Amateur Pulsar Detection Using the RTL SDR

PW East (UK), GM Gancio (Argentina[(1)])

Introduction

This project sought to determine the minimum useful antenna aperture for amateur radio astronomers to successfully detect pulsars around the Hydrogen line frequency of 1420MHz. The technique relied on the collaboration with GM Gancio, who provided RTL SDR data of the Vela pulsar (B0833-45, J0835-4510) and others, collected with a 30m radio telescope. This data was processed to determine the achievable signal-to-noise ratio from which, the minimum useful dish size necessary for some effective amateur work, could be calculated. Two software packages were developed to do synchronous integration, a third to provide a power detection function and a fourth for spectrum analysis to recover pulsar rotation rate.

Pulsar Detection

Pulsar signals are very weak bursts of noise over a very wide frequency range at a regular rate. The duty cycle is typically 5-10%.
The detection process uses recorded complex IQ voltage sampled data collected at a chosen clock frequency f_c over an RF bandwidth equal to $B = f_c$.

The receiver input terminal noise voltage is proportional to, $\sqrt{k(T_p + T_{sys})B}$ within the pulsar pulse and $\sqrt{kT_{sys}B}$ outside. T_p and T_{sys} are the effective pulsar and receiver system equivalent noise temperatures.
Squaring the I and Q components (square-law detection)*, the result is (Figure 1) both DC components, $kT_pB + kT_{sys}B$, $kT_{sys}B$ in and out of the pulse respectively plus AC noise components, of similar magnitude, $kT_pB + kT_{sys}B$, $kT_{sys}B$.

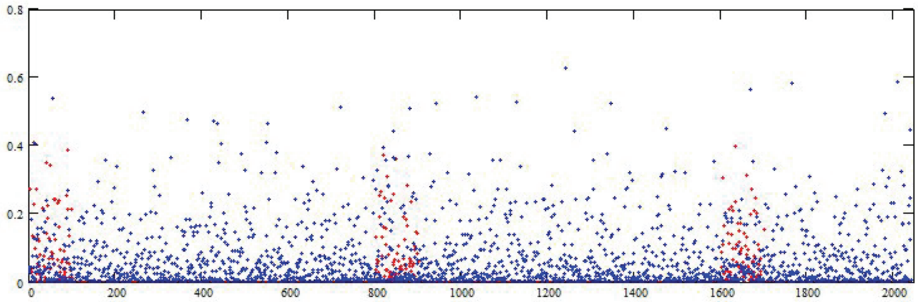

Figure 1 Square-law Detection v Time; System noise(blue) + Pulsar noise(red)

The situation is depicted in Figure 1 showing unipolar system and pulsar noise from which the DC and AC components can be anticipated. Considering the DC terms, these appear as a weak pulsar pulse train sitting on the strong DC platform set by the detected system noise. To recover the DC pulse waveform from the high level AC noise, the technique is to synchronously add the pulses within the pulsar period. Synchronously summing N pulsar periods increases the DC voltage part by N and the AC term by $\sqrt{N}$.
The result is an improvement in pulsar signal-to-noise (voltage) ratio by $\sqrt{N}$ or,

* Linear detection produces the same result if $T_p << T_{sys}$

$$SNR_v = \frac{\sqrt{N}T_p}{T_{sys}}$$

The programs *rapulsar* and *rapulsan* carry out this synchronous integration, requiring the pulsar rotation period measure to be chosen accurately.
The integration process sums the pulsar sampled data within its period divided in a number '*n*' of bins. The choice of bin number '*n*' affects the factor '*N*" controlling the SNR improvement.
If data is collected over a time 'τ', then $Nn = \tau f_c = \tau B = F$; equal to the data file number of samples, *F* as set in *rtl_sdr* command line,
or, $N = \frac{F}{n} = \frac{\tau B}{n}$
So the voltage SNR becomes,

$$SNR_v = \sqrt{\frac{B\tau}{n}} \frac{T_p}{T_{sys}} \qquad (1)$$

Showing sensitivity improves as the square root of the RF bandwidth and data collection time, degrades with the square root of the bin number, but responds directly to a reduction in the receiver system noise temperature.

