



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

PROCESAMIENTO DIGITAL DE SEÑALES
PARA LA DETECCIÓN DE
RADIO-PULSARES

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Ingeniero en Telecomunicaciones

PRESENTA:

Fabricio García Ramírez

DIRECTOR:

Dr. Stan E. Kurtz



Ciudad de México, 2017



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A la Facultad de Ingeniería y a la Universidad, por la formación que me han
dado.
Es gracias a ustedes que es posible el presente trabajo.
En verdad, gracias.
Yo.*

Reconocimientos

A mi familia por haberme apoyado durante todo este tiempo con su amor incondicional, confianza y sabios consejos. A mis padres Fabricio y Dora María por su increíble amor, paciencia y ayuda durante todos estos años de vida. A mis hermanas Alfonsina e Ivette por siempre estar conmigo y apoyarme en cada decisión, a mi querida sobrina Nadia por estar a mi lado.

A mi amigo y tutor Stan Kurtz, por su tiempo, sus consejos, su guía, su amistad y confianza a lo largo de la realización de este proyecto.

A mis sinodales por aceptar serlo y por el tiempo dedicado a la revisión de mi trabajo.

A mis amigos que durante todo este tiempo me han apoyado y por todos sus consejos.

Un agradecimiento especial al Ingeniero Ernesto Andrade y al M.C. Pablo Villanueva por el apoyo que recibí durante la realización de la tesis.

A la Dra. Aina Palau por su apoyo y al Instituto de Radioastronomía y Astrofísica de la Universidad Nacional Autónoma de México.

Agradezco a la DGAPA-UNAM la beca recibida.

Investigación realizada gracias al Programa de Apoyo de Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM, IA102815, Estudio de la formación de estrellas de alta masa y sus cúmulos asociados.

Finalmente un agradecimiento al IRyA, a los estudiantes y al personal por el apoyo durante la realización de la presente tesis.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Fabricio García Ramírez. Ciudad de México, 2017

Resumen

Ésta tesis se enfoca en el procesamiento de datos para la detección de radiopulsares. Los pulsares son estrellas de neutrones que emiten radiación por todo el espectro electromagnético. Por su naturaleza, dicha radiación se puede describir como una señal pulsante que posee un período único y una forma exclusiva para cada una de las fuentes.

Para comprender mejor este tipo de fenómenos en el Capítulo 1 se explica el nacimiento de estas estrellas de neutrones y su mecanismo de emisión. Se define la explicación del modelo faro (*lighthouse*).

Debido a las características especiales de las señales y que estas fuentes suelen ser muy débiles, se crearon métodos matemáticos y algoritmos para procesar datos captados por radiotelescopios. Con estos métodos podemos buscar dentro de una gran cantidad de datos aquellos que puedan o pertenezcan a los pulsares. Debido a que la señal sufre un retraso al atravesar el Medio Interstelar se crearon métodos para la resolución de este problema. Los principales se conocen como dispersión coherente e incoherente. Una vez resuelto el problema de la dispersión debida al medio se debe hacer una búsqueda de periodicidad, por lo que se opta por el método más conocido y usado que es la Transformada de Fourier. Aunque hasta este punto hemos podido resolver dos de los problemas que se presentan al hacer la búsqueda de pulsares aún no podemos dar por finalizado el proceso. El tercer y último paso y consiste en “regenerar” la señal usando una técnica que se conoce como Folding. Este método se describe a mayor detalle en Capítulo 2.

Para procesar los datos que se colectan en los radio-telescopios la comunidad astronómica ha creado programas que realizan los procesos necesarios descritos en el párrafo anterior. Los programas que se destacan para llevar a cabo ésta tarea en los datos son PSRSOFT y PRESTO, ambos escritos en diferentes lenguajes de programación como son Java, Python, C, C++, FORTRAN, entre otros. Estos dos software son de OpenSource lo que produce una ventaja al ser usados, pues se pueden llevar a cabo las modificaciones que uno considere pertinentes en la búsqueda de radiopulsares.

Los programas PSRSOFT y PRESTO usan datos que se encuentran en el formato PSRFITS, por esta razón hago uso de los datos que proporciona la ATNF (Australian Telescope National Facility) y que se encuentran disponibles para su descarga. Los archivos que se usan en esta tesis pertenecen a los pulsares J1456-6843, J1559-4438 y J0835-4510 que han sido elegidos debido a sus propiedades particulares que se muestran en el Capítulo 4.

El objetivo general de ésta tesis es describir las técnicas que se usan para la detección de radio-pulsares y el método empleado para realizar ésta tarea. También se demuestra el funcionamiento de estas técnicas al ser empleadas en dos software que han sido creados específicamente para la tarea de detección de pulsares a partir de datos obtenidos de observaciones reales.

Se desea que esta tesis sirva como guía para aquellas personas que estén interesadas en esta área de la astronomía.

Índice general

Índice de figuras	xii
Índice de tablas	xv
1. Introducción a los pulsares	1
1.1. Estrella de Neutrones	1
1.2. Púlsares	3
1.2.1. Mecanismo de emisión de un púlsar	3
1.3. Aplicaciones de los púlsares	8
2. Detección de púlsares	9
2.1. Radiotelescopio back-end	10
2.1.1. Detector	11
2.1.2. Conversión analógico/digital	11
2.2. Procesamiento de datos para la detección de pulsares	14
2.2.1. Dispersión	14
2.2.2. De-dispersión Incoherente	17
2.2.3. De-dispersión Coherente	18
2.2.4. Búsqueda de Periodicidad	19
2.2.5. Transformada Discreta de Fourier	20
2.3. Reconstrucción de la señal (Folding)	23
3. Uso de PSRSOFT y PRESTO para la búsqueda de pulsares	27
3.1. Método General	27
3.2. Tipos de formatos	29
3.2.1. PSRFITS	29
3.2.2. Filterbanking	31
3.3. Búsqueda de Púlsares con PSRSOFT	33
3.3.1. De-dispersión	33
3.3.2. Búsqueda de periodicidad y selección de candidatos	34
3.3.3. Doblamiento (Folding) y optimización de datos	37

ÍNDICE GENERAL

3.3.4. Resultados	39
3.4. Búsqueda de Púlsares con PRESTO	42
3.4.1. Búsqueda de Interferencia de Radio-Frecuencia (rffind)	42
3.4.2. De-dispersión	45
3.4.3. Transformada de Fourier	46
3.4.4. Búsqueda de periodicidad	47
3.4.5. Construcción del perfil del púlsar (Folding)	48
4. Búsqueda de Radio-púlsares con datos de la ATNF (Australian Telescope National Facility)	51
4.1. Selección de fuentes y características de los datos	52
4.1.1. Datos descargados de la ATNF	52
4.2. PSRSOFT resultados	56
4.2.1. PSR J1456-6843	56
4.2.2. PSR J1559-4438	59
4.2.3. PSR J0835-4510	62
4.3. PRESTO resultados	65
4.3.1. PSR J1456-6843	65
4.3.2. PSR J1559-4438	68
4.3.3. PSR J0835-4510	70
4.4. Evaluación de PSRSOFT y PRESTO	72
5. Conclusiones y trabajo a futuro	79
5.1. Conclusiones	79
5.2. Trabajo a Futuro	81
A. Códigos de Simulación de Folding	83
A.1. Código fuente escrito en MATLAB 2016	83
A.2. Código en Python	86
B. Instalación de PSRSOFT y PRESTO	91
B.1. PSRSOFT instalación	91
B.2. PRESTO INSTALACIÓN	93
C. Acrónimos	97
Bibliografía	101

Índice de figuras

1.1. Composición en capas de una estrella de neutrones. Imagen extraída de Salas and Solís 2006	3
1.2. Modelo de Faro de la emisión de un pulsar. Imagen extraída de Salas and Solís 2006	3
2.1. Radiotelescopios en México. (Figura 2.1 (a) Obtenida de http://wiki.lmtgtm.org/gtm/i Figura 2.1 (b) Obtenida de http://www.mexart.unam.mx/mexart.php)	9
2.2. Esquema básico de un radiotelescopio.	10
2.3. Curva característica I-V de un diodo. La zona A es donde se cumple la ley cuadrada. Imagen obtenida de http://www.ifent.org/lecciones/diodo/curva.asp	
2.4. Muestro periódico de una señal Analógica.	12
2.5. Cuantificador uniforme con compansor.	13
2.6. En la parte superior se encuentra un perfil reconstruido, la parte inferior muestra el efecto de dispersión causado por el MIE a diferentes frecuencias. El eje horizontal representa el tiempo, pero medido en unidades de fase del pulsar. Imagen obtenida de Lorimer and Kramer 2005	16
2.7. Efecto causado por la dispersión si se observará el púlsar en un solo canal. Imagen obtenida de Lorimer and Kramer 2005	17
2.8. Corrección del efecto de dispersión. Imagen obtenida de Lorimer and Kramer 2005	18
2.9. Representación esquemática de la de-dispersión coherente	19
2.10. Se muestra la gráfica de una señal periódica que simula un pulsar.	21
2.11. Señal periódica con ruido Gaussiano	22
2.12. Señal en función de la frecuencia	22
2.13. Simulación de un púlsar.	23
2.14. Al hacer el corte de la señal en los valores de corte Δ se ven patrones con esta forma, que serán promediados para reconstruir el pulso.	24
2.15. Representación del Folding con varias tramas representadas como $S_1, S_2, S_3 \dots S_n$ y promediadas para recuperar el pulso.	24

ÍNDICE DE FIGURAS

2.16. Perfil reconstruido con una $\Delta = 1(s)$ con tiempo de observación de 10(s) y SNR=4.	25
2.17. Perfil reconstruido con un tiempo de observación de 100 segundos y SNR=72.	26
3.1. Ejemplo del contenido del HDU primario en un archivo en formato PSRFITS.	31
3.2. En azul a la izquierda del esquema se encuentran los archivos generados durante el proceso de búsqueda de pulsares con PSRSOFT y a la derecha en rojo son los pasos que se realizan durante este proceso	33
3.3. Archivos generados por la dedispersión	34
3.4. Archivos generados por seek	35
3.5. Contenido del archivo .prd. Cada par de columnas de datos corresponde a un armónico de la señal el primer par corresponde a la fundamental, el segundo par al armónico 2 y así sucesivamente en potencia de dos.	36
3.6. Lista de los mejores candidatos obtenidos con PulsarHunter. La primera columna es el nombre del archivo, la segunda es la SNR, la tercer columna es el período y la columna que se encuentra al final es el rango de aceleración que en este caso es 0.	37
3.7. Imagen obtenida con la herramienta PGPLOT. En el lado superior izquierdo se encuentran los valores de las DM de prueba en función de la fase del pulso. Abajo del lado izquierdo una gráfica de Fase contra tiempo y a su lado de la Fase contra la frecuencia. En la parte inferior se encuentra la reconstrucción del pulso después de haber sumado los canales.	39
3.8. Imagen png obtenida a partir de PGPLOT. Del lado superior izquierdo se encuentra la intensidad de la señal en función del canal de frecuencia y la fase. Del lado superior derecho se encuentran los valores de las DM de prueba en función de la fase del pulso. En el centro del lado izquierdo se encuentra la intensidad de la señal en función de la sub-integración y de la fase. En el centro del lado derecho la relación S/N en función de la DM. En la parte final del lado izquierdo se encuentra el perfil del pulso reconstruido en función de su fase. Del lado inferior derecho se encuentran datos que corresponden al período, Ancho del pulso, SNR y a la DM	40
3.9. Esquema de procesamiento de datos con PRESTO	42

3.10. Se observa en el cuadro superior izquierdo una gráfica donde se encuentra la potencia máxima del intervalo de tiempo de los datos. Las gráficas que se presentan debajo nos sirven de referencia para saber si el tiempo que usamos es correcto o hay que disminuirlo para la búsqueda de RFI. Son gráficas del tiempo contra los canales. Aquellos canales en rojo tienen mucha interferencia y tendrán que ser eliminados esto es perjudicial para los siguientes procesos. Se recomienda disminuir el intervalo de tiempo para evitar eliminar demasiada información.	44
3.11. En este caso hemos disminuido el intervalo de tiempo	45
3.12. Datos en tiempo de un archivo .dat	46
3.13. Espectro de la señal generado con el comando <code>explorefft</code> de PRESTO.	47
3.14. Se muestra la lista de candidatos completa, hay que tener en cuenta que se debe explorar todo el archivo .txt generado con <code>ACCEL_sift.py</code> de manera manual	48
3.15. Output del comando <code>prepfold</code> en la esquina superior derecha se muestran los datos de la búsqueda. Al lado izquierdo de la información se muestra el perfil del pulso reconstruido. En este caso se observan dos picos; esto se debe a que la escala de la fase llega hasta 1.5 (figura que se encuentra del lado inferior izquierdo). La última parte de la imagen nos indica el cambio del período del pulsar.	49
4.1. Telescopio Parkes localizado en New South Wales, Australia. Posee una antena parabólica con diámetro de 64 metros. Imagen obtenida de https://www.atnf.csiro.au/outreach/visiting/parkes/index.html	51
4.2. Información principal del pulsar J1456-6843 del archivo SM003.0741.sf que contiene los datos necesarios para poder llevar a cabo el procesamiento de los datos. Entre los datos más importantes destaca el Telescopio, la posición, el tiempo de muestreo, frecuencia central, ancho de banda, número de canales y ancho por canal	53
4.3. Información principal del pulsar J1559-4438 del archivo S58110.1.sf que contiene los datos necesarios para poder llevar a cabo el procesamiento de los datos. Entre los datos más importantes destaca el Telescopio, la posición, el tiempo de muestreo, frecuencia central, ancho de banda, número de canales y ancho por canal	54
4.4. Información principal del pulsar J0835-4510 contenida en el archivo s1380127.180309.sf que contiene los datos necesarios para poder llevar a cabo el procesamiento de los datos. Entre los datos más importantes destaca el Telescopio, la posición, el tiempo de muestreo, frecuencia central, ancho de banda, número de canales y ancho por canal	55

ÍNDICE DE FIGURAS

4.5. Lista de los mejores candidatos encontrados en los datos de observación del pulsar J1456-6843. La lista ha sido recortado a los 10 mejores candidatos a pulsares.	56
4.6. En la Figura 3.16 observamos la integración del perfil del pulsar J1456-6843, que tiene una $DM=10.85$ y $P=263.40$ [ms].	58
4.7. En la Figura 3.17 observamos la integración del perfil del segundo candidato para J1456-6843 con $DM=72.38$ $P=263.09$ [ms]	59
4.8. Lista de candidatos para el pulsar J1559-4438. Se muestran los mejores diez candidatos.	60
4.9. Datos obtenidos del primer candidato de la lista para J1559-4438 que posee una $DM=55.35$ y un $P=257.04$ [ms].	61
4.10. Datos del segundo candidato de la lista para J1559-4438 que posee una $DM= 22.08$ y un $P=1799.06$ [ms].	62
4.11. Lista de los candidatos del pulsar Vela	63
4.12. Resultado del primer candidato de la lista del pulsar Vela que posee un Período= 89.3 [ms] y una $DM=68.67$	64
4.13. Resultado del segundo candidato de la lista del pulsar Vela que posee un Período= 179.8 [ms] y una $DM=47.15$	65
4.14. Lista de los candidatos con el archivo que se exploro.	66
4.15. Primer candidato del pulsar J1456-6843 $DM=5.0$ $P=263.41$ [ms]	67
4.16. Segundo candidato del pulsar J1456-6843 $DM=13.50$ $P=58.52$ [ms]	68
4.17. Lista de los candidatos con el archivo que se exploró.	69
4.18. Primer candidato para el pulsar J1559-4438 $DM=54.50$ $P=257.07$ [ms]	69
4.19. Segundo candidato del pulsar J1559-4438 $DM=59.50$ $P=73.44$	70
4.20. Lista de los candidatos mas prominentes para el pulsar Vela.	71
4.21. Primer candidato del pulsar Vela que posee una $DM=64.00$ y un $P=89.37$ [ms]	71
4.22. El segundo candidato de la lista posee una $DM=62.00$ $P=44.60$	72
4.23. Perfiles del pulsar J1456-6843 construidos con diferentes programas.	75
4.24. Perfiles del pulsar J1559-4438 construidos con diferentes programas.	76
4.25. Perfiles del pulsar J0835-4510 construidos con diferentes programas.	77

Índice de tablas

2.1. Representación binaria de los números 0 a 9 decimal en código Binario natural, BCD y Gray. Los códigos Binarios natural y BCD son diferentes para valores mayores a 9	14
2.2. Características de la señal	21
3.1. Características del formato PSRFITS	30
4.1. Características de las fuentes. S400 y S1400 son las densidades de flujo promedio de los pulsares observadas a 400[MHz] y 1400 [MHz] en mJy. Datos obtenidos del catálogo de pulsares de la ATNF disponible en línea	52
4.2. Tabla comparativa de los valores de Período obtenidos con PSRSOFT y PRESTO	73
4.3. Tabla comparativa de los valores de DM obtenidos con PSRSOFT y PRESTO	73
4.4. Tabla comparativa de los valores de SNR obtenidos con PSRSOFT y PRESTO	74

Introducción a los pulsares

En 1967, mientras se estudiaba el centelleo en radio de los astros, la estudiante de doctorado Jocelyn Bell descubrió una emisión pulsante entre sus datos. Se confirmó que no era un efecto instrumental, ni alguna forma de interferencia, sino fue el descubrimiento de los *pulsares*.

Un pulsar es una estrella de neutrones cuyo período de rotación va desde unos cuantos milisegundos hasta algunos segundos. Debido a su fuerte campo magnético no alineado con su eje de rotación, la estrella esencialmente es un dipolo magnético oscilante y conforme las leyes de la electrodinámica, ésta emite radiación.

Medio siglo después de su descubrimiento, los pulsares siguen siendo un área de gran interés en la astronomía. Además de ser objetos fascinantes por sí mismos, han resultado ser herramientas extremadamente útiles por sus propiedades especiales: pueden fungir como trazadores de distancia, campo magnético y densidades electrónicas dentro del Medio Interestelar (MIE) por lo que son usados como “sondas” en muchas ramas de la astronomía. Por sus propiedades tan extremas (campo magnético, campo gravitacional y momento angular), también se emplean como laboratorios para la física fundamental, ya que nos da la oportunidad de estudiar regímenes físicos no alcanzables en laboratorios terrestres.

Hasta la fecha los pulsares han resultado en dos premios Nobel y sin duda podrían resultar en varios más.

1.1. Estrella de Neutrones

Cuando las estrellas llegan a la última etapa de su vida tienen tres posibles finales; el camino que elijan depende de su masa. Estrellas con masa menor que $1.4M_{\odot}$ (el llamado “límite de Chandrasekhar”), terminan su vida como enanas

1. INTRODUCCIÓN A LOS PULSARES

blancas, en las cuales la fuerza de colapso gravitacional es contrarrestada por la degeneración electrónica dentro de la estrella.

Para estrellas con masa mayor que $1.4M_{\odot}$ (pero menor que 2 o $3M_{\odot}$ el llamado “límite de Landau-Oppenheimer-Volkoff” o LOV límite) la presión de electrones degenerados no es suficiente como para sostener el peso de la estrella y el colapso gravitacional sigue. Debido a la alta densidad, los electrones y protones comienzan a combinarse, lo que produce neutrones. El colapso se detiene debido a la degeneración de estos neutrones, formando lo que se llama “estrella de neutrones”.

Para estrellas más allá del límite de LOV, ni la degeneración de neutrones puede detener el colapso, y se forma un agujero negro. El enfoque de esta tesis entra en la categoría intermedia de las estrellas de neutrones.

Las estrellas de neutrones poseen radios del orden de 10[km], campos magnéticos muy intensos de hasta 10^{14} [gauss] (el campo magnético del Sol es de 50 [gauss] y el de la Tierra de 0.5 [gauss]) y frecuencia de rotación de hasta 600[Hz]. El campo magnético intenso y la rápida rotación de la estrella de neutrones son consecuencia directa del colapso de la estrella. Como se observa el momento angular, $L = I\omega$, y cuando la estrella colapsa el momento de inercia decremента por varios ordenes de magnitud, la velocidad angular tiene que aumentarse por el mismo factor para mantener el momento angular L constante. De una manera similar el campo magnético de la estrella después del colapso está confinado a una región mucho menor, y así la intensidad del campo aumenta de manera equitativa.

La estructura de las estrellas de neutrones se encuentra formada por tres capas y un núcleo como podemos ver en la Figura 1.1. En la superficie de la estrella, la llamada corteza exterior, coexisten núcleos atómicos y electrones. En esta capa, que es sólida, existen núcleos pesados de hierro, que están rodeados de un gas de electrones libres. El grosor de esta capa es de unos 0.3 [Km] y posee una densidad de 10^6 [g/cm] en su borde más externo y una densidad de 4×10^{11} [g/cm³] en su borde interior. Dentro de la corteza exterior se encuentra la corteza interior con un grosor aproximado de 0.5 [km]. En esta capa se lleva a cabo un proceso conocido como “goteo de neutrones”, que se debe a que los núcleos se han enriquecido tanto de neutrones que ya no pueden atar más, por lo que muchos de ellos quedan libres. La capa que rodea al núcleo posee un grosor de 1.2 [km] y una densidad que va desde 2×10^{14} [g/cm³] hasta 5×10^{14} [g/cm³], tiene una densidad tan alta que los núcleos se disuelven en neutrones y protones libres. En esta capa se forma un superfluido cuántico de neutrones con concentraciones más pequeñas de protones en estado superconductor (sin resistencia para conducir electricidad y que expulsa el campo magnético de su interior) el cual puede sostener corrientes eléctricas y campos magnéticos externos por períodos indefinidos. El núcleo interno tiene una densidad que supera la densidad nuclear ($> 10^{14}$ g/cm³) y donde es posible que existan partículas exóticas, como los quark o strange.

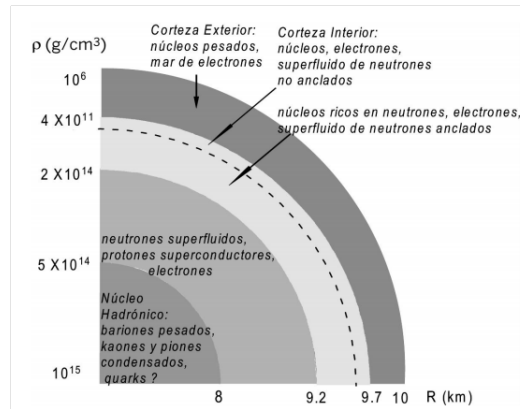


Figura 1.1: Composición en capas de una estrella de neutrones. Imagen extraída de [Salas and Solís 2006](#)

1.2. Púlsares

1.2.1. Mecanismo de emisión de un púlsar

El modelo simple para explicar los púlsares es el mostrado en la Figura 1.2 (modelo lighthouse o faro). En el modelo de faro el eje de rotación y el eje magnético no coinciden.

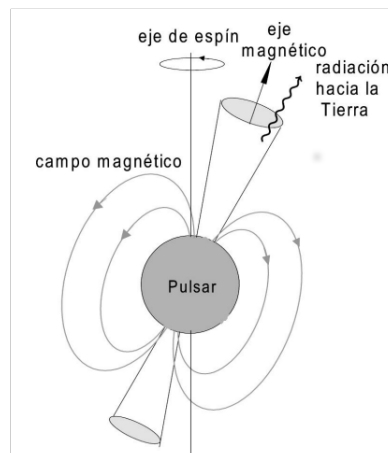


Figura 1.2: Modelo de Faro de la emisión de un púlsar. Imagen extraída de [Salas and Solís 2006](#)

1. INTRODUCCIÓN A LOS PULSARES

Los pulsares emiten radiación por todo el espectro electromagnético, desde el radio hasta los rayos gama. No hay un solo mecanismo físico que produzca esta emisión sino que es una combinación de varios mecanismos, cada uno dominante en rangos distintos del espectro. Aquí solo mencionamos un modelo muy simple — más heurístico que correcto — el de un dipolo magnético oscilante.

Como se observa en la Figura 1.2, el eje magnético y el eje de rotación se encuentran desalineados por un ángulo $\alpha > 0$. Se puede pensar en el pulsar como un dipolo magnético que varía en el tiempo con velocidad constante ω , calculando el momento del dipolo magnético como

$$m(t) = m \cos \alpha \hat{z} + m \sin \alpha [\cos(\omega t) \hat{x} + \sin(\omega t) \hat{y}]. \quad (1.1)$$

La potencia radiada por un dipolo magnético que varía en el tiempo se puede expresar como

$$P_{rad} = \frac{\mu_0 \ddot{m}^2}{6\pi c^3} \quad (1.2)$$

Aquí \ddot{m} es la magnitud de la segunda derivada del momento del dipolo (ecuación 1.1) y la ecuación está en unidades SI. Tomando la derivada de 1.1, obtenemos

$$\ddot{m} = \omega^2 m \sin \alpha (\cos^2 \omega t + \sin^2 \omega t)^{1/2}. \quad (1.3)$$

Sustituyendo 1.3 en 1.2 obtenemos

$$P_{rad} = \frac{\mu_0 \omega^4 m^2 \sin^2(\alpha)}{6\pi c^3}. \quad (1.4)$$

En nuestro modelo el pulsar es una esfera con un campo magnético dipolar, por lo que la potencia radiada que en la ecuación 1.4 aún nos hace falta sustituir el momento de la esfera cargada uniformemente que se puede describir como

$$m = \frac{4\pi B R^3}{\mu_0} \quad (1.5)$$

donde B es el campo magnético de la esfera y R es el radio.

Sustituyendo la ecuación 1.5 en 1.4 obtenemos la potencia radiada por un púlsar

$$P_{rad} = \frac{\mu_0 (4\pi)^2 \omega^4 B^2 R^6 \sin^2 \alpha}{6\mu_0^2 \pi c^3}. \quad (1.6)$$

En términos del período, la ecuación anterior se escribe como

$$P_{rad} = \frac{8\pi B^2 R^6 \sin^2 \alpha}{3\mu_0 c^3} \left(\frac{2\pi}{P} \right)^4 \quad (1.7)$$

La potencia radiada por el púlsar representa una pérdida de energía en ésta, y por lo tanto deben existir cambios en sus propiedades mecánicas. Para explorar estos cambios, analizamos la rotación del púlsar desde la perspectiva de la mecánica clásica. Empezamos con la energía cinética de rotación que se define como

$$E_{rot} = \frac{I\omega^2}{2} = \frac{2\pi^2 I}{P^2} \quad (1.8)$$

donde el momento de Inercia I se define

$$I = \frac{2MR^2}{5}. \quad (1.9)$$

Aclaremos que este valor para I es ligeramente grande, ya que el coeficiente 2/5 corresponde a una esfera uniforme. Debido a la estructura interna del púlsar, y el aumento de densidad hacia el centro, el coeficiente verdadero tendrá un valor menor que 0.4. Sustituyendo el momento de Inercia en la ecuación 1.8 la Energía se escribe como

$$E_{rot} = \frac{4\pi^2 MR^2}{5P^2} \quad (1.10)$$

Donde:

M es la masa del púlsar

R es el radio del púlsar

P es el período del púlsar

Ejemplo 1

Supongamos un pulsar que tiene radio $R=10$ [km] y campo magnético $B = 10^{11}$ [Tesla], con un ángulo $\alpha = .2^\circ$ y con período de rotación de 0.01[s]. Podemos calcular entonces la potencia radiada para este ejemplo, sustituyendo los valores en la ecuación 1.7

$$P_{rad} = \frac{8\pi(10^{11})^2(10^4)^6 \sin^2(.2^\circ)}{3(4\pi \times 10^{-7})(3 \times 10^8)^3} \left(\frac{2\pi}{0.1}\right)^4 = 4.608 \times 10^{33} [watts] \quad (1.11)$$

Que es el equivalente a 12 millones de veces la potencia del Sol.

Ejemplo 2

Para el segundo ejemplo, supondremos un pulsar igual que el anterior, pero con un período de $P=2$ [s] por lo que la potencia radiada será

$$P_{rad} = \frac{8\pi(10^{11})^2(10^4)^6 \sin^2(.2^\circ)}{3(4\pi \times 10^{-7})(3 \times 10^8)^3} \left(\frac{2\pi}{2}\right)^4 = 2.28837 \times 10^{20} [watts] \quad (1.12)$$

1. INTRODUCCIÓN A LOS PULSARES

La dependencia de P_{rad} al término $\left(\frac{2\pi}{P}\right)^4$ es el responsable de la enorme diferencia de orden 13 de magnitud entre los púlsares del ejemplo anterior.

La pérdida de energía por radiación implica una disminución en la velocidad angular, o sea un aumento en el período. A su vez un aumento en el período implica menos pérdidas de energía por radiación. Como consecuencia, la tasa de cambio del período va cambiando con el tiempo. O sea, P y \dot{P} pueden utilizarse como un reloj, para medir la edad del púlsar.

