The Journal of Open Source Software

WILL - Weighted Injector of Luminous Lighthouses

Joseph W Kania [1,2,3] **and Kevin Bandura** [3,4]

1 Jodrell Bank Centre for Astrophysics, Department of Physics and Astronomy, The University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom **2** West Virginia University, Department of Physics and Astronomy, P. O. Box 6315, Morgantown 26506, WV, USA **3** Center for Gravitational Waves and Cosmology, West Virginia University, Chestnut Ridge Research Building, Morgantown 26506, WV, USA **4** Lane Department of Computer Science and Electrical Engineering, 1220 Evansdale Drive, PO Box 6109, Morgantown, WV 26506, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 08 September 2025
Published: unpublished

Summary

Radio transients, including Fast Radio Bursts (FRBs) (Lorimer et al., 2007), Rotating Radio Transients (RRATs) (McLaughlin et al., 2006), and pulsars have gathered significant interest. These sources produce bright pulses that last on order of milliseconds. The community interest in these sources has lead to many dedicated transient backends. These backends have greatly increased the sky-hours searched, leading to increased the rate of discovery for these sources. This trend will continue as new multibeam receivers and telescopes are built. As more bursts are discovered, more science can be extracted from the sources, their environment, and matter the bursts pass through traveling to Earth. To accomplish this science we need to understand the search pipeline. These pipelines contain many steps. The signal must be received by the telescope, then possibly filtered, and finally digitized. The digital is cleaned of Radio Frequency Interference (RFI). The signal has pass through ionized media, this causes a quadratic delay in arrival time as a function of frequency. This must be corrected for a range of possible Dispersion Measures (DMs). The DM-time matrix is then searched over a range of pulse widths. The candidates are then clustered together to report one candidate that may have show up at multiple dispersion measures or widths. See Barsdell (2012) for a discussion of a search pipeline on a Graphical Processing Unit (GPU). Candidates are then reviewed by humans or machine learned models such as Agarwal, Aggarwal, et al. (2020).

To understand the pipeline selection function, the pipeline needs to be tested over a wide variety of pulse morphologies. Pulses can vary in sky location, arrival time, time duration, center frequency, frequency width, number of scintillation patches, scintillation phase, spectral index, scatter time, dispersion measure, and brightness. Even millions of pulses will undersample this eleven dimensional space. (Although not all of these dimensions have the same impact of the pulse search.) Getting millions of pulses is not an easy task, new instruments will not have observed many objects and data from other instruments may not have comparable properties. Also objects might not have been found with certain properties, for example the the highest DM FRB reported on the Transient Name Server is 3038 DM. To understand the sensitivity of searched to higher DMs, we need to make synthetic pulses, which can be done with `WILL`.

Radio Frequency Interference (RFI) are anthropomorphic signals that are inadvertently received by radio telescopes. RFI can degrade the observation by obscuring the astronomical signal, and can produce false positive candidates. There are many RFI removal algorithms, some built into the pipeline (Barsdell, 2012; Ransom, 2001), others as stand alone packages (Kania & Bandura, 2022; Maan et al., 2021; Morello et al., 2022) that clean the data before the pipeline. Ideally these filters completely remove RFI while retaining all of the pulse energy. This does not happen in practice, some RFI remains and bright parts of the pulse are removed. To better understand how these filters interact with the pulse, we can create fake pulses and run them through a search pipeline. We can also use `WILL`'s pulse detection to see how the signal to

noise level changes for a given width and DM.

