

Max-Planck-Institut für Gravitationsphysik, (Albert-Einstein-Institut)  
Astrophysical Relativity

---

**DATA ANALYSIS OF  
CONTINUOUS GRAVITATIONAL WAVES**

**Dissertation**  
zur Erlangung des akademischen Grades  
"doctor rerum naturalium"  
(Dr. rer. nat.)  
in der Wissenschaftsdisziplin "Physik"

eingereicht an der  
Mathematisch-Naturwissenschaftlichen Fakultät  
der Universität Potsdam

von  
Iraj Gholami Ghadikolaei

Potsdam, den 02.11.2007

Online published at the  
Institutional Repository of the Potsdam University:  
<http://opus.kobv.de/ubp/volltexte/2008/1880/>  
<urn:nbn:de:kobv:517-opus-18800>  
[<http://nbn-resolving.de/urn:nbn:de:kobv:517-opus-18800>]





To the memory of: *my father*

&

To: *my mother, my wife*

*and*

*my daughter; Hasti*



## Acknowledgment

First of all I thank my family who supported and encouraged me to continue my study in this level, especially my mother who put all of her efforts to help and give me enough energy to always stand and follow the progress in my life. I would also thank my wife, Azam, who I consider her as a *genius physicist*. Even though my field of study was far far from what she did, but she could easily understand the topics and in many times helped me to understand the mathematics behind of part of my works. She has also provided a warm atmosphere in the family and always encouraged me not to give up, especially during last year when I was a bit tired in my research. I should not certainly forget to thank my sweet daughter, Hasti, who came to our life during the time we were doing our PhD studying. Just give this assurance that her presence in our life was one of the best gift the God has ever offered us to always have hope for the future.

My sincere thanks goes to Dr. Curt Cutler, who has admitted me as a PhD student and supervised me during my research in this level. A great scientist with a very nice personality. I will always proud to say that I was his student. I should have also especial thank to him because of his effort to go through entire of my thesis for corrections with the very precise and valuable comments.

This is also my honor to say that I was working in the Professor Bernard Schutz's group and being his student. A great scientist who has taught me how to think when I face any scientific problem. I cannot forget that how he would think on a problem and be quite precise to find a right solution. I also thank to him because of his financial support and giving me enough time to finish my thesis. I therefore thank to the Max Planck Society, especially the Max Plank Institute for Gravitational Physics – Albert Einstein Institute (AEI) – for their support and hospitality.

I would also thank Dr. Maria Alessandra Papa with the best respect, who was involved in my research work recently and supervised me on my second project. Even being busy by many scientific activities and having two children, she had spent enough time for me, especially when I needed her help to finish my thesis. I should not forget how she was serious in correcting my thesis to be submitted in the proper time.

I had shared an office with Dr. Badri Krishnan for 4 years. We had collaborated in both of my research projects. He was always very kind to help me to understand the topics and had enough patience to let me recover my mistakes. I therefore thank him for all of his kindness and help.

For the second project in my thesis, I have had a very close collaboration with Dr. Reinhard Prix. He is also a capable computer programmer. We had very good times as well as challenging times together!! He always encouraged me to be independent and try to solve my problems alone. I do thank him for all of his effort on this regards.

My especial thank goes to Dr. Markus Pössel who had helped me quite friendly to edit the English of my thesis. He had spent a lot of time on it, especially when he was busy himself but gave me a more priority.

I should also thank Dr. Alicia Sintés, who was not a full time member in AEI, but I have used her help quite sometimes in my research projects.

I also thank Dr. Graham Woan from Glasgow University, UK, and Dr. Alberto Vecchio from Birmingham University, UK, who have accepted my request to be my thesis reviewers.

I would also thank to the AEI's staff who were always available to help us in providing our necessities. My special thanks goes to Ute Schlichting, the secretary of Professor Schutz, who had always time for me to be helpful. I never forget her response when I asked her, "do you have time a couple of minutes, I have some questions?" and her reply was "I am here to have always time for you to solve your problem". She was always helpful even for our personal problems. I should also thank Katharina Henke, who was always kind to solve our personal problems. I especially thank the computer support for all of their help and facilities provided to make our research easier. My thanks to the library staffs of AEI, especially Mrs. Schlenk and Ms. Lehmann who always tried to find the books and scientific papers I needed. I do also thank the personal office of the AEI, especially Mrs. Schneehufer and Ms. Witte.

At the end, I would like to thank everybody who has provided a friendly environments during my research at AEI, especially our football teams that we had a very nice moment every week together to play. Although we could not come up in a good level in our annual tournaments, but we had a very nice

people in our team.

I would like to thank my Iranian friends, Mohammad Mehrnia, Salma Balazadeh, Nima Yazdanbakhsh, Reza Siahpoosh, Masood Sultani, Mohammd Ahmadabadi, Aliyar Javadi and everybody else who I do not remember their name at this moment.



## Abstract

This thesis describes two main projects; the first one is the optimization of a hierarchical search strategy to search for *unknown* pulsars. This project is divided into two parts; the first part (and the main part) is the semi-coherent hierarchical optimization strategy. The second part is a coherent hierarchical optimization strategy which can be used in a project like Einstein@Home [1]. In both strategies we have found that the 3-stages search is the optimum strategy to search for unknown pulsars.

For the second project we have developed a computer software for a coherent Multi-IFO (Interferometer Observatory) search. To validate our software, we have worked on simulated data as well as hardware injected signals of pulsars in the fourth LIGO science run (S4). While with the current sensitivity of our detectors we do not expect to detect any true GW signals in our data, we can still set upper limits on the strength of the gravitational waves signals. These upper limits, in fact, tell us how weak a signal strength we would detect. We have also used our software to set upper limits on the signal strength of known isolated pulsars using LIGO fifth science run (S5) data.



## Zusammenfassung

Zwei Projekte wurden im Rahmen dieser Dissertation ausgeführt. Das erste Projekt bestand aus einer Studie über allgemeine hierarchische Suchstrategien für Gravitationswellen (GW) von unbekanntem, isolierten GW Pulsaren. Insbesondere haben wir dabei hierarchische Algorithmen betrachtet, in denen in jeder Schritt (außer dem letzten) aus einer kohärenten Demodulation über kurze Datenssegmente besteht, gefolgt von dem nötigen “stacking and sliding” der  $\mathcal{F}$ -Statistik Werte von den verschiedenen Segmenten. Diese mehrfachen Schritte dienen dazu, die Anzahl der Kandidaten rasch zu reduzieren. Der letzte Schritt besteht aus der Berechnung einer kohärenten Detektionsstatistik über alle verbleibenden Kandidaten.

Wir haben diese Strategie optimiert durch Minimierung der erforderlichen Rechenleistung  $P$  unter den Nebenbedingungen der gegebenen verfügbaren Datenmenge und des gewünschten Konfidenzlevels. Natürlich hängt  $P$  auch von der Größe des Parameterraumes ab, insbesondere von dem zu durchsuchenden Intervall in Frequenz und des (“spin-down”) Pulsaralters. Bei dieser Optimierung, und durch Variieren der Anzahl der hierarchischen Schritte  $n$ , fanden wir dass die Vorteile dieser mehrfach-Schritt Strategie bei  $n = 3$  “sättigen” (d.h.  $n = 4$  und  $5$  bringen kaum noch Verbesserungen), sowohl für semi-kohärente also auch für kohärente Suchmethoden.

Die angegebenen Werte für die optimierten Suchparameter,  $(N^{(i)}, \Delta T^{(i)}, \mu^{(i)})$  für semi-kohärente Suchen und  $(T^{(i)}, \mu^{(i)})$  für kohärente Suchen, sind das Ergebnis einer Reihe von Vereinfachungen: i) unsere Berechnungen verwenden mehrfach theoretische Abschätzungen der Anzahl der erforderlichen Rechenoperationen anstelle von Messungen der Operationen an existierenden Codes; ii) Fragen von Speicherplatz oder der benötigten Rechenleistung für Monte-Carlo Simulationen wurden nicht berücksichtigt, und iii) das Rauschen in den Detektordaten wurde durchwegs als stationärer Gaußscher Zufallsprozess angenommen. Weiters sind die Formeln für die Anzahl der benötigten Filter (“templates”) genau genommen nur gültig für Beobachtungszeiträume die signifikant kürzer sind als ein Jahr.

Daher sollten unsere Abschätzungen nur als ein erster grober Richtwert betrachtet werden, um eine solche Suche in der Praxis durchzuführen; wenn bessere Approximationen verfügbar sind, sollten diese Werte neu berechnet werden. Trotz dieser Einschränkungen sind wir zuversichtlich, dass diese Ergebnisse nützliche Größenordnungen über die Eigenschaften von solchen optimierten Suchen über den ganzen Himmel liefern.

Diese Studie eröffnet eine Reihe von möglichen weiteren Untersuchungen:

- Um verbesserte Abschätzungen für die Optimierung zu erhalten, wäre ein logischer nächster Schritt die aktuell implementierten Suchprogramme zu verwenden, und diese auf reale Detektordaten anzuwenden. In diesem Zusammenhang sollte man 1) Zeitmessungen der Suchprogramme durchführen, um die tatsächlich benötigte Rechenleistung zu bestimmen, und 2) Monte-Carlo Studien anstellen, um die *tatsächliche*  $\alpha^{(i)}$  für verschiedene Schwellenwerte zu bestimmen (da ja die hier angegebenen Formeln auf der Annahme von Gaußischem stationärem Rauschen basieren). Letzteres könnte mit beachtlichem Arbeitsaufwand verbunden sein, insbesondere wenn die Ergebnisse stark frequenzabhängig sind, sodass gewisse Frequenzbänder günstigeres Verhalten zeigen als andere.
- Es wäre nicht kompliziert, unsere Studie auf Suchen auszudehnen, die weniger Rechenaufwand benötigen als solche über den ganzen Himmel, welche aber immer noch durch den nötigen Rechenaufwand beschränkt sind; z.B. könnte man Suchen nach unbekanntem Neutronensternen in Supernova-Resten (wie etwa SN 1987A) betrachten, wobei die Himmelsposition mit großer Genauigkeit bekannt ist, aber die Frequenz und die Verlangsamungsparameter durch die Suche selbst bestimmt werden müssen. In ähnlicher Weise könnte man Suchen über einen kleinen Teil des Himmels betrachten, z.B. über das galaktische Zentrum, oder einen anderen bestimmten Teil der Galaxie.
- Die Formeln für die Anzahl der Rechenoperationen, Konfidenzintervalle usw. können auch für den Fall berechnet werden, in dem man die Hough-Transformation [62] statt des “stack-slide” Verfahrens im semi-kohärenten Schritt verwendet; die Optimierung von hierarchischen mehrfach-Schritt Suchen mit der Hough Methode würde denselben Schritten folgen wie hier dargelegt.
- Wir vermuten dass die Einsichten, die in diesem Projekt gewonnen wurden, auch in Suchen nach GW Pulsaren in Röntgen-binären von geringer Masse (LMXBs) zum Tragen kommen werden, welche ebenfalls durch die benötigte Rechenleistung limitiert sind [72]. Die Details hierzu müssen jedoch noch ausgearbeitet werden.

- Das hier behandelte Problem der Suche nach GW Pulsaren ist ähnlich einem noch schwierigerem Suchproblem in LISA Daten: die Suche nach den Signalen von der Umkreisung kompakter Objekte von etwa Sonnenmasse um ein schwarzes Loch der Masse  $\sim 10^6 M_\odot$  im Zentrum von Galaxien. Was die nötige Rechenleistung betrifft, stellt dieses Problem eine noch größere Herausforderung dar [89], und wir vermuten, dass die Erfahrungen, die in diesem Projekt gesammelt wurden, auch sehr nützlich für die Formulierung und Optimierung des Such-Algorithmus für solche LISA Signale sein wird.
- Wir haben in diesem Projekt implizit angenommen, dass die Suche von einem einzelnen Computer oder Computercluster ausgeführt wird. Zumindest in den nächsten paar Jahren werden jedoch die rechenintensivsten GW Suchen auf `Einstein@Home` ausgeführt werden [1]. Dieser Zugang bringt zusätzliche Einschränkungen mit sich, die wir hier nicht betrachtet haben, wie z.B. die Rate, mit der Daten und Zwischenergebnisse zwischen Servern und den Computern der Teilnehmer ausgetauscht werden können, und der verfügbare Speicherplatz, den man auf den teilnehmenden Computern verwenden kann. Wir beabsichtigen, die hierarchischen Suchstrategien auch in diesem Zusammenhang zu studieren, um zu erforschen ob die obigen Ergebnisse für `Einstein@Home` anders ausfallen würden.

Im zweiten Projekt, das Teil dieser Dissertation war, haben wir einen existierenden Suchcode (*ComputeFStatistic.c*, von nun an “CFS\_v1”) aus der LAL Softwarebibliothek zur Berechnung der  $\mathcal{F}$ -Statistik (die Detektionsstatistik, die in dieser Dissertation behandelt wird), erweitert und verallgemeinert; der daraus entstandene Code *ComputeFStatistic.v2.c* (“CFS\_v2”) wurde ebenfalls in LAL integriert. Die Hauptverbesserungen sind, dass CFS\_v2 auch multi-Detektor Suchen ausführen kann; weiters verwendet dieser Code SFT-spezifische Gewichtungen entsprechend dem Rausch-Level, was eine bessere Behandlung von nicht-stationärem Rauschen erlaubt. Eine zusätzliche Erweiterung von CFS\_v2 erlaubt den Rückschluss auf die Signalparameter, und es werden die abgeschätzten Parameter des “lautesten” Kandidaten ausgegeben. Durch die Verwendung von einem modifizierten LALDemod ist CFS\_v2 auch schneller als CFS\_v1.

Im ersten Teil dieses Projekts haben wir CFS\_v2 validiert, nämlich durch den Vergleich der Ergebnisse mit der früheren (und ausgiebig getesteten) Version CFS\_v1. In diesem Validierungsprozess fanden wir, dass CFS\_v2 und CFS\_v1 konsistente Ergebnisse lieferte in allen Fällen, wie z.B. bei reinem Rauschen, einem Signal im Rauschen, als auch einem reinen Signal ohne Rauschen. Es stellte sich heraus, dass im Fall von Daten, in denen das Rauschen dominiert, wie etwa bei reinem Rauschen oder Daten mit schwachem Signal, die Verwendung unserer Gewichtungen einen Unterschied zwischen CFS\_v2 und CFS\_v1 im Bereich von -25% to 25% bewirkt. Bei Deaktivierung dieser Option zeigen beide Codes jedoch sehr gute Übereinstimmung.

Wir haben unseren Code weiters an Daten aus dem vierten LIGO “Science Run” (S4) getestet, welche Hardware-“Injektionen” von Signalen enthielten. Wir haben die Parameter der eingeschleusten Signale mit denen verglichen, die unsere Software gemessen hat, und fanden sehr gute Übereinstimmung.

Im nächsten Teil dieses Projekts präsentierten wir unsere Ergebnisse von CFS\_v2 mit Daten von S5, um eine Obergrenze für 33 bekannte isolierte Pulsare zu bestimmen. Diese Pulsare sind aus der Radioastronomie bekannt; ihre Parameter wurden online von ATNF (Australia Telescope National Facility) bezogen. Wir haben unsere Ergebnisse mit denen aus einem “Time Domain Search” (TDS) verglichen, welcher einen grundlegend unterschiedlichen Zugang verwendete. Die Ergebnisse zeigen eine Abweichung zwischen -30% bis 30%, welche wiederum hauptsächlich darauf zurückzuführen sind, wie die verschiedenen Suchtechniken auf nicht-stationäres und nicht-Gaußsches Rauschen “reagieren”. Schließlich haben wir ein Jahr Daten der C03 Kalibrierung aus S5 verwendet, um Obergrenzen für die Gravitationswellenamplitude der obigen 33 Pulsare zu bestimmen. Diese Analyse zeigt, dass bei der Verwendung von einem Jahr Daten die beste erreichbare Empfindlichkeit in einem einzelnen Detektor  $6.0 \times 10^{-26}$  ist, nämlich für den H1 Detektor. Eine Suche mittels aller drei LIGO Detektoren zeigt, dass die beste kombinierte Empfindlichkeit  $4.27 \times 10^{-26}$  ist. Basierend auf diesen Obergrenzen der GW Stärke haben wir Obergrenzen für die Elliptizität aller 33 erwähnten Pulsare bestimmt, und zwar mittels jedes des LIGO Detektoren für sich, als auch für deren Kombination.

Trotz ausgiebiger Studien (im Frequenzraum) mittels unsere neuen verbesserten Software gibt es immer noch viele Bereiche, die weiterer Untersuchung bedürfen. Zum Beispiel, die Möglichkeit nach Signalen von Binärsystemen zu suchen sollte in unsere derzeitige Software integriert werden, eine Suche über das gesamte Datenset muss ausgeführt werden, eine verbesserte Einbindung der Daten von elektromagnetischen Beobachtungen (zum Beispiel für den Crab Pulsar), und schließlich sollten gezielte Suchen in interessanten

Bereichen am Himmel durchgeführt werden, zum Beispiel über unser galaktisches Zentrum, von dem man annimmt dass es viele massive schwarze Löcher enthält.



# Preface

Gravitational Waves (GWs), ripples in space-time which travel at the speed of light, are a fundamental consequence of Einstein's General Theory of Relativity. Due to the great distance to any likely detectable sources of GWs, the signal amplitude reaching us will be very small. Because of limitations in technology, there have been no direct detections of GWs so far. However, with the future generation of detectors, like Advanced LIGO, we should be able to detect a variety of sources.

Different types of GW signals are emitted by different types of sources. For example, we can expect bursts from supernovas, chirping signals from the binary coalescence of compact objects and a stochastic background of GWs from the Big Bang. Another class of signal which is generally referred to as *continuous waves* are emitted by *Neutron Stars*. This is, in fact, the type of signal that we concentrate on in this thesis.

Because of their nature, continuous GWs reaching Earth are expected to be extremely weak, even compared to the other types of waves like bursts. Therefore even with the quite significant sensitivity of our current detectors, it will be quite difficult to detect them. One possible way to increase the overall signal compared to the background noise (signal-to-noise ratio) is to coherently integrate the data for *several days* up to *few years*. For this we need to know the exact shape of the gravitational wave. This is the basic concept of *matched filtering*, which is the optimal detection method in a statistical sense. Throughout this thesis, this technique will be one of our main tool to extract the signal from the noise.

Our templates take into account for the Doppler shift, due to the motion of the Earth compared to the source, which modulates the signal in our data. Therefore the templates are sky-position dependent.

For a coherent all-sky search for unknown sources, the large number of points in parameter space makes the search very computationally expensive after a few hours of coherent integration. However, for a directed search, for which the source position is already known, we can reduce the cost of computing dramatically. And for a targeted search, where all of the source's parameters are previously known by the other means, like radio observations, the search can be performed very fast and just with a single computer by coherently analyzing months of data. It is therefore clear that the major constraint on wide range parameter searching is the computational cost. Beside this fact, we must also take care of the sensitivity of the search which increases with the observation time. Our aim is to develop search techniques that reduce the computational cost while reducing the sensitivity as little as possible.

One technique that can be used to reduce the number of points in parameter space is the so-called semi-coherent search. This method, in which the data is divided into several segments, is less sensitive than a fully coherent search using the same length of data. However, as semi-coherent searches require fewer templates in parameter space, we can use more data in the search in order to compensate for the loss of sensitivity.

To go further in increasing the amount of data and therefore the sensitivity, we can use a hierarchical semi-coherent method. This technique is also based on semi-coherent searching, but performed in several stages. In the first stage we use a small amount of data, a low threshold, and a coarse mesh in parameter space, and we keep just those points that pass the threshold. For the second stage the number of points in parameters space are reduced dramatically, and therefore we can use much more data as well as a finer search in parameter space. This is continued in the further stages until we have our final candidates. The final stage uses quite a long interval of data, and a fine grid, making this technique quite sensitive with low computational cost.

This thesis describes two main projects. The first one is the optimization of a hierarchical search strategy to search for *unknown* pulsars. This project is divided into two parts; the first part (and the main part) is the semi-coherent hierarchical optimization strategy. The second part is a coherent hierarchical optimization strategy which can be used in a project like Einstein@Home [1].

Einstein@Home is a project that uses the idle computers of worldwide users to search for the gravitational waves emitted by neutron stars. In this project, the data taken from the LIGO and GEO detectors is divided into small chunks and then distributed to the users around the world, who have registered themselves for that. The inspiration for this is the earlier project SETI@Home [2] which searches for *Extraterrestrial Intelligence* using radio observation data.

The second project develops the computer software for a coherent Multi-IFO (Interferometer Observatory) search. To validate our software, we have worked on simulated data as well as hardware injected signals of pulsars in the fourth LIGO science run (S4).

While with the current sensitivity of our detectors we do not expect to detect any true GW signals in our data, we can still set upper limits on the strength of the gravitational waves signals. These upper limits, in fact, tell us how weak a signal strength we would detect. We have also used our software to set upper limits on the signal strength of known isolated pulsars using LIGO fifth science run (S5) data. We have compared our results with those obtained by the gravitational-wave pulsar group in University of Glasgow, using a different analysis method.

The first chapter of this thesis is introduction which summarizes the basic knowledge required to work in gravitational-wave astronomy. This chapter covers many things in brief. In the second chapter, we characterize the signal we are searching for, adding a few words about basic statistics and the derivation of the likelihood function (the main tool used in our searches). In chapter 3 we present our work on the hierarchical semi-coherent and coherent search strategy. In chapter 4 we present the theory of a multi-IFO search using the  $\mathcal{F}$ -Statistic as well as its computational implementation. At the end of this chapter we also describe our scheme for estimating the source parameters. In chapter 5, we first validate our software using different types of simulated data and give upper limit results on the signal strength of some known isolated pulsars.



# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>1</b>  |
| 1.1      | Nature of Gravitational Waves . . . . .                           | 1         |
| 1.2      | Sources of Gravitational Waves . . . . .                          | 4         |
| 1.2.1    | Narrow-Band Sources . . . . .                                     | 5         |
| 1.2.2    | Coalescing Binaries . . . . .                                     | 7         |
| 1.2.3    | Burst Sources . . . . .   | 7         |
| 1.2.4    | Stochastic Background . . . . .                                   | 8         |
| 1.3      | Detectors . . . . .   | 9         |
| 1.3.1    | Bar Detectors . . . . .   | 9         |
| 1.3.2    | Laser Interferometric Detectors . . . . .                         | 11        |
| 1.3.3    | Future Detectors . . . . .  | 15        |
| <b>2</b> | <b>Basic Concepts</b>   | <b>19</b> |
| 2.1      | Signal phase model . . . . .                                      | 19        |
| 2.2      | Signal at the detector . . . . .                                  | 21        |
| 2.3      | Basic statistics . . . . .  | 23        |
| 2.4      | Maximum likelihood function ( $\mathcal{F}$ -statistic) . . . . . | 28        |
| 2.5      | Short time-baseline Fourier Transform (SFT) . . . . .             | 30        |
| <b>3</b> | <b>Optimized Stack-Slide Hierarchical Search Strategy</b>         | <b>31</b> |
| 3.1      | The Stack-Slide algorithm (based on re-sampling method) . . . . . | 32        |
| 3.2      | A multi-stage hierarchical search . . . . .                       | 34        |
| 3.2.1    | The general algorithm . . . . .                                   | 34        |
| 3.2.2    | The general optimization scheme . . . . .                         | 34        |
| 3.3      | Metric, template counting, . . . . , computational cost . . . . . | 37        |
| 3.3.1    | Parameter space's metric and template counting formulae . . . . . | 37        |
| 3.3.2    | False dismissal rates and the thresholds . . . . .                | 40        |
| 3.3.3    | Computational Cost . . . . .                                      | 42        |
| 3.4      | Results . . . . .   | 45        |
| 3.4.1    | The optimization method . . . . .                                 | 45        |
| 3.4.2    | The number of semi-coherent stages . . . . .                      | 46        |
| 3.4.3    | The optimal three-stage search parameters . . . . .               | 47        |
| 3.4.4    | The spin-down age and the SNR . . . . .                           | 52        |

|          |  |            |
|----------|--|------------|
| 3.5      | Optimization of coherent hierarchical search strategy . . . . .                  | 53         |
| <b>4</b> | <b>Multi-IFO Coherent Search</b>   | <b>57</b>  |
| 4.1      | Generalized maximum likelihood function for multi-IFO . . . . .                  | 57         |
| 4.2      | Discretization of the $\mathcal{F}$ -statistic's components (LALDemod) . . . . . | 60         |
| 4.3      | Semi-Analytical calculation of the $\mathcal{F}$ -statistic . . . . .            | 64         |
| 4.4      | Parameter estimation for the signal . . . . .                                    | 65         |
| 4.5      | Using noise weights . . . . .  | 67         |
| <b>5</b> | <b>Observational Results</b>   | <b>69</b>  |
| 5.1      | Software validation . . . . .  | 69         |
| 5.1.1    | Consistency software checks . . . . .  | 69         |
| 5.1.2    | Consistency with expectation for pure noise . . . . .                            | 71         |
| 5.1.3    | Consistency with expectation for signal plus noise . . . . .                     | 72         |
| 5.1.4    | Parameter estimation . . . . .   | 72         |
| 5.2      | S5 data - Upper limit for known isolated pulsars . . . . .                       | 76         |
| 5.2.1    | Pulsar selection . . . . .   | 78         |
| 5.2.2    | Frequentist Upper Limits . . . . .   | 79         |
| 5.2.3    | Bayesian Upper Limit . . . . .   | 81         |
| <b>A</b> | <b>Computational cost of the SFT method</b>                                      | <b>103</b> |
| <b>B</b> | <b>Computer codes used in optimization strategies</b>                            | <b>105</b> |
| B.1      | Main file . . . . .  | 105        |
| B.2      | Header file (Header.h) . . . . .   | 108        |
| B.3      | Sensitivity, False Alarm, False Dismissal . . . . .                              | 110        |
| B.4      | Ameba . . . . .  | 118        |
| <b>C</b> | <b>CVS version of the codes</b>  | <b>123</b> |
| <b>D</b> | <b>Curriculum Vitae</b>  | <b>133</b> |
| D.1      | PERSONAL INFORMATION . . . . .   | 133        |
| D.2      | EDUCATIONS . . . . .   | 133        |
| D.3      | PUBLICATIONS . . . . .   | 134        |
| D.4      | SEMINARS & SCHOOLS ATTENDED (during my PhD at AEI) . . . . .                     | 135        |

# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | The effect of gravitational wave passing through a ring of free falling masses. . . . .                             | 3  |
| 1.2  | Sources of gravitational waves and the sensitivities of different detectors. . . . .                                | 5  |
| 1.3  | Three stages of a coalescence black holes binary. . . . .   | 7  |
| 1.4  | A simple view of an Interferometric detector. . . . .   | 12 |
| 1.5  | The LIGO sensitivities during different LIGO science runs (S1-S5). . . . .  | 15 |
| 2.1  | Diagram of the SSB and the detector's position . . . . .  | 20 |
| 2.2  | The orientation of neutron star emitting the gravitational waves . . . . .  | 22 |
| 3.1  | Breaking data in the Stack-Slide algorithm . . . . .  | 33 |
| 3.2  | An illustration of sliding the frequency bins to correct the Doppler shifts . . . . .                               | 33 |
| 3.3  | A hierarchical scheme for the analysis of large parameter space for CGW searches . . . . .                          | 35 |
| 3.4  | Handling data in different stages in the data recycling mode . . . . .  | 36 |
| 3.5  | Handling data in different stages in the fresh-data mode. . . . .   | 37 |
| 3.6  | Computational power versus number of semi-coherent stages . . . . .   | 47 |
| 3.7  | Computational power $P$ as a function of $\Delta T^{(1)}$ and $\Delta T^{(2)}$ . . . . .                            | 51 |
| 3.8  | Computational power $P$ as a function of $N^{(1)}$ , $N^{(2)}$ and $N^{(3)}$ . . . . .                              | 51 |
| 3.9  | Computational power $P$ as a function of $\mu_{\max}^{(1)}$ , $\mu_{\max}^{(2)}$ , and $\mu_{\max}^{(3)}$ . . . . . | 51 |
| 3.10 | The minimum computational power $P$ to analyze 1 year's data as a function of $\tau_{\min}$ . . . . .               | 52 |
| 3.11 | The 1-year SNR (with zero mismatch) as a function of $\tau_{\min}$ , for $P = 10^{13}$ . . . . .                    | 53 |
| 3.12 | Computational cost for a three stages coherent search versus the total observational time. . . . .                  | 54 |
| 3.13 | Computational cost for 10 days of data to be analyzed coherently in a multi stages search. . . . .                  | 55 |
| 5.1  | Histogram of differences between CFS_v2 and CFS_v1 . . . . .  | 70 |
| 5.2  | Histogram of differences between CFS_v2 and CFS_v1 disabling noise weight . . . . .                                 | 70 |
| 5.3  | Histogram of differences between CFS_v2 and CFS_v1 for signal and noise . . . . .                                   | 71 |
| 5.4  | Distribution of the $\mathcal{F}$ -statistic . . . . .  | 72 |
| 5.5  | Relation between the value of $\mathcal{F}$ -statistic and the mismatch between PFS and CFS . . . . .               | 73 |
| 5.6  | Mismatch between PFS and CSF from the value of $\mathcal{F}$ -statistic less than 1500 . . . . .                    | 74 |
| 5.7  | Parameter estimation of simulated signals . . . . .   | 75 |
| 5.8  | Detection of PULSAR3 . . . . .  | 76 |
| 5.9  | PDF of $h_0$ and $\mu$ . . . . .  | 77 |
| 5.10 | PDF of $\psi$ and $\Phi_0$ . . . . .  | 78 |
| 5.11 | The list of 45 known isolated pulsar selected from the ATNF catalogue. . . . .                                      | 80 |

|      |  |    |
|------|--|----|
| 5.12 | A Flow-Chart of how we implemented the frequentist upper limit. . . . .                        | 82 |
| 5.13 | Comparison between the FDS frequentist upper limit with TDS's Bayesian results . . . . .       | 83 |
| 5.14 | Discrepancy in TDS and FDS frequentist upper limit vs the value of $2\mathcal{F}$ . . . . .    | 83 |
| 5.15 | Relation between the sets A, B and S in the definition of conditional probability. . . . .     | 83 |
| 5.16 | Comparison in UL between Mathematica, automated algorithm based on PDF. . . . .                | 86 |
| 5.17 | Periodogram ( $ \tilde{X} ^2$ ) of the 13.5 months of S5 data for H1 detector. . . . .         | 86 |
| 5.18 | Histogram of mismatch between FDS and TDS UL for 11 days of S5 H1 data. . . . .                | 87 |
| 5.19 | Noise floor of the 11days of S5 data for H1 detector. . . . .                                  | 88 |
| 5.20 | Mismatch between TDS and FDS UL for short term of S5 data of, three case. . . . .              | 88 |
| 5.21 | Posterior probability; pulsar J0024-7204N for the 11 days of S5 L1 data. . . . .               | 89 |
| 5.22 | The Posterior probabilities of $h_0$ of simulated pulsars. . . . .                             | 90 |
| 5.23 | The Posterior probabilities of four amplitude parameters of simulated pulsar, PULSAR3. . . . . | 91 |
| 5.24 | Distribution of 95% upper limit on $h_0$ for simulated data. . . . .                           | 91 |
| 5.25 | One year upper limit of H1 and H2 S5 data located on the LIGO sensitivity curve . . . . .      | 94 |
| 5.26 | One year upper limit of L1 and Joint S5 data located on the LIGO sensitivity curve . . . . .   | 95 |
| 5.27 | PDFs of known isolated pulsars of H1 . . . . .   | 96 |
| 5.28 | PDFs of known isolated pulsars of H2 . . . . .   | 97 |
| 5.29 | PDFs of known isolated pulsars of L1 . . . . .   | 98 |
| 5.30 | PDFs of known isolated pulsars of Joint IFO . . . . .  | 99 |

# List of Tables

|      |  |    |
|------|--|----|
| 3.1  | The optimal search parameters in data recycling mode . . . . .                                     | 47 |
| 3.2  | The computational cost to analyze one year of data in data-recycling mode. . . . .                 | 48 |
| 3.3  | Same as Table (3.1), but for fresh-data mode. . . . .  | 48 |
| 3.4  | Same as Table (3.2), but for fresh-data mode. . . . .  | 48 |
| 3.5  | Search parameters for data-recycling mode . . . . .  | 49 |
| 3.6  | The computational cost to analyze one year of data in data-recycling mode. . . . .                 | 49 |
| 3.7  | Search parameters for fresh-data mode . . . . .  | 49 |
| 3.8  | Same as Table (3.6), except for fresh-data mode. . . . .   | 50 |
| 3.9  | The optimal search parameters for coherent hierarchical search in data recycling mode . .          | 54 |
| 3.10 | The distribution of required Computational Cost (CP) in different stages . . . . .                 | 55 |
| 5.1  | Mean and standard deviation of $\mathcal{F}$ -statistic for 129600 individual outputs . . . . .    | 73 |
| 5.2  | Mean and standard deviation for the average of 50 computations of the mean of the $2\mathcal{F}$ . | 73 |
| 5.3  | The parameters of the S4 hardware injected isolated pulsars. . . . .                               | 74 |
| 5.4  | Upper limit on the strength of signal on S4 hardware injected data of different detectors.         | 79 |
| 5.5  | The amount of data used for the short period of time . . . . .                                     | 87 |
| 5.6  | Parameters of 4 pulsars injected for comparison of FDS and TDS upper limit. . . . .                | 89 |
| 5.7  | 95% upper limit of 4 simulated pulsars in FDS and TDS . . . . .                                    | 90 |
| 5.8  | Upper limits on $h_0$ of known isolated pulsars for one year of S5 data. . . . .                   | 92 |
| 5.9  | Ellipticity upper limits of known isolated pulsars for one year of S5 data. . . . .                | 93 |



# Chapter 1

## Introduction

In this chapter, we briefly introduce some essential phenomena and topics that are required to understand the basic physics of gravitational-wave astronomy and the means of its detection. This chapter is basically useful for those who want to have in one place a summary of the most important topics about the nature, generation, sources and the detection of gravitational-waves.

### 1.1 Nature of Gravitational Waves

Einstein's development of the general theory of relativity [3, 4] brought about a new understanding of gravity, that *the force of gravity is actually an effect created by the curvature of space-time around massive bodies*. Early acceptance of General Relativity was based on two important tests. One of them was the anomalous precession in the perihelion of Mercury, which Newton's theory of gravitation could not explain, however its amount could be precisely predicted by the general theory of relativity. Another was the amount of light bending when the light passes near a massive object. In 1916, Einstein found that the general theory of relativity predicts the amount of bending to be twice the value given by Newton's theory of gravitation. In 1919, Eddington discovered that the amount of the light bending around the sun (during the total solar eclipse in that year) confirms the prediction of the general theory of relativity [5].

These evidences convinced people that the theory itself is basically correct. And just as Maxwell's Electromagnetic theory predicts the propagation of electromagnetic waves with the speed of light, so the theory of general relativity also predicts gravitational waves to propagate at the speed of light. This fact can be extracted from the Einstein's field equation (with the notation as used in Landau and Lifshitz [6]),

$$G_{\alpha\beta} \equiv R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R = \frac{8\pi G}{c^4}T_{\alpha\beta}, \quad (1.1)$$

where  $R_{\alpha\beta}$  is the Ricci tensor,  $g_{\alpha\beta}$  is the space-time metric,  $R$  is the scalar curvature,  $G$  is the universal gravitational constant,  $c$  is the speed of light, and  $T_{\alpha\beta}$  is the energy momentum tensor. Of course there can be another term  $\Lambda$ , the cosmological constant, in this equation. However for our purpose we set it to zero here.

In vacuum, the energy momentum tensor vanishes,  $T_{\alpha\beta} \equiv 0$ , and in Transverse Traceless (TT) gauge the field equation reduces to a simple wave equation form [7]

$$\left(-\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2\right)\bar{h}_{\alpha\beta} = 0, \quad (1.2)$$

which leads to a plane-wave solution of the form [8]

$$\bar{h}_{\alpha\beta} = \Re[A_{\alpha\beta} \exp(ik_\mu x^\mu)], \quad (1.3)$$

where  $\Re$  denotes the real part of the complex components; the wave vector  $k_\mu$  and wave amplitude  $A_{\alpha\beta}$  (which can be complex) are constants satisfying

$$k_\mu k^\mu = 0 \quad (\mathbf{k} \text{ a null vector}), \quad (1.4)$$

$$A_{\alpha\mu} k^\mu = 0 \quad (\mathbf{A} \text{ orthogonal to } \mathbf{k}). \quad (1.5)$$

This solution describes a wave with frequency

$$\omega \equiv k^0 = (k_x^2 + k_y^2 + k_z^2)^{1/2}, \quad (1.6)$$

which propagates with the speed of light,  $c$ , in the direction of  $(1/k^0)(k_x, k_y, k_z)$  [8].

Before we go through the effect of the gravitational wave on matter and on the details of its polarizations, let us briefly mention here how a gravitational wave is generated. Just as the generation of electromagnetic waves which is due to a time changing Dipole Moment, so the generation of gravitational waves is due to the changes of the source's Quadrupole Moment in time. The radiation of electromagnetic waves is usually due to the time dependency of its dipole moment

$$d_i = \int \rho x_i d^3x, \quad (1.7)$$

where  $\rho$  is the charge density and  $x_i$  is a Cartesian coordinate and integration is over the volume. If this integral is time-dependent, then the amplitude of the waves is proportional to its first time-derivative,  $\frac{d}{dt}(d_i)$ . However, since in general relativity the mass-energy is conserved, the time-derivative of the dipole moment is just the integral of the momentum density

$$\dot{d}_i = \int \rho v_i d^3x. \quad (1.8)$$

But this is the total momentum in the system and so is constant. Thus there can be no dipole radiation in general relativity. Therefore we need to look one order higher than the dipole approximation terms, namely the quadrupole terms

$$Q_{lm} = \int \rho x_l x_m d^3x. \quad (1.9)$$

Following Schutz [9], the amplitude of the gravitational waves is proportional to the second time derivative of  $Q_{lm}$ ,

$$h_{lm} = \frac{2G}{rc^4} \frac{d^2 Q_{lm}}{dt^2}, \quad (1.10)$$

where  $r$  is the distance to the source.

To summarize: *the Gravitational Waves are produced by masses which move in such a way that their quadrupole moment varies with respect to time.*

To investigate the effects of gravitational waves on matter, let us consider a test point mass which is caught in this wave. The question is; what would be the response of this mass to the change of curvature around it? In reality, if the mass is alone in space, it will not feel any changes. The mass will be freely falling, which means that it will experience no forces. If we consider a pair of point test masses which are arranged in a line perpendicular to the direction of the gravitational wave, then the relative distance between the masses will alternatively stretch and squeeze as the wave passes through them. In this case, the effect of wave will be such that the center of mass of the system is not displaced.

If a third point test mass is added to the system and arrayed perpendicular both to the line between the first two masses and to the direction of travel of the gravitational wave, the distance between the first two masses will be observed to shrink as the distance between the third mass and the first two masses stretches, and vice versa. As with the two-mass system, the wave will affect the masses such that the center of mass of the entire system remains constant. This effect is put to use in the construction of gravitational wave detectors, which we will discuss at length later. Fig. (1.1) shows the effect of a gravitational wave when it passes through a ring of test masses. The ratio of averaged changes in the distance between two relative masses ( $\delta L_1 - \delta L_2$ ), due to the passing of a gravitational wave, over the original distance ( $L$ ) between them will give us the amplitude of the wave

$$h \equiv \frac{\delta L_1 - \delta L_2}{L}. \quad (1.11)$$

In contrary to the above expression, if the masses are arranged in a line parallel to the direction of the gravitational wave, they will experience no change in their relative distance. This is due to the fact that there is no change in the metric in the component along the axis of the wave, as is being discussed in more detail below.



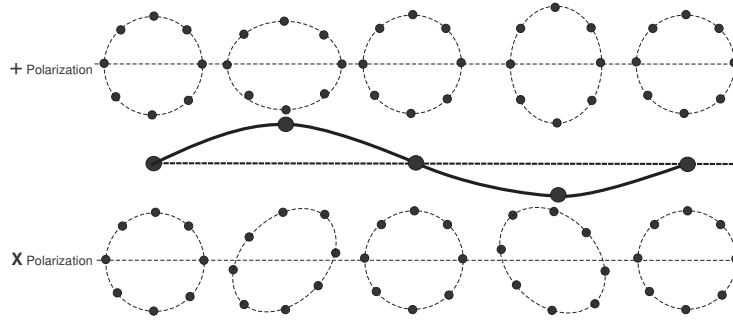


Figure 1.1: The effect of gravitational wave passing perpendicular through a ring of free falling masses. The  $x$  axis indicates the time and the  $y$  axis is the amplitude of the wave,  $h = (\delta L_1 - \delta L_2)/L$ , where  $L$  is the radius of the ring and  $\delta$  indicates the change in it.

Einstein's Theory of Special Relativity applies to a space-time where there is no gravity, *i.e.*, flat space-time. According to this theory, the invariant interval (relative distance between two points in space-time), is expressed as

$$ds^2 = -c^2 dt^2 + dx^2 + dy^2 + dz^2, \quad (1.12)$$

where  $dt$  is the time interval between the two events, and  $dx$ ,  $dy$ , and  $dz$  are the spatial separation of these two events in Cartesian coordinates. We can write the above formula in the form of tensor

$$ds^2 = \eta_{\mu\nu} dx^\mu dx^\nu, \quad (1.13)$$

where  $\mu$  and  $\nu$  are indices independently equal to  $t, x, y$ , or  $z$ , and  $\eta$  is the Minkowski metric tensor:

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.14)$$

We have previously stated that the gravitational wave is a wave of space-time perturbation; that is, a ripple of space-time curvature propagating through space at the speed of light. We have also assigned an amplitude for such a wave,  $h_{\mu\nu}$ , as described in Eq. (1.10), which is in fact the fractional change in the metric caused by a gravitational wave (see below). The metric can then be rewritten as an addition to the Minkowski metric of flat space-time to account for a passing gravitational wave

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu, \quad (1.15)$$

where  $g_{\mu\nu} = (\eta_{\mu\nu} + h_{\mu\nu})$  and  $|h_{\mu\nu}| \ll 1$ . Since in the absence of a gravitational field the space-time is flat, a weak gravitational field is an approximation in which the space-time is assumed to be 'nearly' flat.

Now, let us consider a gravitational wave which is traveling along the  $z$  axis. As previously stated, parallel to the wave traveling direction we have no changes. Therefore the metric change has  $x$  and  $y$  coordinate changes only. This means that  $g_{\mu\nu}$  will only have changes in the tensor with indices of  $x$  and  $y$ , or where  $\mu$  and  $\nu$  are equal to 1 or 2. The tensor  $g_{\mu\nu}$  is symmetric, so  $h_{\mu\nu}$  is also symmetric. The metric perturbation  $h_{\mu\nu}$  then takes the form (assuming we work in Transverse Traceless (TT) gauge)

$$h_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (1.16)$$

where  $h_+$  and  $h_\times$  are constants and denote the two different polarizations of the wave. The  $h_+$  terms describe the metric change in the  $x$  and  $y$  axis directions (since they are in the slots corresponding to  $dx dx$  and  $dy dy$ , or  $dx^2$  and  $dy^2$ ). And since we work in the Traceless gauge, the *trace* of the matrix must be zero (*i.e.*  $h_{xx} + h_{yy} = 0$ , which leads us to the  $h_{xx} = -h_{yy}$ ). The  $h_\times$  terms describes a wave whose

polarization is rotated at a  $45^\circ$  angle to the  $h_+$ . In the case of sinusoidal gravitational wave, these wave polarizations can be expressed as

$$h_+ = h_{xx} = \Re[A_+ e^{-i\omega(t-z/c)}], \quad (1.17)$$

$$h_\times = h_{xy} = \Re[A_\times e^{-i\omega(t-z/c)}], \quad (1.18)$$

where  $A_+$  and  $A_\times$  are the strain amplitude of each polarization.

## 1.2 Sources of Gravitational Waves

In previous section we summarized how the time-dependent behavior of the quadrupole moment of an object causes the emission of gravitational waves. For most cases, the Luminosity of the radiated waves is proportional to the square of the time derivation of the quadrupole moment [9]. The total Luminosity takes the form

$$L_{gw} = \frac{G}{5c^5} \left( \sum_{l,m} \ddot{Q}_{lm} \ddot{Q}_{lm} - \frac{1}{3} \ddot{Q}^2 \right), \quad (1.19)$$

where  $Q$  is the trace of the matrix  $Q_{lm}$  (Eq. 1.9). The dependency of the distance to the source,  $r$ , goes away from this equation because the total Luminosity is an integration over a sphere of radius  $r$ . Apart from the effect of  $\ddot{Q}_{lm}$ , the factor  $G/5c^5 = 5.49 \times 10^{-54} sJ^{-1}$  sets the characteristic gravitational radiation power output to be extremely small. Therefore, the generation of detectable gravitational waves in a laboratory is practically impossible.

To categorize the possible sources of gravitational waves, we therefore need to estimate the maximum wave amplitude reaching us from, for example, the strongest detectable source. To do this we make a very simple estimation such a way that if we consider the motion inside the source is highly non-spherical, then Eq. (1.10) tells us that the strength of the radiated gravitational wave signal is proportional to  $\ddot{Q}/r$ . On the other hand, from Eq. (1.9) we would have an amplitude of  $Mv_{N.S.}^2$  for  $\ddot{Q}$ , where  $v_{N.S.}^2$  is the non-spherical part of the squared velocity inside the source [9]. Therefore,

$$h \sim \frac{2G}{c^4} \frac{Mv_{N.S.}^2}{r}. \quad (1.20)$$

The virial theorem for self-gravitating tells us that  $v_{N.S.}^2$  cannot be larger than the maximum value of the Newtonian gravitational potential *inside* the system,  $\phi_{int}$ . Thus, by having all of the above information in hand, the minimum detectable strength of the source amplitude would follow as [9]

$$h < \frac{2G}{c^2} \frac{M\phi_{int}}{rc^2}, \quad (1.21)$$

where  $M$  is the mass of the system and  $r$  is the distance to the source. According to the above formula, if our source is a neutron star, then we have  $\phi_{int} \sim 0.2c^2$  [9]. If this star is in the Virgo cluster (with  $r \sim 18Mpc$ ), then the upper limit on the amplitude of the radiation would be  $1.5 \times 10^{-21}$ . This means that our detector should be able to observe a wave amplitude at or below this value.

Fig. (1.2) shows the possible sources of gravitational waves and their detectabilities by different type of detectors. We expect to detect the waves from the formation of black holes and neutron stars, and the coalescence of binary neutron stars and the final collapse of such binaries to form a black hole. At low frequencies we expect to find continuous stochastic backgrounds created from large numbers of discrete sources, such as binary stars in our own Galaxy. In our Galaxy we expect to find many quasi-monochromatic sources of gravitational waves such as individual binary star systems, including binary neutron stars as they evolve towards coalescence, and various rotating neutron star systems such as millisecond and x-ray pulsars. We may also be able to observe gravitational waves from the Big Bang, amplified during the inflationary epoch [10]. As the Fig. (1.2) shows, we have still some frequency ranges in which our detectors are not sensitive. For a complete overview of the sources and their detectabilities, we refer to the review article by Cutler and Thorne [11].

We can divide all above mentioned sources into four different classes, according to the methods of data processing and signal extraction [10]. These four classes can be listed as; (i) Narrow-Band Sources, (ii) Coalescing Binaries, (iii) Burst Sources, and (iv) Stochastic Backgrounds, which we describe them in more detail below.

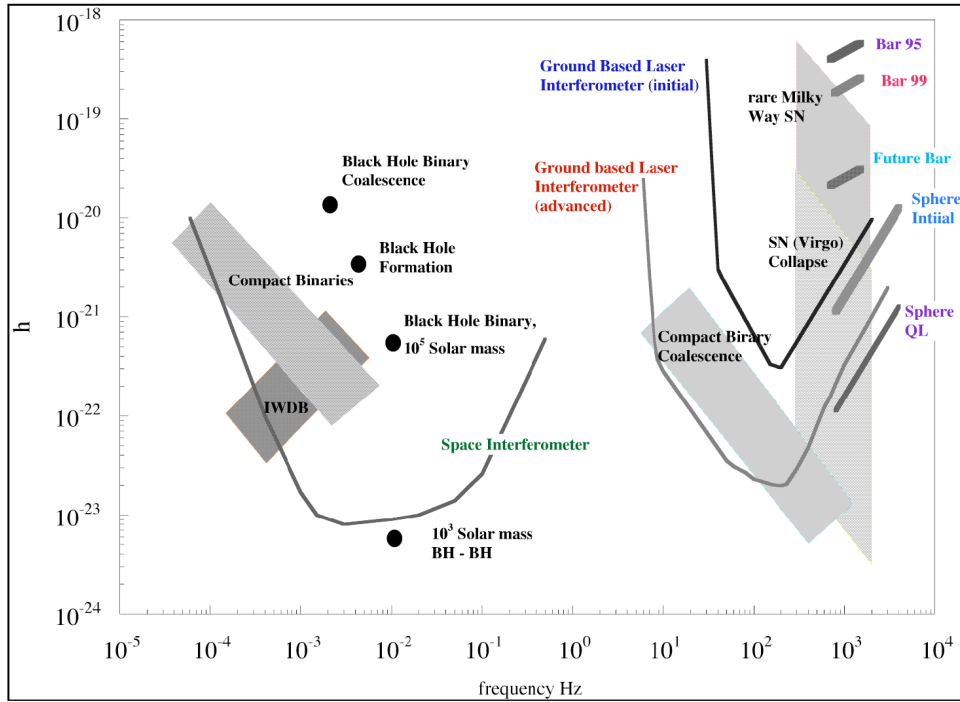


Figure 1.2: The possible sources of gravitational waves and the sensitivities of different type of detectors for different range of frequencies. Image from L. Ju *et al.* [10].

### 1.2.1 Narrow-Band Sources

The sources in this class include the rapidly rotating neutron stars that have non-axisymmetric rotation, as well as binary star systems that are very far from coalescence.

Because the gravitational waves take energy out of the system, the periods generally evolve, so the waveforms will be quasi-periodic. In general they are also periodically Doppler shifted by binary motion and/or the Earth's motion. To increase the signal-to-noise ratio (SNR), and because of the long lifetime of such sources, in principle we can always do a long-term integration of the data to extract the signals from the noise; the scales like  $\sqrt{T}$ , where  $T$  is the total observation time. Integrating the data for a long time of course requires accurate knowledge of the frequency modulations to maintain a coherent integration. With this knowledge, we can use the match filtering technique to extract weak signals from noise. Basically in match filtering technique we multiply the signal by a template and then integrate it.

#### Rotating Neutron Star

The newborn neutron stars, which are formed in a gravitational collapse, may initially rotating very fast. If their period is less than 1 ms, then they may be subjected to a class of gravitational radiation driven instabilities (CFS instability) [12]. Their effect will be to radiate angular momentum away till they spin down to a marginally stable condition.

For an isolated neutron star, two main mechanisms may cause the radiation of gravitational waves: precession of their axis or a bumpy surface due to a mountain. If we assume that the irregularity in its shape is due to a bump of mass  $m$  on its surface, then  $v_{N.S.} = 2\pi fR$  and according to Eq. (1.20) the estimated gravitational amplitude due to this bump would be [9]

$$h_{bump} \approx \frac{8\pi^2 G}{c^4} \frac{mR^2 f^2}{r}, \quad (1.22)$$

where  $R$  is the radius of star,  $f$  is the rotational frequency and  $r$  is the distance to the observer. If we assume that the rotation axis is  $z$ , then

$$mR^2 = I_{xx} - I_{yy} = \epsilon I_{zz}, \quad (1.23)$$

where  $\epsilon \equiv (I_{xx} - I_{yy})/I_{zz}$  is the ellipticity. We can therefore re-write Eq. (1.22) as

$$h \approx 1.05 \times 10^{-24} \left[ \frac{\epsilon}{10^{-6}} \right] \left[ \frac{I_{zz}}{10^{38} \text{kg m}^2} \right] \left[ \frac{f}{500 \text{Hz}} \right]^2 \left[ \frac{1 \text{kpc}}{r} \right], \quad (1.24)$$

where  $I = 10^{38} \text{ kg m}^2$  represent the value of moment of inertia for a typical neutron star with 1.4 solar masses and radius of 10km. This shows that even for the case of rapidly rotating star ( $f = 500 \text{Hz}$ ) and typical ellipticity ( $\epsilon = 10^{-6}$ ), the amplitude of the gravitational wave signal is quite weak. However, as was stated before, since we can integrate their data for a reasonably long period of the time, the SNR can be amplified by the factor of  $\sqrt{T}$ , where  $T$  is the observation time.

### Accreting Neutron Star

A neutron star accreting mass from its binary companion can be spun up until it reaches the CSF instability. The time scale to reach in this instability would be shorter than the accretion timescale (which is in order of few hundred million years for a accretion rate of 0.1 Eddington limit). The unstable mode grows in amplitude until the accreted angular momentum is balanced with the angular momentum radiated away by the gravitational wave. This will generate radiation at single and steady frequency. The accreting matter also emits X-rays where the energy radiated in gravitational wave should be proportional to the x-ray flux. The gravitational wave amplitude is roughly [12]

$$h \approx 2 \times 10^{-28} \left( \frac{300 \text{Hz}}{f} \right) \left( \frac{F_x}{10^{-13} \text{J m}^{-2} \text{s}^{-1}} \right), \quad (1.25)$$

where  $f$  is the frequency of radiating mode and  $F_x$  is the x-ray flux. In the case where the instability exists, the temperature, mass, viscosity and the equation of state of the neutron star will determine the value of  $f$  [12].

### Binary System

A system of two objects rotating around each other is an especially promising source of gravitational waves. If we have two objects of the same mass  $M$  rotating around each other in a circular orbit of radius  $R$ , then the second derivative of the quadrupole moment is  $GM^2/4R$  [9]. Therefore according to Eq. (1.10), the amplitude of the gravitational waves is

$$h_{\text{binary}} \sim \frac{1}{2} \frac{G^2}{c^4} \frac{M^2}{rR}. \quad (1.26)$$

The gravitational wave luminosity therefore takes the form [9]

$$L_{\text{binary}} \sim \frac{1}{80} \frac{G^4}{c^5} \left( \frac{M}{R} \right)^5. \quad (1.27)$$

This tells us that the more massive and closer objects will radiate more energy. As the two masses orbit around each other and radiate energy, their orbit shrinks and they come closer and closer until they merge. Thus the frequency of the wave increases with time. This is called a *chirp*. The timescale for this is [9]

$$t_{\text{chirp}} = \frac{d^2 Q / dt^2}{L_{\text{binary}}} \sim \frac{20c^5}{G^3} \frac{R^4}{M^3}. \quad (1.28)$$

This timescale including with the  $h_{\text{binary}}$  for neutron star binaries observed by the ground-base detectors give an effective amplitude of [9]

$$h_{\text{chirp}} \sim h_{\text{binary}} \sqrt{f_{\text{GW}} t_{\text{chirp}}} \sim \left( \frac{G}{c^2} \right)^{3/4} \frac{(M^3 R)^{1/4}}{r}. \quad (1.29)$$

According to Cutler and Flanagan [14], the frequency evolution of inspiraling binaries follows

$$\frac{df}{dt} = \frac{96}{5} \pi^{8/3} \mathcal{M}^{5/3} f^{11/3}, \quad (1.30)$$

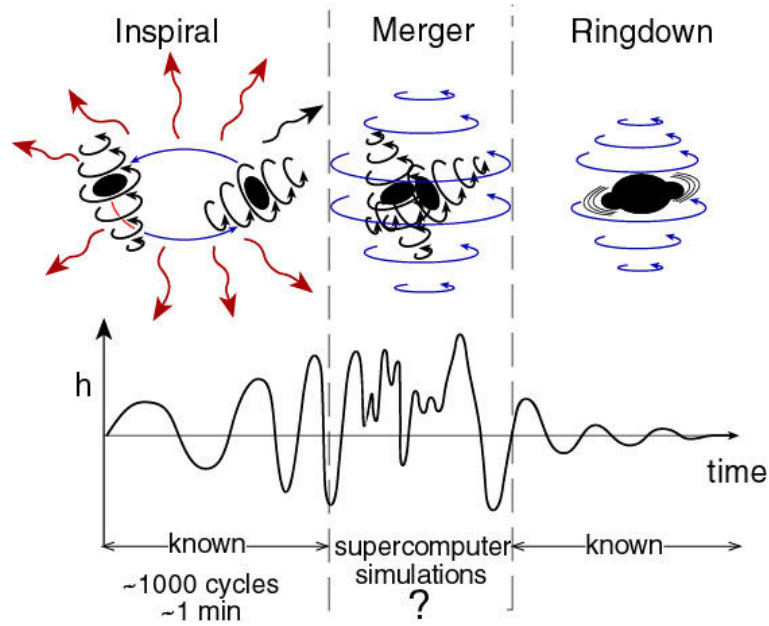


Figure 1.3: Three stages of a coalescence black holes binary. Image from K.S.Thorne [13].

where  $\mathcal{M} \equiv \mu^{3/5} M^{2/5}$  is the chirp mass parameter with  $\mu$  as of reduced mass (for this case,  $\mu = \frac{M_1 M_2}{M_1 + M_2} = 0.5M$ ). So, starting from the same frequency the more massive objects live shorter.

If one can observe the shrinking time by detecting the gravitational wave, then mass of objects can be found. If we also measure the amplitude of the signal, therefore the only unknown parameter would be the distance to the source. Therefore by observing the gravitational wave we can directly measure the distance to the source which is not easy to do by observing their electromagnetic radiation.

### 1.2.2 Coalescing Binaries

The coalescence of compact binaries, consisting either black holes or neutron stars, are the most promising sources of gravitational waves for the ground based detectors. As mentioned above, a binary system loses energy due to the gravitational wave radiation. This loss causes the orbit to shrink until the two bodies merge. If the orbiting objects are black holes, then we can divide the coalescence of this binary system into three parts as *inspiral*, *merger* and *ring-down* (Fig. 1.3).

In the inspiral epoch, the signal frequency would sweeps upward to the maximum of  $f_{max} = 1$  kHz for binary neutron stars and  $f_{max} = 10/(M_1/M_\odot)$  kHz for black holes with the larger mass of  $M_1$  [15]. The characteristic amplitude of the radiated waves from the inspiraling binaries is [15]

$$h_c = 0.237 \left( \frac{\mu^{1/2} M^{1/3}}{r f_c^{1/6}} \right) = 4.1 \times 10^{-22} \left( \frac{\mu}{M_\odot} \right)^{1/2} \left( \frac{M}{M_\odot} \right)^{1/3} \left( \frac{100\text{Hz}}{f_c} \right)^{1/6} \left( \frac{100\text{Mpc}}{r} \right), \quad (1.31)$$

where  $\mu = M_1 M_2 / (M_1 + M_2)$  is the reduced mass,  $r$  the distance to the source and  $f_c$  the characteristic frequency of the wave. By studying the waveform of the received signal using a network of the detectors, we will be able to (i) find the direction of the source by measuring the phase difference in the signal at different detectors with different location; (ii) the direction of the binary's orbital angular momentum by measuring the complex amplitudes of the different polarizations; (iii) the combination  $[\mu^3 (M_1 + M_2)^2]^{1/5}$  of the reduced and total mass of the objects; (iv) the ratio  $(\mu/M)$  from post-Newtonian corrections to Eq. (1.31) and (v) the distance to the source,  $r$  [15].

### 1.2.3 Burst Sources

This class includes sources such as the final coalescence of compact binary star systems, or the formation of neutron stars and black holes in supernova events. The binary coalescence events can consist of binary

neutron stars, binary black holes, or neutron star-black hole binaries.

The burst signal consists of a very short single event (of order of few milliseconds for stellar-mass objects), containing just one or very few cycles. The burst event is relatively broadband. Due to the very short lifetime of this event, to be detectable the signal must be strong enough to dominate over the detector noise in some *rectangle* in the time-frequency plane.

### Supernovae Core-Collapse

The supernova core-collapse produces a neutron star or black hole. The explosions last for a short time ( $\approx 100\text{ms}$ ). The explosion can be observed optically from the optical light emitted from the stellar surface starting hours after the collapse. However the envelope is optically thick and the optical observation cannot give any information about the core. Therefore gravitational waves would be the only possible means of having direct observation of the core collapse. Although the neutrinos will be emitted from the core bounce, but its flux decays with the square of the distance to us and therefore its detectability is quite low.

The signal radiated from the core collapse depends on its speed and its degree of non-sphericity; therefore the generated signal is complex and hard to model. Nevertheless, the waveform's characteristic amplitude can be approximated using the (unknown) total radiated energy,  $\Delta E_{\text{GW}}$ , as [15]

$$h_c \cong \left( \frac{3}{2\pi^2} \frac{\Delta E_{\text{GW}}/f_c}{r^2} \right)^{1/2} = 1.35 \times 10^{-20} \left( \frac{\Delta E_{\text{GW}}}{M_\odot c^2} \right)^{1/2} \left( \frac{1\text{kHz}}{f_c} \right)^{1/2} \left( \frac{20\text{Mpc}}{r} \right), \quad (1.32)$$

where  $M_\odot$  is the mass of the Sun and 20Mpc is the distance to the center of the VIRGO cluster (assuming the Hubble constant to be  $70 \text{ km s}^{-1} \text{ Mpc}^{-1}$ ),  $r$  the distance to the object and  $f_c$  the characteristic frequency. Defining  $\varepsilon \equiv \Delta E/Mc^2$ , this becomes [15]

$$h_c \cong \left( \frac{15}{2\pi} \varepsilon \right)^{1/2} \frac{M}{r} = 5.0 \times 10^{-21} \left( \frac{\varepsilon}{0.01} \right)^{1/2} \left( \frac{10^3\text{Hz}}{f_c} \right)^{1/2} \left( \frac{10\text{Mpc}}{r} \right). \quad (1.33)$$

### 1.2.4 Stochastic Background

Stochastic backgrounds can be produced by a variety of mechanism. They are produced from the integrated effects of many weak individual periodic sources in our Galaxy. Or they can be generated from a large population of burst sources at very large distances. They can also be the remnant of the cosmological processes in the early universe, *i.e.* Big Bang.

Stochastic backgrounds are difficult to detect in a single detector because they are practically indistinguishable from instrument noise. If the source was not isotropically distributed (such as a population of binary stars towards the center of our Galaxy), it might be detectable from the variation of observed instrument noise as the detector orientation varied on the rotating Earth. However, a much better way of detecting stochastic backgrounds is by cross-correlating two nearby detectors. In this case the correlated stochastic signal will integrate up in relation to the uncorrelated instrument noise (assuming both detectors to be truly independent). In this case the signal to noise ratio increases as  $N^{1/4}$ , where  $N$  is the effective number of cycles [10].

Much of our observational knowledge about the early universe comes from the Cosmic Microwave Background Radiation (CMBR), which was produced when the universe had a temperature of about 3000 K, corresponding to  $3 \times 10^5$  years after the Big Bang, and from studying nucleosynthesis, a process that took place in the first few minutes of the Universe's existence. By comparison, the gravitational radiation background was generated earlier than  $10^{-24}$  second after the Big Bang [9]. Therefore detecting the stochastic background of gravitational waves produced in the early universe would give us unique information about the universe very close to the Big Bang.

One of the scenarios that explains the homogeneity of the universe is inflation. If the inflationary picture is true, then it creates a gravitational wave background by parametric amplification.

We can characterize the stochastic background of gravitational wave by the energy density rather than its mean amplitude. For this, we compare (normalize) this energy density of the gravitational wave to the critical energy density required (today) to close the universe [16]. The critical density is determined



by today's expansion rate of the universe. In analogy to other contributions to the universe's energy density, the spectral function  $\Omega(f)$  for gravitational waves can be defined [16] by

$$\Omega_{\text{gw}}(f) = \frac{1}{\rho_{\text{critical}}} \frac{d\rho_{\text{gw}}}{d \ln f}. \quad (1.34)$$

This is roughly the fraction of the energy density in random gravitational waves between the frequency  $f$  and  $2.7f$ . The inflation scenario predicts a flat value of  $\Omega(f)$ , while any other alternative would have different spectrum of the gravitational wave [9].

For our purpose, it is convenient to convert the gravitational wave spectral density to a characteristic amplitude [16], as

$$h_c(f) \cong 3 \times 10^{-20} \sqrt{\Omega(f)} \frac{100\text{Hz}}{f}. \quad (1.35)$$

As an example, if  $\Omega(f) = 10^{-8}$ , over a frequency bandwidth of  $50\text{Hz} < f < 150\text{Hz}$ , the characteristic strain in an ideal detector is  $h_c \approx 10^{-24}$  [16].