Para calibrar este reloj, empezamos con una expresión para el cambio en la energía cinética debido a la rotación como se aprecia en la ecuación 1.13

$$\dot{E} = \frac{-dE_{rot}}{dt} = \frac{-dI\omega^2/2}{dt} = -I\omega\dot{\omega}. \quad (1.13)$$

dado que $\omega = 2\pi/P$ podemos escribir la ecuación anterior en términos de período

$$\dot{E} = \dot{P}P^{-3}4\pi^2I \quad (1.14)$$

La \dot{E} se conoce como *spin-down luminosity* y representa la potencia emitida por el púlsar. \dot{P} es la tasa de decremento del período de rotación.

Igualando la ecuación 1.7 con 1.14 y sustituyendo el momento de Inercia (I) tenemos que:

$$\frac{2\dot{P}P^{-3}MR^24\pi^2}{5} = \frac{8\pi B^2R^6 \sin^2 \alpha}{3\mu_0c^3} \left(\frac{2\pi}{P}\right)^4 \quad (1.15)$$

De la ecuación anterior, despejamos la tasa de decremento del período

$$\dot{P} = \frac{20\pi^2 B^2 R^4 \sin^2 \alpha}{3PMc^3(10^{-7})} \quad (1.16)$$

La ecuación anterior solamente es valida para el espacio vacío

Suponiendo que el decremento en la velocidad angular del pulsar se debe a la radiación magnética del dipolo, se puede mostrar que $\dot{P} \propto P^{-1}$. Para demostrar esto se establece que la potencia radiada del pulsar es igual al decremento en la radiación (spin-down luminosity) es decir que $P_{rad} = \dot{E}$ por lo que reescribiendo la ecuación 1.16 obtenemos que

$$\dot{P}P = \frac{20\pi^2 B^2 R^4 \sin^2 \alpha}{3Mc^3(10^{-7})} = K \quad (1.17)$$

$$\therefore \dot{P} = KP^{-1} \quad (1.18)$$

suponiendo que la constante K no cambia con el tiempo podemos escribir una relación entre \dot{P} y P como:

$$PdP = P\dot{P}dt \quad (1.19)$$

integrando la ecuación anterior por ambos términos

$$\int_{P_0}^P P dP = \int_0^{\tau} dt \quad (1.20)$$

El límite P_0 es el período de nacimiento del pulsar, que es mucho menor al período actual y τ es la edad del pulsar, resolviendo las integrales anteriores tenemos que

$$\frac{P^2 - P_0^2}{2} = P\dot{P}\tau \quad (1.21)$$

dado que $P_0 \ll P$ podemos escribir la edad característica τ como

$$\tau_c = \frac{P}{2\dot{P}}. \quad (1.22)$$

Retomando los datos del ejemplo para el calculo de potencia radiada, para un púlsar con período $P=0.01[s]$ de la ecuación 1.16 tenemos que la tasa de decremento del período es

$$\dot{P} = \frac{20\pi^2(10^{11})^2(10^4)^4 \sin^2(.2^\circ)}{3(.01)(2.7846 \times 10^{30})(3 \times 10^8)^3(10^{-7})} = 1.065 \times 10^{-12} \quad (1.23)$$

y su edad característica es

$$\tau_c = \frac{.01}{2(1.065 \times 10^{-12})} = 4.691 \times 10^9[\text{segundos}] = 148[\text{years}] \quad (1.24)$$

Mientras que para un púlsar con período $P=2[s]$

$$\dot{P} = \frac{20\pi^2(10^{11})^2(10^4)^4 \sin^2(.2^\circ)}{3(2)(2.7846 \times 10^{30})(3 \times 10^8)^3(10^{-7})} = 5.3 \times 10^{-15} \quad (1.25)$$

y su edad característica

$$\tau_c = \frac{2}{2(1.065 \times 10^{-12})} = 1.875 \times 10^{14}[\text{segundos}] = 5.945[\text{My}] \quad (1.26)$$

Estos datos demuestran el patrón de evolución de los pulsares. Estos nacen con muy rápida rotación y períodos de 100 [ms] o menos. Con el tiempo — y la pérdida de energía por rotación— su velocidad angular va disminuyendo, hasta que tengan períodos de varios segundos cuando el pulsar ya tiene una edad de varios millones de años.

Existen poblaciones de pulsares que no se pueden explicar con este modelo. El ejemplo más importante son los llamados “pulsares de milisegundos”. Estos pulsares a pesar de poseer períodos muy rápidos (de 1 a 30 [ms]), son muy viejos. La explicación aceptada para demostrar esta contradicción es que estos pulsares han perdido gran parte de su campo magnético, y por la ec. 1.11, la radiación emitida es baja. Sin embargo, por la acreción del material de una estrella compañera, han aumentado su momento angular y por lo tanto su velocidad angular, con el resultado de períodos muy cortos.

1.3. Aplicaciones de los púlsares

Algunas aplicaciones de los pulsares dependen de la forma especial de su radiación. Un ejemplo es su uso como una red de relojes de alta precisión. Como el período de cada pulsar puede conocerse con muy alta precisión, y como están distribuidos por el plano galáctico, se puede usar un conjunto de pulsares (pulsar timing array) para detectar ondas gravitacionales y su efecto sobre el arreglo cuando estas pasen. El proyecto NANOgrav (<http://nanograv.org/>) es un intento actual para hacer esta detección.

Otro ejemplo, que depende del rango de radio frecuencias presentes en un pulso, es la medición de la dispersión de frecuencias del pulso. La magnitud de la dispersión depende de la densidad de electrones libres sobre la trayectoria del pulso, así que su medición permite al astrónomo crear una técnica para mapear la densidad del plasma inter-estelar de la Vía Láctea.

Otras aplicaciones de los pulsares —sobre todo de las leyes fundamentales de la física— dependen de configuraciones exóticas de pulsares. Un tipo que son particularmente fascinantes son los sistemas binarios que consisten en dos pulsares en órbita mutua, o bien un pulsar y una estrella (típicamente una estrella gigante) que están en órbita mutua. En el segundo caso, comúnmente la estrella secundaria está alimentando material hacia el pulsar por medio de un disco de acreción.

Tales sistemas ofrecen un laboratorio para el comportamiento de la materia en el límite de campos fuertes de la gravedad. El premio Nobel de Física en 1993 fue otorgado a Hulse y Taylor por el decaimiento de la órbita de un pulsar binario, según la teoría de la relatividad general.

El estudio de pulsares es un campo extenso, porque no se trata solamente de detectarlos o de estudiarlos intrínsecamente, sino que ofrece la posibilidad de que a través de su emisión se pueden estudiar otros fenómenos en el Universo y estudiar el entorno en el que nos encontramos.

Deteccción de púlsares

La radioastronomía estudia los objetos y fenómenos que ocurren en el espacio, midiendo la emisión electromagnética que emiten dichos objetos en el radio del espectro.

El radiotelescopio es la herramienta usada para realizar estudios radioastronómicos. Un radiotelescopio comúnmente consiste de una antena o conjunto de antenas; por ejemplo, en México tenemos el GTM y el MEXART (figuras 2.1(a) y 2.1(b) respectivamente). Para acompañar las antenas hay que tener sistemas electrónicos, como son: filtros, amplificadores de bajo ruido o LNA, mezcladores, osciladores locales y una base de registro de datos.



(a) *Gran Telescopio Milimétrico* (b) *MEXART* en Coeneo, Michoacán.
en Sierra Negra, Puebla.

Figura 2.1: Radiotelescopios en México. (Figura 2.1 (a) Obtenida de <http://wiki.lmtgtm.org/gtm/images/sitepicsInauguration/> Figura 2.1 (b) Obtenida de <http://www.mexart.unam.mx/mexart.php>)

2. DETECCIÓN DE PÚLSARES

Los componentes de los radiotelescopios suelen dividirse en dos grandes bloques: front-end y back-end. El primero es la parte de RF (radiofrecuencia) y FI (frecuencia intermedia), es decir toda la instrumentación desde la antena hasta el mezclador, mientras que el back-end son los componentes que procesan la señal previamente tratada por el front-end. La señal que llega a la antena es débil, comúnmente en el orden de $10^{-26} [W/m^2 Hz] = 1 [Jy]$, debido principalmente por la atenuación geométrica de la señal dado que la emisión isotrópica decae con la distancia recorrida al cuadrado. Por eso es necesario amplificar primero la señal, por lo que se agrega en el caso de observaciones a altas frecuencias un LNA (Low Noise Amplifier) o amplificador de bajo ruido. En algunos casos se coloca un filtro para restringir la banda de paso y evitar la interferencia de señales no deseadas. El mezclador que es donde termina la etapa de front-end no es imprescindible, porque depende de la frecuencia a la que observemos. Para las bajas frecuencias, digamos menores de 500 [MHz], puede quitarse. Pero para las altas frecuencias es preferible tener un mezclador que entregue al back-end una frecuencia intermedia (FI), con el objetivo de poder hacer un óptimo muestreo de la señal para su tratamiento.

2.1. Radiotelescopio back-end

El propósito del back-end es extraer información sobre la intensidad de la señal cósmica en función del tiempo, frecuencia y polarización. Para realizar estas funciones, el back-end típicamente cuenta con un detector de potencia, un convertidor analógico o digital, y un sistema para almacenar los datos, como se puede apreciar en la figura 2.2.

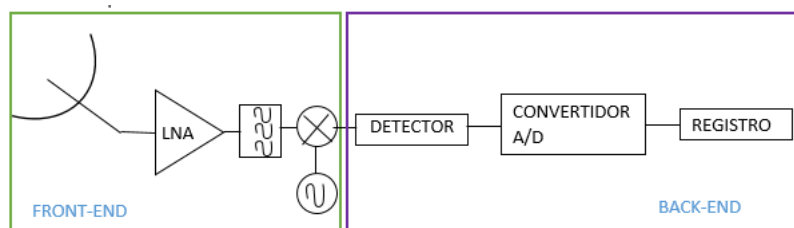


Figura 2.2: Esquema básico de un radiotelescopio.

El back-end suele clasificarse en tres tipos según la forma de tratar la señal:

- **Continúo:** estima la potencia recibida en toda la banda de paso del receptor.

- **Espectral:** estima la densidad espectral de potencia de la señal, dividiendo la banda de paso del receptor en canales adyacentes y midiendo la potencia recibida en cada banda.
- **Pulsar:** estima la densidad espectral de potencia de la señal con alta resolución temporal lo que genera una gran cantidad de datos, generalmente se opta por hacer un pre-procesamiento de la señal para disminuir el tamaño de los archivos.

2.1.1. Detector

Existen varios tipos de detectores, pero el más básico en radio astronomía es el tipo square-law o ley cuadrada. El detector de square-law es el encargado como su nombre lo dice de “detectar” la potencia de la señal que entra del front-end. Para poder hacer esto se utiliza un diodo que sea capaz de cumplir con la ley cuadrada, es decir que la respuesta del diodo sea proporcional al cuadrado del voltaje de entrada. Si se intenta medir la fuerza de la señal directamente de la salida del mezclador los ciclos negativos y positivos se anulan. Además, como la potencia es proporcional a V^2 , si la corriente en el diodo es proporcional al V^2 , nos da una indicación directa de la potencia. Esto se debe a que la curva I-V muestra que para bajas tensiones la curva sigue una forma parabólica, es decir $I \propto V^2$. Dado que la potencia de la señal es proporcional a V^2 , la corriente por el diodo nos da entonces la potencia, siempre que no se salga de la región parabólica de la curva ver la figura 2.3. El diodo de Schottky y el diodo Túnel son de los más usados en radioastronomía por su velocidad de conmutación y sensibilidad.

2.1.2. Conversión analógico/digital

Aunque para descubrir el primer púlsar en 1967 no fue necesario el uso de la conversión analógico/digital, hoy en día la mayor parte de datos se digitalizan. La digitalización de datos ha mejorado las técnicas de análisis de datos y ha favorecido al desarrollo de la astronomía. Por ejemplo en 1967 no se hubieran podido detectar señales débiles de fuentes cósmicas como se hace hoy en día. La conversión de una señal analógica a una digital consiste básicamente de tres pasos que son:

1. **Muestreo:** Consiste en convertir una señal continua en una señal discreta, tomando partes o “muestras” de la señal continua. Para poder hacer este muestreo se debe de respetar el teorema de Nyquist que dice: para reconstruir la señal continua con las muestras tomadas, la frecuencia de muestreo debe ser mayor que el doble de la frecuencia máxima de la señal, es decir: $f_s > 2f_{max}$ y tiene que

2. DETECCIÓN DE PÚLSARES

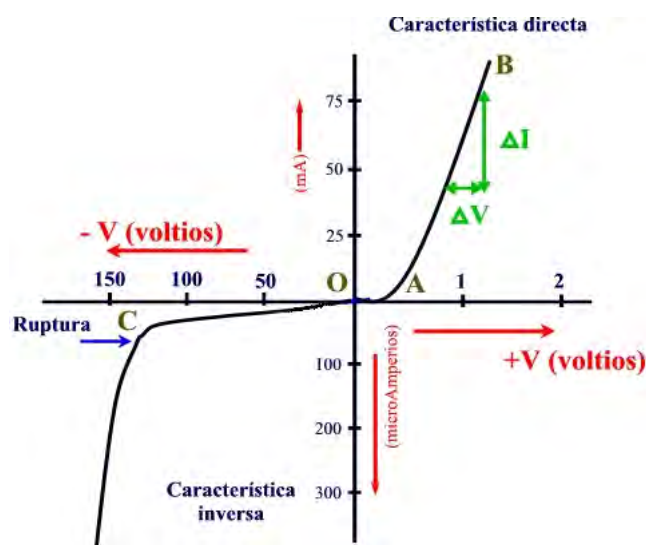


Figura 2.3: Curva característica I-V de un diodo. La zona A es donde se cumple la ley cuadrada. Imagen obtenida de <http://www.ifent.org/lecciones/diodo/curva.asp>

estar limitada en banda”. Este proceso es diferente de la cuantificación, pues es reversible.

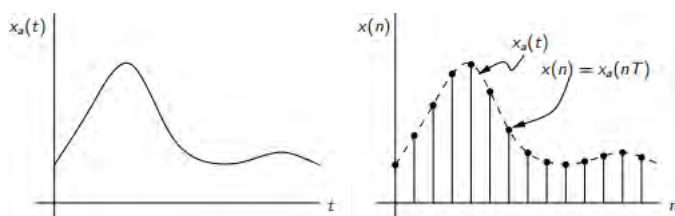


Figura 2.4: Muestro periódico de una señal Analógica.

2. **Cuantificación:** La cuantificación consiste en medir el nivel de tensión de las muestras de la señal analógica y asignarle valores de tensión discretos y finitos de amplitud. De tal manera que hay una correspondencia entre los valores de las muestras y los asignados. Si alguna muestra no alcanza y no se le puede asignar un nivel, entonces se designa al nivel inferior próximo. A la diferencia entre el valor continuo y su aproximación (cuantificación) se le conoce como error de cuantificación. La cuantificación puede ser de dos tipos:

Cuantificación uniforme: En esta cuantificación la distancia entre los **niveles de cuantificación**⁴ es siempre la misma. Una desventaja al usar este método

⁴Los niveles de cuantificación son el número de valores discretos que puede producir un

es que la relación S/N de las amplitudes pequeñas suele ser menor que para valores de amplitud mayores. Como consecuencia, si estas pequeñas amplitudes se repiten más que los valores promedios, el valor promedio de la relación S/N será muy chico.

Cuantificación no uniforme: A diferencia de la cuantificación uniforme las regiones cuantificadas pueden no ser de la misma longitud y no es necesario que los niveles sean equidistantes o se encuentren en puntos medios. Suele usarse la cuantificación no uniforme para señales homogéneas con valores de amplitud pequeños, como es la voz. Se basa en el uso de un cuantificador uniforme y un compansor¹, es decir un compresor a la entrada del cuantificador uniforme y un expansor a la salida del cuantificador ver figura 2.5.

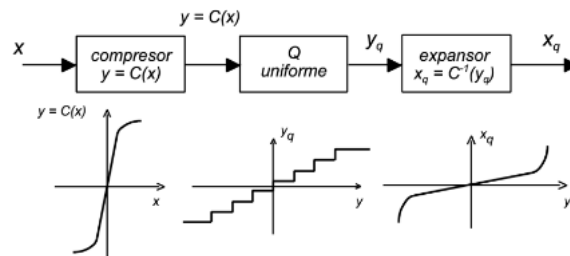


Figura 2.5: Cuantificador uniforme con compansor.

3. Codificación: Es el proceso de representar de manera biunívoca los estados de salida finitos del cuantificador por un símbolo elegido de un alfabeto finito. En electrónica y computación el alfabeto suele ser binario (1 y 0) aunque también son muy usadas las codificaciones en base 8 y base 16 (octal y hexadecimal).

Algunos códigos para hacer la representación binaria son los siguientes:

Código binario natural o directo: se dispone de dos bits o símbolos 0 y 1 con los que se pueden representar infinitas combinaciones numéricas.

Código BCD (Binary Coded Decimal): Los bits se agrupan de cuatro en cuatro, de manera que cada grupo representa una cifra de la cantidad expresada en código decimal.

Código Gray: es un código de distancia unidad y cíclico. En este código dos valores sucesivos difieren solamente en uno de sus dígitos. Es actualmente muy usado para facilitar la corrección de errores en los sistemas de comunicación.

convertidor A/D en el margen de valores analógicos que puede tomar la señal de entrada. Dado que generalmente son almacenados en formato binario, suelen expresarse en potencia de dos.

¹Compansor: es un dispositivo capaz de comprimir una señal y después recuperarla haciendo la expansión de dicha señal.

2. DETECCIÓN DE PÚLSARES

Decimal	Binario	BCD	Gray
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0011
3	0011	0011	0010
4	0100	0100	0110
5	0101	0101	0111
6	0110	0110	0101
7	0111	0111	0100
8	1000	1000	1100
9	1001	1001	1101

Tabla 2.1: Representación binaria de los números 0 a 9 decimal en código Binario natural, BCD y Gray. Los códigos Binarios natural y BCD son diferentes para valores mayores a 9

2.2. Procesamiento de datos para la detección de pulsares

Debido a las características intrínsecas del pulsar, es necesario implementar técnicas diferentes para el procesamiento de los datos que se generan en los observatorios, a continuación menciono algunas de las técnicas que se usan para detección de pulsares.

2.2.1. Dispersión

La dispersión es el fenómeno en donde la velocidad de una onda EM esta en función de su frecuencia. Suele suceder cuando una onda se propaga por un medio (no vacío) que tiene propiedades *dispersivas*. En el caso de las emisiones de los

pulsares, la señal se propaga por el *Medio Interestelar* (MIE) lo cual, debido a la presencia de electrones libres, es un medio dispersivo. Para poder calcular el efecto provocado por el MIE es necesario conocer la *Medida de Dispersión* (DM), que indica cuantos electrones libres la onda encuentra sobre su trayecto (en línea de vista) desde el pulsar hasta la Tierra. La DM se expresa como:

$$DM = \int_0^d n_e dl \quad (2.1)$$

de la ecuación anterior n_e es la densidad de electrones libres (que puede ser una función de posición), d es la distancia a la que se encuentra el pulsar de la Tierra y la integral se evalúa a lo largo de la línea de vista entre la Tierra y el pulsar. La medida de dispersión tiene como unidades $cm^{-3}pc$.

El retardo ocasionado por el MIE está en función de la frecuencia. Es decir que para frecuencias más altas el retardo es menor y viceversa, para frecuencias bajas el retardo es mayor. Para dos frecuencias distintas podemos calcular la diferencia en tiempo de su arribo será usando la ecuación 2.2.

$$\tau_{1,2} = \frac{d}{c} + \frac{e^2}{2\pi m_e c} \times \frac{1}{f^2} DM \quad (2.2)$$

$$\Delta t = 4.15 \times 10^6 \times (f_1^{-2} - f_2^{-2}) \times DM \quad (2.3)$$

donde f_1 y f_2 son frecuencias de observación diferentes en MHz y el valor 4.15×10^6 es la constante de dispersión \mathcal{D} que tiene unidades de $MHz^2 pc^{-1} cm^3 s$.

2. DETECCIÓN DE PÚLSARES

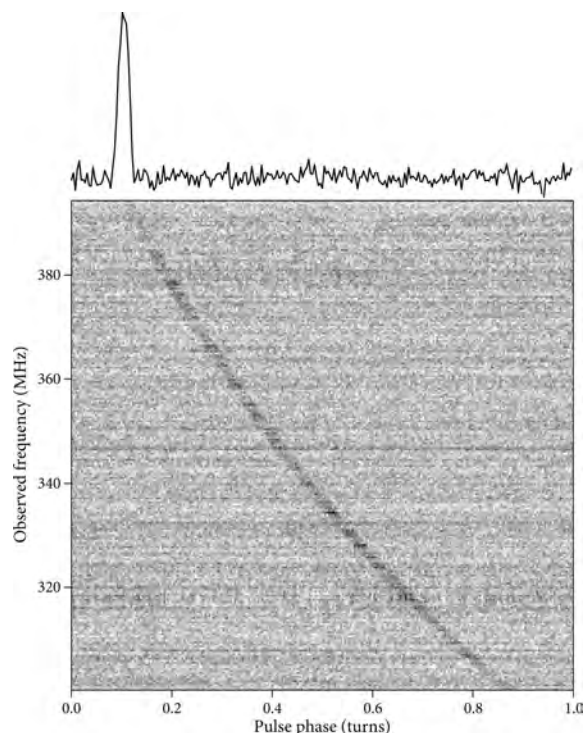


Figura 2.6: En la parte superior se encuentra un perfil reconstruido, la parte inferior muestra el efecto de dispersión causado por el MIE a diferentes frecuencias. El eje horizontal representa el tiempo, pero medido en unidades de fase del pulsar. Imagen obtenida de [Lorimer and Kramer 2005](#)

Para demostrar el efecto de retardo provocado por la dispersión, se puede ver en la figura 2.6 un cuadro en grises, en el que se encuentra el tiempo de arribo de un pulso como función de la frecuencia. La curva que se observa en un tono mas oscuro son las frecuencias que llegan con retardos. Si la dispersión causada por el MIE no existiera en lugar de una curva observaríamos una línea vertical. Es decir que no habría un retardo en la señal en función de la frecuencia, por lo que el trabajo de descubrir pulsares sería sencillo.

De-dispersión

Se han creado diferentes técnicas para contrarrestar el efecto de dispersión. Estos pueden ser englobadas en dos grupos:

1. Dedispersión incoherente: En el que la fase de la señal es ignorada.
2. Dedispersión coherente: En el que la relación fase-frecuencia es de suma importancia y debe preservarse.

A continuación veremos en mayor detalle estos dos grupos.

2.2.2. De-dispersión Incoherente

Para compensar los efectos de la dispersión causada por el MIE se divide la señal de entrada en múltiples canales (entre más canales mejor). Así que el ancho de banda Bw de un receptor se dividirá en n canales con ancho de banda Bw/n . A cada uno de los canales se le aplica una corrección Δt que está dado por la ecuación 2.3.

La ventaja de este método es que se puede aplicar de manera directa. Al aplicar la corrección a un canal rectificamos el retardo. Entre más canales tengamos, podemos hacer una corrección más fina. Pero existe un límite para el número de canales, por lo que hay una limitación para la finura de la corrección.

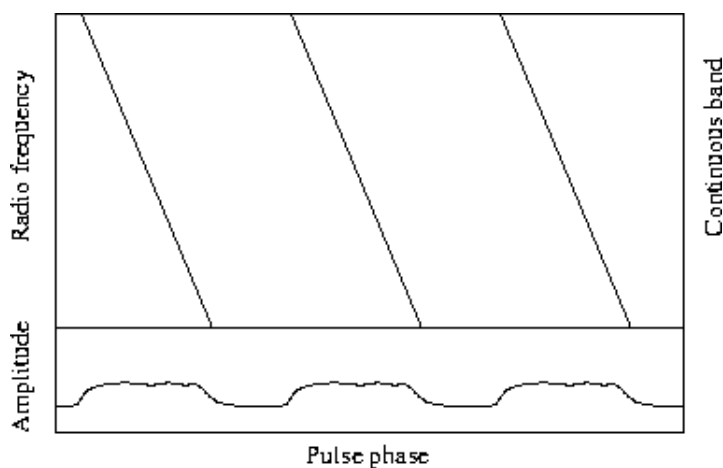


Figura 2.7: Efecto causado por la dispersión si se observará el púlsar en un solo canal. Imagen obtenida de [Lorimer and Kramer 2005](#)

En la figura 2.7 se muestra un gráfico que representa la señal del pulsar en la banda continua (antes de dividirla por canales). Al grabar estos datos su interpretación es de una señal “pulsante” con una duración ensanchada y una amplitud muy reducida.

2. DETECCIÓN DE PÚLSARES

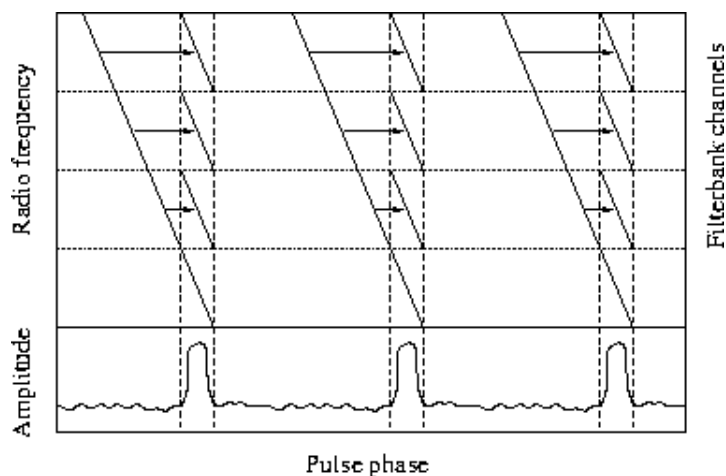


Figura 2.8: Corrección del efecto de dispersión. Imagen obtenida de [Lorimer and Kramer 2005](#)

Una vez dividida la señal en canales de frecuencia y hecha la compensación en tiempo de la señal, se alinean y se suman las señales de los canales como se muestra en la figura 2.8. Esto genera un perfil del pulso que se observa al final de la figura. El perfil obtenido al hacer la corrección en tiempo difiere del pulso ensanchado que observamos en la figura 2.7, en su forma y en su amplitud.

La de-dispersión incoherente tiene la ventaja de su simplicidad y que no requiere de grandes recursos para ser implementada. No obstante es un método bastante engorroso, pues se deben generar “muestras” para valores diferentes de DM (si se trata de un pulsar nuevo) y la resolución en tiempo es baja, por lo que su uso generalmente se limita solamente a la detección de nuevos pulsares.

2.2.3. De-dispersión Coherente

De-dispersión coherente es un proceso que toma en cuenta la fase de la señal. Es conceptualmente más complicado que la de-dispersión incoherente descrito anteriormente, pero es muy superior en sus resultados. La técnica fue desarrollada por Hankins & Ricketts (1975).

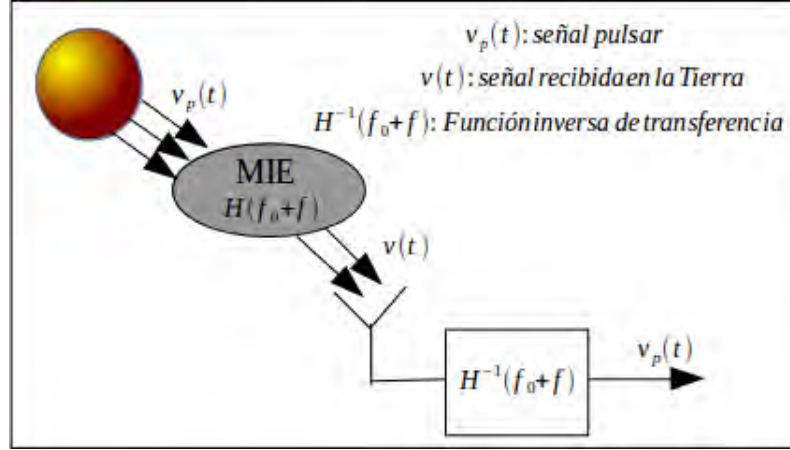


Figura 2.9: Representación esquemática de la de-dispersión coherente

Como se muestra en la figura 2.9, para poder recuperar la señal v_p del pulsar es necesario hacer la convolución de la señal recibida $v(t)$ con la función de transferencia inversa del MIE H^{-1} .

