medición de referencia con la carga caliente. A continuación, se debe ingresar la temperatura actual y la humedad relativa, además de un parámetro cuantitativo de 0 a 3 que indica el grado de cobertura del cielo debido a las nubes y sirve para el registro histórico.

Finalmente, el software realiza el ajuste lineal y entrega el valor  $\tau'_w = 0,423 \pm 0,023$ , además de otros parámetros como la estimación de la eficiencia del telescopio y la temperatura de brillo del agua en el cielo.

Se aprecia que  $|\tau_w - \tau'_w| = 0,166 \pm 0,023$ .

## 4. Observaciones

### 4.1. Nebulosa de Orión

Se observa la nebulosa de Orión (ver figura 2), también conocida como M42 por su identificación en el catálogo Messier. Esta es una nebulosa difusa situada al sur del cinturón de Orión (ver figura 3). La nebulosa de Orión, también conocida como Messier 42, M42, o NGC 1976, es una nebulosa difusa situada al sur del cinturón de Orión. Tiene

un radio de 1.2 ly y su distancia a la Tierra es de 1344 ly, por lo que es muy cercana y eso la convierte en una buena referencia. Además, en ella hay formación de estrellas de alta masa y también variadas moléculas y partículas, encontrando sus abundancias en *surveys*.

Antes de la observación se hace la calibración del telescopio mediante el Hot–Cold Test y Antenna Dipping, tal como se explican en las secciones 2 y 3, respectivamente. Esto es importante para tener observaciones más precisas.

Se hace una cruz en el cielo alrededor y centrada en la fuente, tal como muestra la figura 4. Se apunta en el siguiente orden respecto a la figura: arriba, izquierda, centro, derecha y abajo. Esto se repite consecutivamente para un total de 3 pasadas a la cruz.

Los siguientes parámetros se establecen en el software del MINI.

Se usa el modo observacional *position switching*, que necesita definir los puntos: *on pos*, la posición de la fuente; *off pos*, una posición de referencia y; *home pos*, un punto central del mapa que es necesario para calcular la grilla.

Se usa la coordenada ecuatorial de la nebulosa de Orión RA= $5^{\text{h}}32^{\text{m}}47^{\text{s}}$  y DEC= $-5^{\circ}24'30''$  para *on pos* y *home pos*, mientras que para *off pos* se usa RA= $5^{\text{h}}40^{\text{m}}47^{\text{s}}$  y DEC= $-5^{\circ}10'0''$ .

El ciclo *on-off* es de 30 s en total y 15 s por cada posición

El tiempo de integración es 600 s, por lo que el tiempo de observación es el doble.

Además, constantemente se asigna un tiempo de calibración de 5 s para chequear que la atmósfera no ha cambiado respecto al modelo. El espectro obtenido se normaliza al continuo.

El procedimiento para el análisis de las observaciones se muestra en el código 3 y se detalla a continuación.

### 4.2. Espectros

La observación registra los datos provistos por el equipo docente en los archivos `sdf_1xx_1xx`, donde `xx` va en orden creciente 11 a 25, según el orden de observación de la cruz explicado anteriormente.

Se miden 256 temperaturas en kelvin a distintas velocidades, formando un espectro con una preponderante línea de emisión aproximadamente



Figura 2: Nebulosa de Orión (Imagen: NASA)

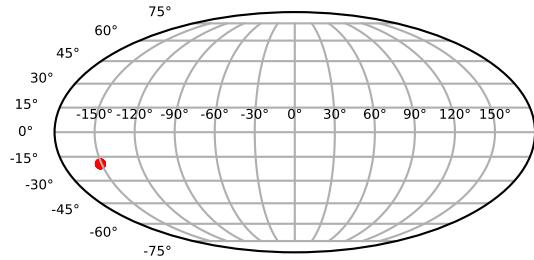


Figura 3: Coordenadas galácticas de la nebulosa de Orión

$9.5 \text{ km s}^{-1}$  a aproximadamente 30 K para el punto central. Se hace un ajuste gaussiano para cada uno de estos espectros y se grafica en las figuras 5, 6 y 7, para la primera, segunda y tercera pasada por la cruz, respectivamente.