Pulsar signal power is usually expressed in Janskys (J); the received source power corresponding to $1J = 10^{-26}$Watts/m^2/Hz.
The equivalent received power in terms of Boltzmann's constant and effective temperature is kT_JB Watts, where $k = 1.38.\ 10^{-23}$Watts/°K/Hz; T_J is measured in °K and again, *B* is the RF bandwidth in Hz.
Equating the powers in these expressions,

$J\,A\,B\ .10^{-26} = 1.38\ .10^{-23} T_J\,B$ Watts, or, $J\,A = 1380T_J$

where A, is the receiver antenna effective collecting area (m^2).
Rearranging the equality,

$$T_J = J.A/1380\ °K$$

As an example, for a 30m dish with 60% aperture efficiency, the equivalent source temperature for a 1Jansky pulsar source is, 0.31/2 = 0.154°K (1/2 factor for receiving a single polarisation).
The larger data collections in this project involved files of 1G samples and an RF bandwidth of 2MHz.
The system noise for the data collection telescope is stated as 110°K.
The temperature measurement uncertainty in each of $n = 100$ time bins within the pulsar period is therefore, $dT = 110°K / \sqrt{(10^9/100)} = 0.035°K$.
The signal-to-noise ratio per Jansky for the 30m dish is now predicted as,
SNR = 0.154/0.035 = 4.4/Jy.
Inspecting Figure 3 and, the plot baseline centre (~0.037373) is equivalent to the system temperature (110°K) and the scale is linear, we can calculate the Vela pulsar peak (0.03907) equates to 114.99°K. The source equivalent peak pulse power in Janskys appears to be, 4.99/0.154 = 32.4Janskys.
The baseline ripple is assumed zero mean and to have no effect on these calculations, but may account for some disagreement with published data. As a secondary check, the SNR measured from the data in Figure 3 is 99.24, using Equation 1, T_p is calculated as 3.45°K, or 22.4Janskys.

Base line level has been known to vary with the RTL SDR temperature, which should be closely controlled[4]; calibration tests have not always produced a consistent result.

Evaluation

Initially, 46 files relating to the Vela pulsar were collected in *rtl_sdr*, '.bin' files at sizes 100MB, 200MB, 400MB, 500MB, 800MB, 1000MB, and nominally 1600MB. The 1600MB files were truncated to just over 1GB due to the byte size limitation of integers in 'C' code derivatives**. Nevertheless the files produce were of excellent quality and more than sufficient for the present task. All data was collected in a 2MHz RF range within the Hydrogen line 1420MHz band using a RTL2832U DVB-T dongle tuned within a 150MHz IF band.

Figure 2 shows the Vela pulsar pulse power integrated over a 50 second 100MB file, combining some 560 pulsar pulses. The rotation period used with software *rapulsar.exe* to synchronise and integrate the data series of pulsar pulses was 89.3905ms.

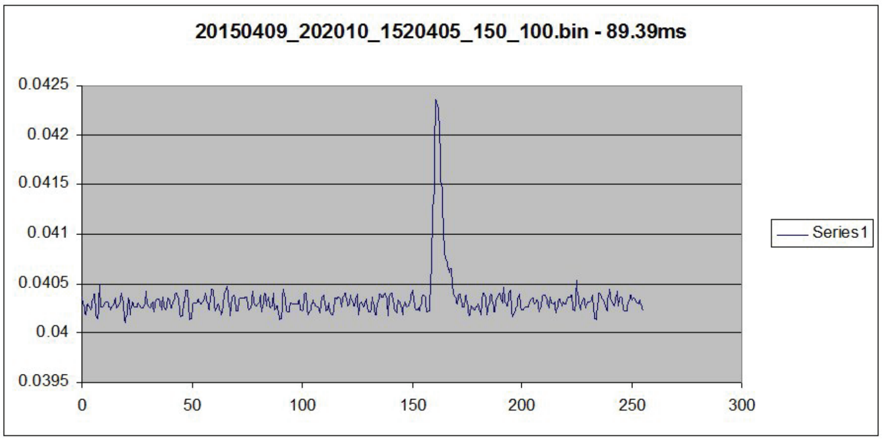

Figure 2 Synchronously Integrated Power, 50sec of Vela Pulses - 256 time bins

The software processing command line to produce this plot was,(see Software Appendix)

rapulsar 100.bin 100a.txt 2.0 256 89.3905

Using pulsar period values slightly either side of this figure broadens the response until the pulse is washed out and the base noise increases.