El efecto de dispersión causado por el MIE en la señal del pulsar se puede describir como un “filtro de fase”, capaz de producir retardos en la señal en función de la frecuencia. La función de transferencia del MIE puede escribirse como

$$H(f_0 + f) = e^{\frac{i2\pi DM f^2}{2.41 \times 10^{-10} (f+f_0) f_0^2}} \quad (2.4)$$

donde f_0 es la frecuencia de referencia, f es la frecuencia de observación, DM es la medida de Dispersión y d es la distancia entre el pulsar y el observador.

El usar este método para mitigar la dispersión es superior al método incoherente, porque tiene la ventaja de remover completamente el efecto de la dispersión independientemente del ancho de los canales. Sin embargo su implementación puede ser costosa debido a que se requieren procesadores independientes para hacer la Transformada en Frecuencia de la señal y un gran número de bancos de filtros (pueden ser digitales o analógicos) para los valores de DM necesarios en la ecuación 9. Por lo tanto para la búsqueda de pulsares no suele usarse este método.

2.2.4. Búsqueda de Periodicidad

Dado que los pulsares son periódicos, es decir emiten pulsos regularmente con periodos bien definidos, podemos usar algunos métodos para buscar periodicidad

2. DETECCIÓN DE PÚLSARES

en las señales como son: Transformada Wavelet, Transformada Z, Autocorrelación, Transformada de Fourier y Folding. Para la tesis, nos enfocaremos en las últimas dos.

2.2.5. Transformada Discreta de Fourier

La Transformada de Fourier consiste en convertir una señal del dominio del tiempo al dominio de la frecuencia y está dada por la ecuación

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (2.5)$$

donde $x(t)$ es un función continua en el dominio del tiempo. Esta expresión nos permite determinar el contenido de una señal temporal en el dominio de la frecuencia.

Al digitalizar la señal convertimos una señal continua a una señal discreta, por lo que la ecuación anterior no nos es útil para el análisis en frecuencia. Por ende recurrimos al uso de la Transformada Discreta de Fourier (TDF), que consiste en pasar una señal discreta en tiempo al espacio Discreto en Frecuencia. La ecuación para la TDF es

$$X(m) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nm/N} \quad (2.6)$$

donde: $x(n)$ es la representación discreta de una señal continua $x(t)$. N es el número de muestras de la secuencia de entrada, mientras que n indica el número de la muestra de entrada en el dominio del tiempo y va desde 0 hasta $N-1$. La m indica el número de la componente de Fourier en el dominio de la frecuencia y j es $\sqrt{-1}$. Implementar directamente la DFT requiere muchos recursos y tiempo de procesamiento, por lo que en análisis de señales se recurre a la Transformada Rápida de Fourier (FFT) que reduce el número de operaciones a $N \log_2(N)$.

A continuación simularé ⁹ un pulsar con las siguientes características

⁹El código se encuentra en el Apéndice A

Características	Valor
Período	1[s]
Ancho del pulso	5 % del ciclo de trabajo
Tiempo de observación	100[s] (variable)
Frecuencia de muestreo	1[KHz]

Tabla 2.2: Características de la señal

Señal pulsante

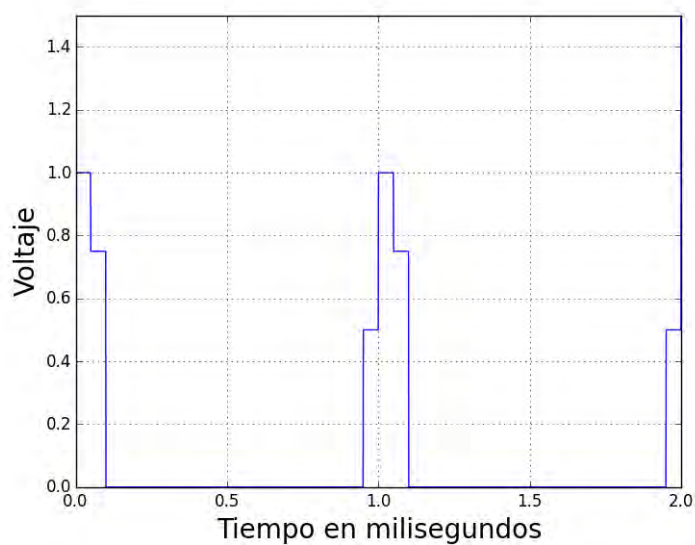


Figura 2.10: Se muestra la gráfica de una señal periódica que simula un pulsar.

2. DETECCIÓN DE PÚLSARES

Agregamos ruido a la señal original

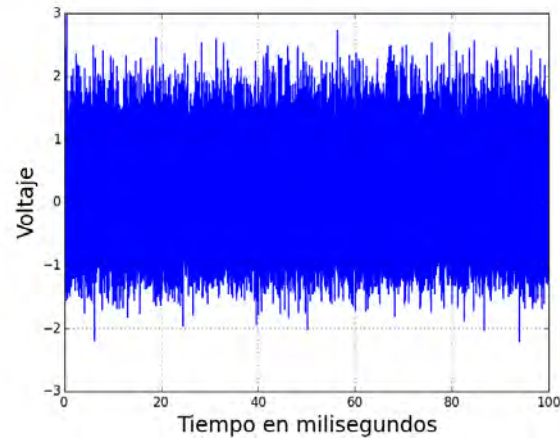


Figura 2.11: Señal periódica con ruido Gaussiano

Convertimos la señal del dominio de tiempo al dominio de la frecuencia por medio de la TDF. El resultado de este proceso se observa en la figura 2.12.

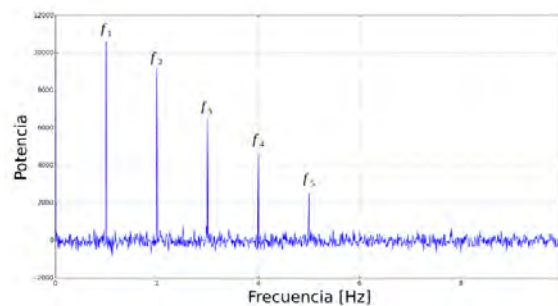


Figura 2.12: Señal en función de la frecuencia

En la figura 2.12, observamos cinco espigas que sobresalen de las demás, se encuentran antes de los 50 [Hz] y las he nombrado f_1 , f_2 , f_3 , f_4 y f_5 . Si hacemos un zoom a cada una de las espigas podemos saber su frecuencia. De esta forma sabemos que las frecuencias son: $f_1 = 1[Hz]$, $f_2 = 2[Hz]$, $f_3 = 3[Hz]$, $f_4 = 4[Hz]$ y $f_5 = 5[Hz]$. De los datos de la señal y los obtenidos por la TDF nos damos cuenta que la espiga 1 corresponde a la señal del pulsar y que las demás espigas son los armónicos 2, 3, 4 y 5 de la señal.

Usamos la TDF para períodos desconocidos porque requiere un número de operaciones menor que Folding. Sin embargo si observamos la figura 2.12 se tiene

poca información de la señal: no sabemos la forma que tiene el pulso, ni su ciclo de trabajo. Para poder obtener mayor información de la señal, es necesario complementar Fourier con otra técnica; comúnmente se hace uso de Folding.

2.3. Reconstrucción de la señal (Folding)

Folding es usado para reconstruir la señal una vez que ya se conoce el período. Ésta técnica consiste en cortar una señal en tramos de tiempo Δ y sumarlos. Por ejemplo si un tramo va de 0-5 segundos el siguiente tiene que ser de 5-10 segundos, así sucesivamente; luego esos tramos se alinean y suman. Este procedimiento se lleva a cabo hasta terminar con todos los datos.

Lo que se pretende con la implementación de Folding es mejorar la relación señal a ruido (SNR), ya que al hacer la suma de las tramas, el ruido se contrarresta consigo mismo mientras que la señal se refuerza. Esto sucederá cuando los pulsos estén en fase y el valor Δ de la trama sea el mismo que el periodo de la señal o bien un múltiplo del período. En caso de que los pulsos de cada trama se encuentren desfasados o que Δ posea otro valor diferente del periodo, se puede producir un efecto de ensanchamiento en tiempo del pulso o solamente observarse ruido; esto depende de la SNR con la que se esté recibiendo la señal y el grado de desfasamiento o el error en Δ .

Suponga que tiene una señal con período $P=1[s]$ con una $SNR=10$ (en la práctica el valor de SNR es mucho menor que 10, pero para el ejemplo es útil). El tiempo de grabación son diez segundos y el ciclo de trabajo (CT) es del 5% (se utiliza este valor de CT porque se ha observado en gran parte de la población de pulsares). Observaremos entonces una señal como en la Figura 2.13

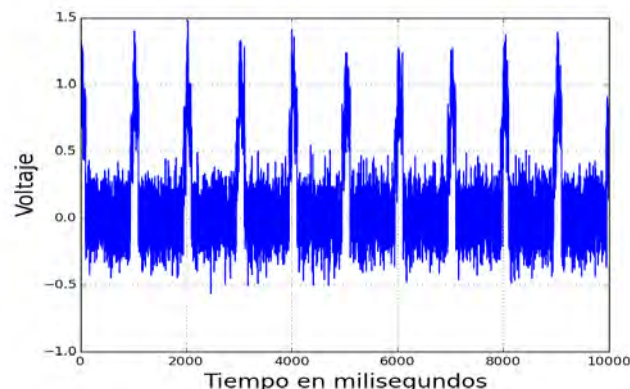


Figura 2.13: Simulación de un púlsar.

2. DETECCIÓN DE PÚLSARES

A continuación la señal se divide en tramas del mismo tamaño, en este caso serán de $\Delta=1$ [s] porque ya conocemos el periodo del pulsar.

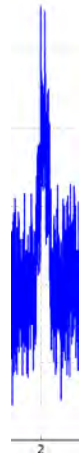


Figura 2.14: Al hacer el corte de la señal en los valores de corte Δ se ven patrones con esta forma, que serán promediados para reconstruir el pulso.

Se deben 'alinear' las tramas para evitar el desfase cuando se recupere la señal. Una vez alineados los datos o tramas, procedemos a hacer el promedio, lo que sirve para reconstruir el pulso.

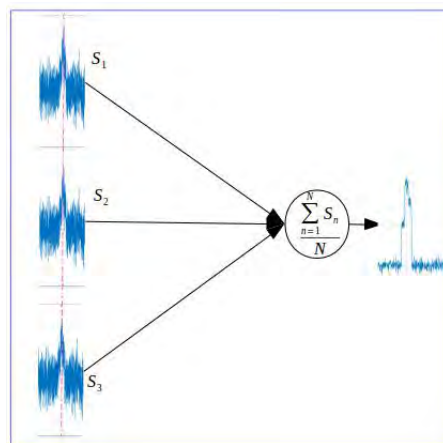


Figura 2.15: Representación del Folding con varias tramas representadas como $S_1, S_2, S_3 \dots S_n$ y promediadas para recuperar el pulso.

Simulación de Folding⁶

En la siguiente simulación muestro el funcionamiento de la técnica de Folding. Tomando de nuevo los datos de la Tabla 2.2 podemos reconstruir el perfil del pulsar que es el mostrado en la Figura 2.10. Al agregar el ruido a la señal se observa la Figura 2.11.

Una vez agregado el ruido, los pulsos no son evidentes como se puede apreciar en la figura 2.10. Por lo tanto no podemos decir cual es la forma de la señal. Entonces lo que se hace es usar un valor de corte $\Delta = P$ para reconstruir el perfil del pulsar, en este caso $\Delta=1[s]$.

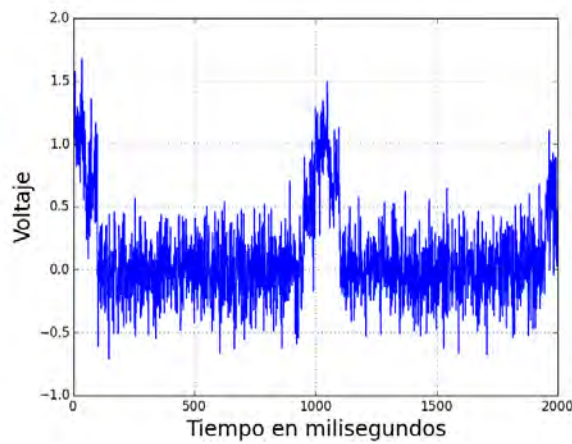


Figura 2.16: Perfil reconstruido con una $\Delta = 1(s)$ con tiempo de observación de 10(s) y SNR=4.

En la figura 2.16 el perfil reconstruido aún tiene mucho ruido (SNR=4), por lo que generalmente se amplía el tiempo de observación para mejorar la relación señal a ruido. Incrementando el tiempo de observación a 100[s] obtenemos el perfil que se observa en la figura 2.17, ahora el valor de la SNR es 72.

⁶El código de la simulación se encuentra en el apéndice A.

2. DETECCIÓN DE PÚLSARES

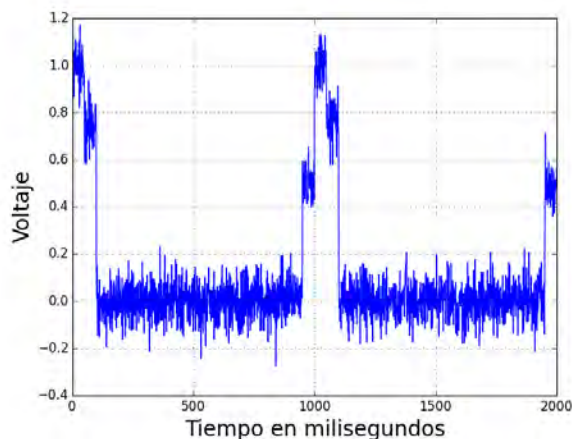


Figura 2.17: Perfil reconstruido con un tiempo de observación de 100 segundos y SNR=72.

Sí seguimos incrementando el tiempo de observación la SNR incrementa, por eso observamos que la señal se ve más “limpia”.

Folding en un principio fue concebido como una técnica para la búsqueda de periodicidad en las señales, pero aunque es un método muy sencillo se complica para periodos desconocidos. Para poder recuperar una señal de la que desconocemos su período usando folding, es necesario adivinar el valor de corte Δ tal que sea igual al período, lo que conlleva un gran tiempo de procesamiento y hace que este método sea ineficiente.

Otro problema al usar Folding es que si no conocemos el período con precisión e incrementamos el tiempo de observación (entre más tiempo observemos más pulsos tenemos) con la intención de mejorar la SNR esperando observar un pulso, estaremos provocando un desfaseamiento en la señal cada vez mayor que no permitirá reconstruir el pulso. Esta relación está dada por la ecuación

$$\frac{\Delta P}{P} = \frac{1}{N_p} \frac{\Delta \phi}{\phi} \quad (15) \quad (2.7)$$

donde P es el período, N_p es el número de pulsos, ϕ es la fase y $\Delta \phi$ el desfaseamiento de la señal.

Como hemos visto hasta este punto la técnica de Folding es sencilla pero ineficiente dado que entre mayor precisión exija la señal, el número de operaciones incrementa de manera cuadrada. Sin embargo en 1969 Staelin et al. crearon un algoritmo que reducía el número de operaciones y optimizaba el tiempo de procesamiento, el cuál es conocido como Fast Folding Algorithm (Staelin 1969).

Uso de PSRSOFT y PRESTO para la búsqueda de pulsares

En este capítulo, primero miraremos las consideraciones generales para la búsqueda de pulsares (sección 3.1), luego muestro el procedimiento de búsqueda de pulsares con PSRSOFT (sección 3.3) y finalizo con el procedimiento usado para PRESTO (sección 3.4)¹.

PSRSOFT es un programa maestro creado por Michael J. Keith. Este programa instala, compila y ejecuta una serie de programas como son: SIGPROC, PULSARHUNTER, TEMPO, DSPSR, PSRCHIVE, entre otros, que son de orígenes distintos.

PRESTO es un programa que fue creado por Scott Ransom. Sirve para la búsqueda y análisis de pulsares. Está escrito principalmente en ANSI C, FORTRAN y Python.

3.1. Método General

El método general para búsqueda de pulsares consiste en:

1. Extracción de datos
2. Búsqueda de la DM
3. Búsqueda de periodo (Transformada de Fourier)
4. Doblamiento de datos (Folding)

¹La instalación de PSRSOFT y PRESTO se puede ver en el Apéndice B de la tesis.

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

5. Inspección de los datos seleccionados

Dependiendo del software que se use, los pasos del método general pueden ser modificados. Como veremos más adelante algunos de estos pasos pueden ser omitidos y se pueden agregar más tareas al método general de ser necesario.

Extracción de datos

Antes de empezar el procesamiento de los datos es necesario convertirlos en un formato compatible con los paquetes del software. Existen muchos instrumentos para la adquisición de datos y cada observatorio usa el que más le convenga según su configuración de back-end. Por esta razón se han creado estándares para guardar los datos que son independientes del back-end que se posea, entre ellos destaca: PuMa, EPN y PSRFITS; este último es el que trataremos en la tesis.

De-dispersión

El primer paso para la búsqueda de pulsares consiste en de-dispersar la señal. A partir del archivo original, se crean otros archivos o series de tiempo (*“time series data”*) con valores de DM diferentes para realizar un ajuste de la dispersión causada por el MIE. Para la detección de pulsares suele usarse la de-dispersión incoherente debido a su fácil implementación y bajo consumo de recursos. Existen diferentes programas para realizar esta tarea como se mostrará más adelante.

Búsqueda de período

Una vez de-dispersada la señal y creados los archivos nuevos se usa la Transformada de Fourier para encontrar en ellos señales periódicas. Debido a que en algunos casos los pulsares no se pueden detectar simplemente con el uso de la Transformada de Fourier es necesario hacer la suma de armónicos después de realizar la FFT.

La FFT (y en su caso la suma de armónicos) se realiza sobre cada archivo, cada uno de-dispersado con un valor distinto de DM. Para cada archivo creado con diferente DM se identifica una lista de candidatos a pulsares, basados en el mayor valor de SNR para uno o más períodos. De tal forma, quedamos con varios candidatos a pulsares con valores distintos para su DM y su período.

Reconstrucción del pulso (Folding)

Ya que hemos logrado encontrar algunas señales periódicas en nuestros datos el siguiente objetivo es reconstruir el perfil del pulso, por lo que recurrimos a hacer Folding o doblamiento de la señal. Este proceso se repite para cada uno de los candidatos identificados en el paso anterior.

Inspección de los datos seleccionados.

Hasta este punto el proceso podríamos considerarlo casi automático. Pero ahora tenemos nosotros que hacer la observación de los perfiles que hemos obtenido anteriormente y determinar por sus características cual de ellos es el más cercano a un pulsar. Una forma de inspeccionarlos es por el valor de su SNR, sin embargo algunos de los candidatos pueden poseer una buena SNR pero tener un perfil poco favorable a ser un pulsar.

3.2. Tipos de formatos

3.2.1. PSRFITS

Cada radiotelescopio tiene un back-end diferente y en muchos casos estos back-ends realizan un pre-procesamiento de los datos antes de que estos sean guardados en el disco. Este pre-procesamiento puede incluir de-dispersión, transformada de fourier y doblamiento — todo hecho en hardware— entre otras posibilidades, este procesamiento sirve para reducir el volumen de datos para guardar y reducir el procesamiento posterior de los datos. Es conveniente estandarizar en un solo formato los datos para facilitar el intercambio de ellos sin importar la forma en que salgan del back-end.

Para lograr este objetivo se creó un formato llamado PSRFITS que es una derivación del formato FITS (*Flexible Image Transport System*) usado en astronomía. A diferencia de FITS que puede ser usado para guardar datos de diferentes fenómenos astronómicos, PSRFITS es usado exclusivamente para guardar y procesar datos de púlsares. El formato PSRFITS consiste en un encabezado primario (HDU o Header Data Unit) seguido de una serie de extensiones del HDU. En la tabla 3.1 se muestran los HDU de PSRFITS.

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

Nombre del HDU	Descripción
Header principal	Información sobre la observación, como es el telescopio usado, el receptor, el nombre de la fuente, fecha de la observación y tiempo de observación.
History	Una lista de todo lo que ha pasado con los datos desde que fueron grabados.
OBSDESCR	Descripción de la observación o procesamiento de la señal en formato ASCII
PSRPARAM	Efemérides del pulsar, usadas para crear o modificar los datos del perfil del pulsar
POLYCO	Elementos del documento polyco creado con TEMPO, usado para predecir el período del pulsar
T2PREDICT	Archivo que sirve para predecir el período del pulsar, usando TEMPO2 %
COHDDISP	Parámetros usados para la de-dispersión coherente en banda-base
BANDPASS	Banda de paso observada por polarización, promediada en cada observación
FLUX_CAL	Temperatura del sistema e inyección de ruido para calibrar los datos como función de la frecuencia en la banda de paso.
CAL_POLN	Polarización aparente del ruido de calibración como función de la frecuencia.
FEEDPAR	Parámetros del acoplamiento cruzado de la alimentación en función de la frecuencia.
SPECKURT	Datos estadísticos para la extracción de RFI
SUBINT	Cadenas de datos o perfiles de pulsos en función del tiempo, frecuencia y polarización
DIG_STAT	Modo de digitalización, ajustes del atenuador y estadísticas de conteo
DIG_CNTS	Modo de digitalización y distribución de la tasa de recuento

Tabla 3.1: Características del formato PSRFITS

PSRFITS nos permite grabar en dos formas los datos, en modo búsqueda (*search mode*) y en modo “Folding”. El primero se usa en la búsqueda de nuevos

pulsares cuando el período aún no es conocido. El segundo se usa para pulsares cuyos períodos ya se saben. En este caso, el doblamiento de los datos se realiza en tiempo real en el back-end, lo cual resulta en una reducción enorme en el volumen de datos grabados en el disco. En la tesis nos enfocaremos en datos en modo búsqueda; estos datos poseen una extensión *.sf*.

En la tabla 3.1 se muestran todos los HDU que existen en PSRFITS. Para datos en modo búsqueda solamente son necesarios tres de estos datos: primary, HISTORY y SUBINT (se pueden agregar más HDU en los archivos, pero se pueden considerar innecesarios). En la figura 3.1 se muestra el encabezado y los HDU de un archivo PSRFITS con extensión *.sf*.

```

Dec J2000 = -44:51:00.000
Dec J2000 (deg) = -44.85
Tracking? = False
Azimuth (deg) = 230.0623
Zenith Ang (deg) = 21.6557
Polarization type = LIN
Number of polns = 1
Polarization order =
Sample time (us) = 300
Central freq (MHz) = 436
Low channel (MHz) = 420.125
High channel (MHz) = 452
Channel width (MHz) = 0.125
Number of channels = 256
Total Bandwidth (MHz) = 32
Beam = 0 of 1
Beam FWHM (deg) = 0.750
Spectra per subint = 1536
Starting subint = 0
Subints per file = 349
Spectra per file = 536064
Time per subint (sec) = 0.4608
Time per file (sec) = 160.8192
bits per sample = 1
bytes per spectra = 256
samples per spectra = 256
bytes per subint = 393216
samples per subint = 393216
zero offset = 0
Invert the band? = False
PSRFITS Specific info:
HDUs = primary, HISTORY, SUBINT
FITS typecode = 11
DATA column = 18
Apply scaling? = True
Apply offsets? = False
Apply weights? = False

```

Figura 3.1: Ejemplo del contenido del HDU primario en un archivo en formato PSRFITS.

3.2.2. Filterbanking

El primer paso antes de usar PSRSOFT y PRESTO es convertir los datos crudos que salen del radiotelescopio a un formato esperado por estos dos paquetes. Para esto es necesario convertir el archivo de PSRFITS al formato filterbanked haciendo uso del programa SIGPROC que viene dentro de los instalados por

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

PSRSOFT o se puede instalar a parte (ver <http://sigproc.sourceforge.net/>).

Como se vio en Capítulo 2 es necesario tener canalizada la señal para la de-dispersión, por eso es necesario hacer “filterbank” a los datos. El programa para dividir los datos se encuentra en SIGPROC y se ejecuta desde terminal de la siguiente manera:

```
filterbank nombredelarchivo.sf > nombredelarchivo.fil
```

Filterbank convierte los datos del archivo .sf en cadenas de datos que se ordenan como:

$$[s_1, c_1][s_1, c_2] \dots [s_1, c_n], [s_2, c_1][s_2, c_2] \dots [s_2, c_n], [s_n, c_1][s_n, c_2] \dots [s_n, c_n] \quad (3.1)$$

Los índices $s_1, s_2 \dots s_n$ son cadenas de datos y los índices $c_1, c_2 \dots c_n$ son los canales de frecuencia. Otra manera de visualizarlo es una matriz $n \times n$, en el cual cada renglón es un tiempo de muestreo, cada columna una frecuencia distinta y los elementos son las muestras.

3.3. Búsqueda de Púlsares con PSRSOFT

Los paquetes instalados por PSRSOFT incluyen varios programas para realizar la búsqueda de púlsares. Los pasos principales se pueden apreciar en la Figura 3.2

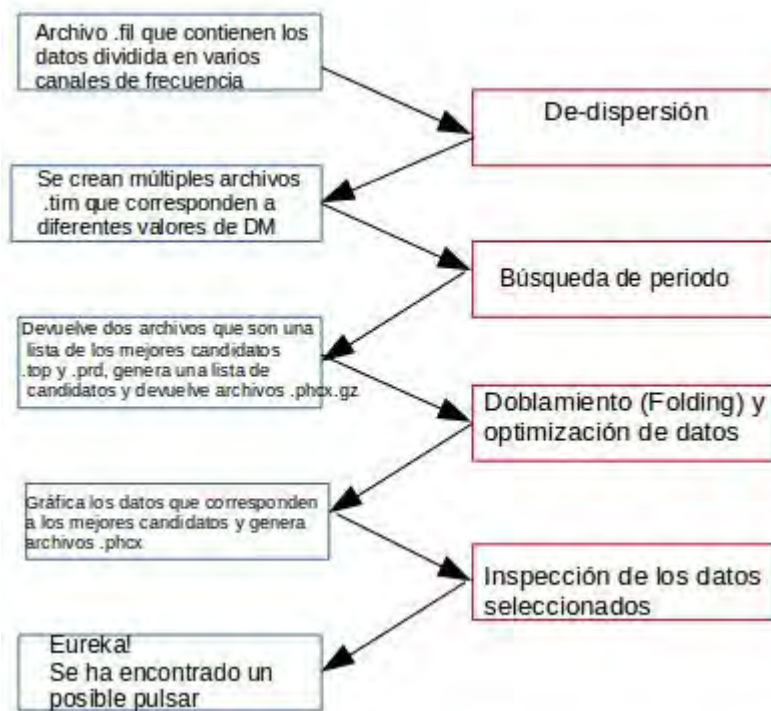


Figura 3.2: En azul a la izquierda del esquema se encuentran los archivos generados durante el proceso de búsqueda de púlsares con PSRSOFT y a la derecha en rojo son los pasos que se realizan durante este proceso

3.3.1. De-dispersión

Para la de-dispersión en PSRSOFT se recurre al comando “dedisperse_all” (pertenece al programa SIGPROC) que efectúa la de-dispersión incoherente sobre los datos y canales, como se vio en el capítulo 2. La sentencia completa se escribe desde terminal como:

```
$ dedisperse_all nombredelarchivo.fil -d 0 80 -g 100000
```

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

En la sentencia anterior “-d” es el rango de valores de DM que se utilizarán para la prueba y la opción -g es el número de muestras que se van a de-dispersar. En este ejemplo se hará la de-dispersión en un rango de 0 a 80 $cm^{-3}pc$ usando un bloque de muestras con tamaño 100000.

Este comando genera una serie de archivos .tim como se muestran en la figura 3.3.

```
:psrsoft: ls
S13405_1.fil
S13405_1.fil.0000.00.tim
S13405_1.fil.0000.28.tim
S13405_1.fil.0000.57.tim
S13405_1.fil.0000.85.tim
S13405_1.fil.0001.13.tim
S13405_1.fil.0001.42.tim
S13405_1.fil.0001.70.tim
S13405_1.fil.0001.98.tim
S13405_1.fil.0002.27.tim
S13405_1.fil.0002.55.tim
S13405_1.fil.0002.84.tim
S13405_1.fil.0003.12.tim
S13405_1.fil.0003.40.tim
S13405_1.fil.0003.69.tim
S13405_1.fil.0003.98.tim
S13405_1.fil.0004.26.tim
S13405_1.fil.0004.55.tim
S13405_1.fil.0004.84.tim
S13405_1.fil.0005.12.tim
S13405_1.fil.0005.41.tim
```

Figura 3.3: Archivos generados por la dedispersión

En este ejemplo, los nuevos archivos tienen un nombre similar a **S13405_1.fil.0000.00.tim**. Los números que se presentan después de la extensión “.fil” son los valores de prueba de la DM. En la figura se observan archivos que van desde 0 hasta 5.41, pero si observamos la carpeta completa se encontrarán archivos que van desde 0 hasta 80. La extensión .tim indica que los datos (en formato filterbank) ya han sido de-dispersados

Es necesario hacer la de-dispersión en un amplio rango de valores de DM, pues desconocemos la DM del objeto y la distancia del pulsar a la Tierra. Estos valores sirven para hacer la corrección del retraso en tiempo ocasionado por el MIE en los canales.