#### 4.3. Temperatura máxima y Pointing

Se calcula la temperatura máxima del espectro para cada una de las quince observaciones. Se promedian las temperaturas correspondientes a las mismas coordenadas en la cruz y se realizan dos ajustes gaussianos, tal como muestra la figura 8. El primer ajuste es a través de la longitud galáctica, manteniendo la latitud galáctica fija, es decir, sobre los tres puntos horizontales de la cruz. El segundo ajuste es a través de la latitud galáctica, manteniendo la longitud galáctica fija, es decir, sobre los tres puntos verticales de la cruz.

El punto central de la cruz tiene una temperatura máxima de 29.215 K a longitud galáctica  $208.006^\circ$  y latitud galáctica  $-19.385^\circ$ , mientras que el modelo del ajuste gaussiano a latitud fija establece un máximo de 29.485 K a longitud  $208.975^\circ$  y el ajuste gaussiano a longitud fija establece un máximo de 29.989 K a latitud  $-19.357^\circ$ .

Los datos anteriores muestran que el pointing del telescopio es aceptable, pues no hay diferencias de más de una unidad. A pesar de que el telescopio tenía ordenado apuntar a la coordenada precisa de la nebulosa de Orión, existe un error que se puede calibrar. Además, el software del telescopio tiene una ventana de tracking que muestra que tiene permitido apuntar dentro de un radio de tolerancia de la coordenada objetivo.

#### 4.4. Temperatura integrada

Se calcula la temperatura integrada del espectro para cada una de las quince observaciones. Se promedian las temperaturas correspondientes a las mismas coordenadas en la cruz y se realizan dos ajustes gaussianos, tal como muestra la figura 8. El primer ajuste es a través de la longitud galáctica, manteniendo la latitud galáctica fija, es decir, sobre los tres puntos horizontales de la cruz. El segundo ajuste es a través de la latitud galáctica, manteniendo la longitud galáctica fija, es decir, sobre los tres puntos verticales de la cruz.

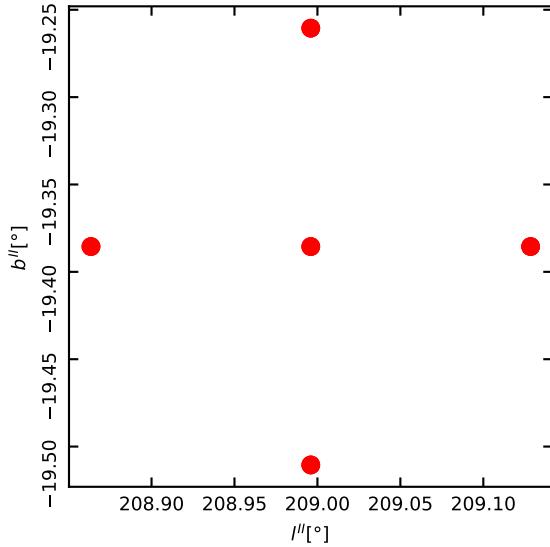


Figura 4: Cruz de observación

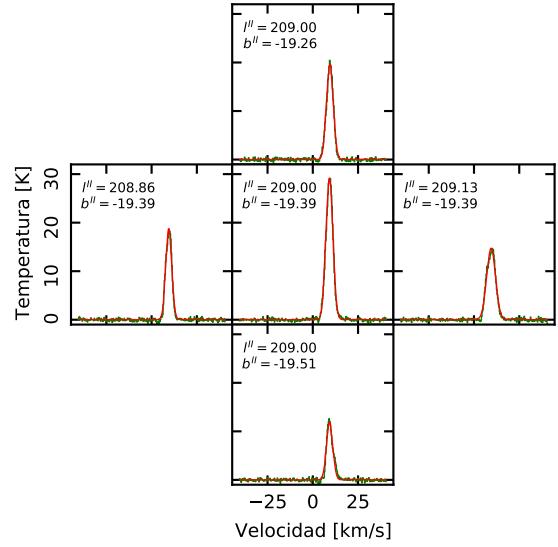


Figura 6: Gráfico del espectro y su ajuste gaussiano, segunda pasada por la cruz. Línea verde es espectro original. Línea roja es ajuste.

