

LIGO's Ability of Detecting Low Mass Gravitational Wave Sources

Jingyi Zhang

Mentor: Prof. Eric Myers

Pioneer Summer Project

Abstract: The goal of this project was to investigate the condition needed for LIGO and future gravitational wave detector like eLISA to detect low mass gravitational wave sources. I modified the template provided by LIGO open science center and added the template to the real noise from LIGO data to create a software injection. The result suggests that present LIGO cannot detect signals from low mass gravitational wave source, yet future eLISA would be more suitable for this job.

Keywords: Gravitational waves; LIGO; eLISA; Binary system

1. Introduction

1.1 Gravitational Waves

Gravitational waves are disturbance in the curvature of spacetime which travel outward from their sources as waves; they can also be described as oscillation of the gravitation field varies with time. Gravitational waves travel at the speed of light and are able to transport energy as the form of gravitational radiation.

In Einstein's general theory of relativity, gravity is the result of curved spacetime; the curvature of space time is cause by mass. The curvature change to reflect the motion and the acceleration of objects, and, in certain circumstances, generate gravitational waves. The larger mass and acceleration would cause greater curvature and curvature changes, or to say, more detectable gravitational wave signals. For the above reason, black holes, neutron stars, supernovas, and even the Big Bang are all good sources for gravitational wave observation as they all involve changes of large masses. [1]

In general, the gravitational waves are generated by objects whose motion involves acceleration and its changes, provided that the motion is not perfectly rotationally symmetric. This means that almost every object would radiate gravitational waves. However, in most of the cases, the wave sources' mass and acceleration are too small to generate detectable gravitational signal that is detectable for current gravitational detectors. Only a few

natural sources are powerful enough to generate detectable gravitational wave signal. Today, scientists classify gravitational wave sources in four categories: continuous gravitational waves, compact binary inspiral gravitational waves, stochastic gravitational waves, and burst gravitational waves. So far, all gravitational waves events detected by LIGO are Compact binary inspiral gravitational waves created by either black hole or neutron star binaries.

Compact binary inspiral gravitational waves are produced by orbiting pairs of massive and dense objects like black holes and neutron stars. When the binaries orbit around each other, they send off gravitational waves and lose their orbital energy. As the system loses more orbital energy, the two objects become closer, orbiting at a shorter period, and eventually send off gravitational waves of higher frequency and greater power. So far, all gravitational waves events detected by LIGO are Compact binary inspiral gravitational waves created by either black hole or neutron star binaries. [2]

In this project, we are considering low mass and low density compact binaries as gravitational wave sources, especially close white dwarf binaries. We would like to see whether current gravitational wave detectors is able to detect these sources.

1.2 White Dwarf Binaries

A white dwarf, also known as a degenerate dwarf, is a low mass, small volume star that has a comparable high density than that of main sequence star. White dwarfs are composed mostly

of electron-degenerate matter and are considered as the final evolutionary state of main sequence stars that do have enough mass to become neutron stars or black holes, which consist 97% of the stars in the Milky Way, including the sun. [3]

The average mass of white dwarfs is around 0.5 to 0.6 solar mass. The maximum mass of a non-rotating white dwarf is 1.44 solar mass, also known as the Chandrasekhar limit. In some cases, white dwarfs in binaries may accrete mass from their companion and their mass increase. When the white dwarfs reach the Chandrasekhar limit, the electron degenerate pressure would not be able to support them, and they may explode as type IA supernovae.

Type IA supernovae can serve as standard candles which help to measure the distance between stars and galaxies. Unfortunately, the exact process of generating type IA supernovae still remains unclear, making it difficult to predict and observe these supernovae events.

Through analysis of the gravitational wave signal generated by white dwarf binaries, we may help to find what are exactly the progenitors of type IA supernovae and explain the exact evolution process of the progenitors. [4][5]

However, detecting gravitational wave signals from white dwarf binaries can be very difficult.

As mentioned before, a typical white dwarf's mass is around 0.5 to 0.6 M_{\odot} , and no white dwarf can be heavier than 1.44 M_{\odot} . [6] The low mass would create weak signals that may not be detectable by current gravitational detectors. Moreover, though the density of white dwarfs is much higher than any main sequence star, it is still much lower than the neutron

stars or black holes that LIGO have detected. The large volume may cause the binaries to merge before they rotate at a noticeable frequency, make the signal even harder to be detected. In later paragraph, I would discuss whether LIGO can detect signal from a merging binary system, and also whether eLISA, a future spaced-based gravitational detector that its designed working frequency is much lower than LIGO, can detect gravitational signal generated from white dwarf binaries. [7]

1.3 The Laser Interferometer Gravitational Wave Observatory

The laser interferometer gravitational wave observatory, also known as LIGO, is a Caltech-MIT project funded by NSF. So far, LIGO has been through two updates, six science runs, and two observation runs, during which it made 6 definite detections of gravitational waves. [8]

segtime*fs

LIGO utilizes strong lasers to make interference and detect small changes of the space. When a gravitational wave transient passes through the interferometer, it would stretch one arm and compress the other arm of a LIGO detector. The phase of the laser arrive at the photodetector would be slightly different from the initial state because of the length change. The phase difference between two lasers would result in a change of the intensity of the combined light, which enable the scientists to calculate the length change of the arm.

