Measuring the Tidal Disruption Time in NSBH Systems

L. Chastain[1], E. Thrane[2], and P. Lasky[2]

[1] *School of Physics and Astronomy, Monash University, Clayton, Victoria 3800, Australia and*

[2] *Centre for Astrophysics, School of Physics and Astronomy, Monash University, VIC 3800, Australia*

(Dated: September 6, 2021)

In the wake of the recent gravitational wave (GW) observations from two neutron star-black hole (NSBH) binary systems, the possibilities to study matter in extreme conditions, including tidal disruption, and the search for potential deviations from general relativity (GR) has broadened. In this work we present an introductory review of gravitational waveform analysis by modeling a binary neutron star (BNS) merger to better understand the physics present in this system. We attempt to simulate tidal disruption by manually obstructing the waveform with a chosen model. We also present signal-to-noise estimates of the simulated disruption to the original waveform.

I. INTRODUCTION

The majority of GW detections with the current ground-based detectors have been black-hole binaries (BBHs). However, LIGO recently detected an NSBH merger, giving rise to new questions and unambiguous data for this elusive system[1]. Although, gravitational wave signals from this event is fairly consistent with BBH and BNS systems. Due to the disruption of the neutron star in systems with one or both objects being neutron stars, there are changes in the GW amplitude at high frequencies, where the detectors have been largely insensitive to the merger and post-merger BNS signal[2]. However, numerical studies of NSBHs have provided insight into different aspects of the merger, investigating how acretion disk, ejecta and jets depend on the mass ratio of the binary, and the spin magnitude and orientation of the BH.[3] As the current detectors are improved and upgraded, we will eventually be able to measure this quiet tidal disruption.

A. Importance of Study

The analysis of gravitational waveforms allows us to learn more about their sources and to estimate their distances and positions in the sky. The importance of studying NSBH systems can be entertained when investigating the neutron star equation of state (EOS)[4,5]. For NSBH-specific contributions, one must ask themselves how important the tidal effects are to the overall model of the waveform for current and future detectors. This will give insight into how distinguishable NSBH-specific effects are from BBH or BNS systems[4].

II. THEORETICAL FRAMEWORK

A. Fundamentals of Gravitational Wave Astronomy

Among the sources of GW signals, binaries are expected to be the most common. Therefore, I present here a few key formulae for back-of-the-envelope calculations used to quickly estimate characteristics. However, the formulae presented are for a binary system in non-relativistic orbits; this is well before merger.

$$h_o = \frac{R_1 R_2}{Dr} \tag{1}$$

where h_o is the amplitude, $R_{1,2}$ are the radii of the objects, and D the distance between them.

$$f_{GW} = 2f_{orb} = \frac{c}{2\pi D}\sqrt{\frac{R_1+R_2}{2D}} \tag{2}$$

where f_{GW} is the gravitational wave frequency, and c the speed of light.

$$\dot{f} = \frac{96}{5}\frac{c^3}{G}\frac{f}{M_c}\left(\frac{G}{c^3}\pi f M_c\right)^{\frac{8}{3}} \tag{3}$$

where $\dot{f}$ is the chirp frequency; this indicated that as the gravitational waves are emitted, they carry energy away from the binary. The gravitational binding energy decreases and the orbital energy increases. The gravitational wave $phase\phi(t)$ evolves in time as

$$\phi(t) = 2\pi\left(ft + \frac{1}{2}\dot{f}t^2\right) + \phi_o \tag{4}$$

where ϕ_o is the initial phase of the binary. A phenomenological form of the waveform is then given by

$$h(t) = h_o cos\phi(t) = h_o cos(2\pi ft + \pi\dot{f}t^2 + \phi_o) \tag{5}$$

B. Theoretical Model of Tidal Disruption Time

To manually simulate the "shut-off" shown in previous models of NSBH mergers, we chose the *tanh* function shown below:

$$tanh\left[\frac{(t-t_o)}{\tau}\right] \tag{6}$$

Where t is the duration, τ is the duration of the neutron star disruption, and t_o is when the neutron star gets pulled apart. The duration is a parameter that we set prior to running the simulation, and t_o and τ are assumed and implemented in the original waveform to isolate the ringdown effect of the tidally disrupted NS.

III. METHODOLOGY

The physical information held within GW signals can be extracted by constructing template waveforms based on theoretical models, which are then compared with the data using a Bayesian Inference framework. To model the BNS waveforms and simulate the effect of tidal disruption, I use the gravitation-wave astronomy Bayesian Inference library Bilby (6). In particular, we utilized the gravitational wave packages, which provided the core functionality for parameter estimation specific to transient gravitational waves. With this, we were able to simulate the outputs of current GW detectors, which produced the waveforms seen in Sec. IV.