Una vez generados los archivos .tim se hará una búsqueda de periodicidad en los datos de cada archivo como se muestra en la siguiente sección.

3.3.2. Búsqueda de periodicidad y selección de candidatos

Para la búsqueda de periodicidad en series de datos en tiempo se usa la función *seek* del programa SIGPROC. Este programa lee las “time series data” generadas

por la función **dedisperse** y realiza la Transformada de Fourier, sumando los armónicos de la señal. El resultado de seek son dos archivos .prd y .top para cada archivo .tim. En el primero se identifican una serie de valores que son la SNR y el período de los posibles candidatos; en el segundo se encuentra el mejor candidato de los que se encuentran en el archivo .prd.

La tarea anterior se puede realizar archivo por archivo de manera que desde terminal se ejecute:

```
$seek nombredelarchivo.fil.DM.tim -fftw
```

Como ejecutar dicha sentencia puede resultar tediosa, pues la cantidad de archivos generados por la de-dispersión puede ir desde unos cientos hasta miles de archivos, se puede ejecutar el siguiente script que minimiza la tarea:

```
file="nombredelarchivo"
for timfile in ${file}.fil.*.tim ; do seek $timfile --fftw -head ; done
cat ${file}.fil.*.prd > ${file}.prd
```

Los archivos que se generan se puede ver en la Figura 3.4, en donde a cada archivo .tim generado anteriormente le corresponden dos archivos uno .prd y uno .top.

```
:psrsott: ls
fftw
S13405_1.fil
S13405_1.fil.0000.00.prd
S13405_1.fil.0000.00.tim
S13405_1.fil.0000.00.top
S13405_1.fil.0000.28.prd
S13405_1.fil.0000.28.tim
S13405_1.fil.0000.28.top
S13405_1.fil.0000.57.prd
S13405_1.fil.0000.57.tim
S13405_1.fil.0000.57.top
S13405_1.fil.0000.85.prd
S13405_1.fil.0000.85.tim
S13405_1.fil.0000.85.top
S13405_1.fil.0001.13.prd
S13405_1.fil.0001.13.tim
S13405_1.fil.0001.13.top
S13405_1.fil.0001.42.prd
S13405_1.fil.0001.42.tim
S13405_1.fil.0001.42.top
S13405_1.fil.0001.70.prd
S13405_1.fil.0005.70.tim
S13405_1.fil.0005.70.top
S13405_1.fil.0005.99.prd
S13405_1.fil.0005.99.tim
S13405_1.fil.0005.99.top
S13405_1.fil.0006.28.prd
S13405_1.fil.0006.28.tim
S13405_1.fil.0006.28.top
S13405_1.fil.0006.57.prd
S13405_1.fil.0006.57.tim
S13405_1.fil.0006.57.top
S13405_1.fil.0006.86.prd
S13405_1.fil.0006.86.tim
S13405_1.fil.0006.86.top
S13405_1.fil.0007.16.prd
S13405_1.fil.0007.16.tim
S13405_1.fil.0007.16.top
S13405_1.fil.0007.45.prd
S13405_1.fil.0007.45.tim
S13405_1.fil.0007.45.top
S13405_1.fil.0007.75.prd
```

Figura 3.4: Archivos generados por seek

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

El archivo .prd contiene la información mostrada en la figura 3.5

```
##BEGIN HEADER##
SOURCEID = G123070
FREQ = 451.9 MHz
TSTART = 48664.6153
TELESCOPE = Parkes
RAJ = 08:32:37.0
DECJ = -44:51:00.0
TSAMP = 300.0 us
PROGRAM = SEEK
VERSION = 4.3
HARM_FOLDS = 1 2 4 8 16
BARYCENTRIC = F
COLS = SNR SPEC PERIOD
##END HEADER##
DM: 33.2081871 AC: 0.00000000 AD: 0.00000000 TSAMP: 3.0000001424923539E-004
50.2 89.27850573 63.1 89.29751329 50.6 89.29434480 50.1 89.29909762 39.6 89.29830545
17.0 17.85975615 9.1 89.39902378 24.5 178.69647941 24.8 267.79276611 19.7 357.07601994
13.2 14.88362305 6.9 178.45571689 8.3 71.43902460 18.0 357.29147718 18.6 446.38065177
11.1 89.37997299 6.7 9907.80519500 7.2 3.54384852 16.3 446.71740828 17.7 624.96616863
9.3 12.75665828 6.7 2.29055512 6.1 89.39584809 15.5 625.47098783 14.3 803.31418707
7.1 4953.90259750 6.6 4.61692183 6.1 4886.56799909 8.9 223.15074436 13.9 268.01379371
6.8 1.28538985 6.4 1.96356734 5.7 3.81617818 7.8 71.43801063 13.7 1160.38387068
6.5 0.88596213 6.4 2.00184111 5.7 16.61253124 7.2 268.19230772 12.7 981.98517980
6.1 1.81809300 6.3 3.80264449 5.7 4.34685715 6.7 125.01962391 11.4 1429.26736875
6.0 0.98178367 6.3 5.46961874 5.6 51.02663313 6.7 9.58524655 10.6 1250.94197566
5.9 0.71392007 6.2 4934.47552849 5.6 3.92713468 5.9 74.42097617 7.9 107.15815666
5.9 5.46954741 6.1 1.21472749 5.6 178.44306314 5.9 12.58483178 6.5 10.25549264
5.8 0.84898642 5.8 1.87889076 5.4 9.58526481 5.8 277.72250947 6.2 536.38461545
5.8 13.32399309 5.8 2.23229921 5.4 5.12777244 5.8 89.19781380 6.0 41.31386624
5.7 0.62871370 5.8 1.77192426 5.4 5.43793208 5.7 16.61269573 6.0 16.31456298
5.6 0.82793323 5.8 1.69797284 5.4 8.56321230 5.6 5.12776722 5.9 39.20108914
5.6 1.15554271 5.7 1.23987295 5.3 2.85202970 5.5 6.00551138 5.8 89.19702342
5.6 1.70144665 5.6 1.79461833 5.3 10.93904727 5.5 32.62886155 5.8 555.44501893
5.6 0.76345305 5.6 1.27947554 5.3 5.81864755 5.4 17.12642459 5.7 10.13069035
5.5 5.53752260 5.5 1.33630047 5.2 2.63461595 5.3 7.80535337 5.6 74.42042597
5.5 89.17726859 5.4 39.33266418 5.2 4.96693592 5.2 5.99409666 5.6 17.12651201
```

Figura 3.5: Contenido del archivo .prd. Cada par de columnas de datos corresponde a un armónico de la señal el primer par corresponde a la fundamental, el segundo par al armónico 2 y así sucesivamente en potencia de dos.

La primera parte del archivo .prd es un encabezado donde se proporciona la información básica sobre el documento, como son el telescopio del cual han sido tomado los datos, la frecuencia central, tiempo de muestreo, entre otros. Las columnas que se aprecian corresponden al valor de SNR y a períodos de diferentes candidatos que se encuentran en los datos.

Por ejemplo en la Figura 3.5 podemos observar que se encuentran los valores *50.2 89.27850573*, estos valores corresponden a la SNR y al Período del armónico 1, para el armónico 2 los valores son *63.1 89.29751329* y así sucesivamente podemos ver que cada par de valores corresponden a un armónico.

El archivo .top corresponde al mejor candidato del archivo .prd, los tres valores que se muestran corresponden al Período, a la SNR y a la DM en ese orden.

89.297513289732805 63.0542336 33.2081871

Es necesario generar una lista de candidatos para poder continuar con el procesamiento de los datos. Para poder crear esta lista de candidatos recurrimos

a PulsarHunter, con este programa podemos leer la información de todos los archivos .prd y determinar cuales son los mejores candidatos (se puede crear la lista de manera manual con los archivos .top, pero no es recomendable porque esto hace mas lento el proceso).

\$ph-best nombredelarchivo.prd nombredelarchivo –maxresults 50

Este comando genera un archivo ASCII, nombredelarchivo.lis, y varios archivos .phcx.gz. El archivo .lis (ver Figura 3.6) contiene la lista de los mejores candidatos. Los archivos .phcx.gz contienen archivos .phcx que son archivos tipo xml que contienen el período, la DM y la SNR del mejor candidato, son necesarios para el siguiente paso (“Folding”).

S13405_1_001.phcx.gz	446.800000	89.29830545	69.264542	0.000000e+00
S13405_1_002.phcx.gz	151.100000	29.76795031	62.837151	0.000000e+00
S13405_1_003.phcx.gz	65.200000	267.79276611	62.837151	0.000000e+00
S13405_1_004.phcx.gz	44.200000	290.28468989	69.264542	0.000000e+00
S13405_1_005.phcx.gz	43.100000	122.78557845	68.430679	0.000000e+00
S13405_1_006.phcx.gz	37.200000	20.00733421	0.000000	0.000000e+00
S13405_1_007.phcx.gz	36.000000	714.22804584	69.264542	0.000000e+00
S13405_1_008.phcx.gz	31.300000	981.98517980	62.837151	0.000000e+00
S13405_1_009.phcx.gz	24.200000	89.39902378	62.837151	0.000000e+00
S13405_1_010.phcx.gz	23.400000	7.76495324	68.430679	0.000000e+00
S13405_1_011.phcx.gz	22.400000	113.65266372	66.789574	0.000000e+00
S13405_1_012.phcx.gz	14.100000	312.51218770	37.630695	0.000000e+00
S13405_1_013.phcx.gz	12.600000	5.76110351	68.430679	0.000000e+00
S13405_1_014.phcx.gz	11.900000	62.11981757	68.430679	0.000000e+00
S13405_1_015.phcx.gz	11.100000	148.84415316	0.849791	0.000000e+00
S13405_1_016.phcx.gz	10.100000	401.83346206	6.864061	0.000000e+00
S13405_1_017.phcx.gz	10.000000	327.4348576	37.630695	0.000000e+00
S13405_1_018.phcx.gz	9.600000	715.03978393	77.188507	0.000000e+00
S13405_1_019.phcx.gz	9.500000	220.08198863	1.609835	0.000000e+00
S13405_1_020.phcx.gz	9.300000	53.36321461	0.000000	0.000000e+00
S13405_1_021.phcx.gz	9.200000	89.50076533	62.837151	0.000000e+00
S13405_1_022.phcx.gz	8.800000	3.71337838	40.796471	0.000000e+00
S13405_1_023.phcx.gz	8.700000	260.09172617	1.133060	0.000000e+00
S13405_1_024.phcx.gz	8.600000	4.68969200	54.844734	0.000000e+00
S13405_1_025.phcx.gz	8.400000	4.04691554	64.392868	0.000000e+00
S13405_1_026.phcx.gz	8.400000	57.14992181	68.430679	0.000000e+00
S13405_1_027.phcx.gz	8.300000	267.50899850	65.982117	0.000000e+00
S13405_1_028.phcx.gz	8.100000	73.36600959	0.000000	0.000000e+00
S13405_1_029.phcx.gz	8.000000	86.71666630	0.000000	0.000000e+00
S13405_1_030.phcx.gz	7.800000	491.16028681	6.864061	0.000000e+00
S13405_1_031.phcx.gz	7.800000	1.21738941	20.758390	0.000000e+00
S13405_1_032.phcx.gz	7.700000	1.22778570	8.933798	0.000000e+00
S13405_1_033.phcx.gz	7.700000	2.96156636	23.010578	0.000000e+00
S13405_1_034.phcx.gz	7.700000	21.55466448	68.430679	0.000000e+00
S13405_1_035.phcx.gz	7.600000	1.25911976	34.150130	0.000000e+00
S13405_1_036.phcx.gz	7.600000	4953.90259750	40.796471	0.000000e+00
S13405_1_037.phcx.gz	7.600000	804.59835969	63.610867	0.000000e+00
S13405_1_038.phcx.gz	7.600000	69.81973476	73.571327	0.000000e+00
S13405_1_039.phcx.gz	7.500000	1.44672586	0.000000	0.000000e+00
S13405_1_040.phcx.gz	7.500000	2.22061829	14.217115	0.000000e+00
S13405_1_041.phcx.gz	7.500000	7.75545320	50.216377	0.000000e+00
S13405_1_042.phcx.gz	7.400000	1.45394920	2.835430	0.000000e+00

Figura 3.6: Lista de los mejores candidatos obtenidos con PulsarHunter. La primera columna es el nombre del archivo, la segunda es la SNR, la tercer columna es el período y la columna que se encuentra al final es el rango de aceleración que en este caso es 0.

3.3.3. Doblamiento (Folding) y optimización de datos

Una vez que se han reducido la cantidad de candidatos a un número razonable, es posible ahora “reconstruir” uno o varios perfiles de la señal. Para esto primero usamos el software **DSPSR**.

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

DSPSR hace uso de la de-dispersión coherente por tener mayor precisión que de-dispersión incoherente debido a que es capaz de remover toda la dispersión de la señal, se usa en este punto porque al ya conocer la Medida de Dispersión consume menos recursos. DSPSR se ejecuta sobre los datos del archivo `.fil` y después los divide en bloques de tiempo t a los que se les llama sub-integraciones. Es necesario especificar el período y la medida de dispersión para poner en fase los datos de las sub-integraciones. Un ejemplo del comando es

```
$ dspsr nombredelarchivo.fil -c 0.08929830545 -D 63.0542336 -L 10  
-t 4 -U 1 -e subint
```

donde `-c` especifica el período de la señal en segundos, `-D` establece el valor de la medida de dispersión, `-L` especifica el tiempo de sub-integración en segundos, `-t` especifica el número de multi-hilos para realizar la ejecución de folding, `-U` el límite de memoria que se permite usar, `-e` especifica la extensión de los archivos de salida, para este caso son archivos `.subint`.

Hay que ejecutar `dspsr` una vez para cada candidato que se quiere considerar del archivo `.lis`. En caso de N candidatos, tendremos $N \times t_{obs}/L$ archivos `.subint`. Estos se deben comprimir en un solo archivo de extensión `.ar`. Se usa la función `psradd` para realizar esta compresión.

```
$ psradd *.subint -o nombredelarchivo.ar
```

`psradd` es la función y `-o` es para especificar el archivo de salida que se va a crear a partir de los archivos `.subint`.

Para reducir el tiempo de procesamiento en los siguiente pasos es recomendable reducir el número de canales, que debe ser un divisor entero del total de canales con el que se hayan tomado datos (información que se encuentra en el encabezado de los archivos), por ejemplo si el archivo posee 96 canales el divisor puede ser 2, 4, 8, 12, 14 etcétera. El comando es `pam` que pertenece a PSRCHIVE y se escribe de la siguiente manera:

```
pam -- setnchn 8 -m nombredelarchivo.ar
```

`setnchn` es para seleccionar al número de canales al que se desea reducir el archivo `.ar`.

Al disminuir el número de canales disminuimos el tiempo de procesamiento, esto no afecta de manera significativa el procesamiento de los datos, suele usarse porque en ocasiones la cantidad de datos es mucha y el procesar todos los canales hace tardado la creación del perfil de la señal.

3.3.4. Resultados

El último paso es crear un archivo gráfico para poder observar los resultados y distinguir aquellos posibles pulsares de los candidatos falsos. Para esto tenemos que usar la función *pdmp* que es parte de PSRCHIVE y se utiliza para mejorar la SNR de los candidatos haciendo un ajuste fino de los datos. Para esto es necesario usar los archivos *.phcx* que se generaron con *PulsarHunter* (se encuentran comprimidos en *.gz*) y el archivo *.ar*.

```
$ pdmp -input-phcx candidato.phcx.gz -output-phcx candidato.phcx
archivo.ar
```

El resultado de *pdmp* es una gráfica en PGPLOT como se muestra en la imagen 3.7.

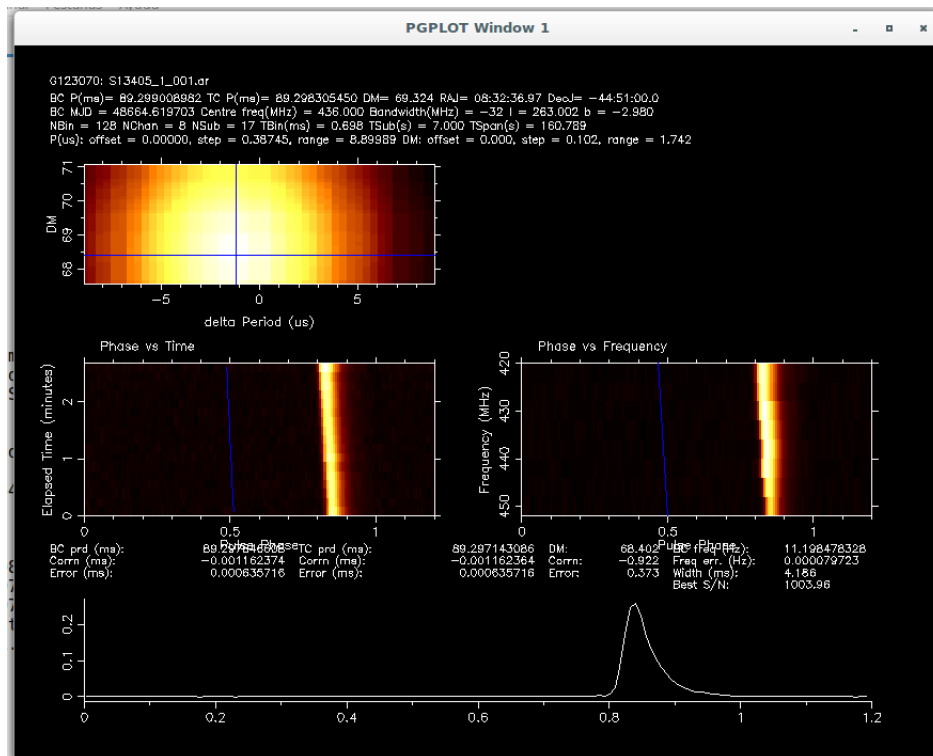


Figura 3.7: Imagen obtenida con la herramienta PGPLOT. En el lado superior izquierdo se encuentran los valores de las DM de prueba en función de la fase del pulso. Abajo del lado izquierdo una gráfica de Fase contra tiempo y a su lado de la Fase contra la frecuencia. En la parte inferior se encuentra la reconstrucción del pulso después de haber sumado los canales.

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

Opcionalmente, por medio del subcomando `--imageoutput` se puede generar el resultado anterior en formato `.png` para que se pueda visualizar con otras aplicaciones.

\$ ph-view-phcx archivo.phcx.gz --imageoutput

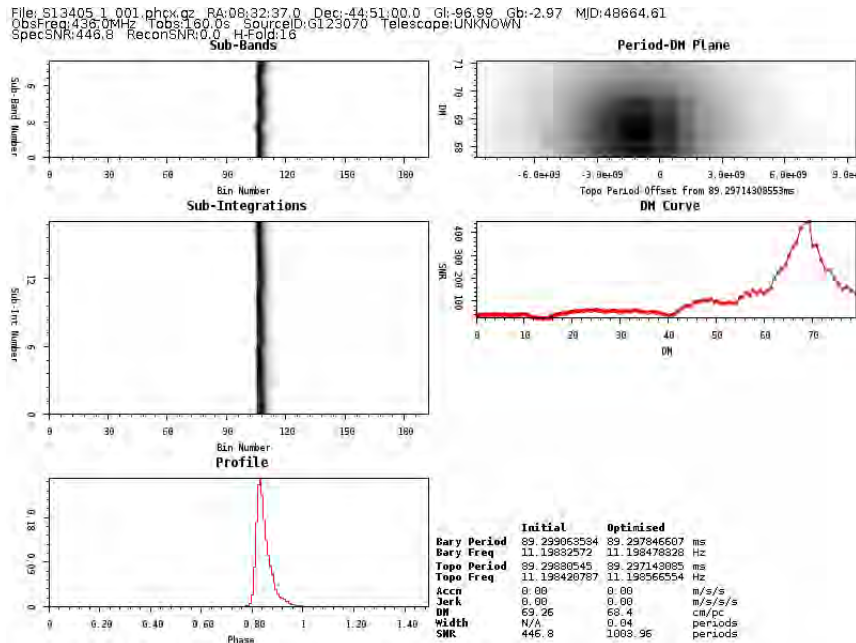


Figura 3.8: Imagen png obtenida a partir de PGPLOT. Del lado superior izquierdo se encuentra la intensidad de la señal en función del canal de frecuencia y la fase. Del lado superior derecho se encuentran los valores de las DM de prueba en función de la fase del pulso. En el centro del lado izquierdo se encuentra la intensidad de la señal en función de la sub-integración y de la fase. En el centro del lado derecho la relación S/N en función de la DM. En la parte final del lado izquierdo se encuentra el perfil del pulso reconstruido en función de su fase. Del lado inferior derecho se encuentran datos que corresponden al período, Ancho del pulso, SNR y a la DM

Hasta este punto, se puede hacer manual todo el proceso de Folding, candidato por candidato. Todo esto muy tardado porque pueden haber desde unos pocos hasta unos cientos de candidatos. Se recomienda usar el siguiente script, que hace toda la parte de Folding de manera automática (desde `dspsr` hasta generar las imágenes `.png`).

```
file="nombredelarchivo"#sin extensiones
while read line ; do
filename=$(echo $line | cut -f1 -d' ')
cand=$(echo $filename | cut -f1 -d'.')
echo $cand
period=$(echo $line | cut -f3 -d' ')
period=$(echo $period/1000|bc -l) #convierte a segundos
los valores de la columna tres
echo $period
dm=$(echo $line | cut -f4 -d' ')
echo $dm
dspsr ${file}.fil -c ${period} -D ${dm} -L 10 -t 4 -U 1 -e subint
#dspsr necesita los valores de period, dm y file#
psradd *.subint -o ${cand}.ar #psradd lo mismo que dspsr
rm *.subint
pam --setnchn 8 -m ${cand}.ar #lo mismo dspsr
pdmp -input-phcx ${cand}.phcx.gz -output-phcx ${cand}.phcx ${cand}.ar
#lo mismo que dspsr
gzip -f ${cand}.phcx
ph-view-phcx ${filename} --imageoutput      #crea las imagenes
de diferentes archivos.
done < ${file}.lis
```

3.4. Búsqueda de Púlsares con PRESTO

PRESTO es un programa escrito por Scott Ransom del GBT (Green Bank Telescope). Este programa contiene rutinas que están escritas en varios lenguajes de programación, Python, C, FORTRAN, entre otros

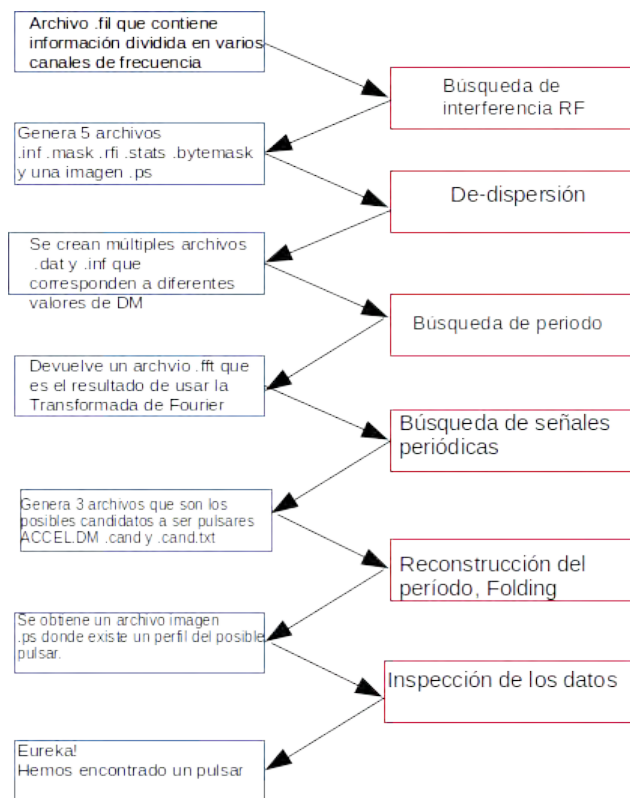


Figura 3.9: Esquema de procesamiento de datos con PRESTO

A diferencia con el proceso que se lleva a cabo con PSRSOFT, con PRESTO se debe hacer una pre-inspección de los datos por búsqueda de Interferencia por Radio Frecuencia (RFI)

3.4.1. Búsqueda de Interferencia de Radio-Frecuencia (rfi-find)

Para hacer la búsqueda de pulsares con PRESTO es necesario primero detectar algunas señales que provoquen interferencia. Estas señales pueden ser periódicas

lo que produce que se observen falsos pulsares. El primer paso para detectar las RFI es de-dispersar los datos con una $DM=0$ y promediar la potencia de todos los canales en un intervalo de tiempo. Una vez obtenido el nivel promedio de la señal, se hace una revisión de la potencia en los canales por el mismo intervalo de tiempo del que se obtuvo el promedio, si se detectan en estos intervalos potencias superiores a la potencia promedio se eliminan los canales.

Por ejemplos si calculamos la potencia de todos los canales en un intervalo de tiempo de dos segundos. Se tiene que comparar este promedio con la potencial en cada canal durante dos segundos. Sí la potencia en cada canal es mayor que la del promedio, el canal debe ser removido.

La sentencia para llevar acabo la detección de RFI con PRESTO es la siguiente:

```
rfifind -time 1.0 -o archivonuevo archivo.fl
```

En el ejemplo anterior se asigna un intervalo de tiempo de 1 segundo, la opción `-time` es variable.

3. USO DE PSRSoft Y PRESTO PARA LA BÚSQUEDA DE PULSARES

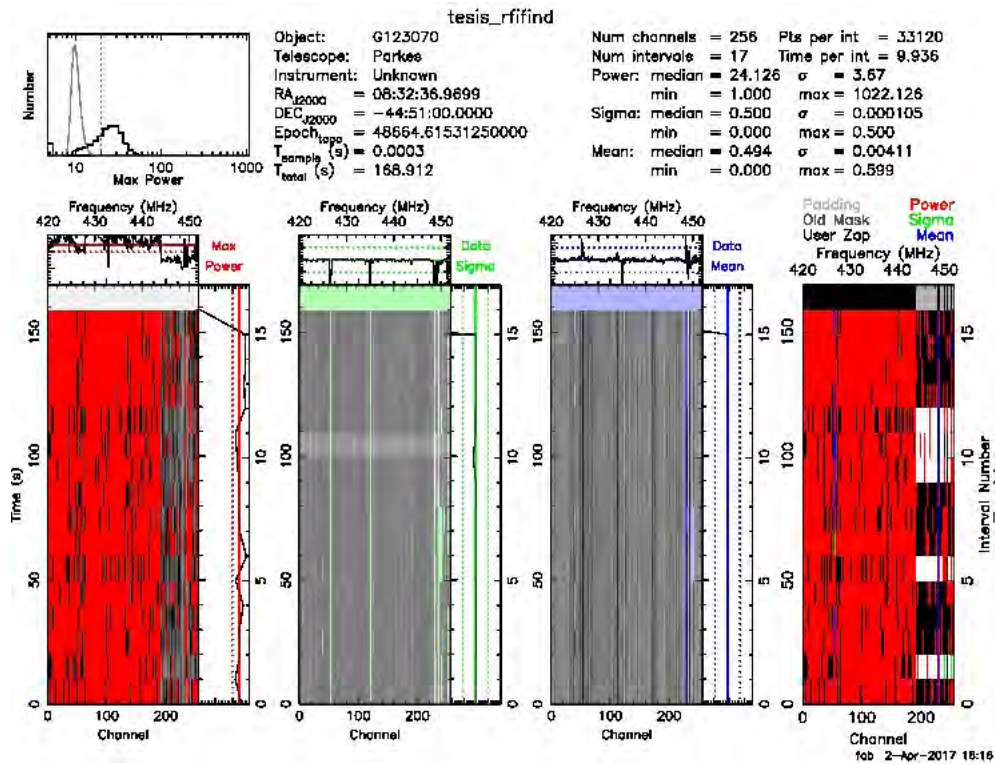


Figura 3.10: Se observa en el cuadro superior izquierdo una gráfica donde se encuentra la potencia máxima del intervalo de tiempo de los datos. Las gráficas que se presentan debajo nos sirven de referencia para saber si el tiempo que usamos es correcto o hay que disminuirlo para la búsqueda de RFI. Son gráficas del tiempo contra los canales. Aquellos canales en rojo tienen mucha interferencia y tendrán que ser eliminados esto es perjudicial para los siguientes procesos. Se recomienda disminuir el intervalo de tiempo para evitar eliminar demasiada información.