Today, the advanced LIGO is able to detect the length difference less than 10^{-18} meter. Make it

the most sensitive gravitational detector in the world and also the only one that has ever detected gravitational wave signal.

1.4 Laser Interferometer Space Antenna

Laser Interferometer Space Antenna, also known as LISA, is a European Space Agency mission consisting of three satellites that designed to accurately detect low frequency gravitational wave. The space craft is planned to launch in 2034. eLISA detectors use the same method as LIGO, but in much larger scale——instead of two 4 kilometer arm, eLISA's arms are 2.5million kilometers long. The longer arms and the quiet space environment would enable eLISA to make more precise detections. [9]

2 Method

LIGO Open Science Center (LOSC) has release data from S5 to O1and also python scripts that can be used to analysis them. I modified the scripts from LOSC, generate new gravitational wave templates of low mass object and make software injections to simulate the observe process.

2.1 Generate new templates

A template is an artificially made signal model that can simulate real gravitational events, by

adding the template model to the known background noise of LIGO and analyze the final signal, we can test whether it is possible that LIGO can hear these signal.

2.1.1 Mass

The template provided by LOSC simulates the gravitational wave signal of black hole binary merging events. The default setting simulates two $10M_{\odot}$ black holes that is one Mpc away from us. [10] Since I want to found out the whether LIGO can detect low mass object, I use two values as typical white dwarf's mass to make the template: $0.5M_{\odot}$, the average White dwarf' mass; and $1M_{\odot}$, the mass of nearest known white dwarf, Sirius B. In each template, I assume that two stars in the binary is identical to make easier calculation and to avoid accretion events between stars that may involve mass change.

2.1.2 Duration

The chirp of the event needs to be included and recognized in the template to show a complete merging event. For lower mass object, more time is needed to make a noticeable chirp as that acceleration of these object is lower than those of the heavy black holes. In the original "find a hard ware injection" script provided by LOSC, the template is 136 seconds long. To see recognizable chirps, the duration I set for the $0.5M_{\odot}$ template is 400 seconds and for the $1.0M_{\odot}$ template is 200 seconds.

2.1.3 Frequency range

So far, we don't exactly know about what would happen after two objects in the binaries

contact. Thus, we are only focusing on the signal generated before any contact happened. This means that we need to set the maximum frequency that a binary can have to stop the template once the binaries contact. The original template uses the innermost stable circular orbit of black hole to calculate the maximum frequency. This may not be suitable for a low mass object because the density of the star would change dramatically below 1.44 solar mass. The maximum frequency should be much lower as the volume increase and the mass decrease. However, the template performs very differently when the frequency changed, which highly affect the final result of the injection. Later in the passage, I would discuss about the influence of mass of template performance. In the injection I made, I still assume that the low mass object follows the same rule as normal black holes and still use the innermost stable circular orbit to limit the frequency range.

2.1.4 Time domain template

The original template is used for making hardware injections. The script would generate frequency domain template which is enough for finding injections. As I need to make new injections, I need to change the template to time domain to add the template to the noise strain in time domain. I use inverse Fourier Transfer to change the original template to time domain one.

2.1.5 Distance

The default setting in the template suggests that the merging event is 1Mpc away from us, which is a reasonable distance for black hole binaries. However, the low mass binaries we are

searching for exist much closer to us. Instead of *megaparsec*, the reasonable distance of white dwarf binaries is on the scale of *parsec*. The nearest known white dwarf Sirius B is 2.6 pc away from us. The strength of signal is inversely proportional to the distance. To adjust this distance change we need to multiply a reasonable factor to the original strain amplitude. I assume that both binaries are 5 pc away from us and multiply their strain amplitude by a factor of 200,000.

2.1.6 Other modification

The find that the beginning of the template has an abnormal high amplitude part which is not consist with the real events. This abnormality is more obvious as the mass decrease. To avoid this unnecessary high amplitude part, I clear out the first 3 seconds of the data at the beginning.