After discussing all four main classes of gravitational waves sources, we see that the neutron stars can produce gravitational waves in all four classes. First, a large population of binary neutron stars in our Galaxy, with the range of orbital periods from days to minutes, can produce a stochastic background of individually unresolvable sources in the frequency band of  $\sim 10^{-2} - 10^{-5}$  Hz. In addition to the binary orbit, the individual rotation of the stars themselves (if they are non-axisymmetric) will also give rise to quasi-periodic gravitational waves. As the binary evolves and radiates away gravitational potential energy, it will gradually spiral inwards. Eventually the stars will coalesce, resulting in a short intense burst of gravitational waves.

## 1.3 Detectors

Although the existence of gravitational waves was observed indirectly by measurements on the Hulse-Taylor binary pulsar, the direct detection of that was always limited by the technology. The pioneering work of Joseph Weber began this field in the late 60's, and since then the actual search for the gravitational waves has been carried out using different types of detectors.

There have been two classes of detectors, **ground based** (under operation) and **space based** (under plan). The ground based detectors can also be divided into two different categories, **Bar** and **Laser Interferometric** detectors, which have different mechanisms of detection. The space based detectors are actually based on Laser Interferometry, but on a larger scale. We give a brief introduction to each below.

### 1.3.1 Bar Detectors

Bar antennas absorb energy from a passing gravitational wave in the frequency band in which they are mechanically resonant. The idea behind these devices is that as a gravitational wave passes through an object, it exerts a tidal force (squeezing and stretching), causing the object to deform. If the object is vibrating at a characteristic frequency and the wave has also the same frequency, then the deformation will appear as a resonance in the vibration. When a gravitational wave hits the bar, a transducer coupled to the bar's longitudinal mode amplifies the oscillation amplitude, producing an electrical signal that is converted into digital information.

The first aluminum bar detector was proposed in 1960 [18] and designed in late 1960s by Joseph Weber of the University of Maryland. He used a massive cylindrical bar, 1.5m long and weighing about a ton, at room temperature and in a vacuum chamber. He selected the frequency of 1660Hz for his experiment and compared the residual vibrations in the bar with the vibrations in similar bars separated by 1000 kilometers [19]. When the separated bars vibrated in coincidence, this was taken to be evidence for the existence of gravitational waves. Weber began to make credible claims of this sort in the late 1960s and early 1970s [19]. However, Weber's claims faltered after other physicists who had built their own more sensitive bar detectors could not find the signals that Weber claimed to have recorded. Work on bar detectors continued to progress, with sensitivity improvements and noise reductions of orders of magnitude being achieved. However, none of these next-generation bar detectors succeeded in definitively recording a gravitational wave signal either.

Weber's detector in fact was very similar to the modern cylindrical resonant bar detectors (called the second generation of bar detectors). The main difference is in the operating temperature (now detectors are kept at very low temperature – in the temperature of liquid Helium – to reduce the thermal noise) and in the conversion of the mechanical vibration signal into an electrical signal [20]. The sensitivity of an antenna (bar) can be improved in a number of ways. First, the force a gravitational wave exerts on the antenna can be increased by increasing the mass of the antenna. Second, the antenna can be made in such a way to be equally sensitive to all directions and polarizations. To fulfill this, a spherical resonant antenna was proposed and is under construction [21]. Third, the noise temperature (level of excitation from non-gravitational sources) of the antenna can be lowered. So, as result of such a proposal, to improve the sensitivity of the detectors, which is limited by quantum mechanical states of the bar, the next generation is moving toward greater masses.

Since the 1990's bar detectors have been sensitive enough to detect burst signals originating from supernovae in our galaxy and they have begun to look for continuous wave signals. They are currently organized as a network to look for gravitational wave bursts. Resonant detectors have also been used to search for coincidences between impulsive strain events observed in massive detectors and supernovae [22].

The current resonant bars are sensitive at frequencies of about 700 to 900 Hz, with band widths of about 50 Hz. Their best strain sensitivity is at the level of  $\sim 10^{-21} - 10^{-22} \text{ Hz}^{-1/2}$  [23], which is sufficient to detect gravitational wave bursts that may occur within our Galaxy. No resonant bar detector has yet detected a source, and the upper limit they provide on the spinning neutron stars is about  $2 \times 10^{-23}$  in the frequency band of 921.00 to 921.76 Hz [24].

Currently there are 5 such instruments in operation around the world. There have been at least two detectors in continuous coincident operation since 1993. The experiments are **ALLEGRO**, **AURIGA**, **EXPLORER**, **NAUTILUS**, and **NIOBI**. Some of the groups associated with these instruments are planning new experiments to make detectors of much greater sensitivity.

**ALLEGRO** (A Louisiana Low temperature Experiment and GRavitational wave Observatory) is an experiment operated by the Louisiana State University in Baton Rouge, Louisiana, USA [25]. This experiment employs an Aluminum bar cooled to 6K. It operates at a frequency of 900Hz, and has a strain sensitivity of  $1 \times 10^{-21} \text{ Hz}^{-1/2}$  [23]. ALLEGRO has had two data runs, with the first covering June 1991 through January 1995, and the second covering 1996 to the present time. A search for coincidences with the EXPLORER detector of the University of Rome was made for the last 6 months of 1991 and for the two months around the time of supernova 1993J. But no coincident events were found.

The **AURIGA** detector is run by the Istituto Nazionale di Fisica Nucleare (INFN) in Legnaro, Italy [26]. It is an Aluminum cylindrical bar of 2300kg mass, 3m long and 60cm diameter cooled to 0.1K. It operates at a frequency of 920Hz, with a sensitivity of  $2 \times 10^{-22} \text{ Hz}^{-1/2}$  [23]. AURIGA has had runs from June 1997 until November 1999 and a second from 2003 until now.

The **EXPLORER** detector is based at CERN and is operated by the University of Rome. It weighs 2300kg and 3m long and it has a diameter of 60cm Aluminum cylindrical bar cooled to 4.2K and operates at the temperature of 2K. It operates at the resonance frequencies of 906 and 923Hz, and has a strain sensitivity of  $6 \times 10^{-22} \text{ Hz}^{-1/2}$  at 923Hz [23].

The **NAUTILUS** detector is run by the Istituto Nazionale di Fisica Nucleare (INFN) at the Laboratori Nazionali di Frascati Italy. It weighs 2300kg, 3m long and has a diameter of 60cm. It is cooled at the temperature of 0.1K by means of a dilution refrigeration system. Its resonance frequencies are around 908 and 924Hz. NAUTILUS has a sensitivity of  $3 \times 10^{-22} \text{ Hz}^{-1/2}$  [23].

The **NIOBE** detector is operated by the University of Western Australia at Perth, Australia. It is a niobium bar operating at a temperature of 5K. It is sensitive at a frequency of 700 Hz and has a sensitivity of  $8 \times 10^{-22} \text{ Hz}^{-1/2}$  [23].

## Sources of Noises in Bar Detectors

There are many non-gravitational wave sources that can excite the resonances of the detector.

**Seismic Noise:** The largest of these sources can be external vibrations, such as ground noise. A sophisticated vibration isolation system is then necessary to keep these forces from exciting the antenna.



**Thermal Noise:** The effect of thermal noise itself in a bar detector is quite significant. As an example, for a typical bar detector like NAUTILUS or AURIGA cooled in 0.1K with mass of 2300kg and the resonant frequency as 910Hz, the rms amplitude of vibration would be [9]

$$\langle \delta l^2 \rangle_{th}^{1/2} = \left( \frac{kT}{4\pi^2 M f^2} \right)^{1/2} \sim 4.3 \times 10^{-18}, \quad (1.36)$$

which is much larger than a typical gravitational wave amplitude. However for a material with quite high quality factor  $Q$  (e.g.  $\sim 10^6$ ) in its fundamental mode, the changes in its thermal amplitude would be like a random walk with the steps that takes a time duration of  $Q/f \approx 1000$ s. This is the time for the mode energy to change by thermal energy [9]. But for a gravitational wave event like a burst, which last about 1ms, the changes due to this random walk would be  $(1000\text{s}/1\text{ms})^{1/2} = Q^{1/2}$  times smaller. Therefore the rms fluctuation in amplitude due to thermal effect would be  $\langle \delta l^2 \rangle_{th: 1\text{ms}}^{1/2} / Q^{1/2} \sim 4.3 \times 10^{-21}$  in this case.

This tells us that if the integration time of the observations is kept sufficiently short, the effect of the thermal noise can be minimized. On the other hand, having a larger observational time increases the signal-to-noise ratio (SNR). So, one of the best way to have a larger SNR in the same time of reducing the thermal noise is to use a high- $Q$  material.

**Sensor (or Series) Noise:** As we have mentioned earlier, the vibration in the bar detector is converted to an electrical signal by a transducer which is attached to the bar. This electrical signal is amplified and then recorded. But sensing the vibration is not perfect and produces some noise. This is an additive and wide-band noise which effectively reduces the sensitive band of the detector to a narrow range around the excitation frequency.

By reducing the amplifier noise as well as increasing the coupling of the transducer to the antenna, the series noise can be decreased. Increased coupling will boost the amount of signal energy transferred to the motion sensor without increasing the series noise, so there will be a net gain in signal-to-noise. However the motion sensors can also generate a force noise that can excite the resonant mass. This type of noise is referred to as back-action noise. Just as the transducer can see what the large mass is doing, the large mass can also see what the transducer is doing. Increasing the coupling causes more back-action noise to be transferred to the antenna.

### 1.3.2 Laser Interferometric Detectors

As was discussed in section 1.1, a gravitational wave alternatively stretches and squeezes the relative distance between two bodies. This motivates using a detector based on a Michelson-Morley interferometer. So, according to the laser interferometric technique, the relative change in the length between two bodies (in each arm),  $\delta L_i$ ,  $i = 1, 2$ , will cause a phase shift in the laser beam reaching the photo-detector. By measuring this phase shift, which appears in the movement of the interference fringes in the detector's photodiode, we can calculate  $\delta L_1$  and  $\delta L_2$ . For expected signal strength from the nearest detectable sources, the phase shifting would be of the order of  $\Delta\phi \approx 10^{-10}$  radian, or equivalently with  $\delta L = \delta L_1 + \delta L_2 \approx 10^{-20} - 10^{-18}m$ . The phase shifting can be formulated as

$$\Delta\phi = \frac{4\pi\delta L}{\lambda_{\text{laser}}} = \frac{4\pi Lh}{\lambda_{\text{laser}}}, \quad (1.37)$$

where  $\lambda_{\text{laser}}$  is the wavelength of the laser used for the experiment. Therefore the longer the light path, the larger is the phase shift for the same  $h$ . So, making the light path as long as possible would be one way to increase the sensitivity of the measurement. But, practically we are limited in doing that. If  $L > \lambda_{GW}/4$ , gravitational wave changes its sign while the laser light is still in the arms, which cancels the effect on  $\Delta\phi$ . Therefore we are limited to the arm lengths to be  $L \leq \lambda_{GW}/4$ . Suppose the frequency of GW is about  $f = 100\text{Hz}$ , therefore its wavelength would be  $\lambda_{GW} = c/f = 3000$  km. So, according to the above constraint we have  $L \leq \lambda_{GW}/4 \approx 750$  km. This is the maximum arm length that can be selected for the interferometer to detect a signal with that particular frequency. However, even though, reaching to that length is not in fact practical. Because it requires construction of a large vacuum tube with this size plus a powerful laser light that survive and does not diverge before reaching to the test mirror on the other end of the interferometer. What is practically possible is to let the laser beam travel many times inside the interferometer to take an approximate traveling distance near the above mentioned limit. This is done by using the Fabry-Perot cavity inside each of the detector's arms (see Fig. 1.4). If we want to reach to the maximum possible laser traveling distance, 750 km for this case, and to search

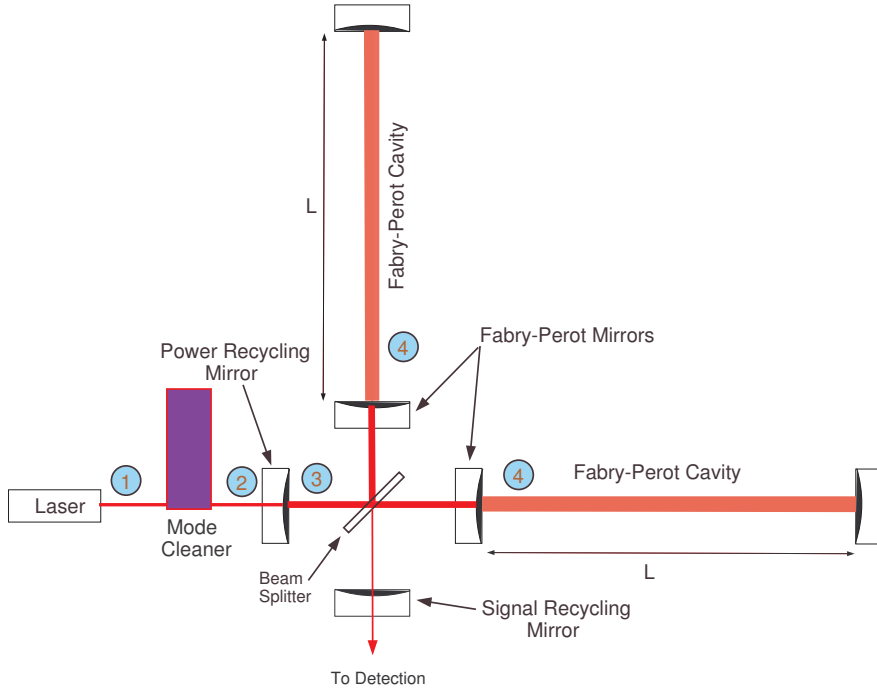


Figure 1.4: A simple view of an Interferometric detector.  $L$  is the length of the detector. The output power of the laser is usually in order of 10W (pointed by a blue circle with the number of 1). Then it is being cleaned and pure by removing noise in Mode cleaner. Therefore it reduces some power before it reaches the Power Recycling Mirror. As an estimate, the power of laser before it enters the main system would be in order of 5W (indicated by point 2). The two Fabry-Perot mirrors plus the power recycling mirror make the power recycling cavity. When the laser beam is being recycled in the recycling cavity, the gain would be by a factor of few tens. As an example, in point 3, the power of laser would be about 250W. At the end, in the Fabry-Perot cavity (point 4) the power of the laser increases significantly. As a rough example, it would be in order of 15KW. The signal recycling mirror, which is currently used in GEO600, allow us to tune the response of the interferometer. For more details we may refer the reader to the explanation in the text body.

for a GW with a frequency of 100Hz by using a 4 km arm length of interferometer, the laser beam must travel  $N \approx 200$  times inside the cavity. In this case the phase shift would be [27]

$$\Delta\phi = N \left( \frac{4\pi\delta L}{\lambda_{\text{laser}}} \right). \quad (1.38)$$

The use of Fabry-Perot cavity is one of the techniques used to increase the sensitivity of phase shift measurement. If the interference at the detection port is locked to the dark fringe, then all the power is reflected towards the laser. By means of an additional mirror placed between the laser and the beam splitter (see Fig. 1.4), the reflected light is re-injected into the interferometer. This technique is called *power recycling*, and the mirror used for this purpose is the *recycling mirror*. This mirror, together with two Fabry-Perot mirrors, make the power recycling cavity [28]. If the cavity length is properly tuned, the light resonates in it and the power entering the beam splitter is increased by a factor  $R$  (called the recycling factor). As a consequence, the interferometer sensitivity is improved by a factor  $\sqrt{R}$  [42].

There are still other noise sources in which must be taken into account. One of the most important, especially at low frequencies, is seismic vibrations. To mitigate this, a complex system of pendulum is used to suspend the mirrors [29].

Fig. (1.4) gives a schematic view of an Interferometric gravitational wave detector. To have a rough idea about the details of the elements shown there, let us review the facilities used in the LIGO detectors<sup>1</sup>

<sup>1</sup>LIGO detectors H1 and H2 correspond to the Hanford, Washington, 4km and 2km detectors respectively. L1 Indicates the 4km detector in Livingston, Louisiana.

[30]. The laser used in these detectors are in a single frequency at 1064nm with output power of 10W and beam radius of 3.6cm. The input power after the mode cleaner and before it enters into the power recycling cavity is 6W, 6W and 2.5W for H1, L1 and H2 respectively. The recycling gain for H1, L1 and H2 is 50, 40 and 60, respectively, which therefore gives an input laser power to the Fabry-Perot cavity of order of 300, 240 and 150 watts respectively. The stored power in the Fabry-Perot cavity is 20KW, 16KW and 10KW for H1, L1 and H2. The size of mirror for all detectors are 25cm in diameter and 10cm in thickness, with mass of 10.7kg. Increasing the mass of the mirror reduces the effect of thermal vibrations of the suspension, as well as fluctuations in radiation pressure. For example, the mass of the mirrors in VIRGO detector is 21.2kg. The displacement in the mirrors reveals itself as a phase shift at the photodiode at the dark port. Then the electrical signal from the photodiode is filtered and digitized continuously at a rate of 16384 Hz.

### Sources of Noise in Interferometric Detectors

In general, for ground-based detectors, the most important limitations to sensitivity are from seismic noise, thermal noise associated with the test masses and their suspensions, shot noise in the photocurrent from the photodiode, and radiation pressure recoil effects on the interferometer mirrors.

**Ground Vibration (Seismic) Noise:** Seismic noise is quite serious for a laser interferometer detector, since the laser beam travels many times inside the detector and each time is affected by these noises. Seismic noise at a reasonably quiet site on the earth follows a spectrum in all three dimensions close to  $10^{-7} f^{-1/2}$  m Hz<sup>-1/2</sup> [40]. So, if we look for a vibration in each test mass to be less than, say  $3 \times 10^{-20}$  ( $\sim 10^{-19}$  in total for 3 mirrors) at, for example, 30 Hz then we should have a reduction in the seismic vibration greater than  $10^9$  in the horizontal direction.

A simple way of reducing the horizontal fluctuation in the mirrors is to use a pendulum. A pendulum is a good filter for vibration above its natural frequency, because the transfer function to the pendulum mass of the horizontal motion of the suspension point falls off as  $1/f^2$  above the pendulum resonance [40]. Using a more complex pendulum system allows one to reduce the seismic noise at quite low frequencies. For example, GEO600, in principle by having a three-stage pendulum system, can have a good sensitivity below 50Hz. Or for the case of VIRGO, the 7 stages system allows the observation ideally to be in desire sensitivity to below 10Hz.

**Thermal Noise:** There are several important sources of the thermal noise, namely from the internal modes of the mirrors and the pendulum mode of suspended mirrors (the so-called violin mode of the suspension wires). These noises come from the thermal energy of the atoms and molecules in the test masses and suspensions. The thermal noise inside the mirrors show themselves as the Brownian motions. This noise occurs since there is a  $kT/2$  energy per degree of freedom in any mechanical system, with the importance of this noise being dependent on the level of internal friction in the system [41]. Therefore in selecting the mirrors as test masses, several requirements must be satisfied. For example, in the case of thermal noise, the mirror should have a low coefficient of thermal expansion and high thermal conductivity.

For long-baseline interferometers like the initial LIGO detectors, the thermal noise is significantly larger at the low frequency end of detector's operating range, say above 40Hz, and continues up to a couple of hundreds Hz.

**Photoelectron Shot Noise:** The laser light photons are quantized. This leads to a random fluctuation in the detected light intensity, which mimics a gravitational wave signal. The more photons, the less the relative fluctuation. The limits that this fluctuation (noise) makes on the phase spectral sensitivity can be given as

$$\Delta\phi_{shot} \sim 1/\sqrt{N}, \quad (1.39)$$

where  $N$  is the photon flux entering the beam splitter [42]. As a consequence of this relation, the detector sensitivity improves when the light power injected into the interferometer is increased.

**Radiation Pressure Noise:** The radiation pressure comes from the forces that laser light's photons exert on the mirror due to their impacts. Therefore for a freely hanging mirror, the radiation pressure pushes on the mirrors and results changes in the length of the cavity. This together with the pendulum restoring force, balance each other to produce an equilibrium position for the mirror. This uncertainty in the position of the mirror results the radiation pressure noise. Based on the origin of this noise, it is clear that a more powerful laser results more radiation pressure noise. But this is in contrast with the nature of shot noise, where a more powerful laser brings a less shot noise. The radiation pressure and the

shot noise together follow the Heisenberg uncertainty principle [43]. Therefore in practice they should be optimized by selecting a suitable power for laser and the mass of the mirror – the larger mass the lower in the radiation pressure noise. This is an issue that in the future detectors will be taken into account, when a more powerful laser will be used.

### Currently operating interferometric detectors

There are several interferometric detectors across the world which have started to collect data. We list them below with a short description.

The **LIGO** detectors are located in the USA. It comprises three detectors, with a 4km interferometer, H1, and a 2km interferometer, H2, using independent components, sharing the same vacuum tube at Hanford, Washington. A single 4km interferometer, L1, occupies a similar vacuum system at Livingston, Louisiana.

The **GEO600** detector, located near Hannover, Germany, is operated by a team of scientists from Germany and the United Kingdom. Its arms are 600 meters large. Because of its shorter arms, GEO600 is not expected to achieve a sensitivity comparable to LIGO, but more advanced suspension and vibration-isolation technologies should close some of that gap.

GEO600 also uses a signal-recycling mirror at the gravitational wave detection port. This mirror causes gravitational wave signals on the light to resonate in the interferometer within a certain band of frequencies. Changing the reflectivity and position of this mirror allows the response of the interferometer to be tuned. One can in principle choose to operate in a relatively broad band mode or as a narrowly tuned detector (Fig. 1.4).

The LIGO detectors, in cooperation with GEO600 (LIGO Scientific Collaboration (LSC)), have already done 5 science runs since 2002 until 1st October 2007.

From 23rd of August till 9th of September 2002 a first common data run (S1) between all three LIGO detectors and GEO600 was undertaken consisting of 17 days of mostly uninterrupted operation. Since the detectors had not yet achieved their designed sensitivity, the aim was to gain experience with data acquisition and data analysis [31].

The second LIGO science run (S2) took place over the period of 14 February 2003 to 14 April 2003.

The third LSC science run, S3, was performed between the 1st of November 2003 and 9th of January 2004. For GEO600, the commissioning work produced a stably-locked Dual Recycled (Power and Signal recycling) Michelson (DRMI) only about two weeks before the start of S3, and thus left little time for testing and sensitivity optimization. Therefore it was decided that GEO600 would run for about one week, near the beginning of S3, then go off-line to allow time to improve the sensitivity and stability of the detector, and finally rejoin the run. Thus, the GEO600 S3 science run was divided into two separate periods of time, having somewhat different interferometer configurations and sensitivities. The first period ran from 5th of November through 11th, and the second, from 30th of December to 13th of January (a few days after the end of the LIGO S3). These are referred to as S3-I and S3-II respectively.

The S4 science run, which lasted from 22 February to 23 March 2005, featured good overall “science mode” duty cycles of 80.5%, 81.4%, and 74.5% for H1, H2, and L1, respectively, corresponding to observation times of 570, 576, and 528 hours. Thanks to improvements made after the S3 run, the detector noise during S4 was within a factor of two of the design goal over most of the frequency band, as shown in Fig. (1.5). The GEO600 interferometer also collected data throughout the S4 run, but was over a factor of 100 less sensitive than the LIGO interferometers at 200 Hz and a factor of few worse at and above the 1kHz frequency range.

On 4 November 2005, LIGO started the S5 science run with the aim of taking data for more than a year. On 2 May 2006 GEO600 joined the S5. The S5 run was stopped at the 00:00 of the 1st October 2007. As can be seen from Fig. (1.5), the S5 run reached the LIGO designed sensitivity in most of the LIGO band. The plan for the future of LIGO detectors are explained in the next section under the *Enhanced* and *Advanced LIGO*.

The **VIRGO** detector is a 3-km instrument situated in Cascina (near Pisa), Italy, operated by a collaboration of scientists from Italy and France. VIRGO employs a similar configuration to the LIGO detectors, but has a more advanced vibration-isolation system. VIRGO’s best sensitivity is comparable with initial LIGO at the frequency range of above 1kHz, but is in a factor of 10 far in the frequency range of 50-300Hz

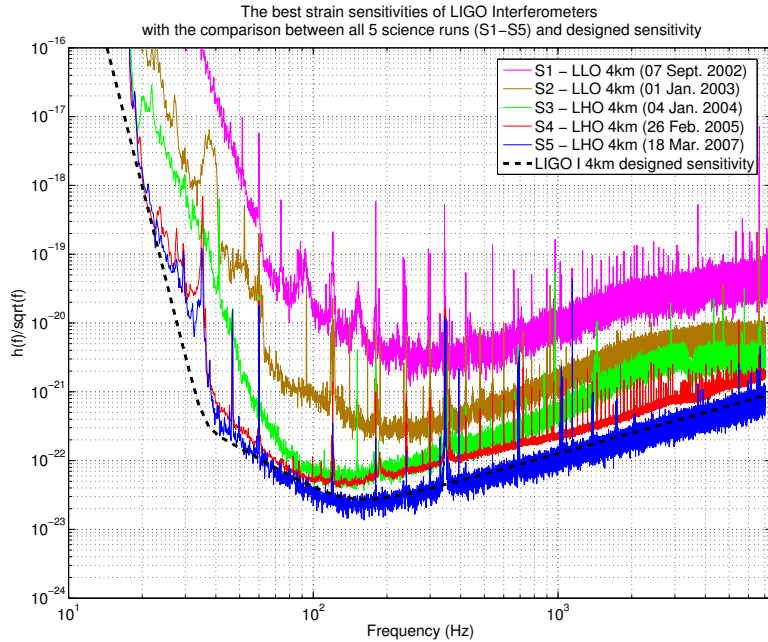


Figure 1.5: The LIGO sensitivities with the comparison among different LIGO science runs (S1-S5). The LIGO design sensitivity is also plotted with a dashed line. This plot shows that the LSC basically reached to the designed sensitivity in S5. In some frequencies the S5 has even better sensitivity than the design one. Data to plot these curves have been taken from [32].

[33]. VIRGO designed strain sensitivity is about  $2 \times 10^{-21} \text{ Hz}^{-1/2}$  at 10 Hz and about  $4 \times 10^{-23} \text{ Hz}^{-1/2}$  at the range of frequencies between 100 and 600 Hz. The best sensitivity that VIRGO could reach is about  $2 \times 10^{-22} \text{ Hz}^{-1/2}$  at the frequencies of 300-500 Hz, but is still far from its design sensitivity at 10 Hz [33]. After steady progress in the commissioning activity, the VIRGO detector is near to its design sensitivity. VIRGO had 7 Week-end Science Run (WSR) since September 2006 until January 2007 [33]. The first science run of the VIRGO started on 18th May 2007 and the plan was to run it for four months [34] which has coincidence partly with the S5 run of LIGO. VIRGO will be upgraded in 2008, mostly by increasing the laser light power from 20 to 50 Watts with an expected sensitivity improvement of about factor 3 [34]. The advanced VIRGO is planned for the year 2011 with an improvement in the sensitivity of factor 10. This requires for it to have laser power of about 200 Watts [34].

The **TAMA300** near Tokyo, Japan has 300-meter arm lengths. The basic optical configuration is similar to that employed in the LIGO and VIRGO detectors. Because of its smaller size, TAMA300 cannot achieve the sensitivity of the larger interferometers, but it was the first of these large-scale detectors to come into operation. It was started in 1995. The purpose of the instrument is to develop technology for a future kilometer-scale instrument. There have been 9 data runs with the instrument. The latest run, called Data Taking 9, started on 28 November 2003, and ended 10 January 2004, and overlapped the last half of LIGO's S3 run. The best sensitivity that TAMA300 could reach was  $5 \times 10^{-19} \text{ Hz}^{-1/2}$  which is several order of magnitude far from LIGO detectors.

### 1.3.3 Future Detectors

#### Enhanced LIGO (LIGO II)

Again, the initial LIGO, known as LIGO I, reached its designed sensitivity during the S5 run. Therefore the plan for the improvement in the sensitivity is to stop the observation for about 2.5 years after S5 (starting from October 2007) and make some hardware improvements that will bring a factor of 2-3 improvement in the sensitivity and 5-10 in the event rate compare to the LIGO I. In this Enhanced LIGO, known as LIGO II, the major changes will be increasing the input laser power from 10W to 180W which gives an input power into power recycling cavity of about 125W instead of 6W used in LIGO I, and increasing the power recycling factor from 50 to 80. The mirrors will also be changed to have 28cm



diameter and 30kg mass instead of 25cm and 10kg as used for LIGO I. The plan is also to use a signal recycling mirror at the detection port (as being used in GEO600). A better suspension system, similar to what is being used in GEO600, will also be used to lower the low-frequency cutoff from 40Hz to 10Hz [35, 36].

### Advanced LIGO

The experience with enhanced LIGO should help with the development of Advanced LIGO. The hardware should be ready for installation at the end of 2010 and launched in 2014 [36]. The improvement in the sensitivity in Advanced LIGO will be about a factor of 10 compared to LIGO I.

The planned test mass optics for Advanced LIGO are about 34cm in diameter to reduce thermal noise contributions and more massive (40kg) to keep the radiation pressure noise to a level comparable to the suspension thermal noise. The laser power in the Fabry-Perot cavities will be on the order of 1MW.

### Space Based Detectors

Like ground-based interferometric GW detectors, a space-based detector (such as LISA) uses the laser interferometric technique. The Laser Interferometer Space Antenna (LISA) is a joint project between European Space Agency (ESA) and NASA. It consists of three spacecraft which each of these spacecraft will carry two telescopes with associated lasers and optical systems. Pointing in directions separated by 60 degrees, the telescopes in each spacecraft will exchange laser beams with the other two spacecrafts, located at the other two corners of a 5 million kilometers equal-sided triangle [37]. Each of the three LISA spacecraft has a shape of cylinder with 1.8 meters diameter and 0.48 meters high. With this design the spacecraft size and shape are optimized to contain the instrument's optical assemblies.

According to the current concept, the three identical LISA spacecrafts will go into space together on a single rocket, and will then independently take up their long-term orbits around the Sun. The orbits will be similar to that of the Earth, but will trail behind our planet at distances of around 50 million kilometers.

It will take one year for the three spacecraft to reach their final position and to start the actual mission. The LISA triangle will be angled at 60 degrees to the plane of Earth's orbit and revolving with the Earth around the Sun. These particular heliocentric orbits for the three spacecrafts were chosen so that the triangular formation is maintained throughout the year, with the triangle appearing to rotate about the center of the formation once per year. The relative movement of the three spacecraft will help to detect the direction of each source and to reveal the nature of the gravitational waves.

Each of these three spacecrafts will play the role of both the laser source and the detection plate. Therefore, in the end, LISA will consist of three independent laser interferometric detectors.

The laser light in the LISA, instead of being reflected using mirrors, is transposed back to the central spacecraft. Also LISA does not use any beam splitter to split laser beam to be sent toward the other test masses at the other corners of triangle. One spacecraft, which acts as a laser light source in this case, transmits the laser beam to the other two spacecrafts by keeping part of the light as reference inside itself. Since the other two spacecrafts are far from the main one, the laser beam reaches them is very weak. Therefore, the two corner spacecrafts rather than sending back the same beam to the main spacecraft, they transmit a new laser light with the same phase of the incoming beam. At the main spacecraft, the signals from the two other spacecrafts are compared to measure the variation in distance between them and the main spacecraft [38].

Apart from the complexity of aiming the laser beams from one small spacecraft to another across 5 million kilometers of space, LISA has to deal with other forces besides gravitational waves which will alter the separation of the spacecrafts - the Solar wind, magnetic fields, thermal effect and the pressure of sunlight, for example. This is due to another difference between LISA and a ground-based laser interferometric detector that the test masses are not suspended by wires, instead they are freely floating inside the spacecraft. This mechanism in fact tries to protect the main part from the movement due to some external forces other than GWs, such as solar winds etc. In the case of un-wanted external forces, the outer shell of spacecraft changes its position whereas the inner part (the main apparatus for interferometry) remains steady. At the next, by using some electrical mechanism, the outer shell will be forced back to its original position. Therefore, the main part will be always protected from the movement due to other forces than gravitational wave's effect.

The distance between the spacecraft determines the frequency range in which LISA can make observations;

it has been carefully chosen to allow observation of most of the interesting sources of gravitational radiation, namely massive black holes and compact binary stars. LISA can cover the frequency range of  $10^{-5} - 10^{-1}$ Hz. The optimal LISA sensitivity, for a one year of observation with a signal to noise ratio of 5, will be about  $10^{-23}$ . LISA's best sensitivity is at a frequency of  $10^{-2}$ Hz [39].





# Chapter 2

## Basic Concepts

In this chapter, we review the standard basic concepts of data analysis that have been used in the course of research for this thesis. It cannot be our aim to provide an exhaustive account of this complex subject; instead, we give a highly selective presentation of those aspects that are most important for the research in question.

In section 2.1, we will give a brief account of the signal phase model for a gravitational wave reaching any ground-based interferometric detector, considering in 2.2 the variation in the detector's response due to the Earth's orbital and rotational motion. In 2.3, we introduce the basic concepts of statistics that are crucial for GW data analysis, establishing our notation for what is to follow. Finally, in 2.4, we explain in more detail the key concept for the research to be described in the later chapters, namely the *maximum likelihood function* or  $\mathcal{F}$ -statistics. The chapter is rounded out by very brief accounts of the Short Fourier Transform.

### 2.1 Signal phase model

In this thesis, the only kind of signal under consideration is the continuous signal from isolated neutron stars. For further simplification, let us assume that the source is at rest relative to the detector, so that proper motion effects do not arise. This poses no restriction on our analysis – even when there is proper motion, the resulting Doppler shift can be absorbed into an overall Doppler shift, and thus is effectively unobservable.

For data analysis making use of time series data, precise knowledge of the evolution of the signal phase is very important. However, in what follows, we shall work in the frequency domain; therefore, we will focus on the signal's power (the square of its amplitude). The only way in which the evolution of the phase contributes is to the frequency evolution.

In general, the intrinsic gravitational wave frequency  $f(t)$  of a continuous signal from a neutron star, as received in the Solar System Barycenter reference frame (SSB), will change but slowly with time (with the exception of possible glitches or other instabilities). Hence, we can expand the time-dependence of the phase as a Taylor series in the SSB, as follow [44]

$$\begin{aligned}\Phi(t_{\text{ssb}}) &= \Phi_0 + 2\pi \left[ f_0 \Delta t_{\text{ssb}} + \frac{1}{2} \dot{f}_0 (\Delta t_{\text{ssb}})^2 + \frac{1}{6} \ddot{f}_0 (\Delta t_{\text{ssb}})^3 + \dots \right] \\ &= \Phi_0 + 2\pi \sum_{k=0}^s \frac{f^{(k)}(t_{\text{ssb}}^{(0)})}{(k+1)!} (\Delta t_{\text{ssb}})^{k+1} \\ &= \Phi_0 + \phi(t),\end{aligned}\tag{2.1}$$

with  $\Phi_0$  the initial phase; generally, we only need to take this expansion up to some specific order  $s$ . Taking the time derivative of this expression, the frequency evolution has the form

$$f(t_{\text{ssb}}) = \sum_{k=0}^s \frac{f^{(k)}(t_{\text{ssb}}^{(0)})}{k!} (\Delta t_{\text{ssb}})^k,\tag{2.2}$$

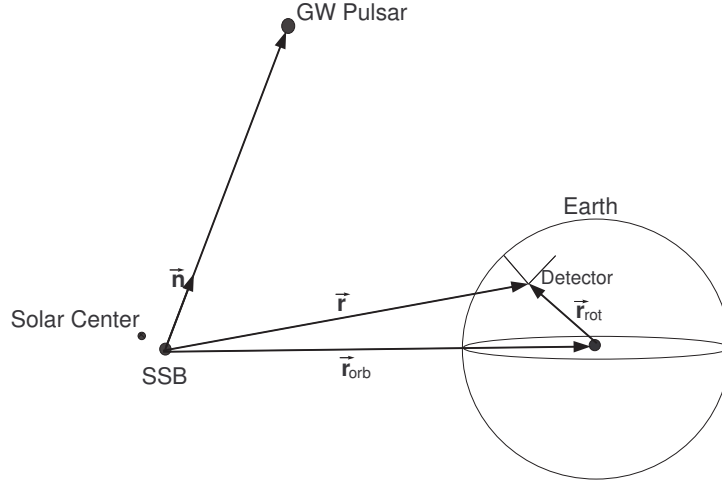


Figure 2.1: Diagram of the SSB and the detector's position and the vectors used to indicate them. The SSB is usually being taken by ignoring any massive objects near the Sun.

where  $t_{\text{ssb}}$  is the time measured in the SSB frame,  $\Delta t_{\text{ssb}} \equiv t_{\text{ssb}} - t_{\text{ssb}}^{(0)}$ , while  $t_{\text{ssb}}^{(0)}$  is the time at which the observation starts;  $f^{(k)}(t_{\text{ssb}}^{(0)})$  is the  $k$ th time derivative of the frequency at the initial time,

$$f^{(k)} = \frac{\partial^k f}{\partial t_{\text{ssb}}^k} \Big|_{t_{\text{ssb}}^{(0)}}. \quad (2.3)$$

The  $s$  terms  $f^{(k)}$  are known as *spin-down parameters*; the name reflects the fact that in the typical case (no accretion), these terms will reflect how the neutron star loses its energy as it radiates away gravitational waves; in consequence, it spins down.

As we change from the SSB frame to an Earth-centered frame, we need to take into account the Doppler shift resulting from the Earth's orbital motion and rotation. Conversely, any signal received by our ground-based detector will need to be demodulated. To this end, let  $\mathbf{n}$  be the unit vector pointing from the SSB toward the pulsar,  $\mathbf{r}(t)$  the position of the detector in the SSB frame (*i.e.*  $\mathbf{r}(t) = \mathbf{r}_{\text{orb}}(t) + \mathbf{r}_{\text{rot}}(t)$ , where  $\mathbf{r}_{\text{orb}}(t)$  is the vector from SSB to the Earth's center and  $\mathbf{r}_{\text{rot}}(t)$  the vector from that center toward the detector). Furthermore, let  $\mathbf{v}(t)$  be detector's velocity, more concretely: its change in position over time  $t$ , where time is measured in the detector frame (see Fig. 2.1).

A precise description would need to take into account a number of relativistic effects, as well, notably the *Shapiro delay* (a time delay that is due to propagation past the Sun) and the *Einstein delay* (a direct consequence of the gravitational redshift). For a full description of the phase evolution including relativistic effects, the reader is referred to Appendix A of Jaranowski, Królak, and Schutz (JKS) [44]. These effects are effectively included in our actual analysis of GW data, but for the exposition here they are a needless complication, and we will henceforth ignore them. With this simplification, a wave reaching the SSB at time  $t_{\text{ssb}}$  will reach the detector at time

$$t = t_{\text{ssb}} - \frac{\mathbf{r}(t) \cdot \mathbf{n}}{c}. \quad (2.4)$$

Calculating  $\Delta t_{\text{ssb}}$  by using Eq. (2.4) in Eq. (2.1), and ignoring higher order terms, we obtain to a good approximation the instantaneous frequency of the signal as seen by the detector. It turns out to be given by the familiar Doppler shift expression

$$f(t) \cong f(t_{\text{ssb}}) \left( 1 + \frac{\mathbf{v}(t) \cdot \mathbf{n}}{c} \right), \quad (2.5)$$

where  $f(t_{\text{ssb}})$  is the instantaneous frequency of the signal in the SSB frame as defined in Eq. (2.2).

Eq. (2.5) describes the approximate frequency modulation of the received signal. The received signal is also amplitude modulated – the antenna pattern changes over time as the detector is carried along by the Earth's rotation. This effect will be discussed in the following section.

## 2.2 Signal at the detector

The signal received at the detector is both phase modulated (cf. the previous section) and amplitude modulated. Both effects are due to the Earth's motion and rotation relative to the SSB. Amplitude modulation is due to the fact that, as the Earth rotates, the detector's orientation relative to the incoming waveform changes; in consequence, so does the detector's response. In order to write down the form of the signal as received by the detector, we will need to take this effect into account.

To this end, let  $x(t)$  be the output of a given detector. In the absence of a GW signal,  $x(t)$  is just noise  $n(t)$ . We will assume  $n(t)$  to be a stationary, Gaussian stochastic process with zero mean (this is a good approximation for the short stretches of real GW data). In the presence of a signal  $h(t)$ ,

$$x(t) = n(t) + h(t). \quad (2.6)$$

The signal  $h(t)$  is a linear combination of the amplitudes for the two independent GW polarizations,  $h_+(t)$  and  $h_\times(t)$  [44]:

$$h(t) = F_+(\mathbf{n}, \psi, t)h_+(t) + F_\times(\mathbf{n}, \psi, t)h_\times(t), \quad (2.7)$$

where  $\psi$  is the signal's polarization angle (see Fig. 2.2), and  $F_{+,\times}$  are the antenna pattern functions (also known as the *detector beam pattern function*). Because of the diurnal motion of the Earth, the beam patterns  $F_+$  and  $F_\times$  are periodic functions of time with a period equal to one sidereal day; they can be expressed as [44]

$$F_+(t) = a(t) \cos 2\psi + b(t) \sin 2\psi, \quad (2.8)$$

$$F_\times(t) = b(t) \cos 2\psi - a(t) \sin 2\psi, \quad (2.9)$$

where  $a(t)$  and  $b(t)$  are functions of right ascension  $\alpha$  and declination  $\delta$ ; they are independent of  $\psi$  and can be expressed as [44]

$$\begin{aligned} a(t) = & \frac{1}{16} \sin 2\gamma (3 - \cos 2\lambda) (3 - \cos 2\delta) \cos[2(\alpha - \phi_r - \Omega_r t)] \\ & - \frac{1}{4} \cos 2\gamma \sin \lambda (3 - \cos 2\delta) \sin[2(\alpha - \phi_r - \Omega_r t)] \\ & + \frac{1}{4} \sin 2\gamma \sin 2\lambda \sin 2\delta \cos[\alpha - \phi_r - \Omega_r t] \\ & - \frac{1}{2} \cos 2\gamma \cos \lambda \sin 2\delta \sin[\alpha - \phi_r - \Omega_r t] \\ & + \frac{3}{4} \sin 2\gamma \cos^2 \lambda \cos^2 \delta, \end{aligned} \quad (2.10)$$

$$\begin{aligned} b(t) = & \cos 2\gamma \sin \lambda \sin \delta \cos[2(\alpha - \phi_r - \Omega_r t)] \\ & + \frac{1}{4} \sin 2\gamma (3 - \cos 2\lambda) \sin \delta \sin[2(\alpha - \phi_r - \Omega_r t)] \\ & + \cos 2\gamma \cos \lambda \cos \delta \cos[\alpha - \phi_r - \Omega_r t] \\ & + \frac{1}{2} \sin 2\gamma \sin 2\lambda \cos \delta \sin[\alpha - \phi_r - \Omega_r t]. \end{aligned} \quad (2.11)$$

In these expressions,  $\lambda$  is the latitude of the detector site,  $\Omega_r$  is the angular velocity of the Earth's rotation; the sum  $\phi_r + \Omega_r t$  denotes the local sidereal time of the detector's site, *i.e.* the angle between the local meridian and the vernal point. The variable  $\gamma$  specifies the orientation of the detector's arms with respect to local geographical directions:  $\gamma$  is measured counter-clockwise from East; its value indicates the direction that exactly bisects the two interferometer arms (which are taken to be at an angle of  $\pi/2$ ).

The two gravitational wave polarization components of the signal can be written as

$$h_+(t) = A_+ \cos \Phi(t), \quad (2.12)$$

$$h_\times(t) = A_\times \sin \Phi(t), \quad (2.13)$$

where  $\Phi(t)$  is the phase of the signal at SSB (cf. Eq. 2.1) and

$$A_+ = \frac{1}{2} h_0 (1 + \cos^2 \iota), \quad (2.14)$$

$$A_\times = h_0 \cos \iota, \quad (2.15)$$

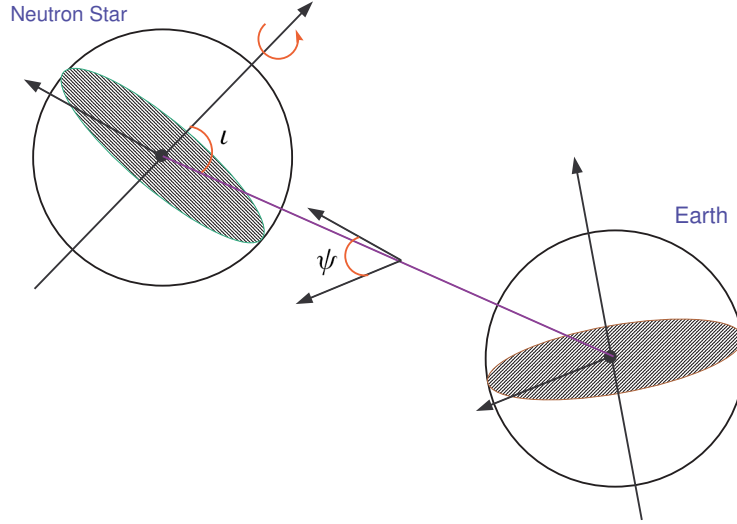


Figure 2.2: The orientation of neutron star emitting the gravitational waves and its position with respect to the Earth. The plane in which the angle of  $\psi$  is located is perpendicular to the direction of wave propagation.

where  $h_0$  (the intrinsic wave amplitude at the source) can be written in terms of the triaxial ellipticity  $\varepsilon = (I_{xx} - I_{yy})/I_{zz}$  and the distance  $d$  from the source as

$$h_0 = \frac{4\pi^2 G}{c^4} \frac{\varepsilon I_{zz} f^2}{d}. \quad (2.16)$$

Putting everything together, the signal  $h(t)$  depends on 8 parameters which can be divided into two sets: on the one hand, we have the intrinsic amplitude  $h_0$ , the wave polarization angle  $\psi$ , the inclination angle  $\iota$  (see Fig. 2.2), and the initial phase  $\Phi_0$ . We will call these the

$$\text{amplitude parameters} \rightarrow h_0, \psi, \iota, \Phi_0.$$

On the other hand, there are quantities like the source's location in the sky, given by  $\alpha$  and  $\delta$ , the wave frequency  $f$  and the spin-down parameters  $f^{(k)}$ , which we will call the

$$\text{Doppler parameters} \rightarrow \alpha, \delta, f, f^{(k)}.$$

The gravitational wave signal  $h(t)$  depends nonlinearly on the amplitude parameters, but, crucially, one can make a simple change of variables – introducing new variables ( $A^1, A^2, A^3, A^4$ ) – such that dependency of  $h(t)$  is linear in the  $A^a$  [52]:

$$h(t) = \sum_{a=1}^4 A^a h_a(t), \quad (2.17)$$

where the detector-dependent wave components  $h_a(t) = h_a(\alpha, \delta, f, f^{(k)}; t)$  are given as

$$h_1(t) = a(t) \cos \phi(t), \quad (2.18)$$

$$h_2(t) = b(t) \cos \phi(t), \quad (2.19)$$

$$h_3(t) = a(t) \sin \phi(t), \quad (2.20)$$

$$h_4(t) = b(t) \sin \phi(t). \quad (2.21)$$

For  $\phi(t)$ , we use the expression given in Eq. (2.1), and the constant (in time) amplitudes  $A^a = A^a(h_0, \psi, \iota, \Phi_0)$  are [52]

$$A^1 = A_+ \cos \Phi_0 \cos 2\psi - A_\times \sin \Phi_0 \sin 2\psi, \quad (2.22)$$

$$A^2 = A_+ \cos \Phi_0 \sin 2\psi + A_\times \sin \Phi_0 \cos 2\psi, \quad (2.23)$$

$$A^3 = -A_+ \sin \Phi_0 \cos 2\psi - A_\times \cos \Phi_0 \sin 2\psi, \quad (2.24)$$

$$A^4 = -A_+ \sin \Phi_0 \sin 2\psi + A_\times \cos \Phi_0 \cos 2\psi. \quad (2.25)$$

## 2.3 Basic statistics

Given the sensitivity of current ground-based detectors, any continuous-wave signal we can realistically expect will be at a level or below the background noise. Hence, any data analysis of detector data will need to rely on statistical techniques to separate the signal from the noise. In this section, we introduce some of the fundamental statistical concepts that are needed to understand GW data analysis methods, at a level that can serve to refresh readers' memories of fundamental statistics, and at the same time establish the notation and conventions used in subsequent chapters.

Assume that we are given a random process, that is, a set of random variables. These variables could be defined as the time series of a single random variable, so we would be studying the random behavior of a single object over a given interval of time. Alternatively, we could be looking at an *ensemble of random variables* – a set of different variables with common properties (such as a common probability density function), evaluated at a single moment in time.

We can define certain statistical quantities to characterize our random process. The first of those is the *average* or *mean value*; for the time series, it is defined as

$$\bar{x} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt, \quad (2.26)$$

while for an ensemble, it is

$$\langle x(t) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N x_i(t) = \int_{-\infty}^{\infty} x(t) P(x, t) dx; \quad (2.27)$$

here,  $P(x, t)$  is the Probability Density Function (PDF) of the random process in question.

In many interesting cases, the time series approach and the ensemble approach are actually equivalent. If, for example, the time average and the ensemble average are the same for some specific statistical phenomenon,

$$\bar{x} = \langle x(t) \rangle, \quad (2.28)$$

the phenomenon is called *ergodic* (in mean). Such is assumed to be the case for the time series considered in this thesis; in consequence, we shall from now on focus exclusively on the statistics of ensembles.

Let us next introduce a number of further statistical properties for ensembles. The *mean-square* is defined as follows:

$$\langle x^2(t) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N x_i^2(t) = \int_{-\infty}^{\infty} x^2(t) P(x, t) dx. \quad (2.29)$$

The *variance* is a measure of how closely the random values are clustered around the mean:

$$\sigma^2 = \langle x^2(t) \rangle - \langle x(t) \rangle^2. \quad (2.30)$$

The *auto-correlation function* (also called the *covariance*) is the correlation between the value of  $x$  at time  $t_1$  and  $t_2$ :

$$\begin{aligned} C_x(t_1, t_2) &= \langle (x(t_1) - \bar{x}(t_1))(x(t_2) - \bar{x}(t_2)) \rangle \\ &= \lim_{T \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N (x_i(t_1) - \bar{x}_i(t_1))(x_i(t_2) - \bar{x}_i(t_2)) \\ &= \int_{-\infty}^{\infty} (x_1 - \bar{x}_1)(x_2 - \bar{x}_2) P(x_1, x_2; t_1, t_2) dx_1 dx_2, \end{aligned} \quad (2.31)$$

where  $\bar{x}$  denotes the average of  $x$ .

For an ensemble with infinitely many members, the mean will tend toward the probability distribution's *expectation value*. For ensembles of large, yet finite size, the two are approximately the same; in many references, the same notation  $E[\cdot]$  is used for both. We will follow this convention and express the averages of a given quantity  $y(t)$  as  $E[y(t)]$ ; for the special case of the mean of our random variable  $x(t)$ , we will in addition use the notation

$$\mu_x = E[x]. \quad (2.32)$$

With this new notation, the variance and the covariance function can be written as

$$\sigma^2 = E[(x(t) - \mu_x(t))^2], \quad (2.33)$$

$$C_x(t_1, t_2) = E[(x(t_1) - \mu_x(t_1))(x(t_2) - \mu_x(t_2))]. \quad (2.34)$$

Having defined basic statistical quantities, we next need to look at special classes of random processes. A *stationary random process* is an ensemble of random processes for which the time dependence of  $P(x_1, x_2; t_1, t_2)$  is a function of the difference  $\tau = t_2 - t_1$  only. More generally, for a *Wide-Sense Stationary (WSS)* process, the mean  $\mu_x(t)$  does not change over time, and the covariance function ( $E[x(t_1)x(t_2)]$ ) depends only on the time difference  $\tau = t_2 - t_1$ . For simplicity, let us assume that our random process has a mean value of zero, then hereafter we replace  $\mu = 0$  in all of our equations. According to Eq. (2.34), the covariance function of a WSS random signal is

$$C_x(t_1, t_2) = C_x(\tau) = E[x(t)x(t + \tau)]. \quad (2.35)$$

Eq. (2.35) tells us that  $C_x(\tau)$  is a symmetric function of  $\tau$ , in other words  $C_x(\tau) = C_x(-\tau)$ , which can be seen from

$$\begin{aligned} C_x(\tau) &= E[x(t)x(t + \tau)] \\ &= E[x(s - \tau)x(s)], \quad t + \tau = s \\ &= E[x(s)x(s - \tau)], \\ &= E[x(t)x(t - \tau)], \quad \text{replacing } s \text{ by } t \\ &= C_x(-\tau); \end{aligned} \quad (2.36)$$

all the steps in this sequence are permissible because the averages are independent of  $s$  as well as of  $t$ .

A key property of the covariance function of a WSS signal is that, if we take  $\tau = 0$ , the result will be the variance

$$C_x(0) = E[x^2(t)] = \sigma^2. \quad (2.37)$$

Given two random processes  $x(t)$  and  $y(t)$  we can define their *cross correlation* as

$$C_{xy}(\tau) \equiv E[x(t)y(t + \tau)] \quad (2.38)$$

(in perfect analogy with the correlation function  $C_x(\tau)$ ). Note that, just like the correlation function, the cross correlation satisfies

$$C_{xy}(-\tau) = C_{yx}(\tau). \quad (2.39)$$

The cross correlation of a random process with itself is equal to its autocorrelation  $C_{yy}(\tau) = C_y(\tau)$ . In fact, given  $x(t)$  and  $y(t)$ , we can define a whole matrix [45]

$$\begin{vmatrix} C_{xx}(\tau) & C_{xy}(\tau) \\ C_{yx}(\tau) & C_{yy}(\tau) \end{vmatrix} = \begin{vmatrix} C_x(\tau) & C_{xy}(\tau) \\ C_{xy}(\tau) & C_y(\tau) \end{vmatrix}, \quad (2.40)$$

which is the correlation matrix for the 2-dimensional random process  $\{x(t), y(t)\}$ .

## Fourier Transformations

The *Fourier Transformation (FT)* of a time series random process,  $x(t)$ , is defined as

$$\tilde{x}(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft} dt. \quad (2.41)$$

Whenever the Fourier transform exists, its inverse is given by

$$x(t) = \int_{-\infty}^{\infty} \tilde{x}(f)e^{i2\pi ft} df. \quad (2.42)$$

For a discrete time series with  $N$  entries, the Discrete Fourier Transform (DFT) is defined as

$$\tilde{x}_k = \sum_{j=0}^{N-1} x_j e^{-i2\pi jk/N}; \quad (2.43)$$

its inverse has the form

$$x_j = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{x}_k e^{i2\pi jk/N}. \quad (2.44)$$

Note that conventions differ where Fourier transforms are concerned. In some references, the exponential in the definition of the transform has a plus sign; in the discrete version, the factor  $1/N$  sometimes occurs in the definition of the transformation (instead of the definition of its inverse); in yet other cases, both definitions have a factor  $1/\sqrt{N}$ .

Unfortunately, the usual definition of the Fourier transform is of limited value when it comes to random processes. In general, a random process is assumed to go on forever ( $T \rightarrow \infty$ ); in consequence, its Fourier transform  $\tilde{x}$  (using the above definition) does not converge!

However, with an additional assumption (a trick, if you like), we can make the Fourier transform well-defined. Let us define a new random process as

$$x_T(t) \equiv \begin{cases} x(t) & -T/2 < t < T/2 \\ 0 & \text{otherwise.} \end{cases} \quad (2.45)$$

With this assumption, the Fourier transform will converge, and in fact it will be finite in any range of intervals. The fact that we can safely make this assumption is a consequence of *Parseval's theorem*, which states that

$$\int_{-\infty}^{\infty} x(t)y^*(t) dt = \int_{-\infty}^{\infty} \tilde{x}(f)\tilde{y}^*(f) df, \quad (2.46)$$

where the asterisk (\*) denotes *complex conjugation*. Setting  $y(t) = x(t)$  in this equation, it can be derived that the integral (or sum) of the square of a function is equal to the integral (or sum) of the square of its Fourier transform:

$$\int_{-\infty}^{\infty} [x(t)]^2 dt = \int_{-\infty}^{\infty} |\tilde{x}(f)|^2 df. \quad (2.47)$$

This has a straightforward physical interpretation: *The total energy, as computed in the time domain, is equal to the total energy as computed in the frequency domain.*

The Fourier transform of our modified function  $x_T(t)$  (as defined in Eq. (2.45)) is, by Eq. (2.47),

$$\begin{aligned} \int_{-T/2}^{T/2} [x(t)]^2 dt &= \int_{-\infty}^{\infty} [x_T(t)]^2 dt \\ &= \int_{-\infty}^{\infty} |\tilde{x}_T(f)|^2 df \\ &= 2 \int_0^{\infty} |\tilde{x}_T(f)|^2 df, \end{aligned} \quad (2.48)$$

where for last step we have used the fact that  $x_T(t)$  is real (so  $\tilde{x}_T^*(f) = \tilde{x}_T(-f)$ ) and, consequently, the integral of  $|\tilde{x}_T(f)|^2$  in the negative range is equal to that in the positive range. Consequently, the limit

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} [x(t)]^2 dt = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} |\tilde{x}(t)|^2 df \quad (2.49)$$

is convergent.

## Power Spectral Density

The *Wiener-Khintchine Theorem* (also known as the Wiener-Khinchin theorem and sometimes as the Khinchin-Kolmogorov theorem) states that the Power Spectral Density (PSD) of a Wide-Sense-Stationary (WSS) random process is the Fourier transform of the corresponding covariance function

$$S_x(f) = \int_{-\infty}^{\infty} C_x(\tau) e^{-2\pi i f \tau} d\tau, \quad (2.50)$$

and that its inverse is

$$C_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{2\pi i f \tau} df, \quad (2.51)$$

where  $C_x(\tau)$  is the covariance function defined in Eq. (2.35), and  $S_x(f)$  is the double-sided power spectral density of the function  $x(t)$ . The Spectral density is an even function of frequency, so in particular  $S_x(f) = S_x(-f)$ . The PSD is not a random variable; it follows directly from the expectation value of the covariance function, which depends only on the probability distribution.

When  $df$  is small, the total average power of the signal in the frequency range  $(f - df/2, f + df/2)$  is  $S_x(f) df$  with the  $S_x(f)$  having the dimension of *inverse frequency* (1/Hz). If the PSD is constant within a given frequency band, the corresponding process is called a *white random process* (in that frequency band); if the PSD instead varies within that frequency band, the process is called a *colored random process*.

An important corollary of the Wiener-Khintchine theorem is the relation between PSD and the Fourier transform of a finite segment (imposing the assumption given in Eq. 2.45) of a WSS signal with zero mean, namely

$$\text{PSD}(f) = \lim_{T \rightarrow \infty} \frac{1}{T} E[|\tilde{x}(f)|^2]. \quad (2.52)$$

For an outline of the proof, the reader is referred to [46]. For a WSS discrete-time random signal (again with the assumption of Eq. 2.45) with zero mean, the relation between PSD and the Fourier transform can be expressed as [47]

$$\text{PSD}(f) = \lim_{N \rightarrow \infty} \frac{1}{N} E[|\tilde{x}(f)|^2]. \quad (2.53)$$

Eq. (2.53) is of paramount importance for data analysis. For example, it enables us to take a short segment of data, compute the Fourier transform, and (supposing that the data contains noise only) use the result to estimate the PSD, *noise floor*.

The main properties of the power spectral density of a WSS signal are

- Reality, and symmetry in  $f$  (as the covariance function is real and symmetric in  $\tau$ )
- For all frequencies, the PSD is non-negative.
- Its integral over all frequencies is proportional to the signal's variance,

$$\int_{-\infty}^{\infty} S_x(f) df = C_x(0) = \sigma^2 \quad (2.54)$$

(which can be seen by substituting  $\tau = 0$  in Eq. (2.51)).

## Gaussian processes

A discretely sampled time series data  $\{x_i : i = 1, \dots, N\}$  which characterized by

$$x_i = x(t_i) \quad (2.55)$$

$$t_i - t_j = (i - j)\Delta t \quad (2.56)$$

$$\Delta t = \frac{T}{N - 1}, \quad (2.57)$$

and zero mean is Gaussian if its probability distribution function (pdf) follows the form [51]

$$P(x) = \frac{\exp \left[ -\frac{1}{2} \sum_{j,k=1}^N C_{jk}^{-1} x_j x_k \right]}{(2\pi)^{N/2} (\det C)^{1/2}}, \quad (2.58)$$

where

$$C = \begin{pmatrix} C_{11} & \cdots & C_{1N} \\ \vdots & \ddots & \vdots \\ C_{N1} & \cdots & C_{NN} \end{pmatrix}, \quad (2.59)$$

with  $C_{ij} \equiv C(t_i, t_j) = C[(i - j)\Delta t]$  being the covariance matrix and  $C_{jk}^{-1}$  its inverse:

$$\delta_{jk} \equiv \sum_l C_{jl} C_{lk}^{-1}. \quad (2.60)$$



Note that the normalization constant in the denominator of the Eq. (2.58) is independent of  $x_i$ . To evaluate Eq. (2.58) in the continuum limit, first note that [51]

$$\delta(t_j - t_k) = \lim_{\substack{\Delta t \rightarrow 0 \\ T \rightarrow \infty}} \frac{1}{\Delta t} \delta_{jk}. \quad (2.61)$$

Consequently [51],

$$\begin{aligned} e^{-2\pi i f t_k} &= \sum_j e^{-2\pi i f t_j} \delta_{jk} \\ &= \lim_{\substack{\Delta t \rightarrow 0 \\ T \rightarrow \infty}} \frac{1}{\Delta t^2} \sum_j \Delta t e^{-2\pi i f t_j} \sum_l \Delta t C_n(t_j - t_l) C^{-1}(t_l, t_k) \\ &= \lim_{\substack{\Delta t \rightarrow 0 \\ T \rightarrow \infty}} \frac{1}{\Delta t^2} \int_{-\infty}^{\infty} dt_j e^{-2\pi i f t_j} \int_{-\infty}^{\infty} dt_l C_n(t_j - t_l) C^{-1}(t_l, t_k) \\ &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t^2} \int_{-\infty}^{\infty} dt_l e^{-2\pi i f t_l} C^{-1}(t_l, t_k) \int_{-\infty}^{\infty} d\tau e^{-2\pi i f \tau} C_n(\tau) \end{aligned} \quad (2.62)$$

$$= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t^2} \widetilde{C}^{-1}(f, t_k) \frac{1}{2} S_n(f). \quad (2.63)$$

This gives

$$\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t^2} \widetilde{C}^{-1}(f, t_k) = \frac{2 e^{-2\pi i f t_k}}{S_n(f)}. \quad (2.64)$$

For the transition from Eq. (2.62) to Eq. (2.63), we use the Wiener-Khintchine theorem [cf. Eq. (2.50)], which relates the PSD,  $S_n(f)$ , and the correlation function  $C_n(\tau)$ ; also, we use the Fourier transform of the inverse of the correlation function, defined as

$$\widetilde{C}^{-1}(f, t_k) \equiv \int_{-\infty}^{\infty} dt C^{-1}(t, t_k) e^{-2\pi i f t}. \quad (2.65)$$

Taking  $\widetilde{C}^{-1}$  of Eq. (2.65), and using Eq. (2.64) and Parseval's theorem, Eq. (2.46), we can evaluate the continuum limit of the argument of the exponential in Eq. (2.58) [51]:

$$\begin{aligned} \lim_{\substack{\Delta t \rightarrow 0 \\ T \rightarrow \infty}} \sum_{j,k=1}^N C_{jk}^{-1} x_j x_k &= \lim_{\substack{\Delta t \rightarrow 0 \\ T \rightarrow \infty}} \frac{1}{\Delta t^2} \sum_{j,k=1}^N \Delta t^2 C^{-1}(t_j, t_k) x(t_j) x(t_k) \\ &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dt_j dt_k C^{-1}(t_j, t_k) x(t_j) x(t_k) \\ &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} df dt_k \widetilde{C}^{-1}(f, t_k) \tilde{x}^*(f) x(t_k) \\ &= 2 \int_{-\infty}^{\infty} df \frac{\tilde{x}^*(f)}{S_n(|f|)} \int_{-\infty}^{\infty} dt_k e^{-2\pi i f t_k} x(t_k) \\ &= 2 \int_{-\infty}^{\infty} df \frac{\tilde{x}(f) \tilde{x}^*(f)}{S_n(|f|)}. \end{aligned} \quad (2.66)$$

Since the detector's output  $x(t)$  is real, we must have  $\tilde{x}^*(f) = \tilde{x}(-f)$ . Hence, we can define the following symmetric inner product of  $(x|y)$

$$\begin{aligned} (x|y) &= 2 \int_{-\infty}^{\infty} \frac{\tilde{x}(f) \tilde{y}^*(f)}{S_n(f)} df \\ &= 2 \int_{-\frac{f_s}{2}}^{\frac{f_s}{2}} \frac{\tilde{x}(f) \tilde{y}^*(f)}{S_n(f)} df \\ &= 4\Re \int_0^{\frac{f_s}{2}} \frac{\tilde{x}(f) \tilde{y}^*(f)}{S_n(f)} df, \end{aligned} \quad (2.67)$$

where  $f_s$  is the bandwidth of the signal that is searched over.