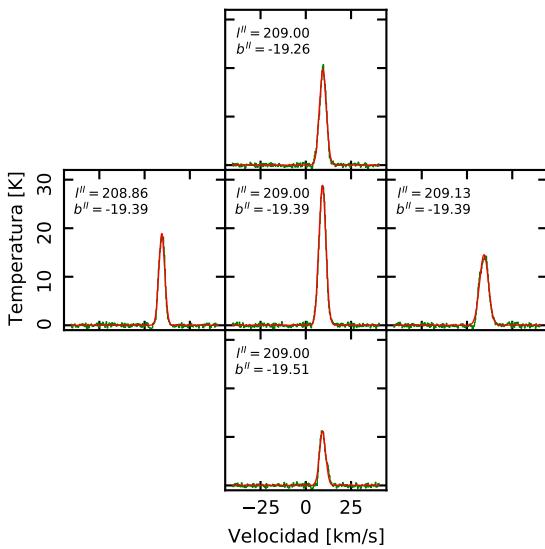


Figura 5: Gráfico del espectro y su ajuste gaussiano, primera pasada por la cruz. Línea verde es espectro original. Línea roja es ajuste.

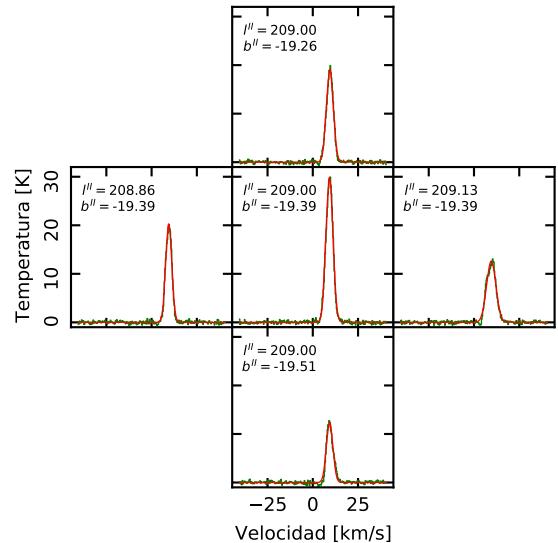


Figura 7: Gráfico del espectro y su ajuste gaussiano, tercera pasada por la cruz. Línea verde es espectro original. Línea roja es ajuste.

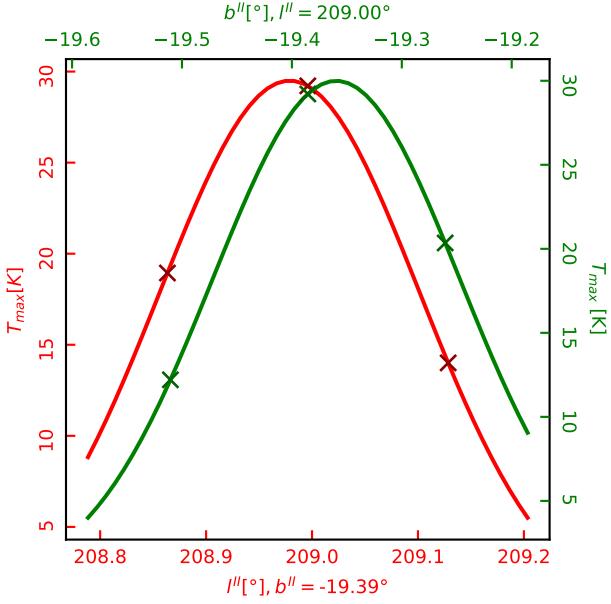


Figura 8: Gráfico de temperatura máxima y su ajuste gaussiano. Línea roja es ajuste de los puntos a latitud fija y se mide en los ejes inferior e izquierdo. Línea verde es ajuste de los puntos a longitud fija y se mide en los ejes superior y derecho. Equis roja es medición en la cruz, en orden creciente de longitud: punto de arriba, centro y abajo. Equis verde es medición en la cruz, en orden creciente de latitud: punto de izquierda, centro y derecha.