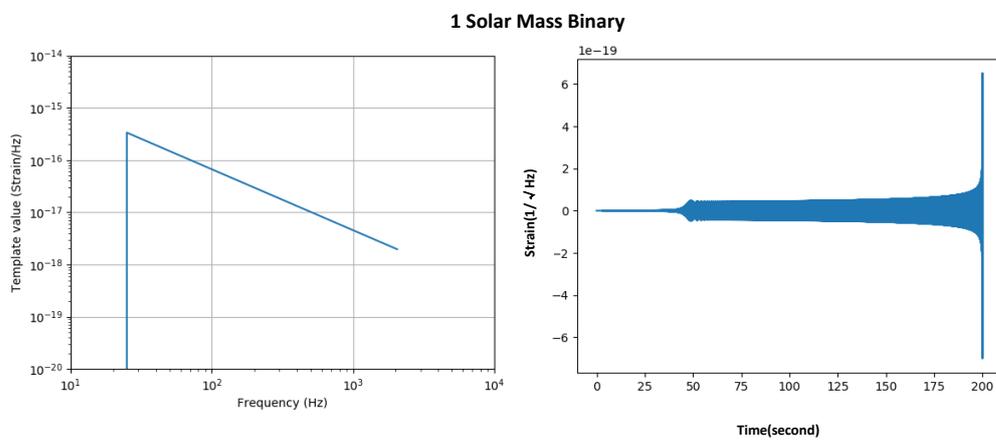
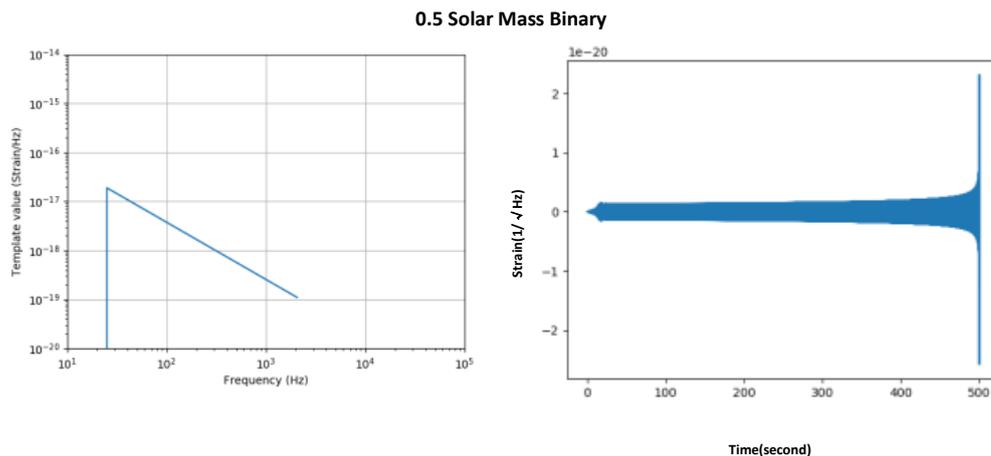


Figure 1. Time & Frequency Domain Gravitational wave signal Template of Low Mass Binaries



2.2 Make a software injection

2.2.1 Get the noise background

In order to make a software injection, we need to put the template in real LOGO noise and to see whether we can find them with the scripts provided by LOSC. LOSC has released the strain data from S5 to O1 and also the list of injection and events. We need to find a piece of data which does not contain any injection or event to make sure that the only thing can be detected is the template we put in. I chose the data in S5 beginning at GPS time 830353408 at Hanford Observatory.

2.2.2 Make a software injection

Making a software injection is very easy. Simply write the template in to hdf5 format and add the template to the noise background. (Appendix B)

2.2.3 Find the software injection

LOSC provided public two useful scripts that may help with this project: *find a hardware injection* and *find an inspiral*. [12] I use the find an inspiral script which does not require any data quality channel to test whether the template I put in the background is detectable.

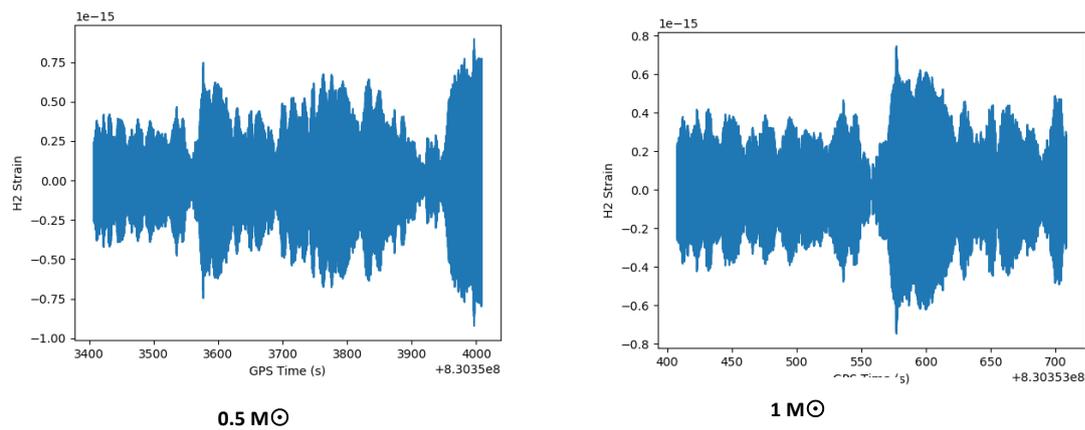


Figure 2. Software Injection Data

(The figures above only present the part which contains the gravitational wave signal template)