Previous literature has found that for low mass ratio systems with more positive spins and/or lower compactness of the NS, the final BH is typically surrounded with massive accretion disks with densities $\geq 10^{12} g/cm^3$ (7). Contrarily, for systems with high mass ratio and low spin priors, there is little to no tidal disruption of the NS before it reaches ISCO and can be swallowed almost completely by the BH. The NS leaves barely any remnants of matter to generate detectable electromagnetic signatures, therefore seeming to behave like a BBH system with almost identical GW signatures (3). Therefore, we chose to simulate a waveform with a low mass ratio in hopes to better achieve the tidal disruption signal.

optimal SNR	17.22
matched filter SNR	18.09-0.50j
mass 1	1.3
mass 2	1.5
chi 1	0.02
chi 2	0.02
luminosity distance	40
theta jn	0.4
psi	2.659
phase	1.3
geocentric time	1126259642.413
ra	1.375
dec	-1.2108
lambda 1	400
lambda 2	450

TABLE I: Injected signal into LIGO-Hanford (H1)

optimal SNR	13.95
matched filter SNR	14.40+0.80j
mass 1	1.3
mass 2	1.5
chi 1	0.02
chi 2	0.02
luminosity distance	40
theta jn	0.4
psi	2.659
phase	1.3
geocentric time	1126259642.413
ra	1.375
dec	-1.2108
lambda 1	400
lambda 2	450

TABLE II: Injected signal into LIGO-Livingston (L1)

optimal SNR	14.78
matched filter SNR	15.16+0.50j
mass 1	1.3
mass 2	1.5
chi 1	0.02
chi 2	0.02
luminosity distance	40
theta jn	0.4
psi	2.659
phase	1.3
geocentric time	1126259642.413
ra	1.375
dec	-1.2108
lambda 1	400
lambda 2	450

TABLE III: Injected signal into Virgo (V1)

IV. RESULTS AND DISCUSSION

Given the time-frame, remote nature, and personal struggles and challenges,I was unable to produce tangible results of the NS tidal disruption nor any implications of the time of disruption. Therefore, I present here the work I was capable of producing. Fig. 1 depicts the original waveform that was produced using Bilby's gravitational waveform generator function with the corresponding parameters given in Table 1, 2, and 3.

The resulting waveforms being plotted are a product of the time domain strain and time array produced when passed through the simulated LIGO-Hanford (H1) interferometer. The code originally produces a waveform with the ringdown at the front of the waveform, and Fig. 1 is the result of a cyclical permutation to shift it to the end;

this can be found in Appendix B.

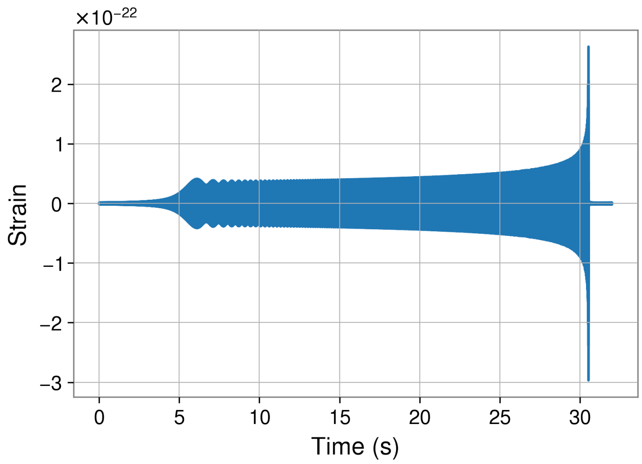

FIG. 1: Original generated waveform with ringdown shifted to the end rather than the beginning. The parameters used to produce this waveform are located in Table 1.

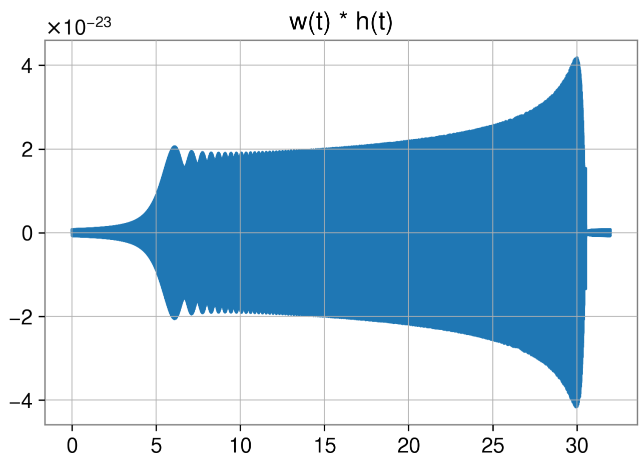

FIG. 2: Extrapolation of tanh model multiplied by the original waveform.