Eqs. (2.67), (2.66) and (2.58) imply that

$$P(x|y) \propto \exp\left(-\frac{1}{2}(x-y|x-y)\right). \quad (2.68)$$

The main reason that it is convenient to work in the frequency domain is that for stationary noise all of its Fourier component are independent; *i.e.*

$$\langle \tilde{x}(f_1)\tilde{x}(f_2) \rangle \propto \delta(f_1 - f_2), \quad (2.69)$$

and therefore for stationary and Gaussian noise

$$P(\tilde{x}(f_1)) \propto e^{-|\tilde{x}(f_1)|^2/2S_n(f_1)}, \quad (2.70)$$

*i.e.*, each Fourier component has a Gaussian probability distribution function.

## 2.4 Maximum likelihood function ( $\mathcal{F}$ -statistic)

The basic problem in GW detection is to identify a gravitational waveform in a noisy background. Because all data streams contain random noise, the data are just a series of random values and therefore the detection of a signal is always a decision based on probabilities. The aim of detection theory is therefore to assess this probability.

The basic idea behind the current methods of signal detection is that the presence of a signal will change the statistical characterization of the data  $x(t)$ , in particular its probability distribution function (pdf)  $P(x)$ . Recall that the pdf is defined so that the probability of a random variable  $x_i$  lies in an interval between  $x(t)$  and  $x(t) + dx$  is  $P(x)dx$ . Let us denote by  $P(x|0)$  the probability of a random process  $x(t)$  (representing our data) in the absence of any signal, and by  $P(x|h)$  the probability of that same process when a signal  $h(t)$  is present. Given a particular measurement  $x(t)$  obtained with our detector, is its probability distribution given by  $P(x|0)$  or  $P(x|h)$ ? In order to make that decision, we need to make a rule called a *statistical test*.

There are several approaches to finding an appropriate test, notably the Bayesian, Minimax and Neyman-Pearson approach (for an overview, we refer the reader to Jaranowski and Królak, [48] and the references listed therein). In the end, however, these three approaches lead to the same test, namely the *likelihood ratio test* [48, 49].

Among the three main approaches, the Neyman-Pearson approach is often used in the detection of gravitational waves [50]. This approach is based on maximizing the detection probability (equivalently minimizing the false dismissal rate) for fixed false alarm rate, where the detection probability is the probability that the random value of a process which contains the signal will pass our test, while the false alarm probability is the probability that data containing no signal will pass the test nonetheless. Mathematically, we can express these probabilities as [50]

$$P_D(R) = \int_R P(x|h)dx \quad \text{Detection Probability,} \quad (2.71)$$

$$P_F(R) = \int_R P(x|0)dx \quad \text{False Alarm Probability,} \quad (2.72)$$

where  $R$  is the detection region (to be determined).

The Likelihood ratio  $\Lambda$  is the ratio of the pdf when the signal is present to the pdf when it is absent:

$$\Lambda = \frac{P(x(t)|h(t))}{P(x(t)|0)}. \quad (2.73)$$

Remembering that  $x(t) = h(t) + n(t)$ , and with the assumption that the noise is a *zero-mean, stationary* and *Gaussian* random process, by using Eq. (2.68), we can write the likelihood ratio as

$$\begin{aligned} \Lambda &= \frac{P(x|h)}{P(x|0)} \\ &= \frac{\exp(-\frac{1}{2}(x-h|x-h))}{\exp(-\frac{1}{2}(x|x))} \\ &= \exp[(x|h) - \frac{1}{2}(h|h)]. \end{aligned} \quad (2.74)$$

This leads to the log of likelihood function as

$$\log \Lambda = (x|h) - \frac{1}{2}(h|h). \quad (2.75)$$

In the time domain, we can rewrite Eq. (2.67) as

$$(x|y) = \frac{2}{S_n(f_0)} \int_{t_0}^{t_p} x(t)y(t) dt, \quad (2.76)$$

where we have assumed that, over the bandwidth of the signal,  $S_n(f)$  is nearly constant and can be replaced by  $S_n(f_0)$ .

We can also rewrite the simple expression of Eq. (2.75) for the likelihood function in terms of the new variables  $A^a$  we had introduced in Eq. (2.17). The result is

$$\log \Lambda = (x|A^a h_a) - \frac{1}{2}(A^a h_a|A^b h_b). \quad (2.77)$$

Since the  $A^a$ s depend neither on the detector properties nor on the frequency or the time, we can take them out of the inner product and write the log of likelihood ratio as

$$\log \Lambda = A^a (x|h_a) - \frac{1}{2} A^a A^b (h_a|h_b). \quad (2.78)$$

Defining the new variables

$$H_a \equiv (x|h_a), \quad (2.79)$$

and

$$M_{ab} \equiv (h_a|h_b), \quad (2.80)$$

we have

$$\log \Lambda = A^a H_a - \frac{1}{2} A^a A^b M_{ab}. \quad (2.81)$$

The maximum detection probability follows from the maximization of the likelihood function: by maximizing the likelihood function with respect to the  $A^a$  (which, again, are independent of the detector), we have

$$\frac{\partial \log \Lambda}{\partial A^a} = 0. \quad (2.82)$$

This leads us to

$$H_a - A_{\text{MLE}}^b M_{ab} = 0, \quad (2.83)$$

and therefore

$$A_{\text{MLE}}^b = (M^{-1})^{ab} H_a. \quad (2.84)$$

The label *MLE* denotes the *Maximum Likelihood Estimator*; it corresponds to the values for the  $A^a$ s we calculate from our data by maximizing the likelihood ratio (so that, in practice, we are calculating  $A^a = E[A_{\text{MLE}}^a]$ ). By definition, the  $\mathcal{F}$ -Statistic is the *maximum of the logarithm of likelihood function*. Substituting Eq. (2.84) into Eq. (2.81), we have

$$\mathcal{F} \equiv \log \Lambda |_{\text{MLE}} = \frac{1}{2} H_a (M^{-1})^{ab} H_b. \quad (2.85)$$

This is the  $\mathcal{F}$ -Statistic; we will return to it (and introduce a generalization) in chapter 4, where we will also introduce a more practical version of this formula.

Again by writing the data as  $x(t) = n(t) + h(t)$ , where  $n(t)$  is the noise and  $h(t)$  is the signal and using Eq. (2.5) of Cutler-Schutz (CS) [52] indicating that  $\langle (x|n)(y|n) \rangle = (x|y)$  with this fact that  $\langle (h|n) \rangle = 0$ , we would have the following

$$\langle 2\mathcal{F} \rangle = 4 + (h|h), \quad (2.86)$$

which follows a  $\chi^2$  distribution with 4 degrees of freedom and non-centrality parameter  $\rho^2 \equiv (h|h)$ . The degrees of freedom come from the 4 *unknown parameters of pulsar* namely the **amplitude** ( $h_0$ ), **inclination angle** ( $\iota$ ), **polarization angle** ( $\psi$ ) and the **initial phase** ( $\Phi_0$ ). As was described in [52], even for a multi-IFO the number of freedom will remain unchanged, since we always have the same 4 unknown parameters in entire the search.

In cases where the amplitude modulation can be ignored (e.g., for short data segments,  $\ll 1$  day long, where the  $a(t)$  and  $b(t)$  can be approximated as constant), we see (in chapter 4) that  $\mathcal{F}$  is proportional to the demodulated Fourier transform which matches just the phase evolution as

$$\mathcal{F} \propto |\tilde{X}(f, \vec{\lambda})|^2, \quad (2.87)$$

where  $\vec{\lambda} := (\mathbf{n}, f^{(k)})$  and

$$\tilde{X}(f, \lambda) = \int_{t_0}^{t_p} x(t) e^{-i\phi(t; \lambda)} dt. \quad (2.88)$$

## 2.5 Short time-baseline Fourier Transform (SFT)

Since the output of a real detector is not stationary, we must break our data into short time-baseline chunks in order to avoid any non-stationarity in our data, and only then take the Fourier transform. The short segment of Fourier transformed data is called *Short time-baseline Fourier Transform*, or **SFT** for short. Longer chunks allow better initial frequency resolution; however one does not want  $S_n(f)$  to vary significantly over the chunk. Also, the length of each SFT should be such that the instantaneous frequency of a putative signal does not shift during that time by more than half a frequency bin, in other words: for each chunk, we should not have significant Doppler modulation effect in a single frequency bin. Following Schutz and Papa [53], this requirement fixes the maximum length for each short chunk at a given frequency as

$$T_{\text{chunk}} = 5.5 \times 10^3 \sqrt{\frac{300\text{Hz}}{f_0}}, \quad (2.89)$$

where  $f_0$  is the frequency of the search. For a search covering a whole range of frequencies, the above constraint must be calculated for the highest frequency within that range. The maximum stored frequency in each SFT of our data is 2048 Hz, which gives a maximum length of 2105 seconds for each short chunk. For their data analysis, the LSC (LIGO Scientific Collaboration) therefore uses 1800 seconds SFTs.

## Chapter 3

# Optimized Stack-Slide Hierarchical Search Strategy

*The work in this chapter was done in collaboration with Curt Cutler and Badri Krishnan[56].*

In analyzing data from Earth-based and space-based gravitational wave (GW) detectors, we will be computationally limited in performing certain types of searches, especially searches for long-lived signals described by several unknown parameters. For such signals, the number of templates required to discretely cover the parameter space (at useful resolution) typically increases rapidly as a function of the observation time. For a blind search (the search which none of the parameters such as frequency, frequency derivatives and sky location are previously known) the number of points in the parameter space for a full coherent search of duration  $T$  follow as

$$N_p \propto T^5. \quad (3.1)$$

For ground-based detectors, such as LIGO, a well-known example is the search for nearly periodic GWs from unknown, isolated, rapidly rotating neutron stars (NSs). We will refer to NSs that are continuously emitting GWs as “GW pulsars”. By “unknown”, we mean that the GW pulsar’s position on the sky, frequency, and frequency derivatives are all unknown, and so must be searched over. (The NS could be unknown either because it is electromagnetically inactive, or because its electromagnetic emission does not reach us - e.g., because we do not intersect its radio pulsar beam.)

Brady *et al.*[57] showed that straightforward matched filtering searches for unknown GW pulsars would be severely computationally limited; for example, searches for young, fast NSs (NSs with GW frequencies as high as 1 kHz and spin-down ages as short as 40 yr) would be limited to observation times of order *one day*. To address this problem, Brady and Creighton [58] (henceforth referred to as BC) were the first to consider hierarchical, multistage, semi-coherent searches for GW pulsars. Briefly, a semi-coherent search is one where a sequence of short data stretches are all coherently searched, using some technique akin to matched filtering, and then the resulting powers from the different stretches are summed. The method is only “semi-coherent” because powers are added instead of complex amplitudes; *i.e.*, information regarding the overall phase of the signal in different stretches is discarded. This allows one to use a much coarser grid on parameter space than would be required in a fully coherent search of the same data. BC developed a “Stack-Slide” method for summing the powers along different tracks in the time-frequency plane. The basic idea of their two-stage search is to identify a list of “candidates” (basically, promising-looking regions in parameter space) in the first stage, using some fraction of the available data, and then to “follow up” those candidates using more data in the second stage. In their scheme, both the first and second stages are semi-coherent.

In our work in this chapter we revisit the problem of constructing efficient, hierarchical searches for GW pulsars. We build on the BC treatment, but we also significantly generalize and otherwise improve upon their work. The most important improvements will be highlighted below.

- 1) BC consider a hierarchical search consisting of exactly two semi-coherent stages. In the present work, we consider a search consisting of an arbitrary number of semi-coherent stages, plus a fully coherent, “follow-up” stage (utilizing all the available data) to assess the significance of any surviving

candidates. The effect of the final, follow-up stage, is to ensure that the overall false alarm (FA) rate (fixed at exactly 1% by BC) is greatly reduced and, for all practical purposes, ceases to be a constraint.

- 2) In BC’s second semi-coherent stage, all the data used in the first-stage is reanalyzed, along with some “fresh” (as yet un-analyzed) data. *A priori*, it is not clear whether this strategy is more efficient than one in which each semi-coherent stage analyzes only fresh data. (*e.g.*, the first stage analyzes 20 days of data and generates candidates, the second semi-coherent stage searches for those candidates in the *next* 50 days of data and generates a list of candidates that have still “survived”, these survivors are searched for again in the *next* 150 days of data, etc.) In this work, we investigate both kinds of schemes: schemes where previously analyzed data is always *recycled* into subsequent stages, and schemes where each semi-coherent stage analyzes only fresh data.
- 3) For simplicity, BC ignored the fact that the GWs have two possible polarizations (in effect pretending that the detectors measure a scalar wave). This is a reasonable approximation when estimating the number of grid points needed to cover the parameter space, but not, say, when trying to estimate the FA and false dismissal (FD) rates as a function of the threshold at some intermediate stage in the search. (Roughly speaking, scalar waves with the same matched-filter SNR would be easier to detect than actual GWs, since with GWs the full SNR is “split” between the two polarizations, in a way that is unknown *a priori*.) In this work we aim to make realistic estimates of a GW pulsar’s delectability for a given matched-filter SNR (and given region of parameter space to be searched over), so we take polarization into account wherever it makes a significant difference. In practice, this just means that we use the  $\mathcal{F}$ -statistic, Eq. (2.85), as our detection statistic.
- 4) When estimating computational costs, BC assume that the demodulations will be done using *stroboscopic re-sampling*, a method modeled closely on the FFT algorithm. A different demodulation method, which we shall refer to as the SFT method, is currently being used by the GW pulsar search codes in the LIGO Scientific Collaboration (LSC) software library [59, 60]. The SFT method takes SFTs (see section 2.5) as its input and can be more efficient than *stroboscopic re-sampling* in cases where only a narrow frequency range of the demodulated time series is of interest. In this work we explore the possibility of using different demodulation methods at different stages of the search, and attempt to find the most efficient combination.

All the above points will be elaborated on in later sections of this chapter.

This chapter is organized as follows:

Section 3.1 reviews the Stack-Slide algorithm for the *re-sampling* method. Our general multistage strategy for searching through large parameter spaces for GW pulsars, using a combination of semi-coherent methods and coherent methods, is explained in section 3.2. Our general search scheme contains a fair number of free parameters (the number and duration of the coherently analyzed stretches in each semi-coherent stage, as well as the coarseness of the discrete grid used to cover the parameter space of sought-for signals), which we can adjust to make the search as efficient as possible. Our general scheme for performing this optimization is described in section 3.2.2. Section 3.3 develops all the formulae we need to evaluate the computational cost of any of our strategies, for any desired sensitivity. More specifically, section 3.3.1 reviews the template counting formulae developed in BC; section 3.3.2 develops the equations relating the thresholds that candidates must pass at different stages (to advance to the following stage) to the FD rates at those stages, and hence to the overall sensitivity of the search; and section 3.3.3 derives estimates for the dominant computational cost of each part of the search. Section 3.4 describes our results: the optimal strategy (within our general scheme) and its sensitivity. In section 3.5 we revisit the hierarchical search strategy for the case that we search over the data in each stage coherently.

### 3.1 The Stack-Slide algorithm (based on re-sampling method)

The Stack-Slide algorithm is best described with reference to the Doppler shift formula of Eq. (2.5). Imagine we have a data stream  $x(t)$  covering an observation time  $T_{\text{used}}$ , and we wish to search for a GW pulsar with some parameters  $\boldsymbol{\lambda}$ . (The modulated frequency is described by the  $s+3$  parameters consisting of  $f_0$  and  $\vec{\lambda} := (\mathbf{n}, \{f^{(k)}\}_{k=1\dots s})$ , recalling from chapter 2,  $s$  is the number of spin-down parameters,  $f^{(k)}$  the  $k^{\text{th}}$  frequency time derivatives and  $\mathbf{n}$  is the unit vector pointing from the SSB toward the pulsar; we shall often denote the pair  $(f_0, \vec{\lambda})$  by the boldface symbol  $\boldsymbol{\lambda}$ .) We break up the data into  $N$  smaller

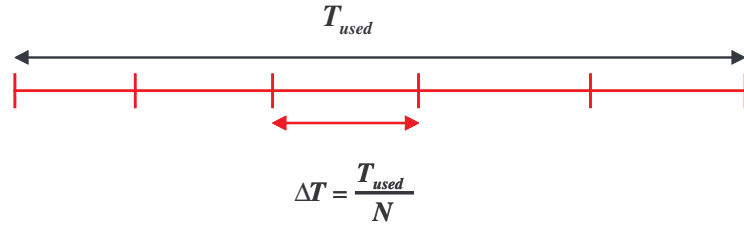


Figure 3.1: In the Stack-Slide algorithm, we first break up data into the smaller segments.

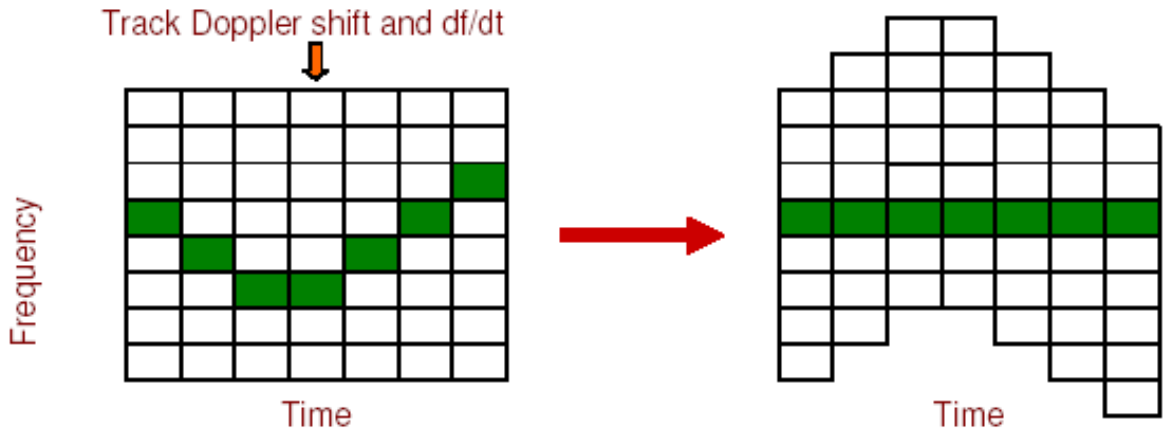


Figure 3.2: An illustration of the discrete frequency bins of the Short Fourier Transform (SFTs) of the data are shown vertically, with the discrete start times of the SFTs shown horizontally. The green pixels (dark pixels in black & white view) represent a signal in the data. Its frequency changes with time due to Doppler shifts and intrinsic evolution of the source. By sliding the frequency bins, the power from a source can be lined up and summed after appropriate weighting or transformation.

segments of length  $\Delta T = T_{used}/N$ , (Fig. 3.1), and calculate the Fourier spectrum of each segment. For now we assume each segment is sufficiently short that the signal frequency remains confined to a single discrete frequency bin. If there is a signal present, it will most likely be too weak to show up in a single segment with any significant signal-to-noise ratio (SNR). However, we can increase the SNR by adding the power from the different segments. We must *not* use the same frequency bin from each segment, but rather must follow the frequency evolution given by Eq. (2.5). Thus, we ‘*stack*’ the power after ‘*sliding*’ each segment in frequency space (Fig. 3.2). Note that the sliding depends on  $\vec{\lambda}$ . Thus, in practice, we choose a grid in the space of  $\vec{\lambda}$ s and the sliding is done differently at each grid point. As described above, the sensitivity of the Stack-Slide algorithm is restricted due to the length of  $\Delta T$ . Therefore we should not take  $\Delta T$  to be too large, since then we would lose SNR due to the signal power being spread over several frequency bins. However, we can gather all the signal power back into a single bin by taking account of the Doppler modulation and spin-down while calculating the spectrum of a segment; *i.e.*, we de-modulate each data segment before summing.

With these concepts at hand, we can now describe the Stack-Slide search for the  $\mathcal{F}$ -statistic. The strategy is very similar to the power summing method described above. Again we break up the data of length  $T_{used}$  into  $N$  segments, each of length  $\Delta T = T_{used}/N$ . We choose a point  $\vec{\lambda}_d$  in parameter space, and demodulate the signal accordingly. We calculate  $\mathcal{F}(f, \vec{\lambda}_d)$  as a function of the frequency for each segment and add the  $\mathcal{F}$ -statistic values after sliding the different segments in frequency space appropriately.

As explained in BC, the resolution of sky- and spin-down-space that suffices for the demodulation is not fine enough for the Stack-Slide step. Thus at each stage we have two grids on parameter space: a coarse one for performing the short-segment demodulations and a fine one for sliding and stacking the short-segment results. We refer the reader to [61] and the appendix of [62] for a detailed derivation of the formula relating the required amount of sliding to the parameters  $\vec{\lambda}_d$ .



## 3.2 A multi-stage hierarchical search

### 3.2.1 The general algorithm

The Stack-Slide search algorithm described in the previous section has two components: 1) calculation of the  $\mathcal{F}$ -statistic for data stretches of length  $\Delta T$ , 2) summation of the resulting  $\mathcal{F}$  values along the appropriate tracks in the time-frequency plane. (If there are  $N$  coherently analyzed segments, then the sums have  $N$  terms.) If we had unlimited computational resources, we would simply do a fully coherent search on all the data; *i.e.*, set  $N = 1$  and take  $\Delta T$  to be the entire observation time. However, the number of templates required for a fully coherent search increases as a fifth power of  $\Delta T$  (Eq. 3.1), making this impractical for all-sky searches.

To illustrate this point, consider an all-sky search for young, fast pulsars, *i.e.*, GW pulsars that have a spin-down age as short as  $\tau_{\min} = 40$  yr and that emit GWs with frequency up to  $f_{\max} = 1000$  Hz. Let us assume that we have 30 days of data available to use. Imagine two different ways of looking for this pulsar: a full 30-day coherent integration versus a semi-coherent method where the available data is broken up into 30 equal segments. The formula for the number of templates required for these searches is given below in Eq. (3.27). It turns out that the full coherent search requires  $\sim 4.2 \times 10^{15}$  templates if we are to not lose more than 30% of the signal power. On the other hand, the semi-coherent search requires only  $\sim 1.5 \times 10^{11}$  templates for the same allowed fractional loss in signal power. The ratio of the the number of templates required for the two types of searches increases rapidly with the observation time; for instance, for an observation time of 40 days, the corresponding numbers are  $\sim 5.5 \times 10^{16}$  and  $\sim 8.3 \times 10^{11}$  for the full coherent and semi-coherent searches respectively.

As illustrated by the above example, semi-coherent searches for unknown GW pulsars are a compromise forced upon us by limited computing power. Such searches will remain computationally limited for the foreseeable future, so it behooves us to organize them as efficiently as possible. In this work we consider a class of multi-stage, hierarchical search algorithms. Since our main “problem” is the large volume of parameter space we need to search over, the basic idea behind these algorithms is to identify and discard unpromising regions of parameter space as fast as possible—without discarding real signals. The type of scheme we consider is illustrated schematically in Fig. (3.3).

The first stage is a semi-coherent search through some fraction of the available data. A threshold is set, and candidates exceeding this threshold are passed to the next stage. The second stage is similar to the first, but includes additional data and generally entails a finer resolution of parameter space. (The latter means that any candidate that survives the first semi-coherent stage gives rise to a little crowd of nearby candidates that are examined in the second semi-coherent stage.) Any candidate that exceeds the second-stage threshold is passed on to the third stage, and so on. In an  $(n+1)$ -stage search, any candidate surviving all  $n$  semi-coherent selections is subjected to a final, coherent search (which we consider the  $(n+1)^{\text{th}}$  stage); if it could pass the final, coherent, threshold then the candidate would be studied in details “by hand”. We impose as a constraint that the false alarm (FA) rate for the entire search must be  $< 1\%$ . For reasons explained below, in realistic examples this inequality is all too easy to satisfy; the actual FA rate for our optimized searches is typically smaller than 1% by many orders of magnitude.

In the end, our search will be able to detect a GW pulsar signal whose rms strength (at the detector)  $h_{\text{RMS}}$  exceeds some threshold value  $h_{\text{th}}$  (with a false dismissal rate of 10% – 15%). We can think of  $1/h_{\text{th}}$  as the search’s sensitivity. We will optimize our search to get the maximum sensitivity for any given computing power or, equivalently, to find the minimum computer power necessary to attain any given sensitivity.

### 3.2.2 The general optimization scheme

In this section we further discuss our search algorithm and its optimization. First we establish some notation. Let  $n$  be the total number of semi-coherent stages. Let  $N^{(i)}$  be the number of stacks used in the  $i^{\text{th}}$  stage and  $\Delta T^{(i)}$  be the length of each stack; the superscript  $i$  will always refer to the  $i^{\text{th}}$  semi-coherent stage. The resolution of the template grid used to cover parameter space is given in terms of the maximum fractional mismatch in signal power  $\mu_{\max}^{(i)}$  [63]. Our detection statistic is  $\rho^{(i)}$ , the sum of



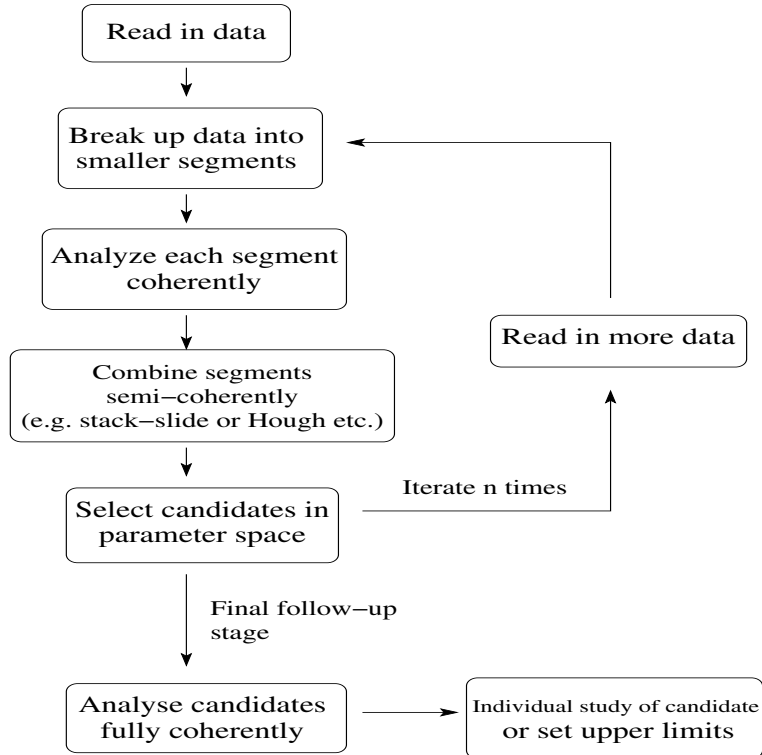


Figure 3.3: A hierarchical scheme for the analysis of large parameter space volumes for continuous wave searches. Each step analyzes only those regions in parameter space that have not been discarded by any of the previous steps.

the  $\mathcal{F}$  values from the different stacks (obtained after sliding appropriately):

$$\rho^{(i)} = \sum_{k=1}^{N^{(i)}} \mathcal{F}_k^{(i)}. \quad (3.2)$$

Denote the distribution of  $\rho^{(i)}$  in the absence of any signal by  $p(\rho^{(i)})$ . In the presence of a GW signal of amplitude  $h_{\text{RMS}}$ , let the distribution of  $\rho^{(i)}$  at the grid-point nearest the actual signal be  $p(\rho^{(i)}|h_{\text{RMS}}, \mu_{\text{max}}^{(i)})$ . Let  $\rho_{\text{th}}^{(i)}$  be the  $i^{\text{th}}$ -stage threshold, which a candidate must exceed to advance to the next stage. The  $i^{\text{th}}$ -stage FA rate (per candidate)  $\alpha^{(i)}$  and FD rate (per candidate)  $\beta^{(i)}$  are given by

$$\alpha^{(i)}(\rho_{\text{th}}^{(i)}) = \int_{\rho_{\text{th}}^{(i)}}^{\infty} p(\rho) d\rho, \quad (3.3)$$

$$\beta^{(i)}(\rho_{\text{th}}^{(i)}; h_{\text{RMS}}, \mu_{\text{max}}^{(i)}) = \int_0^{\rho_{\text{th}}^{(i)}} p(\rho|h_{\text{RMS}}, \mu_{\text{max}}^{(i)}) d\rho. \quad (3.4)$$

Practically identical formulae apply to the final, coherent stage as well.

Again, we require of any search algorithm that, at the very end of the search, it results in a false detection less than 1% of the time. Given this constraint, we parameterize the search's sensitivity by the signal amplitude  $h_{\text{th}}$  such that an embedded signal with  $h_{\text{RMS}} > h_{\text{th}}$  would be detected  $\gtrsim 85\% - 90\%$  of the time. We enforce the latter condition as follows. We set the first-stage threshold  $h_{\text{th}}$  such that a signal of amplitude  $h_{\text{RMS}} = h_{\text{th}}$  will pass to the second stage 90% of the time. At all subsequent stages we set the threshold such that the same signal with strength  $h_{\text{RMS}} = h_{\text{th}}$  has a 99% chance of passing to the next higher stage. That is, we adjust the  $i^{\text{th}}$ -stage threshold  $\rho_{\text{th}}^{(i)}$  so that  $\beta^{(1)} = 0.10$ , while  $\beta^{(i)} = 0.01$  for  $i > 1$ , and  $\beta^{(\text{coh})} = 0.01$  as well. The motivation behind making  $\beta^{(1)}$  lower than  $\beta^{(i)}$  for  $i > 1$  is the following: we believe that a computationally efficient algorithm will have the property that a true signal that is strong enough to pass the first-stage threshold *should* generally pass over all the others. Any source that is not sufficiently strong to make it through to the end of the detection pipeline should be discarded as soon as possible, so as not to waste computing power. This reflects the basic idea behind

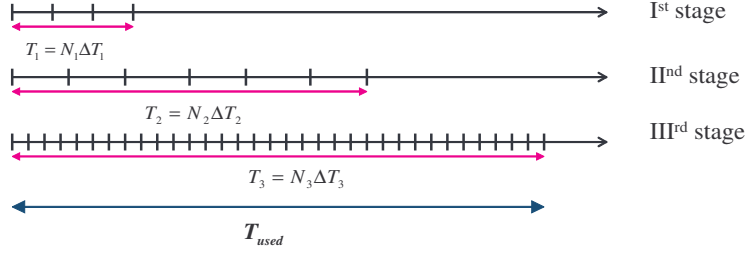


Figure 3.4: The schematic view of how the data are being handled for different stages in the data recycling mode.

our hierarchical searches: to eliminate unpromising regions of parameter space as quickly as possible, so that computational resources can be focused on the more promising regions. Basically, the first-stage threshold determines the sensitivity of the whole search, and subsequent steps whittle down the number of candidates (*i.e.*, the number of small patches in parameter space that perhaps contain a true signal) until any remaining patches can be fully, coherently analyzed.

To fully specify our search algorithm, we have to choose the parameters

$$\Gamma \equiv (n, \{N^{(i)}\}, \{\Delta T^{(i)}\}, \{\mu_{\max}^{(i)}\}, \mu_{\max}^{coh}), \quad (3.5)$$

where  $n$  is the number of semi-coherent stages and  $i = 1, \dots, n$ . In doing so, we are subject to certain requirements or constraints:

- The total amount of data available is no more than some  $T_{\max}$  (say, 1 year).
- We wish to detect (with  $\sim 90\%$  FD rate and  $< 1\%$  overall FA rate) any unknown signal of amplitude  $h_{\text{RMS}}$  greater than  $h_{th}$  (say,  $10^{-26}$ ).

Our task is to choose the parameters  $\Gamma$  that minimize the total required computational power  $P$ , subject to the above constraints. We arrive at a cost function  $P(h_{th})$ , the computational cost of reaching any given sensitivity level. (Really,  $P$  is function of the product  $h_{th}^2 T_{\max} / S_n(f)$ , but we are regarding  $T_{\max}$  and  $S_n(f)$  as fixed.) We can immediately invert this function to determine  $h_{th}(P)$ , the sensitivity achievable for any given computing power.

Let us first deal with the constraint on the total amount of data. We are going to consider simultaneously two different modes of all-sky searches. In *data recycling* mode (Fig. 3.4), at each stage we start back at the beginning of the data, but take progressively larger values of  $N^{(i)} \Delta T^{(i)}$ . Thus the first stage looks at data in the interval  $[T_0, T_0 + N^{(1)} \Delta T^{(1)}]$ , the second stage looks at  $[T_0, T_0 + N^{(2)} \Delta T^{(2)}]$  and so on. The total observation time is thus

$$T_{\text{used}} = N^{(n)} \Delta T^{(n)}. \quad (3.6)$$

In *fresh-data* mode (Fig. 3.5), rather than always starting over from the beginning, we analyze fresh data at each stage. The first stage looks at data in the range  $[T_0, T_0 + N^{(1)} \Delta T^{(1)}]$ , the second stage looks at  $[T_0 + N^{(1)} \Delta T^{(1)}, T_0 + N^{(1)} \Delta T^{(1)} + N^{(2)} \Delta T^{(2)}]$ , etc. The total observation time is thus

$$T_{\text{used}} = \sum_{i=1}^n N^{(i)} \Delta T^{(i)}. \quad (3.7)$$

In either data-recycling or fresh-data mode, one constraint is that  $T_{\text{used}} \leq T_{\max}$  where  $T_{\max}$  is the total amount of data available. Also, in either mode, at each stage we look only at portions of parameter space that exceeded the threshold set at the previous stage.

Next we consider our constraints on the overall FA and FD rates for the pipeline. The final, coherent follow-up stage is expected to be much more sensitive than any of the preceding steps; therefore the overall FA rate is essentially set by the final stage threshold alone. (The earlier stages serve only to whittle down the number of candidates that are analyzed in the final coherent stage.) If the threshold in the final follow-up stage is  $\rho_{th}^{(coh)}$ , then the overall FA rate is no larger than  $\alpha^{(coh)}(\rho_{th}^{(coh)})$  times the number of effectively independent candidates in parameter space. We approximate the latter, crudely,

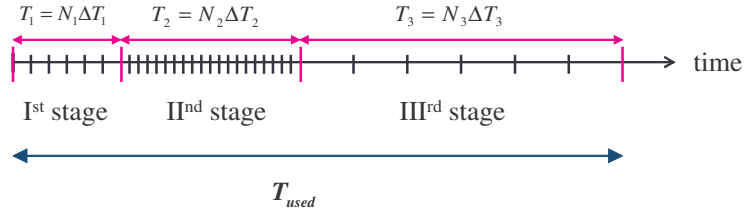


Figure 3.5: Handling data in different stages in the fresh-data mode.

by  $\sim N_p(T_{\max}, 0.2, 1)$ ; in practice  $\alpha^{(coh)}(\rho_{\text{th}}^{(coh)})$  turns out to be so minuscule that the crudeness of this approximation is irrelevant.

The overall false dismissal requirement is also easily handled. Let  $\tilde{\beta}$  be the total false dismissal rate of the multistage search. Each stage has its own threshold  $\rho_{\text{th}}^{(i)}$  and corresponding false dismissal rate  $\beta^{(i)}$ . If each stage, including the follow-up stage, were to analyze completely independent data, we would have

$$\tilde{\beta} = 1 - \prod_{i=1}^{n+1} (1 - \beta^{(i)}) \approx \beta^{(1)} + \dots + \beta^{(n+1)}. \quad (3.8)$$

(where we use “ $\beta^{(n+1)}$ ” interchangeably with  $\beta^{(coh)}$ ). In our fresh-data search mode, the data at different stages are independent, *except* for the final, follow-up stage. And in our recycled-data scheme, the data examined at higher stages includes all the data examined in earlier stages. Then when  $\beta^{(1)} = 0.1$  and

$$\beta^{(2)} = \beta^{(3)} = \dots = \beta^{(n)} = \beta^{(coh)} = 0.01, \quad (3.9)$$

it is clear that  $\tilde{\beta}$  is roughly in the range  $(10 + n - 1)\%$  to  $(10 + n)\%$ , for fresh-data mode and  $10\%$  to  $(10 + n)\%$  for recycled-data mode. Since  $n \approx 3$  turns out to be optimal (see below), we crudely summarize this by saying that our strategies have an overall FD rate of  $10\% - 15\%$  at the threshold value of  $h_{\text{RMS}}$ .

Finally, we turn to the search’s computational cost, which we wish to minimize. Let us denote the total number of floating point operations for the  $i^{\text{th}}$  semi-coherent stage by  $C^{(i)}$  and for the final coherent stage by  $C^{(coh)}$ . Expressions for  $C^{(i)}$  and  $C^{(coh)}$  are given in the next section. For now, it is sufficient to say the total computational cost is

$$C_{\text{total}} = \left( \sum_{i=1}^n C^{(i)} \right) + C^{(coh)}, \quad (3.10)$$

and that if we wish to analyze the data in roughly real time, the required computational power (operations per unit time) is

$$P = \frac{C_{\text{total}}}{T_{\text{used}}}. \quad (3.11)$$

Depending on which mode we are working in,  $T_{\text{used}}$  is given by Eq. (3.6) or Eq. (3.7).

Again, our strategy for optimizing the search is to minimize  $P$ , subject to the constraints listed above.

### 3.3 Metric, template counting, confidence levels, and computational cost

#### 3.3.1 Parameter space’s metric and template counting formulae

The metric used in our optimization strategy is based on the work presented in BC [58] which was originally done by Owen [63]. A more recent work on the metric based on  $\mathcal{F}$ -statistic can be found in [64].

The *match* between parameters of signal and template can be defined as the maximum correlation between these two [63]

$$\mathcal{M}(\boldsymbol{\lambda}, \boldsymbol{\Delta\lambda}) \equiv \max(h(t; \boldsymbol{\lambda}) | h(t; \boldsymbol{\lambda} + \boldsymbol{\Delta\lambda})), \quad (3.12)$$

where,  $\boldsymbol{\lambda} = (f_0, \vec{\lambda})$ , with  $\vec{\lambda} = (\mathbf{n}, f^{(k)})$ . Eq. (3.12), so-called the *ambiguity function* [63], is in fact the fraction of the optimal signal-to-noise ratio using the template with parameters  $(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda})$  over signal with parameters  $(\boldsymbol{\lambda})$

$$\mathcal{M}(\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda}) \equiv \frac{\rho^2(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda})}{\rho^2(\boldsymbol{\lambda})}. \quad (3.13)$$

This, clearly, has a maximum of unity at  $\Delta\boldsymbol{\lambda} = 0$  so can be expanded in a Taylor power series around  $\Delta\boldsymbol{\lambda} = 0$ . By ignoring the terms of  $\mathcal{O}(\Delta\lambda^3)$  and higher, we have

$$\mathcal{M}(\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda}) \approx 1 + \frac{1}{2} \left( \frac{\partial^2 \mathcal{M}}{\partial \Delta\lambda^\alpha \partial \Delta\lambda^\beta} \right)_{\Delta\boldsymbol{\lambda}=0} \Delta\lambda^\alpha \Delta\lambda^\beta. \quad (3.14)$$

If we define the proper distance metric in the parameter space as [63]

$$\mathfrak{g}_{\alpha\beta}(\boldsymbol{\lambda}) \equiv -\frac{1}{2} \left( \frac{\partial^2 \mathcal{M}}{\partial \Delta\lambda^\alpha \partial \Delta\lambda^\beta} \right)_{\Delta\boldsymbol{\lambda}=0}, \quad (3.15)$$

then the *mismatch* between two nearby templates gets the form of

$$m(\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda}) = 1 - \mathcal{M}(\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda}) = \mathfrak{g}_{\alpha\beta}(\boldsymbol{\lambda}) \Delta\lambda^\alpha \Delta\lambda^\beta. \quad (3.16)$$

Note that in the above equations we use the summation rule over the repeating indices. In a simpler word, we have

$$m(\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda}) = \sum_{\alpha, \beta} \mathfrak{g}_{\alpha\beta}(\boldsymbol{\lambda}) \Delta\lambda^\alpha \Delta\lambda^\beta. \quad (3.17)$$

Working on the Stack-Slide algorithm makes it more convenient to express  $\mathfrak{g}_{\alpha\beta}(\boldsymbol{\lambda})$  as a sum of metric computed for the individual stacks

$$\mathfrak{g}_{\alpha\beta}(\boldsymbol{\lambda}) = \frac{1}{N} \sum_{j=1}^N \mathfrak{g}_{\alpha\beta}^{(j)}(\boldsymbol{\lambda}), \quad (3.18)$$

where the individual stack metric,  $\mathfrak{g}_{\alpha\beta}^{(j)}(\boldsymbol{\lambda})$ , are given by [58]

$$\mathfrak{g}_{\alpha\beta}^{(j)}(\boldsymbol{\lambda}) = \langle \partial_{\Delta\lambda^\alpha} \Delta\phi \partial_{\Delta\lambda^\beta} \Delta\phi \rangle_j - \langle \partial_{\Delta\lambda^\alpha} \Delta\phi \rangle_j \langle \partial_{\Delta\lambda^\beta} \Delta\phi \rangle_j. \quad (3.19)$$

The notation  $\langle \dots \rangle_j$  is defined as [58]

$$\langle \dots \rangle_j = \frac{1}{\Delta T} \int_{(j-1)\Delta T}^{j\Delta T} (\dots) dt_{\text{ssb}} \Big|_{\Delta\boldsymbol{\lambda}=0}. \quad (3.20)$$

The phase error,  $\Delta\phi$ , in Eq. (3.19) is given by

$$\Delta\phi[t; (\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda})] = \phi[t; (\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda})] - \phi[t; \boldsymbol{\lambda}], \quad (3.21)$$

where  $\phi[t; \boldsymbol{\lambda}]$  is defined by Eq. (2.1).

Up to here we treated the frequency as one of the search parameter,  $\lambda_0$ , that needs to be matched. However, working in the *stroboscopic re-sampling* method which takes the Fourier transform of required stretch of data, we first look for the local maxima in the frequency parameter  $f$  of the power spectrum. Then we measure the fractional loss in the power due to mismatch in the parameters of  $\vec{\lambda}$ . Technically the maximization of the power should be done over the specific value of  $f$ . But since the search frequency is not known in advanced, we maximize the power over the maximum search frequency  $f_{\text{max}}$ . This ensures us that we never underestimate the mismatch [57, 58].

Based on the above expression, we therefore minimize the mismatch after maximizing the power over  $f$ . The result would be the *projected mismatch* onto the  $(s+2)$ -parameter subspaces [57] as

$$\mu(\vec{\lambda}, \Delta\vec{\lambda}) \equiv \min m(\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda}) = \sum_{i,j} \gamma_{ij}(\vec{\lambda}) \Delta\lambda^i \Delta\lambda^j, \quad (3.22)$$

where

$$\gamma_{ij} = \left( \mathfrak{g}_{ij} - \frac{\mathfrak{g}_{10}\mathfrak{g}_{01}}{\mathfrak{g}_{00}} \right)_{\lambda^0=f_{\text{max}}}. \quad (3.23)$$

By using the above metric we will be able to measure the number of points that must be sampled in our search. To do so, first we need to calculate the total hyper volume  $V_{\mathcal{P}}$  of the space  $\mathcal{P}$  including all parameter values  $\vec{\lambda}$  to be searched over

$$V_{\mathcal{P}} = \int_{\mathcal{P}} d^w \lambda \sqrt{\det|\gamma_{ij}|}, \quad (3.24)$$

where  $w = s + 2$  is the dimension of parameter space  $\mathcal{P}$ , with  $s$  to be the number of spin-down parameters. Note that, since we maximized the power over the frequency before minimizing the mismatch, the parameter  $\mathcal{P}$  is coordinized by  $\vec{\lambda} = \{\alpha, \delta, f^{(k)}\}$ , where  $\alpha$  and  $\delta$  are the positions of the source and  $f^{(k)}$  are the frequency derivatives.

By defining a maximum mismatch between the parameters,  $\mu_{\max}$ , to be the largest fractional loss in the power of the specific source with the parameters  $\mathcal{P}$ , we can define the proper volume of a single patch as [58]

$$V_{\text{patch}} = \left(2\sqrt{\frac{\mu_{\max}}{w}}\right)^w. \quad (3.25)$$

To arrive at the above equation, we assumed that each correction point of the mesh is located at the center of a cube with the side of  $2\sqrt{\mu_{\max}/w}$  [58].

Having the total volume of the parameter space and the volume of a single patch, we can therefore calculate the number of points required for this space to be filled with these patches

$$N_{\mathcal{P}}(\Delta T, \mu_{\max}, N) \equiv \frac{V_{\mathcal{P}}}{V_{\text{patch}}} = \frac{\int_{\mathcal{P}} d^w \lambda \sqrt{\det|\gamma_{ij}|}}{\left(2\sqrt{\mu_{\max}/w}\right)^w}, \quad (3.26)$$

where  $\Delta T$  is the coherent time baseline, and  $N$  the number of stacks.

Note that we defined the  $\mu_{\max}$  to be the maximum loss in the power after power spectra have been added, therefore the number of points mentioned above ( $N_{pf} = N_{\mathcal{P}}(\Delta T, \mu_{\max}, N)$ ) is the the number of fine meshes for the sliding and stacking purpose. There is also a coarse grid, for the demodulation, that tells us the mismatch in each individual power spectra should not exceed  $\mu_{\max}$ . This requires, indeed, the number of points in this coarse mesh to be  $N_{pc} = N_{\mathcal{P}}(\Delta T, \mu_{\max}, N = 1)$ . For simplicity we shall require that the  $\mu_{\max}$  be the same for both fine and coarse grid.

To continue, we use the general form of  $N_p$  – for an all-sky search – as given in Eq. (2.22) of BC [58]:

$$N_p = \max_{s \in \{0,1,2,3\}} \left[ \mathcal{M}_s \mathcal{N}_s G_s \prod_{k=0}^s \left(1 + \frac{0.3r\Omega^{k+1}\tau_{\min}^k}{c k! \sqrt{\mathcal{M}_s}}\right) \right]. \quad (3.27)$$

Here,  $r = 1$  AU is the Earth's orbital radius,  $\Omega = 2\pi/(1\text{yr})$ ,

$$\mathcal{N}_s = \frac{s^{s/2}}{(s+2)^{s/2}} \frac{f_{\max}^s \Delta T^{s(s+3)/2}}{(\mu_{\max}/s)^{s/2} \tau_{\min}^{s(s+1)/2}}, \quad (3.28)$$

$$\mathcal{M}_s = \left(\frac{f_{\max}}{1\text{Hz}}\right)^2 \frac{(s+2)}{4\mu_{\max}} \left(\frac{1}{A^2} + \frac{1}{B^2} + \frac{1}{C^2}\right)^{-1/2}, \quad (3.29)$$

where

$$A = 0.014, \quad B = 0.046 \left(\frac{\Delta T}{1 \text{ day}}\right)^2, \quad C = 0.18 \left(\frac{\Delta T}{1 \text{ day}}\right)^5, \quad (3.30)$$

and the functions  $G_s$  (as given in Appendix A of BC) are

$$G_0(N) = 1 \quad (3.31)$$

$$G_1(N) = \frac{\pi}{6\sqrt{5}}(5N^2 - 4)^{1/2} \quad (3.32)$$

$$G_2(N) = \frac{\pi^2}{180\sqrt{105}}(175N^6 - 840N^4 + 1100N^2 - 432)^{1/2} \quad (3.33)$$

$$G_3(N) = \frac{\pi^3}{75600\sqrt{105}}(3675N^{12} - 58800N^{10} + 363160N^8 - 1053360N^6 + 1484336N^4 - 987840N^2 + 248832)^{1/2}. \quad (3.34)$$

In our numerical work we used the full expressions for the  $G_s$  given above, but for completeness we note that BC also give the following approximate fits to the  $G_s$ , which are valid when  $N \gg 4$ :

$$G_0(N) = 1, \quad (3.35)$$

$$G_1(N) \approx 0.524N, \quad (3.36)$$

$$G_2(N) \approx 0.0708N^3, \quad (3.37)$$

$$G_3(N) \approx 0.00243N^6. \quad (3.38)$$

Roughly speaking, the factor  $\mathcal{M}_s$  in Eq. (3.27) counts distinct patches on the sky as set by the Earth's one-day spin period,  $\mathcal{N}_s$  counts *distinct patches* in the space of spin-down parameters, the  $G_s$  give the dependence of  $N_p$  on the number of stacks,  $N$ , and the factors of the form  $\left(1 + \frac{0.3r\Omega^{k+1}\tau_{\min}^k}{c k! \sqrt{\mathcal{M}_s}}\right)$  effectively account for the increase of search volume required when the frequency derivative  $d^k f/dt^k$  is dominated by the Doppler shift from the Earth's motion around the Sun rather than by the pulsar's intrinsic spin-down.

The  $N_p$  results in BC were derived under the assumption that the observation time is significantly less than one year. As we shall see below, in the cases where the total available data covers an observation time of a year or more, it turns out that for the optimal search, the initial semi-coherent stages typically analyze a few days' to a few months' worth of data. Also, most of the search's computational cost is spent on these early stages. (This is especially true for the young-pulsar search, which is the most computationally challenging.) Therefore, it seems reasonable for our purposes to simply use the  $N_p$  formulae from BC for *all* observation times. Since the cost-errors we make by using the BC formulae will be confined to the later stages, and since the overall sensitivity of the search is effectively set at the first stage, we believe these errors will not significantly affect the total computational cost, for fixed threshold (though they may affect the relative allocation of resources between the different stage). Of course, the validity of this assumption can only really be checked by re-doing the calculation using more accurate expressions for the  $N_p$ 's, appropriate for year-long observation times, but unfortunately such expressions are not currently available.

Even for short observation times, the  $N_p$  calculation in BC used the approximation (2.87), which neglects the amplitude modulation of the signal; however this approximation is not expected to cause significant errors in estimating template numbers.

### 3.3.2 False dismissal rates and the thresholds

In this subsection, we discuss the statistical properties of the Stack-Slide search and solve the false dismissal constraint to obtain expressions for the thresholds.

Recall from chapter 2 that the distribution of the  $\mathcal{F}$ -statistic (or to be more precise,  $2\mathcal{F}$ ), for each coherent search, is given by a non-central  $\chi^2$  distribution [44]. The non-centrality parameter,  $\rho^2 = (h|h)$ , is the optimal signal-to-noise ratio (hereafter we assign the symbol  $\eta$  to it to specify that we talk about mismatched optimal SNR) is given in terms of the signal  $h(t)$  by:

$$\begin{aligned} \eta &= 4 \left(1 - \frac{\mu_{\max}}{3}\right) \int_0^\infty \frac{|\tilde{h}(f)|^2}{S_n(f)} df \\ &= \left(1 - \frac{\mu_{\max}}{3}\right) \frac{2h_{\text{RMS}}^2 \Delta T}{S_n(f)}, \end{aligned} \quad (3.39)$$

where  $\tilde{h}(f)$  is the Fourier transform of  $h(t)$ . We have included a fitting factor of  $(1 - \mu_{\max}/3)$  to account for the average loss in power due to the mismatch between the signal and template.  $h_{\text{RMS}}$  is the root-mean-square value of the signal  $h(t)$ . We can relate  $h_{\text{RMS}}$  to the amplitude  $h_{\text{RMS}}$  defined in Eqs. (2.12) and (2.13), as follows. If one averages  $h_{\text{RMS}}$  over all sky-positions as well as over the polarization parameters  $\iota$  and  $\psi$ , one obtains  $\langle h_{\text{RMS}}^2 \rangle = (2/25)h_0^2$  (see Eq. (93) of [44]).

More explicitly, the distribution is

$$\begin{aligned} p(\mathcal{F}|\eta) &= 2\chi^2(2\mathcal{F}|\eta, 4) \\ &= \left(\frac{2\mathcal{F}}{\eta}\right)^{1/2} I_1(\sqrt{2\mathcal{F}\eta}) e^{-\mathcal{F}-\eta/2}, \end{aligned} \quad (3.40)$$

where  $\chi^2(\cdot|\eta, \nu)$  is the  $\chi^2$  distribution with  $\nu$  degrees of freedom and non-centrality parameter  $\eta$ , and  $I_1$  is the modified Bessel function of first order. The statistic  $\rho$  of interest for the Stack-Slide search is

the sum of the  $\mathcal{F}$ -statistic over  $N$  stacks. Assuming the  $\mathcal{F}$ -statistic for the  $N$  stacks to be statistically independent,  $2\rho$  must follow a  $\chi^2$  distribution with  $4N$  degrees of freedom and non-centrality parameter  $N\eta$

$$p(\rho|\eta, N) = 2\chi^2(2\rho|N\eta, 4N). \quad (3.41)$$

The mean and variance of  $\rho$  are given respectively by

$$\bar{\rho} = 2N + \frac{N\eta}{2}, \quad \sigma_\rho^2 = 2N + N\eta. \quad (3.42)$$

Using the distribution  $p(\rho^{(i)})$ , the false alarm rate for the  $i^{\text{th}}$  semi-coherent stage (defined in Eq. (3.3)) can be evaluated analytically:

$$\alpha^{(i)}(\rho_{\text{th}}^{(i)}) = e^{-\rho_{\text{th}}^{(i)}} \sum_{k=0}^{2N^{(i)}-1} \frac{(\rho_{\text{th}}^{(i)})^k}{k!}. \quad (3.43)$$

As discussed earlier, the overall false alarm probability  $\tilde{\alpha}$  for the search is set by the final coherent follow-up stage. For this stage,  $N = 1$  so that if the threshold on  $\rho$  is  $\rho_{\text{th}}^{(\text{coh})}$ , then it is easy to see from the previous equation that:

$$\tilde{\alpha} = (1 + \rho_{\text{th}}^{(\text{coh})})e^{-\rho_{\text{th}}^{(\text{coh})}}. \quad (3.44)$$

In the presence of a signal, the non-central  $\chi^2$  distribution for  $\rho$  is a little cumbersome to work with, and it is useful to replace it by a Gaussian with the appropriate mean and variance. So we say that the distribution of  $\rho$  must be approximately Gaussian with mean and variance as in Eq. (3.42):

$$p(\rho|\eta, N) = \frac{1}{\sqrt{2\pi\sigma_\rho^2}} e^{-(\rho-\bar{\rho})^2/2\sigma_\rho^2}. \quad (3.45)$$

This approximation is not valid when  $N$  is of order unity. Then for any given  $h_{\text{th}}$ , we should set the threshold of the  $i^{\text{th}}$  stage,  $\rho^{(i)}$ , by the false dismissal requirement:

$$\int_0^{\rho^{(i)}} p(\rho|\eta_{\text{th}}^{(i)}, N)d\rho = \beta^{(i)}, \quad (3.46)$$

where

$$\eta_{\text{th}}^{(i)} := \left(1 - \frac{\mu_{\text{max}}}{3}\right) \frac{2h_{\text{th}}^2 \Delta T^{(i)}}{S_n(f)}. \quad (3.47)$$

Here the factor of  $(1 - \mu_{\text{max}}^{(i)}/3)$  accounts for the average loss in power due to the mismatch between the signal parameters and nearest grid-point parameters. Eq. (3.46) can be solved to find  $\rho_{\text{th}}^{(i)}$  as a function of  $h_{\text{th}}$ ,  $\Delta T^{(i)}$ , and  $\mu^{(i)}$ . Or equivalently, it gives  $h_{\text{th}} = h_{\text{th}}(\rho^{(i)}, \Delta T^{(i)}, \mu^{(i)})$ . This equation can easily be solved by using the properties of the complementary error function. By changing variables in the integral, we can rewrite the false dismissal rate as

$$\beta^{(i)} = \frac{1}{2} \text{erfc} \left( \frac{\bar{\rho}^{(i)} - \rho^{(i)}}{\sqrt{2}\sigma_\rho^{(i)}} \right). \quad (3.48)$$

If  $h_{\text{th}}$  is the smallest value of  $h_{\text{RMS}}$  for which the false dismissal rate is no bigger than  $\beta^{(i)}$ , then according to Eq. (3.48) we have

$$\begin{aligned} \rho^{(i)}(h_{\text{th}}) &= \bar{\rho}^{(i)} - \sqrt{2}\sigma_\rho^{(i)} \text{erfc}^{-1}(2\beta^{(i)}) \\ &\approx 2N^{(i)} \left[ 1 + \frac{\eta_{\text{th}}^{(i)}}{4} \right] - 2\text{erfc}^{-1}(2\beta^{(i)}) \sqrt{N^{(i)}} \sqrt{1 + \frac{\eta_{\text{th}}^{(i)}}{2}}. \end{aligned}$$

In practice, we fix one value of  $h_{\text{th}}$  (our sensitivity goal) for an entire search, and we then set the threshold  $\rho^{(i)}$  at each stage by solving Eq. (3.49), with the false dismissal rates set by  $\beta^{(1)} = 0.1$  and  $\beta^{(i)} = \beta^{\text{coh}} = 0.01$  for  $i \geq 2$ . Our rationale for this choice is as follows: at each stage, one can estimate the signal strength of any successful candidate. If after the first stage, one can already predict that a candidate is not strong enough to pass over the threshold at the second or a higher stage, then one might as well discard it immediately and so not waste computer power on a likely failure. Put the other way, an efficient algorithm should ensure that a true signal that is strong enough to pass over the first stage is also strong enough to pass over all subsequent stages. Then the false dismissal rate for the whole search will be only a little larger than the FD rate of the first stage alone, or a little more than 10%. (An *overestimate* of the total FD rate is the sum of the rates for each of the stages, or 13% for a 3-stage search.)



### 3.3.3 Computational Cost

Let us begin with the first semi-coherent stage. Here, the number of points in the coarse and fine grids are respectively

$$N_{pc}^{(1)} = N_p(\Delta T^{(1)}, \mu_{\max}^{(1)}, 1), \quad (3.49)$$

$$N_{pf}^{(1)} = N_p(\Delta T^{(1)}, \mu_{\max}^{(1)}, N^{(1)}). \quad (3.50)$$

If we are searching in a frequency range from small frequencies up to  $f_{\max}$ , the data must be sampled in the time domain (at least) at the Nyquist frequency  $2f_{\max}$ . The minimum number of data points that we must start out with in the time domain is then  $2f_{\max}\Delta T$ . To calculate the  $\mathcal{F}$ -statistic for each stack, we need to first calculate the quantities  $F_a$  and  $F_b$  which appear in Eq. (4.12). We describe two methods below which may be called the *stroboscopic re-sampling method* and the *SFT method*. Given  $F_a$  and  $F_b$ , the cost of combining them to get  $\mathcal{F}$  is negligible.

*The stroboscopic re-sampling method:* The method suggested in [44] (and also in [58]) is based on the observation that the integrals in Eqs. (4.33) and (4.34) look *almost* like a Fourier transform; the difference being the form of  $\Phi(t)$  in the exponential. However, by suitably re-sampling the time series, effectively re-defining the time variable so that the spectrum of a real signal would look like a spike in a single frequency bin, the integral can be written as a Fourier transform and we can then use the FFT algorithm. Since the cost of calculating an FFT for a time series containing  $m$  data points is  $3m \log_2 m$ , the operations cost of calculating the  $\mathcal{F}$ -statistic for each stack should be approximately  $12f_{\max}\Delta T \log_2(2f_{\max}\Delta T)$ . Repeating this for  $N^{(1)}$  stacks and for each point in the coarse grid, we see that the total cost of calculating  $F_a$  and  $F_b$ , and therefore the  $\mathcal{F}$ -statistic, is approximately

$$12N^{(1)}N_{pc}^{(1)}f_{\max}\Delta T^{(1)}\log_2(2f_{\max}\Delta T^{(1)}). \quad (3.51)$$

We now need to appropriately slide each segment in frequency space and stack them up, *i.e.* add the  $\mathcal{F}$ -statistic values from each stack to get our final statistic  $\rho$ . This has to be done for each point in the fine grid. The cost of sliding is negligible and we need only consider the cost of adding the  $\mathcal{F}$ -statistic values. Since adding  $N^{(1)}$  real numbers requires  $N^{(1)} - 1$  floating point operations, we see that the cost of stacking and sliding for all frequency bins and for all points in the fine grid is approximately

$$f_{\max}\Delta T^{(1)}N_{pf}^{(1)}(N^{(1)} - 1). \quad (3.52)$$

Thus, the computational cost for the first semi-coherent stage is

$$C_{\text{res}}^{(1)} = f_{\max}\Delta T^{(1)}N_{pc}^{(1)} \left[ 12N^{(1)}\frac{\log(2f_{\max}\Delta T^{(1)})}{\log 2} + \frac{N_{pf}^{(1)}}{N_{pc}^{(1)}}(N^{(1)} - 1) \right]. \quad (3.53)$$

The subscript “<sub>res</sub>” indicates that this result is for the stroboscopic re-sampling method.

*The SFT method:* An alternative method is to use as input not the time series, but rather a bank of short time baseline Fourier Transforms (SFTs). This is in fact the method currently being used in the search codes of the LIGO Scientific Collaboration [59]. Here one first breaks up the data into short segments of length  $T_{\text{sft}}$ , and calculates the Fourier transform of each segment. (These segments, which are to be combined *coherently*, are not to be confused with the segments used in the Stack-Slide algorithm which are combined *incoherently*). The exact method of calculating the  $\mathcal{F}$ -statistic from an SFT database is explained in Sec 4.2, and the operations count is derived in Appendix A. The result is (see Eq. (A.4)):

$$\approx 640N^{(1)}N_{pc}^{(1)}f_{\max}\frac{(\Delta T^{(1)})^2}{T_{\text{sft}}} \text{ Flops}. \quad (3.54)$$

Note that the SFT method of calculating the  $\mathcal{F}$ -statistic is  $\mathcal{O}((\Delta T^{(1)})^2)$  while for the stroboscopic re-sampling method it is  $\mathcal{O}(\Delta T^{(1)} \log \Delta T^{(1)})$ .

The total cost of stacking and sliding in the first hierarchical stage using the SFT method is thus:

$$C_{\text{sft}}^{(1)} = f_{\max}\Delta T^{(1)}N_{pc}^{(1)} \left[ \frac{640N^{(1)}\Delta T^{(1)}}{T_{\text{sft}}} + \frac{N_{pf}^{(1)}}{N_{pc}^{(1)}}(N^{(1)} - 1) \right]. \quad (3.55)$$



When all frequencies are to be searched over, stroboscopic re-sampling produces the  $\mathcal{F}$ -statistic about an order of magnitude more cheaply than the SFT method, for typical values of  $\Delta T^{(1)}$ . However when previous stages have narrowed the search to a small fraction of the whole frequency band (for any given  $\bar{\lambda}$ ), the SFT method can be the more efficient one. We should also mention here that it is possible to start with SFTs and combine them in such a way as to get a  $\mathcal{O}(\Delta T^{(1)} \log \Delta T^{(1)})$  operations count; this is in fact the method used in [66]. However, in this thesis, by the ‘‘SFT method’’ we always mean the method described here in Appendix A, with the operation count given above in Eq. (3.55).

It also seems likely that the re-sampling method could be modified so as to be the most efficient one, even when only wanted to demodulate a small frequency band

$$\Delta f = \max \left\{ 1, \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right\} \quad (3.56)$$

around every selected candidate. Presumably the first step would be to heterodyne the data to shift the relevant frequency range to a neighborhood of zero-frequency. Then one would filter out frequencies higher than  $\Delta f$ , followed by the usual demodulation. Eq. (3.60) would then be modified, so that the new cost of demodulating would be  $12N^{(i)}\Delta T^{(i)}\Delta f \log_2(2\Delta T^{(i)}\Delta f)$ . However since the details of this modified demodulation method have not yet been worked out, we will not consider it further in this thesis.

This completes our analysis of the first stage computational costs for both methods. The analysis for the subsequent stages proceeds similarly; the only difference is that subsequent stages analyze only those regions of parameter space that have not been discarded by any of the previous stages. Assuming that almost all the candidates are due to noise, the false alarm rate is a good estimate of the number of candidates produced by any stage. Let us denote by  $F^{(i)}$  the number of candidates which survive the  $i^{\text{th}}$  stage. Since the false alarm rate for the first stage is  $\alpha^{(1)}$ , the number of candidates produced by the first stage is given by

$$F^{(1)} = \max \left\{ 1, f_{\max} \Delta T^{(1)} N_{pf}^{(1)} \alpha^{(1)} \right\}. \quad (3.57)$$

Note that we will always have at least one candidate which makes it through to the next stage. To calculate the cost of a search, we of course must make some assumptions about the data to be processed. Basically, we are assuming that the data consists of Gaussian noise plus one detectable source. (Though we call  $F^{(i)}$  the ‘‘ $i^{\text{th}}$ -stage false alarm rate’’, it is really the ‘‘false alarm rate or the true-source survival rate, whichever dominates’’. In practice, until the last semi-coherent stage, the FA rate always dominates.)