3. Result and Discussion

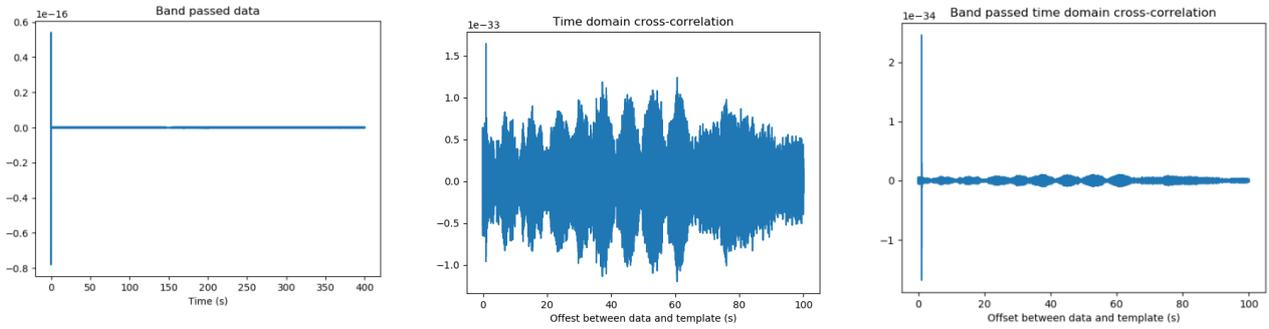
3.1 Results

No obvious detection can be seen from the time domain cross-correlation or the band pass time domain cross-correlation analysis. For both binaries, the band pass cross correlation analysis shows a short fit at the beginning, but as the events I set last 200 and 500 seconds, this cannot be count as a valid detection. I suspect this is because I have cleared out the first 3 second of each template to avoid strange jumps. [figure 3]

Although the signal is very loud as the low mass binaries are very close to us, we still cannot make any detection. I think there are two reasons: First, the frequency of the binaries is too low that they are very far from LIGO's sweet point. Louder environment noise may exist in lower frequency range; make it harder for the detector to distinguish between signal and noise. Second, the merging event of low mass objects takes too long time; the chirp is not obvious

enough as the signal intensity increase slowly with time.

0.5 M \odot



1 M \odot

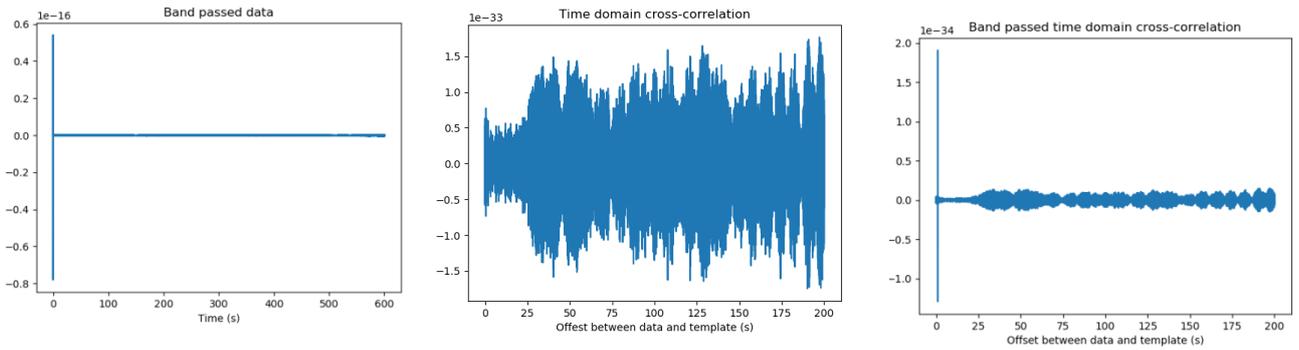


Figure 3. Analysis of the soft ware injections.

3.2 Frequency of the low mass object

As mentioned in 2.1.3, the template provided by LOSC uses the innermost stable circular orbit of black holes to restrain the maximum frequency of the binaries. However, as we are discussing low mass object such as white dwarf stars, this rule could not be applied on them. Since the density of white dwarfs is much lower than black holes or neutron stars, the radius of the star is bigger, and the maximum frequency of white dwarf binaries should be much smaller than those of black holes or neutron stars.

It is reasonable to assume that the white dwarf binary would reach its maximum frequency when they make contact. We can roughly estimate the maximum frequency of white dwarf binaries by their mass. In article *Estimating Stellar Parameters from Energy Equipartition* [12], it gives an easy estimation of the white dwarf's radius. The article uses an energy minimization argument to calculate the relationship between the mass and radius of white dwarfs. The final result suggests that the radius of a white dwarf is inversely proportional to the cube root of its mass, and the formula is:

$$R \approx \frac{N^{5/3} \hbar^2}{2mGM^{1/3}}.$$

Where R is the radius of the white dwarf, N is the number of electrons per unit mass, \hbar is the reduced Planck constant, m is the electron mass, G is the gravitational constant, and M is the mass of the white dwarf.

