Investigating the Efficiency for a Portion of the Hierarchical frequency-Hough Pipeline

Aidan Nyquist

North Park University
Università di Roma: La Sapienza
University of Florida, IREU

Mentors: Pia Astone & Lorenzo Pierini

Università di Roma: La Sapienza

August 20, 2020

The frequency-Hough pipeline has been used to search for various sources of continuous gravitational waves. In this paper, we explore the peakmap creation process by recreating and improving some of the software used in prior searches in a way that allows the verification and inspection the algorithms for past and future experiments. We mainly measure the detection efficiency for different sets of parameters by injecting simulated signals and validating whether went through certain steps of the pipeline.

I. Background

While gravitational wave (GW) detections have become so plentiful that most binary mergers are no longer deserving of their own paper, there are still searches for more elusive continuous gravitational waves (CW). What characterizes CWs, unlike the short and powerful bursts of binary mergers, are their quiet, monotone, and very long lasting 'hum'. They are primarily expected to be produced by asymmetric neutron stars (NS), which to be expected by their name, would have small yet dense imperfections producing CWs when the star is spinning fast. Out of the billion or so to-be-expected NSs in the galaxy, there has only been a few thousand NSs found via telescopes and a handful of NS binary mergers detected via GW detectors [3]. Thus CWs should provide a method for detecting isolated NSs even in the case that they do not emit enough electromagnetic radiation, yet there have not been any signals found in any of the data up till the O2 run. The reasons for these lack of findings are numerous, especially in this context where the source's parameters are unknown. Firstly, not only is there a Doppler shift that cannot be accurately corrected (due to the sources not yet having been found/located), but the act of emitting CW causes the star's rotation to slow down (referred too as spin-down). Thus the frequency shift is two-fold and an unknown, so searches must be done in large *ranges* of frequencies with crossing signals. Secondly, these signals may be indiscernible from noise due to their low amplitudes, and their frequency could be outside of the range of current detectors. Nonetheless, there may be other candidates for a source of a CW if found. Namely, ultralight boson [1] clouds that form around black holes could possibly emit a detectable CW [6]. From the much yet to be learned of the high density environment of NS to beyond Standard Model particles, detecting and analyzing CWs could bring breakthroughs in several areas of research.

[1] Axions, a dark matter canditate that solve the strong CP problem, are example of what ultralight bosons could be.

a. Current Method

It should come at no surprise that there are several CW searches, all uses different techniques on the same data each covering varying parameter spaces [4]. Ours is the frequency-Hough which is a hierarchical pipeline of various signal and image processing techniques. While the fine details [1] are not required for this report, a rough overview goes as follows:

1. Creation of SFDB
   The raw data is split into chunks of 1024 s-16384 s of coherence time where the signal appears more or less of constant frequency during that duration. Thus each chunk can be short-time Fourier transformed (SFT) giving the SFT database (SFDB). Each moment is also assigned metadata such as velocity of the detector, datetime, etc.

2. Autoregression (AR)
   A recursive algorithm (usually done backwards through the frequency bins or right-to-left) smooths the signals with mechanisms to preserve possible candidates.

3. Peakmap creation (PM)
   Peaks are found above a given threshold from the ratio of the original data over the autoregressed data.

4. Hough transform
   The PM is run through a Hough transform—a common pattern recognition technique in image processing. This helps find features of the PM while accounting for a spin-down and Doppler shift.

5. Candidate analysis [2]
   Strong candidates from the Hough transform are refined using a coarse grid in the parameter space, narrowed down to a smaller amount, clustered together, and cross-checked against another search by doing coincidences.

6. Verification
   Any candidates that make it to the end of the pipeline warrant closer inspections, made sure not to be false positives, and ran against a more computationally intensive search.

[2] This is actually several steps, but only the general idea is required for this report.

b. Motivation

As one can imagine, this pipeline is a massive computational undertaking, mostly due to its 'needle in a haystack' problem. Its large amounts of steps brings forth a complicated codebase and even small parts of the pipeline could use further study to verify current results, strengthen future results, and fix inefficiencies or redundant computations. In the process of having to understand what the code is doing, it would be helpful to comment, reformat, and optimize where possible. More specifically, the goal of *this project* is given a SFDB, replicate the AR and PM creation of prior works, inject simulated signals in a variety of ways to better understand the behavior of these algorithms, and modify the parameters and constants to possibly optimize the pipeline - especially for narrow frequency ranges. Although this project will only report sparse regions on a relatively small dataset, it is in the process that the code will be developed that can be reused for more honed analysis and larger computations.