To estimate the computational cost for the  $i^{\text{th}}$  stage, for  $i > 1$ , recall that each of the  $F^{(i-1)}$  candidates produced by the  $(i-1)^{\text{th}}$  stage is in fact a region in parameter space. If we assume that the  $i^{\text{th}}$  stage further refines this region, then we see that the number of  $i^{\text{th}}$ -stage coarse grid points in this region must be, on average,  $N_{pc}^{(i)}/N_{pf}^{(i-1)}$  (again, assuming this ratio to be bigger than 1). Thus, using the stroboscopic re-sampling method, the number of floating point operations to calculate the  $\mathcal{F}$ -statistic in the  $i^{\text{th}}$  stage is

$$F^{(i-1)} \max \left\{ 1, \frac{N_{pc}^{(i)}}{N_{pf}^{(i-1)}} \right\} 12 f_{\max} \Delta T^{(i)} N^{(i)} \log_2(2 f_{\max} \Delta T^{(i)}). \quad (3.58)$$

Each candidate produced by the  $(i-1)^{\text{th}}$  stage occupies a frequency band  $1/\Delta T^{(i-1)}$ , and thus corresponds to  $\Delta T^{(i)}/\Delta T^{(i-1)}$   $i^{\text{th}}$ -stage frequency bins. Thus the operations count for the stacking and sliding is

$$F^{(i-1)} \max \left\{ 1, \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right\} \max \left\{ 1, \frac{N_{pc}^{(i)}}{N_{pf}^{(i-1)}} \right\} \frac{N_{pf}^{(i)}}{N_{pc}^{(i)}} (N^{(i)} - 1) \quad (3.59)$$

floating point operations. Combining these results, we get the computational cost for the  $i^{\text{th}}$  stage ( $i \geq 2$ ):

$$\begin{aligned} C_{\text{res}}^{(i)} &= F^{(i-1)} \max \left\{ 1, \frac{N_{pc}^{(i)}}{N_{pf}^{(i-1)}} \right\} \\ &\times \left[ 12 N^{(i)} f_{\max} \Delta T^{(i)} \frac{\log(2 f_{\max} \Delta T^{(i)})}{\log 2} + \max \left\{ 1, \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right\} \frac{N_{pf}^{(i)}}{N_{pc}^{(i)}} (N^{(i)} - 1) \right]. \quad (3.60) \end{aligned}$$

If instead one uses the SFT method for calculating the  $\mathcal{F}$ -statistic, it is easy to see that the operations count is

$$C_{\text{sft}}^{(i)} = F^{(i-1)} \max \left\{ 1, \frac{N_{pc}^{(i)}}{N_{pf}^{(i-1)}} \right\} \max \left\{ 1, \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right\} \left[ \frac{640N^{(i)}\Delta T^{(i)}}{T_{\text{sft}}} + \frac{N_{pf}^{(i)}}{N_{pc}^{(i)}}(N^{(i)} - 1) \right]. \quad (3.61)$$

After the  $n$  semi-coherent steps, we have the final coherent follow-up stage where the entire stretch of data of duration  $T_{\text{used}}$  is used. For this stage, we analyze  $F^{(n)}$  candidates and simply compute the  $\mathcal{F}$ -statistic without breaking up the data into any smaller stacks. The cost,  $C^{(coh)}$ , for this using the re-sampling method is

$$C_{\text{res}}^{(coh)} = F^{(n)} \max \left\{ 1, \frac{N_p^{(coh)}}{N_{pf}^{(n)}} \right\} 12f_{\text{max}}T_{\text{used}} \frac{\log(2f_{\text{max}}T_{\text{used}})}{\log 2}, \quad (3.62)$$

where  $N_{pf}^{coh} \equiv N_p(T_{\text{used}}, \mu_{coh}, 1)$ , and  $\mu_{coh}$  is the  $\mu_{max}$  of the final, coherent stage. Using the SFT method, we would have

$$C_{\text{sft}}^{(coh)} = F^{(n)} \max \left\{ 1, \frac{N_p^{(coh)}}{N_{pf}^{(n)}} \right\} \max \left\{ 1, \frac{T_{\text{used}}}{\Delta T^{(n)}} \right\} \frac{640T_{\text{used}}}{T_{\text{sft}}}. \quad (3.63)$$

So far, all results in this section are valid whether we are working in fresh-data mode or data-recycling mode. The following formulae, for the number of candidates which survive a given stage, do however depend on which mode we are working in. If we operate in fresh-data mode (analyzing fresh data at every stage—except the last stage, which is a coherent follow-up of all the searched data), we clearly have (for  $i \geq 2$ )

$$F^{(i)} = \alpha^{(i)} \max \left\{ F^{(i-1)}, 1 \right\} \max \left( 1, \frac{N_{pf}^{(i)}}{N_{pf}^{(i-1)}} \right) \max \left( 1, \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right). \quad (3.64)$$

Again, our count assumes that at least one candidate gets “promoted” to the succeeding stage. We note that Eq. (3.64) assumes that the parameter space resolution improves at every stage of fresh-data mode (which seems always to be true for our optimized searches). We also note that Eq. (3.64) is basically identical to Eq. (5.2) of BC, but there it is claimed to be the FA rate for data-recycling mode. That is not correct, in general, as we discuss further below.

If we are in data-recycling mode (at each step, re-analyzing old data, while also adding on new data), then the probabilities of a candidate’s randomly surviving the  $(i-1)^{th}$  and  $i^{th}$  stages are *not* independent, and so Eq. (3.64) is no longer valid. (To see this, consider the limit where only a very tiny bit of data is added on, and the resolution is kept fixed. Then any candidate that survives the  $(i-1)^{th}$  stage has almost a 100% chance of surviving the  $i^{th}$  stage, even if  $\alpha^{(i)}$  is extremely small.) Indeed, the right-hand side (rhs) of Eq. (3.64) is clearly a *lower bound* on the  $i^{th}$ -stage false alarm rate, in data-recycling mode.

We can also place the following *upper bound* on  $F^{(i)}$  for data-recycling mode:

$$F^{(i)} = f_{\text{max}}\Delta T^{(i)} N_{pf}^{(i)} \alpha^{(i)}. \quad (3.65)$$

The rhs of Eq. (3.65) is the number of false alarms that would result if one performed a semi-coherent search of the *entire* parameter space with the given  $(N^{(i)}, \Delta T^{(i)}, \mu^{(i)}, \rho^{(i)})$ , while the left-hand side (lhs) is the false alarms that result from searching only neighborhoods of the points that survived the  $(i-1)^{th}$  stage. Thus for data-recycling mode, we can say that  $F^{(i)}$  is somewhere in the range

$$F^{(i-1)} \left( \frac{N_{pf}^{(i)}}{N_{pf}^{(i-1)}} \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right) \alpha^{(i)} \leq F^{(i)} \leq f_{\text{max}}\Delta T^{(i)} N_{pf}^{(i)} \alpha^{(i)}.$$

Fortunately, when we calculate the total computational cost of some optimized search in data-recycling mode, needed to achieve some given sensitivity  $1/h_{th}$ , if we try plugging in *either* the upper or lower bound for  $F^{(i)}$ , we find the two final results differ from each other by  $\lesssim 18\%$  for a young pulsar ( $\tau_{min} = 40$  year) and  $\lesssim 5\%$  for an old one ( $\tau_{min} = 10^6$  year), which for our purposes is practically insignificant. Moreover, the optimized search parameters obtained when we plug in the upper-limit estimate for  $F^{(i)}$  are quite similar to those we find by plugging in the lower limit instead. Therefore it is safe for us to choose *either* the upper or lower limit as an estimate of  $F^{(i)}$ . For concreteness, in the rest of this chapter we always estimate  $F^{(i)}$  by its upper limit, which slightly overestimates the computational cost of the search.

With these results in hand, we are now ready to calculate the total computational cost of the entire search pipeline. We have a number of choices to make. At each stage, we can use either the stroboscopic re-sampling method or the SFT method in each stage, and we can work in either the data-recycling mode or fresh-data mode from the second stage onwards. For convenience, we somewhat arbitrarily limit the choices by considering only strategies that use either data-recycling mode in every stage or fresh-data mode in every stage. As we shall see below, the efficiencies of these two sorts of searches turn out to be extremely close anyway. Therefore we strongly suspect that more general searches (using fresh-data mode in some stages and data-recycling mode in others) would not give significant improvements.

## 3.4 Results

*The computer codes generated to do the work in this section are available in appendix B.*

### 3.4.1 The optimization method

We next describe our numerical optimization method. The function we want to minimize, the computational power of Eq. (3.11), is a complicated function on a large-dimensional space. Our chosen method is a simulated annealing algorithm [67, 68] based on the downhill simplex method of Nelder and Mead [69]. The downhill simplex method consists of evaluating the function on the vertices of a simplex and moving the simplex downhill and shrinking it until the desired accuracy is reached. The motion of the simplex consists of a prescribed set of “moves” which could be either an expansion of the simplex, a reflection around a face, or a contraction. This method is turned into a simulated annealing method by adding a random fluctuation to the values of the function to be minimized, at the points of the simplex. The temperature of the random fluctuations is reduced appropriately, or in other words “annealed”, until the minimum is found.

There are no universal choices for the rate of annealing or the starting point of the simplex; these depend on the particular problem at hand. For the results presented below, we have used a variety of different starting points and annealing schedules to convince ourselves that the optimization algorithm has converged and that we have indeed found the best minimum. Let us first discuss the starting temperature, whose meaning is as follows. If  $f$  is the the function to be minimized, then the temperature  $\Theta$  parameterizes the amplitude of random fluctuations  $f \rightarrow f + \delta f$  added to  $f$  at the points of the simplex:

$$\delta f = -\Theta \log r, \quad (3.66)$$

where  $0 < r < 1$  is a uniformly distributed random number. A simplex move is always accepted if it takes the simplex downhill, but an uphill step may also be accepted due to these random fluctuations. In our case, we found that a starting temperature of  $\Theta \sim 10^6$ - $10^9$  gives good convergence; this value is to be compared to the typical value  $\sim 10^{13}$  of the computational cost near its minimum for most of the results presented below. We allow a maximum of 500 iterations of the simplex. If the simplex does not converge within 500 iterations, we reduce the temperature by 2% – 5% and restart the iterations from the best minimum found up to that point. These steps are repeated until the simplex converges. The starting point of the simplex cannot be chosen arbitrarily, and for this purpose, it is useful to have a rough idea of the location of the minimum. This requires some experimenting with a sufficiently broad range of starting points; this is especially important when the number of variables is large, as is the case for, say, a search with  $n > 3$ . Having found a suitable starting point for one set of pulsar parameters ( $f_{\max}$  and  $\tau_{\min}$ ), it can be reused for nearby pulsar parameter values. Occasionally, to obtain convergence to a minimum, we have taken as our starting point the minimum we found for nearby pulsar-parameter values.

We next describe how we impose the constraint that the total amount of analyzed data is less than  $T_{\max}$ . One could imagine trying to do this using the method of Lagrange multipliers. However this seemed difficult to implement numerically (for our highly non-linear function  $P$ ), and we found a simpler approach that suffices. The function our algorithm minimizes is not the total computational power  $P$  (defined in Eq. (3.11)) itself, but rather

$$f = P \times \left[ 1 + S \left( \frac{T_{\text{used}}}{T_{\text{max}}} \right) \right], \quad (3.67)$$

where  $S(x)$  is a smooth function such that  $S(x) = 0$  for  $0 < x < 1$  but  $S(x)$  is rapidly increasing exponential function for  $x > 1$ . That is, we impose a very steep penalty for leaving the constraint surface. This works well, and indeed we found it useful to impose some additional (intuitively obvious) constraints in this way, such as requiring the  $N^{(i)}$  and  $\Delta T^{(i)}$  to all be positive; we again multiply  $P$  by factor that is unity when the constraint is satisfied but is very large when the constraint is violated. This “trick” is used to find the *location* of the minimum, but of course the results we report are the values of the function  $P$  there, not  $f$ .

There is one additional technical detail, namely, that our optimization method is meant for the case of continuous, real variables, while our variables  $N^{(i)}$  are strictly integers. We handle this by rounding off  $N^{(i)}$  to the nearest integer while calculating the cost function  $f$ , every time it is called. The downhill simplex algorithm still treats  $N^{(i)}$  as a continuous variable, i.e. we allow arbitrarily small changes to  $N^{(i)}$  when the simplex is moving downhill, but such changes have no effect on  $f$ . We have also tried an alternative approach where  $N^{(i)}$  is kept as a continuous variable throughout, and rounded off only at the very end. We have found that the two approaches yield consistent results.

Finally, we cross-checked our results using two different implementations of the simulated annealing algorithm—those of [70] and [71]—and found that they gave basically equivalent results in our case.

### 3.4.2 The number of semi-coherent stages

The first question we want to answer is: what is the optimum number  $n$  of semi-coherent stages to use in the search? Relatedly, we want to know the most efficient method to use for the  $\mathcal{F}$ -statistic calculation (stroboscopic re-sampling or SFT method) and best mode to work in (fresh-data mode or data-recycling mode). To answer this, we consider an all-sky search for fast/young GW pulsars, by which we mean a search that goes up to frequency  $f_{\max} = 1000$  Hz and that can detect pulsars with spin-down ages  $\tau_{\min}$  as short as 40 yr. We assume the amount of data available is  $T_{\max} = 1$  yr, and ask what is the computational power required to detect pulsar signals whose  $h_{\text{RMS}}$  is or above  $h_{th}$ , given by:

$$\frac{h_{th}^2}{S_n(f)} = 2.5 \times 10^{-5} \text{sec}^{-1}. \quad (3.68)$$

This signal strength corresponds to  $\sqrt{\eta} \approx 39.72$  for a full 1-yr observation time with a perfectly matched template. (Here and below we are implicitly assuming that  $S_n(f)$  hardly varies over the frequency range of the signal.) We choose the  $i^{\text{th}}$ -stage FD rates,  $\beta^{(i)}$ , as given in (and just above) Eq. (3.9), which, along with the detection threshold given by Eq. (3.68), determines the  $i^{\text{th}}$ -stage thresholds  $\rho^{(i)}$ . For simplicity, we set  $\mu^{(coh)} = 0.2$ . While this is a restriction that we simply put in by hand (to slightly reduce the space of search parameters to be optimized), we believe this choice has very little effect on the overall optimized strategy because, as we shall see shortly, the follow-up stage usually accounts for only a tiny fraction of the total computational cost.<sup>1</sup> Thus we are left with  $3n$  parameters to be optimized:  $(\Delta T^{(i)}, \mu_{\max}^{(i)}, N^{(i)})$  for  $i = 1, \dots, n$ , subject to the constraint that the total amount of data analyzed,  $T_{used}$  [given by Eq. (3.6) or (3.7)] is less than 1 yr.

Plots of the minimum computational cost for different  $n$  and for both the data-recycling and fresh-data modes are shown in Fig. (3.6). For each mode, we consider the following three strategies: (i) Use the SFT method in each stage, (ii) Use the re-sampling method in each stage, and (iii) Use the re-sampling method in the first and final follow-up stages, and use the SFT method in all intermediate stages. Therefore there are 6 curves in Fig. (3.6).

The most important lessons from Fig. (3.6) are the following: Strategy (iii) turns out to be better than (i) or (ii). Furthermore, for strategy (iii), there is a significant advantage in a three-stage search as compared to a two-stage or single-stage search, but there is hardly any improvement in computational cost in going to four or more semi-coherent stages. Furthermore, these results are the same whether we use the fresh-data mode or data-recycling mode, and these two modes give very similar total costs. While Fig. (3.6) presents results just for young/fast pulsar searches, we find the same basic pattern for old pulsars, with  $\tau_{\min} \sim 10^6$  yr: strategy (iii) is the most efficient for calculating the  $\mathcal{F}$ -statistic, data-recycling mode and fresh-data mode are almost equally efficient, and having three semi-coherent stages is near-optimal (significantly better than two stages, and practically as good as four). The main difference from the young/fast pulsar case is that the gain in going from 2 to 3 stages is now only a factor of  $\sim 2$  in computational power, i.e., smaller but still significant.

<sup>1</sup>An exception is a search for old pulsars presented in tables 3.7 and 3.8, where the cost for the follow-up stage turns out to be non-negligible. This example in particular will have to be revisited when better formulas for  $N_p$  are available.

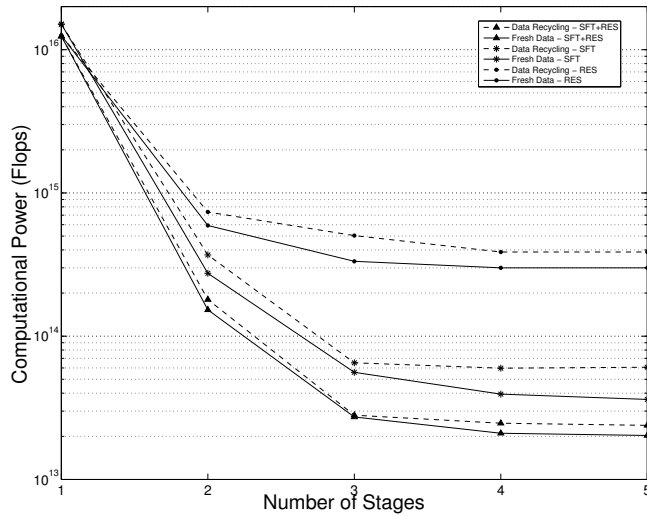


Figure 3.6: Computational power versus number of semi-coherent stages for different methods of calculating the  $\mathcal{F}$ -statistic. RES indicates the stroboscopic re-sampling method (strategy (ii)) and SFT is the SFT method (strategy (i)). SFT+RES corresponds the mixture of these two methods (strategy (iii)). For each strategy, solid lines indicate the result for the fresh-data mode, while the dashed lines are for the data-recycling mode.

In the light of these results, in the rest of this section, we consider only three-stage searches, with the first stage and final follow-up stages employing the re-sampling method and with the second and third stages employing the SFT method. We continue to report results for both data-recycling mode and fresh-data mode.

### 3.4.3 The optimal three-stage search parameters

For the example search described above, (*i.e.*  $f_{\max} = 1000$  Hz,  $\tau_{\min} = 40$  yr,  $T_{\max} = 1$  yr, and  $h_{th}^2/S_n = 2.5 \times 10^{-5} \text{sec}^{-1}$ ) we list the optimal search parameters for the three-stage search in data-recycling mode in Table (3.1). The first two stages analyze about 26 days (divided in 10 segments) and 42 days (divided in 12 segments) of data, respectively, while the third stage analyzes the entire year-long data stretch (divided in 8 segments). The total computational cost is 40.2 TFlops. The cost breakdown among the individual stages, and further cost breakdown into the demodulation piece  $C_{coh}^{(i)}$  and the Stack-Slide  $C_{ss}^{(i)}$  piece in each stage, are given in Table (3.2). There we give the total count of floating point operations required, not the number of operations per second.

| Stage | $\Delta T^{(i)}$ (days) | $\mu^{(i)}$ | $N^{(i)}$ | $T_{\text{used}}^{(i)}$ (days) | $\sqrt{\eta}$ |
|-------|-------------------------|-------------|-----------|--------------------------------|---------------|
| 1     | 2.58                    | 0.7805      | 10        | 25.79                          | 9.08          |
| 2     | 3.51                    | 0.1139      | 12        | 42.13                          | 13.23         |
| 3     | 45.66                   | 0.8196      | 8         | 365.25                         | 33.86         |

Table 3.1: The optimal search parameters in data recycling mode.  $f_{\max} = 1000$ Hz,  $\tau_{\min} = 40$ yr,  $T_{\max} = 1$ yr,  $h_{th}^2/S_n = 2.5 \times 10^{-5} \text{sec}^{-1}$ , and  $\eta$  is defined according to Eq. (3.39).

Our results for fresh-data mode are qualitatively similar, and are given in Table (3.3). In this case, the optimal search analyzes about 24 days of data in the first stage (broken up into 9 segments) and 24 more days in the second stage (broken into 6 segments). The third stage analyzes the rest of the year's worth

| Stage     | $C^{(i)}$ (Flop)      | $C_{\text{coh}}^{(i)}$ (Flop) | $C_{\text{ss}}^{(i)}$ (Flop) |
|-----------|-----------------------|-------------------------------|------------------------------|
| 1         | $9.37 \times 10^{20}$ | $6.21 \times 10^{19}$         | $8.75 \times 10^{20}$        |
| 2         | $3.16 \times 10^{20}$ | $2.46 \times 10^{20}$         | $6.98 \times 10^{19}$        |
| 3         | $1.65 \times 10^{19}$ | $2.73 \times 10^{18}$         | $1.37 \times 10^{19}$        |
| Follow-up | $6.30 \times 10^{15}$ |                               |                              |

Table 3.2: The computational cost to analyze one year of data in data-recycling mode. The search parameters are the same as given in Table 3.1.  $C_{\text{coh}}^{(i)}$  is the cost for the coherent demodulation step and  $C_{\text{ss}}^{(i)}$  for the Stack-Slide step, while  $C^{(i)}$  is the sum of these two. Follow-up indicates the computational cost require for the final follow-up stage.

| Stage | $\Delta T^{(i)}$ (days) | $\mu^{(i)}$ | $N^{(i)}$ | $T_{\text{used}}^{(i)}$ (days) | $\sqrt{\eta}$ |
|-------|-------------------------|-------------|-----------|--------------------------------|---------------|
| 1     | 2.71                    | 0.7829      | 9         | 24.35                          | 8.82          |
| 2     | 4.08                    | 0.0654      | 6         | 24.49                          | 10.17         |
| 3     | 45.20                   | 0.8229      | 7         | 316.42                         | 31.50         |

Table 3.3: Same as Table (3.1), but for fresh-data mode.

of data, divided into 7 segments. The total computational requirement is 34.6 TFlops and its breakdown is given in Table (3.4).

We note the following features of these results. First, in both modes, basically all the data has been analyzed by the end of the third semi-coherent stage. This is not a requirement that we put in by hand, but rather it arises from the optimization: the optimal scheme “gets through” the entire year’s worth of data before the final follow-up stage. Secondly, in data-recycling mode, 73.8% of the computing time is spent in the first stage, 24.9% in the second, 1.3% in the third and a negligible fraction in the follow-up. The results are similar for the fresh-data mode: approximately 74.2% of the computational resources are spent in the first stage, 24.2% in the second stage, 1.6% in the third stage and a negligible amount in the follow up stage. Finally, fresh-data mode entails a slightly lower computational cost than data-recycling mode. However this last fact could be an artifact either of having slightly different overall FD rates in the two cases, or of our using an overestimate of  $F^{(i)}$  in the latter case. The bottom line is that, after optimization, the two modes are almost equally efficient.

If instead we consider a search for older pulsars, with  $\tau_{\text{min}} = 10^6$  yr instead of 40 yr, then the optimal solution for both modes are summarized in Tables (3.5)–(3.8). A larger value of  $\tau_{\text{min}}$  means a smaller number of templates, and therefore a more sensitive search for fixed computational cost.

| Stage     | $C^{(i)}$ (Flop)      | $C_{\text{coh}}^{(i)}$ (Flop) | $C_{\text{ss}}^{(i)}$ (Flop) |
|-----------|-----------------------|-------------------------------|------------------------------|
| 1         | $8.11 \times 10^{20}$ | $6.42 \times 10^{19}$         | $7.46 \times 10^{20}$        |
| 2         | $2.64 \times 10^{20}$ | $2.62 \times 10^{20}$         | $2.67 \times 10^{18}$        |
| 3         | $1.74 \times 10^{19}$ | $5.54 \times 10^{18}$         | $1.19 \times 10^{19}$        |
| Follow-up | $1.62 \times 10^{16}$ |                               |                              |

Table 3.4: Same as Table (3.2), but for fresh-data mode.



| Stage | $\Delta T^{(i)}$ (days) | $\mu^{(i)}$ | $N^{(i)}$ | $T_{\text{used}}^{(i)}$ (days) | $\sqrt{\eta}$ |
|-------|-------------------------|-------------|-----------|--------------------------------|---------------|
| 1     | 14.84                   | 0.3514      | 8         | 118.72                         | 9.06          |
| 2     | 30.06                   | 0.0917      | 6         | 180.34                         | 11.70         |
| 3     | 52.18                   | 0.0986      | 7         | 365.25                         | 16.63         |

Table 3.5: Search parameters for data-recycling mode with  $f_{\text{max}} = 1000\text{Hz}$ ,  $\tau_{\text{min}} = 10^6\text{yr}$ ,  $T_{\text{max}} = 1\text{yr}$ ,  $h_{th}^2/S_n = 4.53 \times 10^{-6}\text{sec}^{-1}$ , and computational power 40.2TFlops;  $\eta$  is defined according to Eq. (3.39).

| Stage     | $C^{(i)}$ (Flop)      | $C_{\text{coh}}^{(i)}$ (Flop) | $C_{\text{ss}}^{(i)}$ (Flop) |
|-----------|-----------------------|-------------------------------|------------------------------|
| 1         | $7.41 \times 10^{20}$ | $2.85 \times 10^{18}$         | $7.39 \times 10^{20}$        |
| 2         | $4.93 \times 10^{20}$ | $3.77 \times 10^{20}$         | $1.16 \times 10^{20}$        |
| 3         | $3.82 \times 10^{19}$ | $1.34 \times 10^{19}$         | $2.48 \times 10^{19}$        |
| Follow-up | $6.18 \times 10^{13}$ |                               |                              |

Table 3.6: The computational cost to analyze one year of data in data-recycling mode. The search parameters are the same as given in Table (3.5).  $C_{\text{coh}}^{(i)}$  is the cost for the coherent demodulation step and  $C_{\text{ss}}^{(i)}$  for the Stack-Slide step, while  $C^{(i)}$  is the sum of these two. Follow-up indicates the computational cost require for the final follow-up stage.

For data-recycling mode, we have lowered the threshold  $h_{th}$  by a factor of 2.35, to the point where the required computational power is again 40.2 TFlops, as in the example of Tables (3.1) and (3.2). The results are shown in Tables (3.5) and (3.6). Compared to the young-pulsar search, the computational power is now spread more evenly over the first two stages: the first stage consumes about 58.27% of the power, the second stage 38.73%, third stage 3.0% and negligible for the follow-up stage.

For the case of fresh-data mode, we have lowered the threshold  $h_{th}$  by a factor 2.36, to the point where the required computational power is again 34.6 TFlops, as in the example of Tables (3.3) and (3.4). Once again, compared to the young-pulsar search, the computational costs are spread more evenly over the first two stages: the first stage consumes about 31.3% of the power, the second stage 30.4%, third stage 14.0%. In this case, the cost for the follow-up stage is 24.0%, which is not negligible. This indicates that, for this case, the earlier stages have not succeeded in reducing the number of candidates to a low level. The overall sensitivity, though, is still almost identical to the data-recycling case.

Let us now discuss the false alarm rate. We require that the overall FA rate be less than 1%, and we claimed in section 3.2.2 that this is automatically satisfied in typical, realistic cases. We can now verify this claim. For the  $\tau_{\text{min}} = 40\text{yr}$  search summarized in tables (3.1)–(3.4), using Eq. (3.39), with  $\mu_{\text{max}} = \mu_{\text{max}}^{(\text{coh})} = 0.2$ , the threshold corresponds to  $\rho_{\text{th}}^{(\text{coh})} \approx \bar{\rho} = 2 + \eta/2 \approx 738$ . By Eq. (3.44), this corresponds to

| Stage | $\Delta T^{(i)}$ (days) | $\mu^{(i)}$ | $N^{(i)}$ | $T_{\text{used}}^{(i)}$ (days) | $\sqrt{\eta}$ |
|-------|-------------------------|-------------|-----------|--------------------------------|---------------|
| 1     | 11.77                   | 0.2074      | 9         | 105.96                         | 9.04          |
| 2     | 10.97                   | 0.0199      | 6         | 65.82                          | 7.13          |
| 3     | 27.64                   | 0.0206      | 7         | 193.47                         | 12.22         |

Table 3.7: Search parameters for fresh-data mode with  $f_{\text{max}} = 1000\text{Hz}$ ,  $\tau_{\text{min}} = 10^6\text{yr}$ ,  $T_{\text{max}} = 1\text{yr}$ ,  $h_{th}^2/S_n = 4.47 \times 10^{-6}\text{sec}^{-1}$ , and computational power 34.6TFlops;  $\eta$  is defined according to Eq. 3.39.

| Stage     | $C^{(i)}$ (Flop)      | $C_{\text{coh}}^{(i)}$ (Flop) | $C_{\text{ss}}^{(i)}$ (Flop) |
|-----------|-----------------------|-------------------------------|------------------------------|
| 1         | $3.46 \times 10^{20}$ | $3.07 \times 10^{18}$         | $3.43 \times 10^{20}$        |
| 2         | $3.35 \times 10^{20}$ | $3.33 \times 10^{20}$         | $1.91 \times 10^{18}$        |
| 3         | $1.54 \times 10^{20}$ | $9.23 \times 10^{19}$         | $6.19 \times 10^{19}$        |
| Follow-up | $2.67 \times 10^{20}$ |                               |                              |

Table 3.8: Same as Table (3.6), except for fresh-data mode. The search parameters are those of Table (3.7).

$\tilde{\alpha} \approx 10^{-318}$  (for either mode). Using Eq. (3.27), the number of independent templates required for a full coherent search of the entire parameter space, using 1 yr of data, is  $f_{\text{max}} T_{\text{max}} N_p (1\text{yr}, 0.2, 1) \approx 10^{34}$ . The overall false alarm rate is thus  $FA = f_{\text{max}} T_{\text{max}} N_p \tilde{\alpha} \approx 10^{-284} \ll 1\%$ <sup>2</sup>.

Similarly, for the case  $\tau_{\text{min}} = 10^6$  yr,  $h_{\text{th}}^2/S_n = 4.53 \times 10^{-6}$  (data recycling mode, Tables (3.5) and (3.6)), we get  $\eta \approx 286$  so that  $\tilde{\alpha} \approx 10^{-61}$ . In this case we have  $N_p \approx 10^{17}$  so that  $FA = f_{\text{max}} T_{\text{max}} N_p \tilde{\alpha} \approx 10^{-34}$ . If we look at fresh data mode (Tables (3.7) and (3.8)) with  $\tau_{\text{min}} = 10^6$  yr,  $h_{\text{th}}^2/S_n = 4.47 \times 10^{-6}$ , we get  $FA \approx 10^{-33}$ . These values are greater than for the case of young pulsars, but still vastly smaller than 1%.

The basic point is simply this: For an all-sky search, sensitivity is limited by computing power, so the detection threshold  $h_{\text{th}}$  in practice is substantially higher than what it would be for infinite computing power. This means that for a signal to be detectable, it must have quite a high SNR (in the matched filter sense)—which means that the FA rate is exponentially small. Being computationally limited means that when we *do* detect something, we can be very confident that it is not simply random noise masquerading as a signal<sup>3</sup>.

How accurate are our numerical results? The total computational cost is a complicated function on a 9-dimensional space and thus is not easy to visualize. We can, however, take appropriate sections of this function to examine its behavior near the minimum. Thus, we can ask whether variations in, say,  $\Delta T^{(1)}$  or  $\Delta T^{(2)}$  away from their optimal values, increase the computational cost (as they should, if should if we have truly found a minimum). To answer this, in Fig. (3.7) we plot the total computational power as a function of  $\Delta T^{(1)}$  and  $\Delta T^{(2)}$ , respectively, for the young-pulsar searches summarized in Tables (3.1) – (3.4). All the other parameters fixed at their optimal values. The minima of these curves agree precisely with our simulated annealing results. Similarly, Figs. (3.8) and (3.9) carry the same message, as well as showing the strong dependence of the computational cost on  $N^{(i)}$  and  $\mu_{\text{max}}^{(i)}$ . (It is not so clear from the plot of  $P$  vs.  $\mu_{\text{max}}^{(3)}$  that this curve has a minimum in the range shown, but it does in fact have a very shallow one.)

For the plot of computational cost versus  $N^{(3)}$ , we are not allowed to keep  $\Delta T^{(3)}$  fixed, since that could violate the constraint  $\Delta T^{(3)} N^{(3)} \leq T_{\text{max}}$ . (Recall that  $\Delta T^{(3)} N^{(3)} = T_{\text{max}} = 1$  yr for the optimal 3-stage solution, which is therefore just at the boundary of the constraint region.) Therefore, we choose to plot the computational cost as a function of  $N^{(3)}$  while simultaneously varying  $\Delta T^{(3)}$  according to  $\Delta T^{(3)} = T_{\text{max}}/N^{(3)}$ .

A noteworthy feature of these plots is that the computational power  $P$  depends more sensitively on the early-stage parameters than the late-stage ones; e.g., more sensitively on  $N^{(1)}$  and  $N^{(2)}$  than on  $N^{(3)}$ . This result should not be surprising since, as mentioned earlier, for the young-pulsar search the computational cost of the higher stages is relatively small.

<sup>2</sup>Strictly speaking, Eq. (3.27) for the number of templates is valid only for observation times which are significantly less than a year. However, unless the discrepancy is many orders of magnitude, the numbers obtained show that it is obviously sufficient for the purposes of this argument.

<sup>3</sup>Of course, no sensible person would ever claim that he or she had detected a GW pulsar with FA probability of less than  $10^{-284}$ . In such a case, the “statistical error” is so ridiculously small that the true FA rate is dominated by the other, hard-to-quantify factors, such as the probability of having some bug in the instrumentation or in the data analysis code.



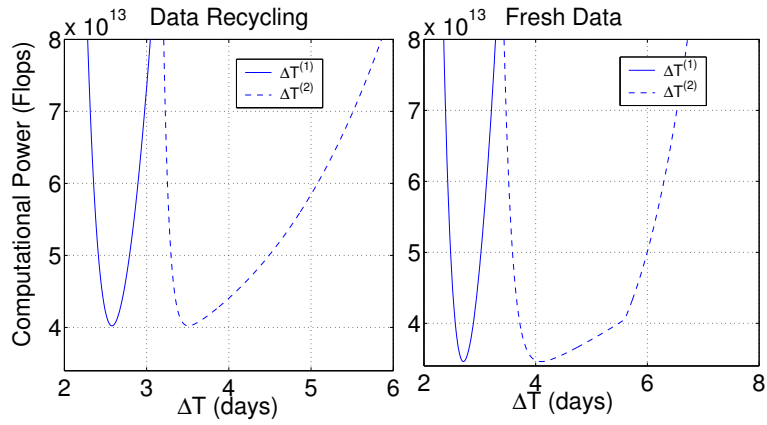


Figure 3.7: Computational power  $P$  as a function of  $\Delta T^{(1)}$  and  $\Delta T^{(2)}$ , with all other parameters fixed to their optimal values.

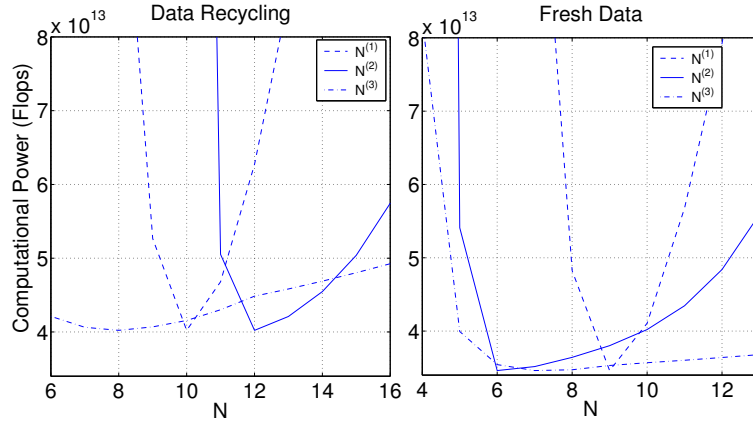


Figure 3.8: Computational power  $P$  as a function of  $N^{(1)}$ ,  $N^{(2)}$  and  $N^{(3)}$ . For the  $N^{(1)}$  and  $N^{(2)}$  plots, all other parameters have fixed to their optimal values. For the  $N^{(3)}$  plot, we have also varied  $\Delta T^{(3)}$  according to  $\Delta T^{(3)} = T_{\max}/N^{(3)}$  in order to satisfy the constraint that the amount of data available is  $T_{\max}$ .

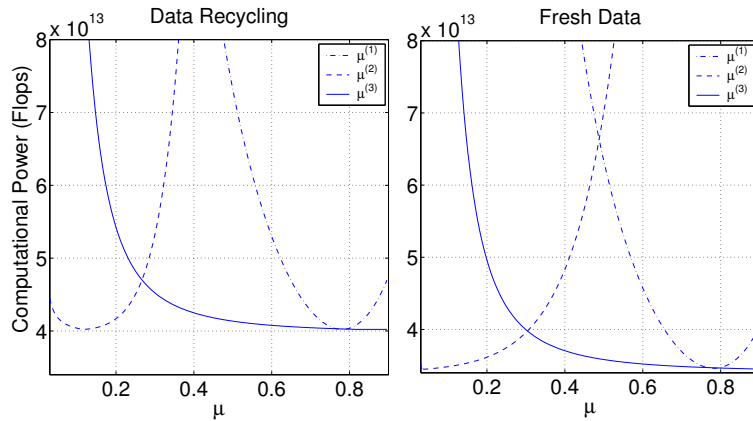


Figure 3.9: Computational power  $P$  as a function of  $\mu_{\max}^{(1)}$ ,  $\mu_{\max}^{(2)}$ , and  $\mu_{\max}^{(3)}$ . For each plot, all other parameters are fixed at their optimal values.

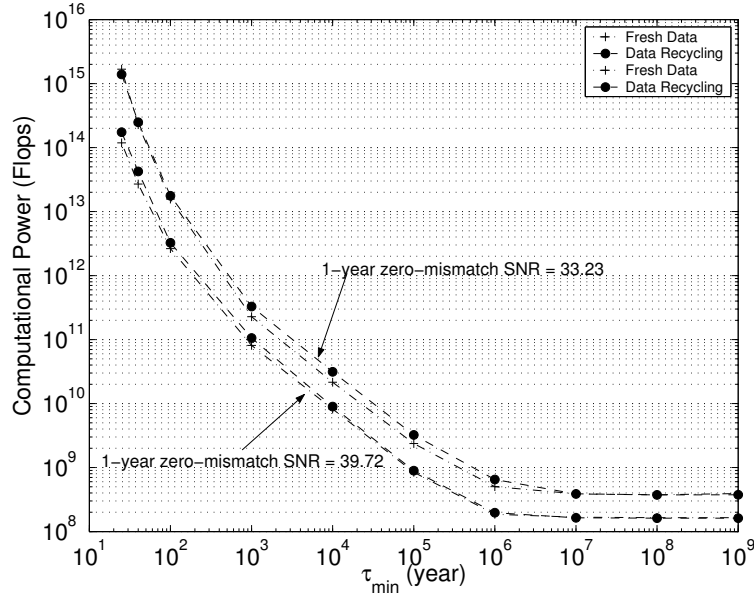


Figure 3.10: The minimum computational power  $P$  required for analyzing 1 year worth of data as a function of the pulsar’s spin-down age  $\tau_{\min}$ . We consider a three-stage search in the both data-recycling and fresh-data mode, for two different signal strengths. The data-recycling mode results are shown with dashed lines, while the fresh-data results are in dotted lines. In parts of the curves, the results for the two modes are so close together that it is hard to distinguish them.

### 3.4.4 The spin-down age and the SNR

How does the (minimum) computational cost depend on the shortest spin-down timescale that we search over,  $\tau_{\min}$ ? Consider again the case where we have one year of data and we perform an all-sky search up to a frequency of  $f_{\max} = 1000$  Hz. Fig. (3.10) shows the result for the both data-recycling and fresh-data mode, for two different values of the 1-year SNR. Note that these results do pass simple sanity checks: the computational cost decreases as the SNR increases (since it is easier to look for stronger signals), and the computational cost decreases as  $\tau_{\min}$  increases (since it is easier to search through a smaller parameter space).

One can also ask: for a given available computational power, how does the threshold SNR scale with  $\tau_{\min}$ ? This is shown in Fig. (3.11). The plot is based on the assumption that we have one year’s worth of data and that we have 10 TFlops of computing power at our disposal. By “SNR”, here we mean the matched-filter SNR, for a perfectly matched filter. Fig. (3.11) tells us that a search for unknown GW pulsars with spin-down ages  $> 10^6$  yr can detect  $\sim 85 - 90\%$  of pulsars whose SNR is  $> 17$  (again, with FA rate  $\ll 1\%$ ). In an all-sky search for very young pulsars, with  $\tau_{\min} = 40$  yr, the SNR required for detection (with the same FD and FA rates) increases to  $\sim 43$ . In comparison, for a source where the sky position and frequency are known in advance (from radio observations), an SNR of only 4.56 is required for detection, with a 10% FD rate and 1% FA rate [54].

Fig. (3.11) strongly suggests that one would like to simultaneously perform at least two different all-sky searches: one for old GW pulsars and another for young ones, with comparable (within a factor ten) computer power devoted to each, but with quite different thresholds. (If one set the same threshold for both old and young pulsars, then almost all computing resources would end up being spent on the young ones.) Clearly, to determine the “best” apportionment of resources between the two types of searches would require some additional inputs/assumptions, but at least Fig. (3.11) seems a good first step towards making an intelligent allocation.

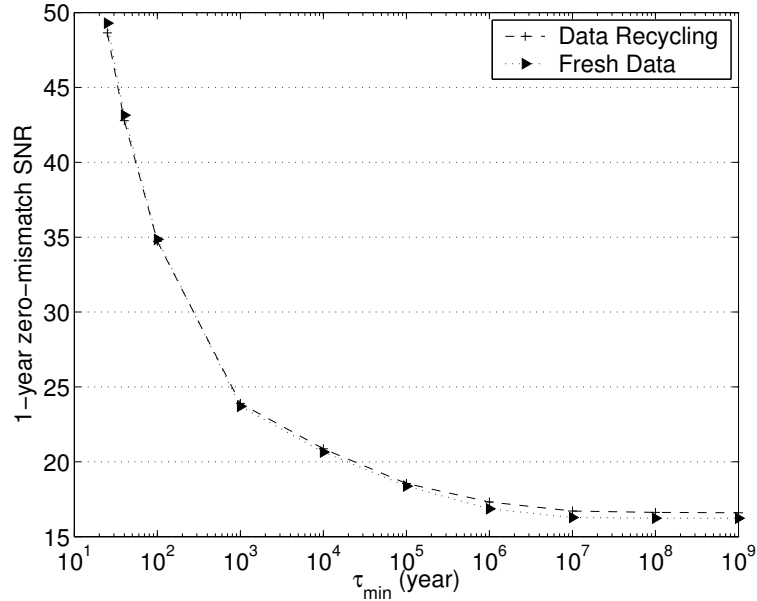


Figure 3.11: The 1-year SNR (with zero mismatch) as a function of  $\tau_{\min}$ , for fixed computational power  $P = 10^{13}$  Flops. The dashed line indicates the result for data-recycling mode and the dotted line for fresh-data. Since these two result are very close to each other, it may be difficult to distinguish them.

### 3.5 Optimization of coherent hierarchical search strategy

Under certain circumstances, it can be advantageous to perform a coherent instead of a semi-coherent search. Notably, the sensitivity for a coherent search will generically be better (albeit at much greater computational cost); with this reasoning, the first stage of the Einstein@Home project used a single stage full coherent search. In this section, we will see how the strategy for a semi-coherent hierarchical search developed in the previous sections might carry over to the coherent case. While this might not lead to the overall optimum strategy, it is an interesting case to study; in fact, the pulsar group of the LIGO-GEO collaboration is developing software based on hierarchical coherent search methods for use with LIGO/GEO data analysis.

As mentioned at the beginning of this chapter, performing a full coherent search is computationally very expensive. Given currently available computer resources, we can only do search over a several hours to a few days. This statement is made more precise by Fig. (3.12): estimating the available computing power at *one TFlops*, then even with a hierarchical coherent search strategy with three optimum stages (more about which below), the maximum length of a data set we can analyze coherently is about 7 days; at the same time, this sets an upper limit for a single stage full coherent search requiring even more computing power. Also, note that the increase is non-linear: comparing coherent analysis of 5 days' worth of data and that of 10 day's worth (amounting to double the data), the required computational cost would increase with a factor of more than 200. With this dependency, it is no wonder that, whenever it can be implemented, a semi-coherent optimization strategy will be preferable. Let us have a closer look at how a coherent hierarchical search scheme can be implemented. In this search, instead of breaking data into smaller segments, analyzing these segments separately, and later combining the results (as would be done in a semi-coherent search), a coherent search will always apply to the data as a whole. More concretely, for the first stage, we take some amount of data and analyze it coherently (*i.e.* as a whole); for the second stage we include more data, but we do not just look at the additional, “fresh” data (cf. the *fresh data mode* of the semi-coherent hierarchical search); rather we take all the available data (our original data plus the fresh data) and analyze the combined data coherently, resulting in a list of candidates that pass our specified threshold. (Note that with this, we are actually following the *data recycling method* mentioned in the context of the semi-coherent hierarchical search scheme.)

We continue this procedure as often as necessary. As is shown in Fig. (3.13), using more than three stages will not result in a significant increase of performance. For this figure, 10 days' worth of data have been analyzed in the data recycling mode over a different number of stages: by doing two stages

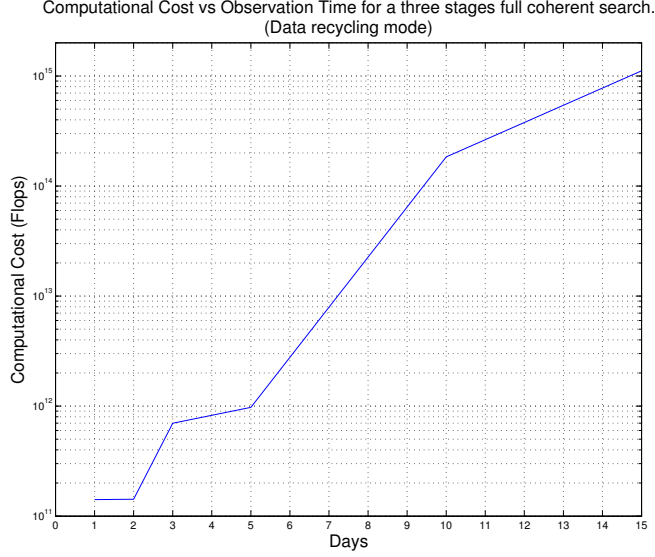


Figure 3.12: Required computational cost for a three stages coherent search versus the total observational time.

| Stage | $T^{(i)}$ (days) | $\mu^{(i)}$ | $\sqrt{\eta}$ |
|-------|------------------|-------------|---------------|
| 1     | 0.74             | 0.3984      | 5.66          |
| 2     | 2.18             | 0.7816      | 9.71          |
| 3     | 5.00             | 0.6450      | 14.70         |

Table 3.9: The optimal search parameters for coherent hierarchical search in data recycling mode.  $f_{\max} = 1000\text{Hz}$ ,  $\tau_{\min} = 40\text{yr}$ ,  $T_{\max} = 5\text{days}$ ,  $h_{th}^2/S_n = 5.0 \times 10^{-4}\text{sec}^{-1}$ , and  $\eta$  is defined according to Eq. (3.39).

instead of one, computational costs is reduced by a factor of 50; for three stages instead of two, the decrease is by a factor of two; using more than three stages we will not result in a significant decrease of computing cost. We conclude that the optimum number of stages in our searches is *three*; the same as for the semi-coherent searches, but with a hugely greater overall computational cost for comparable amounts of data.

Having found the optimum number of stages, the next set of quantities to consider is the optimum amount of data to be used at each stage for lowest overall computational cost. Consider a young, fast pulsar with a minimum spin-down age of 40 years and a frequency of up to 1000 Hz, and set the minimum detectable signal to be  $h_{th}^2/S_n = 5.0 \times 10^{-4}\text{sec}^{-1}$ , and assume that we are looking at a total of 5 days' worth of data. As in the case of the semi-coherent hierarchical search strategy, let us use a 10% false dismissal rate for the first stage and 1% for the subsequent stages. The optimization method described in section 3.4.1 shows the optimum search strategy to be as follows: use about 0.74 days for the first stage, with a maximum mismatch to be  $\mu_1 = 0.40$ ; for the second stage, use 2.18 days of data with  $\mu_2 = 0.78$ . For the final stage, use all 5 days of data with  $\mu_3 = 0.65$ . The computational cost for this particular strategy amounts to 0.98 TFlops. With this search strategy, the 5 days signal-to-noise ratio will be about 14.70; if the target of the search is a significantly stronger signal, computational costs could be reduced even further. The result is shown in more detail in Table (3.9). Note the following interesting differences between a three stage semi-coherent search and a coherent search with the same number of stages: in section 3.4.3, we saw that the distribution of computational cost for a search with three stages was 73.8% in the first stage, 24.9% in the second stage and finally 1.3% in the last stage (Table (3.10)). In contrast, in the case of a coherent search, the computational cost for the first two stages is about the same, and much larger than the cost for stage three; in our particular case, we should spend 44.7% on the first stage, 42% on the second, and a mere 13.3% for the last stage in our optimization scheme. It is quite

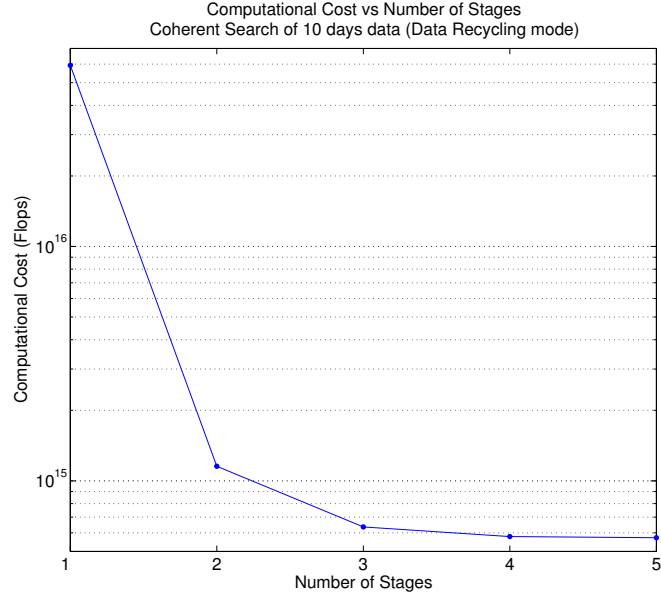


Figure 3.13: Required computational cost for a 10 days of data to be analyzed coherently in a multi stages search.

| Stage | Semi-Coherent<br>(% of total) | Coherent<br>(% of total) |
|-------|-------------------------------|--------------------------|
| 1     | 73.8                          | 44.7                     |
| 2     | 24.9                          | 42.0                     |
| 3     | 1.3                           | 13.3                     |

Table 3.10: The distribution of required Computational Cost (CP) in different stages for different optimization strategies.

significant that we pay roughly half the total cost for analyzing the comparatively small amount of data used in the first stage!

After this brief summary – with its focus on the comparison between semi-coherent and coherent hierarchical searches – let us have a closer look at the mathematical formalism of the coherent search strategy, using the concepts and formulae introduced in the preceding sections. Instead of dividing the data at each stage into different numbers of segments, we fix  $N = 1$  in all cases. We start by calculating the strength threshold for the signal we want to detect: substituting  $N = 1$  in Eq. (3.42), we obtain the following expression for the mean and variance of our statistic:

$$\bar{\rho} = 2 + \frac{\eta}{2}, \quad \sigma_{\rho}^2 = 2 + \eta, \quad (3.69)$$

where  $\eta$  is the non-centrality parameter defined in Eq. (3.39). Substituting these into Eq. (3.49), and fixing both the false dismissal rate and the minimum value of the signal strength through our choice of  $\eta_{\text{th}}$  (Eq. (3.47)), we obtain the threshold

$$\begin{aligned} \rho^{(i)}(h_{\text{th}}) &= \bar{\rho}^{(i)} - \sqrt{2}\sigma_{\rho}^{(i)} \operatorname{erfc}^{-1}(2\beta^{(i)}) \\ &\approx 2 \left[ 1 + \frac{\eta_{\text{th}}^{(i)}}{4} \right] - 2\operatorname{erfc}^{-1}(2\beta^{(i)}) \sqrt{1 + \frac{\eta_{\text{th}}^{(i)}}{2}}. \end{aligned} \quad (3.70)$$

Note that all results mentioned in this section are based on the SFT method (as is the Frequency Domain Search [FDS] of the LSC pulsar group). Still following the same recipe as for the semi-coherent hierarchical

search, our next step is to compute the false alarm probability and computational cost of the first stage. According Eq. (3.43), the false alarm rate for a coherent search undertaken in stages is

$$\alpha^{(i)}(\rho^{(i)}) = (1 + \rho^{(i)})e^{\rho^{(i)}}. \quad (3.71)$$

The number of candidates which pass the first stage is then given as

$$F^{(1)} = \max \left\{ 1, f_{\max} T^{(1)} N_{pc}^{(1)} \alpha^{(1)} \right\} \quad (3.72)$$

(cf Eq. (3.57)).

As we have shown in the context of semi-coherent strategies, the results for the data recycling mode and the fresh data mode are the same; in order to avoid needless repetition, we will focus on the data recycling mode only in the following exposition, in particular since all our results are based on that particular mode.

In this mode, the formulae for the higher stages are the same as for the first; the number of candidates passing on to the next stage is always of the form

$$F^{(i)} = f_{\max} T^{(i)} N_{pt}^{(i)} \alpha^{(i)}, \quad (3.73)$$

which is actually the upper bound on  $F^{(i)}$ . The quantity  $N_{pt}^{(i)}$  mentioned above is the number of grid-points in the parameter space (as the same as  $N_{pc}$  for the semi-coherent search strategy). It can be expressed as

$$N_{pt}^{(i)} = N_p(T^{(i)}, \mu_{\max}^{(i)}), \quad (3.74)$$

(cf. Sec. 3.3.1 for the exact formulation).

The computational cost of a coherent search using the SFT method according to Eq. (3.54) is

$$C^{(1)} = 640 N^{(1)} N_{pt}^{(1)} f_{\max} \frac{(T^{(1)})^2}{T_{\text{sft}}} \text{ Flops}, \quad (3.75)$$

for the first stage; for the subsequent stages

$$C_{\text{sft}}^{(i)} = 640 F^{(i-1)} \max \left\{ 1, \frac{N_{pt}^{(i)}}{N_{pt}^{(i-1)}} \right\} \max \left\{ 1, \frac{T^{(i)}}{T^{(i-1)}} \right\} \times \left[ \frac{T^{(i)}}{T_{\text{sft}}} \right]. \quad (3.76)$$

These are the formulae on which the results mentioned above are based.

## Chapter 4

# Multi-IFO Coherent Search

The  $\mathcal{F}$ -statistic shown in chapter 2, which was originally derived by Jaranowski, Królak and Schutz (JKS)[44], is the optimal statistic for the detection of nearly periodic gravitational waves from unknown GW pulsars. In fact, the  $\mathcal{F}$ -statistic derived by JKS applies just to the case of a single detector with stationary noise (*i.e.*, where spectral density is constant in time). Although JKS mentioned the possibility of using data from a network of detectors to calculate the  $\mathcal{F}$ -statistic, they did not give explicit details on how to implement this. The first detailed calculation for the general case of a network of detectors, with time-varying noise curves and, for a collection of known sources, was published by Cutler and Schutz (CS)[52]. In this chapter, we apply the formalism developed by CS to the case of a *single source* and *N detectors* with *uncorrelated noises*. (As CS have pointed out, it should be straightforward to generalize this to the case of correlated noise between different detectors.)

The main aim of this chapter is to review the multi-IFO  $\mathcal{F}$ -statistic as well as its discretization and implementation in our software. In section 4.1 we present the multi-IFO  $\mathcal{F}$ -statistic for the case of several uncorrelated detectors. In section 4.1 we treat time as a continuous parameter. In section 4.2 we translate these results to the case of discrete timesteps (since real data is sampled discretely). Using the notation of CS, we follow the calculation laid out in the technical note of Itoh, Mendell and Siemens [73], to generalize the formulation of so-called *LALDemod* explained by Siemens [74]. In section 4.3 we review the semi-analytic calculation of  $\mathcal{F}$ -statistic. Section 4.4 is devoted to the question of estimating signal parameters. This is of fundamental importance for any real search for gravitational wave sources, since one main aim is to obtain the best-fit parameters from a given signal for a given type of known source. Finally, section 4.5 is on using *noise weights* when calculating the  $\mathcal{F}$ -statistic. This has two advantages; one corresponds to the fact that our data is in single precision with typical strains  $\tilde{x} \sim 10^{-23}$ , therefore the noise spectral density, which is proportional to the  $|\tilde{x}^2|$ , has a value of order  $10^{-46}$ . Since the range of a single precision floating points is from  $\sim 10^{-44}$  to  $\sim 10^{+38}$ , we cannot use single precision to represent noise floor. Using noise weights, it is possible to make all the numbers that will actually be used in our numerical computations of order unity. At the same time, we also use our noise-weight procedure to correctly account for the non-stationarity of the noise, since we want to give more weight to a chunk with the smaller value of the noise floor and lower weight to the larger one. The details of this come in section 4.5.

### 4.1 Generalized maximum likelihood function for multi-IFO

Going back to the definition of the maximum likelihood function in Eq. (2.73), and considering the fact that we are looking for a single signal buried in all the different data sets from the different detectors, the generalized likelihood function can be written as

$$\Lambda = \frac{P(x^1, x^2, \dots | h)}{P(x^1, x^2, \dots | 0)}, \quad (4.1)$$

where  $x^1$  stands for  $x^1(t)$ ,  $x^2$  for  $x^2(t)$ , etc.

Assuming that there are no correlations between different detector noises, we can readily write the probability of the multi-detector data set as the product of the probabilities for each detector's separate

data set;

$$P(y_1, y_2, \dots) = P(y_1) \cdot P(y_2) \cdots . \quad (4.2)$$

With this factorization, the generalized likelihood function of Eq. (4.1) takes on the form

$$\begin{aligned} \Lambda &= \frac{P(x^1|h)P(x^2|h)\cdots}{P(x^1|0)P(x^2|0)\cdots} \\ &= \frac{P(x^1|h)}{P(x^1|0)} \cdot \frac{P(x^2|h)}{P(x^2|0)} \cdots . \end{aligned} \quad (4.3)$$

But that means that the combined likelihood function for data from multiple detectors is simply the product of the likelihood functions for each detector's data set,

$$\Lambda = \Lambda^1 \times \Lambda^2 \times \cdots . \quad (4.4)$$

In consequence, the log of the general likelihood function (Eq. 4.1) is

$$\log \Lambda = \sum_{\mathbf{x}} \left[ (x^{\mathbf{x}}|h^{\mathbf{x}}) - \frac{1}{2}(h^{\mathbf{x}}|h^{\mathbf{x}}) \right], \quad (4.5)$$

where the index 'x' runs over all the different detectors.

The important lesson is that absence of correlation implies linearity: As long as noises are uncorrelated, the logarithm of the likelihood ratio for the combined set can be computed by adding the logarithms of the likelihood ratio for each detector individually

Redefining the signal  $h(t)$  (Eq. 2.17) for each individual detector 'x' as

$$h^{\mathbf{x}}(t) = \sum_{a=1}^4 A^a h_a^{\mathbf{x}}, \quad (4.6)$$

and making this substitution in the likelihood function (Eq. 4.5), we have

$$\log \Lambda = \sum_{\mathbf{x}} (x^{\mathbf{x}}|A^a h_a^{\mathbf{x}}) - \frac{1}{2} \sum_{\mathbf{x}} (A^a h_a^{\mathbf{x}}|A^b h_b^{\mathbf{x}}). \quad (4.7)$$

Since the  $A^a$ s are independent of the detectors, we can take them out of the summation

$$\log \Lambda = A^a \sum_{\mathbf{x}} (x^{\mathbf{x}}|h_a^{\mathbf{x}}) - \frac{1}{2} A^a A^b \sum_{\mathbf{x}} (h_a^{\mathbf{x}}|h_b^{\mathbf{x}}), \quad (4.8)$$

with all the  $A^a$  in front.

It is helpful to generalize Eqs. (2.79) and (2.80) and introduce the quantities

$$\mathbf{H}_a = \sum_{\mathbf{x}} (x^{\mathbf{x}}|h_a^{\mathbf{x}}), \quad (4.9)$$

and

$$\mathbf{M}_{ab} = \sum_{\mathbf{x}} (h_a^{\mathbf{x}}|h_b^{\mathbf{x}}). \quad (4.10)$$

With their help, we can rewrite Eq. (4.8) as

$$\log \Lambda = A^a \mathbf{H}_a - \frac{1}{2} A^a A^b \mathbf{M}_{ab}. \quad (4.11)$$

By maximizing the log of likelihood function (Eq. 4.11)  $A^a$ s, we obtain the general form of the  $\mathcal{F}$ -statistic, namely

$$\mathcal{F} = \frac{1}{2} \mathbf{H}_a \mathbf{M}^{-1ab} \mathbf{H}_b. \quad (4.12)$$

Note that our vector  $\mathbf{H}$  and matrix  $\mathbf{M}$  in the above equation are the sum over the separate  $H$  and  $M$ , defined for each detector using Eqs. (2.79) and (2.80).



As Eq. (4.12) shows, the  $\mathcal{F}$ -statistic for multiple detectors has the same form as that for a single detector (cf. Eq. 2.85), except that each component represents a sum over all the detectors. In consequence, the recipe for calculating the combined  $\mathcal{F}$ -statistic is straightforward: compute the quantities  $H$  and  $M$  for each detector; add them to construct  $\mathbf{H}$  and  $\mathbf{M}$ ; combine these new quantities to obtain the  $\mathcal{F}$ -statistic.

Actually, following many other references, we will use  $2\mathcal{F}$  instead of the  $\mathcal{F}$  of Eq. (4.12); in other words, we work in terms of

$$2\mathcal{F} = \mathbf{H}_a \mathbf{M}^{-1ab} \mathbf{H}_b. \quad (4.13)$$

In the situation under study here, both the observation time and 1 day [the time scale for variations of  $a(t)$  and  $b(t)$ ] are vastly larger than the period of the sought-for GWs (typically  $10^{-2} - 10^{-3}$ s). In consequence, we can replace  $\cos^2 \phi(t)$ ,  $\sin^2 \phi(t)$ ,  $\cos \phi(t) \sin \phi(t)$  by their time average over one period, namely

$$\cos^2 \phi(t) \rightarrow \frac{1}{2}, \quad \sin^2 \phi(t) \rightarrow \frac{1}{2}, \quad (4.14)$$

and

$$\cos \phi(t) \sin \phi(t) \rightarrow 0. \quad (4.15)$$

As a result,

$$(h_1|h_3) \approx 0, \quad (h_1|h_4) \approx 0, \quad (h_2|h_3) \approx 0, \quad (h_2|h_4) \approx 0. \quad (4.16)$$

We next define the amplitude coefficients  $A$ ,  $B$  and  $C$  as

$$\frac{1}{2}A^X \equiv (h_1^X|h_1^X) = (h_3^X|h_3^X), \quad (4.17)$$

$$\frac{1}{2}B^X \equiv (h_2^X|h_2^X) = (h_4^X|h_4^X), \quad (4.18)$$

$$\frac{1}{2}C^X \equiv (h_1^X|h_2^X) = (h_3^X|h_4^X). \quad (4.19)$$

which, using Eqs. (2.76) and (2.18- 2.21), can be re-written as

$$A^X = \frac{2}{S_h^X(f_0)} \int_{t_0}^{t_p} (a^X(t))^2 dt, \quad (4.20)$$

$$B^X = \frac{2}{S_h^X(f_0)} \int_{t_0}^{t_p} (b^X(t))^2 dt, \quad (4.21)$$

$$C^X = \frac{2}{S_h^X(f_0)} \int_{t_0}^{t_p} a^X(t)b^X(t) dt. \quad (4.22)$$

We use these quantities to define

$$A \equiv \sum_{\mathbf{x}} A^X, \quad B \equiv \sum_{\mathbf{x}} B^X, \quad C \equiv \sum_{\mathbf{x}} C^X. \quad (4.23)$$

Note that in the definitions of  $A^X$ ,  $B^X$  and  $C^X$ , we have used the notation of CS[52], which differs from that of JKS[44] by a factor of  $T/S_h(f)$ , where  $T$  is the observation time.