Disminuyendo el intervalo de tiempo $-t$ buscamos nuevamente la interferencia y esta vez se observa como

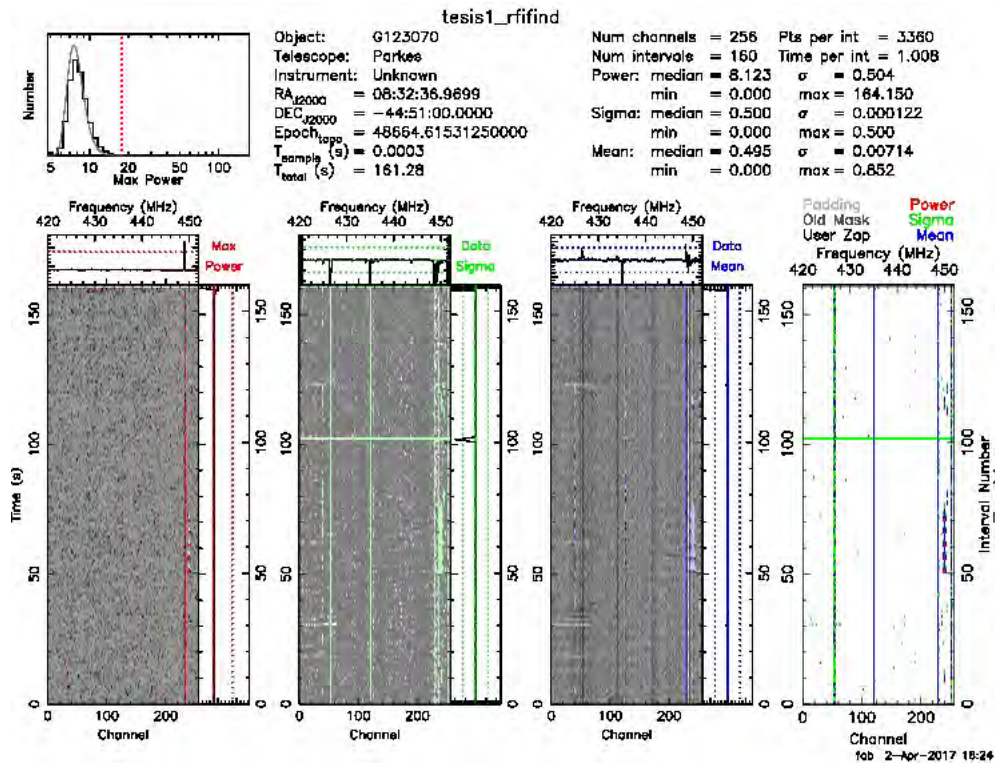


Figura 3.11: En este caso hemos disminuido el intervalo de tiempo

Al realizar una búsqueda de RFI con la sentencia anterior no solamente genera las imágenes mostradas en las Figuras 3.10 y 3.11, también genera otros archivos con las siguientes extensiones `.bytemask`, `.inf`, `.mask`, `.ps`, `.stats`, `.rfi` que usaremos posteriormente.

3.4.2. De-dispersión

Igual como en el procesamiento por PSRSOFT, es necesario realizar una de-dispersión de los datos con PRESTO. La diferencia radica en el código de PRESTO y el algoritmo que usa para llevar a cabo esta tarea. En este caso a parte de poder seleccionar el intervalo en el que queremos efectuar la de-dispersión, también podemos seleccionar el tamaño de los pasos en los que será buscada la DM.

prepsubband -lodm 0.0 -dmstep 1.0 -numdms 24 -mask archivo.mask -o archivonuevo archivo.fil

-lodm indica la DM más baja en la que se empezara a buscar, *-dmstep* el tamaño de los pasos, *-numdms* el número de pasos que hará. En este ejemplo se buscará

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

en un rango de 0 a 24, en pasos de 1.0, `-mask archivo.mask` se usa como referencia de los canales con Interferencia para no tomarlos en cuenta en el archivo `.fil`, `-o` es para indicar el nombre del nuevo archivo que se creará. Los archivos que se obtienen después de la de-dispersión son `.dat`, pero son también series de tiempo (como con PSRSOFT).

El comando `prepsubband` genera varios archivos en formato `.dat` que se pueden explorar como:

exploredat archivo.dat

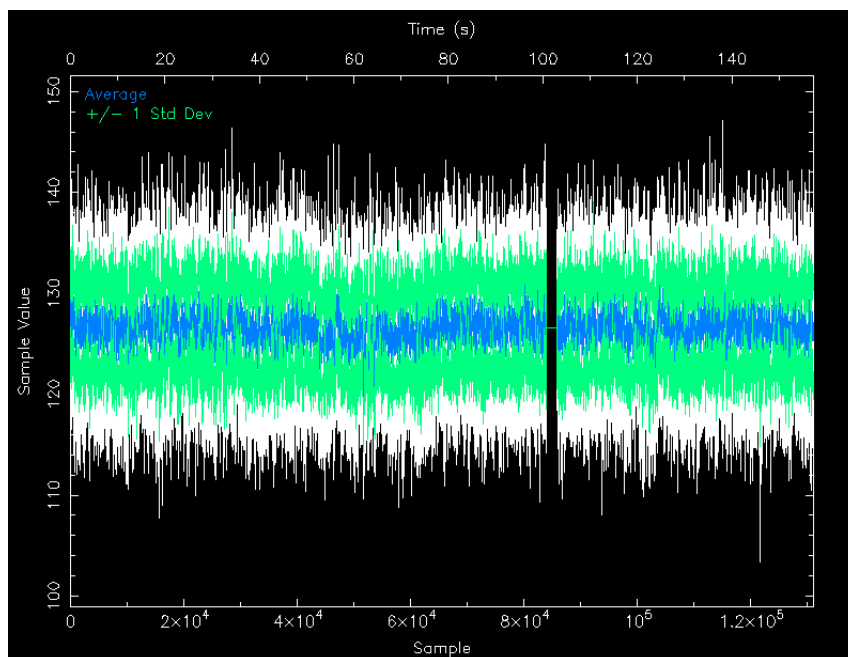


Figura 3.12: Datos en tiempo de un archivo `.dat`

3.4.3. Transformada de Fourier

Una vez que hemos creado los archivos `.dat` (que se encuentran en el dominio del tiempo) hay que pasar al dominio de la frecuencia; para eso hacemos uso de la Transformada de Fourier. En el caso de PRESTO, ejecutamos la siguiente sentencia para poder transformar un archivo:

realfft archivo.dat

El resultado es un archivo `.fft`, que se puede observar ejecutando:

explorefft archivo.fft

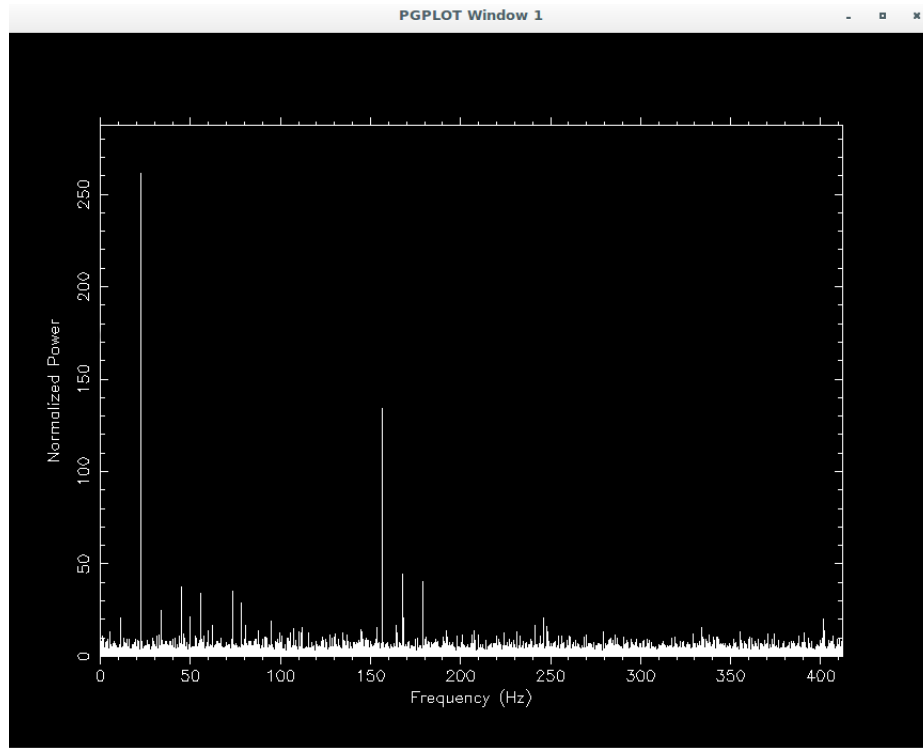


Figura 3.13: Espectro de la señal generado con el comando `explorefft` de PRESTO.

Para disminuir el trabajo y convertir todos los archivos `.dat` de tiempo a Frecuencia podemos realizar el siguiente comando

```
ls *.dat | xargs -n 1 realfft1
```

3.4.4. Búsqueda de periodicidad

Haciendo uso de algunos algoritmos especializados para la búsqueda de pulsares, PRESTO explora los datos en el dominio de frecuencia y realiza una suma de armónicos en el caso de que los hayan.

PRESTO contiene una rutina para la búsqueda de señales periódicas dentro de los datos, algunas veces las señales pueden contener aceleraciones que se deben al tipo de fuente (pulsares binarios), en el caso de nuestra tesis tratamos con radio-pulsares por esta razón la aceleración es 0.

```
accelsearch -zmax 0 archivo.fft
```

¹Es un comando nativo de UNIX.

3. USO DE PSRSOFT Y PRESTO PARA LA BÚSQUEDA DE PULSARES

dónde `accelsearch` es el nombre del programa y `-zmax 0` es el valor de la máxima aceleración, se generan un archivo `.cand` que posteriormente se usaran para crear una lista de candidatos.

Para no tener que realizar de manera manual la sentencia anterior con cada uno de los archivos generados, se puede recurrir a un recurso de UNIX y usar el siguiente comando

```
ls *.fft | xargs -n 1 accelsearch -zmax 0
```

Lo siguiente es crear una lista de posibles candidatos, para después reconstruir los probables perfiles del pulsar y seleccionar alguno. La lista se construye usando un programa en Python escrito por Scott Ransom que se encuentra en la carpeta donde fue descargado previamente PRESTO.

```
python ACCEL_sift.py > lista.txt
```

La lista del ejemplo anterior se muestra en la Figura 3.13

#	filecandname	DM	SNR	sigma	numharm	ipow	cpow	P[no]	r	z	numbits
disper_DM68.50_ACCEL_0:1	68.50	385.29	192.99	0	10689.0	145127.5	09.300758	445.12	0.00	[169]	
disper_DM68.50_ACCEL_0:2	68.50	90.95	78.05	0	3099.1	4026.7	25.513479	1558.00	0.00	[50]	
disper_DM68.50_ACCEL_0:7	68.50	61.87	68.20	0	7110.9	1320.1	33.407708	1107.00	0.00	[31]	
disper_DM68.50_ACCEL_0:8	68.50	61.50	69.22	0	3065.2	3168.0	19.806600	2003.00	0.00	[34]	
disper_DM68.50_ACCEL_0:9	68.50	60.17	43.06	0	102.0	1823.5	16.236082	2440.25	0.00	[14]	
disper_DM68.50_ACCEL_0:9	68.50	36.30	41.62	0	895.0	1165.5	13.730250	2893.36	0.00	[11]	
disper_DM68.50_ACCEL_0:10	68.50	17.23	24.24	1	290.0	290.1	5.252725	7567.50	0.00	[25]	
disper_DM68.50_ACCEL_0:13	68.50	16.12	22.67	1	261.0	261.1	4.699970	8457.50	0.00	[21]	
disper_DM47.00_ACCEL_0:2	47.00	26.66	19.94	0	232.4	3.3	3975.000000	10.00	0.00	[170]	
disper_DM68.50_ACCEL_0:14	68.50	11.96	16.75	1	144.0	143.8	4.464707	8903.00	0.00	[20]	
disper_DM68.50_ACCEL_0:14	68.50	11.96	16.75	1	144.0	143.0	4.464707	8903.00	0.00	[20]	
disper_DM68.50_ACCEL_0:16	68.50	8.03	12.29	1	79.0	79.1	3.002594	10230.00	0.00	[7]	
disper_DM31.00_ACCEL_0:3	31.00	13.83	8.97	0	53.6	117.3	39.864411	597.13	0.00	[14]	
disper_DM67.00_ACCEL_0:13	67.00	8.02	8.04	2	39.0	34.6	13.337807	2980.25	0.00	[3]	
disper_DM67.00_ACCEL_0:13	67.00	8.02	8.04	2	39.0	34.6	13.337807	2980.25	0.00	[3]	
disper_DM24.50_ACCEL_0:4	24.50	11.26	7.67	0	41.9	40.3	31.219321	1273.25	0.00	[2]	
disper_DM41.50_ACCEL_0:3	41.50	10.07	7.41	0	39.7	50.0	55.121023	721.13	0.00	[15]	
disper_DM9.50_ACCEL_0:7	9.50	4.68	6.30	1	22.9	22.9	13.569868	2929.50	0.00	[5]	
disper_DM61.00_ACCEL_0:6	61.00	10.85	6.20	0	40.0	139.8	46.678972	851.56	0.00	[6]	
disper_DM19.50_ACCEL_0:6	19.50	9.51	6.11	0	29.9	29.4	25.652576	1580.25	0.00	[3]	
disper_DM0.50_ACCEL_0:8	0.50	7.30	5.72	0	27.3	15.9	10.570476	3757.63	0.00	[2]	
disper_DM17.00_ACCEL_0:6	17.00	6.69	5.96	0	25.0	73.6	21.777039	1825.25	0.00	[4]	
disper_DM64.00_ACCEL_0:10	64.00	10.11	4.82	0	29.6	3.6	31.165764	1275.84	0.00	[3]	

Figura 3.14: Se muestra la lista de candidatos completa, hay que tener en cuenta que se debe explorar todo el archivo `.txt` generado con `ACCEL_sift.py` de manera manual

3.4.5. Construcción del perfil del pulsar (Folding)

Una vez que tenemos la lista de los mejores candidatos, podemos empezar a construir los perfiles del pulsar con cada uno de los candidatos. Para eso retomamos los archivos `.dat` y seleccionamos aquellos que se muestren en la lista usando el comando `prefold`:

```
prefold -accelcand 1 -accelfile Lband_DM44.00_ACCEL_0.cand
Lband_DM40.00.dat
```

El comando `prefold` sirve para ejecutar el programa que hace folding sobre los datos; `-accelcand` es para seleccionar el mejor candidato del archivo `.cand`, el índice

1 es para decir que se quiere usar el candidato número 1 que se muestra en la lista generada en el paso anterior. La opción *-accelfile* es para acceder al archivo .cand y el archivo .dat es donde se hace el folding. A diferencia de PSRSOFT donde folding se realiza sobre el archivo .phcx, en PRESTO los archivos que han sido de-dispersados y se encuentran en el dominio del tiempo (archivos .dat) son los usados para llevar a cabo Folding.

El resultado de todo este proceso es una gráfica en formato postscript (.ps) el mostrado en la Figura 3.15.

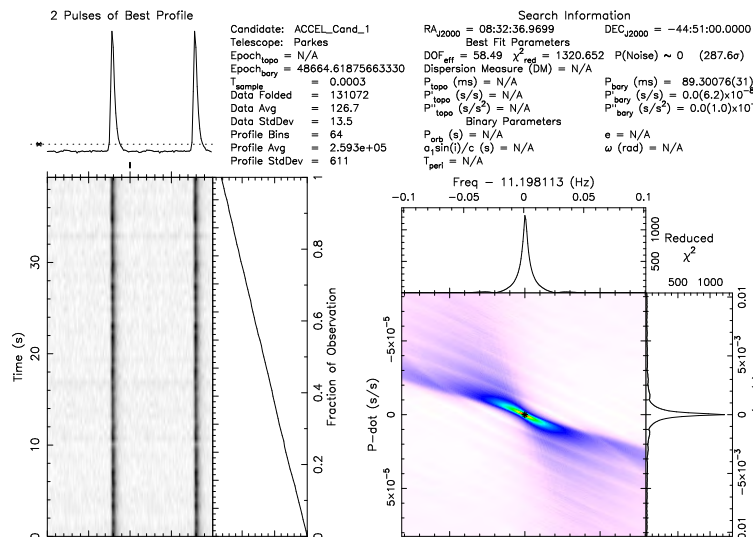


Figura 3.15: Output del comando prepfold en la esquina superior derecha se muestran los datos de la búsqueda. Al lado izquierdo de la información se muestra el perfil del pulso reconstruido. En este caso se observan dos picos; esto se debe a que la escala de la fase llega hasta 1.5 (figura que se encuentra del lado inferior izquierdo). La última parte de la imagen nos indica el cambio del período del pulsar.

Búsqueda de Radio-pulsares con datos de la ATNF (Australian Telescope National Facility)

En este capítulo procesamos datos reales tomados con el radiotelescopio Parkes (Figura 4.1), estos se encuentran disponibles para su descarga gratuita en línea desde la página:

<https://data.csiro.au/dap/public/atnf/pulsarSearch.zul>. En la página anterior se encuentran disponibles al público en general datos de varios pulsares con diferentes extensiones (.sf, .Ftp, .rf, etcétera). La selección de los datos para descargar se puede basar en backends y frecuencias de observación entre otras características. El artículo de [Khoo J. 2012](#) se ven los parámetros de búsqueda de datos en mayor detalle.



Figura 4.1: Telescopio Parkes localizado en New South Wales, Australia. Posee una antena parabólica con diámetro de 64 metros. Imagen obtenida de <https://www.atnf.csiro.au/outreach/visiting/parkes/index.html>.

4.1. Selección de fuentes y características de los datos

En la selección de fuentes tomamos en cuenta que el período del pulsar fuese superior a los 100[ms] para poder implementar el método de búsqueda general y no recurrir a procesos más especializados.

Los candidatos seleccionados para la búsqueda de pulsares son:

- PSR J1456-6843
- PSR J1559-4438
- PSR J0835-4510

1

Nombre del púlsar	Período [ms]	DM	S400	S1400
PSR J1456-6843	263.3768148933	8.6	350	80
PSR J1559-4438	257.00560976508	56.1	110	40
PSR J0835-4510	89.328385024	67.99	5000	1100

Tabla 4.1: Características de las fuentes. S400 y S1400 son las densidades de flujo promedio de los pulsares observadas a 400[MHz] y 1400 [MHz] en mJy. Datos obtenidos del catálogo de pulsares de la ATNF disponible en línea

4.1.1. Datos descargados de la ATNF

Todos los datos que se muestran a continuación son archivos descargados desde la base de datos de la ATNF disponible en línea. Las observaciones fueron hechas a los 1400 [MHz] (banda UHF), la extensión de los formatos descargados en línea es .sf.

¹Aunque el pulsar J0835-4510 tiene un periodo menor de 100[ms] es seleccionado, debido a que es uno de los más observados.

4.1 Selección de fuentes y características de los datos

```
1: From the PSRFITS file 'SM003_0741.sf':
    Telescope = Parkes
    Observer =
    Source Name = J1456-6843
    Frontend = MULTI
    Backend = FB_1BIT
    Project ID = P309
    Obs Date String = 1998-06-22T13:07:22
    MJD start time (DATE-OBS) = 50986.54678240740741
    MJD start time (STT *) = 50986.54674768518518
    RA J2000 = 14:56:00.1020
    RA J2000 (deg) = 224.000425
    Dec J2000 = -68:43:37.935
    Dec J2000 (deg) = -68.72720416666667
    Tracking? = True
    Azimuth (deg) = 197.245
    Zenith Ang (deg) = 39.949
    Polarization type = LIN
    Number of polns = 2 (summed)
    Polarization order = AA+BB
    Sample time (us) = 250
    Central freq (MHz) = 1374
    Low channel (MHz) = 1231.5
    High channel (MHz) = 1516.5
    Channel width (MHz) = 3
    Number of channels = 96
    Total Bandwidth (MHz) = 288
    Beam = 0 of 1
    Beam FWHM (deg) = 0.240
    Spectra per subint = 4096
    Starting subint = 0
    Subints per file = 117
    Spectra per file = 479232
    Time per subint (sec) = 1.024
    Time per file (sec) = 119.808
    bits per sample = 1
    bytes per spectra = 96
    samples per spectra = 96
    bytes per subint = 393216
    samples per subint = 393216
    zero offset = 0
    Invert the band? = True
PSRFITS Specific info:
    HDUs = primary, HISTORY, SUBINT
    FITS typecode = 11
    DATA column = 20
    Apply scaling? = False
    Apply offsets? = False
    Apply weights? = False
```

Figura 4.2: Información principal del pulsar J1456-6843 del archivo SM003_0741.sf que contiene los datos necesarios para poder llevar a cabo el procesamiento de los datos. Entre los datos más importantes destaca el Telescopio, la posición, el tiempo de muestreo, frecuencia central, ancho de banda, número de canales y ancho por canal

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

```
1: From the PSRFITS file 'S58110_1.sf':
  Telescope = Parkes
  Observer =
  Source Name = J1559-4438
  Frontend = UNKNOWN
  Backend = FB_1BIT
  Project ID = P050
  Obs Date String = 1994-00-00T09:25:51
  MJD start time (DATE-OBS) = 0.39295138888889
  MJD start time (STT *) = 49540.39287037037037
  RA J2000 = 15:59:41.5050
  RA J2000 (deg) = 239.9229375
  Dec J2000 = -44:38:45.764
  Dec J2000 (deg) = -44.6460455555556
  Tracking? = True
  Azimuth (deg) = 128.3249
  Zenith Ang (deg) = 22.5094
  Polarization type = LIN
  Number of polns = 2 (summed)
  Polarization order = AA+BB
  Sample time (us) = 1200
  Central freq (MHz) = 1412.26
  Low channel (MHz) = 1254.76000976562
  High channel (MHz) = 1569.76000976562
  Channel width (MHz) = 5
  Number of channels = 64
  Total Bandwidth (MHz) = 320
  Beam = 0 of 1
  Beam FWHM (deg) = 0.750
  Spectra per subint = 6144
  Starting subint = 0
  Subints per file = 19
  Spectra per file = 116736
  Time per subint (sec) = 7.3728
  Time per file (sec) = 140.0832
  bits per sample = 1
  bytes per spectra = 64
  samples per spectra = 64
  bytes per subint = 393216
  samples per subint = 393216
  zero offset = 0
  Invert the band? = False
  PSRFITS Specific info:
    HDUs = primary, HISTORY, SUBINT
    FITS typecode = 11
    DATA column = 20
    Apply scaling? = False
    Apply offsets? = False
    Apply weights? = False
```

Figura 4.3: Información principal del pulsar J1559-4438 del archivo S58110_1.sf que contiene los datos necesarios para poder llevar a cabo el procesamiento de los datos. Entre los datos más importantes destaca el Telescopio, la posición, el tiempo de muestreo, frecuencia central, ancho de banda, número de canales y ancho por canal

4.1 Selección de fuentes y características de los datos

```
I: From the PSRFITS file 's130127_180309.sf':
  Telescope = Parkes
  Observer = GH,
  Source Name = J0835-4510
  Frontend = MULTI
  Backend = PDFB3
  Project ID = P737
  Obs Date String = 2013-01-27T18:03:10
  MJD start time (DATE-OBS) = 56319.75219907407407
  MJD start time (STT *) = 56319.75231481481751
  RA J2000 = 08:35:20.610
  RA J2000 (deg) = 128.835875
  Dec J2000 = -45:10:34.800
  Dec J2000 (deg) = -45.17633333333333
  Tracking? = True
  Azimuth (deg) = 236.7726
  Zenith Ang (deg) = 45.094
  Polarization type = LIN
  Number of polns = 2 (summed)
  Polarization order = AA+BB
  Sample time (us) = 100
  Central freq (MHz) = 1369
  Low channel (MHz) = 1241
  High channel (MHz) = 1496.5
  Channel width (MHz) = 0.5
  Number of channels = 512
  Total Bandwidth (MHz) = 256
  Beam = 0 of 1
  Beam FWHM (deg) = 0.248
  Spectra per subint = 2048
  Starting subint = 0
  Subints per file = 670
  Spectra per file = 1372160
  Time per subint (sec) = 0.2048
  Time per file (sec) = 137.216
  bits per sample = 1
  bytes per spectra = 512
  samples per spectra = 512
  bytes per subint = 1048576
  samples per subint = 1048576
  zero offset = 0.5
  Invert the band? = True
  PSRFITS Specific info:
    HDUs = primary, HISTORY, BANDPASS, SUBINT
    FITS typecode = 11
    DATA column = 20
    Apply scaling? = True
    Apply offsets? = True
    Apply weights? = False
```

Figura 4.4: Información principal del pulsar J0835-4510 contenida en el archivo s1380127_180309.sf que contiene los datos necesarios para poder llevar a cabo el procesamiento de los datos. Entre los datos más importantes destaca el Telescopio, la posición, el tiempo de muestreo, frecuencia central, ancho de banda, número de canales y ancho por canal

Podemos ver que cada archivo posee tiempos de muestreo diferentes, con números de canales diferentes que dependen del backend que se use en cada observación (en los datos no se especifica que backend se está ocupando pero el telescopio Parkes puede tener diferentes configuraciones).

4.2. PSRSOFT resultados

El primer programa que usaremos para procesar los datos crudos es PSRSOFT. Los datos serán primero convertidos en archivos con extensión .fil como se explicó en el capítulo 3.

4.2.1. PSR J1456-6843

Haciendo uso del método mostrado en el Capítulo 3, procesamos los datos de la fuente J1456-6843 como se muestra en la Figura 4.5

```
$ dedisperse_all SM0003_0741.fil -d 0 100 -g 100000
$ file="SM0003_0741.fil"
$ for timfile in ${file}.fil.*.tim ; do seek $ timfile --fftw -head ;
done $ cat ${file}.fil.*.prd > ${file}.prd
$ ph-best SM0003_0741.prd SM0003_0741 -- maxresults 50
```

El resultado es una lista de los mejores candidatos como la que se muestra a continuación:

#nombredelarchivo	SNR _r	P [ms]	DMI
SM0003_0741_001.phcx.gz	162.600000	263.411681171	10.8525321
SM0003_0741_002.phcx.gz	30.800000	263.105993741	72.3872531
SM0003_0741_003.phcx.gz	27.900000	2106.103037521	72.3872531
SM0003_0741_004.phcx.gz	27.300000	20.003549251	0.0000001
SM0003_0741_005.phcx.gz	26.300000	263.859096581	10.8525321
SM0003_0741_006.phcx.gz	24.300000	262.801014991	10.8525321
SM0003_0741_007.phcx.gz	20.600000	856.067802681	10.8525321
SM0003_0741_008.phcx.gz	19.300000	473.959455251	0.8234991
SM0003_0741_009.phcx.gz	17.900000	966.095634971	10.8525321
SM0003_0741_010.phcx.gz	17.200000	737.719496831	30.9909651

Figura 4.5: Lista de los mejores candidatos encontrados en los datos de observación del pulsar J1456-6843. La lista ha sido recortado a los 10 mejores candidatos a pulsares.

Haciendo una pequeña modificación al script que se muestra en el capítulo 3 y cambiamos el nombre del documento, reescribiendo el script queda como:

```
file="SM0003_0741"
```

```
while read line ; do
filename=$(echo $line | cut -f1 -d' ')
cand=$(echo $filename | cut -f1 -d'.')
echo $cand
period=$(echo $line | cut -f3 -d' ')
period=$(echo $period/1000|bc -l)
echo $period
dm=$(echo $line | cut -f4 -d' ')
echo $dm

dspsr ${file}.fil -c ${period} -D ${dm} -L 10 -t 4 -U 1 -e subint
psradd *.subint -o ${cand}.ar
rm *.subint
pam --setnchn 8 -m ${cand}.ar

pdmp -input-phcx ${cand}.phcx.gz -output-phcx ${cand}.phcx ${cand}.ar
gzip -f ${cand}.phcx
ph-view-phcx ${filename} --imageoutput
done < ${file}.lis
```

Para ejecutar el script anterior desde la consola de Linux escribimos:

```
$ bash script
```

El resultado de ejecutar el script anterior son imágenes .png donde podemos ver los perfiles que se construyen a partir del archivo .fil.