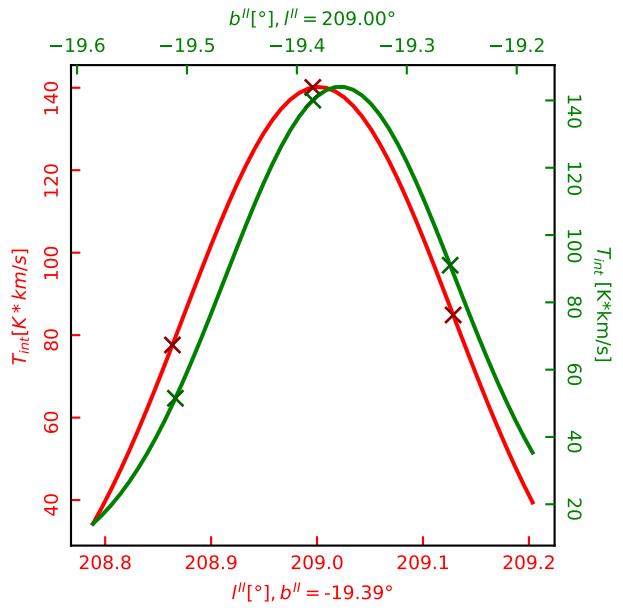


Figura 9: Gráfico de temperatura integrada y su ajuste gaussiano. Línea roja es ajuste de los puntos a latitud fija y se mide en los ejes inferior e izquierdo. Línea verde es ajuste de los puntos a longitud fija y se mide en los ejes superior y derecho. Equis roja es medición en la cruz, en orden creciente de longitud: punto de arriba, centro y abajo. Equis verde es medición en la cruz, en orden creciente de latitud: punto de izquierda, centro y derecha.

Pasada	Arriba	Izquierda	Centro	Derecha	Abajo
Primera	0.583	0.573	0.579	0.555	0.552
Segunda	0.621	0.601	0.640	0.628	0.593
Tercera	0.637	0.619	0.626	1.084	0.599

Cuadro 3: Cociente entre error de los puntos de la cruz promediados y error de los puntos sin promediar

#### 4.5. Error RMS y comparación

Sea  $\Delta T_A$  el error total para  $N$  muestras promediadas,  $T_{\text{sys}}$  el error para una única muestra,  $\tau$  el tiempo total,  $\Delta\nu$  el ancho de banda. La estadística gaussiana establece la siguiente relación entre el error para las muestras promediadas y el error para una única muestra,

$$\Delta T_A = \frac{T_{\text{sys}}}{\sqrt{N}} = \frac{T_{\text{sys}}}{\sqrt{\tau\Delta\nu}}, \quad (4.1)$$

pero en la práctica hay un factor  $C_{\text{obs}}$  que depende del modo de observación,

$$\Delta T_A = \frac{C_{\text{obs}}T_{\text{sys}}}{\sqrt{N}} = \frac{C_{\text{obs}}T_{\text{sys}}}{\sqrt{\tau\Delta\nu}}, \quad (4.2)$$

donde la constante observacional  $C_{\text{obs}}$  para el modo *position switching*, que apunta tanto a la fuente como fuera de esta, es igual a  $\sqrt{2}$ .

Por un lado, se utiliza la técnica de *sigma clipping* para obtener la base de ruido de los espectros y eliminar la línea de emisión para el siguiente cálculo, utilizando  $\sigma = 3$  y las iteraciones necesarias hasta que el algoritmo converja. Se calcula el error RMS del ruido para cada uno de los quince espectros.

Por otro lado, se median los espectros correspondientes a la misma coordenada en la cruz. Se les aplica la técnica de *sigma clipping* con los mismos parámetros anteriores. Se calcula el error RMS del ruido para cada uno de los cinco nuevos espectros.

Los resultados se muestran en la tabla 3, que tiene los valores  $\Delta T_A/T_{\text{sys}}$ .

La ecuación 4.1 y la cantidad de datos promediados establecen que el cuociente teórico entre el error del promedio y el error de una muestra única es  $1/\sqrt{3} = 0,577$ , pero si se considera la constante observacional entonces es  $\sqrt{2}/\sqrt{3} = 0,816$ .

atmósfera que se observa y, por lo tanto, su cantidad de emisión observada, es también mayor.

### 5. Conclusiones

Se concluye gracias a tabla 2 que, a menor elevación, mayor es la potencia, pues la cantidad de

## 6. Anexo

Código 1: Hot–Cold Test

```
1 #!/usr/bin/python3
2
3 def dBm2W(W):
4     """Converts an arbitrary power 'W' in dBm to W."""
5     return 10 ** ((W - 3) / 10)
6
7 def T_rec(T_hot, T_cold, y_factor):
8     """Calculates receiver noise temperature via hot & cold temperatures and y factor."""
9     return (T_hot - y_factor * T_cold) / (y_factor - 1)
10
11 def Y_factor(P_hot, P_cold):
12     """Y factor via hot & cold power in W."""
13     return P_hot / P_cold
14
15 T_hot = 300 # K
16 T_cold = 77 # K
17
18 W_hot = -44.5 # dBm
19 W_cold = -47.94 # dBm
20
21 Y = Y_factor(dBm2W(W_hot), dBm2W(W_cold))
22
23 print("Y = {}".format(Y))
24 print("T_rec = {} K".format(T_rec(T_hot, T_cold, Y)))
```

