USING 2D-FFT TO SEARCH FAST RADIO BURST

C.H. NIU
Central China Normal University
Luoyu Road,Wuhan, China
National Astronomy Observatory , Chinese Academy of Sciences
Datun(A) Road,No.30,Beijing, China and
University of California Berkeley, Campbell Hall 339, Berkeley CA 94720
(Revised April 3, 2018)
submitted to APJ

ABSTRACT

Fast Radio Burst have been found from pulsar data for many years. There are several FRB search algorithm like tree algorithm, FDMT et. Here we proposed a different FRB searching algorithm which basically trace a curve in frequency-time image. This algorithm is mainly realized by two dimensional Fast Fourier Transform. We take a 2D FFT on the $f^{-2}(t)$ data map, Then trace the signal along the angle of straight line. In this searching method, it's easier to remove RFI in large scale and will bring a speed up benefit in well-developed 2D FFT library both in CPU and GPU code.

Keywords: 2DFFT ,algorithm, Fast radio burst

1. INTRODUCTION

Fast radio bursts (FRBs) are bright ($\sim$Jy) and short ($\sim$ms) bursts of radio emission that have dispersion measures (DMs) in excess of the line of sight DM contribution expected from the electron distribution of our Galaxy. To date 18 FRBs have been reported — most of them detected at the Parkes telescope (**?????????**) and one each at the Arecibo (**?**) and Green Bank telescopes (**?**).

A plethora of source models have been proposed to explain the properties of FRBs (see e.g. **?**, for a brief review). According to the models, the excess DM for FRBs may be intrinsic to the source, placing it within the Galaxy; it may arise mostly from the intergalactic medium, placing a source of FRBs at cosmological distances ($z \sim 0.2 - 1$) or it may arise from the host galaxy, placing a source of FRBs at extragalactic, but not necessarily cosmological, distances ($\sim 100\,\mathrm{Mpc}$).

Since the only evidence to claim an extragalactic origin for FRBs has been the anomalously high DM, some models also attempted to explain the excess DM as a part of the model, thus allowing FRBs to be Galactic. All FRBs observed to date have been detected with single dish radio telescopes, for which the localization is of order arcminutes, insufficient to obtain an unambiguous association with any object. To date, no independent information about their redshift, environment, and source could be obtained due to the lack of an accurate localization of FRBs. **?** attempted to identify the host of FRB 150418 on the basis of a fading radio source in the field that was localized to a $z = 0.492$ galaxy. However, later work identified the radio source as a variable active galactic nucleus (AGN) that may not be related to the source (**????**).

Repeated radio bursts were observed from the location of the Arecibo-detected , with the same DM as the first detection, indicating a common source. As discussed by, it is unclear whether the repetition makes unique among known FRBs, or whether radio telescopes other than Arecibo lack the sensitivity to readily detect repeat bursts from other known FRBs.

? used the Karl G. Jansky Very Large Array (VLA) to directly localize the repeated bursts from with 100-mas precision and reported an unresolved, persistent radio source and an extended optical counterpart at the location with a chance coincidence probability of $\approx 3 \times 10^{-4}$ — the first unambiguous identification of multi-wavelength counterparts to FRBs. Independently, used the European VLBI Network (EVN) to localize the bursts and the persistent source and showed that both are co-located within ~ 12 milliarcseconds.

Here we report a new algorithm to search FRBs .

2. BASICS OF INCOHERENT DEDISPERSION

The dispersion of the electromagnaetic wave pulse cause a delay in arrival time at frequency ν compared with the reference frequency ν_0, which is given by :

$$\Delta t(\nu) = -D(\nu^{-2} - \nu_0^{-2}) \tag{1}$$

where D is the dispersion measure times a dispersion constant. Thus , We may model a burst with a very short intrinsic width as :

$$I(t,\nu) = I_0(\nu)\delta_D(t - t_s - \frac{D}{\nu^2}) \tag{2}$$

Where δD is the Dirac delta function, t_s marks the signal starting time for infinitely high frequency. If the bandwidth is small, we can approximate

$$\frac{D}{\nu^2} \approx \frac{D}{\nu_0^2}(1 - 2\frac{\nu - \nu_0}{\nu_0})$$

denote $\Delta\nu \equiv \nu - \nu_0$, and assume that the spectrum is not too steep such that within the observing band the

signal is constant, then

$$I(t,\nu) \approx I_0 \delta_D(t - t_s - \frac{D}{\nu_0^2}(1 - 2\frac{\Delta\nu}{\nu_0}))$$
$$= I_0 \delta_D(t - t_0 + \frac{2D}{\nu_0^3}\Delta\nu) \tag{3}$$

where t_0 is the arrival time of the signal at thre reference frequency ν_0.