II. Description of Software

What ended up being developed was a library that while used for this report, has the potential to be an alternative to some of the existing code in the current pipeline. It runs on Matlab and was kept compatible with GNU Octave. Besides standard Matlab toolboxes (and their Octave counterparts) the only dependency is the Snag V2 library [5]. A large sum of the code had already been written, but it was often tangled in a way that made it hard to tweak small parts, exchange components, or verify its correctness. So the first task was to not only understand the code, but break it up into smaller documented functions. This allowed for the creation of a test suite that used randomly generated data to compare the results and performance of copyand-pasted fragments of the old code against the refactored or optimized version of corresponding procedures. There were also triple-checks against existing 3rd party libraries when possible. While the optimization of the code was not explored in great detail, the few **functions that were optimized and tested either sped up or maintained the same speed.** This means that there is no computational cost known to adoption of any parts of the codebase. Nonetheless, the room for optimization and per-function testing is still quite vast.

The focus of the project was just on two steps of the frequency-Hough pipeline, so care was taken to remain compatible with the step before and the steps after. It often calls to other modules that are actively worked on by other members of the group such that newer version of their code is seamlessly compatible. Thus the final library provides infrastructure to introspect its own speed, behavior, and correctness, while being easy to incorporate with other steps of the pipeline.

III. Efficient Parameters

In order to measure how well the pipeline performed given some parameters, the efficiency (η) is measured at the frequency of an injected signal by counting the peaks across several PMs (one for every SFDB index) where

$$\eta_f = \frac{\text{Detected peaks at } f}{\text{Injected peaks at } f}$$

and f is some frequency bin across several moments of the SFT. Thus $\eta \leq 1$ because the peaks are indexed by the resolution of the SFT. Signals were injected on a pre-made section of the SFDB for the O2 run.

One of the key components needed was a baseline that would be a useful measure of the background noise across the frequency domain. In order to do this, a numerical solution was found for the amplitude (h) of an injected signal that would correspond to $\eta = .95$. This will give a context for the amplitude of injections in this report, but its relatively small data-set leaves much to be desired for characterizing O2 data.

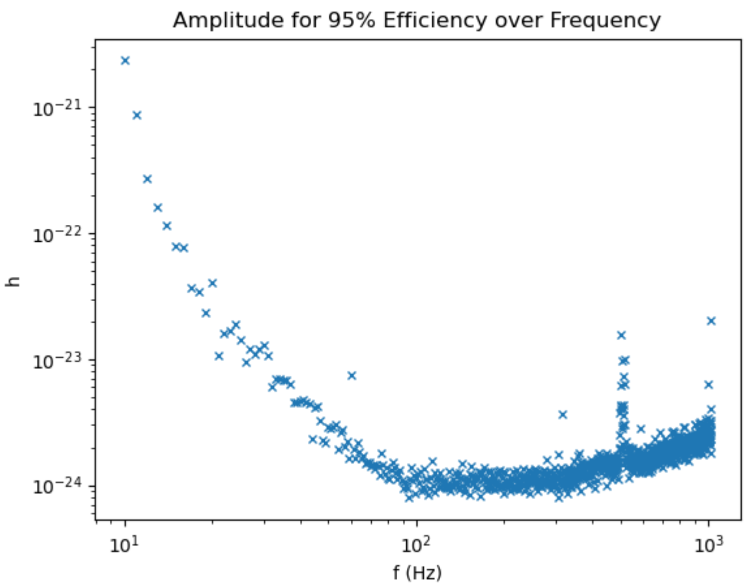

Figure 1: *A model of the background noise based off a bounded numeric solver using injected sinusoidal signal with static frequency.*

The results shown in Figure 1 nearly match the full 95% O2 upper limits of the frequency-Hough pipeline [6, FIG 1] which was also used here to estimate lower and upper bounds to help guarantee a global solution for each frequency. The reason for the reduced sensitivity is that the spin-down was not included, the Hough transform was not used, and the dataset was much smaller. Still, this not only helps validate the method of injection and detection, but also serves as a reference point when measuring the efficiency of parameters within certain frequency ranges.

a. Optimal Parameters

Both the AR and PM process involve input parameters that govern their behavior. For example the PM has a threshold (i.e. a minimum amplitude that must each peak must have in order to be valid), but because this threshold has an analytical solution [2], there is no need to explore any variations thereof. However, the AR - which will be treated as a black box due to its lengthy definition - does have parameters that could be of interest.