Integer multiples of this value inserts a corresponding number of pulses in the plot. When close to the correct period, but using twice or three times the expected period, plot symmetries can aid the period search task. Search time can also be shortened using *rapulsan.exe* on long files to reduce the amount of file data processed. Selecting a low (<100) number of time bins also helps period matching if power ripple or regular interference is present.

The period figure, found for the present data, exceeds recognised published values for Vela rotation[2], possibly attributable to the RTL SDR clock accuracy.

Figure 3 is the result of integrating a 1GB file of some 500seconds duration. Much improved signal-to-noise ratio is observed

** G Gancio has modified Osmocom tool 'rtl_sdr' (for linux –> rtl_sdr2) to overcome this limitation, allowing bin files to be recorded in excess of 30GB. Software developed in this project has also been modified to accept these larger files.

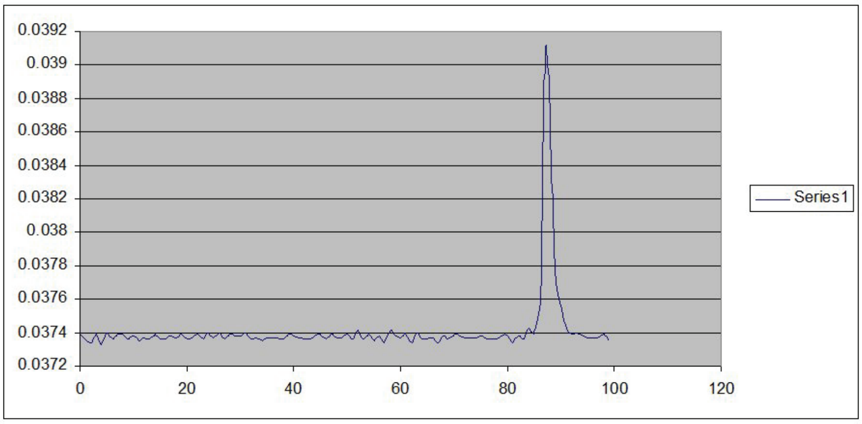

Figure 3 500sec Synchronous Power Integration - 100 time bins

Figure 4 depicts the same response except as an experiment, the pulse power data has been artificially reduced in the plot by a factor of 25 as might be expected from a dish of 6m diameter.
This pulse level is suggested as just sufficient for an amateur to do some reasonable work. Once an integrated pulse has been identified, there would be no problem in overcoming the RTL bin file 1GB limitation and stacking file results to obtain improved signal-to-noise ratio.

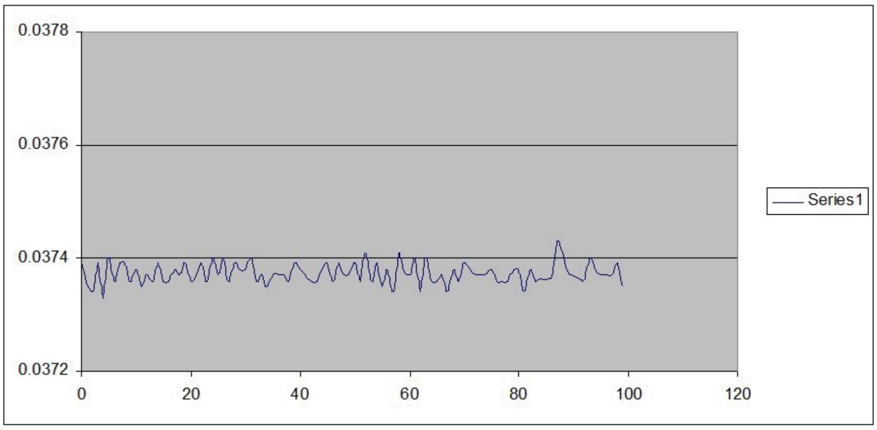

Figure 4 500sec Synchronous Integration (T_p/25; SNR~x4)

It is noted in Figure 2 that the baseline appears to exhibit some regular variation and this was shown to be caused by very low level, possibly power supply, ripple modulating the data. Figure 5 shows this feature using square-law detection software to show a short data time sequence.
In Figure 5, the red curve is the raw data using the *pdetect.exe* program and shows the unwanted ripple in the baseline clearly. *pdetect* simulates a square-law detector, in this case, set for a 1kHz video band width and performs this function by averaging the power in blocks of data numbering 1/video bandwidth, and outputting the results at the end of each block. The red amplitude plot is such that individual pulsar pulses are obscured. To assess the effect of the ripple not being present, a sine wave of the same frequency and phase (blue) is simulated and subtracted from the raw data to produce the lower black plot.
This still contains unwanted noise but individual pulses can now be observed, for clarity, aligned to the negative peaks of the magenta sine wave, which has been adjusted for the pulsar rotation period of 89.39ms.