Statement of Need

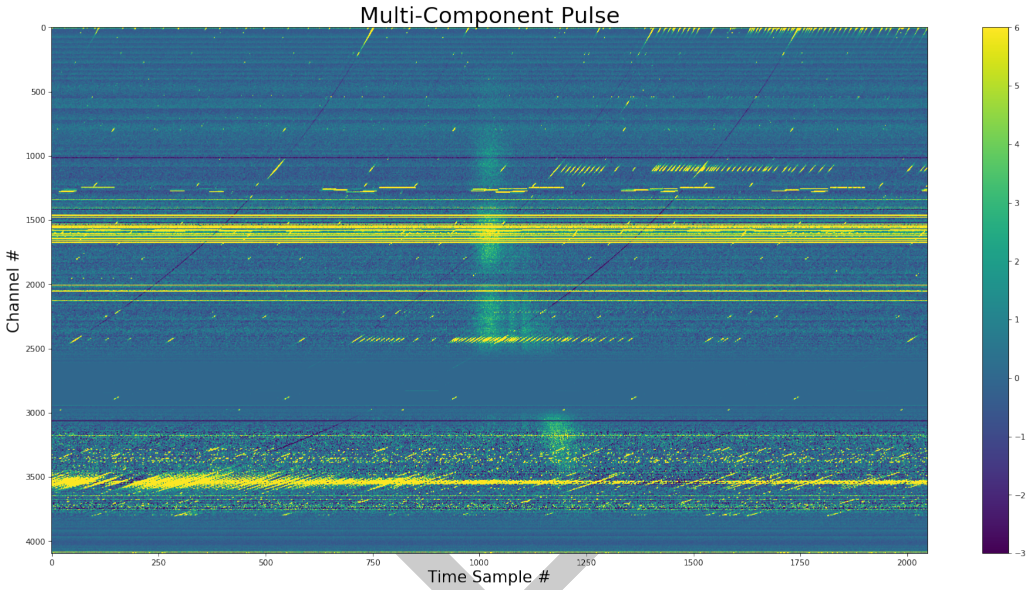
We looked for pulse simulation software, and we found seven existing pulse simulators. However none met all of our needs. Many had missing or incomplete documentation, making it impossible to know how to use the software or port it to our data. Other burst simulators failed to run on our data. Finally some simulators only work on synthetic noise, this includes both pulse simulators. We then proceeded to develop `WILL` to overcome these problems. Will uses Aggarwal et al. (2020) to write and write file files, allowing `WILL` to read both Filerbank (Lorimer, 2011) and PSRFITS (Hotan et al., 2004) files. We also have a documentation website which has function documentation as well as example notebooks showing how to create & inject pulses, pulse detection, pulsar analysis, and an example showing `Will` being used with SciPy to optimize filter inputs.

While writing `WILL` we came up with several improvements to the fidelity of the synthetic pulses. Many pulse simulators attempt to inject a pulse at a given Signal to Noise Ratio (SNR), the notable exception being FRB Faker. The problem with this injection methodology is that RFI can drastically change the noise level. Injecting at constant SNR will mean the pulse will also get brighter. This will lead to overestimating the sensitivity and uniformity of the search pipeline. `will.create` creates a pulse profile and then samples that profile for a given number of samples. Thus the total power in the pulse in know, the pulse can then be added to a dynamic spectra. `will.detect` can be used to then calculate the SNR, if desired. A unique feature of `will.create` is the ability to take into account the changing sensitivity of radio telescopes. This change can be due to band roll off or band stop filters. The sensitivity weights can be user provided and we have functions that can estimate weights. Taking into account sensitivity changes is important when evaluating the effectiveness of RFI filters. The variance of channels that are blocked by a bandstop filter are close to zero. A filter may then remove any pulse found here because the pulse does not follow the surrounding statistics. This would lead to the incorrect conclusion that the filter is malfunctioning. `will.create` can take pulses fitted with `burstfit` (Aggarwal et al., 2021) and reinject them.

`will.detect` provides straightforward pulse detection at given width and DM. This allows analysis of pulsar pulses over an observation. You can also see the folded pulse dynamic spectra, time series, and SNR. You can use this functionality to see how see how the pulses respond to RFI filters using a full pulse search pipeline, reducing the degrees of freedom searched.

`WILL` uses NumPy (Harris et al., 2020), rich, SciPy (Virtanen et al., 2020), jess (Kania & Bandura, 2022), and your (Aggarwal et al., 2020). We found `injectfrb` (Connor, 2020) to have useful pulse modeling functions, these were used in (Agarwal, Lorimer, et al., 2020), and built upon in `will.create`.