The AR parameters are as follows:

Name	Summary	Default
τ	The memory of the auto-regressive mean	0.02 Hz
dead frequency	The minimum frequency difference of two candidates	τ/δ_ν

where δ_ν is the frequency resolution, and the parameters will be their default values unless otherwise specified.

In order to make sure the parameters had sane defaults, the Nelder-Mead method was run to maximize the average efficiency for the frequencies {10, 11, ... , 1024} of a signal injected with $h = 10^{-25}$. Using an initial guess of the default values returned that the optimum was indeed the default values. **This means that on average the existing parameters are at the very least locally optimal.** However, this may not be the case for certain ranges of frequencies, and there may be a better global solution.

b. Inspecting Frequency Ranges

The behavior of smaller ranges of frequency pose an interest, as it allows for finer tuning of parameters when searching for more specific sources of CWs. By going across various values of τ, the average efficiency of 1 injected signal per integer frequency was found - the interquartile range (IQR) is also given as errorbars. The amplitude that was selected was $\bar{h}$ for that range based off of the calculations given in Figure 1. There are a few observations that can be made from Figure 2. Firstly, the default value for τ (0.02Hz) is well within a relatively flat zone for every frequency range. However, even this flat range seems to subtlety increase in every case over large changes in magnitude, and one cannot go much lower than the default value without beginning to see noticeable reductions in efficiency. Another subtle note is that the placement of the averages with consideration to the IQR also reflect the slope and density of points in Figure 1 as to be expected. The default τ is fine for the entire spectrum, but increasing it for other frequencies may help.

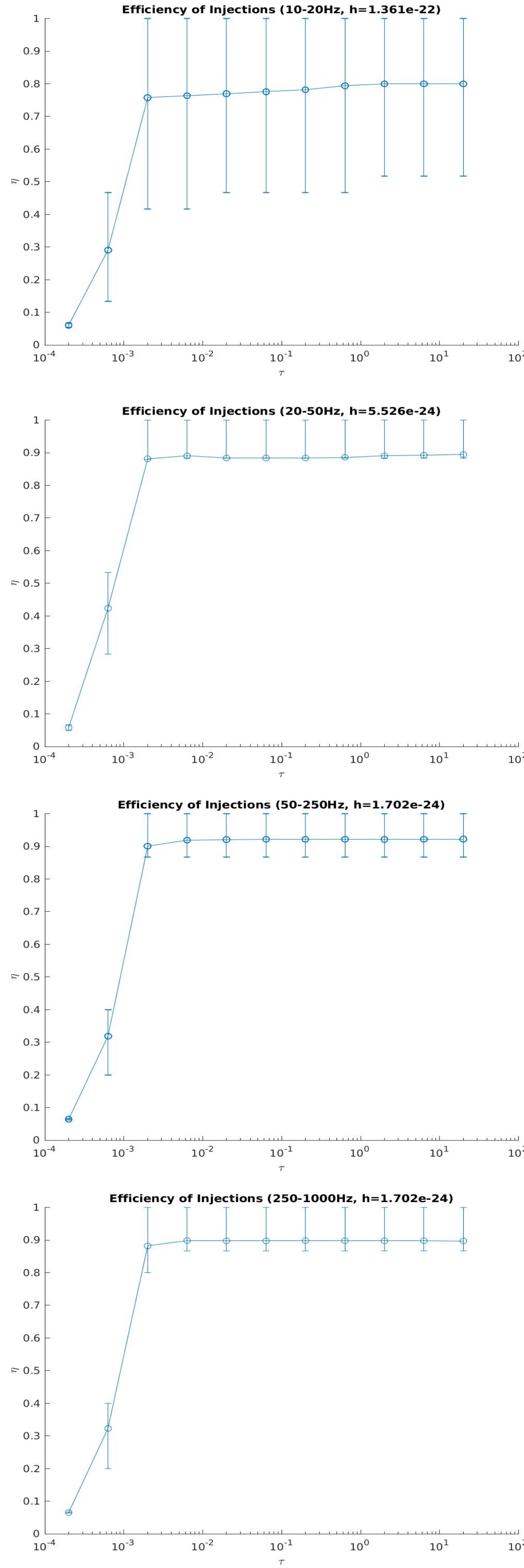

Figure 2: ***Charts of $\bar{\eta}$ and IQR of η over τ.***

c. Introducing Dynamic Signals

So far, every signal that has been injected contained no frequency shift, but the actual expected signals would indeed have dynamic frequency. Thus Doppler shifted signals obtained by random distribution of CW signals over the sky were not only injected but also corrected and detected. There is likelihood that future detectors with improved sensitivity as well as some models of aforementioned boson clouds would have many signals close together. So there is particular interest to verify the behavior of when multiple signals that are injected close together:

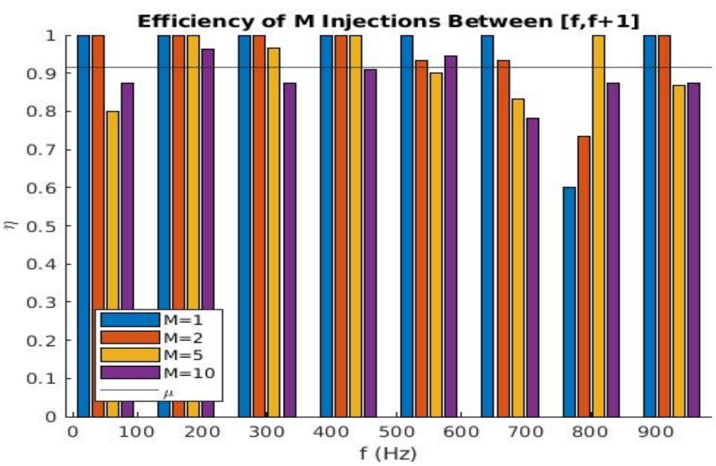

Figure 3: ***Far apart*** *Doppler shifted signals injected and corrected in an integer Hz frequency range with $h \sim 10^{-20}$.*

Yet injecting many signals into a single Hz range is still not a good enough indicator because the frequency resolution of the data used is $\delta_\nu = 4.8828 \cdot 10^{-4} Hz$. Nonetheless Figure 3, gives a baseline of how injected Doppler shifted signals may act. Notice how increasing the number of injected signals brings finer variability, but the range remains unchanged. A much better measure would be injecting signals spaced by a certain number (N) of frequency bins so the distance in frequency domain (Δ_f) between signals would be:

$$\Delta_f = N \cdot \delta_\nu$$

When $N = 0$ the injected signals are not in the same frequency bin until after correction, so correcting and counting each individually still works.

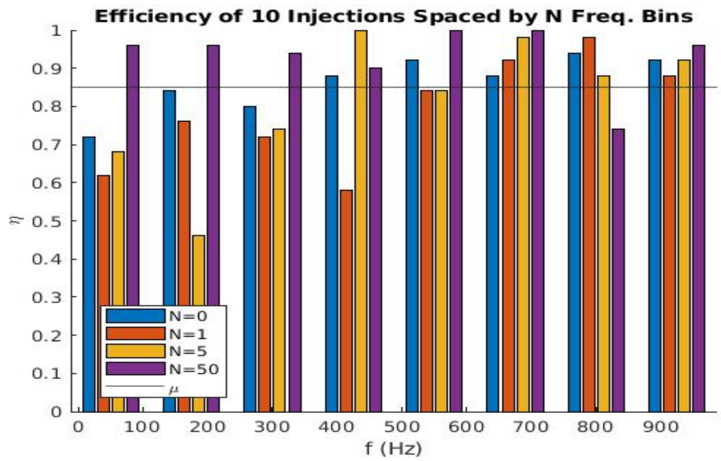

Figure 4: ***Extremely close*** *Doppler shifted signals injected and corrected to every Nth frequency bin with $h \sim 10^{-20}$. $N = 0$ has 10 injected signals as well.*

Even though there is some variance to be expected due to the random variables generated to simulate the Doppler shift, Figure 3 has a slightly higher average than Figure 4, but this difference is within the standard deviation of both. Meaning that for clumped together signals **injections that are far apart and extremely close have similar efficiencies.**

IV. Significance of Results

Most of the time on this project was spent developing the tools that were used. This was a slow and iterative process that often came with roadblocks and failures. Hopefully the methodical process combined with rigorous testing led to a bug free analysis. Yet one should consider that the data used was just a small portion of the O2 run. While the size of the data was large, it is hard to say for certain whether it accurately represents other sections of time, or if the results can be extrapolated for past or future runs.