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

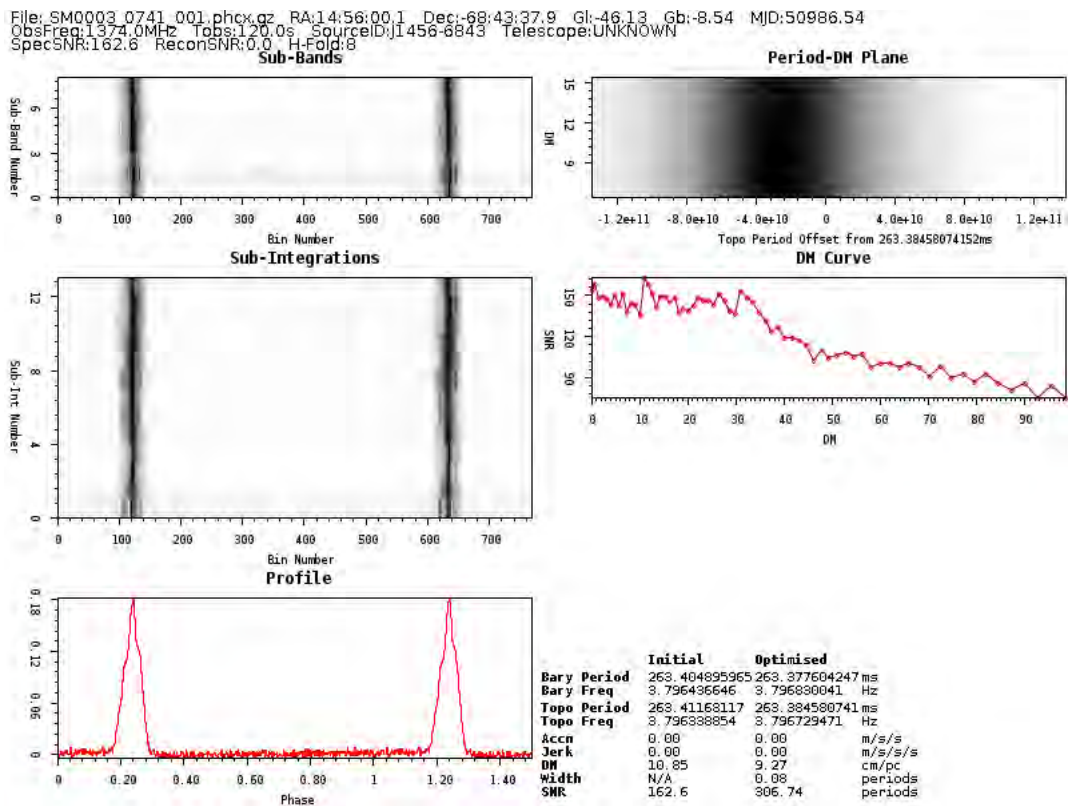


Figura 4.6: En la Figura 3.16 observamos la integración del perfil del pulsar J1456-6843, que tiene una $DM=10.85$ y $P=263.40$ [ms].

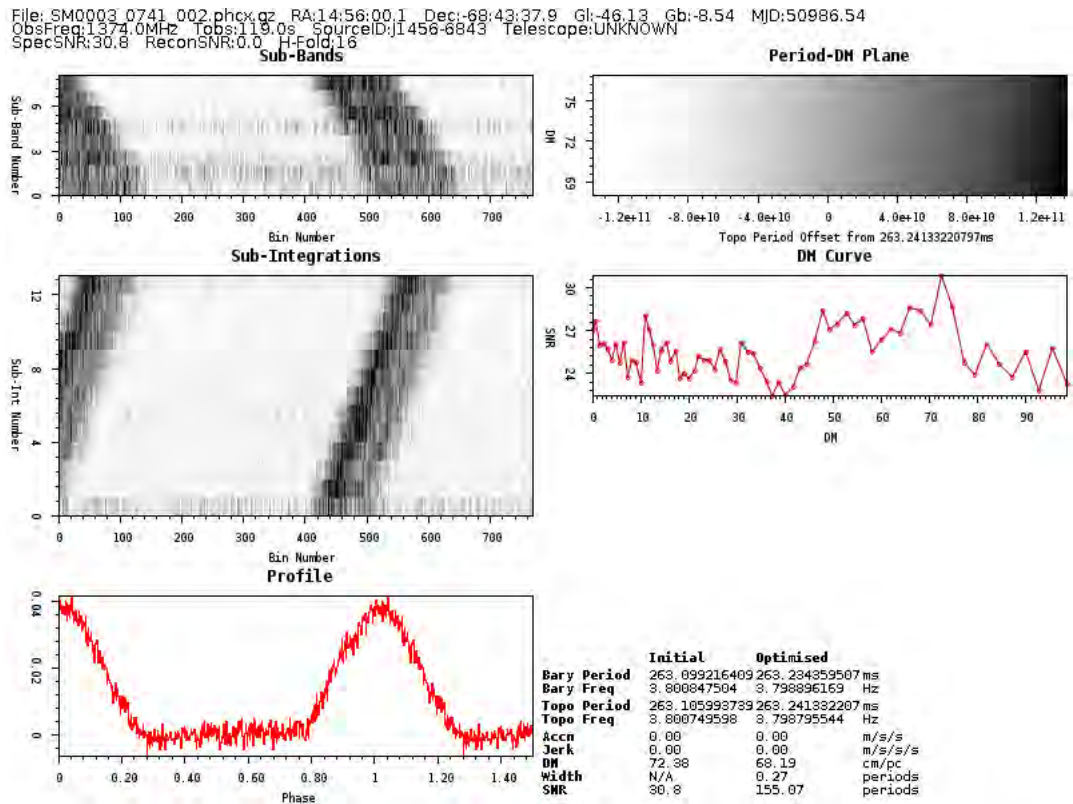


Figura 4.7: En la Figura 3.17 observamos la integración del perfil del segundo candidato para J1456-6843 con DM=72.38 P=263.09[ms]

4.2.2. PSR J1559-4438

Haciendo uso del método mostrado en el Capítulo 3, procesamos los datos de la fuente J1559-4438; usamos el archivo S58110_1.fil.

```
$ dedisperse_all S58110_1.fil -d 0 100 -g 100000
$ file="S58110_1.fil"
$ for timfile in ${file}.fil.*.tim ; do seek $ timfile --fftw -head ;
done $ cat ${file}.fil.*.prd > ${file}.prd
$ ph-best S58110_1.prd S58110_1 -- maxresults 50
```

Una vez de-dispersados los datos, se crea una lista de los posibles candidatos como se muestra en la Figura 4.8

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

#nombre del archivo	SNR _I	I	P [ms] _I	I	DM
S58110_1_001.phcx.gz	332.900000	I	257.06300156	I	55.365234
S58110_1_002.phcx.gz	71.700000	I	7.40750858	I	3.812527
S58110_1_003.phcx.gz	69.800000	I	256.68936348	I	3.812527
S58110_1_004.phcx.gz	55.900000	I	1799.16534015	I	22.881117
S58110_1_005.phcx.gz	51.000000	I	20.00399446	I	3.812527
S58110_1_006.phcx.gz	40.900000	I	2058.13332205	I	22.881117
S58110_1_007.phcx.gz	34.200000	I	2054.77241848	I	22.881117
S58110_1_008.phcx.gz	32.400000	I	565.68306143	I	55.365234
S58110_1_009.phcx.gz	31.700000	I	578.29207090	I	46.841473
S58110_1_010.phcx.gz	31.100000	I	835.31076908	I	46.841473

Figura 4.8: Lista de candidatos para el pulsar J1559-4438. Se muestran los mejores diez candidatos.

Haciendo una pequeña modificación al script que se muestra en el capítulo 3 y cambiamos el nombre del archivo el script queda como:

```
file="S58110_1"
while read line ; do
filename=$(echo $line | cut -f1 -d' ')
cand=$(echo $filename | cut -f1 -d'.')
echo $cand
period=$(echo $line | cut -f3 -d' ')
period=$(echo $period/1000|bc -l)
echo $period
dm=$(echo $line | cut -f4 -d' ')
echo $dm
dspsr ${file}.fil -c ${period} -D ${dm} -L 10 -t 4 -U 1 -e subint
psradd *.subint -o ${cand}.ar
rm *.subint
pam --setnchn 8 -m ${cand}.ar
pdmp -input-phcx ${cand}.phcx.gz -output-phcx ${cand}.phcx ${cand}.ar
gzip -f ${cand}.phcx
ph-view-phcx ${filename} --imageoutput
done < ${file}.lis
```

Para ejecutar el script anterior desde la consola de Linux escribimos:

```
$ bash script
```

El resultado de ejecutar el script anterior son imágenes .png donde podemos ver los perfiles que se construyen con PSRSOFT a partir del archivo .fil.

Para el primer candidato de la lista con una $DM=53.35$ y un $P=257.04$ [ms] obtenemos la información que se presenta en la Figura 4.9.

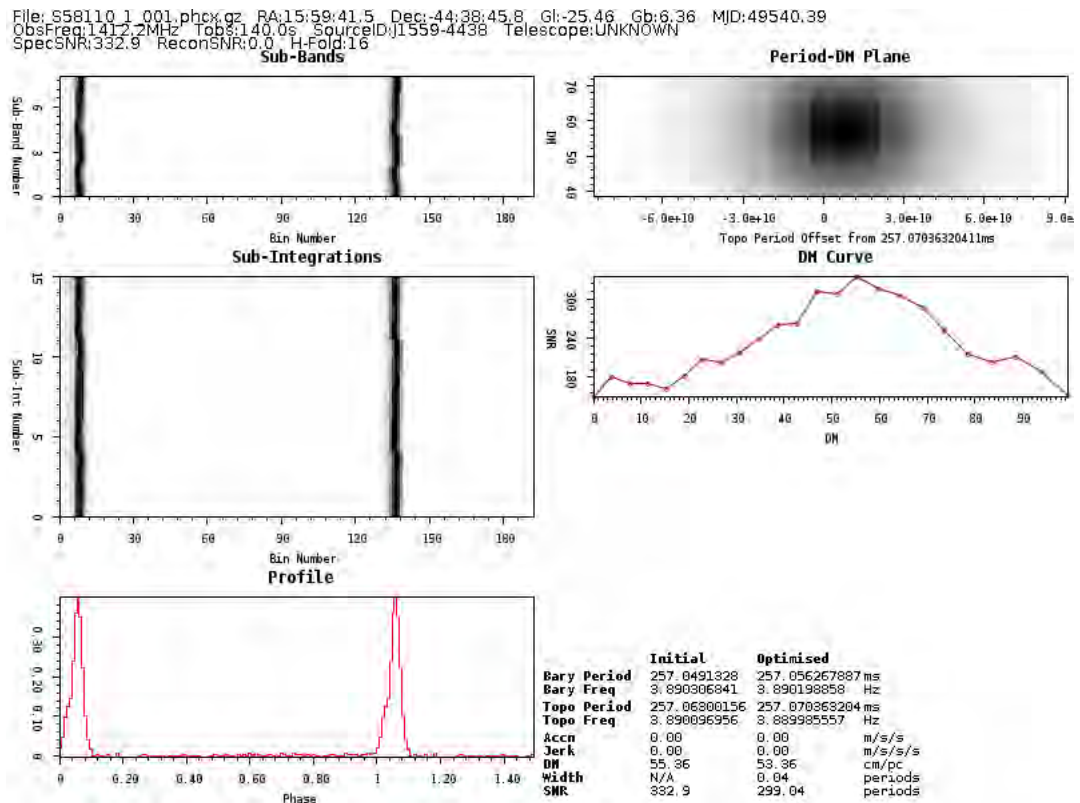


Figura 4.9: Datos obtenidos del primer candidato de la lista para J1559-4438 que posee una $DM=55.35$ y un $P=257.04$ [ms].

En la lista el segundo candidato posee una $DM=22.08$ y $P=1799.06$ [ms]. Generamos la siguiente información:

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

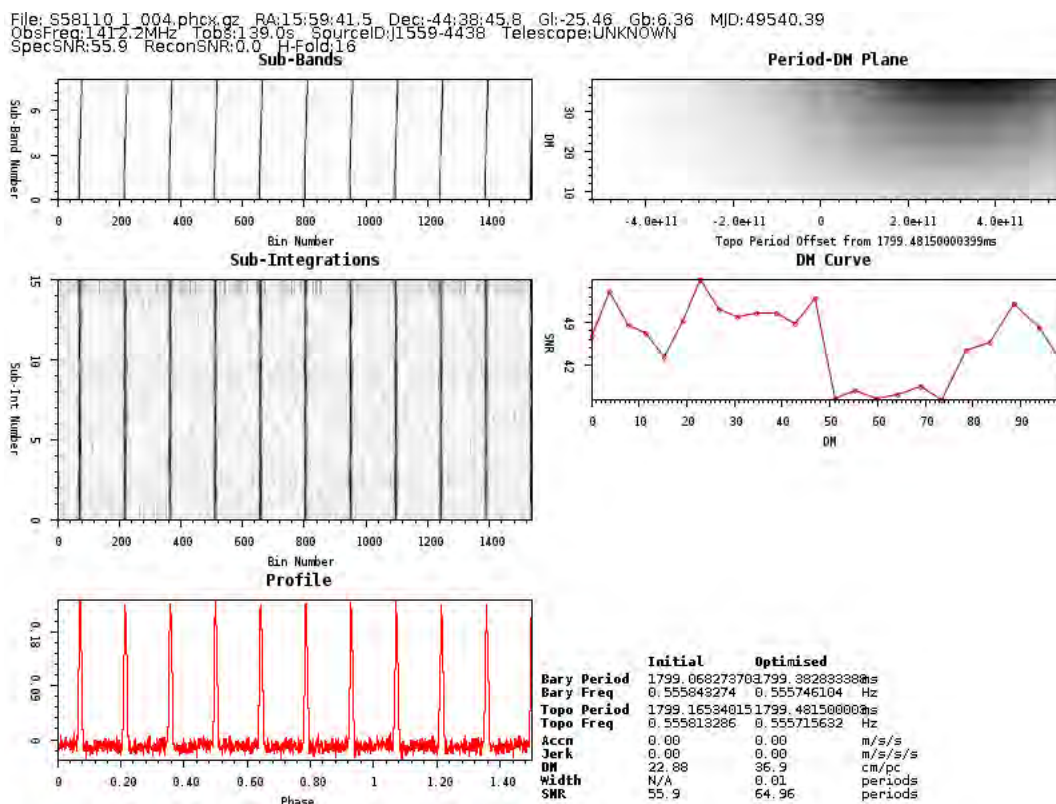


Figura 4.10: Datos del segundo candidato de la lista para J1559-4438 que posee una $DM = 22.08$ y un $P = 1799.06$ [ms].

4.2.3. PSR J0835-4510

Para el pulsar J0835-4510 o el pulsar Vela realizamos el mismo proceso que se mostró en el Capítulo 3 y uso el archivo s130127_180309.fil como archivo principal.

```
$ dedisperse_all s130127_180309.fil -d 0 100 -g 100000
$ file="S58110_1.fil"
$ for timfile in ${file}.fil.*.tim ; do seek $ timfile --ftw -head ;
done $ cat ${file}.fil.*.prd > ${file}.prd
$ ph-best s130127_180309.prd s130127_180309 -- maxresults 50
```

Una vez de-dispersados los datos, se crea una lista de los posibles candidatos como la siguiente:

#nombredelarchivo	I	SNRi	P [ms]	DM
s130127_180309_001.phcx.gz	684.100000	89.37956968	68.673531	
s130127_180309_002.phcx.gz	127.000000	178.80438640	47.158764	
s130127_180309_003.phcx.gz	99.800000	232.37945325	52.475998	
s130127_180309_004.phcx.gz	53.500000	20.02556199	0.390148	
s130127_180309_005.phcx.gz	51.700000	245.79299822	99.704849	
s130127_180309_006.phcx.gz	47.800000	267.93173755	36.188526	
s130127_180309_007.phcx.gz	42.900000	1.39652323	68.119537	
s130127_180309_008.phcx.gz	42.500000	804.75911146	47.158764	
s130127_180309_009.phcx.gz	39.900000	446.46971129	1.950702	
s130127_180309_010.phcx.gz	38.500000	1162.05822173	0.000000	

Figura 4.11: Lista de los candidatos del pulsar Vela

Haciendo una pequeña modificación al script que se muestra en el capítulo 3 y cambiamos el nombre del archivo el script queda como:

```
file="s130127_180309"
while read line ; do
filename=$(echo $line | cut -f1 -d' ')
cand=$(echo $filename | cut -f1 -d'.')
echo $cand
period=$(echo $line | cut -f3 -d' ')
period=$(echo $period/1000|bc -l)
echo $period
dm=$(echo $line | cut -f4 -d' ')
echo $dm
dspsr ${file}.fil -c ${period} -D ${dm} -L 10 -t 4 -U 1 -e subint
psradd *.subint -o ${cand}.ar
rm *.subint
pam --setnchn 8 -m ${cand}.ar
pdmp -input-phcx ${cand}.phcx.gz -output-phcx ${cand}.phcx ${cand}.ar
gzip -f ${cand}.phcx
ph-view-phcx ${filename} --imageoutput
done < ${file}.lis
```

Para ejecutar el script anterior desde la consola de Linux escribimos:

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

\$ bash script

El resultado de procesar los datos candidatos más prominentes se muestran en las Figuras 4.12 y 4.13.

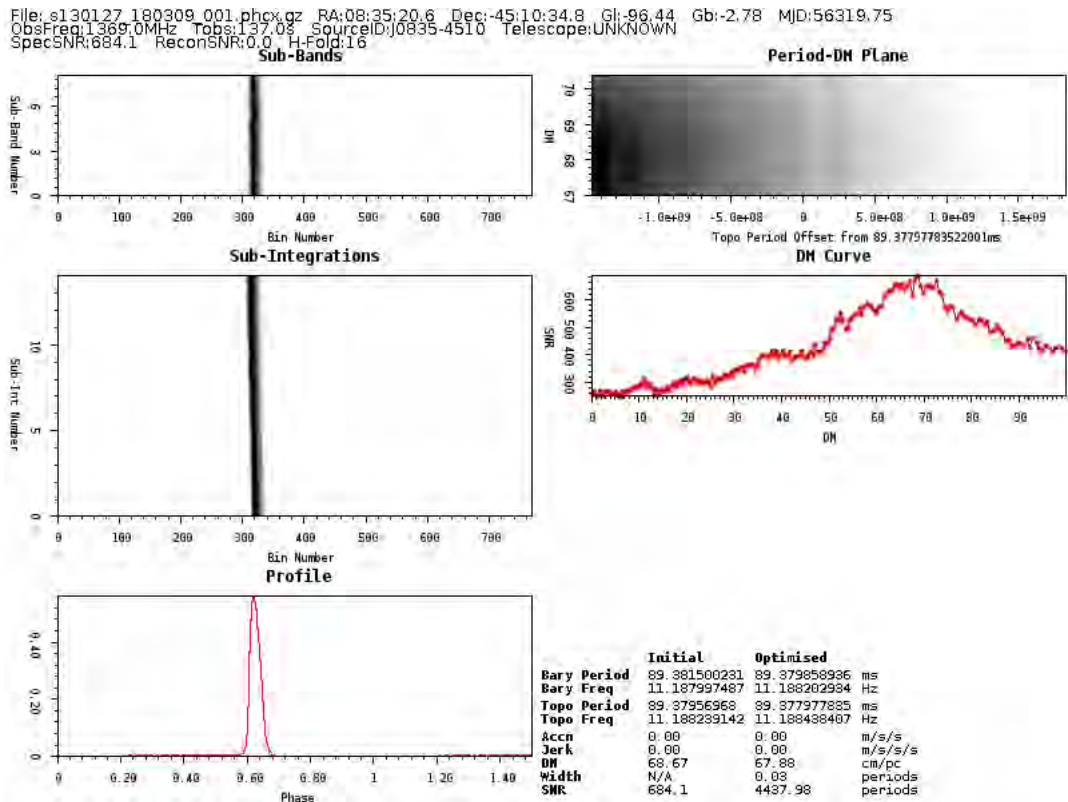


Figura 4.12: Resultado del primer candidato de la lista del pulsar Vela que posee un Período=89.3[ms] y una DM=68.67.

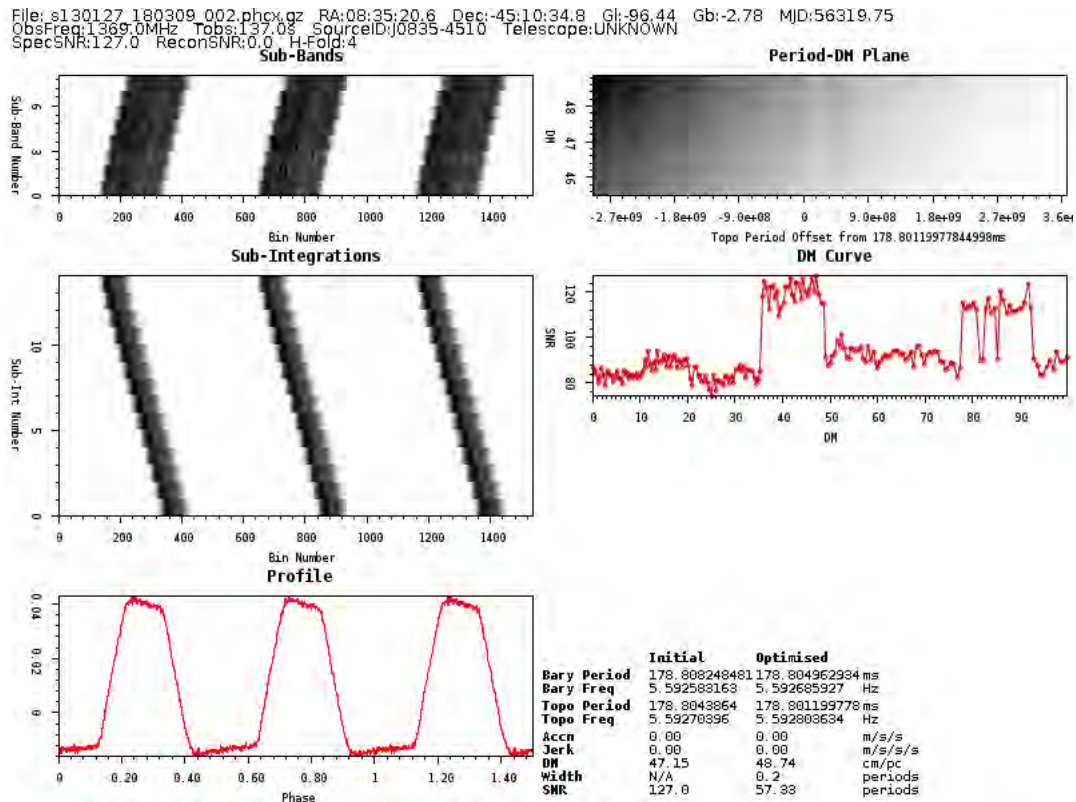


Figura 4.13: Resultado del segundo candidato de la lista del pulsar Vela que posee un Período=179.8[ms] y una DM=47.15.

4.3. PRESTO resultados

Aquí muestro el resultado de procesar los mismos datos que en la sección anterior, pero con el paquete de software PRESTO, desarrollado por S. Ransom.

4.3.1. PSR J1456-6843

Siguiendo el proceso del Capítulo 3, lo primero es buscar interferencia por radiofrecuencia:

```
$ rfifind -time 1.0 -o rfi_find SM0003_0741.fil
```

```
$ prepsubband -lodm 0.0 -dmstep 0.5 -numdms 100 -numout 132500
```

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

```
-mask rfifind_rfifind.mask -o disp SM0003_0741.fil
$ ls *.dat | xargs -n 1 realfft
$ ls *.fft | xargs -n 1 accelsearch -zmax 0
$ python ACCEL_sift.py > candidatos.txt
```

En este punto ya tenemos la lista de los mejores candidatos, generado por el programa ACCEL_sift.py.

#	file:candnum	DM	SNR	sigma	numharm	ipow	cpow	P[ms]
disp_DM5.00_ACCEL_0:1		5.00	126.08	62.74	8	2018.0	15464.8	263.419483
disp_DM13.50_ACCEL_0:4		13.50	14.35	17.52	2	162.3	205.1	58.524735
disp_DM7.00_ACCEL_0:3		7.00	13.24	17.13	4	164.0	170.4	98.768561
disp_DM6.00_ACCEL_0:5		6.00	9.15	10.71	2	64.8	90.0	47.885797
disp_DM75.00_ACCEL_0:3		75.00	9.97	10.04	4	64.4	15.7	112.862010
disp_DM6.50_ACCEL_0:6		6.50	13.63	7.78	8	52.6	137.8	40.522951
disp_DM56.00_ACCEL_0:4		56.00	8.08	7.17	4	37.8	36.3	75.498575
disp_DM3.50_ACCEL_0:7		3.50	12.19	5.22	8	32.3	1.0	68.857925
disp_DM13.00_ACCEL_0:7		13.00	11.89	5.08	8	31.3	10.8	8.495359

Figura 4.14: Lista de los candidatos con el archivo que se exploró.

Una vez con la lista de candidatos procedemos a obtener manualmente el perfil de cada candidato. A diferencia de PSRSOFT que es automático, aquí debemos generarlo uno por uno.

Para el primer candidato ejecutamos la siguiente sentencia:

```
$ prepfold -accelcand 1 -accelfile disp_DM5.00_ACCEL_0.cand
disp_DM5.00.dat
```

El resultado del primer candidato se muestra en la siguiente Figura 4.15.

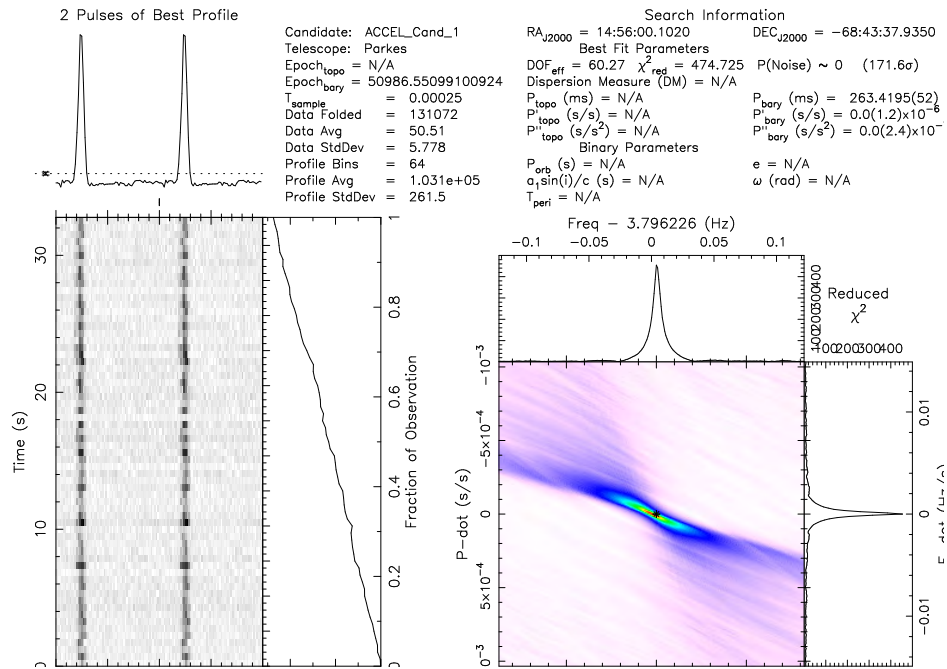


Figura 4.15: Primer candidato del pulsar J1456-6843 DM=5.0 P=263.41[ms]

Para el segundo candidato ejecutamos:

```
$ prepfold -accelcand 4 -accelfile disp_DM13.50_ACCEL_0.cand
disp_DM13.50.dat
```

El perfil del segundo candidato es

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

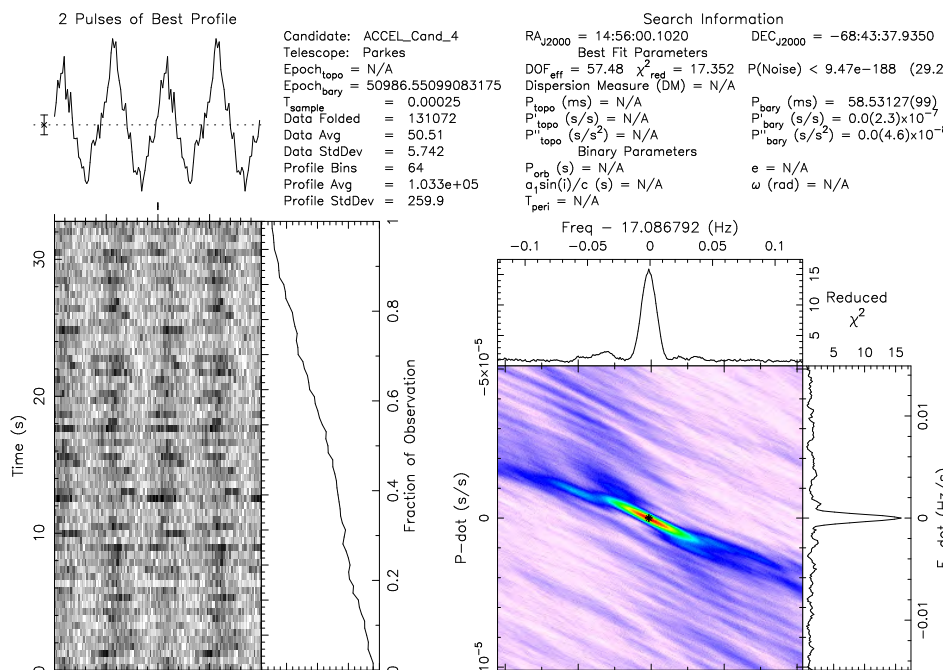


Figura 4.16: Segundo candidato del pulsar J1456-6843 DM=13.50 P=58.52[ms]

4.3.2. PSR J1559-4438

Siguiendo el proceso del Capítulo 3, lo primero es buscar interferencia por radiofrecuencia:

```
$ rfifind -time 0.5 -o rfi_find S58110_1.fil
$ prepsubband -lodm 0.0 -dmstep 0.5 -numdms 100 -numout 132500
-mask rfifind_rfifind.mask -o dedisp SM0003_0741.fil
$ ls *.dat | xargs -n 1 realfft
$ ls *.fft | xargs -n 1 accelsearch -zmax 0
$ python ACCEL_sift.py > candidatos.txt
```

En este punto ya tenemos la lista de los mejores candidatos, generado por el programa ACCEL_sift.py

#	file:candnum	DM	SNR	sigma	numharm	ipow	cpow	P[ms]
dedisp_DM54.50_ACCEL_0:1		54.50	182.07	90.60	8	4159.0	32308.9	257.073565
dedisp_DM59.50_ACCEL_0:3		59.50	52.01	39.46	8	821.7	184.8	73.445349
dedisp_DM56.50_ACCEL_0:7		56.50	33.32	36.04	4	671.7	1005.4	46.737213
dedisp_DM47.50_ACCEL_0:5		47.50	35.24	35.29	8	664.2	611.0	96.396470
dedisp_DM53.50_ACCEL_0:6		53.50	40.35	33.23	8	592.6	350.6	57.124893
dedisp_DM58.00_ACCEL_0:8		58.00	25.95	27.42	4	396.4	199.0	39.547320
dedisp_DM55.50_ACCEL_0:13		55.50	16.88	19.93	2	207.9	128.7	13.529036
dedisp_DM53.00_ACCEL_0:16		53.00	11.05	15.45	1	123.0	122.8	11.176326
dedisp_DM59.50_ACCEL_0:14		59.50	10.72	14.99	1	116.0	115.9	12.852639
dedisp_DM54.00_ACCEL_0:17		54.00	12.47	14.94	4	128.0	158.2	30.846833
dedisp_DM54.50_ACCEL_0:12		54.50	13.88	14.83	2	118.3	106.3	15.121255

Figura 4.17: Lista de los candidatos con el archivo que se exploró.