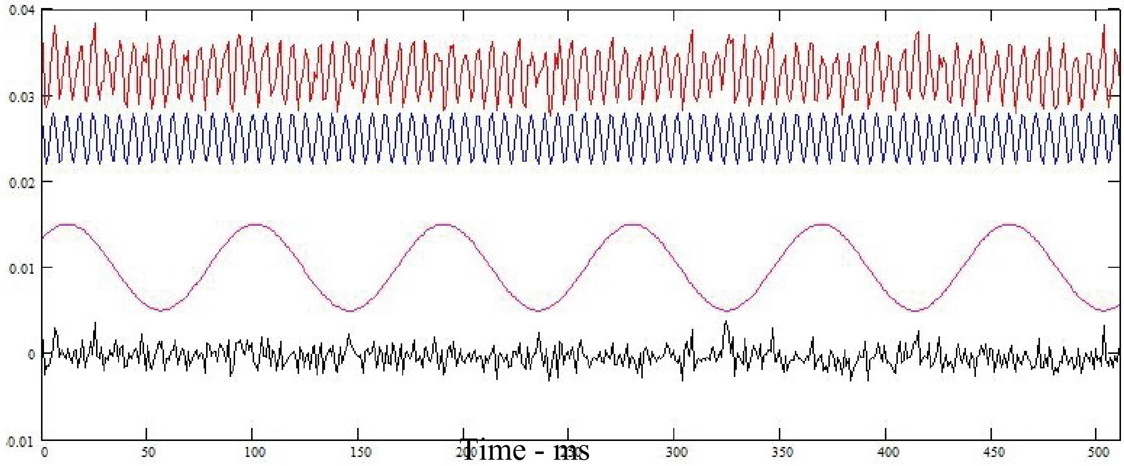

Figure 5 Time-base Plot of a Typical Data File

Determining Pulsar Rotation Rate

In principle, the pulsar rotation period and rate can be determined from the timebase plot, using the *pdetect* program by identifying pulsar pulses and measuring the pulse arrival times. Most pulsars are not so powerful however and may not be visible, but spectral integration using long or multiple averaged FFTs can recover the rate information.

The pulsar signal is effectively pulse modulated wideband noise and indistinguishable from the receiver system noise. The detection process demodulates this to a DC pulse train repeating at the pulsar rotation period, plus video noise. The resulting pulse train FFT spectrum comprises a number of spectral lines spaced at 1/(pulsar period) together with broadband system and pulsar video noise spread over the FFT spectral range. Detection of the system noise also produces a constant DC component in all FFT bins.

The signal-to-noise ratio of pulsar components can be improved in either of two ways; one by increasing the FFT length and the second by averaging multiple shorter FFTs. Both methods are equally effective in improving the SNR; longer FFTs giving better resolution. FFT averaging is feasible over many processed data records to get sufficient integration for weak pulsar signals.

Best period accuracy is assured using the frequency of the highest identified period harmonic.

If 'k_n' is the pulsar FFT data n^{th} harmonic bin number, the pulsar rotation rate is given by, $R_p = k_n B_v/(nP)$ or the period $T_p = nP/ k_n B_v$.

where, 'B_v' is the video band specified in the *pdetect* command line and 'P' is the number of FFT points/bins.

The real data FFT spectrum plot covers the frequency range 0 to $B_v/2$ with a frequency interval/bin of B_v/P; the best choice of these parameters depends on the pulsar rotation rate and timing accuracy expected. Ideally there should be several video samples ($1/B_v$) within the pulsar pulse and the number of FFT bins P should be large enough so that the first pulsar spectral line (at $P/T_p B_v$) is far enough away from zero to be clearly visible.