A problem with Figure 1 is that the numeric results can get stuck on the ends where the efficiency is 1 or near 0 because the function is too flat causing the algorithm to terminate. To solve this a small slope was artificially imposed to aid the numeric solver in finding where the actual non-flat regions were. While this slope was several magnitudes below significant, it is possible that this occasionally misled the solver. Nonetheless, the results were almost exactly what was expected based off of previous work while using a different section of data and different technique. Similarly, when finding the optimal parameters - the solution being equal to the input can easily be a sign that the solver was able to find no better nor worse solution.

For injected Doppler shifted signals, a higher number of injected signals on a larger dataset would be needed to get infallible results. Ultimately, there is a lot more investigation that could be done on the parameters and verification of how the pipeline treats signals, but this should probably be done instead on tightly focused frequency ranges.

a. Failed Results

There were a lot of ideas that initially seemed to be interesting or helpful explorations, but in reality were dead-ends. Hopefully, a short-mention will prevent any reputations of the same mistakes. For example, Figure 1 was attempted to be duplicated using dynamic signals to verify that the process for injecting and correcting led to similar efficiencies. This did not work because Doppler shifts are generated at random, so every iteration produced a unique correction and thus different numbers of false-positives. At no fault of the existing code that was used, this variability ruined the optimization algorithm causing the results be off by orders of magnitude. Perhaps this could be corrected by identifying every random variable and giving it a seed or by averaging several runs every iteration of the optimization. Similarly, Figure 1 was also going to include the same calculation for the locally optimal parameters (which in a way it did). The SNR was also going to be used in lieu of injected signal amplitude, but this was another calculation for already computationally intensive procedures that was hard to tack as an afterthought. In the end, it was decided that using signal amplitude alongside providing Figure 1 would not only prove sufficient, but potentially more helpful in comparing with relevant works.

There were definitely more dead-ends, but the few that were mentioned here led to most amount of time wasted. Even so, every failure often led to better insights on the work yet to be done.

b. Future Results

In order to provide enough information for every interesting frequency range would require page after page of charts and tables - not to mention computationally challenging. Such a massive undertaking characterizing so many regions is one of the possibilities that the software was intended for and would use a much larger dataset than used in this report. For the time being, the library that was provided should be flexible enough that whenever an answer to a question of a certain frequency, parameter, or dataset is needed - it should be simple to use the existing code to quickly get the results needed. There is more work that could be done on analyzing the dynamic signals and AR parameters in general. In terms of data analyzed, this report is just the tip of the iceberg, and with new data often coming it does not seem to be melting. The entire library could always be expanded upon, and the goal was for it to be well written enough that improving and reusing is easier than re-implementing any specific functionality. The vast room for further tests and optimizations will of course only grow the more the software is used.

V. Conclusion

The final product was both a highly flexible codebase as well the verification of the expected behavior of various algorithms. This was the first time that signals were injected close together in the pipeline, something that must be handled correctly to accommodate a large set of possible sources. Many prior assumptions were either correct or more than good enough. The recreation required an in-depth look into past code and significant designing of new code. This new code may be used for future characterization of frequency ranges by parameters. When everything worked, there were few surprises. Meaning, this project has only strengthened the past and future works using the frequency-Hough pipeline.

VI. Acknowledgments

Thank you to the folks at the University of Florida for putting together this international REU, and especially thanks to everyone involved at the University of Rome for being such a wonderful host institution. This project would not have been possible without the amazing funding provided by the National Science Foundation [Grant Numbers: 1460803 & 1950830].

References

[1] P Astone, S Frasca, and C Palomba. The short FFT database and the peak map for the hierarchical search of periodic sources. *Classical and Quantum Gravity*, 22(18):S1197–S1210, September 2005.

[2] Pia Astone. Peak–maps: definitions and statistics.

[3] Abbott et al. GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral. *Phys. Rev. Lett.*, 119(16):161101, October 2017.

[4] Abbott et al. All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO O2 data. *Physical Review D*, 100(2):024004, July 2019.

[5] Sergio Frasca, Cristiano Palomba, Roberto Ruffato, and Ettore Majorana. SNAG, A Toolbox For Gravitational Wave Data Analysis. *International Journal of Modern Physics D*, 09(03):299–302, June 2000. Publisher: World Scientific Publishing Co.

[6] C. Palomba, S. D'Antonio, P. Astone, S. Frasca, G. Intini, I. La Rosa, P. Leaci, S. Mastrogiovanni, A. L. Miller, F. Muciaccia, O. J. Piccinni, L. Rei, and F. Simula. Direct Constraints on the Ultralight Boson Mass from Searches of Continuous Gravitational Waves. *Physical Review Letters*, 123(17):171101, October 2019.