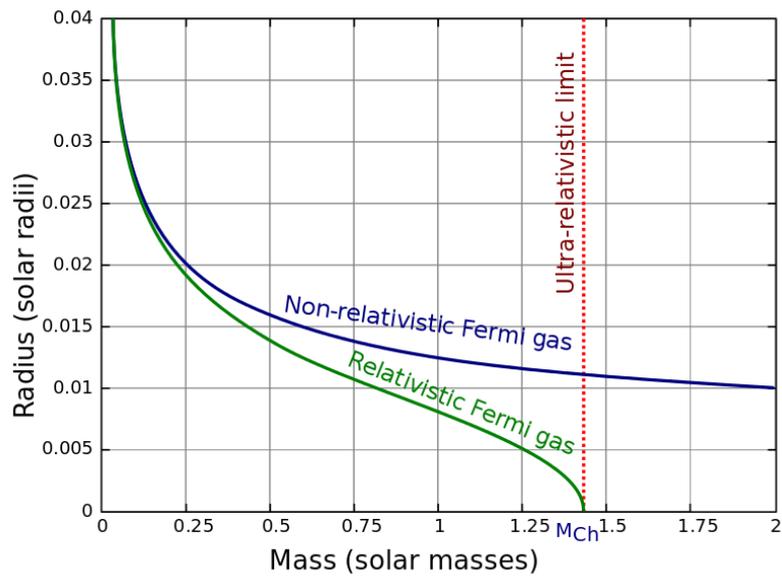
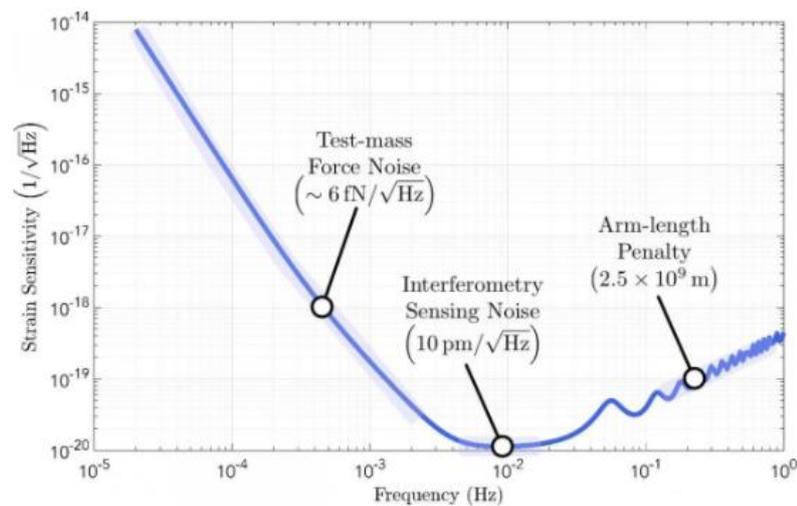


Figure 4. White Dwarf's Mass and Radius Relationship[11]

We assume that the gravitational force between two stars and their own centripetal force is roughly the same. (I do not take in acceleration of the star into account; otherwise the calculation would be too complicated.) By knowing the mass and the radius of the white dwarfs, we can estimate the velocity of their circular motion and their frequency. The final result of my estimation suggest that the maximum frequency of white dwarf is on the scale of 1Hz, typically around 2 to 3 Hz, much lower than we used in the template.

Basically, LIGO's minimum working frequency is around 25Hz. Any lower frequency would be undistinguishable to LIGO as the noise in this range would be huge. In this case, the low frequency of real white dwarf binaries made them impossible to be detected by LIGO, no matter how close they are. Scientist just simply could not make valid conclusion about lower frequency signal as they are highly influenced by environmental noises.

However, this does not mean that the gravitational wave generated by white dwarf binaries would remain unclear. Future eLISA detector is made for these low mass objects. The designed sweet point of eLISA is around 10^{-2} Hz, exactly meet the possible frequency of white dwarf binaries. At this sweet point, eLISA can reach its highest sensitivity, detecting strain signal as small as 10^{-20} . Though this sensitivity is not as good as LIGO due to its extremely long arm, it would be enough to detect smaller stars as they are much closer to us



and more common to see than black holes or neutron stars.

Figure 5. Sensitivity of eLISA[13]

3.3 About the template script

The script I used to generate the template of low mass objects' gravitational wave signals still has some limits, and I have not completely understand them. I list the question below in case any reader wants to make improvements.