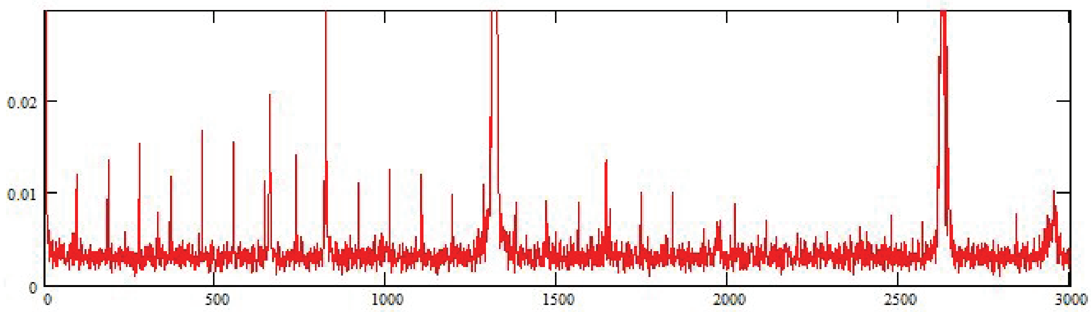

Figure 6 Average of 4 x 8192-point FFTs

Figure 6 shows the average data of 4 consecutive 8192-point FFTs derived from 1kHz bandwidth *pdetect* data (from *rtl_sdr* IQ sample pairs at 2MHz rate), the pulsar rotation rate harmonic spurs are clearly evident, as are some large interference components (around bin 1326 and 2652, equivalent to ~ 162 and 324Hz).
The 31st harmonic lies in bin number 2841. The Vela pulsar rate and period are then derived as described above; period = 31 x 8192/2841 = 89.388244ms – very close to the value found experimentally using *rapulsar* program.
The result using all the data in a single 131,072-point FFT is 31 x 131072/45455 = 89.39021ms.
The program *pafft.exe* enables the various key parameters to be chosen and outputs a text file for Excel to produce a spectrum display similar to Figure 6. The parameters can be varied to optimise the presentation and aid the estimation of the pulsar rotation rate/period.

Conclusions

Useful data has been collected on the powerful Vela pulsar with a 30m dish facility using an RTL2832U SDR dongle. Simple integration software based on a known/derived test value of the pulsar period has shown to effectively improve the pulse to system noise ratio so that pulse detail can be observed.
For the observed conditions and resulting signal to noise ratio, a reasonable estimate can be made of the minimum useful antenna aperture for amateurs to effect detection. For the Vela pulsar this aperture estimate, from the process leading to Figure 4, is 6m.
Bearing in mind that the Vela pulsar is probably the most powerful that has been discovered (maybe not as strong as calculated here) and most pulsar powers are much less than 1 Jansky, opportunities for amateur radio astronomers appear limited.
Synchronous pulsar pulse integration is effective and can be extended to several hours if necessary. Enabling software includes developments to accommodate much larger files or to collate many short (1GB) sections.
Other routes to improving pulsar detection with smaller dish sizes are to reduce the system noise temperature, the use of a wider RF bandwidth SDR/bank of RTL SDRs, or alternatively moving a lower RF band.
For Vela, an improved 50°K system temperature (x2.2) coupled with a 10MHz RF bandwidth SDR (x√5) or 5 suitably tuned 2MHz RTL SDRs*** together with processing 30GB data files (x√30) would enable a 2.2x√5x√30 antenna area reduction This brings the dish diameter required to about 5.8m to match the performance to the present 30m dish system. A 3.5m aperture with these improvements could achieve an SNR of 36 compared to 99 in Figure 3. 30GB need not be collected in one session but the sensitivity can be recovered from an equal set of smaller files. Smaller files must be sufficient to identify and

*** Stability of multiple RTL SDRs can be much improved with air cooling(4). Uncooled, the tuned rf frequency and gain can drift significantly during operation.

correct the pulse phase so that re-alignment and synchronous adding/averaging can be performed.
PSR B0329+54 is the strongest pulsar in the northern hemisphere and is logged at about 1/5 of the power of the Vela pulsar but should still be detectable with an optimised 3.5m dish system. This may still be within many amateur radio astronomer's capability, especially if prepared to combine several 30G observations.
Wider band SDRs (RASDR or Airspy) and lower frequencies may require de-dispersion pre-processing.
Finally, longer term tracking and integration, reduced system temperature, reduced interference, lower frequencies, de-dispersion and wider SDR bandwidth coupled to an antenna of 3.5m upwards seems to offer the amateur way ahead.