Now consider an integral of this signal between frequency ν_1 and ν_2 , the signal strength would be

$$s = \int d\nu \int dt I(t,\nu) = (\nu_2 - \nu_1)I_0 = I_0 B \tag{4}$$

Where $B = \nu_2 - \nu_1$ is the bandwidth. Now consider the noise. Suppose the data is digitized with time interval δt and frequency channel bandwidth $\delta\nu$. For the incoherent dedispersion, the signal within each time interval and frequency channel is

$$I_n = \frac{2kT_{sys}}{A_{\text{eff}}\sqrt{\delta\nu\delta t}} \tag{5}$$

Suppose we are observing between ν_1,ν_2 with a total of N_ν channels, and processing a time interval $T = N_t \delta t$ where $T \geq \Delta t(\nu_1) - \Delta t(\nu_2)$, i.e. the whole of the dispersed signal is within the data frame.

For incoherent dedispersion , in the absence of the pulse signal, the whole read out of the data frame is given by

$$n = \int d\nu \int dt I_n = \frac{2kT_{sys}}{A_{\text{eff}}}\frac{(\nu_2 - \nu_1)T}{\sqrt{\delta\nu\delta t}}$$
$$= \frac{2kT_{sys}}{A_{\text{eff}}}B^{1/2}T^{1/2}N_\nu^{1/2}N_t^{1/2} \tag{6}$$

So the raw signal to noise ratio is given

$$\text{SNR}_{raw} = \frac{I_0 A_{\text{eff}}}{2kT_{sys}}\left(\frac{B}{N_\nu N_t T}\right)^{1/2} \tag{7}$$

In a perfect incoherent dedispersion , we sum up all the signal, which is still given by s. However, we compare it with the noise in the same dedispersion $\nu - t$ track, not the whole data frame. The noise along the same track is given by

$$n = \int d\nu \int dt I_n \delta_D(t - t_0 + \frac{2D}{\nu_0^3}\Delta\nu) = BI_n \tag{8}$$

Then

$$\text{SNR}_{opt} = \frac{I_0}{I_N} = \frac{I_0 A_{\text{eff}}}{2kT_{sys}}\left(\frac{BT}{N_\nu N_t}\right)^{1/2} \tag{9}$$

Now consider a pulse of finite width. We replace the Dirac δ function by a Gaussian function with the same normalization

$$\delta_D(t - t') \rightarrow g(t - t') \equiv \frac{1}{(2\pi)^{1/2}\sigma}\exp[-\frac{(t - t')^2}{2\sigma_t^2}] \tag{10}$$

If the pulse intrinsic width $\sigma > \delta t$, then in a dedispersion along the track only the part of the signal within one time bin would be included, which gives

$$\int_{-\delta t}^{+\delta t} d\Delta t \frac{1}{\sqrt{2\pi}\sigma}e^{-\Delta t^2/2\sigma^2} = erf(\frac{\delta t}{\sqrt{2}\sigma}) \approx \sqrt{\frac{2}{\pi}}\frac{\delta t}{\sigma} \tag{11}$$

Where the last holds for the case $\delta t \ll \sigma$, so in this case

$$s = I_0 B\sqrt{\frac{2}{\pi}}\frac{\delta t}{\sigma} \tag{12}$$

While the noise is still given by Eq.(8), so in this case

$$\text{SNR}_{fin} = \frac{I_0}{I_n} = \frac{I_0 A_{\text{eff}}}{2kT_{sys}}\left(\frac{BT}{N_t N_\nu}\right)^{1/2}\sqrt{\frac{2}{\pi}}\frac{\delta t}{\sigma} \tag{13}$$

3. IMPLEMENTED 2DFFT ON BURST SEARCH

Assuming data output from Transient Incoherent Search backend are in (f,t) format with size $[N_{ch}, N_{tsamp}]$. We could get the signal line are showing up as a curve line which are complied with relationship of $f^{-2} \sim t$ in raw data map from eq. (1). As the form of curve after 2D-FFT is much more complex than a straight line. Hence we take the 2D-FFT on (f^{-2}, t) map. In order to get (f^{-2}, t) map, we use interpolation along the frequency axis in (f,t) map. The total pixel number of (f^{-2}, t) map keep same, so that total information could get conservation. In this case, re-bin data still has shape of $[N_{ch}, N_{tsamp}]$,