Figure



A dedispersed four component pulse created with `WILL`, the last component is rotated compared to the other components. The pulse has no power between channels 2500 to 3000, correctly accounting for the bandstop filter at these frequencies.

Acknowledgment

`WILL` was developed with support from NSF Award 2006548. We thank Devansh Agarwal and Kshitij Aggarwal for useful discussions.

References

Agarwal, D., Aggarwal, K., Burke-Spolaor, S., Lorimer, D. R., & Garver-Daniels, N. (2020). FETCH: A deep-learning based classifier for fast transient classification. *497*(2), 1661–1674. https://doi.org/10.1093/mnras/staa1856

Agarwal, D., Lorimer, D. R., Surnis, M. P., Pei, X., Karastergiou, A., Golpayegani, G., Werthimer, D., Cobb, J., McLaughlin, M. A., White, S., Armour, W., MacMahon, D. H. E., Siemion, A. P. V., & Foster, G. (2020). Initial results from a real-time FRB search with the GBT. *497*(1), 352–360. https://doi.org/10.1093/mnras/staa1927

Aggarwal, K., Agarwal, D., Kania, J., Fiore, W., Thomas, R., Ransom, S., Demorest, P., Wharton, R., Burke-Spolaor, S., Lorimer, D., Mclaughlin, M., & Garver-Daniels, N. (2020). Your: Your Unified Reader. *The Journal of Open Source Software*, *5*(55), 2750. https://doi.org/10.21105/joss.02750

Aggarwal, K., Agarwal, D., Lewis, E. F., Anna-Thomas, R., Cardinal Tremblay, J., Burke-Spolaor, S., McLaughlin, M. A., & Lorimer, D. R. (2021). Comprehensive analysis of a dense sample of FRB 121102 bursts. *arXiv e-Prints*, arXiv:2107.05658. https://arxiv.org/abs/2107.05658

Barsdell, B. R. (2012). *Advanced architectures for astrophysical supercomputing* [PhD thesis]. Swinburne University of Technology.

Connor, L. (2020). *Injectfrb*. https://github.com/liamconnor/injectfrb

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hotan, A. W., van Straten, W., & Manchester, R. N. (2004). PSRCHIVE and PSRFITS: An Open Approach to Radio Pulsar Data Storage and Analysis. *21*(3), 302–309. https://doi.org/10.1071/AS04022

Kania, Joseph, & Bandura, K. (2022). *High cadence radio frequency interference filters*.

Lorimer, D. R. (2011). *SIGPROC: Pulsar Signal Processing Programs* (p. ascl:1107.016). Astrophysics Source Code Library, record ascl:1107.016.

Lorimer, D. R., Bailes, M., McLaughlin, M. A., Narkevic, D. J., & Crawford, F. (2007). A Bright Millisecond Radio Burst of Extragalactic Origin. *Science*, *318*(5851), 777. https://doi.org/10.1126/science.1147532

Maan, Y., van Leeuwen, J., & Vohl, D. (2021). Fourier domain excision of periodic radio frequency interference. *650*, A80. https://doi.org/10.1051/0004-6361/202040164

McLaughlin, M. A., Lyne, A. G., Lorimer, D. R., Kramer, M., Faulkner, A. J., Manchester, R. N., Cordes, J. M., Camilo, F., Possenti, A., Stairs, I. H., Hobbs, G., D'Amico, N., Burgay, M., & O'Brien, J. T. (2006). Transient radio bursts from rotating neutron stars. *439*(7078), 817–820. https://doi.org/10.1038/nature04440

Morello, V., Rajwade, K. M., & Stappers, B. W. (2022). IQRM: real-time adaptive RFI masking for radio transient and pulsar searches. *510*(1), 1393–1403. https://doi.org/10.1093/mnras/stab3493

Ransom, S. M. (2001). *New search techniques for binary pulsars* [PhD thesis]. Harvard University, Massachusetts.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2