Substituting the above quantities into the definition of Eq. (4.10), we obtain the generalized version of the matrix  $\mathbf{M}$ , namely

$$\mathbf{M} = \frac{1}{2} \begin{pmatrix} \mathcal{C} & \mathcal{O} \\ \mathcal{O} & \mathcal{C} \end{pmatrix}, \quad (4.24)$$

with

$$\mathcal{C} = \begin{pmatrix} A & C \\ C & B \end{pmatrix}, \quad \text{and} \quad \mathcal{O} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \quad (4.25)$$

The inverse of matrix  $\mathbf{M}$  is readily written down to be

$$\mathbf{M}^{-1} = \frac{2}{D} \begin{pmatrix} \mathcal{C}^{-1} & \mathcal{O} \\ \mathcal{O} & \mathcal{C}^{-1} \end{pmatrix}, \quad (4.26)$$

with

$$\mathcal{C}^{-1} = \begin{pmatrix} B & -C \\ -C & A \end{pmatrix}, \quad (4.27)$$

where  $D \equiv AB - C^2$ .

Returning to the general form of equation of  $\mathcal{F}$ -statistic (Eq. 4.13), we have

$$\begin{aligned} 2\mathcal{F} &= \mathbf{H}_a (\mathbf{M}^{-1})^{ab} \mathbf{H}_b \\ &= \frac{2}{D} \{B[(\mathbf{H}_1)^2 + (\mathbf{H}_3)^2] + A[(\mathbf{H}_2)^2 + (\mathbf{H}_4)^2] - 2C[\mathbf{H}_1\mathbf{H}_2 + \mathbf{H}_3\mathbf{H}_4]\}. \end{aligned} \quad (4.28)$$

If we define  $F_a$  and  $F_b$  as

$$2F_a \equiv \sqrt{(\mathbf{H}_1)^2 + (\mathbf{H}_3)^2} = \sum_{\mathbf{x}} (x^{\mathbf{x}} | h_1^{\mathbf{x}} - i h_3^{\mathbf{x}}) = 2 \sum_{\mathbf{x}} F_a^{\mathbf{x}}, \quad (4.29)$$

$$2F_b \equiv \sqrt{(\mathbf{H}_2)^2 + (\mathbf{H}_4)^2} = \sum_{\mathbf{x}} (x^{\mathbf{x}} | h_2^{\mathbf{x}} - i h_4^{\mathbf{x}}) = 2 \sum_{\mathbf{x}} F_b^{\mathbf{x}}, \quad (4.30)$$

where in addition

$$(h_1^{\mathbf{x}} - i h_3^{\mathbf{x}}) = a^{\mathbf{x}}(t) e^{-i\phi^{\mathbf{x}}(t)}, \quad (4.31)$$

$$(h_2^{\mathbf{x}} - i h_4^{\mathbf{x}}) = b^{\mathbf{x}}(t) e^{-i\phi^{\mathbf{x}}(t)}, \quad (4.32)$$

so that the result is

$$2F_a^{\mathbf{x}} = \frac{2}{S_h^{\mathbf{x}}(f_0)} \int_{t_0}^{t_p} x^{\mathbf{x}}(t) a^{\mathbf{x}}(t) e^{-i\phi^{\mathbf{x}}(t)} dt, \quad (4.33)$$

$$2F_b^{\mathbf{x}} = \frac{2}{S_h^{\mathbf{x}}(f_0)} \int_{t_0}^{t_p} x^{\mathbf{x}}(t) b^{\mathbf{x}}(t) e^{-i\phi^{\mathbf{x}}(t)} dt. \quad (4.34)$$

The  $\mathcal{F}$ -statistic can then be written as

$$2\mathcal{F} = \frac{8}{D} \{B|F_a^2| + A|F_b^2| - 2C\Re(F_a F_b^*)\}. \quad (4.35)$$

Note that, again, the multi-detector components of  $\mathcal{F}$ -statistic (Eq. 4.35), namely  $F_a, F_b$  and  $A, B, C$  are all simply *sums* over detector-specific quantities  $F_a^{\mathbf{x}}, F_b^{\mathbf{x}}$  and  $A^{\mathbf{x}}, B^{\mathbf{x}}, C^{\mathbf{x}}$ , which can be calculated individually and then combined.

## 4.2 Discretization of the $\mathcal{F}$ -statistic's components (LALDemod)

In practice, data analysis is performed numerically with discrete data. This calls for a discretized expression for the  $\mathcal{F}$ -statistic derived in the previous section. The first technical note to derive such an expression for the case of a single detector, was done by Siemens [74]. We will briefly repeat his analysis here, generalizing the  $\mathcal{F}$ -statistic for multiple detectors; also, in [74], Siemens assumed that over the bandwidth of the signal  $S_h(f)$  is constant and equal to  $S_h(f_0)$  (just as JKS did), where  $f_0$  is the frequency of the signal at  $t = 0$ . However while we will assume that  $S_h(f)$  is constant in each SFT, it can differ from one SFT to the other.

We can write the discrete times of which the data is sampled as

$$t_{\alpha,j} = (\alpha - 1)T_{\text{sft}} + j\Delta t, \quad (4.36)$$

with  $j = 1, \dots, N$ , and  $\alpha = 1, \dots, M$  and  $T_{\text{sft}} = N\Delta t$ , where  $N$  is the number of time-steps per SFT,  $M$  is the number of SFTs and  $T_{\text{sft}}$  is the duration of each SFT.

As a first step towards discretization, let us calculate the Amplitude Modulation coefficients. Since the formalism is the same for all three coefficients  $A, B$  and  $C$ , we will give a detailed description for  $A$ , and only the results for  $B$  and  $C$ . In Eqs. (4.17 and 4.20),  $A$  was defined as

$$\begin{aligned} A &= 2 \sum_{\mathbf{x}} (h_1^{\mathbf{x}} | h_1^{\mathbf{x}}) \\ &= 2 \times \sum_{\mathbf{x}} \frac{2}{S_h^{\mathbf{x}}(f_0)} \int_{t_0}^{t_p} h_1^{\mathbf{x}}(t) h_1^{\mathbf{x}}(t) dt. \end{aligned} \quad (4.37)$$

Note that, by our assumption that the spectral density  $S_h(f)$  of the noise is constant in each SFT, we can move it into the summation over different SFTs, and by using Eq. (2.18), we obtain

$$\begin{aligned} A &= 4 \sum_X \sum_{l=1}^{NM^X} \frac{h_{1l}^X h_{1l}^X}{S_{h_l}^X(f)} \Delta t \\ &= 4 \sum_X \sum_{\alpha=1}^{M^X} \frac{a_\alpha^X a_\alpha^X}{S_{h_\alpha}^X(f)} \sum_{j=1}^N \cos \phi_{\alpha j}^X \cos \phi_{\alpha j}^X \Delta t. \end{aligned} \quad (4.38)$$

Using the approximation given in Eq. (4.14), we have

$$\begin{aligned} \sum_{j=1}^N \cos \phi_j^X \cos \phi_j^X \Delta t &\approx \frac{N}{2} \Delta t \\ &= \frac{T_{\text{sft}}}{2}; \end{aligned} \quad (4.39)$$

inserting this expression into Eq. (4.38), the result is

$$\begin{aligned} A &\approx 2 T_{\text{sft}} \sum_X \sum_{\alpha=1}^{M^X} \frac{(a_\alpha^X)^2}{S_{h_\alpha}^X(f)} \\ &= 2 T_{\text{sft}} \sum_X \sum_{\alpha=1}^{M^X} \left( \frac{a_\alpha^X}{\sqrt{S_{h_\alpha}^X(f)}} \right)^2. \end{aligned} \quad (4.40)$$

Note that  $T_{\text{sft}}$  is going to be fixed for data of different detectors. However, since we may have different amount of data for different detectors, the number of SFTs,  $M$ , can vary. Therefore, in the summation, we will give  $M$  an index  $x$ , indicating that it depends on the detector in question.

As was mentioned before, the  $B$  and  $C$  coefficients can be derived following the same procedure as for  $A$ . The results are

$$B = 2 T_{\text{sft}} \sum_X \sum_{\alpha=1}^{M^X} \left( \frac{b_\alpha^X}{\sqrt{S_{h_\alpha}^X(f)}} \right)^2, \quad (4.41)$$

$$C = 2 T_{\text{sft}} \sum_X \sum_{\alpha=1}^{M^X} \left( \frac{a_\alpha^X}{\sqrt{S_{h_\alpha}^X(f)}} \frac{b_\alpha^X}{\sqrt{S_{h_\alpha}^X(f)}} \right). \quad (4.42)$$

In order to simplify these expressions, we introduce the variables

$$\hat{a}_\alpha^X \equiv \left( \frac{a_\alpha^X}{\sqrt{S_{h_\alpha}^X(f)}} \right), \quad (4.43)$$

$$\hat{b}_\alpha^X \equiv \left( \frac{b_\alpha^X}{\sqrt{S_{h_\alpha}^X(f)}} \right), \quad (4.44)$$

as well as

$$\hat{A}^X \equiv \sum_{\alpha=1}^{M^X} (\hat{a}_\alpha^X)^2, \quad (4.45)$$

$$\hat{B}^X \equiv \sum_{\alpha=1}^{M^X} (\hat{b}_\alpha^X)^2, \quad (4.46)$$

$$\hat{C}^X \equiv \sum_{\alpha=1}^{M^X} \hat{a}_\alpha^X \hat{b}_\alpha^X. \quad (4.47)$$

Using these variables, the Amplitude Modulation coefficients can be written as

$$A = 2 T_{\text{sft}} \sum_{\mathbf{X}} \hat{A}^{\mathbf{X}}, \quad (4.48)$$

$$B = 2 T_{\text{sft}} \sum_{\mathbf{X}} \hat{B}^{\mathbf{X}}, \quad (4.49)$$

$$C = 2 T_{\text{sft}} \sum_{\mathbf{X}} \hat{C}^{\mathbf{X}}. \quad (4.50)$$

The above expressions will give us the following form for the coefficient  $D$ :

$$\begin{aligned} D &= AB - C^2 \\ &= 4 T_{\text{SFT}}^2 \sum_{\mathbf{X}} (\hat{A}^{\mathbf{X}} \hat{B}^{\mathbf{X}} - (\hat{C}^{\mathbf{X}})^2) \\ &= 4 T_{\text{SFT}}^2 \sum_{\mathbf{X}} \hat{D}^{\mathbf{X}}. \end{aligned} \quad (4.51)$$

Next, we will calculate the quantities  $F_a$  and  $F_b$ , which depend on the detector orientations and positions. From Eq. (4.33), we derive the discretized version of  $F_a^{\mathbf{X}}$ ,

$$F_a^{\mathbf{X}} = \sum_{\alpha=1}^{M^{\mathbf{X}}} \frac{a_{\alpha}^{\mathbf{X}}}{S_{h_{\alpha}}^{\mathbf{X}}(f)} \Delta t \left( \sum_{j=1}^N x_{\alpha,j}^{\mathbf{X}}(t) e^{-i\phi_{\alpha,j}} \right), \quad (4.52)$$

where we have made the assumption that the duration of a SFT (in our case 30 minutes) is such that the amplitude modulation functions  $a(t)$  and  $b(t)$  (the quantities used in Eqs. (2.18) and (2.19) which have been defined in [44]) do not change significantly. We define the quantity inside parenthesis of Eq. (4.52) to be the SFT, which is the Fourier transform of our data for this short time-baseline ( $N\Delta t$ )

$$\tilde{x}_{\alpha}^{\mathbf{X}}(f) = \sum_{j=1}^N x_{\alpha,j}^{\mathbf{X}}(t) e^{-i2\pi f t_{\alpha,j}}. \quad (4.53)$$

Recalling the definition of Discrete Fourier Transform (DFT) and its inverse from Eqs. (2.43) and (2.44)

$$\tilde{x}_k = \sum_{j=0}^{N-1} x_j e^{-i2\pi jk/N}, \quad (4.54)$$

with its inverse

$$x_j = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{x}_k e^{i2\pi jk/N}, \quad (4.55)$$

we define the Discrete Fourier Transform (DFT) for this case as

$$\tilde{x}_{\alpha,k}^{\mathbf{X}}(f) = \sum_{j=1}^N x_{\alpha,j}^{\mathbf{X}}(t) e^{-i2\pi kj/N}. \quad (4.56)$$

We translate between Eqs. (4.56) and (4.53) by equating the arguments of the exponential as

$$f_k t_{\alpha,j} = kj/N, \quad (4.57)$$

with  $t_j = j\Delta t$  and  $T_{\text{sft}} = N\Delta t$ , giving us that  $f_k = k/N\Delta t$  and therefore  $\Delta f = 1/T_{\text{sft}}$ , which is the frequency resolution of the data.

The inverse DFT of our data is then given by

$$x_{\alpha,j}^{\mathbf{X}}(t) = \frac{1}{N} \sum_{k=1}^N \tilde{x}_{\alpha,k}^{\mathbf{X}}(f) e^{i2\pi kj/N}. \quad (4.58)$$

Substituting the  $x_{\alpha,j}^X(t)$  into Eq. (4.52), we are led to

$$F_a^X = \frac{\Delta t}{N} \sum_{\alpha=1}^{M^X} \frac{a_{\alpha}^X}{S_{h_{\alpha}}^X(f)} \sum_{k=1}^N \tilde{x}_{\alpha,k}^X(f) \sum_{j=1}^N \left( e^{(-i\phi_{\alpha,j} + i2\pi kj/N)} \right). \quad (4.59)$$

The arguments in the exponential relate to the evolution of the phase of our short segments of data (SFT). This segment is short enough that the phase evolution of the signal can be considered as linear within it. Therefore we can make a Taylor expansion of  $\phi_{\alpha,j}$  around the middle of each segments (SFTs) as

$$\phi_{\alpha,j} = \phi_{\alpha,1/2} + \dot{\phi}_{\alpha,1/2}(t_{\alpha,j} - t_{\alpha,1/2}), \quad (4.60)$$

with  $t_{\alpha,j}$  as defined in Eq. (4.36), so that

$$\phi_{\alpha,j} = \phi_{\alpha,1/2} + \dot{\phi}_{\alpha,1/2}(j - N/2)\Delta t = \phi_{\alpha,1/2} + \dot{\phi}_{\alpha,1/2}(j/N - 1/2)T_{\text{sft}}. \quad (4.61)$$

Inserting this expression for the Taylor expansion into Eq. (4.59), we have

$$\begin{aligned} F_a^X &= \frac{\Delta t}{N} \sum_{\alpha=1}^{M^X} \frac{a_{\alpha}^X}{S_{h_{\alpha}}^X(f)} \sum_{k=1}^N \tilde{x}_{\alpha,k}^X(f) \sum_{j=1}^N e^{-i(\phi_{\alpha,1/2} + \dot{\phi}_{\alpha,1/2}(j/N - 1/2)T_{\text{sft}} - 2\pi kj/N)} \\ &= \frac{\Delta t}{N} \sum_{\alpha=1}^{M^X} \frac{a_{\alpha}^X}{S_{h_{\alpha}}^X(f)} e^{-i(\phi_{\alpha,1/2} - \dot{\phi}_{\alpha,1/2}T_{\text{sft}}/2)} \sum_{k=1}^N \tilde{x}_{\alpha,k}^X(f) \sum_{j=1}^N e^{-i(\dot{\phi}_{\alpha,1/2}T_{\text{sft}} - 2\pi k)j/N}. \end{aligned} \quad (4.62)$$

We can simplify this further with the help of the following quantities:

$$\zeta \equiv \phi_{\alpha,1/2} - \dot{\phi}_{\alpha,1/2}T_{\text{sft}}/2, \quad (4.63)$$

$$\varepsilon \equiv \dot{\phi}_{\alpha,1/2}T_{\text{sft}} - 2\pi k. \quad (4.64)$$

Then Eq. (4.62) can be brought into the form

$$F_a^X = \frac{\Delta t}{N} \sum_{\alpha=1}^{M^X} \frac{a_{\alpha}^X}{S_{h_{\alpha}}^X(f)} e^{-i\zeta} \sum_{k=1}^N \tilde{x}_{\alpha,k}^X(f) \sum_{j=1}^N e^{-i\varepsilon j/N}. \quad (4.65)$$

Note that

$$\sum_{j=1}^N e^{-i\varepsilon j/N} = \frac{1 - e^{-i\varepsilon}}{1 - e^{-i\varepsilon/N}}. \quad (4.66)$$

and that, furthermore, in the limit of very large  $N$ , the exponent in the denominator will be small, so that

$$\begin{aligned} \frac{1 - e^{-i\varepsilon}}{1 - e^{-i\varepsilon/N}} &\approx \frac{1 - e^{-i\varepsilon}}{1 - (1 - i\varepsilon/N)} = \frac{iN}{\varepsilon}(e^{-i\varepsilon} - 1) \\ &= N\left(\frac{\sin \varepsilon}{\varepsilon} - i\frac{1 - \cos \varepsilon}{\varepsilon}\right). \end{aligned} \quad (4.67)$$

Therefore,

$$F_a^X \approx \Delta t \sum_{\alpha=1}^{M^X} \frac{a_{\alpha}^X}{S_{h_{\alpha}}^X(f)} e^{-i\zeta} \sum_{k=1}^N \tilde{x}_{\alpha,k}^X(f) P_{\alpha,k}, \quad (4.68)$$

where

$$P_{\alpha,k} = \left(\frac{\sin \varepsilon}{\varepsilon} - i\frac{1 - \cos \varepsilon}{\varepsilon}\right) \quad (4.69)$$

is known as the Dirichlet kernel; it has a sharp peak around  $\varepsilon = 0$  which, according to Eq. (4.64) would correspond to a sharp peak close to the value of the frequency index  $k^* = \dot{\phi}_{\alpha,1/2}T_{\text{sft}}/2\pi$ . This suggests a

further expansion, this time around  $k^*$ ; keeping only the lowest-order terms in  $\Delta k$ , the result for  $F_a$  as well as  $F_b$  is

$$F_a^X \approx \Delta t \sum_{\alpha=1}^{M^X} \frac{a_\alpha^X}{S_{h_\alpha}^X(f)} Q_\alpha \sum_{k=k^*-\Delta k}^{k^*+\Delta k} \tilde{x}_{\alpha,k}^X(f) P_{\alpha,k}, \quad (4.70)$$

$$F_b^X \approx \Delta t \sum_{\alpha=1}^{M^X} \frac{b_\alpha^X}{S_{h_\alpha}^X(f)} Q_\alpha \sum_{k=k^*-\Delta k}^{k^*+\Delta k} \tilde{x}_{\alpha,k}^X(f) P_{\alpha,k}, \quad (4.71)$$

where

$$Q_\alpha = e^{-i\zeta}. \quad (4.72)$$

There is still one further step to take before we reach the final form of  $F_{a,b}$ . In the end, we are going to compute the  $\mathcal{F}$ -statistic numerically. As it stands, the quantities to be evaluated (and later, combined) in that calculation are of widely disparate orders of magnitude. The solution is to re-define the quantities needed for calculating the  $\mathcal{F}$ -statistic so that all of them are of order unity. The expectation value of input data,  $x(t)$ , has a value of order  $\sim h_0 \sim 10^{-23}$ ; we will normalize it by the noise floor, resulting in a quantity of order unity. Using the Wiener-Khinchine theorem (Eq. 2.52), we can estimate power spectral density of the noise as

$$S_{h_\alpha}(f) \sim \frac{1}{T_{\text{sft}}} E[|\tilde{x}_\alpha|^2] \implies \sqrt{S_{h_\alpha}(f) T_{\text{sft}}} \sim E[|\tilde{x}|], \quad (4.73)$$

where  $E$  denotes the expectation value. Therefore

$$E \left[ \tilde{X}_\alpha(f) \right] = E \left[ \frac{\tilde{x}_\alpha(f)}{\sqrt{S_{h_\alpha}(f) T_{\text{sft}}}} \right] \sim \mathcal{O}(1). \quad (4.74)$$

In consequence, we normalize the SFT as  $\tilde{X}_{\alpha,k}^X(f) = \tilde{x}_{\alpha,k}^X(f) / \sqrt{S_{h_\alpha}^X(f) T_{\text{sft}}}$ . A correspondingly normalized version of  $F_a$  is

$$F_a^X \approx \Delta t \sqrt{T_{\text{sft}}} \sum_{\alpha=1}^{M^X} \frac{a_\alpha^X}{\sqrt{S_{h_\alpha}^X(f)}} Q_\alpha \sum_{k=k^*-\Delta k}^{k^*+\Delta k} \frac{\tilde{x}_{\alpha,k}^X(f)}{\sqrt{S_{h_\alpha}^X(f) T_{\text{sft}}}} P_{\alpha,k}. \quad (4.75)$$

Using Eq. (4.43) and taking  $\tilde{X}'_{\alpha,k}(f) = \Delta t \tilde{X}_{\alpha,k}^X(f)$ , we arrive at the final form of  $F_a$ ,

$$F_a^X \approx \sqrt{T_{\text{sft}}} \sum_{\alpha=1}^{M^X} \hat{a}_\alpha^X Q_\alpha \sum_{k=k^*-\Delta k}^{k^*+\Delta k} \tilde{X}'_{\alpha,k}(f) P_{\alpha,k}. \quad (4.76)$$

Equivalently,  $F_b$  can be written as

$$F_b^X \approx \sqrt{T_{\text{sft}}} \sum_{\alpha=1}^{M^X} \hat{b}_\alpha^X Q_\alpha \sum_{k=k^*-\Delta k}^{k^*+\Delta k} \tilde{X}'_{\alpha,k}(f) P_{\alpha,k}. \quad (4.77)$$

Our data analysis software (which will be described in more detail in the following chapter) computes the components  $P_{\alpha,k}$ ,  $\tilde{X}'_{\alpha,k}(f)$ ,  $Q_\alpha$ ,  $\hat{a}_\alpha^X$  and  $\hat{b}_\alpha^X$  and uses them to calculate  $F_{a,b}$  and finally  $\mathcal{F}$ -statistic.

### 4.3 Semi-Analytical calculation of the $\mathcal{F}$ -statistic

Sometimes it is useful to estimate (predict) the value of  $\mathcal{F}$ -statistic without analyzing the data. The importance of this shows itself when we want to validate our analysis' process. If the computed  $\mathcal{F}$ -statistic were not close to the predicted one we would not trust our analysis. A detailed discussion of this is saved for the next chapter, where we validate our software.

In our work on predicting (semi-analytic calculation) the  $\mathcal{F}$ -statistic, we improved on the previous computational procedure. Previously, a value of noise floor was inserted by hand in the calculation of

semi-analytic  $\mathcal{F}$ , however, our improvement was to let the software compute the noise floor from our data.

Our method of calculating the predicted  $\mathcal{F}$ -statistic is to replace the quantity  $H_a$  in Eq. (4.11) by the equivalent expression defined in Eq. (2.83), namely

$$H_a = A_{MLE}^b M_{ab}. \quad (4.78)$$

The result is

$$\begin{aligned} \log \Lambda &= A_{MLE}^a M_{ab} A_{MLE}^b - \frac{1}{2} A_{MLE}^a M_{ab} A_{MLE}^b \\ &= \frac{1}{2} A_{MLE}^a M_{ab} A_{MLE}^b, \end{aligned} \quad (4.79)$$

so that

$$2\mathcal{F} = A_{MLE}^a M_{ab} A_{MLE}^b; \quad (4.80)$$

the quantity on the right hand side is the *square of the signal-to-noise ratio*,  $\rho^2$ . According to Eq. (2.86) we can write the above equation as

$$E[2\mathcal{F}] = 4 + A_{MLE}^a M_{ab} A_{MLE}^b. \quad (4.81)$$

As we can see directly from this expression, just taking the detector independent elements of the search ( $h_0, \psi, \iota, \phi_0$ ), and the amplitude modulation coefficients ( $a(t)$  and  $b(t)$ ), we can predict the possible values of the  $\mathcal{F}$ -statistic without doing a full analysis. However, as stated before, still we need the data to estimate the (level of) noise floor.

## 4.4 Parameter estimation for the signal

In this section, we will discuss techniques for estimating the signal amplitude parameters ( $h_0, \mu (= \cos \iota), \Phi_0$  and  $\psi$ ) given the four amplitudes,  $A^a$  (Eqs. 2.22-2.25), based on the formalism used in [73]. To start with, let us define new quantities

$$\mathcal{A}_s^2 \equiv \sum_{a=1}^4 (A^a)^2 = A_+^2 + A_\times^2, \quad (4.82)$$

$$\mathcal{D}_a \equiv A^1 A^4 - A^2 A^3 = A_+ A_\times. \quad (4.83)$$

Solving for  $A_+$  and  $A_\times$ , we have

$$2A_+^2 = \mathcal{A}_s^2 \pm \sqrt{\mathcal{A}_s^4 - 4\mathcal{D}_a^2}, \quad 2A_\times^2 = \mathcal{A}_s^2 \pm \sqrt{\mathcal{A}_s^4 - 4\mathcal{D}_a^2}; \quad (4.84)$$

two solutions for  $A_+^2$  and two for  $A_\times^2$ . According to Eqs. (2.14) and (2.15), we have  $|A_+| \geq |A_\times|$ , and therefore we will assign the '+' solution to  $A_+$  and the '-' to  $A_\times$ . On the other hand, we know from Eq. (2.14) that the sign of  $A_+$  is always positive, so that

$$A_+ = \left( \frac{\mathcal{A}_s^2 + \sqrt{\mathcal{A}_s^4 - 4\mathcal{D}_a^2}}{2} \right)^{1/2}. \quad (4.85)$$

For  $A_\times$ , we obtain the expression

$$A_\times = \pm \left( \frac{\mathcal{A}_s^2 - \sqrt{\mathcal{A}_s^4 - 4\mathcal{D}_a^2}}{2} \right)^{1/2}. \quad (4.86)$$

Note that

$$\mathcal{A}_s^2 - 2\mathcal{D}_a = (A^1 - A^4)^2 + (A^2 + A^3)^2 \geq 0, \quad (4.87)$$

and

$$\mathcal{A}_s^2 + 2\mathcal{D}_a = (A^1 + A^4)^2 + (A^2 - A^3)^2 \geq 0. \quad (4.88)$$

Therefore,

$$\mathcal{A}_s^4 - 4\mathcal{D}_a^2 = (\mathcal{A}_s^2 - 2\mathcal{D}_a)(\mathcal{A}_s^2 + 2\mathcal{D}_a) \geq 0, \quad (4.89)$$

which tells us that  $A_+$  and  $A_\times$  are always *real*, and we have

$$\sqrt{\mathcal{A}_s^4 - 4\mathcal{D}_a^2} = A_+^2 - A_\times^2 \geq 0. \quad (4.90)$$

Unfortunately, Eq. (4.86) on its own will not tell us the sign of  $A_\times$ ; for this, we will need to refer to Eq. (4.83): since  $A_+$  is always positive, the equation tells us that  $A_\times$  has the same sign as  $D_a$ .

Given these expressions for  $A_+$  and  $A_\times$ , we can use Eqs. (2.12) and (2.13) to determine the values of  $h_0$  and  $\cos \iota$ , namely

$$h_0 = A_+ + \sqrt{A_+^2 - A_\times^2}, \quad (4.91)$$

$$\cos \iota = \frac{A_\times}{h_0}. \quad (4.92)$$

However, according to Itoh [75], these expression, as they stand, are prone to numerical errors: when  $A_+ \cong A_\times$  (or  $\mu \cong \pm 1$ ), the error  $dh_0$  could be unexpectedly large even for very small errors  $dA_{+, \times}$ , as

$$dh_0 = dA_+ + \frac{A_+ dA_+ - A_\times dA_\times}{\sqrt{A_+^2 - A_\times^2}}. \quad (4.93)$$

While this shows that  $dh_0$  is finite, we need to make some estimate of its magnitude. Suppose for simplicity's sake that  $dA_+ = 0$ . Defining  $\gamma = A_\times/A_+$ , assume that the true value of this is  $\gamma = \gamma_*$ , while the value we obtain in our measurements is  $\gamma_s = \gamma_* + \epsilon$ . Then [75]

$$\frac{h_0(\gamma_* + \epsilon)}{A_+} - \frac{h_0(\gamma_*)}{A_+} = \sqrt{1 - \gamma_*^2} - \sqrt{1 - \gamma_s^2}; \quad (4.94)$$

for  $\gamma_* = 1$ , we obtain

$$\frac{h_0(\gamma_* + \epsilon) - h_0(\gamma_*)}{h_0(\gamma_*)} \cong \sqrt{2\epsilon}. \quad (4.95)$$

Therefore, an error of, say 1% in  $\gamma$  ( $\epsilon = 0.01$ ), leads to an error of 10% in  $h_0$  ( $\sqrt{\epsilon} \sim 0.1$ ).

As Itoh [75] pointed out, to avoid this numerically “error-unsafe” quantity, we will define alternative, “error-safe” quantities as

$$\mathcal{A}_s^2 = \sum_{a=1}^4 (A^a)^2 = \frac{1}{4} h_0^2 (1 + 6\mu^2 + \mu^4), \quad (4.96)$$

$$A_+ = \frac{1}{2} (1 + \mu^2) h_0. \quad (4.97)$$

Furthermore, let us introduce a new variable

$$\beta = \frac{A_+}{A_\times} = \frac{1 + \mu^2}{2\mu}. \quad (4.98)$$

Using Eqs. (2.22)-(2.25), it can be shown that

$$A^1 \cos 2\psi + A^2 \sin 2\psi = A_+ \cos \phi_0, \quad (4.99)$$

$$A^4 \cos 2\psi - A^3 \sin 2\psi = A_\times \cos \phi_0, \quad (4.100)$$

$$A^1 \cos \phi_0 - A^3 \sin \phi_0 = A_+ \cos 2\psi, \quad (4.101)$$

$$A^4 \cos \phi_0 + A^2 \sin \phi_0 = A_\times \cos 2\psi. \quad (4.102)$$

Thus, dividing Eq. (4.99) by (4.100), and Eq. (4.101) by (4.102), and using the newly defined  $\beta$  (Eq. 4.98), it follows that  $\psi$  and  $\phi_0$  are given by

$$\psi = \frac{1}{2} \arctan \left( \frac{\beta A^4 - A^1}{\beta A^3 + A^2} \right), \quad (4.103)$$

$$\phi_0 = \arctan \left( \frac{A^1 - \beta A^4}{A^3 + \beta A^2} \right). \quad (4.104)$$



It needs to be mentioned that the values of  $A^a$  are invariant under simultaneous transformations  $\{\psi \rightarrow \psi + \pi/2, \phi_0 \rightarrow \phi_0 + \pi\}$ , and, in consequence, under  $\{\psi \rightarrow \psi + \pi, \phi_0 \rightarrow \phi_0 + 2\pi\}$ . However, there is a sign ambiguity of the  $A^a$  as we replace  $\phi_0 \rightarrow \phi_0 + \pi$ , which we must seek to resolve by finding which solution for the  $\psi$  and  $\phi_0$  is acceptable from a physics point of view. The simplest way is to reconstruct the  $A^a$  using the computed parameters  $\psi, \phi_0, A_+$  and  $A_-$ ; if the sign of  $A^1$  differs from that of Eq. (2.22), we simply replace  $\phi_0 \rightarrow \phi_0 + \pi$ .

Note that still there are some special cases for which we cannot find unique values for the amplitude parameters. For example, the coordinate singularity  $h_0 = 0$  or  $\mu = \pm 1$  corresponds to  $A^a = 0$  or  $A^1 = \pm A^4, A^2 = \pm A^3$ , respectively; in either case, we cannot determine the parameters ( $\psi, \phi_0, h_0$  and  $\mu$ ) because we do not have enough independent variables – if  $A^a = 0$ , then we can take  $h_0 = 0$  and any  $\psi, \phi_0, \mu$  we prefer. If  $A^1 = \pm A^4, A^2 = \pm A^3$ , then we can determine  $h_0, \mu = \pm 1$  and  $\psi \pm \phi_0$  uniquely, but we cannot determine  $\phi$  and  $\phi_0$  independently.

## 4.5 Using noise weights

As can be seen from the final expressions for the A, B and C (Eqs. 4.45 - 4.47) as well as for  $F_a$  and  $F_b$  (Eqs. 4.76 and 4.77), we need to normalize the  $a(t)$  and  $b(t)$ , dividing by the noise spectral density  $S_h(f)$ . To implement this, we introduce the *Noise Weight* which enables us to keep all the numbers we use in the numerical computation of order unity. Also it takes care of the non-stationarity in our data, as described below. For simplicity's sake, let us here ignore the summation  $\sum_X$  over the different detectors and concentrate on the case of a single detector; we can readily reconstitute the full expression afterwards. Let the noise weight be defined as

$$w_\alpha = \frac{S_{h\alpha}^{-1}(f)}{\hat{S}} \quad (4.105)$$

with

$$\hat{S} \equiv \langle S_{h\alpha}^{-1}(f) \rangle = \frac{1}{M} \sum_{\alpha}^{M-1} S_{h\alpha}^{-1}(f). \quad (4.106)$$

According to this definition,  $\hat{S}$  is the mean value of the inverse of noise floor for our entire data, so its contribution is fixed in the calculation of the *weight*. Then, Eq. (4.105) tells that the larger value of  $S_{h\alpha}(f)$  gives smaller value of its weight and vice versa.

Therefore, wherever we have  $S_{h\alpha}^{-1}$ , we can substitute

$$S_{h\alpha}^{-1}(f) = w_\alpha \hat{S}; \quad (4.107)$$

notably, we can write

$$A = 2 T_{\text{sft}} \hat{S} \sum_{\alpha=0}^{M-1} (\sqrt{w_\alpha} a_\alpha)^2 = 2 T_{\text{sft}} \hat{S} \bar{A} \quad (4.108)$$

$$B = 2 T_{\text{sft}} \hat{S} \sum_{\alpha=0}^{M-1} (\sqrt{w_\alpha} b_\alpha)^2 = 2 T_{\text{sft}} \hat{S} \bar{B} \quad (4.109)$$

$$C = 2 T_{\text{sft}} \hat{S} \sum_{\alpha=0}^{M-1} (\sqrt{w_\alpha} a_\alpha \sqrt{w_\alpha} b_\alpha) = 2 T_{\text{sft}} \hat{S} \bar{C}. \quad (4.110)$$

$$\begin{aligned} D &= AB - C^2 \\ &= 4 T_{\text{SFT}}^2 \hat{S}^2 (\bar{A}\bar{B} - \bar{C}^2) \\ &= 4 T_{\text{SFT}}^2 \hat{S}^2 \bar{D} \end{aligned} \quad (4.111)$$

$$F_a = \sqrt{T_{\text{sft}}} \sqrt{\hat{S}} \sum_{\alpha=0}^{M-1} \sqrt{w_\alpha} a_\alpha Q_\alpha \sum_k \tilde{X}_{\alpha k} P_{\alpha k} = \sqrt{T_{\text{sft}}} \sqrt{\hat{S}} \bar{F}_a \quad (4.112)$$

$$F_b = \sqrt{T_{\text{sft}}} \sqrt{\hat{S}} \sum_{\alpha=0}^{M-1} \sqrt{w_\alpha} b_\alpha Q_\alpha \sum_k \tilde{X}_{\alpha k} P_{\alpha k} = \sqrt{T_{\text{sft}}} \sqrt{\hat{S}} \bar{F}_b \quad (4.113)$$

Inserting these expression into the Eq. (4.35) for the  $\mathcal{F}$ -statistic, we have

$$2\mathcal{F} = \frac{8\{B|F_a^2| + A|F_b^2| - 2C\Re(F_a F_b^*)\}}{D} \quad (4.114)$$

$$= \frac{8\{2 T_{\text{sft}} \hat{S} \bar{B}|F_a^2| + 2 T_{\text{sft}} \hat{S} \bar{A}|F_b^2| - 4 T_{\text{sft}} \hat{S} \bar{C}\Re(F_a F_b^*)\}}{4 T_{\hat{S}FT}^2 \hat{S}^2 \bar{D}} \quad (4.115)$$

$$= \frac{4\{\bar{B}|F_a^2| + \bar{A}|F_b^2| - 2\bar{C}\Re(F_a F_b^*)\}}{T_{\text{sft}} \hat{S} \bar{D}} \quad (4.116)$$

$$= \frac{4\{\bar{B} T_{\text{sft}} \hat{S} |\bar{F}_a^2| + \bar{A} T_{\text{sft}} \hat{S} |\bar{F}_b^2| - 2 T_{\text{sft}} \hat{S} \bar{C}\Re(\bar{F}_a \bar{F}_b^*)\}}{T_{\text{sft}} \hat{S} \bar{D}} \quad (4.117)$$

$$= \frac{4\{\bar{B}|\bar{F}_a^2| + \bar{A}|\bar{F}_b^2| - 2\bar{C}\Re(\bar{F}_a \bar{F}_b^*)\}}{\bar{D}} \quad (4.118)$$

Thus, our final formula for the  $\mathcal{F}$ -statistic including the noise weight turns out to be

$$\mathcal{F} = \frac{2\{\bar{B}|\bar{F}_a^2| + \bar{A}|\bar{F}_b^2| - 2\bar{C}\Re(\bar{F}_a \bar{F}_b^*)\}}{\bar{D}} \quad (4.119)$$

As is readily seen, this is no major change – where previously, we had to use  $a_\alpha(t)$  in our calculation, we now use  $\sqrt{w_\alpha} a_\alpha(t)$ .

These expressions for A, B, C,  $F_a$ ,  $F_b$  and  $\mathcal{F}$  are the ones currently in use in the ComputeFStatistic\_v2 code; the actual implementation of noise-weight calculation in that code is due to Badri Krishnan and Reinhard Prix.

# Chapter 5

## Observational Results

The LAL (LIGO Algorithm Library) is a free distribution software package developed by the LIGO Scientific Collaboration (LSC) in order to do gravitational wave data analysis [59, 60]. As part of this thesis work, we have extended and generalized an existing code from the LAL (*ComputeFStatistic.c* henceforth “CFS\_v1”) for computing the  $\mathcal{F}$ -statistic (the detection statistic, as discussed in the preceding chapters); the result, a code named *ComputeFStatistic\_v2* (“CFS\_v2”) has since been integrated into the LAL. The main improvements are that the CFS\_v2 can perform a coherent multi-IFO (Interferometric Observatory) search; also, it uses SFT-specific noise-weights that allow a better handling of non-stationary noise. Another extension in CFS\_v2 is that it can estimate the signal parameters and outputs the estimated parameters of the loudest event – an important feature when it comes to analyzing the hardware-injected data, and to providing estimates for direct comparison with the parameters chosen for that data. Due to use of modified LALDemod, CFS\_v2 is also slightly faster than CFS\_v1.

In the first part of this chapter, we describe the validation procedures of our CFS\_v2, namely the results of a comparison between its output and that of the previous (and extensively tested) version CFS\_v1. Also, we give an account of test runs using hardware-injected data from the fourth LIGO Science run (S4), where we have compared the parameters of the injected signals as estimated by using our software to the true parameter values used for generating these signals. In the second part of this chapter, we present our results from using CFS\_v2 to analyze the S5 data and set the upper limit on the gravitational wave amplitude from 33 known isolated pulsars. We compare our results with those obtained in the Time Domain Search (TDS), which uses a completely different approach. These pulsars are known from radio-astronomical data; their parameters have been taken from the ATNF (Australia Telescope National Facility) catalogue [78].

All results presented and discussed in this chapter were generated using CFS\_v2; the only occasion on which CFS\_v1 has been used was for comparing the performance of the two codes. The CVS version of the CFS\_v2 is given in Appendix C.

### 5.1 Software validation

#### 5.1.1 Consistency software checks

As mentioned above, CFS\_v2 is a new code we have developed for our data analysis purpose; before it can be used in any applications, it is essential that its validity be checked as rigorously as possible.

A natural first step is to compare the results obtained with the new code with those of CFS\_v1, which has undergone extensive testing and meticulous reviews by the LSC pulsar-group committee (among other things, this software is now being used as the central search code in the Einstein@Home project). Since both CFS\_v1 and CFS\_v2 compute the  $\mathcal{F}$ -statistic, we can use them both to compute the  $\mathcal{F}$ -statistic for the same data sets and compare the results. For this purpose, we have generated simulated data (in SFT format) of the duration of one day with pure noise (that is, without any signal) using *lalapps\_Makefakedata* (a LAL based software to generate both time series data and data in SFT format) [59]. For this simulated data, we have computed 100,000 values of  $\mathcal{F}$ -statistic.

In Fig. (5.1) the absolute and relative differences between CFS\_v2 and CFS\_v1 are shown. The small

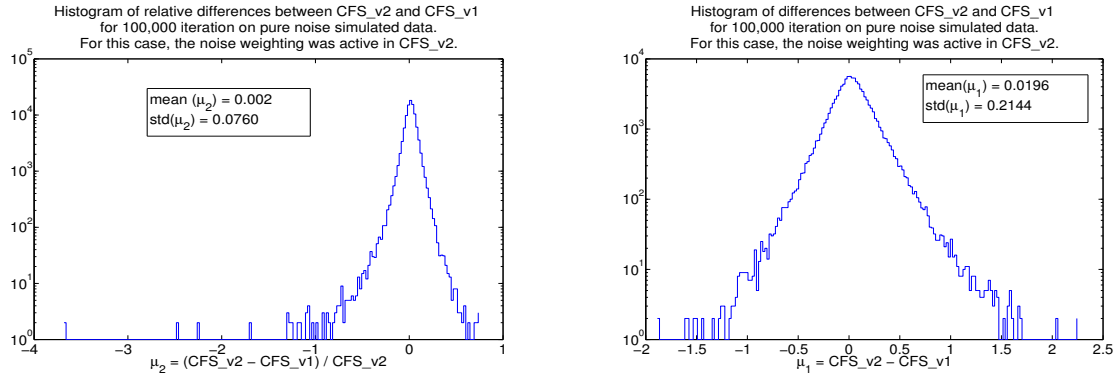


Figure 5.1: Histogram of relative (left) and absolute (right) differences between CFS\_v2 and CFS\_v1. The results are based on the randomly distributed search parameters which in each run were the same for both of the codes. The results shown are for 100,000 trials, and based on the H1 detector sensitivity.

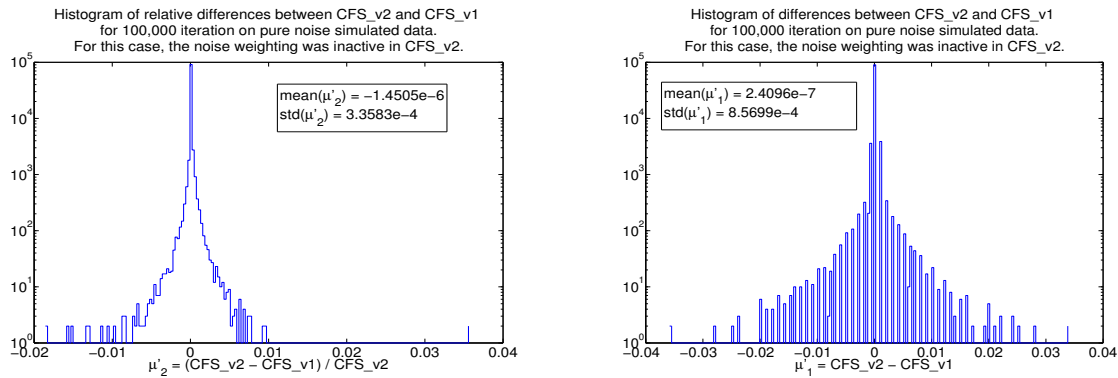


Figure 5.2: Histogram of relative (left) and absolute (right) differences between CFS\_v2 and CFS\_v1 when the option of noise weighting in CFS\_v2 is disabled. The results are based on the randomly distributed search parameters which in each run were the same for both of the codes. The results shown are for 100,000 trials, and based on the H1 detector sensitivity.

population of points at very large relative mismatch have values of  $\mathcal{F}$ -statistic which are significantly lower than 4. For the worst case shown in Fig. (5.1), the high relative mismatch is  $\sim -3.7$  from values of  $\text{CFS}_{v1}=0.122$  and  $\text{CFS}_{v2}=0.026$ . Even if we ignore the few points with a very high relative mismatch, we see that the bulk of the distribution lies between -50% to 50% with a standard deviation of 8% (0.21 in the absolute difference). Such differences cannot be ascribed to different rounding errors in the two codes; they are due to the different way that the two codes treat the noise. As we have already explained in the previous chapter, in CFS\_v2 we have added a noise weight procedure for noise floor estimation. If we disable the noise weighting, the two codes yield consistent results, as it is shown in Fig. (5.2) (a more details of this investigation can be found in [79]): the bulk of the distribution now lies around  $10^{-6}$  with a standard deviation of order  $10^{-4}$ .

Fig. (5.3) shows a comparison between the two codes for the signal plus noise cases starting from very low SNR ( $=1$ ) to a high value ( $=100$ ). The relative mismatch between these two codes reduces significantly with growing SNR because the values of the  $\mathcal{F}$ -statistic are dominated by the signal rather than the noise, and the signal is indeed treated in the two codes in the same manner. At very low SNR we observe the distribution of Fig. (5.1) whereas, for example for SNR=100, we see that the mismatch is distributed in the range of just -2% to 2%. The same kind of investigation for the signal-only case shows consistency among the two codes.

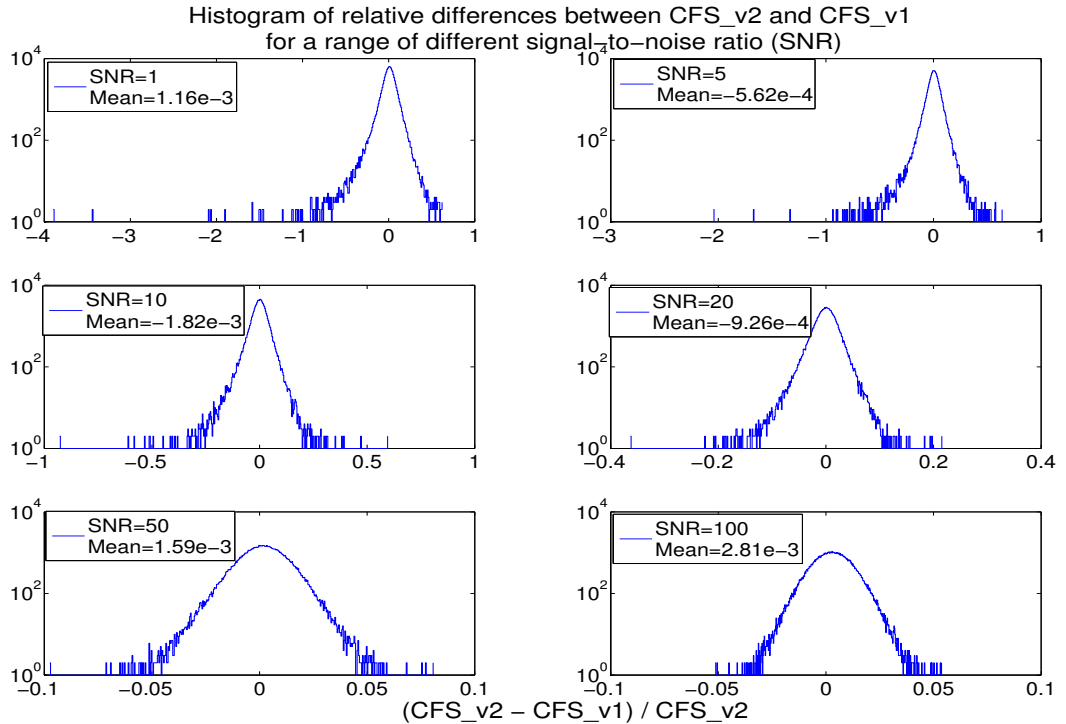


Figure 5.3: Histogram of relative differences between CFS\_v2 and CFS\_v1 for the case of signal plus noise with different SNR. The results are based on the randomly distributed search parameters which in each run were the same for both of the codes. The results shown are for 100,000 trials, and based on the H1 detector sensitivity.

### 5.1.2 Consistency with expectation for pure noise

As was discussed earlier, in chapter 2, for Gaussian noise the distribution of the  $\mathcal{F}$ -statistic ( $2\mathcal{F}$ ) follows a  $\chi^2$  distribution with 4 degrees of freedom, with the expectation value for the perfectly matched case given as (Eq. 2.86)

$$\langle 2\mathcal{F} \rangle = 4 + \rho^2. \quad (5.1)$$

Here, the non-centrality parameter,  $\rho^2 = (h|h)$ , is the square of optimal signal-to-noise ratio (SNR); for the case of a pure noise data, where we have no signal, so that  $\rho^2 = 0$ , we should have

$$\langle 2\mathcal{F} \rangle = 4. \quad (5.2)$$

By the above criterion, if we plot the histogram of the output  $\mathcal{F}$ -statistic for a significantly large number of trials (in our case, 100,000 trials on Gaussian noise), it should follow a  $\chi^2$  distribution with 4 degrees of freedom. The plots given in Fig. (5.4), appear to show that this is indeed the case but let us look at this in more detail. We repeat 129600 trials for different detectors and their combinations. Each time we compute the average value of  $2\mathcal{F}$  and its standard deviation. The results are shown in Table (5.1). Both the mean values and the standard deviation appear to suffer from a positive bias: the expected values are 4.0 and 2.8284 for the mean and standard deviation respectively.

To get a better statistic, we have repeated the 129600 trials for each detector and detector combination 50 times. Table (5.2) shows the average value of the mean and of the standard deviation: over the 50 experiments a very significant deviation from the expectations is evident. This is due to a positive bias in the finite sample estimator of the noise power spectral density used in our software [76] due to the fact that, quite generally,

$$E \left[ \frac{1}{x} \right] \neq \frac{1}{E[x]}. \quad (5.3)$$

This bias in noise estimation due to the Eq. (5.3) depends on the size of the running median window being used to estimate the noise PSD (a similar effect has been reported on Stack-Slide techniques using

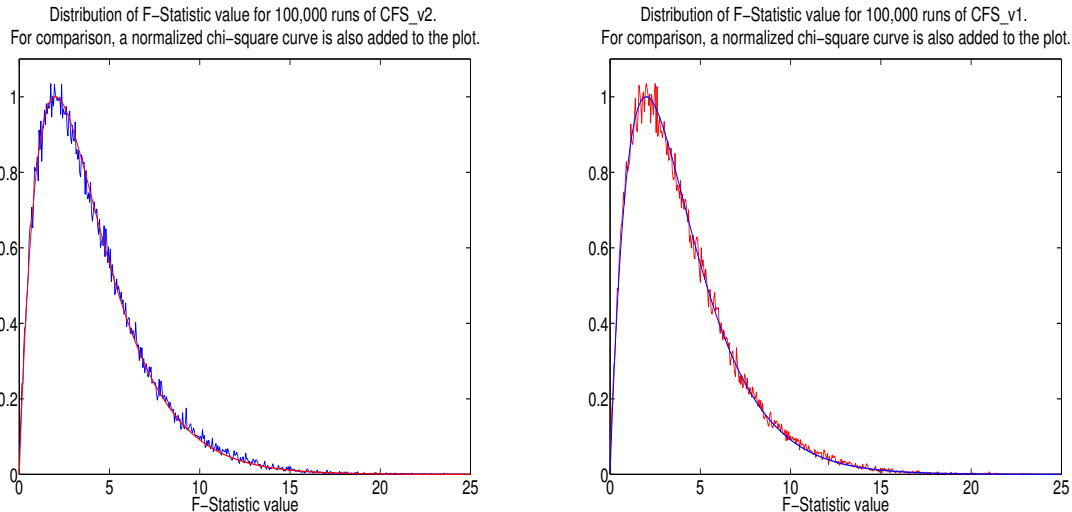


Figure 5.4: Probability distribution function (pdf) of  $\mathcal{F}$ -statistic for both CFS\_v2 (left) and CFS\_v1 (right). The pdf of a  $\chi^2$  distribution with 4 degrees of freedom (smooth curves) are also shown for comparison. The output is based on the sensitivity of the H1 detector with 100,000 trials.

CFS\_v1 [77].) The discrepancies measured here are consistent with this predicted bias, which, for the running median window size used here ( $N = 50$ ) is expected to be  $\approx 0.2\%$ . We have verified that an increase in the size of the running median window will result in a decrease in the amount of observed bias. This, however, also reduces the search efficiency, as with the larger window size the noise power spectral density varies over the window, whereas we want to keep the window size as small as possible to have a constant  $S_n(f)$  across the window.

### 5.1.3 Consistency with expectation for signal plus noise

In section 4.3, we reviewed the theoretical formalism for estimating the expected value of  $\mathcal{F}$ -statistic in a data set containing a known signal. We have developed a code, *PredictFStat* (PFS) [59], that performs such estimations.

For the test, we have performed a Monte Carlo search on 100,000 independent random signals, each of which consists of 20 hours of simulated data plus noise. The signal parameters ( $f_0, f^{(k)}, \alpha, \delta, A_+, A_\times, \psi, \Phi_0$ ) are randomly distributed. For each signal, we calculated the expected  $\mathcal{F}$ -statistic first by using PFS, then we calculated it by using CFS\_v2 (hereinafter referred to simply as CFS), and then compared the results. Since the standard deviation of  $2\mathcal{F}$  distribution is proportional to the  $\langle 2\mathcal{F} \rangle$ , with a higher value of  $2\mathcal{F}$  we will have a larger mismatch between the expectation value, PFS, and the actual value, CFS, of  $2\mathcal{F}$  as shown in Fig. (5.5). For simplicity we will restrict ourself to the value of  $2\mathcal{F} \leq 1500$  in our analysis.

Fig. (5.6) shows the mismatch between CFS and PFS for more than 29000 runs and in each run we used the same data to calculate the CFS and PFS. The mean value of the mismatch is positive, with a value of about 6%. This is consistent with the expectations [54]: we expect to loose SNR due to the way that the  $\mathcal{F}$ -statistic calculations have been implemented, optimizing for speed.

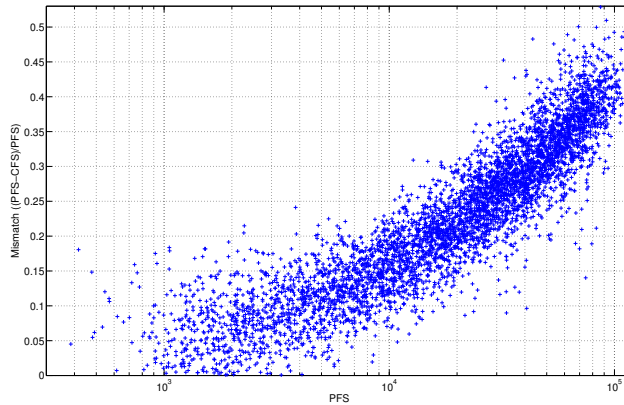
### 5.1.4 Parameter estimation

One of the main purpose of any gravitational waves data analysis is to extract the wave's parameters. These are the parameters of the wave that, in fact, give us the information about the mechanism and nature of the source. Therefore, as a next validation process, we perform the test on the estimation of simulated data. To do so, we have generated 5 artificial pulsars with SNR of 1, 5, 10, 20 and 50. We have generated one day of data and selected the signal strength to satisfy the above values of SNR in each data set. The other parameters, such as  $\cos \iota, \psi, \Phi_0$  and the frequency and location of the source have been chosen randomly. We have computed the maximum likelihood estimators for the signal parameters (the red lines showing the extracted parameters in Fig. (5.7)) and have also derived the posterior PDFs

| IFO      | Mean(CFS_v2) | Std(CFS_v2) |
|----------|--------------|-------------|
| H1       | 4.0643       | 2.8695      |
| H2       | 4.0671       | 2.8816      |
| L1       | 4.0754       | 2.8883      |
| G1       | 4.1065       | 2.9070      |
| H1H2     | 4.0748       | 2.8681      |
| H1L1     | 4.0624       | 2.8709      |
| H1G1     | 4.0804       | 2.8954      |
| H2L1     | 4.0716       | 2.8787      |
| H2G1     | 4.0839       | 2.9000      |
| L1G1     | 4.0705       | 2.8757      |
| H1H2L1   | 4.0535       | 2.8613      |
| H1H2G1   | 4.0647       | 2.8658      |
| H1L1G1   | 4.0819       | 2.9005      |
| H2L1G1   | 4.0758       | 2.8826      |
| H1H2L1G1 | 4.0477       | 2.8684      |

Table 5.1: Mean and standard deviation of 129600 outputs of  $\mathcal{F}$ -statistic for pure noise data using CFS\_v2.

| IFO      | $\langle Mean \rangle$ | Std(Mean) |
|----------|------------------------|-----------|
| H1       | 4.060018               | 0.020039  |
| H2       | 4.061292               | 0.021610  |
| L1       | 4.055642               | 0.015677  |
| G1       | 4.058904               | 0.021304  |
| H1H2     | 4.061833               | 0.019438  |
| H1L1     | 4.064068               | 0.019175  |
| H1G1     | 4.062683               | 0.014066  |
| H2L1     | 4.060648               | 0.015691  |
| H2G1     | 4.062865               | 0.014752  |
| L1G1     | 4.055830               | 0.014303  |
| H1H2L1   | 4.059847               | 0.018078  |
| H1H2G1   | 4.058302               | 0.012509  |
| H1L1G1   | 4.058325               | 0.015382  |
| H2L1G1   | 4.061718               | 0.016247  |
| H1H2L1G1 | 4.059154               | 0.016055  |

Table 5.2: Mean and standard deviation for the average of 50 computations of the same kind as reported in Table (5.1) (each computation with 129600 outputs of  $2\mathcal{F}$  for pure noise data using CFS\_v2, as above).Figure 5.5: Relation of mismatch between PFS and CFS and the value of  $\mathcal{F}$ -statistic. A linear fit in this plot would show that the mismatch is proportional to the value of  $2\mathcal{F}$  for  $2\mathcal{F} > 1500$ . Number of runs = 5,000.

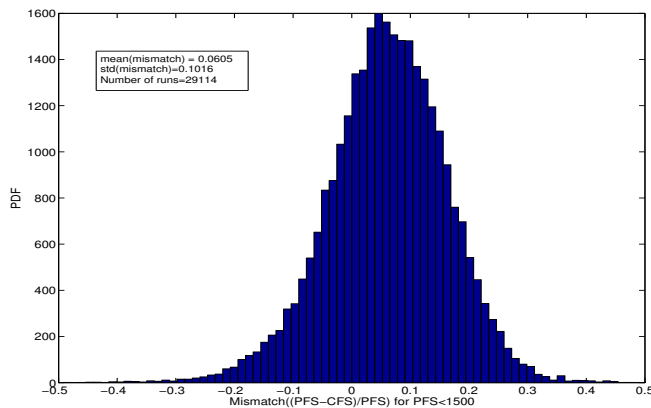


Figure 5.6: Relative mismatch between PFS and CSF  $((\text{PFS}-\text{CFS})/\text{PFS})$  for more than 29000 different runs. In each run the search parameters were chosen randomly. The results presented here are from the value of  $2\mathcal{F} \leq 1500$ .

| PULSAR | $\dot{f}$ | $f$     | $h_0$     | SNR    | $\iota$ | $\psi$ | $\Phi_0$ | DEC     | RA     |
|--------|-----------|---------|-----------|--------|---------|--------|----------|---------|--------|
| 0      | -4.15E-12 | 265.58  | 4.933e-25 | 9.0    | 0.65    | 0.77   | 2.66     | -0.9812 | 1.2488 |
| 1      | -3.00E-10 | 849.07  | 4.240e-24 | 26.6   | 1.09    | 0.36   | 1.28     | -0.5140 | 0.6526 |
| 2      | -1.37E-13 | 575.16  | 8.037e-24 | 71.4   | 2.76    | -0.22  | 4.03     | 0.0601  | 3.7569 |
| 3      | -1.46E-17 | 108.86  | 3.255e-23 | 748.8  | 1.65    | 0.44   | 5.53     | -0.5836 | 3.1132 |
| 4      | -2.54E-08 | 1402.11 | 4.562e-22 | 1708.3 | 1.29    | -0.65  | 4.83     | -0.2176 | 4.8867 |
| 5      | -4.03E-18 | 52.81   | 9.701e-24 | 47.4   | 1.09    | -0.36  | 2.23     | -1.4632 | 5.2818 |
| 6      | -6.73E-09 | 148.44  | 2.769e-24 | 70.7   | 1.73    | 0.47   | 0.97     | -1.1418 | 6.2614 |
| 7      | -1.12E-09 | 1220.93 | 1.319e-23 | 50.1   | 0.71    | 0.51   | 5.25     | -0.3569 | 3.8995 |
| 8      | -8.65E-09 | 193.95  | 3.175e-23 | 569.9  | 1.50    | 0.17   | 5.89     | -0.5833 | 6.1329 |
| 9      | -1.45E-17 | 763.85  | 8.130e-24 | 56.0   | 2.24    | -0.01  | 1.01     | 1.3210  | 3.4712 |

Table 5.3: The parameters of the S4 hardware injected isolated pulsars. The reference time for all of these parameter has been set to 793130413 GPS.

of the signal strength in a Bayesian framework (this approach is being explained in more details in the section corresponding to the Bayesian upper limit). The PDFs look consistent and the extracted values in all cases show that the estimations were quite correct. If we set 95% upper limits on  $h_0$  using these PDFs, we find that in all cases the upper limits are larger than the injected signal strength, as expected. Independently of the shape of the PDF one can always set an upper limit. However, in an actual search one would do this only if the PDFs of  $h_0$  did *not* show a significant peak (as in the case of our injections at SNR 1 and 5). In these circumstances the upper limit value may also be taken as an indicator of the sensitivity of the search.

As to analyze these PDFs better, we look at the correlations between the signal parameters, for example between  $h_0$  and  $\cos \iota$  as shown in Fig. (5.7). All of these PDFs seem to be plausible. This is also shown that as the signal gets stronger the correlation between  $h_0$  and  $\mu$  gets narrower, as expected.

We also performed tests on hardware-injected data. This is a somewhat more complete (end-to-end) assessment of the entire pipeline. For this purpose pulsar signals have been injected in hardware in the LIGO detectors during the fourth LIGO science run (S4), which took place between noon CST on 22 February and midnight CST on 23 March 2005 [80], corresponding to a duration of 29.5 days. During this time the duty cycle was about 72%, translating into 20.9 days' worth data with 1004 SFTs of each 1800 seconds. The hardware-injected data contained the simulated signals from two binary pulsars and ten isolated pulsars. Since the present version of our software is not set up to deal with binaries, we focused on the isolated pulsars. Table (5.3) gives an overview of the parameters for these pulsars [81]. The frequency ranges have been selected to be in the range of LIGO detectors sensitivity. The typical strain sensitivity in this run was within a factor of two of the design goals [80].

PULSAR3 is one of the strongest injected pulsars with an SNR  $\approx 750$ . We searched for this signal in



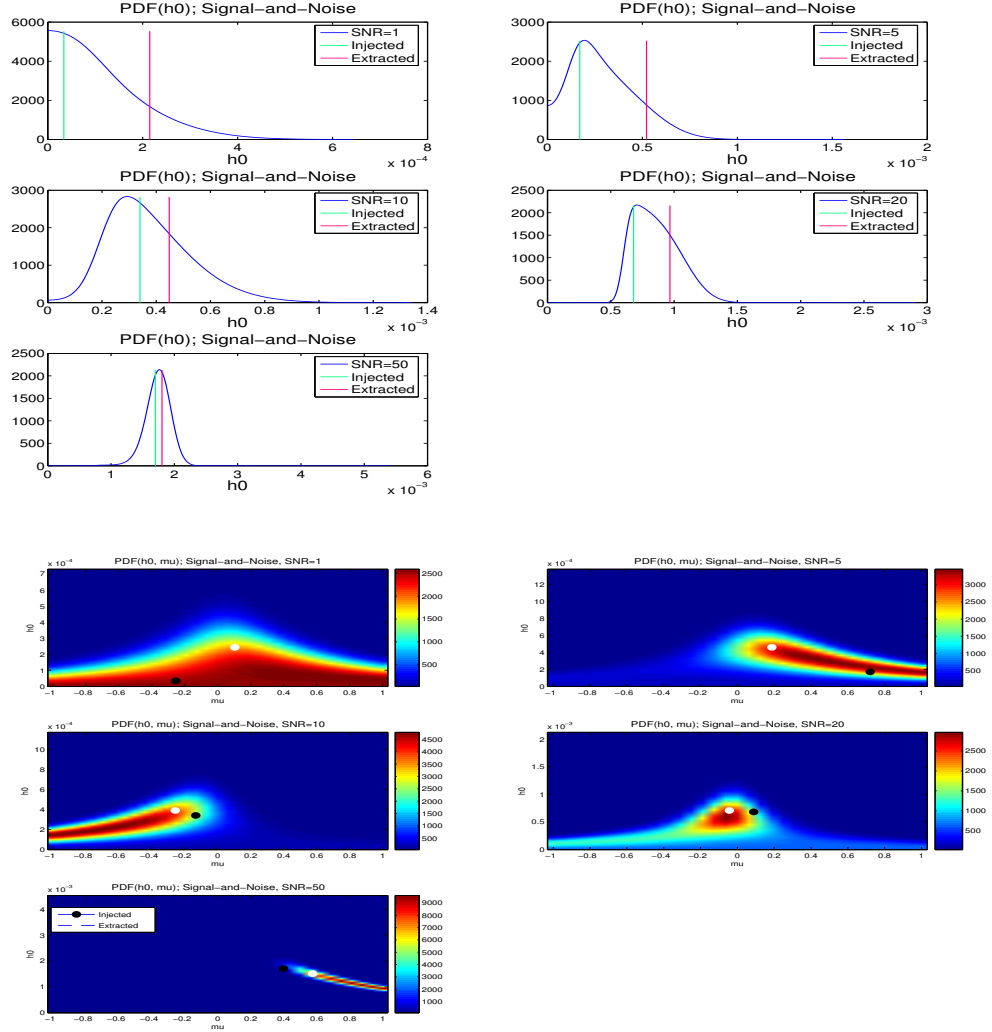


Figure 5.7: Parameter estimation of simulated signals with different SNR. The top plots are the probability distributions (PDFs) for the signal amplitudes,  $h_0$ , in the case of signal plus noise by marginalizing the other three parameters ( $\cos \iota$ ,  $\psi$  and  $\Phi_0$ ) with the injected and extracted values indicated by green and red vertical lines respectively. The bottom plots are the correlated PDFs of  $h_0$  and  $\cos \iota$  by marginalizing over the other two parameters,  $\psi$  and  $\Phi_0$ .

a very narrow patch in the longitude ( $\alpha$ ) and latitude ( $\delta$ ), with a size of 0.001 radian. The location resolution was set to  $10^{-5}$  radian which is quite high resolution, and in fact, showed correlation over sky pixels, as expected. The result can be seen in Fig. (5.8) (top figure): the right location is indeed included and it shows the highest value of detection statistic, but the overall result is a narrow region in the sky. We repeated the search in a larger region of 0.5 radian in both longitude and latitude and reducing the resolution to 0.001 radian. The result can be seen in Fig. (5.8) (bottom figure). Clearly, the pulsar was now detected in its exact location, the highest  $\mathcal{F}$ -statistic pixel is very well identified. For clearance, we have also added a zoomed area around the actual location of the pulsar, as shown in top left of bottom figure of Fig. (5.8).