From eq.(1) , define $k_1 = -\frac{1}{D}$. We could rewrite eq.(1) as

$$f^{-2} = k_1 t + b \tag{14}$$

where b is decided by start time. Apply Fourier pair: $(f^{-2}, t) \sim (v, u)$, Then process 2D FFT on eq.(14) :

$$\int\int \delta(k_1 t + b - f^{-2})e^{-i2\pi(ut + vf^{-2})}dtdf^{-2}$$
$$= \delta(u + k_1 v) \cdot e^{-i2\pi vb} \tag{15}$$

In (v, u) map ,Signal line format has been changed into $v = -k_1^{-1}u$ with slope $k_2 = -\frac{1}{k_1} = -D$, where $D = DM \cdot 4.15 \times 10^{-6}ms \cdot MHz^2 \cdot pc^{-1} \cdot cm^3$. Considered unit of f^{-2} and t axis after FFT will become $[\max(f^{-2}) - \min(f^{-2})]^{-1}$ and T^{-1}. Actually DM we got from slope k_2 is:

$$DM = -\frac{1}{C} \cdot \frac{T}{\max(f^{-2}) - \min(f^{-2})} \cdot k_2 \tag{16}$$

And the b comes to module factor in $e^{-i2\pi vb}$. It is still straight but perpendicular to previous straight. Furthermore, wherever the signal arise in raw data map , it always go cross the center of 2D-FFT map. Take advantage of these property, we could search transient signal on 2D-FFT map along specific angle.

For the phase term $e^{-i2\pi vb}$, it will modulate the signal intensity monotonically along straight line of 2D-FFT map. if we sum along straight line directly, we might not get the highest signal noise ratio. An effective and feasible method is to take a second FFT along straight line, then whole signal will turn into a spot. The whole process will take 4 steps to accomplish, we called them : Re-bin, 2D-FFT, Polar transform, 1D-FFT. Data after these process called map with process suffix separately,e.g. data after Re-bin Re-bin map.
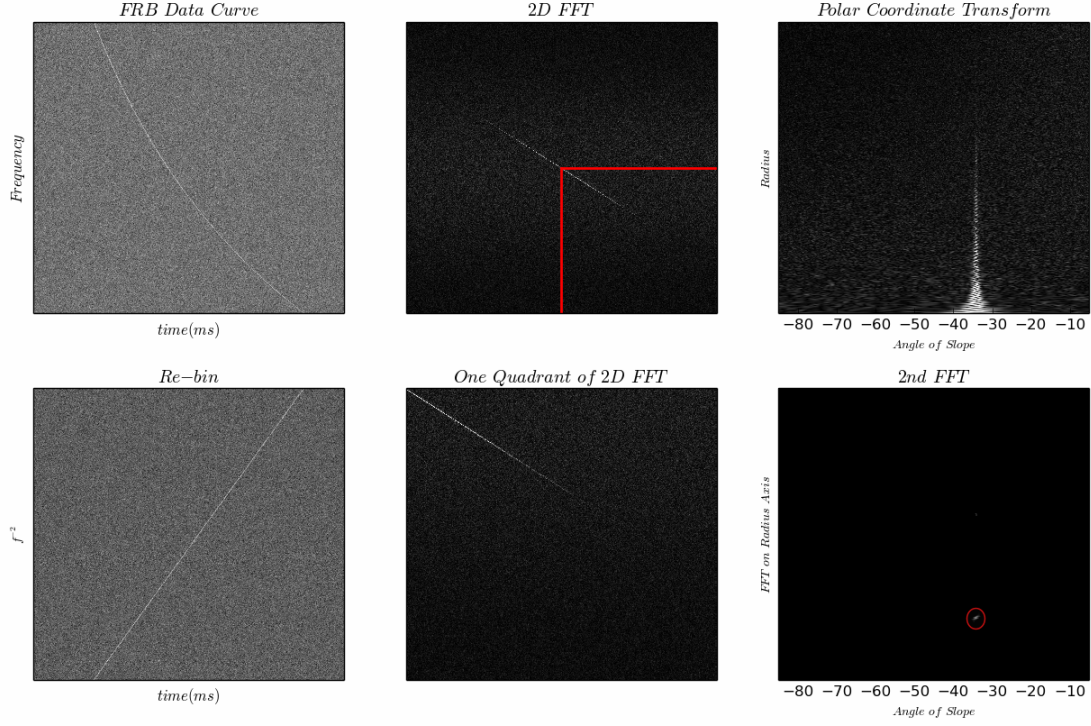
Figure 1. The whole process of 2DFFT algorithm. Left top is Raw data of FRB data curve in $f(t)$ form which are follow Eq(1). Left bottom is Re-bin process on Raw data,after this process curve in the Left top will comes to a straight line in $f^{-2}(t)$ form. Mid top is Re-bin Data after 2D FFT. Straight line in Left bottom turn into a straigt line that rotate $90°$ and go through center. As discussed in Sec 3.1, we could only take one quadrant data. Here we take data inside red line area of Mid top. One quadrant data are showed in Mid bottom. In Mid bottom image, Left top point is taken as origin and top horizontal line as origin axis, Then calculate the radius and angle of slope of each pixel and change this image into polar coordinate image of $(radius, angle)$ form as showed in Right top. As phase term influence, interference texture could be found on 2D FFT image. Then final step is doing 1D FFT along radius axis on polar coordinate data, then entire signal line will converge to several spots as showed in Right bottom.