Código 2: Antenna Dipping

```
1 #!/usr/bin/python3
2
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6
7 df = pd.read_excel("../antdip_AE2021A.xlsx", engine='openpyxl')
8
9 _secz = df.iloc[:-1, 3]
10 lndP = df.iloc[:-1, 7]
11
12 p = np.poly1d(np.polyfit(_secz, lndP, 1))
13 print("tau_w = {}".format(p.coefficients[0]))
14 print("ln(deltaW) = -sec(z) * {} + {}".format(*p.coefficients))
15
16 plt.rcParams.update({'font.size': 7})
17 fig, ax = plt.subplots(figsize=(3.25, 3.25))
18
19 ax.plot(_secz, p(_secz), label="y = {1:.3f} + x * {0:.3f}".format(*p.c), color="red")
20 ax.plot(_secz, lndP, label="Datos", marker='x', fillstyle="none", linestyle="", color="green")
21 ax.legend()
22
23 ax.set_xlabel(r"$-\sec(\varphi)$")
24 ax.set_ylabel(r"$\ln(\Delta W)$")
25 ax.yaxis.set_tick_params(rotation=90)
26 ax.tick_params(direction="in", top=True, right=True)
27
28 fig.savefig("../informe/taufit.pdf")
29 plt.show()
```

Código 3: Observaciones

```
1 #!/usr/bin/python3
2
3 import glob
4 import itertools
5 import numpy as np
6 import scipy.optimize
```

```

7 import scipy.integrate
8 import astropy.stats
9 import matplotlib.pyplot as plt
10
11 paths = sorted(glob.glob("../sec_mierc_semi_2021/sdf*"))
12
13 lii = np.zeros(len(paths))
14 bii = np.zeros(len(paths))
15 v = np.zeros((len(paths), 364-109+1))
16 T = np.zeros((len(paths), 364-109+1))
17 for i, path in enumerate(paths):
18     with open(path) as f:
19         _, lii[i], _, bii[i] = np.genfromtxt(itertools.islice(f, 22, 24, None))
20         v[i], T[i] = np.genfromtxt(path, unpack=True, skip_header=108)
21
22 liirad = np.radians((lii - 180) % 360 - (360 - 180))
23 biirad = np.radians((bii - 180) % 360 - (360 - 180))
24
25 plt.rcParams.update({'font.size': 5})
26 fig = plt.figure(figsize=(3.25, 3.25/2))
27 ax = fig.add_subplot(111, projection="mollweide")
28 ax.grid(True)
29 ax.scatter(liirad, biirad, marker='.', color="red")
30 fig.savefig("../informe/lb.pdf")
31 plt.show()
32
33 plt.rcParams.update({'font.size': 7})
34 fig, ax = plt.subplots(figsize=(3.25, 3.25))
35 ax.scatter(lii, bii, color="red")
36 ax.set_xlabel(r"$l^{\wedge} \{II\} [\text{degree}]$")
37 ax.set_ylabel(r"$b^{\wedge} \{II\} [\text{degree}]$")
38 ax.yaxis.set_tick_params(rotation=90)
39 ax.tick_params(direction="in", top=True, right=True)
40 fig.savefig("../informe/cruz.pdf")
41 plt.show()
42
43 def gaussian(x, height, center, width):
44     return height * np.exp(-0.5 * ((x - center) / width) ** 2)
45
46 Tmax = np.zeros((5, 3))
47 for j in range(3):
48     fig, axs = plt.subplots(3, 3, sharex=True, sharey=True, figsize=(3.25, 3.25))
49     axs = axs.flatten()
50     for i in range(9):
51         axs[i].set_axis_off()
52         for i, n in enumerate([2, 4, 5, 6, 8]):
53             axs[n-1].set_axis_on()
54             axs[n-1].set_ylim((-1,32))
55             axs[n-1].yaxis.set_tick_params(rotation=90)
56             axs[n-1].tick_params(direction="in", top=True, right=True)
57             popt, pcov = scipy.optimize.curve_fit(gaussian, v[5*j+i], T[5*j+i], p0=[20, 10, 1])
58             axs[n-1].plot(v[5*j+i], T[5*j+i], c="green", drawstyle="steps-mid", ls="-", lw=0.5)
59             axs[n-1].plot(v[5*j+i], gaussian(v[5*j+i], *popt), c="red", ls="-", lw=0.5)
60             Tmax[i, j] = T[5*j+i].max()
61             plt.text(0.3, 0.85, r"\$l^{\wedge} \{II\} = \$" + f"{lii[5*j+i]:.2f}", fontsize=5,
62                     ha='center', va='center', transform=axs[n-1].transAxes)
63             plt.text(0.3, 0.75, r"\$b^{\wedge} \{II\} = \$" + f"{bii[5*j+i]:.2f}", fontsize=5,
64                     ha='center', va='center', transform=axs[n-1].transAxes)
65             axs[8-1].set_xlabel(r"Velocidad [km/s]")
66             axs[4-1].set_ylabel(r"Temperatura [K]")
67             plt.subplots_adjust(wspace=0, hspace=0)
68             fig.savefig("../informe/specfit{}.pdf".format(j+1))
69             plt.show()
70
71 Tmax = Tmax.mean(axis=1)
72 fig = plt.figure(figsize=(3.25, 3.25))
73
74 # https://doi.org/10.1109/MSP.2011.941846
75 p = np.poly1d(np.polyfit(lii[0:5][[1, 2, 3]], np.log(Tmax[[1, 2, 3]]), 2))
76 heightlii = np.exp(p.c[2]-p.c[1]*p.c[1]/(4*p.c[0]))
77 centerlii = -p.c[1]/(2*p.c[0])
78 widthlii = np.sqrt(-1/(2*p.c[0]))