Para obtener el perfil del primer candidato tenemos que ejecutar:

```
$ prepfold -accelcand 1 -accelfile disp_DM54.50_ACCEL_0.cand
disp_DM54.50.dat
```

El resultado del primer candidato es:

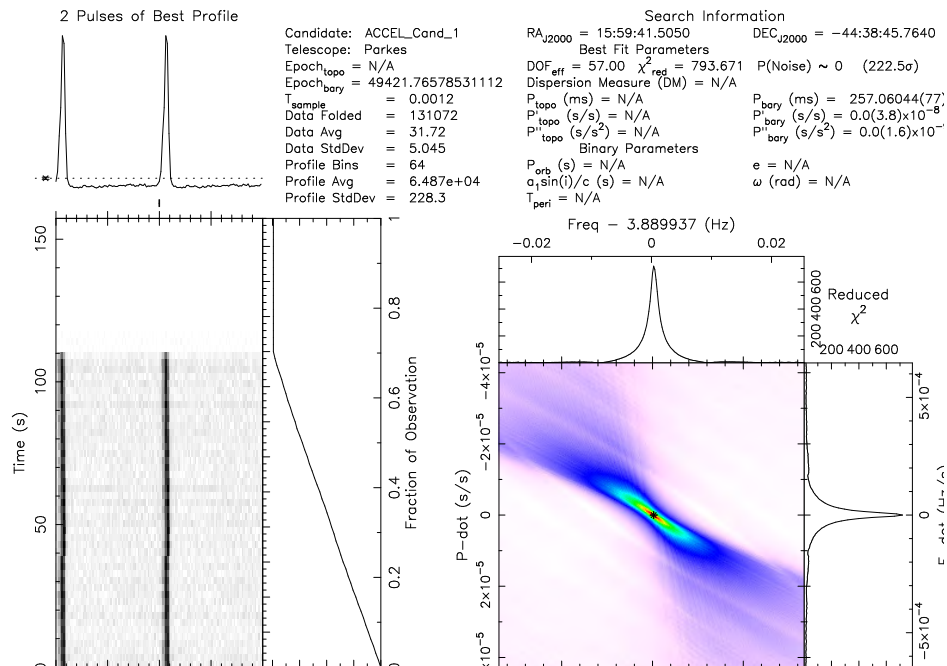


Figura 4.18: Primer candidato para el pulsar J1559-4438 DM=54.50 P=257.07[ms]

El segundo candidato se obtiene de la misma manera que el primero:

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

```
$ prepfold -accelcand 3 -accelfile dedisp_DM59.50_ACCEL_0.cand
dedisp_DM59.50.dat
```

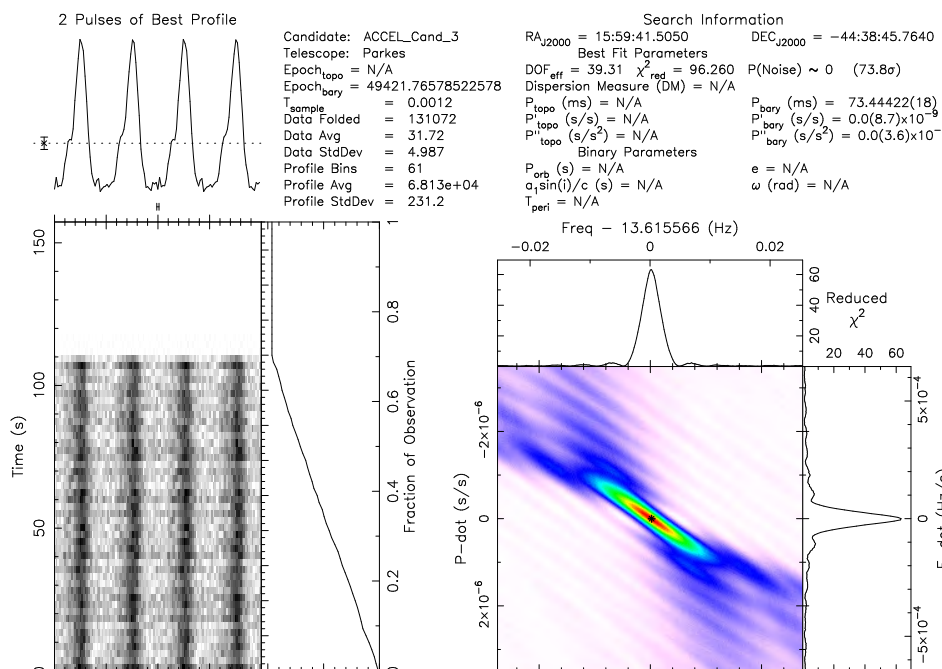


Figura 4.19: Segundo candidato del pulsar J1559-4438 DM=59.50 P=73.44

4.3.3. PSR J0835-4510

Para el pulsar Vela hacemos el mismo proceso que en los archivos anteriores. Lo primero es buscar interferencia por radiofrecuencia:

```
$ rfifind -time 0.5 -o rfi_find s130127_180309.fil
$ prepsubband -lodm 0.0 -dmstep 0.5 -numdms 100 -numout 132500
-mask rfifind_rfifind.mask -o prepvela SM0003_0741.fil
$ ls *.dat | xargs -n 1 realfft
$ ls *.fft | xargs -n 1 accelsearch -zmax 0
$ python ACCEL_sift.py > candidatos.txt
```

En este punto ya tenemos la lista de los mejores candidatos, generado por el programa ACCEL_sift.py

#	file:candnum	DM	SNR	sigma	numharm	ipow	cpow	P[ms]
prepvela_DM64.00_ACCEL_0:1		64.00	386.33	193.03	8	18696.0	147368.9	89.376054
prepvela_DM62.00_ACCEL_0:11		62.00	328.21	167.55	8	14101.0	107122.3	44.601044
prepvela_DM48.50_ACCEL_0:9		48.50	196.96	138.95	8	9715.1	1725.9	177.183452
prepvela_DM62.00_ACCEL_0:19		62.00	107.12	87.04	8	3842.7	4321.2	25.507589
prepvela_DM89.50_ACCEL_0:17		89.50	130.42	86.30	8	3778.5	399.4	22.318721
prepvela_DM67.50_ACCEL_0:5		67.50	111.43	82.67	8	3471.0	134.2	19.862276
prepvela_DM63.00_ACCEL_0:12		63.00	106.42	79.69	8	3228.4	5669.9	11.168181
prepvela_DM75.50_ACCEL_0:18		75.50	87.34	71.95	8	2640.4	1276.2	9.924512
prepvela_DM67.50_ACCEL_0:7		67.50	95.00	71.80	8	2629.5	6566.8	16.250814
prepvela_DM68.00_ACCEL_0:8		68.00	80.44	64.99	8	2162.0	3781.2	13.750829
prepvela_DM82.00_ACCEL_0:3		82.00	55.38	57.13	8	1680.6	1216.0	26.812527

Figura 4.20: Lista de los candidatos mas prominentes para el pulsar Vela.

Para obtener el perfil del primer candidato de la lista, ejecutamos la siguiente sentencia:

```
$ prepfold -accelcand 1 -accelfile disp_DM64.00_ACCEL_0.cand
disp_DM64.00.dat
```

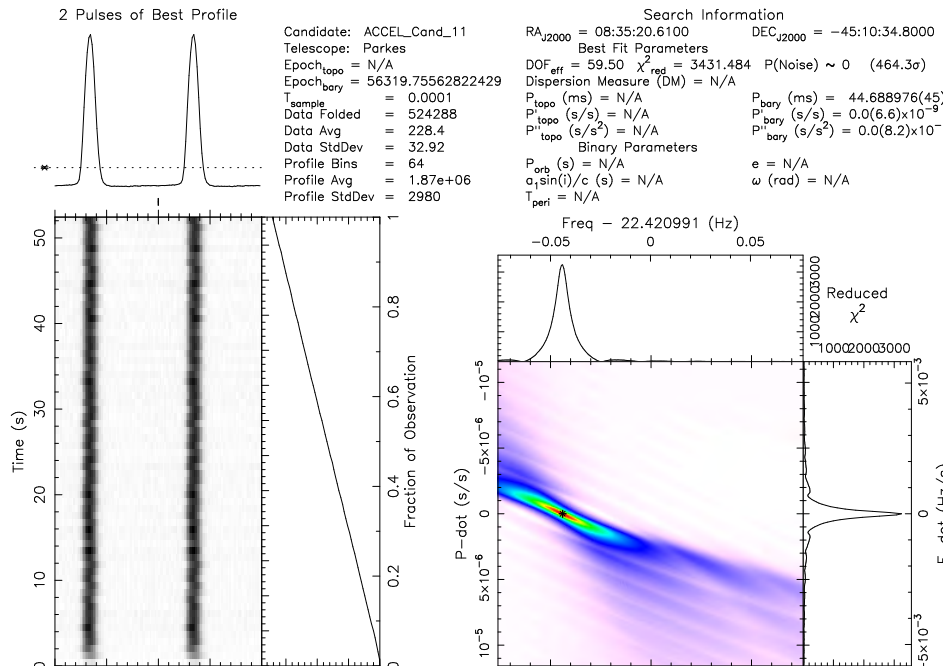


Figura 4.21: Primer candidato del pulsar Vela que posee una DM=64.00 y un P=89.37[ms]

Siguiendo el orden de la lista el perfil del segundo candidato se obtiene de la siguiente manera:

4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

```
$ prepfold -accelcand 11 -accelfile disp_DM62.00_ACCEL_0.cand
disp_DM62.00.dat
```

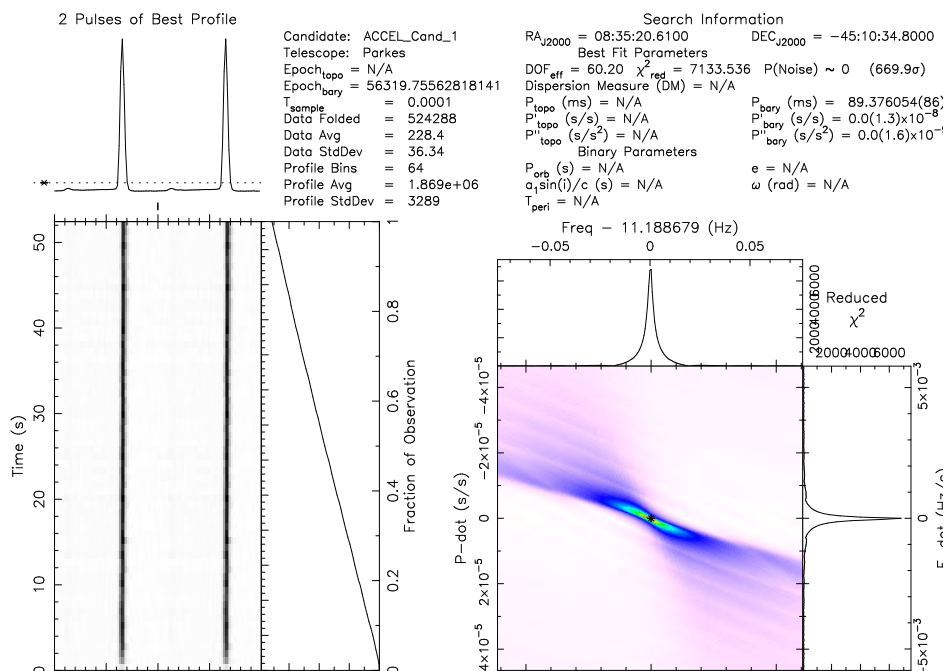


Figura 4.22: El segundo candidato de la lista posee una DM=62.00 P=44.60

4.4. Evaluación de PSRSOFT y PRESTO

Hasta este punto se muestra que usando PSRSOFT el procesamiento de datos para detección de pulsares es casi automático mientras que con el uso de PRESTO en gran parte del procesamiento de datos tiene que ser manual.

Nombre del púlsar	Período con PSRSOFT	Período con PRESTO
PSR J1456-6843	263.41168117 [ms]	263.419483[ms]
PSR J1559-4438	257.06300156 [ms]	257.073565 [ms]
PSR J0835-4510	89.37956968 [ms]	89.376054 [ms]

Tabla 4.2: Tabla comparativa de los valores de Período obtenidos con PSRSOFT y PRESTO

Nombre del púlsar	DM con PSRSOFT	DM con PRESTO
PSR J1456-6843	10.852532 [$cm^{-3}pc$]	5.0 [$cm^{-3}pc$]
PSR J1559-4438	55.365234 [$cm^{-3}pc$]	54.50 [$cm^{-3}pc$]
PSR J0835-4510	68.673531 [$cm^{-3}pc$]	64.0 [$cm^{-3}pc$]

Tabla 4.3: Tabla comparativa de los valores de DM obtenidos con PSRSOFT y PRESTO

En las tablas 4.2 y 4.3 se muestran los mejores perfiles obtenidos a partir del procesamiento de los datos con PSRSOFT y PRESTO. Se observa que los períodos obtenidos con PSRSOFT y PRESTO no son significativamente diferentes entre ellos y el que ya ha sido aceptado por la comunidad. Sin embargo, las DM que resultan del procesamiento de los datos difieren de manera significativa entre ambos programas y entre los valores que han sido estandarizados por la comunidad, esta diferencia en la DM repercute en la SNR de las señales, en la tabla 4.4 se muestra porque es necesario calcular de la manera más correcta posible la DM.

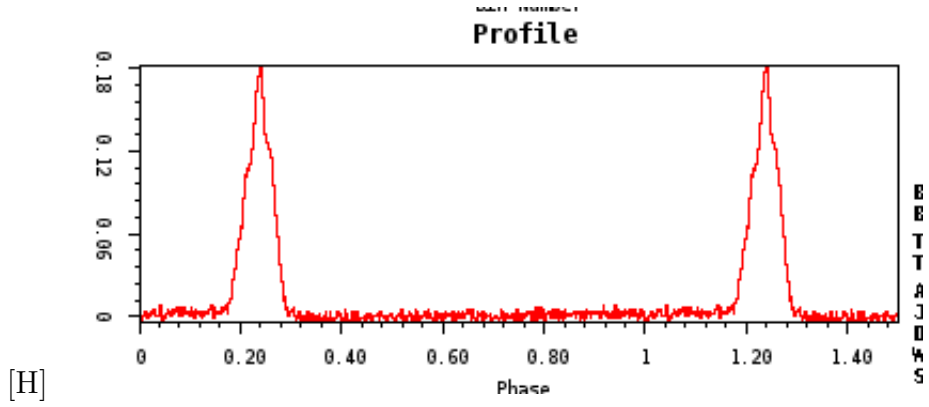
4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)

Nombre del púlsar	SNR con PSRSOFT	SNR con PRESTO
PSR J1456-6843	162	126
PSR J1559-4438	332	182
PSR J0835-4510	684	386

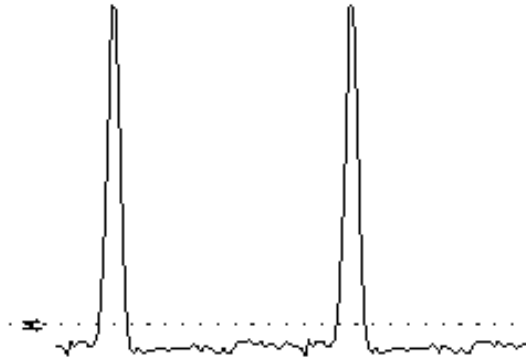
Tabla 4.4: Tabla comparativa de los valores de SNR obtenidos con PSRSOFT y PRESTO

Comparando los resultados obtenidos en la tabla 4.1 con las tablas 4.2 y 4.3 me atrevo a decir que PSRSOFT ha tenido mayor precisión con el cálculo de Período y DM, aún hay que hacer una tercera comparación que se debe hacer a la construcción del perfil hecha por ambos software, por lo que haciendo uso de datos de la EPN (European Pulsar Network), podemos comparar los perfiles del pulsar construidos por nosotros con aquellos que ya han sido previamente observados, como se observa a continuación.

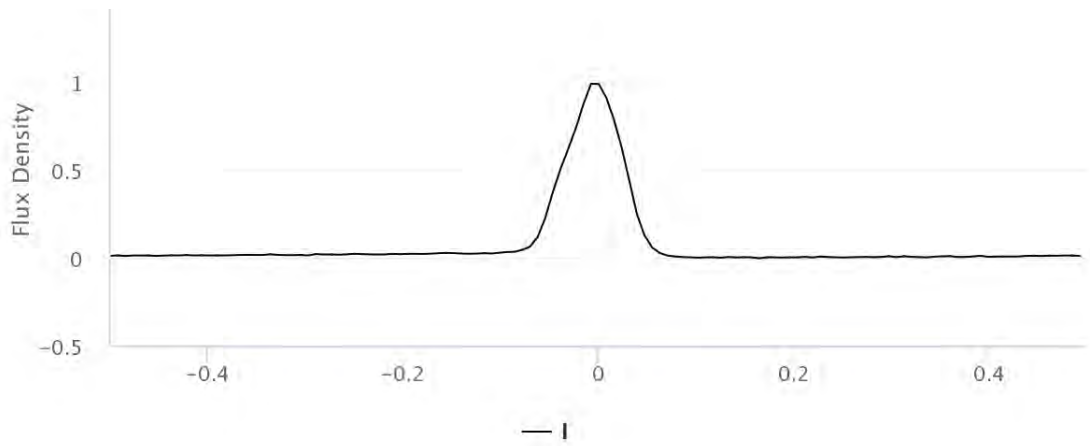
Observando los perfiles de señal obtenidos con los programas mostrados en esta tesis y aquellos que se descargaron de EPN, puedo decir que el programa PSRSOFT tiene mejor resolución en la construcción de los perfiles que PRESTO. Hasta este punto PSRSOFT se ha visto mejor que PRESTO, sin embargo este último posee características para procesar otro tipo de datos como BURST, pulsares de milisegundos o pulsares binarios, que con PSRSOFT es un poco más complicado, pero este tema no lo trataremos en la tesis.



(a) Perfil construido con PSRSOFT



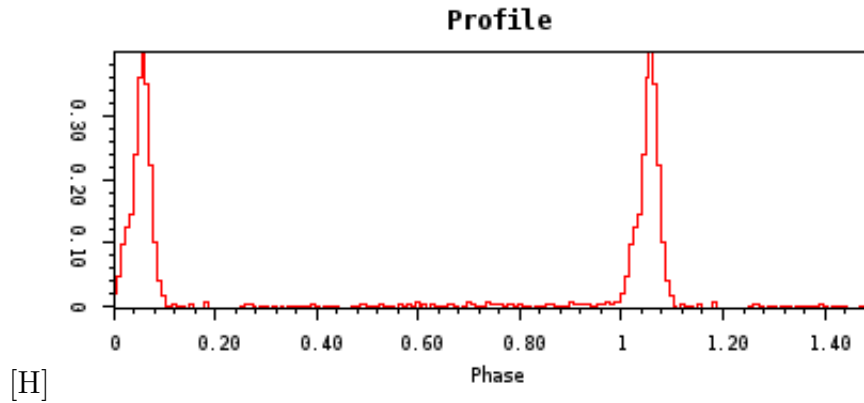
(b) Perfil construido con PRESTO



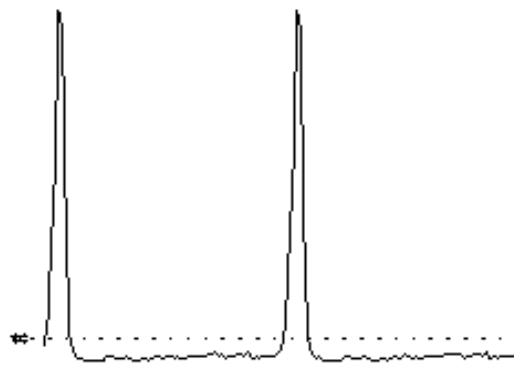
(c) Perfil obtenido de la EPN

Figura 4.23: Perfiles del pulsar J1456-6843 construidos con diferentes programas.

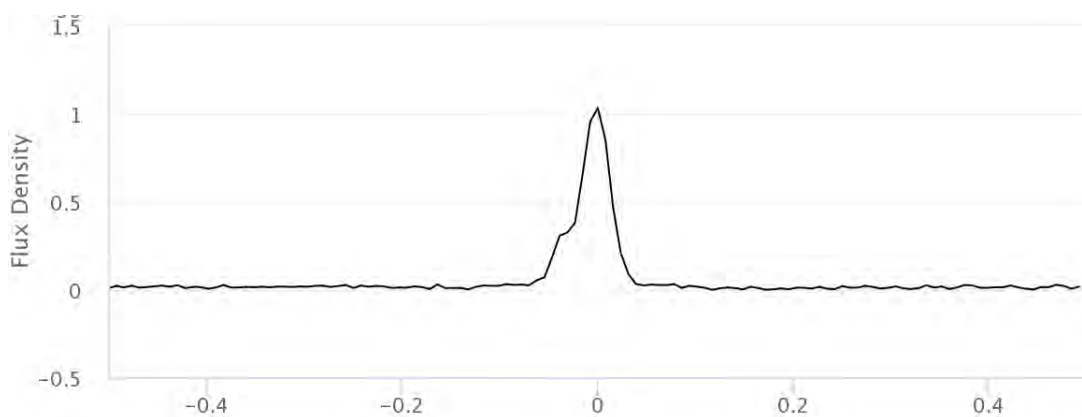
4. BÚSQUEDA DE RADIO-PULSARES CON DATOS DE LA ATNF (AUSTRALIAN TELESCOPE NATIONAL FACILITY)



(a) *Perfil construido con PSRSOFT*

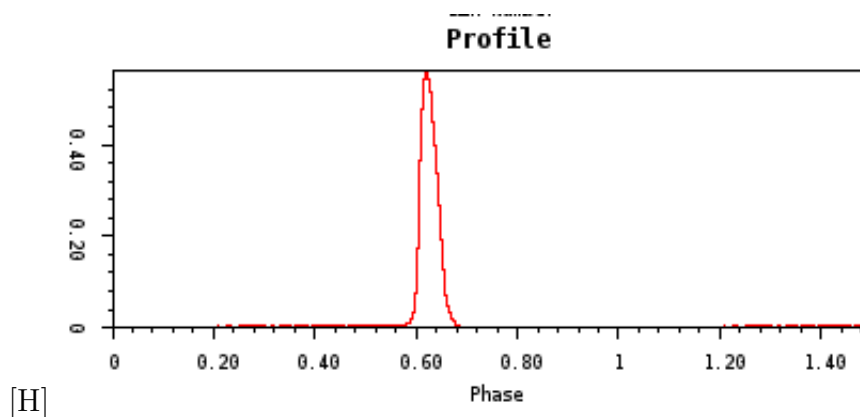


(b) *Perfil construido con PRESTO*

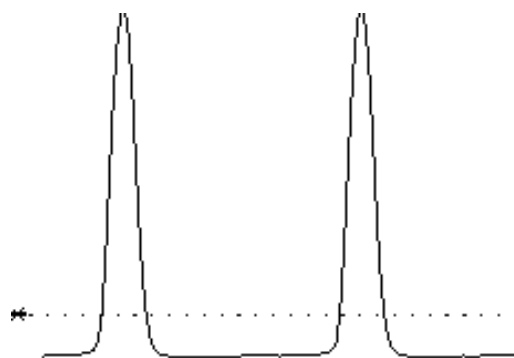


(c) *Perfil obtenido de la EPN*

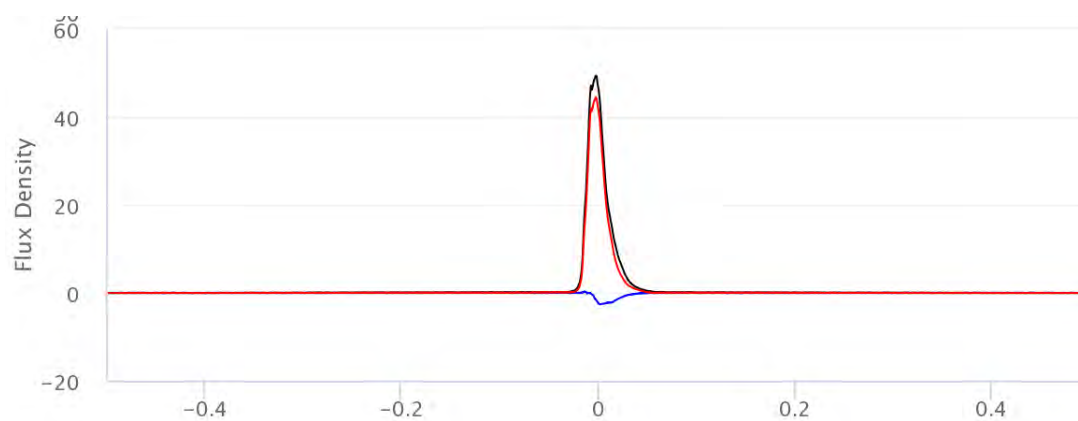
Figura 4.24: Perfiles del pulsar J1559-4438 construidos con diferentes programas.



(a) Perfil construido con PSRSOFT



(b) Perfil construido con PRESTO



(c) Perfil obtenido de la EPN

Figura 4.25: Perfiles del pulsar J0835-4510 contruidos con diferentes programas.

Conclusiones y trabajo a futuro

5.1. Conclusiones

Esta tesis se enfocó en el procesamiento de datos para la detección de radiopulsares, que son señales que tienen una forma única comparado a otras señales astronómicas debido a su naturaleza.

Para comprender mejor este tipo de fenómenos en el Capítulo 1 se explica el nacimiento de estas estrellas de neutrones y su mecanismo de emisión para de esta forma comprender su origen y su radiación. Se define la explicación del modelo *lighthouse*. Los pulsares poseen características que ayudan a entender el comportamiento del Universo que nos rodea.

