The Wilbur Single Dispersed Pulse Search System Ver. 1

J. Dowell[*] G. B. Taylor[†]

August 31, 2020

Contents

[*]University of New Mexico. E-mail: jdowell@unm.edu
[†]University of New Mexico. E-mail: gbtaylor@unm.edu

1 Introduction

The purpose of this document is to describe that Wilbur Single Dispersed Pulse Search System system that is deployed at the Long Wavelength Array station located on the Sevilleta National Wildlife Refuge (LWA-SV). This includes not only the data products and how they are made but also the hardware that the software runs on.

Briefly, Wilbur is a realtime system for searching for single dispersed pulses over the dispersion measure (DM) range of 10 to 500 pc cm^{-3}. The system runs commensally on beam 1 and uses the fast dispersion measure transform (FDMT; [3]) for searching.

2 Hardware

The Wilbur Single Dispersed Pulse Search System runs on a 4U Silicon Mechanics Rackform R2504.v7 platform. The machine has:

- Two 8-core Intel Xeon Silver 4208 processors with a base clock of 2.10GHz,
- 96 GB of 2666 MHz DDR4 memory,
- Two Nvidia Titan RTX GPUs with 24 GB of memory each,
- A 46 TB RAID6 array,
- A 480 GB Intel Optane 900P NVMe drive, and
- A Mellanox ConnectX-3 10GbE fiber card.

The Wilbur software is bound to one of the CPUs and one of the GPUs, uses the NVMe drive as a large data buffer, and writes raw DRX data corresponding to events to the RAID6 array. The other resources on the machine are used by the Orville Wideband Imager system which is described in LWA Memo 215.

3 Software

The Wilbur is written in Python using the Bifrost framework (Cranmer et al. [1]) and the source code is available on GitHub at `https://github.com/lwa-project/wilbur_frb_search`. Figure 1 shows the Bifrost blocks used for Wilbur. At the top of the diagram are the four data sources, one for each of the four tuning/polarizations of beam 1 from the ADP backend at LWA-SV. These data are 4+4-bit complex integer voltage time series data with a sample rate of up to 19.6 MHz.

Once the data have been received and reordered by the `udp_capture` block, there are three blocks that make use of the data. The first is the `SpectrometerOp` block. This block takes the time domain data and generates XX and YY averaged spectra with integration times ~ms. The spectra are then sent to the `BandpassOp` block where they are further processed. In this block the data are:

1. At the start of each unique frequency setup the number of channel ranges of DTV channels are determined. These will be used later in the processing for excising DTV flares that reflect off meteor trails in the upper atmosphere.
2. The higher of the two tunings is selected and only data from that tuning is kept. This is to increase the range of DM that can be searched.

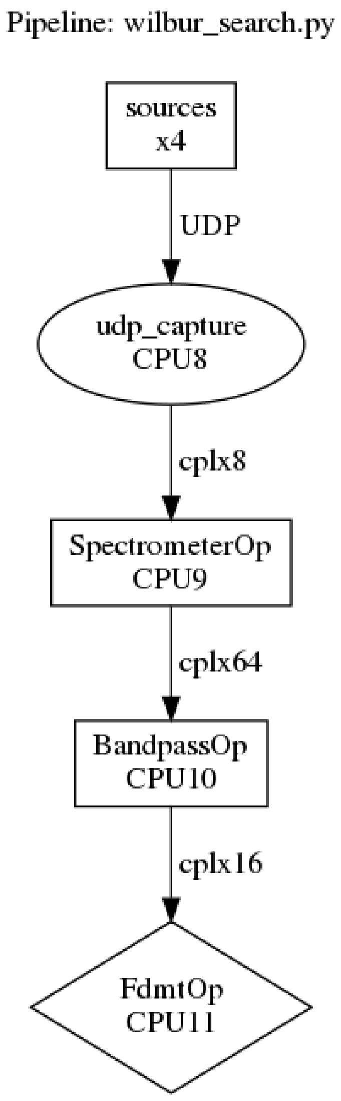

Figure 1: Bifrost block diagram for the Wilbur Single Dispersed Pulse Search System.