Figs. (5.9) show the posterior PDFs for  $h_0$  and  $\mu$  and Figs. (5.10) for  $\psi$  and  $\Phi_0$ . The distributions show agreement between the computed PDFs and the injected parameter values for all but the strongest pulsars: PULSAR3 with SNR  $\sim 750$ , PULSAR4 with SNR  $\sim 1700$  and PULSAR8 with SNR  $\sim 570$ . The results from PULSAR9 shows a deviation probably due to a statistical fluctuation. The reason for the discrepancies for the strongest signals is due to the fact that the parameter estimation does not function correctly in the case of extremely high SNR events because the noise estimate that is used to normalize the data is contaminated by the signal. In practice it is extremely unlikely that we will encounter signals

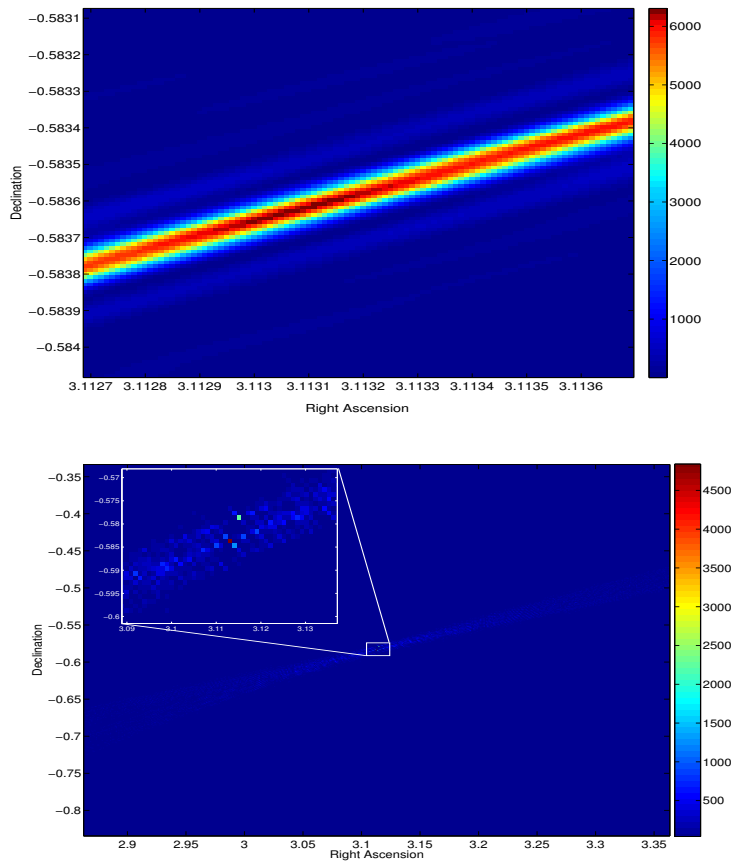


Figure 5.8: Detection of PULSAR3 in a 0.001 radian (top) and 0.5 radian (bottom) patch in the sky location.

with such high SNR, so this is, in practice, not a problem. A way to avoid this would be to not perform the estimate of the noise by using the data itself, but rather to input a constant upfront estimate of the noise floor. This is not a desirable way forward because the non-stationary and non-gaussian behavior of the instrument would not be tracked and this would impact the sensitivity of our searches towards our most important class of signals – the weak ones.

Using the posterior probabilities for  $h_0$ , we can set upper limits on the signal amplitudes. The results for the 95% upper limits on  $h_0$  of S4 data with hardware injections are shown in Table (5.4). Apart from extremely high SNR conditions, our upper limits are correct and consistent with being higher than the injected signals. As discussed above, our search method is designed to search for signals in a noise-dominated regime. In the signal-dominated regime, our technique still detects the signal but the parameter estimation is not reliable.

## 5.2 S5 data - Upper limit for known isolated pulsars

As was briefly mentioned before, due to the current sensitivity of our detectors it is unlikely that we can lay claim to any detection, but we can still set an upper limit (UL) on the signal strength for any particular targeted source – also an indication of our detectors’ sensitivity. In this section, we present the results of our evaluation of upper limits on the signal amplitude of the gravitational wave emission from 33 isolated radio pulsars, based on data from the fifth science runs (S5) of the LIGO gravitational wave detectors, namely on the data from LIGO Hanford Observatory (labeled as H1 (4km) and H2 (2km)) and LIGO Livingston Observatory (labeled as L1).

The analysis resulting in upper limit is, in fact, a targeted search that depends on having available initially all information about our sources: frequency, frequency derivatives, position of the source and a reference

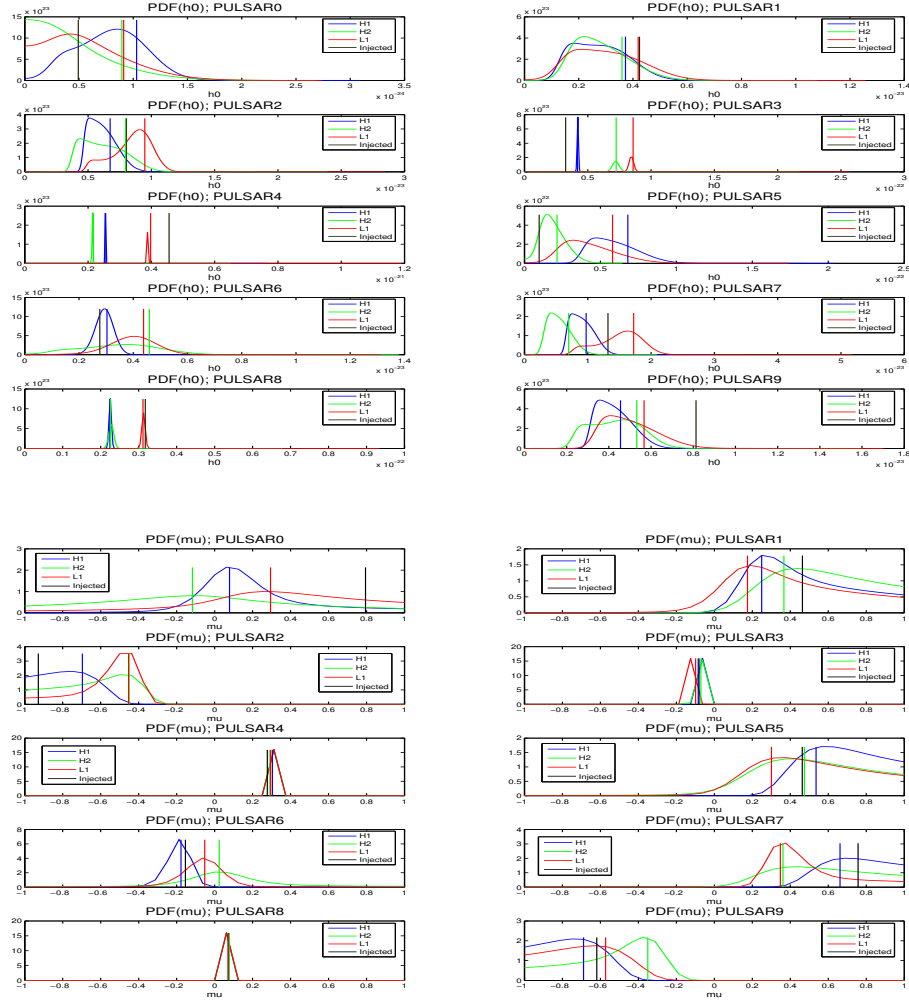


Figure 5.9: Posterior probability of signal amplitudes,  $h_0$ , and the inclination angle,  $\mu = \cos i$ , of the 10 hardware injected pulsars in S4 data. The vertical lines indicate the estimated values of the parameter.

time in which all of this information have been observed. For our purpose, we have taken all necessary information for our 33 selected pulsars from astronomical data obtained via radio observations by the Australian Telescope National Facility (ATNF)[78]. We have performed this search with two different approaches, *Frequentist* and *Bayesian*, comparing our results with the Bayesian ULs for the same data resulting from the Time Domain Search (TDS) group at the University of Glasgow.

Our first result presented in this section is the Frequentist ULs using 8 months of S5 data of L1 IFO taken between 15 November 2005 (roughly the beginning of the S5 run) and 15 July 2006. Comparison with the Bayesian UL of TDS shows a large mismatch between the two results; in order to exclude as its cause differences resulting from the different approaches taken, we performed an UL search with a Bayesian technique, using the H1 data.

At the time of analysis, the data from the measurement period indicated above had significant gaps, especially where the L1 detector was concerned. For H1, H2 and L1, respectively, we had 145 days, 161 days and 114 days of data, corresponding to duty factors of 60%, 67% and 47% . Among the above-mentioned IFOs, for the Bayesian analysis we mostly focus on the data of the H1 detector: with twice the arm-length, it is more sensitive than the H2, and the proportion of available data is significantly greater than for L1.

Repeating our evaluation of the UL for  $h_0$  using a Bayesian scheme, we found that there were still significant differences between our results and those of TDS. We conjectured that this might be due to the different timescales of the data, and decided that, in addition to using the same technique as TDS

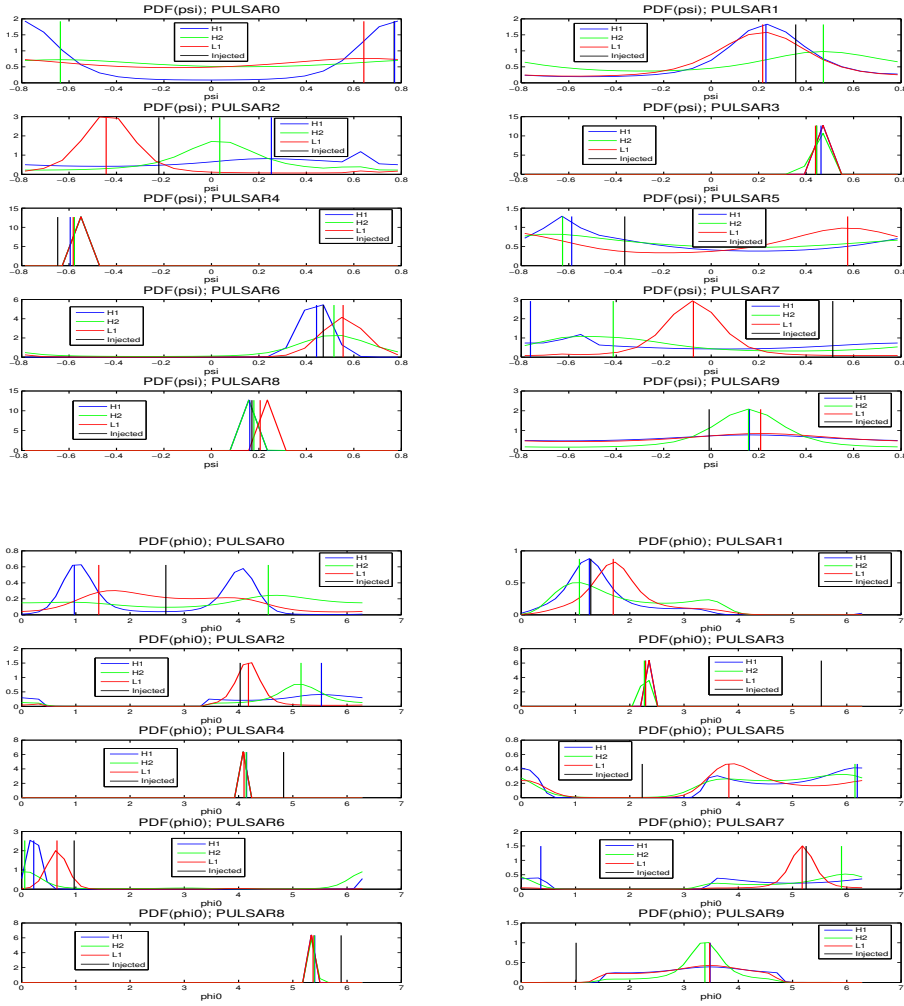


Figure 5.10: Posterior probability of wave polarization angle,  $\psi$ , and the initial phase,  $\Phi_0$ , of the 10 hardware injected pulsars in S4 data. The vertical lines indicate the estimated values of the parameter.

(Bayesian), we should also use correspondingly shorter data sets (11 days). Still, the relative discrepancy amounted to about  $\pm 30\%$ , so we decided to work on simulated noise data. The result was that the relative differences dropped to about  $\pm 6.5\%$ , a strong indication that the previously observed large discrepancies were due to the different way that non-gaussian and non-stationary artifacts are handled by the different searches.

### 5.2.1 Pulsar selection

In selecting the pulsars for our upper-limits estimate, the key criterion was their frequency. Since our detectors' sensitivity is poor in the range of GW frequencies below 50Hz, the natural starting point was to restrict ourselves to pulsars with a rotational frequency above 25Hz (since the frequency of the emitted GWs is twice the source's rotational frequency). This left about 170 pulsars in the ATNF catalogue; among these, only 68 are isolated pulsars. We focus on those, since our software cannot yet handle the search for binary pulsars. Next, we considered the reference time at which the position and the period of pulsars have been measured, figuring that more recent observations would correspond to better accuracy in the estimates of both the current position and the frequency drift. The calculations in our software are based on GPS time, so in particular, observations made before GPS=1 (6 January 1980) were not acceptable. This reduced our roster of acceptable candidate to 45 pulsars, which we used for our calculation of Frequentist ULs. However, as we wanted to compare our results with those of the TDS group, we applied the Bayesian technique only to those 33 pulsars that were both in our list and in theirs

| Pulsar  | $f_0$       | $h_0$ (Injected) | H1          | H2          | L1          |
|---------|-------------|------------------|-------------|-------------|-------------|
| PULSAR0 | 265.5771052 | 4.933e-25        | 1.28373e-24 | 1.15501e-24 | 1.27244e-24 |
| PULSAR1 | 849.0832962 | 4.240e-24        | 4.46999e-24 | 4.63768e-24 | 5.15681e-24 |
| PULSAR2 | 575.163573  | 8.036e-24        | 8.09504e-24 | 9.20007e-24 | 1.05282e-23 |
| PULSAR3 | 108.8571594 | 3.255e-23        | 4.27917e-23 | 7.60553e-23 | 8.75419e-23 |
| PULSAR4 | 1403.163331 | 4.562e-22        | 2.55350e-22 | 2.14594e-22 | 3.87887e-22 |
| PULSAR5 | 52.80832436 | 9.700e-24        | 8.57537e-23 | 3.38050e-23 | 7.39505e-23 |
| PULSAR6 | 148.7190257 | 2.769e-24        | 3.45567e-24 | 5.72055e-24 | 5.32309e-24 |
| PULSAR7 | 1220.979581 | 1.319e-23        | 1.22817e-23 | 8.79135e-24 | 1.99240e-23 |
| PULSAR8 | 194.3083185 | 3.175e-23        | 2.28816e-23 | 2.37439e-23 | 3.17600e-23 |
| PULSAR9 | 763.8473165 | 8.130e-24        | 5.73382e-24 | 6.22571e-24 | 7.22950e-24 |

Table 5.4: Upper limit on the strength of signal on S4 hardware injected data of different detectors.

[82, 83]. In their work, they have explained a very precise details of the pulsars selection procedure that we do not need to repeat them here again. The details of the parameters of the 45 selected pulsars are shown in the table quoted as Fig. (5.11).

### 5.2.2 Frequentist Upper Limits

The frequentist probability of an event represents the expected frequency of occurrence of that event. The result for our upper limits depends crucially on the experimental data under examination. We have set frequentist upper limits on the amplitude of gravitational waves from various sources. The confidence value associated with these upper limits indicates the expected occurrence of detection statistics values more significant than the one that we have measured in the presence of signals whose amplitude is equal to the upper limit value.

To set the frequentist upper limit for the amplitude of gravitational waves, we use the  $\mathcal{F}$ -statistic as an optimal detection statistic. To start with, we need to assign a confidence level  $\mathcal{C}$  – roughly speaking, our criterion will then be that, for our “repeated measurements”, in  $\mathcal{C}$ -percent of the time the value of  $2\mathcal{F}$  is above a specified threshold.

Let us explain how this works in detail for the example of setting a 95% upper limit on  $h_0$ . For this, we need to find at which  $h_0$  it is true that 95% of the values of the  $\mathcal{F}$ -statistic are above the initial value  $2\mathcal{F}$  derived from the data. To do so, we proceed step by step as follows:

1. Compute the  $\mathcal{F}$ -statistic of a perfectly matched signal using the exact values for the signal parameters (such as frequency, longitude, latitude and frequency derivatives). Let us call the resulting value of the  $\mathcal{F}$ -statistic  $\mathcal{F}^*$ .
2. Estimate the signal amplitude,  $h_0$ , using our parameter estimation routine.
3. Take this  $h_0$  as the initial value of the search.
4. Since we assume that there is no signal in the data – that it is pure noise –, we can randomly assign arbitrary values to the other signal parameters (such as  $\phi_0$ ,  $\psi$  and  $\cos \iota$ ).
5. To determine the probability distribution of  $\mathcal{F}$ -statistic, we take a random frequency value with a band of 0.1Hz around the actual pulsar frequency (as was proposed in the S1 paper [54]) and inject the artificial signal. With this choice, we are sure to be on the safe side; we use a large amount of data (in order of several months up to a year), so that the 0.1Hz band will not lead to any spurious correlations between the search parameters.
6. After injection, compute the  $\mathcal{F}$ -statistic once again. Let us designate the resulting value of  $2\mathcal{F}$  as  $\mathcal{F}'$ ; store this value for later use.
7. Repeat the injections, computing of  $\mathcal{F}$ -statistic for 150 times. Save all resulting values of  $\mathcal{F}'$ . (The number of iterations used here is a heuristic value.)

|    | Freq              | Alpha             | Delta               | fldot         | f2dot      | f3dot   | refTimeOP | refTimeMJI | SpinAge   | JName             |
|----|-------------------|-------------------|---------------------|---------------|------------|---------|-----------|------------|-----------|-------------------|
| 1  | 50.5929585112000  | 5.20531394344503  | 0.573828129740219   | -7.480318E-12 | 2.924E-23  | NOVALUE | 483926410 | 49845.000  | 1.070E5   | J1952+3252        |
| 2  | 53.1065091260000  | 5.62853778656359  | 0.212335360534228   | -2.800000E-15 | NOVALUE    | NOVALUE | 292768133 | 47632.520  | 2.980E8   | J2129+1210G       |
| 3  | 55.6969441074200  | 5.03239073541471  | 0.177843989323391   | -5.223520E-12 | NOVALUE    | NOVALUE | 643939213 | 51697.000  | 1.690E5   | J1913+1011        |
| 4  | 59.0728600000000  | 1.72482163328339  | 0.0969191805607962  | NOVALUE       | NOVALUE    | NOVALUE | NOVALUE   | NOVALUE    | NOVALUE   | J0635+0533        |
| 5  | 59.5537757000000  | 1.45967505304477  | 0.384224810048590   | -7.458491E-10 | 2.438E-20  | NOVALUE | 816048013 | 53669.000  | 1.240E3   | J0534+2200 (Crab) |
| 6  | 69.3203466993000  | 2.785758765417540 | 0.00951972387207786 | -1.600000E-16 | NOVALUE    | NOVALUE | 756518413 | 53000.000  | 6.820E9   | J1038+0032        |
| 7  | 124.0523791916000 | 1.47388395301818  | -1.20728490977605   | -3.984544E-10 | 1.220E-20  | NOVALUE | 675417325 | 52061.330  | 4.930E3   | J0537-6910        |
| 8  | 152.0074920000000 | 5.67392001230687  | -0.404562433690380  | NOVALUE       | NOVALUE    | NOVALUE | 684115213 | 52161.997  | NOVALUE   | J2140+2310B       |
| 9  | 192.7244691280000 | 4.37065927243073  | 0.636247836976843   | NOVALUE       | NOVALUE    | NOVALUE | 295678949 | 47666.210  | NOVALUE   | J1641+3627A       |
| 10 | 221.3543839680000 | 5.02163339857005  | -1.04693313381492   | -2.360000E-14 | NOVALUE    | NOVALUE | 662342413 | 51910.000  | 1.490E8   | J1910-5959D       |
| 11 | 237.0765061126000 | 4.66035931911933  | -0.432434469180103  | 1.704000E-14  | NOVALUE    | NOVALUE | 580089612 | 50958.000  | NOVALUE   | J1748-2446C       |
| 12 | 239.2297465691400 | 5.02160669503249  | -1.04691061906757   | 2.288000E-14  | NOVALUE    | NOVALUE | 670118413 | 52000.000  | NOVALUE   | J1910-5959B       |
| 13 | 246.220578359594  | 4.58306357347041  | -0.402741531681774  | -6.126200E-16 | NOVALUE    | NOVALUE | 524966411 | 50320.000  | 6.370E9   | J1730-230A        |
| 14 | 296.5865451300000 | 5.62855471625733  | 0.212345657976815   | -1.060000E-15 | NOVALUE    | NOVALUE | 292768133 | 47632.520  | 4.450E9   | J2129+1210H       |
| 15 | 333.299812081480  | 4.31596431886465  | -1.20507863906860   | -5.560000E-16 | NOVALUE    | NOVALUE | 635558413 | 51600.000  | 9.510E9   | J1629-6902        |
| 16 | 347.416437932106  | 0.104018132760358 | -1.25795321422485   | 3.008400E-15  | NOVALUE    | NOVALUE | 635558413 | 51600.000  | NOVALUE   | J0024-7204C       |
| 17 | 360.127249394540  | 4.00795349939460  | 0.0364946601262762  | -2.673000E-15 | NOVALUE    | NOVALUE | 362966407 | 48445.000  | 2.130E9   | J1518+0205A       |
| 18 | 364.234475332780  | 1.88453514233707  | -1.19578127014323   | -9.883600E-16 | NOVALUE    | NOVALUE | 534038411 | 50425.000  | 5.840E9   | J0711-6830        |
| 19 | 367.646870835540  | 4.81568943375662  | -0.529900384994325  | -2.287072E-13 | 1.020E-24  | NOVALUE | 524880011 | 50319.000  | 2.550E7   | J1823-3021A       |
| 20 | 371.4485544032000 | 5.02309930060880  | 0.0179887595344551  | 1.380000E-16  | NOVALUE    | NOVALUE | 756518413 | 53000.000  | NOVALUE   | J1911+0101B       |
| 21 | 373.303339713676  | 0.105729079026088 | -1.25801307901819   | 2.390000E-16  | NOVALUE    | NOVALUE | 635558413 | 51600.000  | NOVALUE   | J0024-7204D       |
| 22 | 378.979742140380  | 5.02258896633551  | -1.04748692678658   | -1.580000E-16 | NOVALUE    | NOVALUE | 662342413 | 51910.000  | 3.800E10  | J1910-5959C       |
| 23 | 382.90182418620   | 5.28296032574068  | -0.233854746612543  | -3.540000E-16 | NOVALUE    | NOVALUE | 713318413 | 52500.000  | 1.720E10  | J2010-1323        |
| 24 | 385.713042033820  | 5.16685247178468  | 0.159228482054545   | -1.274200E-15 | NOVALUE    | NOVALUE | 749001613 | 52913.000  | 4.800E9   | J1944+0907        |
| 25 | 387.431373382060  | 2.72552797652999  | -0.127791611567622  | -1.390600E-15 | -6.000E-27 | NOVALUE | 585273612 | 51018.000  | 4.410E9   | J1024-0719        |
| 26 | 405.587794469976  | 5.60569666264274  | -0.593044728403452  | -1.689400E-15 | NOVALUE    | NOVALUE | 522201611 | 50288.000  | 3.800E9   | J2124-3358        |
| 27 | 411.061398549860  | 0.132894605234604 | 0.0848409785382199  | -8.400000E-16 | NOVALUE    | NOVALUE | 582370572 | 50984.400  | 7.710E9   | J0030+0451        |
| 28 | 415.8000000000000 | 0.105083307202850 | -1.25780067244822   | NOVALUE       | NOVALUE    | NOVALUE | 583718412 | 51000.000  | NOVALUE   | J0024-7204V       |
| 29 | 415.936333604394  | 6.11900480243835  | 0.365660789992678   | -8.390000E-16 | NOVALUE    | NOVALUE | 402278408 | 48900.000  | 7.850E9   | J2322+2057        |
| 30 | 416.4233760200000 | 5.62856134850849  | 0.212347054240217   | 9.320000E-13  | NOVALUE    | NOVALUE | 292768133 | 47632.520  | NOVALUE   | J2129+1210D       |
| 31 | 429.9748158120000 | 5.62855506532318  | 0.212390163872741   | -1.640000E-14 | NOVALUE    | NOVALUE | 292768133 | 47632.520  | 4.140E8   | J2129+1210E       |
| 32 | 437.467715178600  | 5.02161437448120  | -1.04691672771995   | 4.182000E-14  | NOVALUE    | NOVALUE | 662342413 | 51910.000  | NOVALUE   | J1910-5959E       |
| 33 | 460.175492583000  | 0.104939346345148 | -1.25807660900296   | 1.292400E-14  | NOVALUE    | NOVALUE | 635558413 | 51600.000  | NOVALUE   | J0024-7204L       |
| 34 | 490.852247414612  | 4.64471314050690  | -0.202141066906255  | -1.077040E-15 | NOVALUE    | NOVALUE | 534816011 | 50434.000  | 7.220E9   | J1744-1134        |
| 35 | 495.003050193040  | 0.105298506299621 | -1.25799300773179   | 5.164000E-15  | NOVALUE    | NOVALUE | 635558413 | 51600.000  | NOVALUE   | J0024-7204G       |
| 36 | 496.6423615200000 | 5.62848158696167  | 0.212362413137634   | -4.000000E-15 | NOVALUE    | NOVALUE | 292768133 | 47632.520  | 1.990E9   | J2129+1210F       |
| 37 | 528.484692377160  | 4.99806115622554  | 0.069866035561085   | -6.780000E-16 | NOVALUE    | NOVALUE | 685065613 | 52173.000  | 1.240E10  | J1905+0400        |
| 38 | 543.974457577860  | 0.104318852990477 | -1.258240449551972  | 5.686000E-15  | NOVALUE    | NOVALUE | 635558413 | 51600.000  | NOVALUE   | J0024-7204M       |
| 39 | 551.709426733100  | 4.72046671322877  | -0.249458688257073  | -8.040000E-16 | NOVALUE    | NOVALUE | 699494413 | 52340.000  | 1.090E10  | J1801-1417        |
| 40 | 571.978696023800  | 4.52619058713394  | -0.435489144436943  | -9.600000E-16 | NOVALUE    | NOVALUE | 635558413 | 51600.000  | 9.390E9   | J1721-2457        |
| 41 | 654.811330299780  | 4.81943648112773  | -0.434059021647859  | -3.470410E-13 | -6.600E-26 | NOVALUE | 6.6E-33   | 485049610  | 49858.000 | 2.990E7           |
| 42 | 654.8886372351000 | 0.105387867157323 | -1.25794064785423   | 4.690000E-15  | NOVALUE    | NOVALUE | 635558413 | 51600.000  | NOVALUE   | J0024-7204N       |
| 43 | 662.317327313100  | 0.105000054997530 | -1.25800819209628   | -1.874140E-14 | NOVALUE    | NOVALUE | 635558413 | 51600.000  | 6.440E8   | J0024-7204F       |
| 44 | 1083.61950814668  | 4.90301261955703  | -0.195918269991194  | -5.630000E-15 | NOVALUE    | NOVALUE | 702432013 | 52374.000  | 3.050E9   | J1843-1113        |
| 45 | 1283.85652221362  | 5.14716201622954  | 0.376695983254263   | -8.663498E-14 | 8.000E-26  | NOVALUE | 315855206 | 47899.500  | 2.350E8   | J1939+2134        |

Figure 5.11: The list of 45 known isolated pulsar selected from the ATNF catalogue.

8. As we are looking for a 95% upper limit, proceed as follows: if the confidence level (the percentage of instances in which  $\mathcal{F}'$  is greater than  $\mathcal{F}^*$ ) was less than 90% or above 98% (say  $x$ ), multiply the  $h_0$  by the ratio of  $\frac{95}{x}$  and take this value as the initial  $h_0$  for the next step.
9. Repeat steps “6” and “7” until the confidence level is in one of the following ranges: a) 90% – 95%, or b) 95% – 98%.
10. For case a), multiply  $h_0$  by 1.05; for case b), multiply by 0.90. (The factors 1.05 and 0.90 are, again, heuristic.)
11. Repeat the calculations of step “7” and following, but this time with 1000 injections in each run (instead of 150) to improve the statistics.
12. Repeat step “11” for 6 times; in each run follow the instructions in step “10”. (The number of repetitions is heuristic; it is chosen in a way that the range of computed confidence levels will always include values higher and lower than 95%; therefore we can make an “interpolation” fit instead of having to extrapolate.)

A flow chart version of this procedure can be found in Fig. (5.12).

In this way, we have obtained the 95% upper limits for 45 known isolated pulsars which are shown in Fig. (5.13). As can be seen from this plot, there are many pulsars where our result deviates from the upper limit (UL) obtained by the TDS group using a Bayesian approach for the same data. There are also a few pulsars for which the UL is very low, compared to the S5 sensitivity curve.

To investigate these anomalies, we have plotted the ratio of TDS over FDS upper limit versus the value of  $\mathcal{F}$ -statistic as shown in Fig. (5.14). As this figure shows, at roughly  $2\mathcal{F} = 4$  the ratio is close to unity; however, for values of less than 4 for the  $\mathcal{F}$ -statistic, the ratio increases rapidly, reaching significant magnitudes close to  $2\mathcal{F} \approx 1$ . In order to understand why this is so, note that, for very low values of  $\mathcal{F}^*$  (say about 1, which is true for the majority of anomalous points mentioned above), the False Alarm probability becomes quite high (Eq. 3.44),

$$\alpha = (1 + \mathcal{F})e^{-\mathcal{F}} = (1 + 1)\exp(-1) = 0.74. \quad (5.4)$$

This high false alarm probability means that *noise alone* has a high chance of producing an  $\mathcal{F}$ -statistic value greater than the  $\mathcal{F}^*$  produced by our data set and our template. In other words, our realization of the noise is one that is particularly *unlikely* to look like it contains our signal, and this statistical fluctuation yields a low upper limit value, due to the nature of the procedure that we use to determine the upper limit. The Bayesian approach is less sensitive to such fluctuations. Note that the resulting frequentist upper limit is *not* an artifact of our technique, but it is still a perfectly correct and consistent upper limit in the Frequentist framework.

A similar upper limit comparison can be found in GEO600’s UL as stated in the S1 paper [54]; a subsequent investigation by Yousuke Itoh [55] demonstrates that it will not always be the case that the frequentist upper limit gives a higher value in comparison with Bayesian, although the origin of this effect is not clearly discussed.

### 5.2.3 Bayesian Upper Limit

The Bayesian probability is a measure of *degree of belief* in the occurrence of a statistical process. In contrast with the Frequentist probability, in the Bayesian approach, we do not need for an event of that particular type to have actually happened; all we need is to find a measure for the degree to which a person believes that a given proposition is true.

The key ingredient of the Bayesian approach is Bayes’ theorem. As a simple example, consider a sample space  $S$  which contains of two subsets  $A$  and  $B$  (Fig. 5.15) [84]. Assuming that  $P(B) \neq 0$ , we can define the *conditional probability*  $P(A|B)$  as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (5.5)$$

The same will be true for  $P(B|A)$  (assuming  $P(A) \neq 0$ ), as

$$P(B|A) = \frac{P(B \cap A)}{P(A)}. \quad (5.6)$$



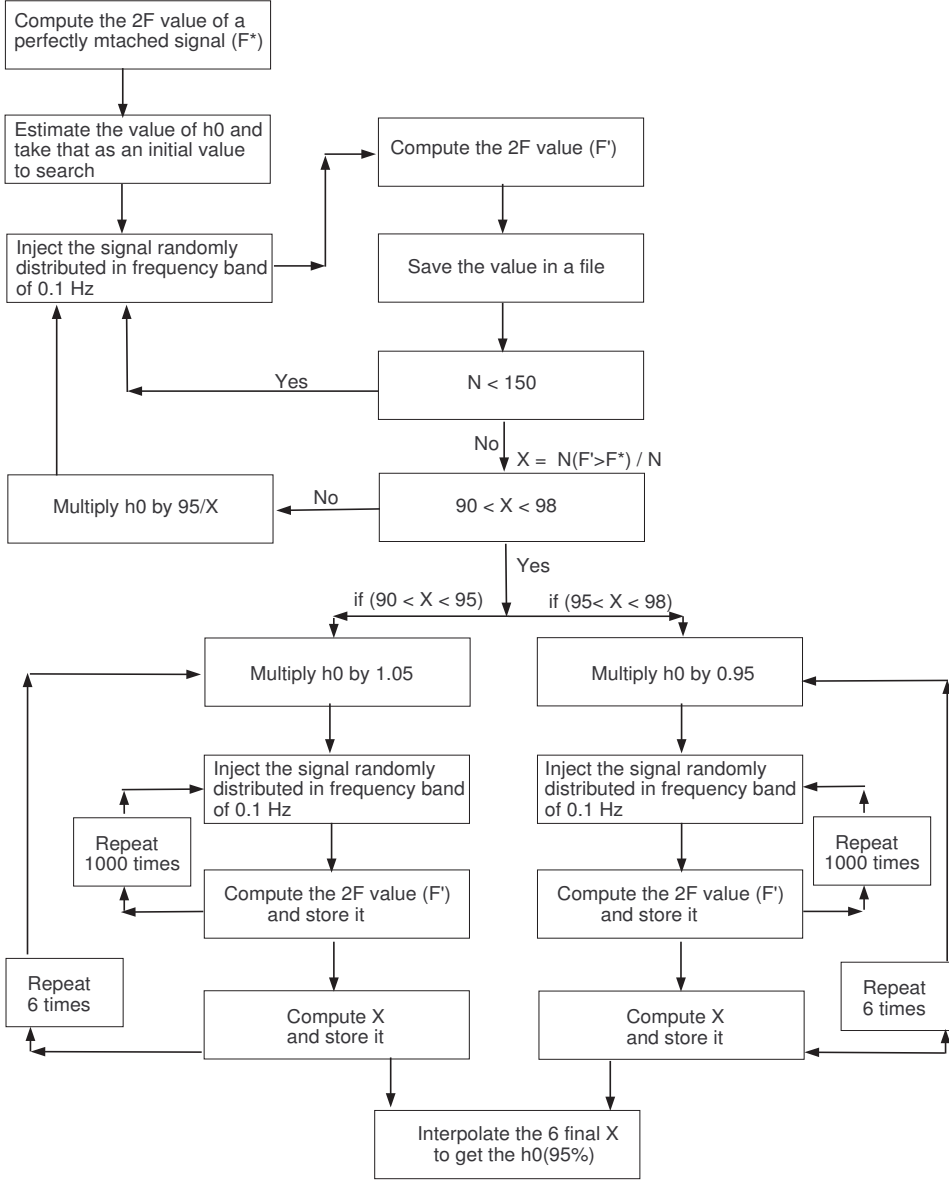


Figure 5.12: A Flow-Chart of how we implemented the frequentist upper limit.

Since  $A \cap B = B \cap A$ , we can combine these two equations and obtain

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A), \quad (5.7)$$

or

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (5.8)$$

This last equation represents Bayes' theorem. The term at the left hand side is called the *posterior probability*, while  $P(A)$  is the *prior probability* which reflects our initial knowledge about the quantity  $A$ . The term  $P(B|A)$  is called the *likelihood function*; the log of this is, in fact, the  $\mathcal{F}$ -statistic to be computed from our data. Our goal is to set an upper limit on the strength of the gravitational wave amplitude,  $h_0$ , using a given amount of available data from observations. Therefore, our data plays the role of  $B$  in Eq. (5.8); in line with the conventions in the other parts of this thesis, we will henceforth denote it by  $s$ . The term  $A$  is the quantity about which we intend to draw conclusions using our data; in our case, this is the upper limit  $h_0$ , so we will substitute  $h_0$  for  $A$  in what follows. With these substitutions, Eq. (5.8) now reads

$$P(h_0|s) = \frac{P(s|h_0) \times P(h_0)}{P(s)}, \quad (5.9)$$



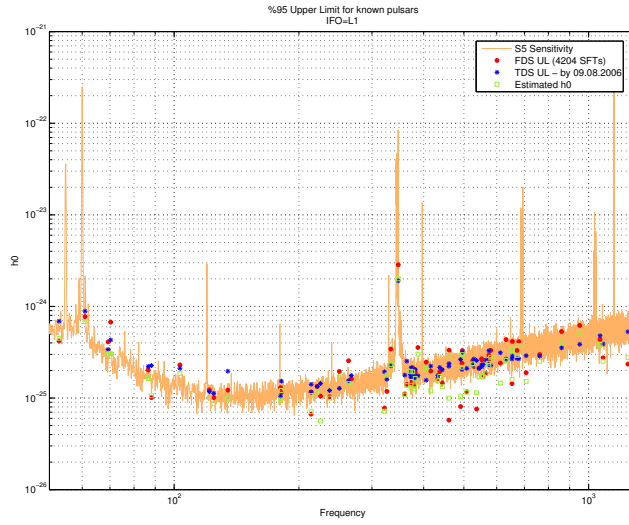


Figure 5.13: Comparison between the FDS frequentist upper limit with TDS's Bayesian results.

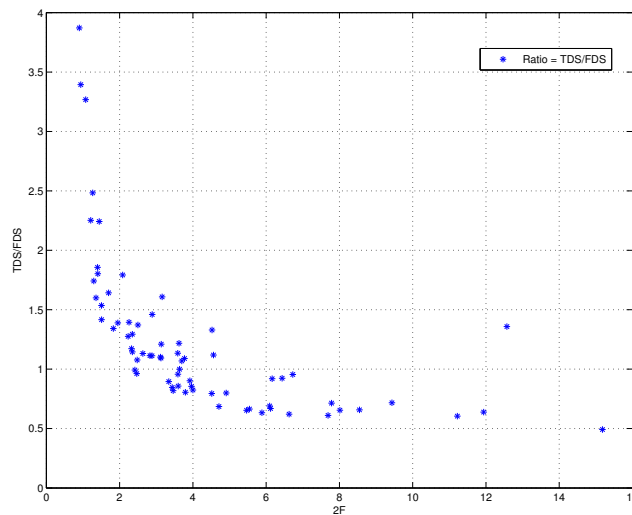


Figure 5.14: The relation between the discrepancy in TDS and FDS frequentist upper limit and the value of  $2\mathcal{F}$ .

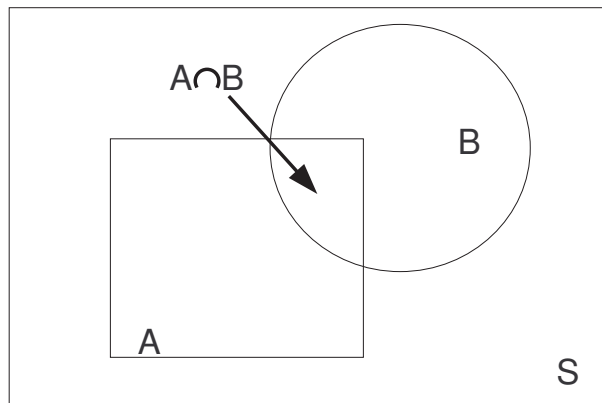


Figure 5.15: Relation between the sets A, B and S in the definition of conditional probability.

where  $P(h_0|s)$  is the conditional probability of  $h_0$  (posterior probability) given the data  $s$ ,  $P(s|h_0)$  is the likelihood function (to be defined below), and  $P(h_0)$  represents our prior knowledge about the distribution of  $h_0$ .

Since the term  $P(s)$  is independent of our signal, we can consider it as the constant normalization factor; it will cancel out automatically when we compute the confidence level. Therefore, we can rewrite our Bayes' theorem for the general case of all signal parameters as

$$P(h_0, \psi, \iota, \Phi_0|s) \propto P(s|h_0, \psi, \iota, \Phi_0) \times P(h_0, \psi, \iota, \Phi_0), \quad (5.10)$$

where, again,  $P(h_0, \psi, \iota, \Phi_0|s)$  is our posterior probability (to be calculated),  $P(s|h_0, \psi, \iota, \Phi_0)$  is the likelihood function and  $P(h_0, \psi, \iota, \Phi_0)$  is the prior probability of  $h_0, \psi, \iota, \Phi_0$ .

There are two common choices for estimating prior probability, known as a *flat prior* and as *Jeffrey's prior*. In the flat prior, the prior probability is chosen to be constant ( $P(h_0) \equiv \text{constant}$ ), while in Jeffrey's prior, it is taken to vary inversely proportional to the value of  $h_0$  ( $P(h_0) \equiv 1/h_0$ ). More details can be found in Réjean Dupuis' PhD thesis [85].

In the following we will focus on a flat prior; we are aiming for a valid comparison of our results with those of TDS, which are based on a flat prior as well.

To obtain the posterior probability, we need to calculate the likelihood function. By Eq. (2.85), it can be expressed as

$$P(s|h_0, \psi, \iota, \Phi_0) \propto e^{-\frac{1}{2}M_{ab}(A-A^0)^a(A-A^0)^b} = G(h_0, \psi, \iota, \Phi_0), \quad (5.11)$$

where  $A = (A^1, A^2, A^3, A^4)(h_0, \psi, \iota, \Phi_0)$  are the four amplitude parameters defined in Eqs. (2.22-2.25). The  $A^0 = (A^{10}, A^{20}, A^{30}, A^{40})$  are also the best fit for the  $A$ s resulting from our calculation of the  $\mathcal{F}$ -statistic.

For proper normalization, we first compute the integral

$$I \equiv \int_0^\infty P(h_0)dh_0 \int_{-1}^1 d\mu \int_{-\pi/4}^{\pi/4} d\psi \int_0^{2\pi} d\Phi_0 G(h_0, \psi, \iota, \Phi_0), \quad (5.12)$$

where  $\mu \equiv \cos \iota$ , and where we have taken ' $p(h_0) \equiv \text{constant}$ '. To find the Upper Limit (UL), we use  $h_0^{max}$  as the upper bound in the integration over  $h_0$ ,

$$I_{UL} \equiv \int_0^{h_0^{max}} P(h_0)dh_0 \int_{-1}^1 d\mu \int_{-\pi/4}^{\pi/4} d\psi \int_0^{2\pi} d\Phi_0 G(h_0, \psi, \iota, \Phi_0). \quad (5.13)$$

We select  $h_0^{max}$  in such a way that the ratio  $I_{UL}/I$  gives us the desired confidence level. In our example, we are looking for the 95% Upper Limit, therefore

$$\frac{I_{UL}}{I} = 0.95. \quad (5.14)$$

### Practical implementation

To implement the above formalism, let us first construct the function  $G(h_0, \psi, \iota, \Phi_0)$ . To do so, we will make use of several equations from the preceding chapters: substituting Eqs. (2.12 and 2.13) into Eqs. (2.22-2.25) and expanding the matrix of Eq. (4.24) (for the case of a single detector) as

$$M_{11} = \frac{A}{2}, \quad M_{12} = \frac{C}{2}, \quad M_{13} = 0, \quad M_{14} = 0, \quad (5.15)$$

$$M_{21} = \frac{C}{2}, \quad M_{22} = \frac{B}{2}, \quad M_{23} = 0, \quad M_{24} = 0, \quad (5.16)$$

$$M_{31} = 0, \quad M_{32} = 0, \quad M_{33} = \frac{A}{2}, \quad M_{34} = \frac{C}{2}, \quad (5.17)$$

$$M_{41} = 0, \quad M_{42} = 0, \quad M_{43} = \frac{C}{2}, \quad M_{44} = \frac{B}{2}, \quad (5.18)$$

we can construct the four elements

$$G_1 = \frac{A}{2}(A^1 - A^{1_0})^2 + \frac{C}{2}(A^1 - A^{1_0})(A^2 - A^{2_0}), \quad (5.19)$$

$$G_2 = \frac{B}{2}(A^2 - A^{2_0})^2 + \frac{C}{2}(A^2 - A^{2_0})(A^1 - A^{1_0}), \quad (5.20)$$

$$G_3 = \frac{A}{2}(A^3 - A^{3_0})^2 + \frac{C}{2}(A^3 - A^{3_0})(A^4 - A^{4_0}), \quad (5.21)$$

$$G_4 = \frac{B}{2}(A^4 - A^{4_0})^2 + \frac{C}{2}(A^4 - A^{4_0})(A^3 - A^{3_0}), \quad (5.22)$$

to construct the final form of  $G(h_0, \psi, \iota, \Phi_0)$  in Eqs. (5.12 and 5.13) as

$$G = \exp\left[-\frac{1}{2}(G_1 + G_2 + G_3 + G_4)\right]. \quad (5.23)$$

This is the core equation for our upper-limit analysis, to be constructed using the output of our CFS\_v2 code. With CFS\_v2, we calculate the four amplitudes  $A^i$  as well as the matrix elements  $M_{ab}$  (namely the amplitude modulation coefficients  $A, B$  and  $C$ ). Practically, we can calculate the UL value in Eq. (5.14) in two ways. One is to follow the exact procedure spelled out above, first calculating the normalization in Eq. (5.12) and then trying to find a value of  $h_0^{max}$  for which the ratio of Eq. (5.14) will be satisfied. This can be done using the *Numerical Integration* routines in mathematical software like *Mathematica* and *Maple* (the initial results quoted in this thesis were calculated using *Mathematica*, and the most important results were checked using *Maple*). Another way would be to calculate the posterior probability of  $h_0$  by marginalizing over the other three parameters. This can be expressed in mathematical form as

$$p(h_0|s) \propto \int \int \int G(h_0, \psi, \iota, \Phi_0) d\psi d\mu d\Phi_0. \quad (5.24)$$

Once the posterior probability for  $h_0$  is known, one can then integrate it over a sufficient range of  $h_0$  to find out the area covered; the result can be used for proper normalization (namely unit total area). Next, we can find out at which  $h_0$  the fraction of area would satisfy our required confidence level.

Our initial analysis used the first algorithm; however, we have also recently developed a code which implements the second procedure. As would be expected, the results are consistent, providing a further internal consistency check for our software implementation. The comparison is summarized in Fig. (5.16).

## Results

We have performed our upper-limit search several times, using data sets of different sizes taken from the S5 run, and comparing our results with those of the TDS. In this section we will describe in some detail our efforts to reconcile the results obtained by the different techniques in order to illustrate the type of effects that one has to deal with in these searches. The comparisons revealed a significant mismatch. In order to account for these, a number of different problems had to be resolved both on our (FDS) side and on the TDS side. The first problem from our side was that the available SFTs in our data set were not calibrated in a uniform manner. To explain it more, let us look at Fig. (5.17). It shows how the data can be divided according to quality into four different epochs. The first epoch starts at the beginning of the S5 run (07 November 2005) and lasts until 31 January 2006. The available data for this interval was analyzed using the version one calibration (so-called C01) of time-series data. Given the poor quality of data, the pulsar group of LSC recently recommended that it should not be used for analysis purposes. Figure (5.17) clearly shows the non-stationarity of the Fourier-transformed data during this interval. After 31 January 2006, the data was processed using version two of the calibration (C02). From 22 June 2006 until 03 October 2006 (3rd epoch) and from 03 October 2006 afterward (4th epoch), one might think that there had been great improvements in the quality of the (processed) data; however, this apparent improvement is due to the software applying an improper calibration: the PSD of these two epochs should in fact be multiplied by a factor of 1.18 and 1.22 for the 3rd and 4th epochs respectively.

The TDS group, in turn, found a programming error in their software which resulted in an incorrect value for all the upper limits given.

At this point in time – the TDS group having improved their software, and us having found out about the calibration issues with S5 data –, both groups decided to perform an analysis strictly in parallel, working

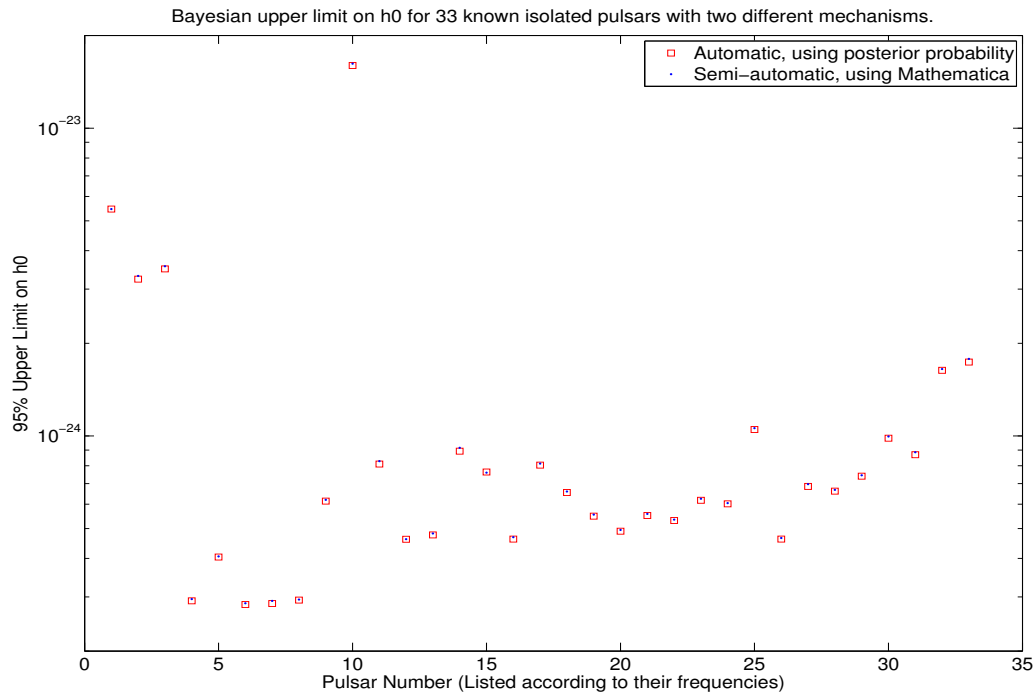


Figure 5.16: Comparison between the upper limit set using the Mathematica, as our initial algorithm and automated algorithm based on the posterior probability.

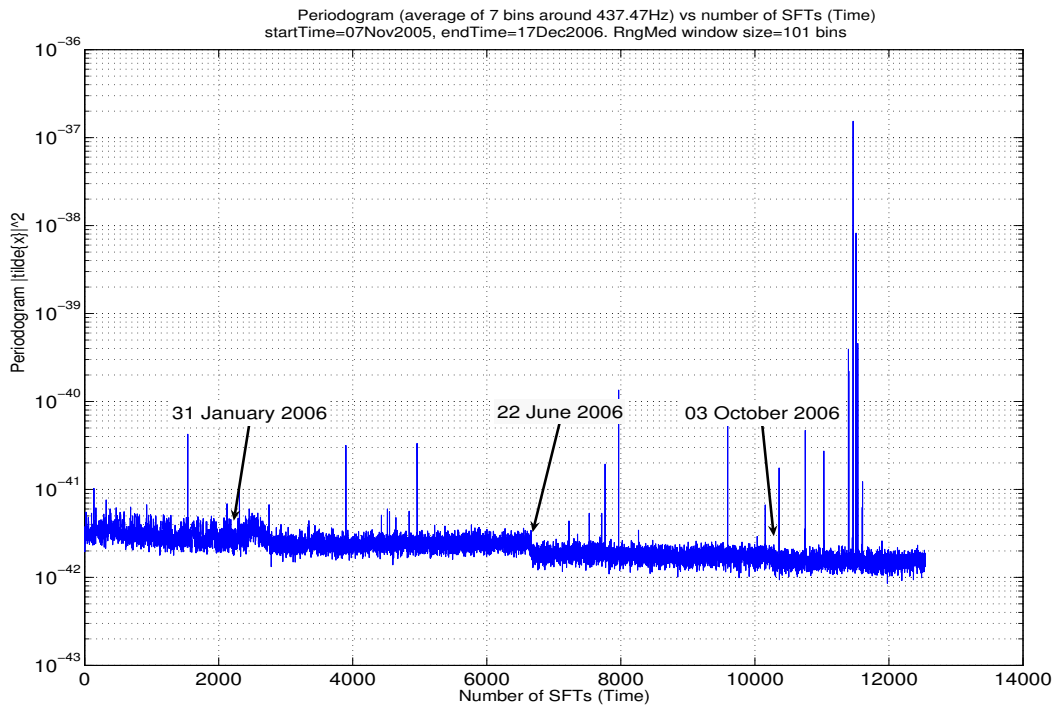


Figure 5.17: Periodogram ( $|\tilde{X}|^2$ ) of the 13.5 months of S5 data for H1 detector.

on data taken during the same short period of time, using the same calibration of data and, therefore, obtaining results of equal quality. We selected 11 days' worth of data starting from 01:33:32 GMT in 22nd of March 2006 and ending on 07:38:48 GMT 2nd of April 2006 (827026426 GPS to 827998742), corresponding to a time interval of 972316 seconds. With this choice, we are firmly within the epoch in which the available data uses the proper calibration; within that period, the interval was chosen so as to ensure a maximal number of available SFTs for the H1 detector in our data base: in this period, we had 463 SFTs for H1, which covers about 9.6 days (833400 secs with 86% duty factor); in addition, we had 292 SFTs for L1, corresponding to about 6 days (525600 secs with 54% duty factor). The available data for the TDS group were: 915945 secs for H1 with the 94% of duty factor and for L1 666643 secs with the duty factor of 69%. These values are summarized in Table (5.5).

|    | Run Intervals<br>(secs) | Available Data<br>TDS (secs) | % Duty<br>Factor | Available Data<br>FDS (secs) | % Duty<br>Factor | Ratio<br>TDS/FDS |
|----|-------------------------|------------------------------|------------------|------------------------------|------------------|------------------|
| H1 | 972316                  | 915945                       | %94              | 833400                       | %86              | 1.10             |
| L1 | 972316                  | 525600                       | %69              | 666643                       | %54              | 1.27             |

Table 5.5: The amount of data used for the short period of time (11 days interval) for both H1 and L1 detectors. The period starts from 01:33:32 GMT in 22nd of March 2006 till 07:38:48 GMT 2nd of April 2006 (827026426 GPS to 827998742).

Using data from the above-mentioned time period, we have set the upper limit for 32 known isolated pulsars, as discussed above (from the list of 33 pulsars given there, one, namely J0537-6910 at a frequency of 124.05 Hz, was removed: at that frequency, the data showed a peak that, by LSC Pulsar group consensus, very probably represented some instrument-specific resonance). The histogram of relative differences between the UL of TDS and FDS for H1 is shown in Fig. (5.18); on the right, it also includes a plot showing each individual mismatch.

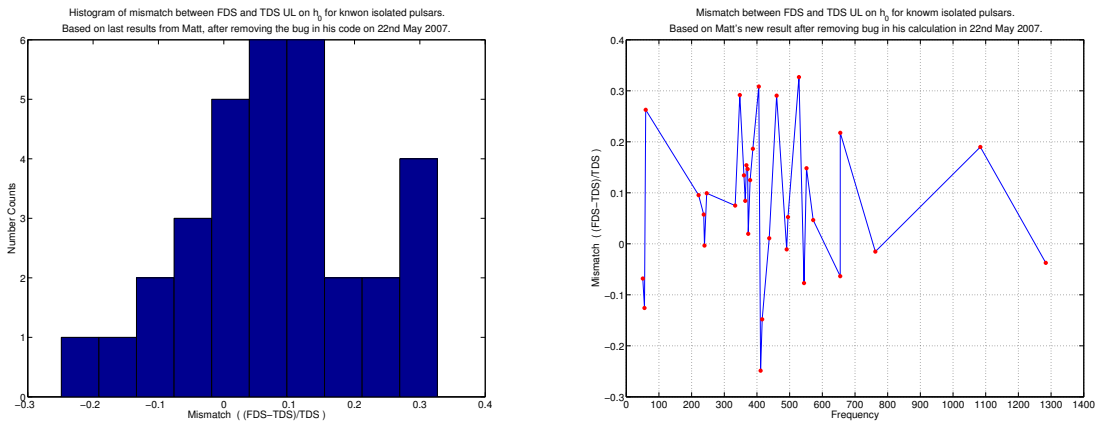


Figure 5.18: Histogram of mismatch between FDS and TDS upper limit for 11 days of S5 data for H1 detector. To show the mismatch more precisely, the plot of each individual mismatch (right) is also shown.

As we can see from Table (5.5), in this analysis, the TDS available data for H1 is about 10% more than what we had in our data base. This increases the SNR by a factor of 5% ( $=\sqrt{1.1}$ ) for the time domain data, which should, in fact, reduce the value of upper limit by the same amount (5%) for the TDS. If we want to compensate for the effect of this additional data in the TDS data base, we need to shift the relative UL mismatch by about 5% toward the negative side in Fig. (5.18); the reason for this is that the relative mismatch is  $(UL_{FDS} - UL_{TDS})/UL_{TDS}$ , hence in order to compensate the TDS upper limit, we need to increase their UL by 5%, which makes the entire histogram move towards more negative values. In its final form, the overall mismatch varies between -30% and +30% and does not show a systematic bias. The fact that our mismatch is not systematic is unfortunate, as it means that there is little information to point us to a specific cause of the mismatch. As things stood, we were forced to proceed on a trial-and-error basis – making assumptions for what could have caused the discrepancy,

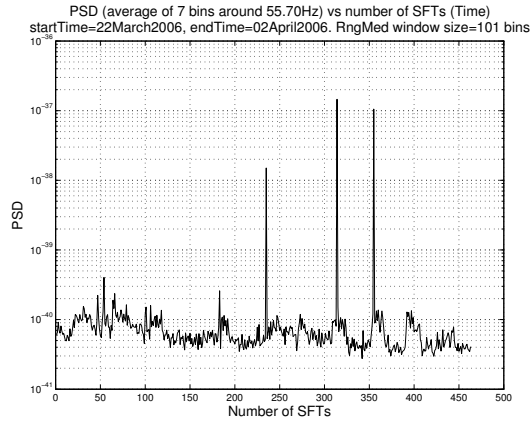


Figure 5.19: Noise floor of the 11days of S5 data for H1 detector.

changing our analysis procedure accordingly, repeating the modified analysis, and checking whether or not the mismatch was reduced.

First, we noted the existence of a few (three) SFTs that had very high PSD compare to the background noise (Fig. 5.19). We removed these SFTs from our list and repeated the search; unsurprisingly, this resulted in no significant changes, as our code, by using the noise weight, will mostly protect the final result from change brought about by the inclusion of a few bad SFTs like these (see Fig. (5.20) and the additional information for this figure given in the subsequent paragraphs).

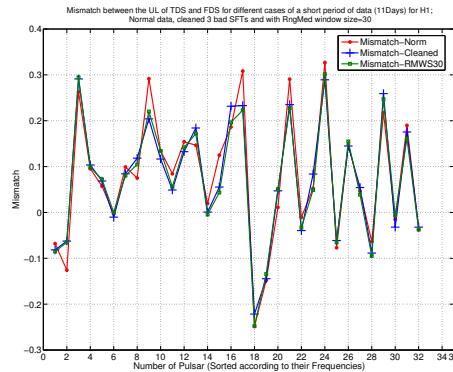


Figure 5.20: Mismatch between TDS and FDS UL for short-term S5 data of H1 detector in three case. i) with *All available SFTs*, ii) *cleaning three bad SFTs* and iii) *taking the running median window size to be 30*

Another issue that we explored was noise estimation by the TDS group, as explained in more details in [85] and [82]. Simply put, they estimate the noise floor for one minute of data but; they then assume that their data is stationary over 30 minutes, with the average equal to the previously estimated noise floor. This is roughly the same as what we assume for a 30 minute duration of our SFT. However, the frequency bin they use for taking this one minute average is 1/60 Hz. According to our frequency resolution (size of frequency bin in a single SFT; 1/1800 Hz), this correspond to the 30 frequency bins average around the Doppler-shift corrected pulsar frequency. Given this, we explored the consequences of adapting our own noise floor estimation accordingly. More concretely, our estimate is based on a running median; we reduced the window size for that medium from 50 to 30 bins and, with these new value, repeated our search once again. The result, together with the cleaned data and the normal data, is displayed in Fig. (5.20). As is evident from these plots, the results did not change significantly (although there is improvement by less than 10% for a few of the pulsars). The interesting part of this comparison is that the value of mismatch with the cleaned SFTs and the running median window size=30 are very very close to each other.

As is also evident from the figure, these particular tests were not sufficient to elucidate the origin of the discrepancies. Next, we tried to find out whether both groups went the right way about estimating the posterior probability of the pulsar amplitude parameters. In order to do so, we looked at the probability distribution curve of  $h_0$  of a single pulsar and compared them in FDS and TDS approach. The plots showed in Fig. (5.21) correspond to pulsar *J0024-7204N*, which has maximum mismatch in upper limit on the L1 IFO. Also shown is the estimate for the 95% upper limit on  $h_0$  for both approaches, TDS and FDS ( $UL_{TDS}$ ,  $UL_{FDS}$ ); both results are internally consistent with what we can see from their posterior distributions, hence we are left to investigate whether the discrepancies might result from differences in the quality of the data. To this end, we compare the results obtained for one and the same set of *simulated data*, as explained in the next section.

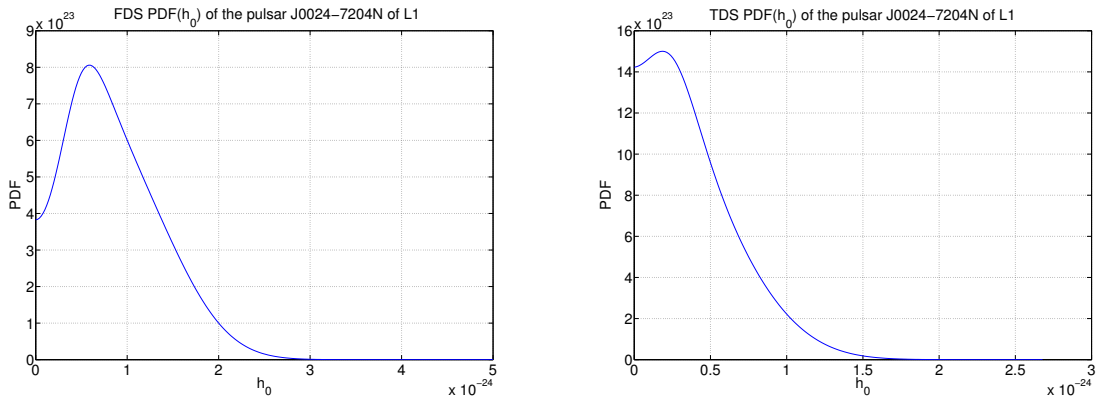


Figure 5.21: The Posterior probability of signal amplitude of pulsar J0024-7204N for the 11 days of S5 data from L1 detector for FDS, left, and TDS, right.