Fig. 2 shows the result of the original waveform, $h(t)$, being multiplied by the *tanh* model, $w(t)$, mentioned in Eq. 6. This was achieved by inputting place-holders for t_o and τ, which were 30.5 and 0.3 respectively. These specific numbers were obtained through a trial-and-error test, and eventually decided on because they produced the best waveform when multiplied by the original. This process was meant to just be an initial step towards measuring a value for these numbers, however that next step was not achieved in the time frame.

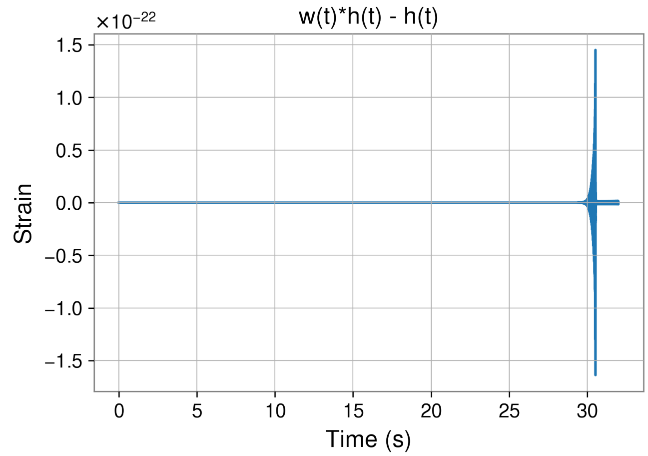

FIG. 3: The difference between the extrapolated waveform and the original waveform, produced to isolate the ringdown.

Fig. 3 is obtained when subtracting the original waveform from the extrapolated waveform. As seen, most of the waveform is cancelled out, leaving only the ringdown. A zoomed-in picture of the end of the waveform can be seen in Fig. 4. The signal-to-noise ratio of this signal to the original is given in *Summary of Results* section with more detail.

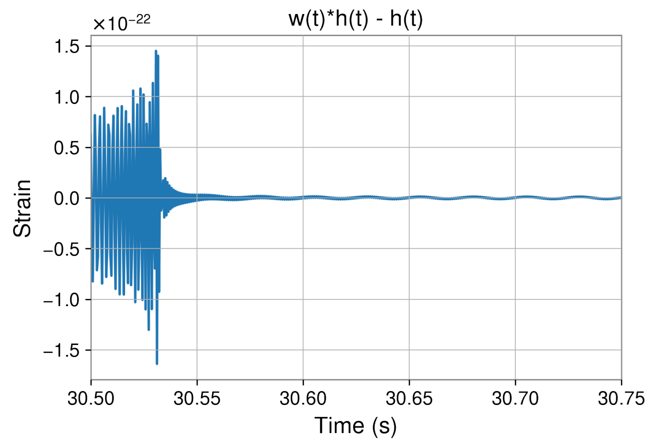

FIG. 4: Same resulting graph as Fig. 3 with different x-limits to show the resulting ringdown achieved.

Summary of Results

We calculate the matched-filter signal-to-noise ratio, S/N. For a strain time series $h(t)$, S/N is

$$S/N = \langle h, n\rangle / \sqrt{\langle u, u\rangle} \tag{7}$$

where $u(t)$ is the template, and

$$\langle a, b\rangle \equiv 4Re \int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_h(f)} df \tag{8}$$

where $S_h(f)$ is the noise power spectral density. Fig. 5 shows the optimal S/N over multiple distances from the detector to the binary source.

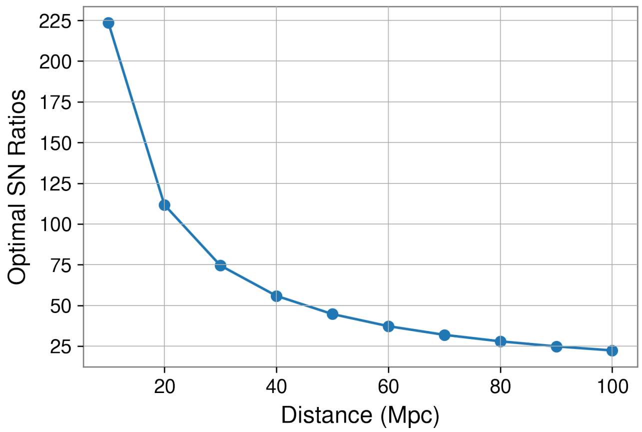

FIG. 5: Optimal S/N over multiple distances. Calculated as the inner product defined in the matched filter statistic.