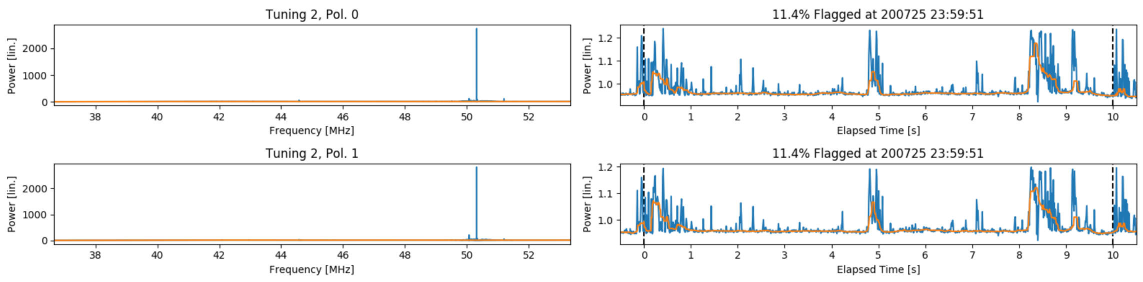

Figure 2: Example `BandpassOp` diagnostic plot. The two panels on the left show the integrated spectra for both XX (top) and YY (bottom) data. The two panels on the right show the drift curve (integrated power as a function of time) for XX and YY. The panels on the right are labeled by flagging fraction and time.

3. The data are bandpassed to remove the spectral response of the telescope. This is done by:
   (a) Creating a mean bandpass for each tuning/polarization in the data by averaging each bandpass section of data over time.
   (b) Smoothing the mean bandpasses in frequency with a median filter to remove any narrow spikes in the data.
   (c) Dividing the data by the filtered bandpasses to remove the frequency response of the instrument. The data should be mean 1 at this point.
   (d) Flagging any points in time/frequency that are $\gg$1 or $\ll$1 since these are likely transient features.
4. DTV flares are flagged by:
   (a) Looking at the power in the pilot tone for each DTV channel in the data and comparing it with the power in a nearby clean section of the DTV channel. The pilot carrier is located 309.4 kHz above the lower edge of the channel.
   (b) Flagging the full DTV channel for any time where the power in the pilot is some multiplicative threshold greater than the clean section.
5. The drift curve (integrated power as a function of time) is flattened to remove power variations. This is done by:
   (a) Creating a mean drift curve for each tuning/polarization in the data by averaging the data over frequency.
   (b) Smoothing these mean drift curves in time with a median filter to remove any narrow spikes in the data.
   (c) Subtracting these filtered drift curves from the data to remove the overall power drift in the instrument. The data should be mean 0 at this point.
   (d) Flagging any points in time/frequency that are $\gg$0 or $\ll$0 since these are likely transient features.
6. After the data have been cleaned the XX and YY polarizations are summed to form Stokes I.
7. Finally, the Stokes I data are scaled and re-quantized from 32-bit floating point to 8-bit integers to save memory on the GPU where the FDMT search occurs.

As part of its processing the `BandpassOp` also writes out diagnostic plots for the data quality. An example of one of these plots is show in Figure 2