### Comparison between TDS and FDS Bayesian upper limit using simulated data

Let us specify what our simulated data set is supposed to look like. We would like to simulate the search for four known pulsars, whose frequency, frequency drift and sky position are assumed to be known from astronomical observations. In order to make our simulation realistic, we also assume that there is a significant noise level so that the pulsar signals cannot actually be detected by our analysis. The simplest way to achieve that is to set all four signal amplitudes to zero. The resulting parameters for the 4 pulsars we generated using `lalapps_Makefakedata` are listed in Table (5.6). Next, both the TDS

| Pulsar Parameters | PULSAR0      | PULSAR1      | PULSAR2      | PULSAR3      |
|-------------------|--------------|--------------|--------------|--------------|
| IFO               | H1           | H1           | H1           | L1           |
| $h_0$             | 0.0          | 0.0          | 0.0          | 0.0          |
| $f_0$ (Hz)        | 304.934124   | 504.934124   | 713.934124   | 182.934124   |
| $\dot{f}$ (Hz/s)  | 6.571779e-09 | 1.571779e-11 | 2.571779e-12 | 3.571779e-13 |
| $\alpha$ (rad)    | 2.359236     | 1.359236     | 1.059236     | 0.359236     |
| $\delta$ (rad)    | 0.730642     | 0.30642      | 1.030642     | 1.230642     |
| $\sqrt{S_h}$      | 1.0          | 1.0          | 2.0          | 3.0          |

Table 5.6: Parameters of 4 pulsars that we search for in a simulated noise data set in order to compare the FDS and TDS upper limit. The duration of the data set is 10 days.

and FDS Bayesian approaches were used to set an upper limit using this simulated data. Note that the data set used here is exactly the same for the TDS and FDS. The upper limits for the simulated pulsars are listed in Table (5.7). Comparing FDS and TDS, we see that the mismatch between the different search schemes has been reduced to an average  $\pm 6.5\%$ . The reduction is a strong indication that the discrepancies among the results of the two searches is due to the way in which the two search techniques

“react” to non-Gaussianity and non-stationarities of the real data. This is consistent with the conclusion that we have previously drawn.

|          | PULSAR0                | PULSAR1                | PULSAR2                | PULSAR3                |
|----------|------------------------|------------------------|------------------------|------------------------|
| UL(FDS)  | $1.143 \times 10^{-2}$ | $1.423 \times 10^{-2}$ | $1.375 \times 10^{-2}$ | $3.105 \times 10^{-2}$ |
| UL(TDS)  | $1.069 \times 10^{-2}$ | $1.347 \times 10^{-2}$ | $1.489 \times 10^{-2}$ | $3.291 \times 10^{-2}$ |
| Mismatch | +7.5%                  | +5.7%                  | -7.6%                  | -5.6%                  |

Table 5.7: 95% upper limit for 4 simulated pulsars in FDS and TDS and the relative mismatch between them. The mismatch is defined as [mismatch = (FDS-TDS)/TDS]

The comparison for the posterior probabilities of  $h_0$  of PULSAR0, PULSAR1 and PULSAR2 are shown in Fig. (5.22). Comparing the probabilities of  $h_0$  of each simulated pulsar in TDS and FDS, we see that all of them have quite similar distributions. Likewise, as a more comparison, we plotted the posterior probabilities of all four amplitude parameters of PULSAR3 as shown in Fig. (5.23). It shows that, apart from  $\phi_0$ , the rest of three show a very similar distributions.

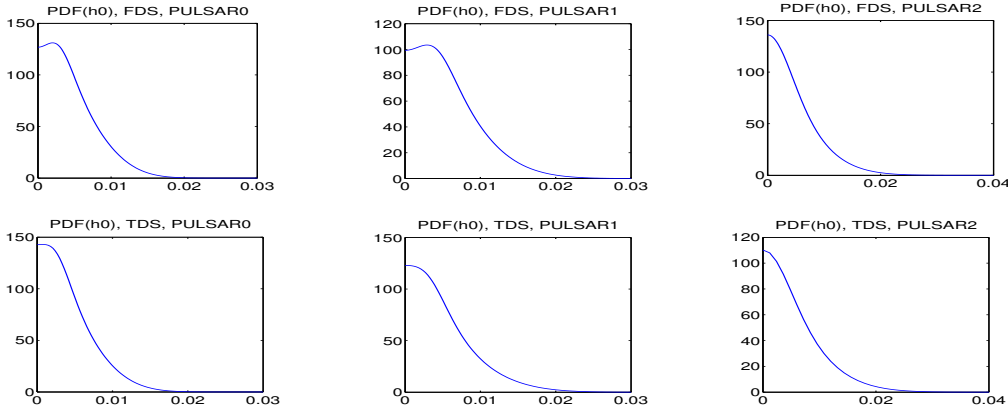


Figure 5.22: The Posterior probabilities of  $h_0$  of simulated pulsars; PULSAR0, PULSAR1 and PULSAR2 as was discussed in the text body.

### Distribution of 95% Bayesian upper limits on $h_0$ using simulated data in frequency domain

As a next step, we examine more closely the distribution for UL results (95% limit for  $h_0$ ) computed using our method and compare it to that predicted in [86]. We start with the idealized case of a large number (5500) of simulated data set with no signal in them, and we compute the 95% upper limit on  $h_0$  of each data set. The search sky locations are chosen randomly in such a way that their distribution over the solid angle is uniform; detector position is picked randomly from a list consisting of the locations of H1, L1, VIRGO and GEO600. The resulting mean upper limit,  $\langle h_0^{95\%} \rangle$ , is:

$$\langle h_0^{95\%} \rangle = (10.67 \pm 0.04) \sqrt{\frac{S_h(f)}{T}}. \quad (5.25)$$

In order to compare our result with that of [86], we repeated this experiment with only the H1, L1 and GEO600 detectors, as in their analysis. The result was  $\langle h_0^{95\%} \rangle = (10.59 \pm 0.04) \sqrt{S_h(f)/T}$ . The value in [86] is  $\langle h_0^{95\%} \rangle = (10.8 \pm 0.2) \sqrt{S_h(f)/T}$ . The shape of the two distributions is also very similar. The results discussed above show that the TDS and FDS upper limit searches yield consistent results on theoretical noise. This confirms the result of the previous section, that the discrepancies between the TDS and FDS results should be ascribed to the way that the two search techniques react to the deviations of the noise from the idealized Gaussian and stationary case.



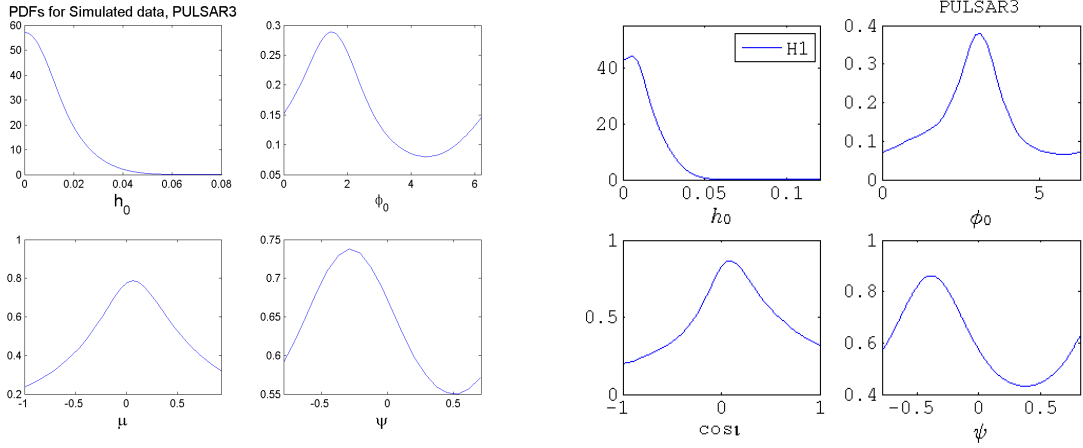


Figure 5.23: The Posterior probabilities for four amplitude parameters of simulated pulsar, PULSAR3, as was discussed in the text body. Left figure correspond to FDS and the right for the TDS.

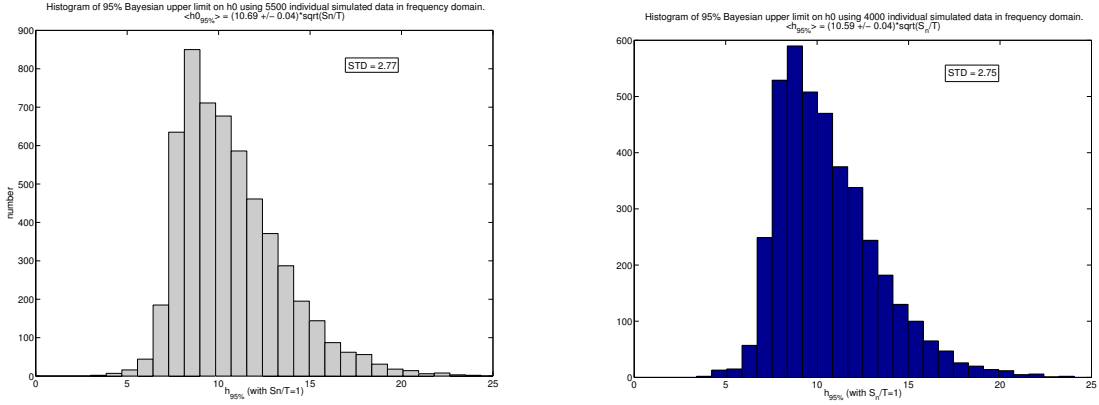


Figure 5.24: Distribution of 95% upper limit on  $h_0$  for 5500 and 4000 individual runs on simulated data.

### 95% Bayesian upper limit on $h_0$ using one year of V3 calibration of S5 data

The last results we present here have been obtained from work on one year's worth of newly generated version of SFTs for the S5 data (the so-called V3 version or C03) which we have used to place upper limits for  $h_0$  on the 33 known isolated pulsars among those listed in table of Fig. (5.11).

This version of SFTs, in fact, is the most proper version compared to what was generated and published before, (named as V1 and V2 calibration). The V3 calibration supersedes the V2 one, in that all the data calibrated under V2 are re-calibrated as V3, plus V3 applies for a much longer time frame in S5. Owing to changes on interferometers, a calibration can become "stale", and no longer accurately describe the response of the system to a gravitational wave. In the case of V2, there were several instances (two epochs) in which the staleness was mostly in amplitude, mostly flat, and about 18% and 22% respectively. However, there is more to it than just amplitude: filters are different in the later epochs, making for gain changes around, for instance, violin frequencies, and effects in phase at those frequencies as well [87, 88].

The S5 upper limits on  $h_0$  from the individual LIGO detectors as well as a joint (H1+H2+L1) upper limits are presented below. The H1 data upper limits are smaller than  $10^{-25}$  for 9 pulsars among the 33, for the L1 data there are only 2 and for the H2 data there are none. The coherent multi-IFO search yields 13 upper limit values below  $10^{-25}$ . The best sensitivity achieved among all three detectors corresponds to the pulsar J1910-5959B with the frequency of 239.30 Hz with the upper limit of  $6.0 \times 10^{-26}$ . For the joint IFOs, the best upper limit is  $4.27 \times 10^{-26}$  for the pulsar J1910-5959D with the frequency of 221.35Hz, close to the best sensitivity of the LIGO detectors around 150Hz. All the upper limits values are plotted

| Pulsar Name | Frequency | H1        | H2        | L1        | Joint     |
|-------------|-----------|-----------|-----------|-----------|-----------|
| J1952+3252  | 50.59     | 6.418e-25 | 2.774e-24 | 7.181e-25 | 4.665e-25 |
| J1913+1011  | 55.70     | 5.105e-25 | 1.925e-24 | 6.422e-25 | 4.434e-25 |
| J0534+2200  | 59.56     | 4.271e-25 | 1.502e-24 | 6.422e-25 | 5.109e-25 |
| J0537-6910  | 124.05    | 6.484e-26 | 1.560e-25 | 1.293e-25 | 6.484e-26 |
| J1910-5959D | 221.35    | 6.950e-26 | 1.577e-25 | 6.507e-26 | 4.273e-26 |
| J1748-2446C | 237.08    | 7.535e-26 | 1.442e-25 | 8.115e-26 | 5.618e-26 |
| J1910-5959B | 239.30    | 6.002e-26 | 1.048e-25 | 1.134e-25 | 6.326e-26 |
| J1730-2304  | 246.22    | 6.715e-26 | 1.359e-25 | 1.068e-25 | 5.310e-26 |
| J1629-6902  | 333.30    | 7.753e-26 | 1.668e-25 | 1.925e-25 | 5.574e-26 |
| J0024-7204C | 347.42    | 2.312e-24 | 1.036e-24 | 1.735e-24 | 1.109e-24 |
| J1518+0205A | 360.13    | 2.032e-25 | 2.560e-25 | 1.094e-25 | 1.040e-25 |
| J0711-6830  | 364.23    | 8.774e-26 | 1.726e-25 | 2.143e-25 | 8.642e-26 |
| J1823-3021A | 367.65    | 1.494e-25 | 2.747e-25 | 1.184e-25 | 7.558e-26 |
| J1911+0101B | 371.45    | 1.403e-25 | 2.823e-25 | 1.801e-25 | 1.007e-25 |
| J0024-7204D | 373.30    | 1.149e-25 | 2.190e-25 | 1.350e-25 | 1.074e-25 |
| J1910-5959C | 378.98    | 9.594e-26 | 2.558e-25 | 1.523e-25 | 7.030e-26 |
| J1024-0719  | 387.43    | 1.409e-25 | 3.253e-25 | 1.967e-25 | 1.087e-25 |
| J2124-3358  | 405.59    | 1.322e-25 | 1.455e-25 | 1.541e-25 | 8.407e-26 |
| J0030+0451  | 411.06    | 1.753e-25 | 2.487e-25 | 1.111e-25 | 1.073e-25 |
| J2322+2057  | 415.94    | 1.512e-25 | 2.029e-25 | 1.655e-25 | 1.137e-25 |
| J1910-5959E | 437.47    | 1.213e-25 | 2.169e-25 | 2.116e-25 | 9.364e-26 |
| J0024-7204L | 460.18    | 8.825e-26 | 1.687e-25 | 1.822e-25 | 8.177e-26 |
| J1744-1134  | 490.85    | 1.520e-25 | 2.466e-25 | 2.191e-25 | 1.085e-25 |
| J0024-7204G | 495.00    | 1.698e-25 | 1.948e-25 | 2.250e-25 | 1.092e-25 |
| J1905+0400  | 528.49    | 1.724e-25 | 3.150e-25 | 2.641e-25 | 1.525e-25 |
| J0024-7204M | 543.98    | 1.422e-25 | 2.160e-25 | 1.405e-25 | 8.517e-26 |
| J1801-1417  | 551.71    | 2.133e-25 | 3.156e-25 | 2.110e-25 | 1.346e-25 |
| J1721-2457  | 571.98    | 1.458e-25 | 2.868e-25 | 2.006e-25 | 1.290e-25 |
| J1824-2452  | 654.81    | 1.235e-25 | 3.211e-25 | 3.177e-25 | 1.212e-25 |
| J0024-7204N | 654.89    | 1.349e-25 | 4.932e-25 | 2.123e-25 | 1.382e-25 |
| J0024-7204F | 762.32    | 2.233e-25 | 2.603e-25 | 2.373e-25 | 1.159e-25 |
| J1843-1113  | 1083.62   | 2.163e-25 | 5.569e-25 | 3.975e-25 | 2.085e-25 |
| J1939+2134  | 1283.86   | 5.713e-25 | 6.712e-25 | 5.459e-25 | 3.614e-25 |

Table 5.8: Upper limits on amplitude of known isolated pulsars using one year of V3 calibration of S5 data. Pulsar J0534+2200 is the Crab pulsar.

against the expected curve based on the noise estimates in the detectors during the observation times in Figs. (5.25) and (5.26). Likewise, the PDFs of all 33 pulsars for all three detectors and their combination are shown in Figs. (5.27 - 5.30). These PDFs show that there is no indication of a detection of a signal from any of the searches that have been carried out: the measured PDFs are quite broad and no significant peak is evident. As we have seen in the analysis of S4 hardware injections as well in the simulated signal with quite large SNR, in the presence of a signal the peaks of PDFs were very sharp. This result is not unexpected: for all the pulsars but one, the upper limits that we have derived do not constrain the gravitational wave emission more than what can be inferred from the measured period spin-down rates, assuming that all the lost energy is indeed radiated away by gravitational waves – which in itself is also an extremely over-optimistic assumption.

From the upper limit on the amplitude of the gravitational wave, we can set an upper limit on the *equatorial ellipticity*,  $\varepsilon$ , of the pulsars. If we assume that the mechanism of emitting the gravitational waves are due to the triaxiality with the GW frequency of twice the rotation frequency, Eq. (2.16) gives us the following value for the ellipticity [83]

$$\varepsilon = 0.237 \left( \frac{h_0}{10^{-24}} \right) \left( \frac{r}{1\text{kpc}} \right) \left( \frac{1\text{Hz}}{\nu} \right)^2 \left( \frac{10^{38}\text{kg m}^2}{I_{zz}} \right), \quad (5.26)$$

where  $\nu$  is the rotational frequency of pulsar and we have assume a fiducial moment of inertia value of

| Freq    | H1         | H2         | L1         | Joint $\epsilon^{95\%}$ |
|---------|------------|------------|------------|-------------------------|
| 50.59   | 5.94217e-4 | 2.56835e-3 | 6.64877e-4 | 4.31915e-4              |
| 55.70   | 7.12557e-4 | 2.63561e-3 | 8.79239e-4 | 6.07082e-4              |
| 59.56   | 2.28326e-4 | 8.03048e-4 | 6.30580e-4 | 2.73091e-4              |
| 124.05  | 1.97331e-4 | 4.74663e-4 | 3.93347e-4 | 1.97320e-4              |
| 221.35  | 5.37902e-6 | 1.22074e-5 | 6.27445e-6 | 3.30669e-6              |
| 237.08  | 1.10566e-5 | 2.11624e-5 | 1.19078e-5 | 8.24351e-6              |
| 239.30  | 3.97486e-6 | 6.93911e-6 | 7.50781e-6 | 4.18935e-6              |
| 246.22  | 5.35539e-7 | 1.08402e-6 | 8.51880e-7 | 4.23501e-7              |
| 333.30  | 8.99745e-7 | 1.93542e-6 | 2.23401e-6 | 6.46888e-7              |
| 347.42  | 8.71454e-5 | 3.90598e-5 | 6.54133e-5 | 4.18232e-5              |
| 360.13  | 1.15849e-5 | 1.45956e-5 | 6.23479e-6 | 5.92873e-6              |
| 364.23  | 6.52077e-7 | 1.28297e-6 | 1.59274e-6 | 6.42252e-7              |
| 367.65  | 8.27753e-6 | 1.52181e-5 | 6.55756e-6 | 4.18753e-6              |
| 371.45  | 7.13204e-6 | 1.43541e-5 | 9.15900e-6 | 5.11987e-6              |
| 373.30  | 3.75029e-6 | 7.14929e-6 | 4.40857e-6 | 3.50780e-6              |
| 378.98  | 2.53296e-6 | 6.75347e-6 | 4.02126e-6 | 1.85611e-6              |
| 387.43  | 4.71608e-7 | 1.08902e-6 | 6.58431e-7 | 3.63908e-7              |
| 405.59  | 1.90520e-7 | 2.09587e-7 | 2.21961e-7 | 1.21118e-7              |
| 411.06  | 2.95074e-7 | 4.18576e-7 | 1.87064e-7 | 1.80572e-7              |
| 415.94  | 6.46127e-7 | 8.67036e-7 | 7.07230e-7 | 4.85915e-7              |
| 437.47  | 2.40412e-6 | 4.29788e-6 | 4.19328e-6 | 1.85548e-6              |
| 460.18  | 1.89631e-6 | 3.62409e-6 | 3.91586e-6 | 1.75709e-6              |
| 490.85  | 2.87080e-7 | 4.65734e-7 | 4.13758e-7 | 2.04873e-7              |
| 495.00  | 3.15409e-6 | 3.61806e-6 | 4.17900e-6 | 2.02856e-6              |
| 528.48  | 7.84280e-7 | 1.43260e-6 | 1.20126e-6 | 6.93383e-7              |
| 543.97  | 2.18736e-6 | 3.32227e-6 | 2.16008e-6 | 1.30967e-6              |
| 551.71  | 1.19599e-6 | 1.76897e-6 | 1.18288e-6 | 7.54725e-7              |
| 571.98  | 6.59182e-7 | 1.29638e-6 | 9.06617e-7 | 5.83280e-7              |
| 654.81  | 1.33833e-6 | 3.47839e-6 | 3.44137e-6 | 1.31324e-6              |
| 654.89  | 1.43072e-6 | 5.23248e-6 | 2.25240e-6 | 1.46667e-6              |
| 762.32  | 1.74866e-6 | 2.03847e-6 | 1.85831e-6 | 9.07609e-7              |
| 1083.62 | 4.07681e-7 | 8.85690e-7 | 6.32261e-7 | 3.31608e-7              |
| 1283.86 | 1.18295e-6 | 1.38976e-6 | 1.13028e-6 | 7.48374e-7              |

Table 5.9: Ellipticity upper limits of known isolated pulsars using one year of V3 calibration of S5 data. Pulsar with frequency of 59.56 Hz is the Crab pulsar.

$I_{zz} = 10^{38} \text{kg m}^2$ . The upper limits of the ellipticity of all 33 analyzed pulsars for the three LIGO detectors and their combination are shown in Table (5.9). For these specific objects the same considerations made regarding the significance of the amplitude upper limits also apply to the ellipticity upper limits. However it is interesting to note that with ellipticity values now lower than  $10^{-6}$ , LIGO is indeed starting to probe potentially interesting regions of the parameter space.

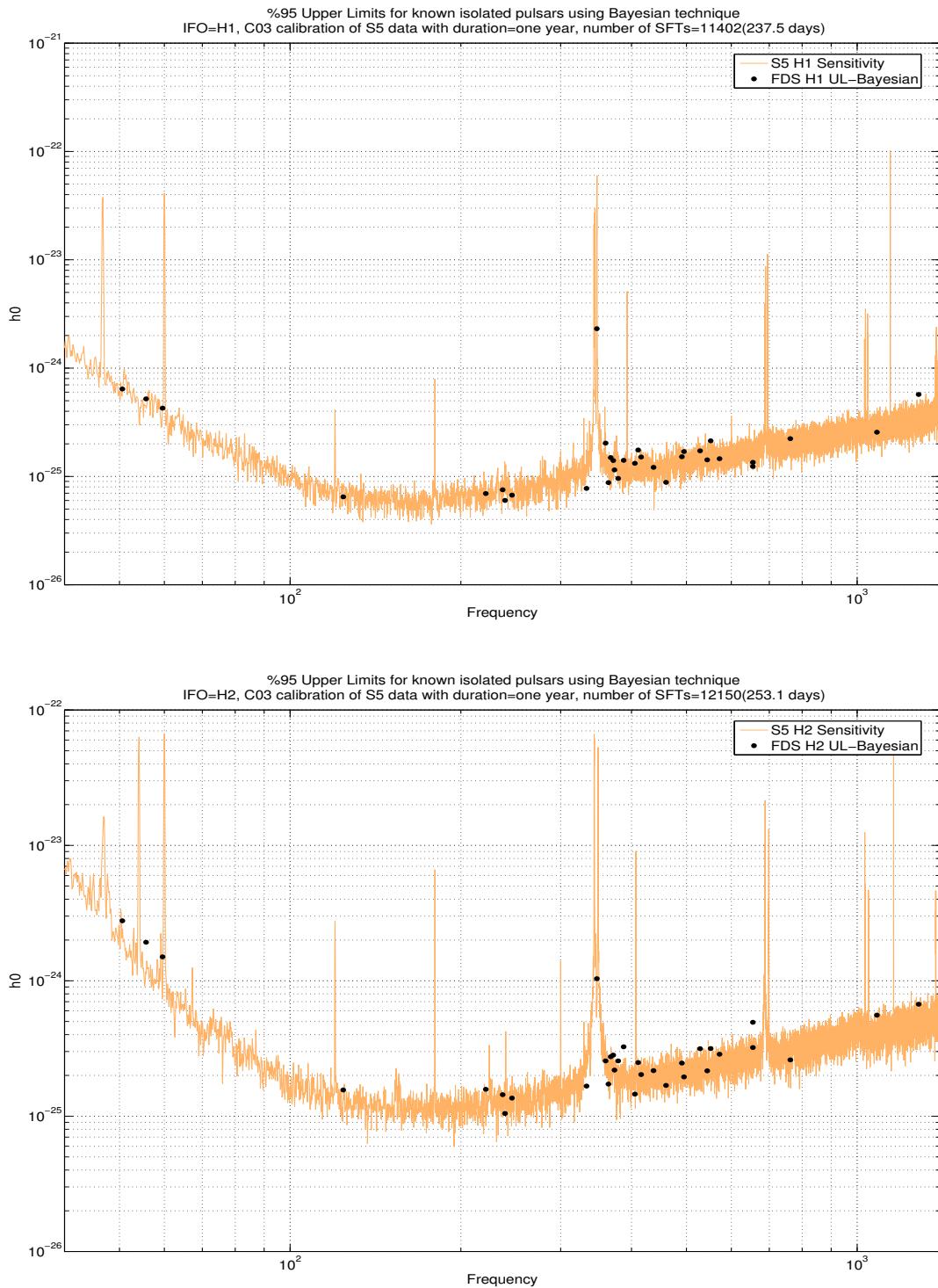


Figure 5.25: The upper limit on  $h_0$  using one year of C03 calibration of S5 data for H1 (top) and H2 (bottom) detector together with LIGO sensitivity curve.

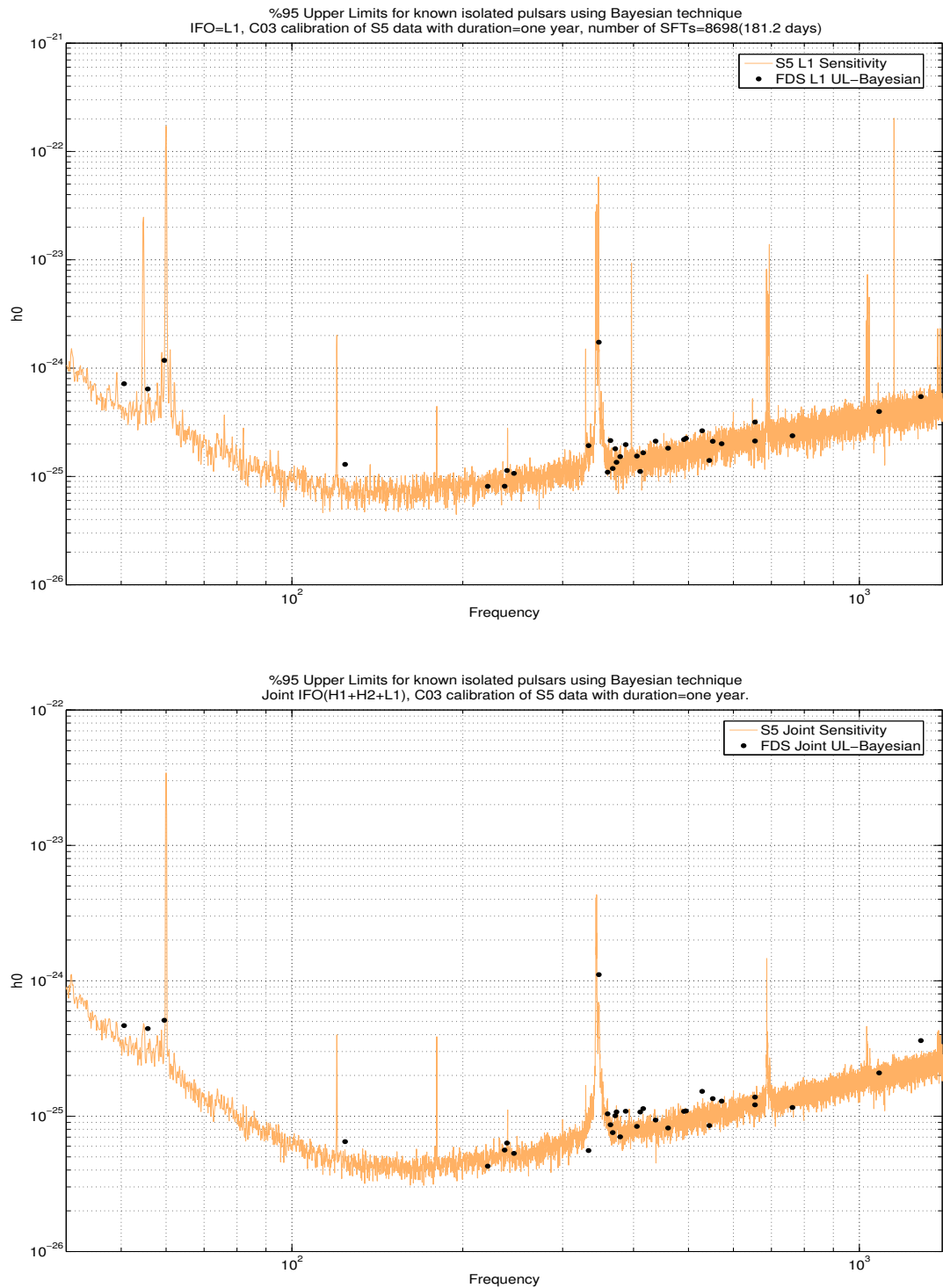


Figure 5.26: The upper limit on  $h_0$  using one year of C03 calibration of S5 data for L1 and Joint (H1+H2+L1) detector together with LIGO sensitivity curve.

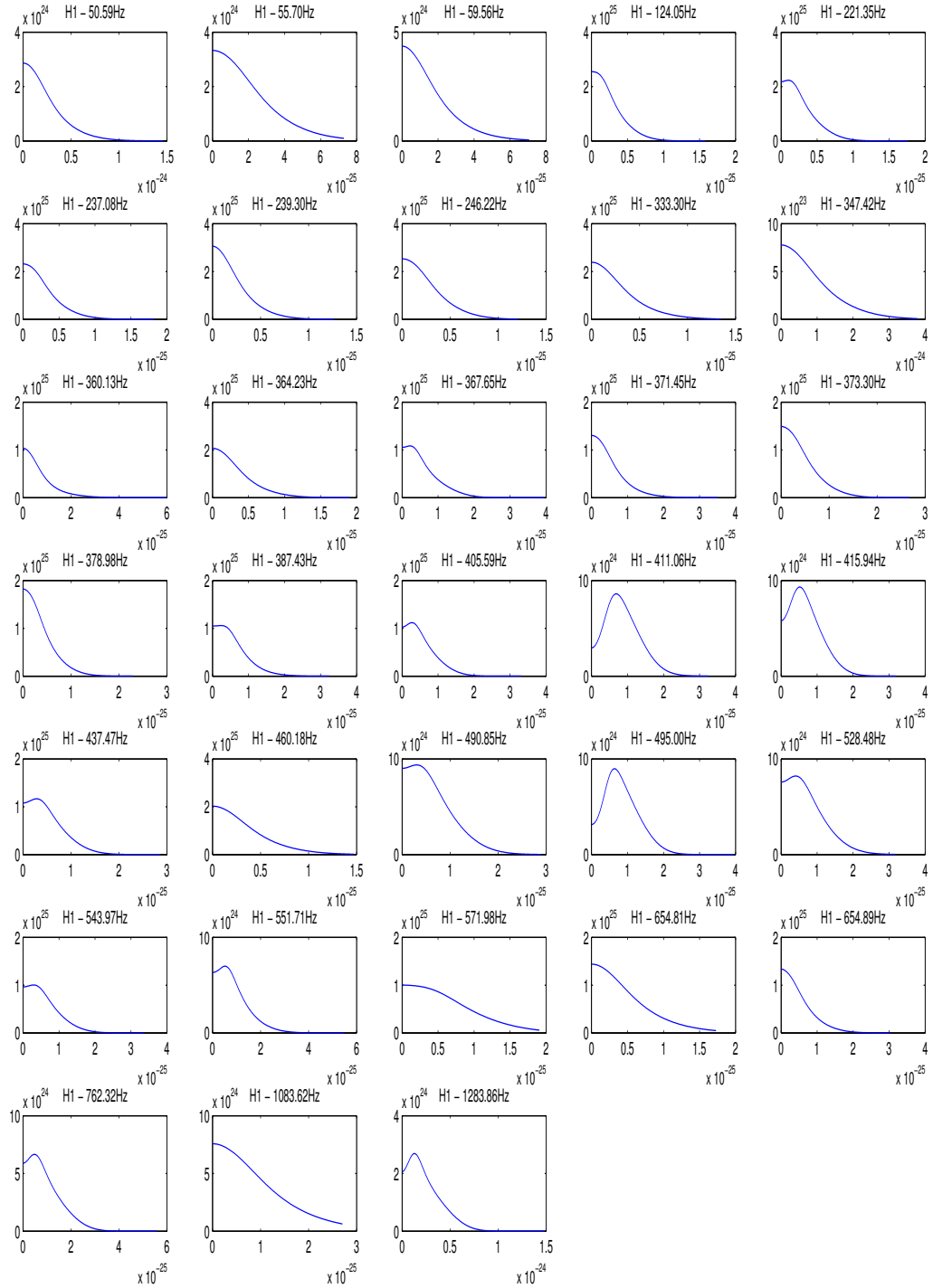


Figure 5.27: PDFs of known isolated pulsars of H1.

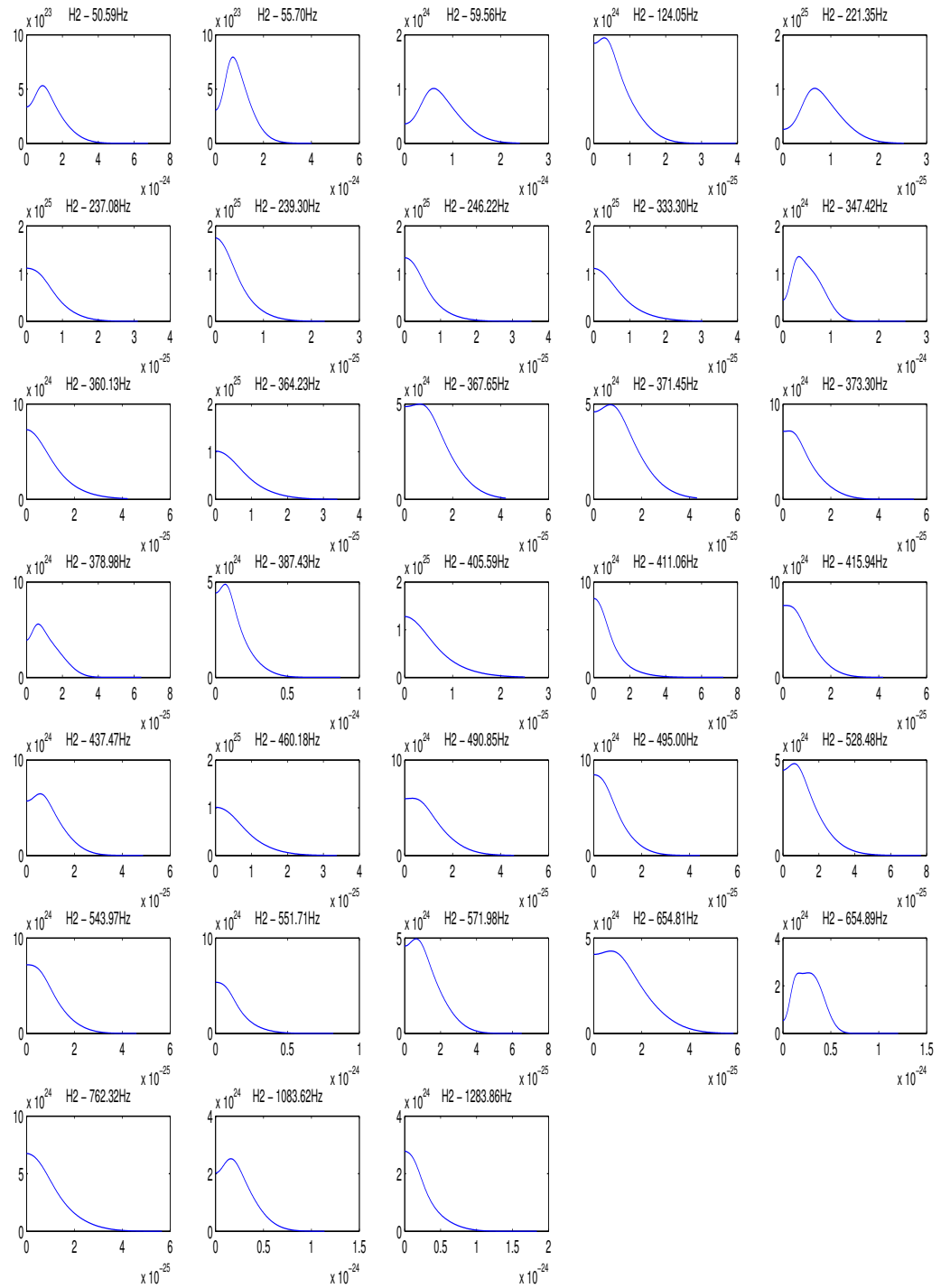


Figure 5.28: PDFs of known isolated pulsars of H2.

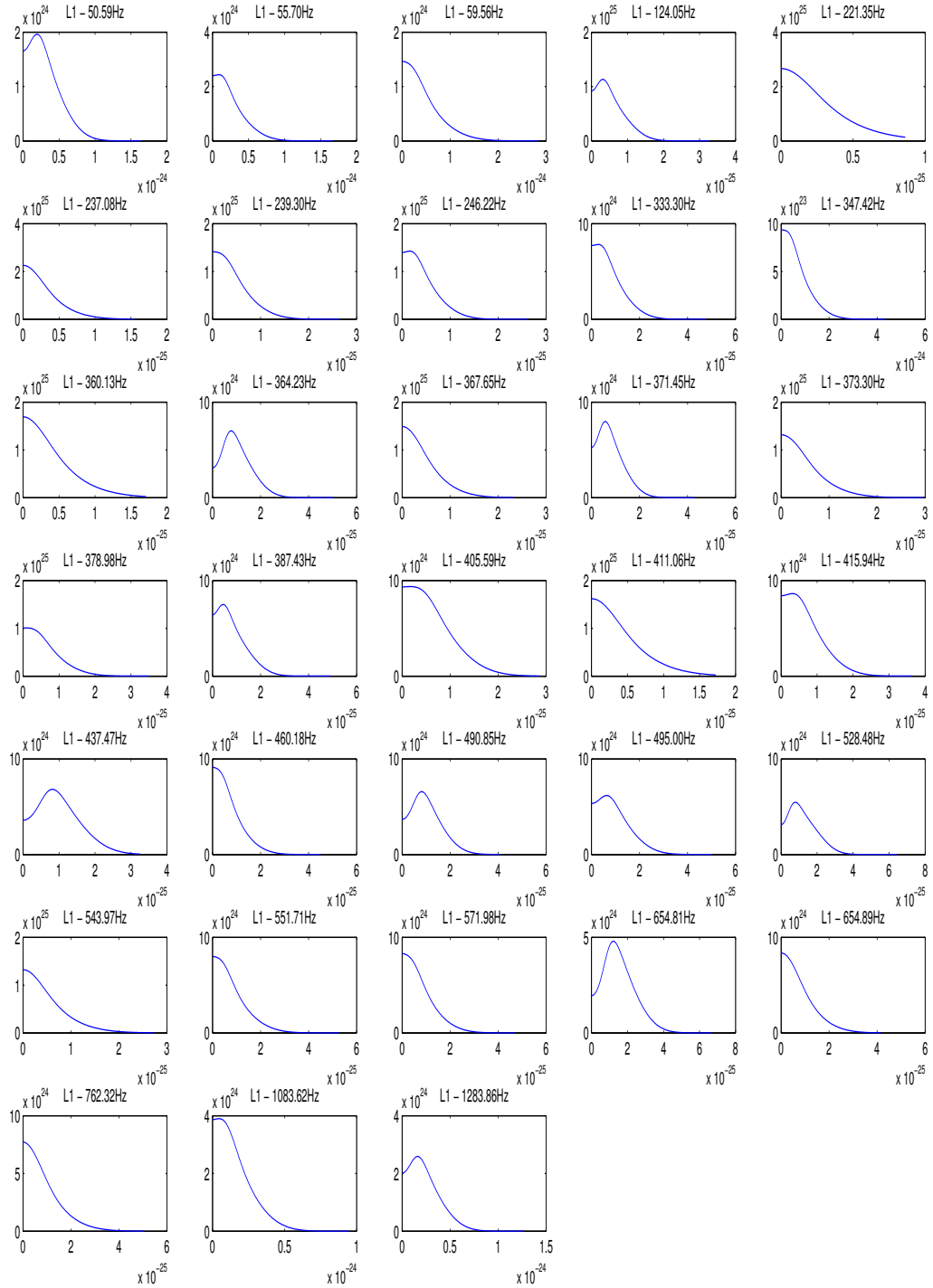


Figure 5.29: PDFs of known isolated pulsars of L1.



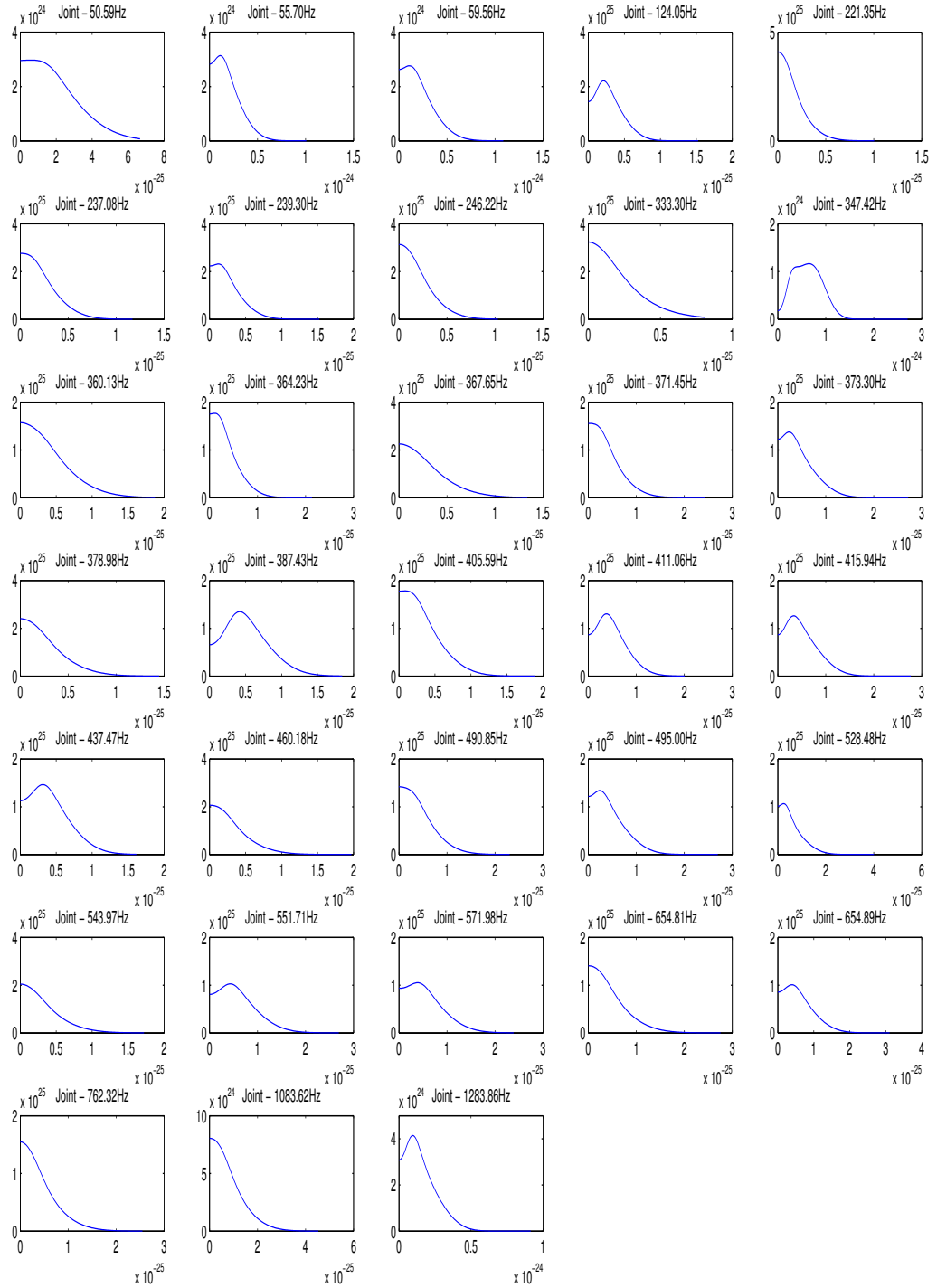


Figure 5.30: PDFs of known isolated pulsars of Joint IFO.



# Summary

Two projects have been carried out for this thesis. At the first, we have studied general hierarchical search strategies for gravitational waves from unknown, isolated GW pulsars. In particular, we have considered multistage hierarchical algorithms where, in the semi-coherent search algorithm, each stage (except the last) consists of a coherent demodulation of short stacks of data followed by appropriate sliding and stacking of the  $\mathcal{F}$ -statistics results from the different stacks. The successive stages serve to quickly reduce the number of candidates; they are followed by a final coherent follow-up stage to fully analyze the remaining candidates.

We have optimized this strategy by minimizing the computational cost  $P$  subject to the constraints which specify the total amount of data available and the desired confidence levels. Of course,  $P$  still depends on the size of the parameter space, in particular on the range of frequencies and spin-down ages that are searched over. Carrying out the optimization, and varying over the number of stages,  $n$ , we found that the advantages of the multistage approach saturate at  $n = 3$  (i.e.,  $n = 4$  and  $5$  are scarcely better) in both semi-coherent and coherent searches.

The reported values for the optimized search parameters,  $(N^{(i)}, \Delta T^{(i)}, \mu^{(i)})$  in the semi-coherent search and  $(T^{(i)}, \mu^{(i)})$  in the coherent search, are the result of a number of simplifications: i) our calculations have more than once used theoretical estimates of the operations counts instead of those obtained by profiling existing codes; ii) we have not considered issues of memory storage or the cost of performing Monte-Carlo simulations, and iii) the detector noise has throughout been assumed to be Gaussian and stationary. Furthermore, the template counting formulae used in this work are, strictly speaking, valid only for observation times significantly less than a year. Hence, our present estimates should not be considered as more than a rough guide for carrying out a search in practice; once better approximations become available, the values in question should be calculated anew. In spite of these limitations, we believe that our results do provide useful ballpark figures for the properties of an optimized all-sky search.

The present research opens up a number of avenues for future work:

- In order to obtain improved estimates for optimization, a re-analysis using actual search codes, applied to actual data would be a logical next step. In this context, one should then 1) profile the codes to determine the actual computational cost of the different operations, and 2) do Monte Carlo studies to determine the *actual*  $\alpha^{(i)}$  for different thresholds. (Recall that the formulae given here are based on the assumption stationary, Gaussian noise.) The latter could require considerable work, especially if the results are strongly frequency-dependent, with some bands being much “better behaved” than others.
- It would be straightforward to extend our work to searches that are less computationally challenging than all-sky ones, but that are still computationally limited; e.g., one could consider searches for unknown NSs in supernova remnants (such as SN 1987A), in which case the sky-position is known with great precision, but the frequency and spin-down parameters must be determined by the search. Similarly, one could consider a search over a small fraction of the sky, e.g., a portion containing only the Galactic center, or a specific portion of the disk.
- The formulae for operations counts, confidence levels etc. can also be derived for case when the Hough transform [62] instead of the stack-slide method is used in the semi-coherent stages; the optimization of multistage, hierarchical Hough-type searches would then proceed in the same way as developed here.
- We expect that the lessons learned in this project will carry over to searches for GW pulsars in

low-mass X-ray binaries, which are also limited by computational power [72]; the details, however, are yet to be worked out.

- The GW pulsar search problem treated here is similar to another more challenging search problem involving LISA data: the search for inspiral signals of stellar-mass compact objects captured by  $\sim 10^6 M_\odot$  BHs in galactic nuclei. Where computational power is concerned, this problem is an even greater challenge [89], and we expect that the lessons learned in this project will also be very useful in formulating and optimizing a search algorithm for LISA capture sources.
- In this project, we have tacitly assumed that the search is performed by a single computer or computing cluster. However, at least in the next few years, the most computationally intensive GW searches will be directed by `Einstein@Home` [1]. With this setup, there are additional constraints that we have not yet considered, involving factors such as the rate at which data and intermediate results can be sent back and forth between the `Einstein@Home` servers and participants' computers, and the amount of storage space available for use on participants' computers. We intend to study hierarchical searches in this context, as well, to see which – if any – of the lessons learned here need to be modified for application to `Einstein@Home`.

For the second project fulfilled as part of this thesis, we have extended and generalized an existing code from the LAL (`ComputeFStatistic.c` henceforth “CFS\_v1”) for computing the  $\mathcal{F}$ -statistic (the detection statistic, as discussed in this thesis); the result, a code named `ComputeFStatistic_v2.c` (“CFS\_v2”) has since been integrated into the LAL. The main improvements are that the CFS\_v2 can handle a multi-IFO (Interferometric Observatory) search; also, it uses SFT-specific noise-weights that allow a better handling of non-stationary noise. Another extension in CFS\_v2 is that it can estimate the signal parameters and outputs the estimated parameters of the loudest event. Due to use of modified LALDemod, CFS\_v2 is also faster than CFS\_v1.

In the first part of this project, we have validated our CFS\_v2, namely the results of a comparison between its output and that of the previous (and extensively tested) version CFS\_v1. In this validation process, we have seen that CFS\_v2 and CFS\_v1 are quite consistent in all the cases such as pure noise, signal plus noise as well as signal only data. It turned out that in a noise dominated data, as a pure noise or a data with a weak signal, using our noise weight option gives a discrepancy of the range of -25% to 25% between CFS\_v2 and CFS\_v1. However, by disabling this option the two codes show a very good consistency.

We have also tested our code by using the hardware-injected data from the fourth LIGO Science run (S4). As the result, we have compared the parameters of the injected signals with those of estimated by our software. They show a very good agreement.

In the next part of this project, we presented our results using CFS\_v2 to analyze the S5 data to set an upper limit on the 33 known isolated pulsars. These pulsars are known from radio-astronomical data; their parameters have been taken from the ATNF (Australia Telescope National Facility). We compared our results with those obtained in the Time Domain Search (TDS), which uses a completely different approach. Our results show a discrepancies between -30% to 30%, which are, again, largely due to how the search techniques “react” to non-stationary and to non-Gaussian noise. Finally, we have used one year of the C03 calibration of S5 data to set the upper limit on the strength of gravitational wave amplitude of the above mentioned 33 pulsars. This analysis shows that by using a one year data, the best sensitivity that can be achieved using a single detector is  $6.0 \times 10^{-26}$  corresponding to H1 detector. A multi-IFO search using three LIGO detectors shows that the best joint sensitivity is  $4.27 \times 10^{-26}$ . Based on these upper limits on the strength of GW, we have also set the upper limits on the ellipticity of all 33 mentioned pulsars for each LIGO detectors and their combination.

Although very extensive work have been done in frequency domain by using our newly improved software, still there are many avenues to explore in the future. For example, the possibility of searching for signals from the binary systems should be integrated in our current software, a full analysis over the entire data set needs to be performed, a more sophisticated use of the information that the electromagnetic observations provide (for example for the Crab pulsar), finally directed searches towards interesting regions in the sky, for example looking at our Galaxy center which is assumed to be the center of many massive black holes, should be carried out

## Appendix A

# Computational cost of the SFT method

Here we estimate the computational cost (in floating point operations) of calculating the  $\mathcal{F}$ -statistic via the SFT method. This result is used in Sec 3.3.3.

The SFT method was already described in Sec 4.2 in details. Eqs. (4.76) and (4.77) are our final approximations for  $F_a$  and  $F_b$ . The final  $\mathcal{F}$ -statistic is calculated from  $F_a$  and  $F_b$  using Eq. (4.35). Thus, with the SFT method, for each point  $\boldsymbol{\lambda}$  in parameter space we need to calculate  $Q_\alpha(\boldsymbol{\lambda})$ ,  $P_{\alpha,k}(\boldsymbol{\lambda})$ , and the amplitude modulation functions,  $a_\alpha(t)$  and  $b_\alpha(t)$ , and then to perform the sums in Eqs. (4.76) and (4.77). It is then easy to see that to calculate the  $\mathcal{F}$ -statistic for  $n$  frequency bins, for a fixed value of  $\vec{\lambda}$ , the number of floating point operations required is roughly some constant times  $nMD$ . To estimate the constant, let  $C_1$  be the cost of calculating  $P_{\alpha,k}$  for each  $\alpha$  and  $k$  value. Since multiplying two complex numbers requires 6 operations, and adding two complex numbers requires 2 operations, we see that calculating the sum over  $k'$  in Eq. (4.76) requires

$$(C_1 + 6)(2D + 1) + 4D \quad (\text{A.1})$$

operations. Similarly, if  $C_2$  is the cost of calculating  $\hat{a}_\alpha Q_\alpha$  (Eq. 4.76) for every  $\alpha$ , then the cost of calculating  $F_a$  is

$$M[(C_1 + 6)(2D + 1) + 4D] + MC_2 + 6M + 2(M - 1). \quad (\text{A.2})$$

Thus, to find  $F_a$  and  $F_b$  for every frequency bin requires  $\approx 2 \times (2C_1 + 16)DM$  operations. Since the cost of combining  $F_a$  and  $F_b$  to get  $\mathcal{F}$  is negligible compared to this, and assuming  $C_1$  to be of order unity, we see that the operation count for calculating the  $\mathcal{F}$ -statistic for  $n$  frequency bins is approximately

$$\sim 40nMD. \quad (\text{A.3})$$

For the first stage in the hierarchical search, the  $\mathcal{F}$ -statistic is evaluated for  $N^{(1)}N_{pc}^{(1)}f_{\max}\Delta T$  bins. So, taking  $D = 16$  and using  $M = \Delta T^{(1)}/T_{\text{sft}}$ , the cost is

$$\approx 640N^{(1)}N_{pc}^{(1)}f_{\max}\frac{(\Delta T^{(1)})^2}{T_{\text{sft}}}. \quad (\text{A.4})$$

At higher stages, we evaluate the  $\mathcal{F}$ -statistic

$$F^{(i-1)} \max \left\{ 1, \frac{N_{pc}^{(i)}}{N_{pf}^{(i-1)}} \right\} \max \left\{ 1, \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right\} N^{(i)} \quad (\text{A.5})$$

times, so the operations count is

$$F^{(i-1)} \max \left\{ 1, \frac{N_{pc}^{(i)}}{N_{pf}^{(i-1)}} \right\} \max \left\{ 1, \frac{\Delta T^{(i)}}{\Delta T^{(i-1)}} \right\} \times \left[ \frac{640N^{(i)}\Delta T^{(i)}}{T_{\text{sft}}} \right]. \quad (\text{A.6})$$



## Appendix B

# Computer codes used in optimization strategies

### B.1 Main file

```
/******  
 * Optimization program for Hierarchical Stack-Slide pulsar searches  
******/  
  
/* Author: Iraj Gholami and Badri Krishnan */  
  
#include "Header.h"  
  
double *qtry=NULL; double tt; long idum;  
  
int main(int argc, char *argv[]) {  
    int L;  
    double tauyear[] = {25.0, 40.0, 100.0, 1000.0, 10000.0, 100000.0,  
                        1000000.0, 10000000.0, 100000000.0, 1000000000.0};  
    for(L=1; L<=5; L++)  
    {  
  
        /* Calculating the time elapsed for running this Code.*/  
        clock_t start, end;  
        double cpu_time_used;  
  
        start = clock();  
        /*******/  
  
        int nDim, i, j, k;  
        int count;  
        double *values=NULL;  
        double **p= NULL;  
        double *factor=NULL;  
        double *pb=NULL; /* Values of the simplices which indicate the best  
                           minimum*/  
  
        params par;  
        /* starting point */  
        double TStart[] = {1.00, 1.00, 1.00, 1.00, 1.00, 1.00,  
                           1.00, 1.0, 1.0, 1.0};  
        double MuStart[] = {0.6, 0.1, 0.7, 0.4, 0.4, 0.4, 0.4,  
                             0.4, 0.4, 0.4};  
        double NStart[] = {50.0, 50.0, 50.0, 50.0,50.0, 50.0,  
                            150.0,150.0,150.0,150.0};  
  
        double tempvalues[100000];  
  
        double yb; /* A large number involved to be the initial best  
                   minimum used in Simulated Annealing Algorithm.  
                   The best minimum in the rest of the code is  
                   replaced to this parameter*/  
        double tempr; /* The Temperature used in the Simulated Annealing  
                       Method*/  
        int iter; /* Number of Iteration in fixed temperature */  
        FILE *fp=NULL; /* output file */  
  
        /* default values for InParams */  
  
        par.InPar.tauMin = tauyear[1] * YEARSEC; /* Smallest spindown age of  
                                                  pulsar searched over*/  
        par.InPar.fMax = 1000.0; /* largest frequency searched over */
```

```

par.InPar.ModeFlag = 0;
par.InPar.SkyFlag = 1;
par.InPar.h2Sn = 5.0e-05;
par.InPar.Tsft = 1800;
par.InPar.MaxTime = 0.5*YEARSEC;
par.InPar.nStages = L; /* largest number of stages included in search */
nDim = 3*par.InPar.nStages; /*number of parameters to be optimized over*/
par.InPar.beta = NULL;
par.InPar.beta = vector(1, par.InPar.nStages);
par.InPar.beta[1] = 0.10;
for (i=2; i <= par.InPar.nStages; i++)
    par.InPar.beta[i] = 0.01;

fp = fopen("hstack.out", "a");

/* allocate memory */
pb = vector(1,nDim);
p = matrix(1,nDim+1,1,nDim);
values = vector(1,nDim+1);
factor = vector(1,nDim);

par.OutPar.FalseAlarm = NULL;
par.OutPar.CDF = NULL;
par.OutPar.Npf = NULL;
par.OutPar.Npc = NULL;
par.OutPar.Threshold = NULL;
par.OutPar.Compt = NULL;
par.OutPar.inCohCompt = NULL;
par.OutPar.CohCompt = NULL;

par.OutPar.Npf = vector(1,par.InPar.nStages);
par.OutPar.Npc = vector(1,par.InPar.nStages);
par.OutPar.Compt = vector(1,par.InPar.nStages + 1);
par.OutPar.inCohCompt = vector(1,par.InPar.nStages);
par.OutPar.CohCompt = vector(1,par.InPar.nStages);
par.OutPar.FalseAlarm = vector(1,par.InPar.nStages);
par.OutPar.CDF = vector(1,par.InPar.nStages);
par.OutPar.Threshold = vector(1,par.InPar.nStages);

qtry=vector(1,nDim);

/* choose starting point */
for (i=1; i<= par.InPar.nStages; i++)
{
    p[1][3*(i-1)+1] = TStart[i-1] * 86400.0;
    p[1][3*(i-1)+2] = MuStart[i-1];
    p[1][3*(i-1)+3] = NStart[i-1];
}

/* choosing other points of simplices: x_i[j] = p[1][j] + factor[j] */
/* first choose factor[i] which sets sizes of the simplex in the
different directions */
for (i=1; i<=par.InPar.nStages; i++)
{
    factor[3*(i-1)+1] = 0.05 * 86400.0;
    factor[3*(i-1)+2] = 0.005;
    factor[3*(i-1)+3] = 1.0 /*0.1*/;
}

/* now add factor to p[1] to get other points of simplex */
for (i=1; i<=nDim; i++)
{
    for (j=1; j<=nDim; j++)
    {
        k = i-j;
        if ((!k)) p[i+1][j] = p[1][j] + factor[j];
        else p[i+1][j] = p[1][j];
    }
}

/* calculate values of function at simplices */
for (i=1; i <= nDim+1; i++)
    values[i] = funk(p[i], &par);

/***** Minimization *****/
temptr = 5.0e6;
yb = 1.0e21;
count=0;
do
{
    iter=500;          /* Initial value 1000*/
    count+=1;
    if(count%25==0)   /* Initial value = 25*/
        temptr = 0.98*temptr; /* Initial value (temptr * 0.97)*/

    tempvalues[count] = yb;
}

```



```

    if(count%30==0)
    {
        if(tempvalues[count]==tempvalues[count-25])
            temptr = 1.015*temptr; /* Initial value (temptr * 1.015)*/
        }
    minimize(p, values, nDim, pb, &yb, TINY, funk, &iter, temptr, &par);
    printf(" %d iter = %d temptr = %1.2e result = %1.14e
           bestresult = %1.14e\n", count, iter, temptr, values[1], yb);
    }
while(iter<0 && temptr > 1.0e-7);

/* print to output file */
PrintInParams(fp, &par);
PrintOutParams(fp, &par);

fprintf(fp, "Stage      DeltaT (days)      muMax      N
           Tobs (days)\t SNR \n");
PrintValues(fp, pb, &par);
fprintf(fp, "\n");
fprintf(fp, "Minimum computational power required: %6.3e Flops\n", yb);

end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;

printf("\n\n Time Used = %.2f Sec ( %.2f Min )\n\n",
       cpu_time_used, cpu_time_used/60);
fprintf(fp, "\nTime Used = %.2f Sec ( %.2f Min )\n",
       cpu_time_used, cpu_time_used/60);

fclose(fp);

/* free memory */
free_vector(par.InPar.beta, 1, par.InPar.nStages);
free_vector(par.OutPar.Npf, 1, par.InPar.nStages);
free_vector(par.OutPar.Npc, 1, par.InPar.nStages);
free_vector(par.OutPar.Compt, 1, par.InPar.nStages+1);
free_vector(par.OutPar.inCohCompt, 1, par.InPar.nStages+1);
free_vector(par.OutPar.CohCompt, 1, par.InPar.nStages+1);
free_vector(par.OutPar.FalseAlarm, 1, par.InPar.nStages);
free_vector(par.OutPar.Threshold, 1, par.InPar.nStages);
free_vector(par.OutPar.CDF, 1, par.InPar.nStages);
free_vector(qtry, 1, nDim);
free_vector(pb, 1, nDim);
free_matrix(p, 1, nDim+1, 1, nDim);
free_vector(values, 1, nDim+1);
free_vector(factor, 1, nDim);
}
return 0;
}

/*-----*/
void PrintValues(FILE *fp, double x[], params *par) {
    int i;

    for(i=0; i<par->InPar.nStages; i++)
    {
        fprintf(fp, "\n");
        fprintf(fp, "%d      %12.3lf      %5.4lf      %7.3lf      %11.3lf\t %5.2lf\n",
                i+1, x[3*i+1]/86400.0, x[3*i+2], x[3*i+3], (x[3*i+1]/86400.0)*
                x[3*i+3], sqrt(par->InPar.h2Sn*(x[3*i+1])* x[3*i+3]));
    }

    return;
}

void PrintInParams(FILE *fp, params *par) {
    int nStages, i;

    nStages = par->InPar.nStages;
    fprintf(fp, "\n##### INPUT #####\n");
    if ( (par->InPar.ModeFlag == 1) )
        fprintf(fp, "BC mode\n");
    else
        fprintf(fp, "Fresh mode\n");

    if ( (par->InPar.SkyFlag == 1) )
        fprintf(fp, "All sky search\n");
    else
        fprintf(fp, "Directed search\n");

    fprintf(fp, "Minimum spindown age: %4.2lf years\n",
            par->InPar.tauMin / YEARSEC);
    fprintf(fp, "Maximum frequency searched over: %6.2lf Hz\n",
            par->InPar.fMax);

    fprintf(fp, "False dismissal rates: ");