3.3.1 Suitable frequency range

When low frequency motion is taken into consideration, the template is no longer usable. Instead of showing a chirp, the template would perform a very random pattern which doesn't give any valuable information. Even lengthening the time of the whole events would not help. In this script, there only exists a simple linear relationship between frequencies and their corresponding intensity, I suspect that the low yet powerful strain would be very noticeable in the whole template, creating weird patterns that cannot be understood. In further investigation, I think instead of using a linear relationship, we may try to make modifications to avoid unreasonable powerful low frequency signal.

3.3.2 The beginning.

As I mentioned before, the beginning part of the template has a weird "jump." I am not sure about what exactly makes the jump. In my calculation, I simply clear out the data of the first 3 seconds but I believe that a better solution exists.

3.3.3 The end

The end of the template also has a weird part. Instead of constantly growing stronger and stronger and suddenly stop because of the maximum frequency we set, the template often has an unstable ending where the intensity sometimes decreases. This phenomenon is especially obvious when the template applies to low mass objects.

3.3.4 Different template

The *find an inspiral* script provided publicly a different time domain template that already been

written into hdf5 format. I found that the template made by the template script is quite different from the already written template. The hdf5 template seems much better- made as it shows a very smooth increasing curve which does not contain any unreasonable jump or gap. I think this template would be helpful to improve the existing script. To get better comparison, I think we can use Fourier transform to make the hdf5 template into frequency domain and see the difference between two scripts.

3.5 Further investigation

So far, scientists still don't exactly know what would happen after two white dwarf stars merge. Since white dwarf stars are very faint, it is very hard to observe them using optical method. The detection of gravitational wave signal would give us more clues about the merging process, contribute to the study supernovae and star evolution.

4 Conclusions

The present LIGO detector can hardly hear from low mass objects. It is impossible for LIGO to detect white dwarf binaries because of their low frequency. Future eLISA would be much more suitable for low mass gravitational wave source detection as these objects are loud enough but in low frequency.

- [1] Krauss, LM; Dodelson, S; Meyer, S (2010). "Primordial Gravitational Waves and Cosmology". *Science*. 328 (5981): 989–992. arXiv:1004.2504
- [2]"Sources and Types of Gravitational Waves." LIGO Lab | Caltech. LIGO Caltech, n.d. Web. 26 July 2017. <https://www.ligo.caltech.edu/page/gw-sources>.
- [3]Johnson, J. (2007). "Extreme Stars: White Dwarfs & Neutron Stars". Lecture notes, *Astronomy 162*. Ohio State University. Retrieved 17 October 2011.
- [4]S.E.Woosley, Thomas A.W., Sub-chandrasekhar mass models for type Ia Supernova, *The astrophysical journal*,1994.3.1:371-379
- [5]Bo Wang, Zhanwen Han, Progenitors of type Ia supernovae, 2012.6,*New Astronomy Reviews*, Volume 56, Issue 4, p. 122-141.
- [6]Barstow, M. A.; Bond, Howard E.; Holberg, J. B.; Burleigh, M. R.; Hubeny, I.; Koester, D. (2005). "Hubble Space Telescope spectroscopy of the Balmer lines in Sirius B". *Monthly Notices of the Royal Astronomical Society*. 362 (4): 1134–1142. arXiv:astro-ph/0506600 .
- [7] Andrzej Krolak, " White Dwarf binaries as gravitational wave sources " in: *Einstein Online* **Vol. 04** (2010),1001
- [8]Cervantes-Cota, J.L.; Galindo-Uribarri, S.; Smoot, G.F. (2016). "A Brief History of Gravitational Waves". *Universe*. 2 (3): 22. arXiv:1609.09400.
- [9] eLISA website.
<https://www.lisamission.org/articles/lisa-mission/first-gravitational-wave-observatory-space/sensitivity>. Sep 2018
- [10] Allen et. al. 2011. *PHYSICAL REVIEW D* 85, 122006 (2012). FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries.

arXiv:gr-qc/0509116

[11] <https://www.gw-openscience.org/inj/step1/>. 2018.11

[12] Estimating Stellar Parameters from Energy Equipartition.

<http://www.sciencebits.com/StellarEquipartition>. 2018.9

[13] AllenMcC.

https://en.wikipedia.org/wiki/White_dwarf#/media/File:ChandrasekharLimitGraph.svg