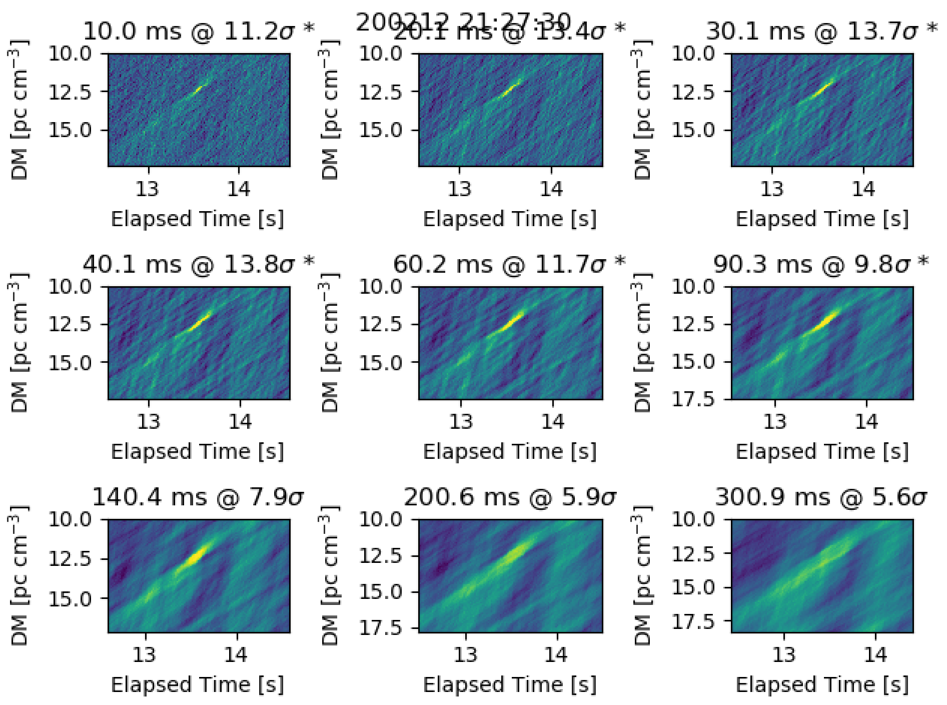

Figure 3: Example event detected by Wilbur. Each panel represents a different pulse width and are labeled by the pulse width in ms and the strength of the detection. This particular event is a single pulse from PSR B1919+21 and the characteristic "bow time" is clearly seen in pulse widths from 10.0 and 140 ms with a peak detection at width of 40 ms.

The final stage of the Wilbur processing is to run a FDMT search for pulses at a variety of pulse widths. This block does the actual search for single dispersed pulses in the data using the fast DM transform method of Zacky & Ofek [3]. This is done by:

1. Loading in the data from `BandpassOp` and transposing the axis order.
2. Running the FDMT algorithm on the data to create a search space in DM.
3. Smoothing the data with boxcar filters of different widths from 10 to 300 ms to do a basic matched filter on pulse width.
4. Searching for peaks in the DM–time domain across the different pulse widths. Each peak is then converted to a significance using the mean and standard deviation of the data. Peaks above the pipeline's signal-to-noise threshold, S, are considered events. Peaks in the range of $[S-1, S)$ are considered sub-events.

The results of this are shown in Figure 3.

4 Data

For any event identified by the search pipeline, the raw DRX data is dumped from the large NVMe buffer. The data are written out as standard DRX files with filenames that correspond to the event time and peak DM. It should be noted that these files do not contain the exact same packets as

Wilbur received. Rather, they are recreated by Wilbur from what is stored in the NVMe buffer so that the file corresponds to a set amount of time both before and after the event. For sub-events only the diagnostic plots, like the one shown in Figure 3, are generated.

5 Future Directions

The Wilbur system is still under development in the areas of RFI rejection, event detection, and overall goals. For the RFI flagging the open issues are:

- Is the current scheme of flagging in time and frequency, and searching for DTV flares sufficient?
- Are the flagging parameters tuned appropriately for typical running conditions?

For event detection the main issue is how to move from the simple "highest peak" approach to something more sophisticated that considers the data jointly across different pulse widths. This could include some kind of machine learning-based triage to further validate candidates.

For the overall goals the question is whether or not it makes sense to continue using Wilbur in a blind search mode or if it better to run it in a targeted mode with coherent dedispersion.

6 Document History

- Version 1 (Aug 31, 2020): First version.

References

[1] M. Cranmer et al., "Bifrost: a Python/C++ Framework for High-Throughput Stream Processing in Astronomy", 2017, JAI, 650007.

[2] J. Dowell, S. Varghese, & G. B. Taylor "The Orville Wideband Imager," Ver. 1, Long Wavelength Array Memo 215, Aug 28, 2020. [online] `http://www.phys.unm.edu/~lwa/memos/index.html`.

[3] B. Zackay & E. Ofek "An Accurate and Efficient Algorithm for Detection of Radio Bursts with an Unknown Dispersion Measure, for Single Dish Telescopes and Interferometers", 2017, ApJ, 385, 11.