3.1. polar coordinate transform

As DM always bigger than zero, so k_2 is always positive. It indicate signal after 2D-FFT is a straight line which will only appear in $1/2$ of 2DFFT map. For example, Fig.2 shows a signal line location after 2DFFT, it won't show up in quadrant II and IV, but only quadrant I and III. nevertheless quadrant I and III are conjugate to each other, we can only take 1 of them.

For purpose of search transient on 2D-FFT map, we need trace every slope of line which go cross center. It's easier to get each straight which pass through center from raius and angle of slope space (r, θ). Thus it's better to take a coordinate transform. In Fig. 2, If we make use of quadrant I, we can take map center as origin O, and horizontal axis as original axis, then each line go cross center will have radius and angle information.

As tiny scale before FFT will decide large structure after FFT. Taking care of direction of FFT, Signal length in 2D-FFT map depend on width of signal line on re-bin map. Similarly, For polarization coordinates transform, the angle resolution could get from the length of signal line on re-bin map. Note width and length of signal before 2D-FFT is w_s and l_s, the length and width of signal after 2D-FFT will be $l_{FFT} = 1/w_s$ and $w_{FFT} = 1/l_s$ separately. Dimension of pixel need to be considered when actually process. Thus It's not necessary to take the whole data of quadrant I but pixels whose radius value
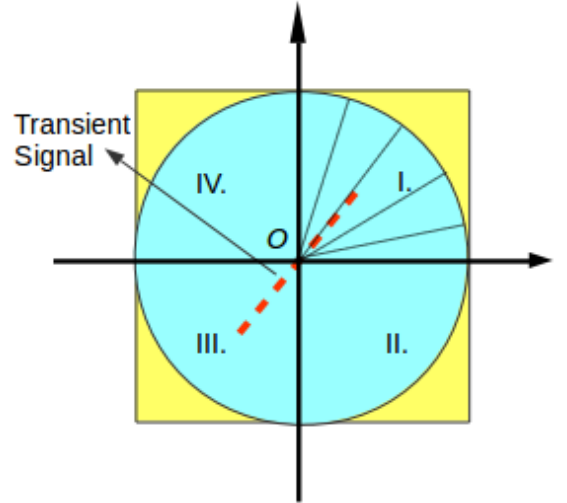

Figure 2. Signal on 2DFFT map

smaller tan l_{FFT}. Furthermore, if DM range are already set up, the boundary angle of slope $[\theta_{DM_{min}}, \theta_{DM_{max}}]$ are also been decided according to Eq 16. Thus pixel whose angle of slope are beyond boundary could be cast through.

There are a lot of ways to take polar coordinate transform. But here we are taking interpolation which might

be more smooth.

After polar coordinate transform, signal line will exist at some exact angle θ. As shows in right top of Fig 1.