Una vez que he comprendido como funciona un pulsar es momento de analizar las técnicas usadas para poder detectar estas señales. Como vimos en el Capítulo 2 el Universo no está vacío, por lo que cualquier señal que lo atraviese se verá afectada por lo que conocemos como MIE. Estas afectaciones debidas al Medio (que estudié muchas veces durante la carrera) provocan que la señal se atenúe y retarde; por estas razones al hacer observaciones en el espacio se tienen que calcular maneras de compensar estos problemas. En el Capítulo 2 hablo de dos técnicas para compensar el retardo de la señal debido al medio; estas técnicas se conocen como de-dispersión coherente e incoherente. Pero esto no es todo debido a la naturaleza periódica de la emisión de pulsares podemos recurrir a técnicas de Búsqueda de Periodicidad como es la Transformada de Fourier y a partir de aquí detectar dichas señales. La técnica que complementa la búsqueda de pulsares se conoce como Folding que sirve para construir un perfil de la señal emitida por el pulsar que se desee estudiar.

En el campo de la detección de los pulsares se han desarrollado una gran cantidad de algoritmos computacionales para implementar las técnicas mostradas en el Capítulo 2. Estos algoritmos tienen sus variantes y han sido adaptados a la

5. CONCLUSIONES Y TRABAJO A FUTURO

instrumentación que se tenga en el momento.

Los programas que se muestran en el Capítulo 3 son el trabajo de años en colaboración de varios investigadores y programadores que han optimizado y mejorado los algoritmos que ya se encontraban disponibles. Se muestra como se realiza el procesamiento de datos con el uso de estos dos programas y la detección de radio-pulsares con un archivo ejemplo descargado de la ATNF. Este observatorio mantiene la base de datos con mayor cantidad de observaciones y archivos disponibles al público.

En el capítulo 4 haciendo uso de datos captados por la ATNF, comparé y mostré el funcionamiento de PSRSOFT y PRESTO para la detección de radio-pulsares con archivos reales de observaciones con el radio-telescopio Parkes. Cada uno de los archivos corresponde a un pulsar diferente. Cada uno de los archivos posee tiempos de muestreo diferentes según sean las necesidades de la observación, número de canales (que depende del banco de filtros que se este usando para la observación). Estas diferencias en los archivos de las observaciones son importantes pues según sean estas características habrán algunas ligeras variaciones en los scripts que se muestran en el Capítulo 3 para cada programa.

PRESTO y PSRSOFT son software libre, por lo que su descarga e instalación requieren de Sistemas Operativos basados en UNIX. Ambos programas fueron instalados en Debian Jessie y se muestra el procedimiento de instalación en el Apéndice B. Al ser software libre ambos programas cuentan con la facilidad de poder ser modificados por el usuario si así se requiere, en esta tesis no llevamos a cabo ninguna modificación al código fuente porque no fue necesario.

PSRSOFT es un software de fácil instalación pero que requiere demasiadas librerías adicionales debido a que maneja muchos lenguajes de programación, lo que no es un problema significativo. Al contrario de PSRSOFT la instalación de PRESTO exige una cantidad menor de librerías pero un mejor manejo de UNIX.

En cuanto a la manera para procesar datos con PSRSOFT resulta ser más sencillo para el usuario porque no requiere de aprender muchas sentencias y no es necesario tener un conocimiento profundo de la plataforma UNIX para llevar a cabo todo el proceso. Incluso es posible crear un script como el mostrado en Capítulo 3 para automatizar algunas de las tareas. Sin embargo PRESTO exige un conocimiento más profundo de UNIX y del programa en si para poder ejecutar desde la terminal varios de los programas necesarios para el procesamiento de los datos para la detección de pulsares. Esto no es un grave problema pero si el usuario no tiene gran conocimiento en UNIX y en programación yo recomendaría hacer uso de PSRSOFT.

Respecto al tiempo de procesamiento; ambos presentan características muy similares y consumieron una cantidad semejante de recursos al procesar estos datos.

Las tablas mostradas en Capítulo 4 se observa que PSRSOFT y PRESTO

tienen la misma precisión al obtener los períodos de los candidatos a pulsares, sin embargo se observa que cuanto al cálculo de la Medida de Dispersión PSRSOFT es superior a PRESTO. El error en DM obtenida por ambos software afecta directamente a la SNR de los candidatos, por lo que aquel que tenga una SNR superior es considerado más preciso. En este caso PSRSOFT ha demostrado ser más eficiente en cuanto a la búsqueda de pulsares se refiere.

Otro punto a observar es que al construir el perfil de las señales con uso de Folding en los programas y compararlos visualmente con aquellos obtenidos en la base de datos de la EPN se observa que PSRSOFT es un poco superior pues los perfiles creados con este programa se asemejan más que los creados a partir de PRESTO.

Si bien se mostró que cada uno de los programas detectó diferentes candidatos, aquellos que fueron más fuertes se mostraron como el candidato ideal para cada una de nuestras observaciones. Esto se debe a que cada quien ha implementado algoritmos en sus programas. También se muestra que cada señal tiene una firma diferente; es decir, tiene un perfil que lo distingue de otros. Esto es importante porque algunas veces al realizar búsqueda de pulsares podemos observar mismos períodos y al hacer la correspondencia con el perfil puede ser alguna señal espuria. El perfil es una manera de distinguir también de que tipo de pulsar hablamos.

Durante la realización de mi tesis observe la relación tan estrecha que existe entre la Ingeniería y la Astronomía pues esta última requiere de mucha instrumentación para llevar a cabo su observaciones. Aprendí muchas cosas de la radioastronomía pero lo más importante para mi fue que pude aplicar muchos de mis conocimientos adquiridos a lo largo de mi carrera en situaciones diversas que se presentaron durante la realización de mi tesis.

5.2. Trabajo a Futuro

La siguiente etapa es poder crear nuestros propios datos a partir de observaciones a frecuencias menores entre 100 y 400 [MHz], esto se logrará a partir de la creación de un back-end que posea las características de los que ya se encuentran en operación, como son los del Jodrell Bank, del telescopio Parkes o del GBT. Aunque en la actualidad la comunidad de amateur de astronomía ha desarrollado programas para la detección de pulsares, estos no nos son del todo útiles, porque “esconden” mucha de su información y no son compatibles con los formatos de archivos que usamos en esta tesis.

De la necesidad de poder hacer nuestra propia detección de pulsares surge la propuesta para el desarrollo de un back-end con el que seamos capaces de guardar nuestros datos de observaciones consiste en un programa elaborado en

5. CONCLUSIONES Y TRABAJO A FUTURO

GNURADIO con el uso de un USRP y el uso de una FPGA con el programa PSRDADA para guardar los datos en archivos en formato PSRFITS y después realizar el procesamiento de los datos con los programas que se presentaron en esta tesis.

Los retos para realizar un back-end con las características suficientes para poder crear nuestros propios datos de observación y procesarlos con PSRSOFT y PRESTO son bastante amplios, pues se trata de hacer una reproducción del trabajo que otros investigadores han venido desarrollando a lo largo de años, esto no hace más complicado el reto sino más interesante.

El reproducir a escala un back-end como el DFB4 (Digital Filter Bank 4, que se encuentra en el Telescopio Parkes) es una idea en desarrollo bastante interesante, pues podríamos crear adaptabilidad entre varios Front-end. Con este back-end se podría tomar datos en el radiotelescopio MEXART de la UNAM o en una antena de 5 metros, lo que proporcionaría además una herramienta más para aquellos interesados en la astronomía al ser de fácil acceso.

Códigos de Simulación de Folding

Se presentan dos códigos, el primero fue escrito en Matlab 2016 y el segundo en Python 2.7 para realizar una simulación de la técnica de Folding que se presenta en el capítulo 2.

A.1. Código fuente escrito en MATLAB 2016

```
%Creación de la señal
T=.1; %periodo del pulsar en segundos
w=(T*.05)/3; %ancho del pulso suele ser 5% del ciclo de trabajo
fs=1*103; %frecuencia de muestro en Hz
dura=100; %Duración de la señal en segundos o tiempo de observación
d=.1; %tiempo en el que se haran los cortes para hacer folding
SNRe=3; %Relación señal a ruido elegida o designada por el
usuario en dB
t=1:1/fs:dura;
n_p=dura/T; %numero de pulsos
x=pulstran(t,[1:n_p]*T,'rectpuls',w); %con esto generamos la primer
componente de la señal, lo que esta entre [] es el número de pulsos
de la señal, en este caso no la desfasamos.
```

A. CÓDIGOS DE SIMULACIÓN DE FOLDING

```
x1=.5*pulstran(t,([1:n_p]*T-w),'rectpuls',w); %lo mismo que el
código anterior pero con un adelanto de la señal principal con una
amplitud de .25
x2=.75*pulstran(t,([1:n_p]*T+w),'rectpuls',w); %ahora hacemos un
atraso con una amplitud de .75
suma=x+x1+x2; %es la suma de las 3 señales para generar un perfil
del pulsar
ruido=awgn(suma,SNRe); %generamos ruido gaussiano aditivo

%ahora debo dividir la señal en tramas
delta=round(d*fs); %Es el tiempo en el que se dividirá cada trama
matriz=vec2mat(ruido,delta); %es el arreglo de matriz que se hace
a partir del acomodo de los arreglos.

%obtenemos el perfil de los datos

[m,n]=size(matriz); %tamaño de la matriz
perfil=0;
for i=1:1:m
perfil=perfil+matriz(i,:);
end
promedio=perfil/m;

%Creamos gráficas
%Grafica de pulsar.

xlabel('tiempo en segundos')
```

```
ylabel('Amplitud')
subplot(3,1,1) %graficar
figura1 = plot(t,suma);
axis([1,2,0,1.5])
title('Perfil pulso sin ruido')
grid on

%grafica de pulso+ruido

subplot(3,1,2)
figura2 = plot (t,ruido);
grid on
title('Perfil pulso con ruido')
xlabel('Tiempo en segundos')
ylabel('Amplitud')
axis([1,10,-5,5])

%Grafica de perfil reconstruido

j=1:1:delta;
%subplot(3,1,3) %graficar
figura2 = plot (j,promedio);
title('Perfil reconstruido a partir del Folding')
ylabel('Amplitud')
axis ([0,delta,-.5,2])
xlabel('Tiempo en milisegundos [ms]')
grid on
```


Nota: El código escrito en MATLAB 2016 tiene funciones exclusivas de esta versión, por lo que quizá sea necesario hacer algunas modificaciones si se usa una versión distinta.

A.2. Código en Python

El código que se muestra a continuación fue escrito en Python 2.7.

Aunque el Apéndice presenta dos códigos, uno en MATLAB y otro en PYTHON, el usado en la tesis es este último.

```
from scipy import signal
import scipy as sc
import matplotlib.pyplot as plt
import numpy as np

#####
##### parametros de la senal
#####
T=.123 #periodo del pulso
DC=.05#ciclo de trabajo
w=DC*T #ancho del pulso
fs=1000
sf=1/T #es la frecuencia de la senal
ts=100 #tiempo de observacion
t = np.linspace(0, ts, fs*ts, endpoint=False) #ts*fs indica el numero
de pulsos que hay entre 0, ts
```

```
#####  
##### construimos la senal  
#####  
sig=signal.square(2 * np.pi * sf * t,duty=DC) # genera la senal cuadrada  
del pulso1  
sig1=signal.square(2*np.pi*sf*(t+w),duty=DC)#pulso2  
sig2=signal.square(2*np.pi*sf*(t-w),duty=DC)#pulso3  
ps=(sig+1)/2 #perfil de 0 a 1  
ps1=(sig1+1)/2 #perfil de 0 a 1 de la segunda senal  
ps2=(sig2+1)/2  
ns=ps+(.5*ps1)+(.75*ps2) #suma  
noise=np.random.normal(0,1,size=fs*ts) #generacion de ruido gaussiano  
suma=noise+ns  
  
#####  
##### Folding metodo  
#####  
delta=.125 #en tiempo puede ser ms o s  
columnas=round (delta*fs)  
renglones=int(round (len(ns)/columnas))  
matriz=np.resize(suma,(renglones,columnas))  
  
#####  
##### Sumando renglones  
#####  
reconstruido=np.sum(matriz,axis=0)/renglones
```

A. CÓDIGOS DE SIMULACIÓN DE FOLDING

```
#####  
##### graficamos senal  
#####  
##### Grafica pulso  
#####  
plt.subplot(1,2,1)  
plt.plot(t,ns,'g')  
plt.ylim(0, 1.1)  
plt.xlim(0,1)  
plt.grid()  
plt.xlabel('Tiempo en segundos')  
plt.ylabel('Voltaje')  
  
#####  
##### Grafica reconstruida  
#####  
tiempo= np.arange(0, len(reconstruido),1)  
plt.subplot(1,2,1)  
plt.plot(tiempo,reconstruido)  
plt.grid()  
plt.xlabel('Tiempo en milisegundos')  
plt.ylabel('Voltaje')  
plt.show()#Mostrar en pantalla  
  
#obtener la SNR del perfil reconstruido  
i=0  
contadorr=0
```

```
contadors=0
ruido=0
signalaverage=0
promedio=np.mean(reconstruido)
for i in range(0,len(reconstruido)):
    if (reconstruido[i]<=promedio):
        contadorr=contadorr+1 #saber cuantas veces se itero esta sentencia
        ruido=ruido+reconstruido[i]
    else:
        contadors=contadors+1 #saber cuantas veces se itero esta sentencia
        signalaverage=signalaverage+reconstruido[i]
#una vez obtenido estos valores podemos ahora obtener la SNR
promruido=ruido/contadorr
promsignal=signalaverage/contadors

SNR= abs(promsignal/promruido)
```


Instalación de PSRSOFT y PRESTO

PSRSOFT es un conjunto de programas para buscar púlsares, desarrollado por Michael J. Keith. PRESTO es un programa escrito por Scott Ransom para búsqueda de pulsares.

B.1. PSRSOFT instalación

PSRSOFT ¹ se puede instalar en sistemas operativos que estén basados en Debian o en RedHat, esto depende del gusto del usuario, mientras que en este apéndice mostramos la instalación en ambas distribuciones de Linux.

La instalación que se muestra a continuación fue realizada en Fedora y Centos 7 de 64 bits.

Hay que instalar las herramientas de desarrollo (“Development Tools”) con el siguiente comando: `sudo yum groupinstall “Development Tools”` (Para poder hacer la instalación se requieren permisos de usuario raíz)

Los programas o herramientas que contiene PSRSOFT se ejecutan sobre lenguaje C, python, Java, Fortran, entre otros por lo que debemos tener instalados los compiladores adecuados para la instalación.

```
yum install compat-gcc-44-gfortran.x86_64
```

(Librería necesaria para la instalación de PSRCHIVE)

```
yum install libxml2
```

¹Nota: PulsarHunter funciona sobre 64 bits, por lo que resultará incompatible instalar PSRSOFT sobre alguna que sea de 32 bits.

B. INSTALACIÓN DE PSRSOFT Y PRESTO

```
yum install libxml2-devel
```

Para instalar Java:

```
yum install java-1.8.0-openjdk-.x86_64
```

```
yum install java-1.8.0-openjdk-devel.x86_64
```

Para instalar PSRSOFT lo primero será descargar del siguiente link el programa:

```
http://www.pulsarastronomy.net/psrsoft/psrsoft.tar.gz
```

Una vez descargado el programa debemos entonces extraerlo:

```
tar -xzf psrsoft.tar.gz
```

```
cd psrsoft
```

```
ls psrsoft
```

```
/bin /config
```

```
cp ./config/profile.example ./config/profile
```

Si se instalan las librerías que se señalaron previamente, podemos ejecutar el siguiente comando e instalar las herramientas necesarias para PSRSOFT con el siguiente comando:

```
./bin/psrsoft sixproc dspsr psrchive pulsarhunter
```

y no deberá haber algún problema para la instalación de las herramientas.

Para la instalación en Debian es necesario primero instalar los siguientes paquetes:

```
apt-get install gcc
```

```
apt-get install gfortran
```

```
apt-get install libxml2-dev
```

```
apt-get install git
```

```
apt-get install build-essential
```

```
apt-get install libX11-dev
```

```
apt-get install libtool
```

```
apt-get install automake
apt-get install m4
apt-get install openjdk-7-dev
```

Para instalar PSRSOFT hay que descargar de la siguiente página el programa PSRSOFT <http://www.pulsarastronomy.net/psrsoft/psrsoft.tar.gz> Una vez descargado el programa debemos entonces extraerlo:

```
tar -xzf psrsoft.tar.gz
cd psrsoft
ls psrsoft
/bin /config

cp ./config/profile.example ./config/profile
```

Si se instalan las librerías que se señalaron previamente, podemos ejecutar el siguiente comando e instalar las herramientas necesarias para PSRSOFT con el siguiente comando:

```
./bin/psrsoft sixproc dspsr psrchive pulsarhunter
```

y no deberá haber algún problema para la instalación de las herramientas.

B.2. PRESTO INSTALACIÓN

Instalar PRESTO en RedHat de 64 bits representa varios problemas debido a que hay incompatibilidad en las librerías de PYTHON de 64 bits con las de 32 bits que son sobre las que funciona PRESTO, por eso no pondré la instalación de PRESTO en RedHat de 64 bits, pero existe la posibilidad de instalarlo en RedHat de 32 bits, en la siguiente página vienen instrucciones detalladas para su instalación:

<http://54.153.202.3/pulsarref/pulsar-software-install-centos.html>

Instalar PRESTO es mas sencillo usando Debian, debido a que existe compatibilidad entre las librerías de 32 bits y 64 bits, lo que nos reduce el trabajo. Necesitamos instalar las siguientes librerías antes de comenzar con PRESTO.

```
apt-get install csh
apt-get install tk
```


B. INSTALACIÓN DE PSRSOFT Y PRESTO

```
apt-get install tk-dev
apt-get install libpng-dev
apt-get install libgd2-xpm-dev
apt-get install cvs
apt-get install autoconf
apt-get install gsl-bin
apt-get install libgsl0-dev
apt-get install flex
apt-get install bison
apt-get install fort77
apt-get install libglib2.0-dev
apt-get install gnuplot
apt-get install gnuplot-X11
apt-get install python-dev
apt-get install python-numpy
apt-get install python-scipy
apt-get install python-matplotlib
apt-get install ipython
apt-get install python-notebook
apt-get install python-pandas
apt-get install python-sympy
apt-get install python-nose
apt-get install swig
apt-get install fftw3
apt-get install libfftw3
apt-get install libfftw3-dev
```

Se deben crear los siguientes PATH's de preferencia:

```
# Path to the pulsar software installation directory eg:
export SOFTWARE_DIR=/home/{user}/pulsar_software

# OSTYPE
export OSTYPE=linux

# Tempo
export TEMPO=$SOFTWARE_DIR/tempo

# PGPLOT
export PGPLOT_DIR=$SOFTWARE_DIR/pgplot_build
export PGPLOT_DEV=/xwindow

# PRESTO
export PRESTO=$SOFTWARE_DIR/presto

# LD_LIBRARY_PATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SOFTWARE_DIR/lib:
$SOFTWARE_DIR/pgplot_build:$SOFTWARE_DIR/presto/lib

# PATH
# Some Presto executables match sigproc executables so keep separate -
# all other executables are found in $SOFTWARE_DIR/bin
export PATH=$PATH:$SOFTWARE_DIR/bin:$SOFTWARE_DIR/presto/bin
```

B. INSTALACIÓN DE PSRSOFT Y PRESTO

```
# Python path
```

```
export PYTHONPATH=$PRESTO/lib/python:/usr/lib/python2.6/site-packages:  
$SOFTWARE_DIR/lib/python2.6/site-packages
```

```
#FFTWPATH
```

```
export FFTW_PATH=/usr/lib/x86_64-linux-gnu
```

Y se debe instalar TEMPO, PGPLOT, CFITSIO en una carpeta de direcciones a la que podamos acceder de manera sencilla.

Esto lo podemos hacer creando una carpeta como pulsar_software y descargar en este lugar los archivos.

CFITSIO.

```
$ cd cfitsio-{version}
```

```
$ ./configure --prefix=$SOFTWARE_DIR CFLAGS=-fPIC FFLAGS=-fPIC
```

```
$ make shared
```

```
$ make install
```

```
$ make clean
```

PGPLOT

Se debe crear un directorio vacío (pgplot_build) y copiaremos el archivo drivers.list de pgplot a pgplot_build.

Acrónimos

MIE	Medio Interestelar
DM	Medida de Dispersión
LNA	Low Noise Amplifier
GTM	Gran Telescopio Milimétrico
MEXART	Mexican Array Telescope
LOV	Landau-Oppenheimer-Volkoff
PuMa	Pulsar Machine
EPN	European Pulsar Network
PSRFITS	Pulsar Flexible Image Transport System
TEMPO	Pulsar timing data analysis
DSPSR	Digital Signal Processing Software for Pulsar Astronomy
PSRCHIVE	Pulsar Data Archival
PSRSOFT	Pulsar Software
PRESTO	Pulsar Exploration and Search Toolkit
NANOgrav	North American Nanohertz Observatory for Gravitational Waves
RF	Radiofrecuencia
FI	Frecuencia Intermedia
BCD	Binary Coded Decimal
TDF	Transformada Discreta de Fourier

FFT	Transformada Rápida de Fourier
SNR	Signal to Noise Ratio
FITS	Flexible Image Transport System
HDU	Header Data Unit
GBT	Green Bank Telescope
UHF	Ultra High Frequency
DFB	Digital Filterbank
USRP	Universal Software Radio Peripheral
FPGA	Field-programmable gate array
PSRDADA	Pulsar Distributed Acquisition and Data Analysis

Bibliografía

- [1] A. Richard Thompson, James M. Moran, G. W. S. J. (1986). *Interferometry and Synthesis in Radio Astronomy*. Wiley-Vch, New York, primera edición.
- [2] Arthur, S. J. (2012). Capítulo 4. etapas finales de la evolución estelar. Instituto de Radioastronomía y Astrofísica, UNAM.
- [3] Autores, V. (2014). Psrfits documentation v5.4. [urlhttp://www.atnf.csiro.au/people/pulsar/index.html?n=PsrfitsDocumentation.Txt](http://www.atnf.csiro.au/people/pulsar/index.html?n=PsrfitsDocumentation.Txt).
- [4] B. R. Barsdell, M. Bailes, D. G. B. C. J. F. (2012). Accelerating incoherent dedispersion. *Monthly Notices of the Royal Astronomical Society*, 422:379–392.
- [5] Backer, D. C. (1998). Recent results with berkeley pulsar procesors. University of California.
- [6] Barr, E. D. (2013). *Searching for Pulsar with the Effelsberg Telescope*. PhD thesis, RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITAT BONN, Faculty of Mathematics and Natural Sciences.
- [7] Burns, W. R. and Clark, B. G. (1969). Pulsar search techniques. *Astronomy and Astrophysics*, 2:280–287.
- [8] D. Ait Allal, R. W. and Cognard, I. (2009). Puma-ii: A wide band pulsar machine for the westerbork synthesis radio telescope. *Signal Processing Conference, 2009 17th European*, s.i:2052–2056.
- [9] Deane, R. (2006). *Developing a Pulsar Search Algorithm*. PhD thesis, University of Cape Town, Department of Electrical Engineering.
- [10] E. J. Korpela, E. M. H. and Werthimer, D. (2000). Pulse detection algorithms for use in seti@home. *American Astronomical Society*, 32:1492–1500.

BIBLIOGRAFÍA

- [11] Fisher, J. R. (2002). Back-ends. *Single-Dish Radio Astronomy: Techniques and Applications*, 278:113–122.
- [12] G. Hobbs, e. a. (2011). The Parkes Observatory pulsar Data Archive. *Publications of the Astronomical Society of Australia*, 28:1–14.
- [13] Garrindo, L. L. (2008). *Introducción a la Física del Cosmos*. Universidad de Granada, España, primera edition.
- [14] Griffiths, D. J. (1999). *Introduction to Electrodynamics*. Prentice Hall, New Jersey, third edition. 4
- [15] Gupta, Y. (s.f.). Pulsar observations. National Centre for Radio Astrophysics. Tata Institute of Fundamental Research, Pune. 26
- [16] Hankins, H. T. and Rickett, B. J. (1975). Pulsar signal processing. *Academic Press*, 14:55–129.
- [17] Hewish, A. (1979). Pulsars. *Royal Institution of Great Britain*, 49:201–207.
- [18] J. M. Cordes, M. Kramer, T. J. W. L. B. W. S. D. C. B. S. J. (2004). Pulsars as tools for fundamental physics and astrophysics. *New Astronomy Reviews*, 48:1413–1438.
- [19] J. M. Ford, R. M Prestage, M. B. (2014). Experiences with the design and construction of wideband spectral line and pulsar instrumentation with casper hardware and software: The digital backend system. *Proceedings of the SPIE*, 9152:1.
- [20] .Keith, M. J. (2012). Installation and use of Pulsar Search Software. *Astronomical Research and Technology*, 9:1–10.
- [21] Khoo J., Hobbs G., M. R. N. M. D. D. J. (2012). Using the parkes pulsar data archive. *Arxiv*, s.i:1–9. 51
- [22] Kovelakas, K. K. (2013). *Search for weak periodicities in astronomical data using the FFT and FFA algorithms*. PhD thesis, Aristotle University of Thessaloniki Faculty of Sciences – Department of Physics Section of Astrophysics, Astronomy and Mechanics.
- [23] Li-Yong Liu, E. A. and Zhang, J. (2006). Pulsar coherent de-dispersion experiment at urumqi observatory. *Chinese Journal of Astronomy and Astrophysics*, 6:53–55.

- [24] Lorimer, D. and Kramer, M. (2005). *Handbook of Pulsar Astronomy*. Cambridge University Press, New York, segunda edición. [XI](#), [XI](#), [XI](#), [XI](#), [XI](#), [XI](#), [16](#), [17](#), [18](#)
- [25] Lynch, R. (s.f.a). From the Telescope to the Collaboratory. Department of Astronomy, University of Virginia.
- [26] Lynch, R. S. (s.f.b). Searching for and Identifying Pulsars. Department of Physics, McGill University.
- [27] Lyne, A. and Graham-Smith, F. (1998). *Pulsar Astronomy*. Cambridge University Press, United Kingdom, segunda edición.
- [28] Lyons, R. G. (2008). *Understanding Digital Signal Processing*. Prentice Hall, New Jersey, segunda edición.
- [29] M. Kramer, A. G. Lyne, B. C. J. T. I. C. J. A. J. H. (2001). Cobra – a digital receiver at jodrell bank. University of Manchester.
- [30] Marton, L. (1976). *Methods in Experimental Physics*. Academic Press Inc, New York, primera edición.
- [31] Moya, J. P. A. (2011). Procesamiento digital de señales. Tecnológico de Costa Rica. Escuela de Ingeniería.
- [32] Ng, C. W. Y. (2014). *Pulsar searching and timing with the Parkes telescope*. PhD thesis, RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN, Faculty of Mathematics and Natural Sciences.
- [33] P.C. van Haren, J. L. L. Voûte, T. D. B. D. D. M. L. A. K. and Langerak, J. J. (2000). Puma, the first fully digital pulsar machine. *IEEE transactions on Nuclear Science*, 47:91–98.
- [34] R. Karuppusamy, B. Stappers, W. v. S. (2008). Puma-ii: A wide band pulsar machine for the westerbork synthesis radio telescope. *Publications of the Astronomical Society of Pacific*, 120:191–202.
- [35] Ransom, S. (s.f). Searching for Pulsars with PRESTO. NRAO.
- [36] S. M. Ord, H. S. and Camilo, F. (2000). Coherent de-dispersion observations at jodrell bank. *Pulsar Astronomy*, 202:1–2.
- [37] Salas, P. and Solís, M. A. (2006). Estrella de neutrones. *Revista Mexicana de Física*, 52:37–46. [XI](#), [XI](#), [XI](#), [XI](#), [3](#)

BIBLIOGRAFÍA

- [38] Staelin, D. H. (1969). Fast folding algorithm for detection of periodic pulse trains. *Proceedings of the IEEE*, 57:724–725. [26](#)
- [39] Taylor, J. H. (1973). Pulsar receivers and data processing. *Proc. IEEE*, 61:1295–1298.
- [40] Thomas L. Wilson, Kristen Rohlfs, S. H. (2013). *Tools of Radioastronomy*. Springer, Washington D.C, sexta edition.
- [41] V. J. Martínez, J. A. Miralles, E. M. D. G.-E. (2005). *Astronomía Fundamental*. Universitat de València, España, primera edition.
- [42] W. Van-Straten, M. B. (2011). dspsr: Digital signal processing software for pulsar astronomy. *Publications of the Astronomical Society of Australia*, 28:1–14.