Postscript
This has been very much a project undertaken with minimal pulsar experience. However, pulsar data recorded using a cheap RTL DVB-T dongle has responded to some fairly basic signal processing algorithms although some results are at variance with published data. With design care plus a tracking 3.5m dish system, it seems feasible for amateurs to detect and analyse the stronger pulsars. For the future, there are a number of areas that could be explored for the further development, including,
1. De-dispersion, by coarse FFT channelling and post integration combination.
2. Filtering interference, or,
3. Pulsar rate FFT spur filtering + inverse FFT prior to synchronous integration.
4. Multiple SDR injected-noise timing management with RasPi back-end processing[+].
5. Investigate amateur use of Tempo2.
May many others get the thrill from seeing a pulsar pulse shape appear out of noise as the rotation period is adjusted to the perfect value.

References

1. http://www.iar.unlp.edu.ar/index.html
2. http://www.atnf.csiro.au/research/pulsar/psrcat/
3. http://zmtt.bao.ac.cn/pulsar_cof_beijing/afternoon_509/mkeith_search.pdf
4. http://www.y1pwe.co.uk/RAProgs/FrequencyStabilityU2.doc
5. http://iopscience.iop.org/0004-637X/498/1/365/fulltext/

[+] see Appendix 3

Appendix 1. - Software[****]

***rapulsar2.exe* - Synchronous Integration**

This software processes RTL .bin data collected with an RTL SDR receiver system.
It integrates the IQ power data synchronously on a timebase adjusted to or very near to the pulsar period. The result is a text file that can be viewed with Excel or mathcad programs showing the integrated pulse power shape at some arbitrary phase dependant on the initial file timing conditions.
A typical command line is....

rapulsar2 <datfile.bin> <outfile.txt> < RTL data rate (MHz)> <No. data points> <Pulsar period (ms)>

For large files when adjusting the pulsar period, a smaller subset can be inspected using the modified program....

rapulsan2 <datfile.bin> <outfile.txt> < RTL data rate (MHz)> <No. data points> <Pulsar period (ms)> <File divisor>

The file divisor integer shortens the processed file to the file divisor factor.

***pdetect2.exe* - Square-law Detector**

This software processes RTL .bin data collected with an RTL SDR receiver system.
It acts as a software, square-law detector. The result is a text file that can be viewed with Excel or mathcad programs showing the video integrated detected power - time response. It integrates data in blocks proportional to the inverse of the video band figure and outputs the block result.
A typical command line is....

pdetect2 <datfile.bin> <outfile.txt> < RTL data rate (MHz)> <Video Band (kHz)> <File divisor>

As before, the file divisor integer shortens the processed file to the file divisor factor.

***pafft2.exe* - Pulsar Rotation Spectrum**

This software processes RTL .bin data collected with an RTL SDR receiver system.
It calculates and averages FFT spectra formed from square-law detected data as specified in the command line. The result is a text file that can be viewed with Excel or mathcad programs showing the pulsar pulse spectrum, from which rotation rate harmonics may be identified and the rotation rate/period can be calculated.
The period information can then be input to *rapulsar* to seed accurate timing estimation.
A typical command line is....

pafft2 <datfile.bin> <outfile.txt> < RTL data rate (MHz)> <Video Band (kHz)> <FFT size>

Software Link

http://www.y1pwe.co.uk/RAProgs/NewSW2.zip

[****] Latest versions can now process .bin files much larger than 2GB, identified with the post nominal '2'.

Appendix 2. Pulsar J0437- 4714 Experiment

This appendix summarises RTL SDR results for a weaker, faster rotation rate pulsar - J0437-4714 using the software developed for this project.
Note that the pulsar spectral lines are very small compared to the interfering signal level and requires 1GB files and expected harmonic component relations to identify them.
Techniques to accumulate harmonic components have been described(see Reference 3).