3.2. *Turn signal line into spot*

As we discussed before, the phase term $e^{-i2\pi vb}$ could disturb the signal. Since phase term is monotonically and data are complex number, if we integrate along radius directly, signal intensity will be blended. We also can't take absolute value then sum, due to noise will be amplified. A reasonable method is to take another FFT along radius axis, so that the influence of phase term will be eliminated. After this we could get Signal Noise Ratio by

$$SNR = \frac{Max\ Value}{std}$$

When we know where the highest SNR is , we can got the slope of it , finally calculate the DM Value from Eq.(16). The entire procedure are showed by Fig1 The Data are generated by SIGPROC and parameter are simulated from TianLai Array FRB backend request which Time resolution is 1 millisecond, Band width is 100 MHz and central Frequency is 750 Mhz, with 1024 Frequency Channels.

3.3. *Advantage on RFI removing*

There are two sorts of regular RFI: At certain frequency last long time, At certain time stamp but with large bandwidth, corresponding to horizontal and vertical line separately in image of Re-bin data. After 2D FFT , they will rotate 90°, but all of them go through center no matter what frequency or time stamp they are. Therefore, Regular type of RFI could be removed simply by discarding data after 2D FFT at horizontal and vertical line that go cross center.An example could be seen from Fig. 3.

4. DETECTION OF OBSERVED FRB DATA

Observed FRB data are tested through this algorithm.

5. INCOHERENT DE-DISPERSION ALGORITHMS AND COMPUTE COMPLEXITY

There are a lot of incoherent transient search algorithm already in use, But I'd like to difference them into four types: 1) *Bruteforce* 2) *Treealgorith* , 3) *Imageprocessing*.

BruteForce is implemented in a lot of software like *SIGPROC*, *HEIMDALL*. In general, the compute complexity of transient search algorithm is compute dependency on number of time samples $'N'_t$, number of frequency channels $'N'_\nu$ and number of DM $'N'_{DM}$ of observation. The basics of *BruteForce* is to shift time delays back for each frequency channel according to Eq1. As every DM are processed with every time sample and frequency channel, so the total compute complexity is $(N_t N_\nu N_{DM})$. It has the slowest compute complexity along the 3 types, However it is easier to paralleliz and implement on GPUs which are good at high performance computing. *HEIMDALL* is an example to take use GPU to De-disperse. It has been used for a lot of real-time survey like UTMOST ?? and ??? and already detected lots of FRB instances.
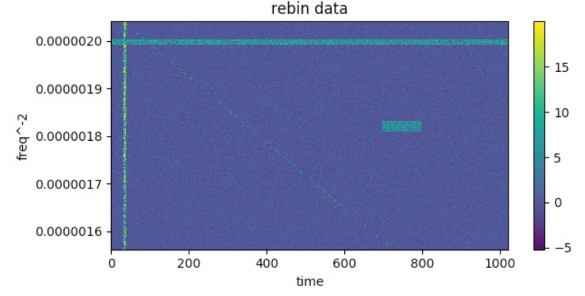
(a) Add RFI manually on Re-bin map
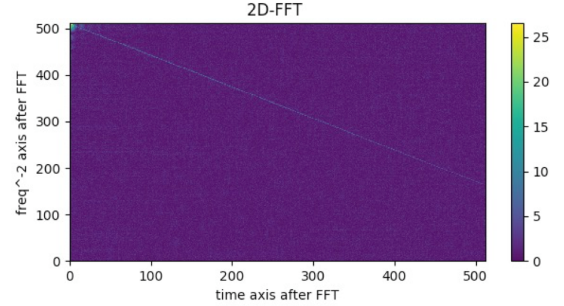
(b) RFI are removed on 2D FFT map

Figure 3. Plot (a) shows Re-bin map which are add 3 RFI manully: Time domain and frequency domain and short bad data area. Seeing from Plot (b), RFI are removed clearly.

Different with *BruteForce* algorithm, Talor proposed *Tree* algorithm ?? to accelarate *BruteForce* de-disperse algorithm. Under asumption of Eq 2, Tree algorithm trace signal across a straight line and reuse some element to get rid of redundant calculation. It has same mechanisim as FFT and compute complexity is $(N_t N_\nu \log_2 N_\nu)$. Though Tree algorithm is much faster than *BruteForce*, it is difficult at pratical application. The result are iterative at each step, This algorithm has worse parallelism. As for Linear aproximate, it also face SNR lost. Advanced 'Tree' method are raised up. People are using sub frequency band to get curve more like straigt and also can parallel at that time. Moreover, 'FDMT' are proposed to get accurate curve calculated in decent subband. In 'FDMT' algorithm, the compute complexity is $\max\{N_t N_{DM} \log_2(N_\nu), 2N_\nu N_t\}$.