```

```

79
80 xlii = np.linspace(lii.min()-0.075, lii.max()+0.075)
81 ylii = gaussian(xlii, heightlii, centerlii, widthlii)
82
83 axlii = fig.add_subplot(111, label=r"$l^{\{II\}}$")
84 axlii.plot(xlii, ylii, color="red")
85 axlii.plot(lii[0:5][[1, 2, 3]], Tmax[[1, 2, 3]], marker='x', color="darkred", linestyle="")
86 axlii.set_xlabel(r"$l^{\{II\}} [\text{degree}]", b^{\{II\}}=$"+f"\{bii[2]:.2f}"+"r"\degree", c="red")
87 axlii.set_ylabel(r"$T_{\{max\}} [K]", color="red")
88 axlii.tick_params(axis='x', colors="red")
89 axlii.tick_params(axis='y', colors="red")
90 axlii.yaxis.set_tick_params(rotation=90)
91 axlii.tick_params(direction="in")
92
93 p = np.poly1d(np.polyfit(bii[0:5][[0, 2, 4]], np.log(Tmax[[0, 2, 4]]), 2))
94 heightbii = np.exp(p.c[2]-p.c[1]*p.c[1]/(4*p.c[0]))
95 centerbii = -p.c[1]/(2*p.c[0])
96 widthbii = np.sqrt(-1/(2*p.c[0]))
97 xbii = np.linspace(bii.min()-0.075, bii.max()+0.075)
98 ybii = gaussian(xbii, heightbii, centerbii, widthbii)
99 axbii = fig.add_subplot(111, label=r"$b^{\{II\}}$", frame_on=False)
100 axbii.plot(xbii, ybii, color="green")
101 axbii.plot(bii[0:5][[0, 2, 4]], Tmax[[0, 2, 4]], marker='x', color="darkgreen", linestyle="")
102 axbii.xaxis.tick_top()
103 axbii.yaxis.tick_right()
104 axbii.set_xlabel(r"$b^{\{II\}} [\text{degree}]", l^{\{II\}}=$"+f"\{lii[2]:.2f}"+"r"\degree", c="green")
105 ylabelaxbii = axbii.set_ylabel(r"$T_{\{max\}} [K]", color="green")
106 ylabelaxbii.set_rotation(-90)
107 axbii.xaxis.set_label_position('top')
108 axbii.yaxis.set_label_position('right')
109 axbii.tick_params(axis='x', colors="green")
110 axbii.tick_params(axis='y', colors="green")
111 axbii.yaxis.set_tick_params(rotation=-90)
112 axbii.yaxis.set_label_coords(1.1175, 0.5)
113 axbii.tick_params(direction="in")
114
115 fig.savefig("../informe/tmax.pdf")
116 plt.show()
117
118 print("Data: Tmax = {}, lii = {}".format(Tmax[2], lii[2]))
119 print("Model: Tmax = {}, lii = {}".format(ylii.max(), xlii[np.argmax(ylii)]))
120 print("Data: Tmax = {}, bii = {}".format(Tmax[2], bii[2]))
121 print("Model: Tmax = {}, bii = {}".format(ybii.max(), xbii[np.argmax(ybii)]))
122
123 Tint = -scipy.integrate.simpson(T, v, axis=1).reshape((3, 5)).T.mean(axis=1)
124 fig = plt.figure(figsize=(3.25, 3.25))
125 p = np.poly1d(np.polyfit(lii[0:5][[1, 2, 3]], np.log(Tint[[1, 2, 3]]), 2))
126 heightlii = np.exp(p.c[2]-p.c[1]*p.c[1]/(4*p.c[0]))
127 centerlii = -p.c[1]/(2*p.c[0])
128 widthlii = np.sqrt(-1/(2*p.c[0]))
129 xlii = np.linspace(lii.min()-0.075, lii.max()+0.075)
130 ylii = gaussian(xlii, heightlii, centerlii, widthlii)
131 axlii = fig.add_subplot(111, label=r"$l^{\{II\}}$")
132 axlii.plot(xlii, ylii, color="red")
133 axlii.plot(lii[0:5][[1, 2, 3]], Tint[[1, 2, 3]], marker='x', color="darkred", linestyle="")
134 axlii.set_xlabel(r"$l^{\{II\}} [\text{degree}]", b^{\{II\}}=$"+f"\{bii[2]:.2f}"+"r"\degree", c="red")
135 axlii.set_ylabel(r"$T_{\{int\}} [K\text{km/s}]", color="red")
136 axlii.tick_params(axis='x', colors="red")
137 axlii.tick_params(axis='y', colors="red")
138 axlii.yaxis.set_tick_params(rotation=90)
139 axlii.tick_params(direction="in")
140 p = np.poly1d(np.polyfit(bii[0:5][[0, 2, 4]], np.log(Tint[[0, 2, 4]]), 2))
141 heightbii = np.exp(p.c[2]-p.c[1]*p.c[1]/(4*p.c[0]))
142 centerbii = -p.c[1]/(2*p.c[0])
143 widthbii = np.sqrt(-1/(2*p.c[0]))
144 xbii = np.linspace(bii.min()-0.075, bii.max()+0.075)
145 ybii = gaussian(xbii, heightbii, centerbii, widthbii)
146 axbii = fig.add_subplot(111, label=r"$b^{\{II\}}$", frame_on=False)
147 axbii.plot(xbii, ybii, color="green")
148 axbii.plot(bii[0:5][[0, 2, 4]], Tint[[0, 2, 4]], marker='x', color="darkgreen", linestyle="")
149 axbii.xaxis.tick_top()
150 axbii.yaxis.tick_right()