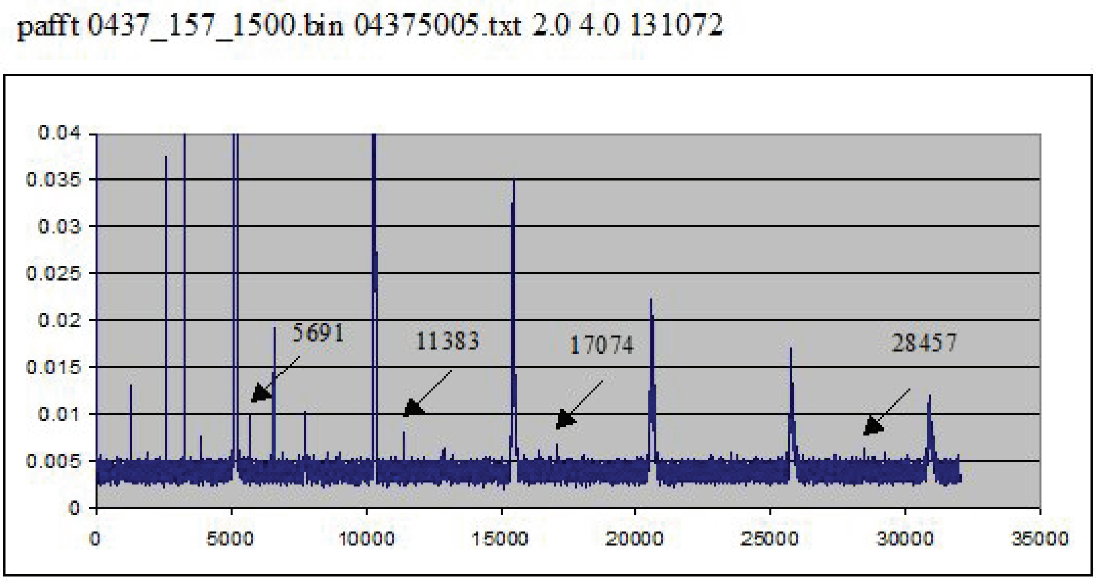

Figure A1 Detected Power Spectrum - J0437-4714

Calculated rotation period, T_{fft} = 131072 x 5/28457/4.0 = 5.75746ms.

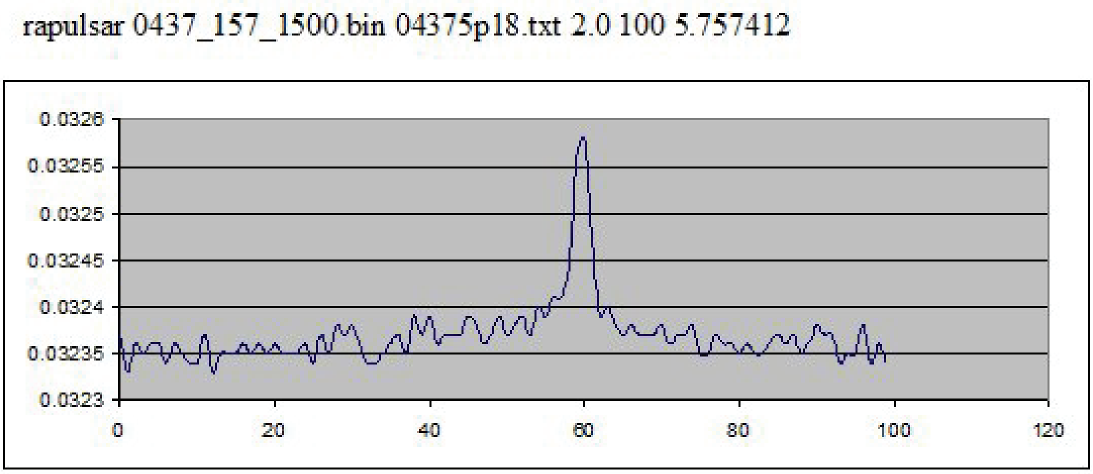

Figure A2 Synchronous Integration - J0437-4714

Optimum period result by trial and error, T_{trial} = 5.757412ms. SNR_v ~x20.

From Figure A2, pulsar peak temperature = 110° x0.03258/0.3235 = 0.78°K, o r ~ 5.06Jy. This compares favourably to the peak average expected of 5.6Jy from Reference (5), Radio Pulse Properties of the Millisecond Pulsar PSR J0437-4715. I. Observations at 20 Centimeters., Jenet, FA, et al., 1998.

Appendix 3 - Quad RTL SDR Pulsar receiver
Introduction
Multiple RTL SDR's locked by a single master crystal oscillator and ambient air cooled are an economical approach for multi-baseline interferometry and high sensitivity wide-band pulsar detection receivers. Figure A3 shows a 4-channel digital receiver block diagram producing complex data files driven and collected by Raspberry Pi mini-computers. For interferometry, the SDR's are tuned to the same frequency and calibrated to phase track. For Pulsars or wide band operation, the SDR's are tuned to adjacent bands. Calibration and synchronism of data files is similar in both applications. The receiver cost, less the controlling PC, is around £300.