Algorithm disscuessed in this paper is belong to *Imageprocessing* way to De-disperse. This type of algorithm foucus on finding a signal line on given (f,t) data. Excpet algorithm talked in this paper, there are a lot of ways to complish this algorithm, like Radnom transform and Machine learning. The compute complexity is varied with different implement method. For 2D-FFT algorithm talked in this papaer, The most time consuming step is 2D-FFT. Though a second 1-D FFT are executed on (r,θ) data, this doesn't change due to N_t, N_ν, or N_{DM}. Because data shape after polar coordinate transform is only decided by supposed signal width w_s and assumed signal length l_s which are constant during a observation. The scale need to process are much smaller than 2D-FFT . Thus the compute complexity got from this algorithm are mainly caused by 2D-FFT which are $2N_t N_\nu \log_2(N_t N_\nu)$.

Table 1 shows comparation with some Transients search algorithm like *BruteForce*, *Tree*, *FDMT*. This

Table 1
Algorithm comparison

2DFFT	FDMT	Brute force	Tree
Computational complexity $2N_t N_\nu \log_2(N_t N_\nu)$	$\max\{N_t N_\Delta \log_2(N_f), 2N_f N_t\}$	$N_f N_t N_\Delta$	$N_f N_t \log_2(N_f)$
Parallelization friendly Yes	Yes	Yes	No

Note. — Comparison of the FDMT algorithm with two other approaches to incoherent dedispersion, brute force (e.g.,It is clear that the FDMT algorithm dominates in all parameters.

algorithm doesn't have simplest compute complexity. Take Tree algorithm for example, this method will twice the complexity. However there are advantages that we might be interested.
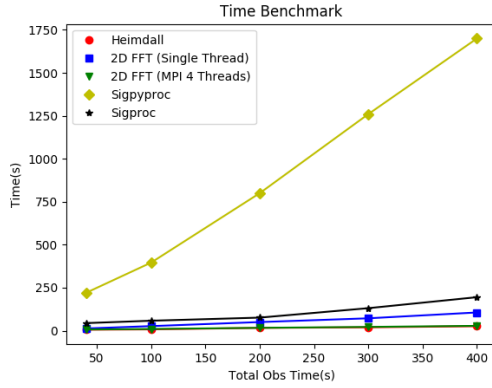
5.1. Benchmark

Figure 4. Benchmark of Transient search software: Sigproc, Sigpyproc, Heimdall , 2DFFT

REFERENCES

Barsdell, B. R., Bailes, M., Barnes, D. G., & Fluke, C. J. 2012, MNRAS, 422, 379

Brady, Martin L. "A fast discrete approximation algorithm for the Radon transform." SIAM Journal on Computing 27, no. 1 (1998): 107-119.

Clarke, N., Macquart, J.-P., & Trott, C. 2013, ApJS, 205, 4

Clarke, N., D'Addario, L., Navarro, R., & Trinh, J. 2014, Journal of Astronomical Instrumentation , 3, 50004

Gotz, W. A., and H. J. Druckmuller. "A fast digital Radon transform. An efficient means for evaluating the Hough transform." Pattern Recognition 29, no. 4 (1996): 711-718.

Manchester, R. N., Lyne, A. G., D'Amico, N., et al. 1996, MNRAS, 279, 1235

Magro, A., Karastergiou, A., Salvini, S., et al. 2011, MNRAS, 417, 2642

McLaughlin, M. A., & Cordes, J. M. 2003, ApJ, 596, 982

Lorimer, D. R., Bailes, M., McLaughlin, M. A., Narkevic, D. J., & Crawford, F. 2007, Science, 318, 777

Lorimer, D. R., & Kramer, M. 2012, Handbook of Pulsar Astronomy, by D. R. Lorimer , M. Kramer, Cambridge, UK: Cambridge University Press, 2012,

Petroff, E., van Straten, W., Johnston, S., et al. 2014, ApJ, 789, L26

Taylor, J. H. 1974, A&AS, 15, 367

Thompson, D. R., Wagstaff, K. L., Brisken, W. F., et al. 2011, ApJ, 735, 98

Thornton, D., Stappers, B., Bailes, M., et al. 2013, Science, 341, 53

van Leeuwen, J., & Stappers, B. W. 2010, A&A, 509, A7