The figure presented is what we would expect to see, given that $h \sim 1/d$ and $SNR \sim h$, therefore $SNR \sim 1/d$.

V. CONCLUSION

At the beginning of this project, I knew next to nothing about gravitational waves. The loose "*results*" presented here are a small testament to the magnitude of content that I had to grasp over the duration of ten weeks. Most of the graphs, calculations, and coding was an overview of general data reduction and analysis procedures common in gravitational wave astronomy - something I am more familiar with than I was when I began this project. In conclusion, even without the tangible results that were sought after at the beginning, I still ended with a greater understanding of gravitational wave astronomy than I began with.

VI. ACKNOWLEDGMENTS

I would like to express thanks to Eric Thrane and Paul Lasky for their patience and mentorship granted to me during the duration of this project. I thank Colm Talbot for his contribution to the code. I would also like to thank Paul Fulda, Peter Wass, and Guido Mueller of University of Florida for their advisement during the project. I thank the National Science Foundation NSF PHY-1950830 and NSF PHY-1460803 grants for funding the program.

VII. REFERENCES

[1] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, and et al., The Astrophysical Journal Letters **915**, L5 (2021).

[2] J. E. Thompson, E. Fauchon-Jones, S. Khan, E. Nitoglia, F. Pannarale, T. Dietrich, and M. Hannam, Physical Review D **101** (2020), 10.1103/physrevd.101.124059.

[3] B. Khamesra, M. Gracia-Linares, and P. Laguna, Classical and Quantum Gravity **38**, 185008 (2021).

[4] E. Fauchon-Jones, "Full stack development for gravitational waveform modelling," (2020).

[5] F. Foucart, M. Duez, L. Kidder, S. Nissanke, H. Pfeiffer, and M. Scheel, (2019).

[6] G. Ashton, M. Hübner, P. D. Lasky, C. Talbot, K. Ackley, S. Biscoveanu, Q. Chu, A. Divakarla, P. J. Easter, B. Goncharov, and et al., The Astrophysical Journal Supplement Series **241**, 27 (2019).

[7] M. Shibata and K. Taniguchi, "Coalescence of black hole-neutron star binaries," (2011).

Appendix A: Python BNS Waveform Simulation Code

The following code is a tutorial to demonstrate running parameter estimation on a binary neutron star system taking into account tidal deformabilities. This example estimates the masses using a uniform prior in both component masses and also estimates the tidal deformabilities using a uniform prior in both tidal deformabilities.

```
from IPython.core.interactiveshell import
    InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

import numpy as np
from matplotlib import pyplot as plt
import bilby

import json
import os
from math import fmod

from scipy.interpolate import interp1d
from scipy.special import i0e

# Specify the output directory and the name of the
    simulation.
outdir = 'outdir'
label = 'bns_example'
bilby.core.utils.setup_logger(outdir=outdir, label=
    label)
```

```
# Set up a random seed for result reproducibility.
    This is optional!
np.random.seed(88170235)

# We are going to inject a binary neutron star
    waveform. We first establish a
# dictionary of parameters that includes all of the
    different waveform
# parameters, including masses of the two black
    holes (mass_1, mass_2),
# aligned spins of both black holes (chi_1, chi_2),
    etc.
injection_parameters = dict(
    mass_1=1.3, mass_2=1.5, chi_1=0.02, chi_2=0.02,
        luminosity_distance=100.,
    theta_jn=0.4, psi=2.659, phase=1.3, geocent_time
        =1126259642.413,
    ra=1.375, dec=-1.2108, lambda_1=400, lambda_2
        =450)

# Set the duration and sampling frequency of the
    data segment that we're going
# to inject the signal into. For the
# TaylorF2 waveform, we cut the signal close to the
    isco frequency
duration = 32
sampling_frequency = 2 * 1024
start_time = injection_parameters['geocent_time'] +
    2 - duration

# Fixed arguments passed into the source model. The
    analysis starts at 40 Hz.
waveform_arguments = dict(waveform_approximant='
    IMRPhenomPv2_NRTidal',
                          reference_frequency=50.,
                              minimum_frequency=40.0)

# Create the waveform_generator using a LAL Binary
    Neutron Star source function
waveform_generator = bilby.gw.WaveformGenerator(
    duration=duration, sampling_frequency=
        sampling_frequency,
    frequency_domain_source_model=bilby.gw.source.
        lal_binary_neutron_star,
    parameter_conversion=bilby.gw.conversion.
        convert_to_lal_binary_neutron_star_parameters
        ,
    waveform_arguments=waveform_arguments)

# Set up interferometers. In this case we'll use
    three interferometers
# (LIGO-Hanford (H1), LIGO-Livingston (L1), and
    Virgo (V1)).
# These default to their design sensitivity and
    start at 40 Hz.
interferometers = bilby.gw.detector.
    InterferometerList(['H1', 'L1', 'V1'])
for interferometer in interferometers:
    interferometer.minimum_frequency = 40
interferometers.
    set_strain_data_from_power_spectral_densities(
    sampling_frequency=sampling_frequency, duration=
        duration,
    start_time=start_time)
interferometers.inject_signal(parameters=
    injection_parameters,
                              waveform_generator=
                                  waveform_generator)
```