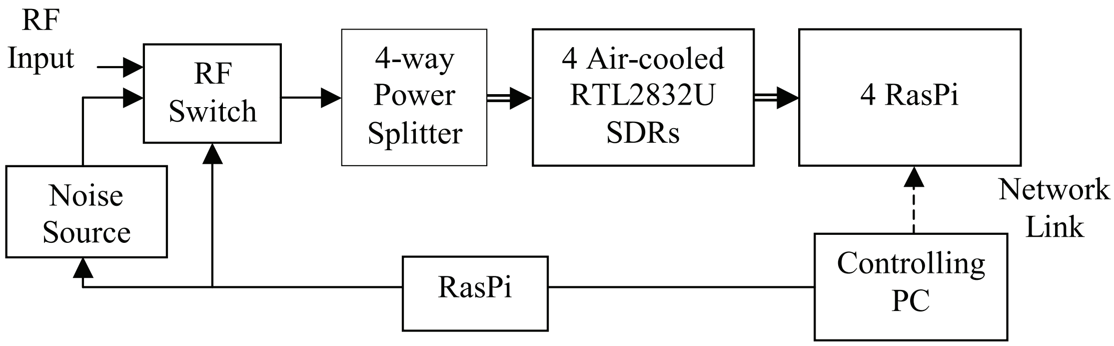

Figure A3 10MHz Bandwidth 4xSDR Digital Receiver

Hardware Notes
1. RF switch - 1-pole, 2-way, electromechanical (ms switch speed) or PIN (us switch speed). The latter type may be best for more accurate file offset correlation (Mini-Circuits, ZMSW-1211 is suitable ~£60).
2. Noise source - (ebay SMA noise tracking source ~ £15).
3. Power splitters - three 2-way or one 4-way RF power divider.
4. Bare RTL2832U's with mini heatsinks on all ICs, run from a single temperature compensated crystal at 28.8MHz, all mounted in an air-cooled metal box(4).
5. RasPi with 64GB memory (Sony micro SDXC or USB).

Operation
PuTTY[a] on the controlling PC manages operations on all RasPi's via SSH[b]. Programs are run in RasPi terminals via the controlling PC. *rn_rtlat2* runs the Osmocom *rtl_sdr* data acquisition program at a set time to nearly synchronise all 4 SDR data collection.
The delay between start command and the program run time may vary due to the processor housekeeping, clock requests and clock accuracy. Delays are unpredictable but limited to few or low tens of milliseconds.
On completion of data recording, data files are downloaded to the controlling PC for final processing. Accurate synchronism can be achieved by switching on and off a noise signal and using correlation software to identify offsets required to synchronise SDR data files. Synchronism accuracy for pulsars only need to be a fraction of the pulse dwell whereas alignment within a clock cycle is necessary for interferometry to limit phase offset which can be achieved by noise amplitude correlation.

---

[a] http://www.putty.org/
[b] apt-get install openssh-server openssh-client

Operation Sequence - Pulsars
Data collection
1. Set switch to select noise source.
2. Start 4 SDR's recording using *rn_rtlat2* at programmed time, 4 SDR's with set data rate at 2.5MHz to cover 10MHz band.
3. Turn on noise using 5th RasPi with *rn_prgat.*
4. Random on-off noise modulation for 1sec to simplify correlation.
5. Switch to signal path.
6. Complete recording.

Data correlation, Matching
1. Select SDR1 data file as master.
2. Run *cor_tim2* on 1+2, 1+3 and 1+4 data files, note noise source timing.
3. Determine file offsets for correlated alignment.
4. Calculate de-dispersion times and adjust offsets.
5. Run *f_align2* on all 4 files to match start positions and file lengths.
6. Run *pafft2* on the four data files, average spectrum data and inspect for rotation spurs.
7. Run *rapulsan2* on files to for synchronous integration of pulsar pulse shape.

Conclusions
The architecture and processes described offer an economical approach to broad-banding and time calibrating a pulsar receiver to improve system sensitivity and also provide a simple and convenient method of de-dispersion.
The method is suitable for extending the frequency band and provides raw data files that can be investigated at leisure.
Data files of particular sources can be collected on multiple occasions and the results added and combined to further increase the receiver effective sensitivity.

PW East, GM Gancio. May 2015