2018.9

[14] <https://www.elisascience.org/multimedia/image/lisa-sensitivity>. 2018.9

Appendix A

```
# Generate a template for detector strain
# as a function of frequency using eq. 3.4
# in Allen et. al. 2011 (title: FINDCHIRP)
# arXiv:gr-qc/0509116
#
# Adopted from a module
# by Ashley Disbrow 2013
import cmath
import numpy as np
import math
#import matplotlib
#matplotlib.use('Agg')
#import matplotlib.pyplot as plt
#####
#Define all functions which generate template
#####
#Calculate strain using frequency domain
def h(f, Amp, Psi, D_eff,imaginary):
h_f = (1.0/D_eff)*Amp*f**(-7./6)*np.exp(-imaginary*Psi)
return h_f
#Amplitude equation 3.4b
def Amp(ang_mom):
G=6.67e-11 #Units: m^3/(kg*s^2)
c=3e8 #m/s
Mpc = 3e22 #m
m_sun = 2e30 #kg
c1=-math.sqrt(5./(24*math.pi)) #first term eq. 3.4b
c2=(G*m_sun/(c**2*Mpc)) #unitless
c3=(math.pi*G*m_sun/c**3)**(-1./6) #(T)**-(1/6)
c4=ang_mom**(5./6) #unitless
return c1*c2*c3*c4 #with units of (time)**-(1/6)
#eq. 3.4c
def Psi(f,eta,Mtot):
G=6.67e-11*2e30 #Units: m^3/(M_sun*s^2)
c=3e8
v = (G*Mtot*math.pi*f/c**3)**(1./3) #unitless
t1=(3715./756+55.*eta/9)
t2=15293365./508032+27145.*eta/504+3085.*eta**2/72
t_0 = 0 #assume time is negative until coalescence
phi_0 = 0 #equal to Phi_c, assuming i=1 and F_cross=0
Psi = 2*math.pi*f**t_0-2*phi_0-math.pi/4+(3./
(128*eta))*(v**(-5)+t1*v**(-3))-16*math.pi*v**(-2)+t2/v)
```

```

return Psi
#####
#Main - Give parameters and generate template
#####
def createTemplate(fs,dataChunk, m1, m2):
#Definitions necessary for math
Deff = 1. #Mpc
j = cmath.sqrt(-1) #define the imaginary variable
dt=dataChunk/2
Mtot=m1+m2
eta = float(m1*m2)/(Mtot**2)
ang_mom=eta**(3./5)*Mtot
# Create array of frequencies
nyquist = fs/2
f_i = 1./(2*dt)
# f_i = 25
frequency = np.arange(0,nyquist+1./(2*dt),1./(2*dt))
frequency[0] = 1./(2*dt)
#use frequencies to find strain:
amplitude = Amp(ang_mom)
psi_vector = Psi(frequency,eta,Mtot)
strain = h(frequency,amplitude,psi_vector,Deff,j)
# The template should stop at f_isco, the innermost
# stable circular orbit.
c = 3e8
G = 6.67e-11
m_sun = 2e30 #kg
f_isco = c**3/(6*math.sqrt(6)*math.pi*G*m_sun*Mtot)
#strain[f_isco<frequency]=0
strain[2.11949<frequency]=0

return strain,frequency

```

Appendix B

```
#Jingyi Zhang
#Pioneer Program
#2018.9
#This script is a adaptation version, the original script comes from
#Ashley Disbrow and Prof. Eric Myers
#This script add the gravitational wave signal template of a low mass
#binary to real LIGO background
import h5py # for HDF5 files. No need for readligo.py
import numpy as np # h5py uses numpy arrays
import sys # To use sys.modules to detect pylab
import matplotlib.pyplot as plt
import template
#create the template
seg_time=200
fs=4096
dis=200000 #times closer than 1 Mpc
#create the strain
temp, temp_freq = template.createTemplate(fs,seg_time,1,1)
#clear out the low frequency part that out of LIGO range
temp[temp_freq < 25] = 0
#ploting the original template
plt.figure()
plt.loglog(temp_freq, dis*abs(temp))
plt.axis([10, 100000, 1e-20, 1e-14])
plt.title("Template in time domain")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Strain (?/hz)")
plt.grid()
plt.show()
#change the template to time domain
temp_time=np.fft.irfft(temp,n=None,axis=-1,norm=None)
print("The type of the template is",type(temp_time))
# change the distance
temp_large=[]
a=0
while a<seg_time*fs:
b=temp_time[a]*200000
temp_large.append(b)
a=a+1
print(len(temp_large))
#create the time sequence
time=[]
```

```

i=0
while i<fs*seg_time:
time.append(i/fs)
i=i+1
#modify the beginning
c=0
while c<3*fs:
temp_large[c]=0
c=c+1
#plotting the time domian template
plt.figure()
plt.plot(time,temp_large)
plt.title("Template in time domain")
plt.xlabel("time (s)")
plt.ylabel("strain")
##
plt.show()
import pylab as plt # just to plot the results
#-----
# Open the data file using 'r+' so that we can make changes
#-----
fileName = 'H-H2_LOSC_4_V1-830353408-4096.hdf5'
dataFile = h5py.File(fileName, 'r+')
# This is the "dataset" of strain values. It's a part of the HDF5
file.
strainDataset = dataFile['strain']['Strain']
print (type(strainDataset))
# This is a NumPy ndarray of the strain value, extracted from the
file.
strain = strainDataset.value
print (type(strain) )
#-----
# Zero out some data
#-----
startN = 4096 # start at 1 second
p = startN
q=0
while p < (startN + 4096*seg_time) : # end 1 second later
strainDataset[p] = strainDataset[p]+temp_large[q] #
must use strainDataset, not strain!
p = p + 1
q=q+1
# This will write changes to the HDF5 file.
# But don't close the file yet, or you cannot access the dataset