```

```

151 axbii.set_xlabel(r"$b^{\{II\}}~[\degree]$, $l^{\{II\}}$={lii[2]:.2f}" + r"$\degree$", c="green")
152 ylabelaxbii = axbii.set_ylabel(r"$T_{\{int\}}~[K\cdot km/s]$", color="green")
153 ylabelaxbii.set_rotation(-90)
154 axbii.xaxis.set_label_position('top')
155 axbii.yaxis.set_label_position('right')
156 axbii.tick_params(axis='x', colors="green")
157 axbii.tick_params(axis='y', colors="green")
158 axbii.yaxis.set_tick_params(rotation=90)
159 axbii.yaxis.set_label_coords(1.1175, 0.5)
160 axbii.tick_params(direction="in")
161 fig.savefig("../informe/tint.pdf")
162 plt.show()
163
164 masked = astropy.stats.sigma_clip(T, axis=1, sigma=3, maxiters=None)
165 rms = np.sqrt((masked**2).mean(axis=1)).reshape((3, 5)).T
166 meanmasked = astropy.stats.sigma_clip((T[0:5]+T[5:10]+T[10:15])/3, sigma=3, axis=1, maxiters=None)
167 rmsmean = np.sqrt((meanmasked**2).mean(axis=1))
168 print("deltaT_A/T_sys =", rmsmean.reshape((5, 1)) / rms)
169 print("1/sqrt(3) =", 1/np.sqrt(3))
170 print("sqrt(2/3) =", np.sqrt(2/3))

```