Appendix B: Python Waveform Plotting

The following code plots the original waveform, shifts it so the ringdown will be at the end rather than the beginning, and manually applies the chosen model that reduces the waveform to isolate the ringdown.

```
# Plot original waveform

plt.plot(waveform_generator.time_array,
    waveform_generator.time_domain_strain()['plus'
    ])

# Perform and cyclical permutation to shift the
    ringdown to the end of the graph

strain = waveform_generator.time_domain_strain()['
    plus']
shifted_strain = np.roll(strain,-3000)

plt.plot(waveform_generator.time_array,
    shifted_strain)
plt.xlabel('Time (s)')
plt.ylabel('Strain')
plt.title('h(t)')

# Create a window function using tanh and find the
    values that will isolate the ringdown

tanh_model = np.tanh((30.5 - waveform_generator.
    time_array)/0.3)
plt.plot(waveform_generator.time_array,tanh_model*
    shifted_strain)
plt.title('w(t) * h(t)')

# Take the difference of the original waveform and
    the tanh function waveform then plot

diff = shifted_strain - (tanh_model*shifted_strain)
plt.plot(waveform_generator.time_array,diff)
plt.xlabel('Time (s)')
plt.ylabel('Strain')
plt.title('w(t)*h(t) - h(t)')
#plt.xlim(30.5,30.75)
```

Appendix C: Python S/N Estimation

The following code performs a single-sided fast Fourier Transformation of the time domain strain, as well as calculates the inner product as defined by the match filter statistic and plots the results.

```
def nfft(time_domain_strain, sampling_frequency):
    """Returns
    =======
    frequency_domain_strain, frequency_array: (
        array_like, array_like)
        Single-sided FFT of time domain strain
            normalised to units of
        strain / Hz, and the associated
            frequency_array.

    """
    frequency_domain_strain = np.fft.rfft(
        time_domain_strain)
    frequency_domain_strain /= sampling_frequency

    frequency_array = np.linspace(
        0, sampling_frequency / 2, len(
            frequency_domain_strain))

    return frequency_domain_strain, frequency_array

frequency_domain_strain = nfft(shifted_strain,
    sampling_frequency)[0]
frequency_array = nfft(shifted_strain,
    sampling_frequency)[1]
diff_domain_strain = nfft(diff,sampling_frequency)
    [0]

def inner_product(aa, bb, frequency, PSD):
    """
    Calculate the inner product defined in the
        matched filter statistic

    Parameters
    ==========
    aa, bb: array_like
        Single-sided Fourier transform, created, e.g
            ., by the nfft function above
    frequency: array_like
        An array of frequencies associated with aa,
            bb, also returned by nfft
    PSD: bilby.gw.detector.PowerSpectralDensity

    Returns
    =======
    The matched filter inner product for aa and bb

    """
    psd_interp = PSD.
        power_spectral_density_interpolated(
        frequency)

    # calculate the inner product
    integrand = np.conj(aa) * bb / psd_interp

    df = frequency[1] - frequency[0]
    integral = np.sum(integrand) * df
    return 4. * np.real(integral)

np.sqrt(inner_product(frequency_domain_strain,
    diff_domain_strain,frequency_array,
    interferometer.power_spectral_density))

distances = [10,20,30,40,50,60,70,80,90,100]
optimal_sn_ratio =
    [223.501,111.751,74.500,55.875,44.700,37.250,31.929,27.938,


plt.plot(distances,optimal_sn_ratio, 'o-')
plt.xlabel('Distance (parsecs)')
plt.ylabel('Optimal SN Ratios')
```