```

```
object.  
dataFile.flush()  
#-----  
# Optionally show the result, but only if pylab is loaded.  
# This part is based on plot_strain.py from losc.ligo.org  
if 'pylab' in sys.modules:  
# Get metadata about the file  
meta = dataFile['meta']  
gpsStart = meta['GPSstart'].value  
duration = meta['Duration'].value  
gpsEnd = gpsStart + duration  
detector = meta['Detector'].value  
# Create a time vector  
ts = dataFile['strain']['Strain'].attrs['Xspacing']  
time = np.arange(gpsStart, gpsEnd, ts)  
# Plot the time series  
numSamples = fs*300  
plt.plot(time[0:numSamples], strainDataset[0:numSamples])  
plt.xlabel('GPS Time (s)')  
plt.ylabel( str(detector,"utf-8") +' Strain')  
plt.show()  
# Close the data file  
dataFile.close()
```

Appendix C

```
#Jingyi Zhang
#Pioneer Program
#2018.9
#This code write the time domain gravitattional wave signal
#template of a low mass binary into hdf5 file
import h5py
import numpy as np
import template
seg_time=200
fs=4096
dis=20000 #times closer than 1 Mpc
#create the strain
temp, temp_freq = template.createTemplate(fs,seg_time,1,1)
#clear out the low frequency part that out of LIGO range
temp[temp_freq < 25] = 0
#change the template to time domain
temp_time=np.fft.irfft(temp,n=None,axis=-1,norm=None)
print("The type of the template is",type(temp_time))
# change the distance
temp_large=[]
a=0
while a<seg_time*fs:
b=temp_time[a]*200000
temp_large.append(b)
a=a+1
#modify the beginning
c=0
while c<3*fs:
temp_large[c]=0
c=c+1
#write in hdf5 file
#create a file
file=h5py.File('1mass.hdf5','w')
#create group and dataset
group=file.create_group('strain')
dataset=group.create_dataset('Strain',(segtime*fs,),h5py.h5t.IEEE_F32BE)
file.close()
print("created")
#write the data into the file
file=h5py.File('1mass.hdf5','r+')
dataset=file['/strain/Strain']
data=[]
```

```
i=0
while i<segtime*fs:
j=temp_large[i]
data.append(j)
i=i+1
print ("Writing data...")
dataset[...] = data
print(dataset)
#read the data
print ("Reading data back...")
data_read = dataset[...]
print ("Printing data...")
print (data_read)
file.close()
```

Appendix D

```
#Jingyi Zhang
#Pioneer Program
#2018.9
#This script is a adaptation version of the
#"find the inspiral" script provided by LOSC
#This script would try to find the software injection
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig
import h5py
import sys
#-----
#-- Read in data and template
#-----
fs = 4096
dataFile = h5py.File('injection for 1 solar mass.hdf5', 'r')
largedata = dataFile['strain/Strain'][...]
dataFile.close()
time = np.arange(0, 400, 1./fs)
data=[]
i=0
while i<400*4096:
j=largedata[i]
data.append(j)
i=i+1
templateFile = h5py.File('1mass.hdf5', 'r')
template = templateFile['strain/Strain'][...]
temp_time = np.arange(0, template.size / (1.0*fs), 1./fs)
templateFile.close()
#-----
# Plot data and template
#-----
plt.figure()
plt.plot(time,data)
plt.xlabel('Time (s)')
plt.ylabel('Strain')
plt.title('Strain')
plt.figure()
plt.plot(temp_time, template)
plt.xlabel('Time (s)')
plt.ylabel('Strain')
plt.title('Template')
```

```

plt.show()
#-----
# Apply a bandpass filter to the data
#-----
(B,A) = sig.butter(4, [80/(fs/2.0), 250/(fs/2.0)], btype='pass')
data_pass= sig.lfilter(B, A, data)
plt.figure()
plt.plot(time, data_pass)
plt.title('Band passed data')
plt.xlabel('Time (s)')
plt.show()
#-----
# Time domain cross-correlation
#-----
correlated_raw = np.correlate(data, template, 'valid')
correlated_passed = np.correlate(data_pass, template, 'valid')
plt.figure()
plt.plot(np.arange(0, (correlated_raw.size*1.)/fs, 1.0/
fs),correlated_raw)
plt.title('Time domain cross-correlation')
plt.xlabel('Offset between data and template (s)')
plt.figure()
plt.plot(np.arange(0, (correlated_passed.size*1.)/fs, 1.0/fs),
correlated_passed)
plt.xlabel('Offset between data and template (s)')
plt.title('Band passed time domain cross-correlation')
plt.show()

```