```

```

for (i = 1; i <= nStages; i++)
    fprintf(fp, "%4.3lf ", par->InPar.beta[i]);
fprintf(fp, "\n");

fprintf(fp, "Smallest signal that can cross thresholds: %5.2e\n",
        par->InPar.h2Sn);
fprintf(fp, "%4.2lf-Year Signal to Noise Ratio (SNR): %5.2lf\n",
        par->InPar.MaxTime/YEARSEC, sqrt(par->InPar.h2Sn*
        par->InPar.MaxTime));
fprintf(fp, "Total data available: %4.2lf years\n",
        par->InPar.MaxTime/YEARSEC);
fprintf(fp, "Length of sfts if sft method used: %6.2lf seconds\n",
        par->InPar.Tsft);
fprintf(fp, "\n");
return;
}

void PrintOutParams(FILE *fp, params *par) {
    int nStages, i;
    nStages = par->InPar.nStages;

    fprintf(fp, "##### OUTPUT #####\n\n");

    fprintf(fp, "False alarm rates: ");
    for(i = 1; i<=nStages; i++)
        fprintf(fp, "%8.3e ",(par->OutPar.FalseAlarm[i] ));
    fprintf(fp, "\n");

    fprintf(fp, "False alarm prob.: ");
    for(i = 1; i<=nStages; i++)
        fprintf(fp, "%8.3e ", (par->OutPar.CDF[i] ));
    fprintf(fp, "\n");

    fprintf(fp, "Npf: ");
    for(i = 1; i<=nStages; i++)
        fprintf(fp, "%6.3e ", par->OutPar.Npf[i]);
    fprintf(fp, "\n");

    fprintf(fp, "Npc: ");
    for(i = 1; i<=nStages; i++)
        fprintf(fp, "%6.3e ", par->OutPar.Npc[i]);
    fprintf(fp, "\n");

    fprintf(fp, "NpfCoh: %6.3e\n", par->OutPar.NpfCoh);

    fprintf(fp, "Thresholds: ");
    for(i = 1; i<=nStages; i++)
        fprintf(fp, "%6.3e ", par->OutPar.Threshold[i]);
    fprintf(fp, "\n");

    fprintf(fp, "\nComputational Cost: \n");
    fprintf(fp, "Incoherent Part: \t ");
    for(i = 1; i<=nStages; i++)
        fprintf(fp, "%6.3e ", par->OutPar.inCohCompt[i]);
    fprintf(fp, "\n");
    fprintf(fp, "Coherent Part: \t\t ");
    for(i = 1; i<=nStages; i++)
        fprintf(fp, "%6.3e ", par->OutPar.CohCompt[i]);
    fprintf(fp, "\n");
    fprintf(fp, "Total: \t\t\t ");
    for(i = 1; i<=nStages+1; i++)
        fprintf(fp, "%6.3e ", par->OutPar.Compt[i]);
    fprintf(fp, "\n");

    fprintf(fp, "\nTotal amount of data: %6.2lf years\n",
            par->OutPar.TcoH/YEARSEC);

    fprintf(fp, "\n");

    return;
}

```

## B.2 Header file (Header.h)

```

/*****
 * Header file for hstack.c and its modules
 *****/
/* Authors: Iraj Gholami and Badri Krishnan */

/* double inclusion protection */
#ifndef HSTACKH
#define HSTACKH

```

```

#include<unistd.h> #include<math.h> #include<stdlib.h>
#include<stdio.h> #include <time.h> #include<gsl/gsl_cdf.h>

/* used in amebssa */
#define GET_PSUM \
    for (j=1; j<=nDim; j++){\
        for (sum=0.0, i=1; i <= mpts; i++) sum += p[i][j];\
        psum[j] = sum;}

/* used in amebssa */
#define SWAP(a,b) {swap = (a); (a)=(b); (b) = wap;}

#define TINY 1.0e-12

#define PIE 3.1415926535897932 /* yearly ang. vel. of earth in rad/s*/

#define OMEGA 1.99238499e-07

/* speed of light in m/s */
#define LIGHT 3.00e08

/* one astronomical unit in metres */
#define AU 149597870691.0

/* number of seconds in a year */
#define YEARSEC 31557600.0

/* Elements needed for the ran1 function*/
#define IA 16807

#define IM 2147483647
#define IQ 127773
#define IR 2836

/* Elements needed for the ran2 function*/

#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)

#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 2774
#define IR1 12211
#define IR2 3791
#define NTAB 32

#define NDIV (1+IMM1/NTAB)
#define EPS 1.0e-9
#define RMX (1.0-EPS)

/* global variables */
extern double *qtry; extern long idum; extern double tt;

/* this structure contains parameters that must be specified in the
beginning as requirements/condition */
typedef struct tagInParams {
    double tauMin; /* minimum spindown age of pulsar in seconds */
    double fMax; /* largest frequency searched over in Hz */
    int nStages; /* number of incoherent stages in hierarchical search */
    int ModeFlag; /* 1 for BC mode and Fresh otherwise */
    int SkyFlag; /* 1 for all-sky and directed otherwise */
    double h2Sn; /* signal strength to search for */
    double Tsft; /* length of sfts if we use sft method in sec */
    double MaxTime; /* total amount of data available in sec*/
    double *beta; /* desired false dismissal rates */
} InParams;

/* these are the quantities that can be calculated if we know the InParams
and the values of N[i], DeltaT[i] and muMax[i] */
typedef struct tagOutParams{
    double *FalseAlarm; /* false alarm rates at each step (this is 1-CDF) */
    double *CDF; /* CDF part of false alarm */
    double *Npf; /* points in fine grid in each stage */
    double *Npc; /* points in coarse grid in each stage */
    double NpfCoh; /* points in grid used for final coherent stage */
    double *Threshold; /* threshold set at each incoherent stage */
    double *Compt; /* number of floating point operations at each stage */
    double *inCohCompt; /* number of floating point operations at each stage
for Incoherent part*/
    double *CohCompt; /* number of floating point operations at each stage for
Coherent part*/
    double Tcoh; /* observation time considered in each stage */
} OutParams;

```

```

/* the grand structure containing all the information except N[i], DeltaT[i]
   and muMax[i] which are to be optimized over */
typedef struct tagParams {
    InParams InPar;
    OutParams OutPar;
} params;

/*****/ /* function prototypes */

double funk(double x[], params *par);

double testfun(double x[], params *par);

void CalcOutParams( params *par, double x[]);

void PrintValues( FILE *fp, double x[], params *par);

void PrintOutParams(FILE *fp, params *par);

void PrintInParams(FILE *fp, params *par);

double TotalTime(double x[], params *par);

double FDismissal(double, double, double, double, double, double);

void FalseAlarm (double x[], params *par);

void Flops1(double, double, double, params *par);

double Flops2(int, double x[], params *par);

double FlopsCoh(double x[], params *par);

double Threshold(double , double , double, double, double);

double NP(double , double , double, double, double);

double Ns(int s, double deltaT, double muMax, double fMax, double tauMin);

double NPallsky(double deltaT, double muMax, double N, double fMax, double tauMin);

double MskyS(double deltaT, double muMax, double N, int s, double fMax);

double stirling(double );

double Gs(int s, double N);

double ran1(long *idum);

double ran2(long *idum);

int factorial(int N);

void minimize(double **p, double values[], int nDim, double pb[],
              double *yb, double fTol, double (*funk)(double [], params*),
              int *iter, double temptr, params *par);

double amotsa(double **p, double values[], double psum[], int nDim,
              double pb[], double *yb, double (*funk)(double [], params *par),
              int ihi, double *yhi, double fac, params *);

double *vector(long nl, long nh);

double **matrix(long nrl, long nrh, long ncl, long nch);

void free_vector(double *v, long nl, long nh);

void free_matrix(double **m, long nrl, long nrh, long ncl, long nch);

void nrerror(char error_text[]); /* end of function prototypes */

/* end double inclusion protection */ #endif

```

### B.3 Sensitivity, False Alarm, False Dismissal

```

/*****
 * Functions for sensitivity, false alarm, false dismissal and computing power
 *****/

#include "hstack.h"

/* Function to be minimized: sensitivity for directed search */ /*
   x[1] : deltaT1

```

```

x[2] : MuMax1
x[3] : N1 */

double funk(double x[], params *par)
{
  int i, nStages;
  double constraint, output;
  double C, TotTime;

  nStages = par->InPar.nStages;

  /* The below constraints say that the Total Observational Time should be
  less than a year, Threshold should get higher in the next stage, False
  Alarm rate should get lower in the next stage. Finally for the BC-Mode, the
  observational time in each stage should be longer than its previous one.*/

  CalcOutParams(par, x);
  TotTime = par->OutPar.TcoH;

  C = 0.0;
  for (i=1; i<=nStages+1; i++)
    C += par->OutPar.Compt[i]/TotTime;

  if (par->InPar.ModeFlag == 1)
  {
    constraint = 1.0
    + exp(10.0 * (TotTime - par->InPar.MaxTime))* T_Obs<=one year.*/
    + exp(100.0 * (0.01*86400.0 - x[1]))/*Constraint on DeltaT > 0.01 */
    + exp(100.0 * (x[2] - 0.90)) /*Constraint on Mu < 0.90 */
    + exp(100.0 * (0.001 - x[2])) /*Constraint on Mu > 0.001 */
    + exp(100.0 * (5.0 - x[3])) /*Constraint on N > 1.0 */
    + exp(100.0 * (par->OutPar.CDF[nStages] - 0.0099));/*CDF[nStages]<0.01*/

    for(i=2; i<=nStages; i++)
      constraint += exp(10.0 * (x[3*(i-2)+1]*x[3*(i-2)+3]/(x[3*(i-1)+1]*
      x[3*(i-1)+3]) - 1.0 ))
      + exp(100.0 * (0.01*86400.0 - x[3*(i-1)+1]))/*Constraint on DeltaT>0.01*/
      + exp(100.0 * (x[3*(i-1)+2] - 0.90)) /*Constraint on Mu < 0.90 */
      + exp(100.0 * (0.001 - x[3*(i-1)+2])) /*Constraint on Mu > 0.001 */
      + exp(100.0 * (5.0 - x[3*(i-1)+3])); /*Constraint on N > 1.0 */
  }
  else
  {
    constraint = 1.0
    + exp(10.0 * (TotTime - par->InPar.MaxTime))* T_Obs<=one year.*/
    + exp(100.0 * (0.01*86400.0 - x[1]))/*Constraint on DeltaT > 0.01 */
    + exp(100.0 * (x[2] - 0.90)) /*Constraint on Mu < 0.90 */
    + exp(100.0 * (0.001 - x[2])) /*Constraint on Mu > 0.001 */
    + exp(100.0 * (5.0 - x[3])) /*Constraint on N > 1.0 */
    + exp(100.0 * (par->OutPar.CDF[nStages] - 0.0099));/*CDF[nStages]<0.01*/

    for(i=2; i<=nStages; i++)
      constraint += exp(100.0 * (0.01*86400.0 - x[3*(i-1)+1]))
      /*Constraint on DeltaT > 0.01 */
      + exp(100.0 * (x[3*(i-1)+2] - 0.90)) /*Constraint on Mu < 0.90 */
      + exp(100.0 * (0.001 - x[3*(i-1)+2])) /*Constraint on Mu > 0.001 */
      + exp(100.0 * (5.0 - x[3*(i-1)+3])); /*Constraint on N > 1.0 */
  }

  output = C*constraint;
  return output;
}

void CalcOutParams(params *par, double x[]) {
  double *deltaT=NULL;
  double *muMax=NULL;
  double *N=NULL;
  int i, nStages;
  double fMax, tauMin;

  nStages = par->InPar.nStages;
  fMax = par->InPar.fMax;
  tauMin = par->InPar.tauMin;

  deltaT = vector(1,nStages);
  muMax = vector(1,nStages);
  N = vector(1,nStages);

  for (i = 1; i <= nStages; i++)
  {
    deltaT[i] = x[3*(i-1)+1];
    muMax[i] = x[3*(i-1)+2];
    N[i] = x[3*(i-1)+3];
    par->OutPar.Threshold[i] = Threshold(deltaT[i], muMax[i], N[i],
    par->InPar.h2Sn, par->InPar.beta[i]);
  }

  for (i=1; i<=nStages; i++)

```

```

{
  if(par->InPar.SkyFlag==1)
  {
    par->OutPar.Npf[i] = NPallsky(deltaT[i], muMax[i], N[i], fMax, tauMin);
    par->OutPar.Npc[i] = NPallsky(deltaT[i], muMax[i], 1.0, fMax, tauMin);
  }
  else
  {
    par->OutPar.Npf[i] = NP(deltaT[i], muMax[i], N[i], fMax, tauMin);
    par->OutPar.Npc[i] = NP(deltaT[i], muMax[i], 1.0, fMax, tauMin);
  }
}

par->OutPar.Tcoh = TotalTime(x, par);
par->OutPar.NpfCoh = (par->InPar.SkyFlag == 1) ? NPallsky(par->OutPar.Tcoh,
  0.2, 1.0, fMax, tauMin) :
  NP(par->OutPar.Tcoh, 0.2, 1.0, fMax, tauMin);

FalseAlarm(x, par);

Flops1(deltaT[1], muMax[1], N[1], par);
for (i=2; i<=nStages; i++)
  par->OutPar.Compt[i] = Flops2(i, x, par);
par->OutPar.Compt[nStages+1] = FlopsCoh(x, par);

free_vector(deltaT,1,nStages);
free_vector(N,1,nStages);
free_vector(muMax,1,nStages);

return;
}

/*-----*/
/*Function to compute the total time: TotalTime = sum(N_i *Delta T_i) */
/*Note: this is only relevant in "fresh mode" and not in "BC mode" */

double TotalTime(double x[], params *par)
{
  double temp = 0.0;
  int i;

  if ( (par->InPar.ModeFlag == 1) )
  {
    for (i=1; i<=par->InPar.nStages; i++)
    {
      if (x[3*(i-1)+1] * x[3*(i-1)+3] > temp)
        temp = x[3*(i-1)+1] * x[3*(i-1)+3];
    }
  }
  else
  {
    for (i=1; i <= par->InPar.nStages; i++)
      temp += x[3*(i-1)+1] * x[3*(i-1)+3];
  }
  return temp;
}

/*-----*/
/* Function for log of eq.(5.2) of Brady-Creighton for false alarm rate */
/*Function for log of False Alarm for nth-Stage in Fresh and BC Mode. */

void FalseAlarm (double x[], params *par)
{
  double B, C, fMax;
  int i, nStages;
  double *deltaT=NULL;
  double *muMax=NULL;
  double *N=NULL;

  fMax = par->InPar.fMax;
  nStages = par->InPar.nStages;

  deltaT = vector(1,nStages);
  muMax = vector(1,nStages);
  N = vector(1,nStages);

  for (i=1; i<=nStages; i++)
  {
    deltaT[i] = x[3*(i-1)+1];
    muMax[i] = x[3*(i-1)+2];
    N[i] = x[3*(i-1)+3];
  }
  /* this is actually 1-CDF */
  {
    par->OutPar.CDF[1] = gsl_cdf_gaussian_Q(par->OutPar.Threshold[1]

```

```

        - 2*N[1], sqrt(2*N[1]));
    par->OutPar.FalseAlarm[1] = fMax *deltaT[1] * par->OutPar.Npf[1]
        * par->OutPar.CDF[1];
}
if ( (par->OutPar.FalseAlarm[1] < 1.0))
    par->OutPar.FalseAlarm[1] = 1.0;

/* if we are in BC mode */
if (par->InPar.ModeFlag == 1)
{
    for (i=2; i<=nStages; i++)
    {
        /* log of the CDF */
        par->OutPar.CDF[i] = gsl_cdf_gaussian_Q(par->OutPar.Threshold[i]
            - 2*N[i], sqrt(2*N[i]));
        par->OutPar.FalseAlarm[i] = fMax * deltaT[i] * par->OutPar.Npf[i]
            * par->OutPar.CDF[i];
        if ( (par->OutPar.FalseAlarm[i] < 1.0) )
            par->OutPar.FalseAlarm[i] = 1.0;
    }
}
else
{
    for (i=2; i<=nStages; i++)
    {
        B = (par->OutPar.Npf[i]/par->OutPar.Npf[i-1] < 1.0) ? 1.0 :
            par->OutPar.Npf[i]/par->OutPar.Npf[i-1];
        C = (deltaT[i]/deltaT[i-1] < 1.0) ? 1.0 : deltaT[i]/deltaT[i-1];
/*   par->OutPar.CDF[i] = B * C * gsl_cdf_gaussian_Q(par->OutPar.Threshold[i]
        - 2*N[i], sqrt(2*N[i]));*/

        par->OutPar.CDF[i] = gsl_cdf_gaussian_Q(par->OutPar.Threshold[i]
            - 2*N[i],sqrt(2*N[i]));
        par->OutPar.FalseAlarm[i] = B * C * par->OutPar.FalseAlarm[i-1]
            * par->OutPar.CDF[i];
        if ( (par->OutPar.FalseAlarm[i] < 1.0) )
            par->OutPar.FalseAlarm[i] = 1.0;
    }
}

/* free memory */
free_vector(deltaT,1,nStages);
free_vector(muMax,1,nStages);
free_vector(N,1,nStages);

return;
}

/*****

/* function for No. of floating point operations in first stage from
eq.(5.4) of Brady-Creighton */

void Flops1(double deltaT1, double muMax1, double N1, params *par)
{
    double E, fMax, Npf1, Npc1;
    fMax = par->InPar.fMax;
    Npc1 = par->OutPar.Npc[1];
    Npf1 = par->OutPar.Npf[1];

    E = (Npf1/Npc1 > 1.0) ? Npf1/Npc1 : 1.0;
    par->OutPar.inCohCompt[1] = fMax* deltaT1*Npc1* E* (N1-1.0);
    par->OutPar.CohCompt[1] = fMax* deltaT1*Npc1*12.0*N1*log(2.0*fMax*deltaT1)
        /log(2.0);
    par->OutPar.Compt[1] = par->OutPar.inCohCompt[1] + par->OutPar.CohCompt[1];

    return;
}

/*****

/* function for computing power in any higher stage incoherent stage based on
eq.(5.5) of Brady Creighton */
/* Explanation: the value of the computing power for the ith stage depends
on the variables of the ith stage and also of the (i-1)th
stage. DeltaT1, muMax1, N1 and X1 represent the variables of the (i-1)th
stage while deltaT2, muMax2, N2 represent the vairables of the ith stage
(there is no dependence on X2) */

double Flops2(int k, double x[], params *par)
{
    double temp, A, B, D;
    int i;
    double *deltaT=NULL;
    double *muMax=NULL;
    double *N=NULL;

```

```

deltaT = vector(1,k);
muMax = vector(1,k);
N = vector(1,k);

for (i=1; i<=k; i++)
{
    deltaT[i] = x[3*(i-1)+1];
    muMax[i] = x[3*(i-1)+2];
    N[i] = x[3*(i-1)+3];
}

A = (par->OutPar.Npc[k]/par->OutPar.Npf[k-1]<1.0)? 1.0 :
    par->OutPar.Npc[k]/par->OutPar.Npf[k-1];

B = (deltaT[k]/deltaT[k-1] < 1.0)? 1.0: deltaT[k]/deltaT[k-1];

D = (par->OutPar.Npf[k]/par->OutPar.Npc[k] < 1.0) ? 1.0 :
    par->OutPar.Npf[k]/par->OutPar.Npc[k];

par->OutPar.inCohCompt[k] = par->OutPar.FalseAlarm[k-1]
    * A*B*(D*(N[k]-1.0));
par->OutPar.CohCompt[k] = par->OutPar.FalseAlarm[k-1]* A*B
    *(640.0*N[k]*deltaT[k]/par->InPar.Tsft);
temp = par->OutPar.inCohCompt[k] + par->OutPar.CohCompt[k];
/* temp = par->OutPar.FalseAlarm[k-1]* A*B*(640.0*N[k]
    *deltaT[k]/par->InPar.Tsft + D*(N[k]-1.0));*/

/* de-allocate memory */
free_vector(deltaT,1,k);
free_vector(muMax,1,k);
free_vector(N,1,k);

return temp;
}

/*****
/* Computing power for final coherent stage */ /* Uses formula for
$C^{coh}$ from Curt's notes */

double FlopsCoh(double x[], params *par)
{
    double Fn, fMax, Tcoh, temp, A;
    int nStages;

    /* Assuming that the n^{th} stage has the longest total observation time so
    that coherent observation time must be Tcoh = Nn*deltaTn. Note: this is
    only valid for the "BC mode", In "fresh mode" we should take a sum over
    the Ni*deltaTi */
    /* This should use a better value for muMax but for now use the arbitrary
    value 0.8 */

    nStages = par->InPar.nStages;
    A = (par->OutPar.NpfCoh/par->OutPar.Npf[nStages] <1.0)?
        1.0 : par->OutPar.NpfCoh/par->OutPar.Npf[nStages];

    Fn = par->OutPar.FalseAlarm[nStages];
    fMax = par->InPar.fMax;
    Tcoh = par->OutPar.Tcoh;
    temp = Fn*A*12.0*fMax*Tcoh*log( (2*fMax*Tcoh))/log(2.0);/* resampling method*/

    /*temp = exp(Fn)*A*(Tcoh/ x[3*(nStages-1)+1])*640.0*(Tcoh/T_sft); C(i)_sft*/

    return temp;
}
/*-----*/
/* Solution of the false dismissal constraint */

double Threshold(double deltaT, double mu, double N, double h, double
FalseDiss)
{
    double temp, erfcinv, mean, sigma;

    erfcinv = gsl_cdf_ugaussian_Qinv( FalseDiss )/sqrt(2.0);
    mean = 2.0*N + N*(1.0-mu/3.0)*h*deltaT/2.0; /* the mean */
    sigma = sqrt(2.0*N + N*(1.0-mu/3.0)*h*deltaT ); /* the std. dev. */

    temp = mean - sqrt(2.0)*erfcinv*sigma;

    if (temp < 0.0) temp=0.0;

    return temp;
}

/*-----

```



```

* Modules for calculating No. of patches in parameter space for
* directed and all sky Stack-Slide searches
*-----*/

/*-----*/

double Gs(int s, double N)
{
    double out;

    switch (s)
    {
        case 0: out = 1.0;
            break;
        case 1 : out = (PIE/(6.0*sqrt(5.0))) * sqrt(5.0*N*N-4.0);
            break;
        case 2 : out = (PIE*PIE/(180.0*sqrt(105.0))) * sqrt(175.0*pow(N,6.0)
            - 840.0*pow(N,4.0)
            + 1100.0*N*N -432.0);
            break;
        case 3 : out = (PIE*PIE*PIE/(75600.0*sqrt(105.0))*sqrt(3675.0*pow(N,12.0)
            -58800.0*pow(N,10.0)
            +363160.0*pow(N,8.0)
            -1053360.0*pow(N,6.0)
            +1484336.0*pow(N,4.0)
            -987840.0*N*N+248832.0);

            break;
    }

    return out;
}

/*-----*/

double Ns(int s, double deltaT, double muMax, double fMax, double
tauMin) {
    double out;

    out = 1.0;
    if (s > 0)
    {
        out = pow(fMax,s) * pow(deltaT, 0.5*s*(1.0*s+3.0))
        / ( pow(muMax/(1.0*s), 0.5*s) * pow(tauMin, 0.5*s*(1.0*s+1.0)));
    }

    return out;
}

/*-----*/

/* Calculation of number of patches in parameter space for Direct
search */ double NP(double deltaT, double muMax, double N, double
fMax, double tauMin) {
    double NS[4], G[4], Np, temp;
    int k;

    for (k=0; k<=3; k++)
    {
        NS[k] = Ns(k, deltaT, muMax, fMax, tauMin);
        G[k] = Gs(k, N);
    }

    Np = NS[0] * G[0];
    if (Np < NS[1] * G[1]) Np = NS[1] * G[1];
    if (Np < NS[2] * G[2]) Np = NS[2] * G[2];
    if (Np < NS[3] * G[3]) Np = NS[3] * G[3];

    temp=Np;
    if (Np<1.0) temp=1.0;
    return temp;
}

/*-----*/

/* Number of patches in parameter space for all-sky search:
eq.(2.22) of BC */

double NPallsky(double deltaT, double muMax,
double N, double fMax, double tauMin)
{
    double H[4], X[4], out1, out2, out, temp;
    int s, k;

    for (s=0; s<=3; s++) H[s] = 1.0;

    for (s = 0; s <= 3; s++)
    {

```

```

    for (k=0; k <= s; k++)
    {
        H[s] *= 1.0 + 0.3 * AU * pow(OMEGA,k+1) * pow(tauMin,k)
        / (LIGHT * factorial(k) * sqrt(MskyS(deltaT, muMax, N, s, fMax)));
    }
    X[s] = MskyS(deltaT, muMax, N, s, fMax) * Ns(s, deltaT, muMax, fMax, tauMin)
        * Gs(s, N) * H[s] * pow(1.0*s, 0.5*s) / (pow(1.0*s+2.0, 0.5*s));
}

/* output is largest element of X[s] */
out1 = (X[0] > X[1]) ? X[0] : X[1];
out2 = (X[2] > X[3]) ? X[2] : X[3];
out  = (out1 > out2) ? out1 : out2;

temp=out;
if (out<1.0) temp=1.0;
return temp;
}

/*-----*/

/* Number of sky-patches */

double MskyS(double deltaT, double muMax, double N, int S, double fMax)
{
    double A, B, C;
    double temp, Ms;

    /* notation from Brady-Creighton eq. (2.23) */
    A = 0.014;
    B = 0.046 * pow(deltaT/86400.0,2);
    C = 0.18 * pow(deltaT/86400.0,5) * pow(N,3);
    temp = sqrt(1.0/(A*A) + 1.0/(B*B) + 1.0/(C*C));

    Ms = 0.25 * (1.0*S+2.0) * fMax * fMax / (muMax * temp);

    return Ms;
}

/*-----
 * Routines for memory allocation taken mostly from numerical recipes
 *-----*/
/* allocation a vector with subscript range nl..nh */

double *vector(long nl, long nh)
{
    double *v;

    v = (double *)malloc((size_t) ((nh-nl+1)*sizeof(double)));
    if (!v) nrerror("allocation failure in vector()");
    return v-nl+1;
}

/*-----*/

/* allocation of a matrix with range(nrl..nrh,ncl..nch) */

double **matrix(long nrl, long nrh, long ncl, long nch)
{
    long i, nrow = nrh-nrl+1, ncol = nch-ncl+1;
    double **m;

    /* allocate pointers to rows */
    m = (double **)malloc((size_t)((nrow+1)*sizeof(double*)));
    if (!m) nrerror("allocation failure 1 in matrix()");
    m += 1;
    m -= nrl;

    /* allocate rows and set pointers to them */
    m[nrl] = (double *)malloc((size_t)((nrow*ncol+1)*sizeof(double)));
    if (!m[nrl]) nrerror("allocation failure 2 in matrix()");
    m[nrl] += 1;
    m[nrl] -= ncl;

    for (i=nrl+1; i<=nrh; i++)
        m[i] = m[i-1]+ncol;

    return m;
}

/*-----*/

/* free memory allocated to vector */

void free_vector(double *v, long nl, long nh)

```

```

{
    free((char*) (v+n1-1));
    return;
}

/*-----*/

/* free memory allocated to matrix */

void free_matrix(double **m, long nrl, long nrh, long ncl, long nch)
{
    free((char*) (m[nrl]+ncl-1));
    free((char*) (m+nrl-1));
}

/*-----*/

/* error messages */ void nrerror(char error_text[]) {
    fprintf(stderr, "Numerical Recipes run-time error...\n");
    fprintf(stderr, "%s\n", error_text);
    fprintf(stderr, "...now exiting to system...\n");
    exit(1);
    return;
}

/*-----*/

/* factorial function (to be used only for small integers) */
int factorial(int N)
{
    int k, out;

    out = 1;

    if (N > 0)
    {
        for (k = 1; k <= N; k++) out *= k;
    }

    return out;
}

/*-----*/

/* Log(N!) by Stirling's approximation */
double stirling(double N)
{
    double x;
    x = 0.5*log(2.0*PIE) + (N+0.5)*log(N) - N;
    return x;
}

/*-----*/

/* Random generator function ran1*/

double ran1(long *idum)
/*Minimal random number generator of Park and Miller with Bays-Durham
shuffle and added safeguards. Returns a uniform random deviate between
0.0 and 1.0 (exclusive of the endpoint values). Call with idum a negative
integer to initialize; thereafter, do not alter idum between successive
deviates in a sequence. RNMN should approximate the largest floating
value that is less than 1.*/

{
    int j;
    long k;
    static long iy=0;
    static long iv[NTAB];
    double temp;

    if (*idum <= 0 || !iy)
    {
        /*Initialize.*/

        if (-(*idum) < 1) *idum=1; /*Be sure to prevent idum =0 */
        else *idum = -(*idum);

        for (j=NTAB+7;j>=0;j--)
        {
            /*Load the shuffle table (after 8 warm-ups).*/
            k=(*idum)/IQ;
            *idum=IA*(*idum-k*IQ)-IR*k;
            if (*idum < 0) *idum += IM;
            if (j < NTAB) iv[j] = *idum;
        }
        iy=iv[0];
    }

    k=(*idum)/IQ; /*Start here when not initializing.*/

```

```

*idum=IA*(idum-k*IQ)-IR*k; /*Compute idum=(IA*idum) % IM without*/
/*overflows by Schrage's method.*/

if (*idum < 0) *idum += IM;
j=iy/NDIV; /* Will be in the range 0..NTAB-1*/

iy=iv[j]; /*Output previously stored value and refill the shuffle table.*/

iv[j] = *idum;
if ((temp=AM*iy) > RNMx)
    return RNMx; /*Because users don't expect endpoint values.*/

else
    return temp;
}

/* Random generator function ran2*/

double ran2(long idum)

/*Long period (> 2 ?? 10^18) random number generator of L'Ecuyer
with Bays-Durham shuffle and added safeguards. Returns a uniform
random deviate between 0.0 and 1.0 (exclusive of the endpoint values).
Call with idum a negative integer to initialize; thereafter, do not
alter idum between successive deviates in a sequence. RNMx should
approximate the largest floating value that is less than 1. */

{
int j;
long k;
static long idum2=123456789;
static long iy=0;
static long iv[NTAB];
double temp;

if (*idum <= 0)
{
/*Initialize.*/
if (-(*idum) < 1) *idum=1; /*Be sure to prevent idum = 0.*/
else *idum = -(*idum);
idum2=(*idum);

for (j=NTAB+7;j>=0;j--)
{
/*Load the shuffle table (after 8 warm-ups).*/
k=(*idum)/IQ1;
*idum=IA1*(idum-k*IQ1)-k*IR1;
if (*idum < 0) *idum += IM1;
if (j < NTAB) iv[j] = *idum;
}
iy=iv[0];
}

k=(*idum)/IQ1; /*Start here when not initializing.*/
*idum=IA1*(idum-k*IQ1)-k*IR1; /*Compute idum=(IA1*idum) % IM1 without
overflows by Schrage's method.*/

if (*idum < 0) *idum += IM1;
k=idum2/IQ2;
idum2=IA2*(idum2-k*IQ2)-k*IR2; /*Compute idum2=(IA2*idum) % IM2 likewise.*/
if (idum2 < 0) idum2 += IM2;
j=iy/NDIV; /*Will be in the range 0..NTAB-1.*/
iy=iv[j]-idum2; /*Here idum is shuffled, idum and idum2 are
combined to generate output.*/

iv[j] = *idum;
if (iy < 1) iy += IMM1;
if ((temp=AM*iy) > RNMx)
    return RNMx; /*Because users don't expect endpoint values.*/

else
    return temp;
}

```

## B.4 Ameba

```

/*-----
* This module contains the Amebsa routines for minization (Simulated
* Annealing Method) and is taken from Numerical recipes
*-----*/

#include "Header.h"

/*-----*/

```

```

/* Function for performing the minimization */

void minimize(double **p, double values[], int nDim, double pb[], double *yb,
             double fTol, double (*funkt)(double [], params *),
             int *iter, double temptr, params *par)

/*Multidimensional minimization of the function funk(x) where
x[1..ndim] is a vector in ndim dimensions,by simulated
annealing combined with the downhill simplex method of Nelder
and Mead.The input matrix p[1..ndim+1][1..ndim] has ndim+1
rows,each an ndim dimensional vector which is a vertex of the
starting simplex.Also input are the following:the vector
y[1..ndim+1],whose components must be pre-initialized to the
values of funk eval- uated at the ndim+1 vertices (rows)of p
ftol the fractional convergence tolerance to be achieved in the
function value for an early return;iter and temptr The routine
makes iter function evaluations at an annealing temperature
temptr then returns.You should then decrease temptr according
to your annealing schedule,reset iter and call the routine
again (leaving other arguments unaltered between calls).If iter
is returned with a positive value, then early convergence and
return occurred. If you initialize yb to a very large value on
the rst call,then yb and pb[1..ndim] will subsequently return
the best function value and point ever encountered (even if it
is no longer a point in the simplex).*/

{
int i, ihi, ilo, j, n, mpts=nDim+1;
double rtol, sum, swap, yhi, ylo, ynhi, ysave, yt, ytry, *psum;

psum = vector(1,nDim);
tt = -temptr;
GET_PSUM
for (;;)
{
ilo=1;          /*Determine which point is the highest (worst), */
ihi=2;          /*next-highest, and lowest (best).*/
ynhi=ylo=values[1]+tt*log(ran2(&idum)); /*Whenever we look at */
yhi=values[2]+tt*log(ran2(&idum)); /* a vertex, it gets a random*/
if (ylo > yhi) /*thermal fluctuation.*/
{
ihi=1;
ilo=2;
ynhi=yhi;
yhi=ylo;
ylo=ynhi;
}

for (i=3;i<=mpts;i++)
{
/*Loop over the points in the simplex.*/
yt=values[i]+tt*log(ran2(&idum)); /*More thermal fluctuations.*/
if (yt <= ylo)
{
ilo=i;
ylo=yt;
}
if (yt > yhi)
{
ynhi=yhi;
ihi=i;
yhi=yt;
}
else if (yt > ynhi)
{
ynhi=yt;
}
}
rtol=2.0*fabs(yhi-ylo)/(fabs(yhi)+fabs(ylo));
/*Compute the fractional range from highest to lowest and return if
satisfactory.*/
if (rtol < fTol || *iter < 0)
{
/*If returning, put best point and value in slot 1.*/
swap=values[1];
values[1]=values[ilo];
values[ilo]=swap;
for (n=1;n<=nDim;n++)
{
swap=p[1][n];
p[1][n]=p[ilo][n];
p[ilo][n]=swap;
}
break;
}
*iter -= 2;
/*Begin a new iteration.First extrapolate by a factor -1 through the
face of the simplex across from the high point,i.e.,relect the
simplex from the high point.*/

```

```

ytry=amotsa(p,values,psum,nDim,pb,yb,funk,ihl,&yhi,-1.0, par);
if (ytry <= ylo)
{
  /*Gives a result better than the best point,so try an additional
  extrapolation by a factor of 2*/
  ytry=amotsa(p,values,psum,nDim,pb,yb,funk,ihl,&yhi,2.0, par);
}
else if (ytry >= ynhi)
{
  /*The reflected point is worse than the second-highest,so look for
  an intermediate lower point, i.e., do a one-dimensional
  contraction.*/
  ysave=yhi;
  ytry=amotsa(p,values,psum,nDim,pb,yb,funk,ihl,&yhi,0.5, par);
  if (ytry >= ysave)
  {
    /*Can't seem to get rid of that high point.*/
    for (i=1;i<=mpts;i++)
    {
      /*Better contract around the lowest (best) point.*/
      if (i != ilo)
      {
        for (j=1;j<=nDim;j++)
        {
          psum[j]=0.5*(p[i][j]+p[ilo][j]);
          p[i][j]=psum[j];
        }
        values[i]=(*funk)(psum, par);
      }
    }
    *iter -= nDim;
    GET_PSUM          /*Recompute psum*/
  }
}
else ++(*iter);          /*Correct the evaluation count.*/
}
free_vector(psum,1,nDim);
}

double amotsa(double **p, double values[], double psum[], int nDim,
double pb[], double *yb,double (*funk)(double [], params *),
int ihl, double *yhi, double fac, params *par)
/*Extrapolates by a factor fac through the face of the simplex across
from the high point, tries it, and replaces the high point if the new
point is better. */
{
int j;
double fac1, fac2, yflu, ytry, *ptry;

ptry=vector(1,nDim);
fac1=(1.0-fac)/nDim;
fac2=fac1-fac;
for (j=1;j<=nDim;j++)
{
  ptry[j]=psum[j]*fac1-p[ihl][j]*fac2;

  /****** Tries to get rid of the negative values of vertices *****/

  switch(j%3)
  {
case(1):
  if (ptry[j]>=0.01*86400)
    qtry[j]=ptry[j];
  else
    ptry[j] = qtry[j];
  break;
case(2):
  if (ptry[j]>=0.001 && ptry[j]<0.90)
    qtry[j]=ptry[j];
  else
    ptry[j] = qtry[j];
  break;
case(0):
  if (ptry[j]>=5.0)
    qtry[j]=ptry[j];
  else
    ptry[j] = qtry[j];
  break;
}
}
/******
ytry=(*funk)(ptry, par);
if (ytry <= *yb)
{
  /*Save the best-ever.*/
  for (j=1;j<=nDim;j++) pb[j]=ptry[j];
  *yb=ytry;
}
yflu=ytry-tt*log(ran2(&idum));/*We added a thermal fluctuation to all */
if (yflu < *yhi)/* the current vertices, but we subtract it here, so as*/
{
  /*to give the simplex a thermal Brownian motion: */

```

```
    values[ihi]=ytry; /*It likes to accept any suggested change. */
    *yhi=yflu;
    for (j=1;j<=nDim;j++)
    {
        psum[j] += ptry[j]-p[ihi][j];
        p[ihi][j]=ptry[j];
    }
    free_vector(ptry,1,nDim);
    return yflu;
}
```





## Appendix C

# The CVS versions of ComputeFStatistic\_v2 code used for the analysis done in chapter 5

ComputeFStatistic\\_v2:

```
LALRCSID.h,v 1.6 2007/06/08 14:41:52 bema
LALAtomicDatatypes.h,v 1.9 2007/06/08 14:41:52 bema
LALDatatypes.h,v 1.19 2007/06/08 14:41:52 bema
AVFactoriesH.m4,v 1.2 2004/11/04 22:48:56 jolien
LALStdio.h,v 1.2 2007/06/08 14:41:53 bema
LALMalloc.h,v 1.4 2007/06/08 14:41:52 bema
XLALError.h,v 1.12 2007/06/08 14:41:53 bema
LALError.h,v 1.3 2007/06/08 14:41:52 bema
LALStatusMacros.h,v 1.25 2007/06/08 14:41:52 bema
LALStdlib.h,v 1.12 2007/06/08 14:41:53 bema
LALConstants.h,v 1.12 2007/06/08 14:41:52 bema
LALDetectors.h,v 1.8 2007/08/13 16:16:59 whelan
DetectorSite.h,v 1.14 2007/08/13 16:16:59 whelan
LALBarycenter.h,v 1.12 2007/06/08 14:41:50 bema
LALInitBarycenter.h,v 1.4 2007/07/17 13:34:03 bema
StringInput.h,v 1.4 2007/06/08 14:41:53 bema
ConfigFile.h,v 1.11 2005/07/12 13:12:19 reinhard
SFTutils.h,v 1.31 2007/02/15 12:25:03 reinhard
Date.h,v 1.65 2007/06/08 14:41:43 bema
SkyCoordinates.h,v 1.19 2007/06/08 14:41:47 bema
RngMedBias.h,v 1.3 2007/06/08 14:41:59 bema
LALRunningMedian.h,v 1.8 2007/06/08 14:41:59 bema
UserInput.h,v 1.21 2007/02/22 16:32:27 reinhard
SeqFactoriesH.m4,v 1.3 2004/11/10 20:41:21 jolien
PulsarDataTypes.h,v 1.18 2006/11/22 20:30:27 reinhard
SFTfileIO.h,v 1.34 2006/12/13 16:56:24 bema
ExtrapolatePulsarSpins.h,v 1.9 2006/10/30 18:27:34 reinhard
NormalizeSFTRngMed.h,v 1.16 2007/08/21 00:39:15 reinhard
ComputeFstat.h,v 1.35 2006/12/06 15:14:19 reinhard
DetResponse.h,v 1.11 2007/06/08 14:41:56 bema
DetectorStates.h,v 1.4 2006/10/30 10:09:40 reinhard
LUT.h,v 1.11 2007/02/26 13:16:26 badri
PHMD.h,v 1.9 2006/03/14 10:48:40 badri
HoughMap.h,v 1.9 2006/03/14 10:48:40 badri
LALHough.h,v 1.11 2006/02/20 17:28:02 badri
LogPrintf.h,v 1.4 2007/02/01 14:37:04 reinhard
lalapps.h,v 1.2 2007/06/08 15:29:50 bema
PtoleMetric.h,v 1.19 2007/06/08 14:41:50 bema
PulsarTimes.h,v 1.11 2007/06/08 14:41:50 bema
StackMetric.h,v 1.11 2007/06/08 14:41:50 bema
```

DopplerScan.h,v 1.54 2007/09/01 19:04:17 reinhard  
 DopplerFullScan.h,v 1.4 2007/02/01 21:57:05 reinhard  
 HeapToplist.h,v 1.15 2007/06/08 15:29:51 bema  
 ComputeFStatistic\_v2.c,v 1.248 2007/09/17 20:31:16 reinhard  
 LALRCSID.h,v 1.6 2007/06/08 14:41:52 bema  
 HeapToplist.h,v 1.15 2007/06/08 15:29:51 bema  
 HeapToplist.c,v 1.20 2007/06/08 15:29:51 bema  
 LALRCSID.h,v 1.6 2007/06/08 14:41:52 bema  
 LALAtomicDatatypes.h,v 1.9 2007/06/08 14:41:52 bema  
 LALDatatypes.h,v 1.19 2007/06/08 14:41:52 bema  
 LALMalloc.h,v 1.4 2007/06/08 14:41:52 bema  
 XLALError.h,v 1.12 2007/06/08 14:41:53 bema  
 LALError.h,v 1.3 2007/06/08 14:41:52 bema  
 LALStatusMacros.h,v 1.25 2007/06/08 14:41:52 bema  
 LALStdlib.h,v 1.12 2007/06/08 14:41:53 bema  
 LALDetectors.h,v 1.8 2007/08/13 16:16:59 whelan  
 DetectorSite.h,v 1.14 2007/08/13 16:16:59 whelan  
 LALStdio.h,v 1.2 2007/06/08 14:41:53 bema  
 LALConstants.h,v 1.12 2007/06/08 14:41:52 bema  
 LALBarycenter.h,v 1.12 2007/06/08 14:41:50 bema  
 PulsarTimes.h,v 1.11 2007/06/08 14:41:50 bema  
 StackMetric.h,v 1.11 2007/06/08 14:41:50 bema  
 AVFactoriesH.m4,v 1.2 2004/11/04 22:48:56 jolien  
 StringInput.h,v 1.4 2007/06/08 14:41:53 bema  
 ConfigFile.h,v 1.11 2005/07/12 13:12:19 reinhard  
 Date.h,v 1.65 2007/06/08 14:41:43 bema  
 LALInitBarycenter.h,v 1.4 2007/07/17 13:34:03 bema  
 Velocity.h,v 1.10 2006/02/11 19:29:47 badri  
 TwoDMesh.h,v 1.8 2007/06/08 14:41:50 bema  
 LogPrintf.h,v 1.4 2007/02/01 14:37:04 reinhard  
 SkyCoordinates.h,v 1.19 2007/06/08 14:41:47 bema  
 PtoleMetric.h,v 1.19 2007/06/08 14:41:50 bema  
 SFTutils.h,v 1.31 2007/02/15 12:25:03 reinhard  
 RngMedBias.h,v 1.3 2007/06/08 14:41:59 bema  
 LALRunningMedian.h,v 1.8 2007/06/08 14:41:59 bema  
 PulsarDataTypes.h,v 1.18 2006/11/22 20:30:27 reinhard  
 ComputeFstat.h,v 1.35 2006/12/06 15:14:19 reinhard  
 DetResponse.h,v 1.11 2007/06/08 14:41:56 bema  
 DetectorStates.h,v 1.4 2006/10/30 10:09:40 reinhard  
 DopplerScan.h,v 1.54 2007/09/01 19:04:17 reinhard  
 DopplerScan.c,v 1.110 2007/09/02 22:42:38 reinhard  
 LALRCSID.h,v 1.6 2007/06/08 14:41:52 bema  
 LALAtomicDatatypes.h,v 1.9 2007/06/08 14:41:52 bema  
 LALDatatypes.h,v 1.19 2007/06/08 14:41:52 bema  
 LALMalloc.h,v 1.4 2007/06/08 14:41:52 bema  
 XLALError.h,v 1.12 2007/06/08 14:41:53 bema  
 LALError.h,v 1.3 2007/06/08 14:41:52 bema  
 LALStatusMacros.h,v 1.25 2007/06/08 14:41:52 bema  
 LALStdlib.h,v 1.12 2007/06/08 14:41:53 bema  
 LALDetectors.h,v 1.8 2007/08/13 16:16:59 whelan  
 DetectorSite.h,v 1.14 2007/08/13 16:16:59 whelan  
 AVFactoriesH.m4,v 1.2 2004/11/04 22:48:56 jolien  
 LALConstants.h,v 1.12 2007/06/08 14:41:52 bema  
 SkyCoordinates.h,v 1.19 2007/06/08 14:41:47 bema  
 LALStdio.h,v 1.2 2007/06/08 14:41:53 bema  
 LALBarycenter.h,v 1.12 2007/06/08 14:41:50 bema  
 PtoleMetric.h,v 1.19 2007/06/08 14:41:50 bema  
 LatticeCovering.h,v 1.5 2007/01/31 00:18:41 reinhard  
 LogPrintf.h,v 1.4 2007/02/01 14:37:04 reinhard  
 StringInput.h,v 1.4 2007/06/08 14:41:53 bema  
 ConfigFile.h,v 1.11 2005/07/12 13:12:19 reinhard  
 DetectorStates.h,v 1.4 2006/10/30 10:09:40 reinhard  
 Date.h,v 1.65 2007/06/08 14:41:43 bema  
 DetResponse.h,v 1.11 2007/06/08 14:41:56 bema

SFTutils.h,v 1.31 2007/02/15 12:25:03 reinhard  
 RngMedBias.h,v 1.3 2007/06/08 14:41:59 bema  
 LALRunningMedian.h,v 1.8 2007/06/08 14:41:59 bema  
 PulsarDataTypes.h,v 1.18 2006/11/22 20:30:27 reinhard  
 FlatPulsarMetric.h,v 1.5 2007/01/31 00:32:26 reinhard  
 PulsarTimes.h,v 1.11 2007/06/08 14:41:50 bema  
 StackMetric.h,v 1.11 2007/06/08 14:41:50 bema  
 ComputeFstat.h,v 1.35 2006/12/06 15:14:19 reinhard  
 DopplerScan.h,v 1.54 2007/09/01 19:04:17 reinhard  
 DopplerFullScan.h,v 1.4 2007/02/01 21:57:05 reinhard  
 DopplerLatticeCovering.h,v 1.3 2007/02/09 11:57:09 reinhard  
 DopplerFullScan.c,v 1.14 2007/02/09 11:56:59 reinhard  
 LALRCSID.h,v 1.6 2007/06/08 14:41:52 bema  
 DetectorStates.h,v 1.4 2006/10/30 10:09:40 reinhard  
 LALAtomicDatatypes.h,v 1.9 2007/06/08 14:41:52 bema  
 LALDatatypes.h,v 1.19 2007/06/08 14:41:52 bema  
 LALMalloc.h,v 1.4 2007/06/08 14:41:52 bema  
 XLALError.h,v 1.12 2007/06/08 14:41:53 bema  
 LALError.h,v 1.3 2007/06/08 14:41:52 bema  
 LALStatusMacros.h,v 1.25 2007/06/08 14:41:52 bema  
 LALStdlib.h,v 1.12 2007/06/08 14:41:53 bema  
 LALStdio.h,v 1.2 2007/06/08 14:41:53 bema  
 LALConstants.h,v 1.12 2007/06/08 14:41:52 bema  
 LALDetectors.h,v 1.8 2007/08/13 16:16:59 whelan  
 DetectorSite.h,v 1.14 2007/08/13 16:16:59 whelan  
 SkyCoordinates.h,v 1.19 2007/06/08 14:41:47 bema  
 Date.h,v 1.65 2007/06/08 14:41:43 bema  
 DetResponse.h,v 1.11 2007/06/08 14:41:56 bema  
 LALBarycenter.h,v 1.12 2007/06/08 14:41:50 bema  
 SFTutils.h,v 1.31 2007/02/15 12:25:03 reinhard  
 RngMedBias.h,v 1.3 2007/06/08 14:41:59 bema  
 LALRunningMedian.h,v 1.8 2007/06/08 14:41:59 bema  
 PulsarDataTypes.h,v 1.18 2006/11/22 20:30:27 reinhard  
 PulsarTimes.h,v 1.11 2007/06/08 14:41:50 bema  
 AVFactoriesH.m4,v 1.2 2004/11/04 22:48:56 jolien  
 PtoleMetric.h,v 1.19 2007/06/08 14:41:50 bema  
 FlatPulsarMetric.h,v 1.5 2007/01/31 00:32:26 reinhard  
 FlatPulsarMetric.c,v 1.15 2007/05/06 12:38:46 reinhard  
 LALRCSID.h,v 1.6 2007/06/08 14:41:52 bema  
 LALAtomicDatatypes.h,v 1.9 2007/06/08 14:41:52 bema  
 LALDatatypes.h,v 1.19 2007/06/08 14:41:52 bema  
 LALMalloc.h,v 1.4 2007/06/08 14:41:52 bema  
 XLALError.h,v 1.12 2007/06/08 14:41:53 bema  
 LALError.h,v 1.3 2007/06/08 14:41:52 bema  
 LALStatusMacros.h,v 1.25 2007/06/08 14:41:52 bema  
 LALStdlib.h,v 1.12 2007/06/08 14:41:53 bema  
 LALDetectors.h,v 1.8 2007/08/13 16:16:59 whelan  
 DetectorSite.h,v 1.14 2007/08/13 16:16:59 whelan  
 AVFactoriesH.m4,v 1.2 2004/11/04 22:48:56 jolien  
 LALConstants.h,v 1.12 2007/06/08 14:41:52 bema  
 SkyCoordinates.h,v 1.19 2007/06/08 14:41:47 bema  
 LALStdio.h,v 1.2 2007/06/08 14:41:53 bema  
 LALBarycenter.h,v 1.12 2007/06/08 14:41:50 bema  
 PtoleMetric.h,v 1.19 2007/06/08 14:41:50 bema  
 LatticeCovering.h,v 1.5 2007/01/31 00:18:41 reinhard  
 LogPrintf.h,v 1.4 2007/02/01 14:37:04 reinhard  
 DetectorStates.h,v 1.4 2006/10/30 10:09:40 reinhard  
 Date.h,v 1.65 2007/06/08 14:41:43 bema  
 DetResponse.h,v 1.11 2007/06/08 14:41:56 bema  
 SFTutils.h,v 1.31 2007/02/15 12:25:03 reinhard  
 RngMedBias.h,v 1.3 2007/06/08 14:41:59 bema  
 LALRunningMedian.h,v 1.8 2007/06/08 14:41:59 bema  
 PulsarDataTypes.h,v 1.18 2006/11/22 20:30:27 reinhard  
 FlatPulsarMetric.h,v 1.5 2007/01/31 00:32:26 reinhard

PulsarTimes.h,v 1.11 2007/06/08 14:41:50 bema  
StackMetric.h,v 1.11 2007/06/08 14:41:50 bema  
ComputeFstat.h,v 1.35 2006/12/06 15:14:19 reinhard  
DopplerScan.h,v 1.54 2007/09/01 19:04:17 reinhard  
DopplerFullScan.h,v 1.4 2007/02/01 21:57:05 reinhard  
DopplerLatticeCovering.h,v 1.3 2007/02/09 11:57:09 reinhard  
DopplerLatticeCovering.c,v 1.9 2007/08/20 22:52:43 reinhard  
LALRCSID.h,v 1.6 2007/06/08 14:41:52 bema  
LALAtomicDatatypes.h,v 1.9 2007/06/08 14:41:52 bema  
LALDatatypes.h,v 1.19 2007/06/08 14:41:52 bema  
lalapps.h,v 1.2 2007/06/08 15:29:50 bema  
LALMalloc.h,v 1.4 2007/06/08 14:41:52 bema  
XLALError.h,v 1.12 2007/06/08 14:41:53 bema  
LALError.h,v 1.3 2007/06/08 14:41:52 bema  
LALStatusMacros.h,v 1.25 2007/06/08 14:41:52 bema  
LALInitBarycenter.c,v 1.10 2007/08/15 17:11:30 bema  
SFTfileIO.c,v 1.126 2007/08/27 14:13:17 bema  
UserInput.c,v 1.39 2006/07/10 09:38:25 reinhard  
ConfigFile.c,v 1.25 2007/09/08 19:04:24 reinhard  
FileIO.c,v 1.19 2007/08/22 16:17:19 bema  
StreamSequenceInput.m4,v 1.4 2005/02/18 03:09:47 jolien  
StringToken.c,v 1.2 2007/06/08 14:41:53 bema  
VectorFactories.m4,v 1.5 2003/09/29 00:33:24 dwchin  
NormalizeSFTRngMed.c,v 1.34 2007/09/17 12:47:31 reinhard  
RngMedBias.c,v 1.4 2007/06/08 14:41:59 bema  
SkyCoordinates.c,v 1.12 2007/06/08 14:41:47 bema  
GPStoFloat.c,v 1.11 2007/06/08 14:41:43 bema  
GPStoUTC.c,v 1.31 2007/06/08 14:41:43 bema  
IncrementGPS.c,v 1.16 2007/06/08 14:41:43 bema  
ComputeFstat.c,v 1.70 2007/09/17 12:47:31 reinhard  
ProjectMetric.c,v 1.5 2007/06/08 14:41:51 bema  
DetectorStates.c,v 1.8 2006/10/30 10:09:44 reinhard  
PtoleMetric.c,v 1.31 2007/06/08 14:41:51 bema  
SFTutils.c,v 1.52 2007/08/21 00:25:50 reinhard  
ExtrapolatePulsarSpins.c,v 1.10 2006/10/23 18:44:47 reinhard  
TwoDMesh.c,v 1.9 2007/06/08 14:41:51 bema  
TBaryPtolemaic.c,v 1.6 2007/06/08 14:41:51 bema  
DTEphemeris.c,v 1.7 2007/06/08 14:41:51 bema  
CoherentMetric.c,v 1.8 2007/06/08 14:41:51 bema  
TComp.c,v 1.3 2007/06/08 14:41:51 bema  
LALBarycenter.c,v 1.10 2007/07/18 16:32:02 bema  
LatticeCovering.c,v 1.4 2007/01/31 00:18:48 reinhard  
GetEarthTimes.c,v 1.4 2007/06/08 14:41:51 bema  
TwoDMeshInternal.c,v 1.12 2007/06/08 14:41:51 bema  
TSpin.c,v 1.4 2007/06/08 14:41:51 bema  
LALGSL.c,v 1.3 2007/06/08 14:41:53 bema  
CreateDetector.c,v 1.21 2007/08/13 16:16:59 whelan  
VectorSequenceFactories.m4,v 1.4 2001/08/21 04:19:49 jolien  
LALRunningMedian.c,v 1.13 2007/06/08 14:41:59 bema  
TerrestrialCoordinates.c,v 1.18 2007/06/08 14:41:47 bema  
CelestialCoordinates.c,v 1.9 2007/06/08 14:41:47 bema  
LMST1.c,v 1.26 2007/06/08 14:41:43 bema  
HeapSort.c,v 1.11 2007/06/08 14:41:59 bema

# Bibliography

- [1] The web address for Einstein@Home is <http://www.einsteinathome.org/about/index.html>
- [2] The web address for Seti@Home is <http://setiathome.berkeley.edu/>
- [3] A. Einstein, *Die Feldgleichungen der Gravitation (The Field Equations of Gravitation)*, Sitzungsberichte der Preussischen Akademie der Wissenschaften zu Berlin, November 25, 1915, 844-847.
- [4] A. Einstein. *The Foundation of the General Theory of Relativity*, Annalen der Physik, **49**, 769 (1916).
- [5] F.W. Dyson, A.S. Eddington, and C.R. Davidson, *A Determination of the Deflection of Light by the Sun's Gravitational Field, from Observations Made at the Total Eclipse of May 29, 1919*, Mem. R. Astron. Soc., **220**, 291-333 (1920).
- [6] L.D. Landau and E.M. Lifshitz *The Classical Theory of fields*, Pergamon Press, 1959.
- [7] Bernard F. Schutz, *A First Course in General Relativity*, Cambridge University Press, 1990.
- [8] C.W. Misner, K.S. Thorne and J.A. Wheeler (MTW), *Gravitation*, W.H. Freeman and Company, 1973.
- [9] Bernard F. Schutz, *Gravitational Wave Astronomy*, CQG, **16**, A131 (1999). Or in a compact version *Gravitational Radiation*, gr-qc/0003069.
- [10] L. Ju, D.G. Blair and C. Zhaoi, *Detection of Gravitational Waves*, Rep. Prog. Phys., **63**, 1317 (2000).
- [11] Curt Cutler and Kip S. Thorne, *An Overview of Gravitational-wave Sources*, gr-qc/0204090.
- [12] Brnard F. Schutz, *Gravitational wave sources and their detectibility*, CQG, **6**, 1761 (1989).
- [13] Kip S. Thorne, *Gravitational Waves, A Web-Based Course*, <http://elmer.tapir.caltech.edu/ph237/> .
- [14] Curt Cutler and Éanna E. Flanagan, Phys. Rev. D49, 2658 (1994).
- [15] Kip, S. Thorne, 300 Years of Gravitation (in Gravitational Radiation), Edited by: Stephen Hawking and Werner Israel. Cambridge University Press, Cambridge, 1987.
- [16] Bruce Allen, *The stochastic gravity-wave background: sources and detection*, gr-qc/9604033.
- [17] Brnard F. Schutz, *Gravitational wave sources*, CQG, **13**, A216 (1996).

- [18] Joseph Weber, *Phys. Rev.* **117**, 306 (1960).
- [19] Joseph Weber, *Phys. Rev. Lett.* **22**, 1320 (1969).
- [20] Francesco Ronga, *Detection of gravitational waves with resonant antennas*, J. Phy.: Conference Series, **39**, 18 (2006).
- [21] Warren W. Johnson and Stephen M. Merkowitz, *Truncated icosahedral gravitational wave antenna*, *Phys. Rev. Lett.* **70**, 2367 (1993).
- [22] W.O. Hamiltona *et al.*, *Resonant detectors and interferometers can work together*, <http://sam.phys.lsu.edu/Papers/hawaiiypeset.pdf> .
- [23] L. Baggio, *Bar detectors: present and future*, [http://www.auriga.inl.infn.it/auriga/papers\\_src/tama99.ps.gz](http://www.auriga.inl.infn.it/auriga/papers_src/tama99.ps.gz) .
- [24] P. Astone *et al.*, *All-sky upper limit for gravitational radiation from spinning neutron stars*, *Class. Quantum Grav.* **20** S665 (2003).
- [25] <http://sam.phys.lsu.edu/>
- [26] <http://www.inl.infn.it/auriga/>
- [27] A.J. Weinstein, *LIGO Technical Report No. LIGO-G000165-00-R* (2000).
- [28] S. Sato *et al.*, *High-gain power recycling of a Fabry-Perot Michelson interferometer for a gravitational-wave antenna*, *Applied Optics*, **39**, No. 25, 4616 (2000).
- [29] G. Losurdo *et al.*, *Inertial control of the mirror suspensions of the VIRGO interferometer for gravitational wave detection*, *Review of Scientific Instruments*, **72**, No. 9, 3653 (2001).
- [30] LSC, *LIGO: The Laser Interferometer Gravitational-Wave Observatory*, LIGO-P070082-01 (2007) (to be appeared in Science).
- [31] <http://www.geo600.uni-hannover.de/results.html> .
- [32] [http://www.ligo.caltech.edu/jzweizig/distribution/LSC\\_Data/](http://www.ligo.caltech.edu/jzweizig/distribution/LSC_Data/) .
- [33] Status of the Virgo Ddetector,  
[http://wwwcascina.virgo.infn.it/vsb/slides/Vajente\\_GWDAW11\\_StatusOfVirgo.ppt](http://wwwcascina.virgo.infn.it/vsb/slides/Vajente_GWDAW11_StatusOfVirgo.ppt)
- [34] [www2.cnrs.fr/sites/en/fichier/dossier\\_de\\_presse\\_en\\_anglais.pdf](http://www2.cnrs.fr/sites/en/fichier/dossier_de_presse_en_anglais.pdf) . Or in HTML format at [http://www.milde-marketing.de/html/virgo\\_2007\\_-\\_dossier\\_de\\_presse.html](http://www.milde-marketing.de/html/virgo_2007_-_dossier_de_presse.html) .
- [35] *LSC White Paper*, LIGO-T990080-00.
- [36] P. Fritschel, R. Adhikari and R. Weissd, *Enhancements to the LIGO S5 Detectors*, LIGO-T050252-00-I, November 2005.
- [37] [http://www.esa.int/esaSC/120376\\_index\\_0\\_m.html](http://www.esa.int/esaSC/120376_index_0_m.html) .

- [38] [http://lisa.jpl.nasa.gov/TECHNOLOGY/LISA\\_interfer2.html](http://lisa.jpl.nasa.gov/TECHNOLOGY/LISA_interfer2.html) .
- [39] Thomas A. Prince, Massimo Tinto, and Shane L. Larson, *LISA Optimal Sensitivity*, *Phys. Rev. D* **66**, 122002 (2002).
- [40] Jim Hough and Sheila Rowan, *Gravitational Wave Detection by Interferometry (Ground and Space)*, *Living Rev. Relativity* **3**, 3 (2000). URL (cited on 11.07.2007): <http://www.livingreviews.org/lrr-2000-3>
- [41] S. Rowan, J. Hough and D.R.M. Crooks, *Thermal noise and material issues for gravitational wave detectors*, *Phys. Lett. A*, **347**, 25 (2005).
- [42] F. Beauville *et al.*, *Improvement in the shot noise of a laser interferometer gravitational wave detector by means of an output mode-cleaner*, *CQG*, **23**, 3235 (2006).
- [43] Norna A. Robertson, *Laser interferometric gravitational wave detectors*, *Class., Quantum Grav.*, **17**, R19 (2000).
- [44] P. Jaranowski, A. Królak, and B.F. Schutz (JKS), *Phys. Rev.* **D58**, 063001 (1998).
- [45] Kip S. Thorne, *Ph 136: Applications of Classical Physics, Chapter 5: Random Processes*, <http://www.pma.caltech.edu/Courses/ph136/yr2006/text.html> .
- [46] Boaz Porat, *A Course in Digital Signal Processing*, Section 2.6.3, John Wiley & Sons, 1997.
- [47] Boaz Porat, *A Course in Digital Signal Processing*, Section 2.9, John Wiley & Sons, 1997.
- [48] P. Jaranowski, A. Królak, *Living Rev. Relativity* **8**, 3, (2005).
- [49] M.H.A. Davis, “A review of Statistical Theory of Signal Detection”, in B.F. Schutz, ed., *Gravitational Wave Data Analysis*, Proceeding of the NATO Advanced Research Workshop, held at Dyffryn House, St. Nicholas, Cardiff, Wales, 6-9 July 1987, vol. **253** of NATO ASI Series C, 73-94.
- [50] Bernard F. Schutz, *Introduction to the Analysis of Low-Frequency Gravitational Wave Data*, gr-qc/9710080.
- [51] Lee S. Finn, *Phys. Rev.* **D46**, No.12, 5236 (1992).
- [52] Curt Cutler and Bernard F. Schutz, *Phys. Rev.* **D72**, 063006 (2005).
- [53] Bernard F. Schutz and Maria Alessandra Papa, *End-to-end algorithm for hierarchical area searches for long-duration GW sources for GEO 600*, gr-qc/9905018.
- [54] B. Abbott *et al.*. (The LIGO Scientific Collaboration), *Phys. Rev.* **D69**, 082004 (2004).
- [55] Yousuke Itoh, *Is the “Conservative” frequentist upper limit always worse than the “Optimistic” Bayesian upper limit ? NoA Simple Counterexample*. LIGO Technical Notes LIGO-T030248-00-Z (2003).
- [56] Curt Cutler, Iraj Gholami and Badri Krishnan, *Phys. Rev. D* **72**, 042004 (2005).



- [57] P.R. Brady, T. Creighton, C. Cutler, and B.F. Schutz, *Phys. Rev.* **D57**, 2101 (1998).
- [58] P.R. Brady and T. Creighton, *Phys. Rev.* **D61**, 082001 (2000).
- [59] The LAL software can be found on the following websites:  
<http://www.lsc-group.phys.uwm.edu/daswg/projects/lal.html>  
<http://www.lsc-group.phys.uwm.edu/daswg/projects/lalapps.html>
- [60] The doxygen version of the LAL softwares can be found in this site:  
<http://www.lsc-group.phys.uwm.edu/lal/slug/nightly/doxygen/html/dirs.html>
- [61] R. Prix and Y. Itoh, *gr-qc/0504006*.
- [62] B. Krishnan, A.M. Sintes, M.A. Papa, B.F. Schutz, S. Frasca, and C. Palomba, *Phys. Rev.* **D70**, 082001 (2004).
- [63] B. Owen, *Phys. Rev.* **D53**, 6749 (1996).
- [64] Reinhard Prix, *Phys. Rev.* **D75**, 023004 (2007).
- [65] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, Springer (1991).
- [66] P. Astone, K.M. Borkowski, P. Jaranowski, and A. Królak, *Phys. Rev.* **D65**, 042003 (2003).
- [67] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
- [68] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, *Science* **220**, No. 4598, 671 (1983).
- [69] J.A. Nelder and R. Mead, *Comput. J.* **7**, 308-313, 1965.
- [70] W.H. Press, *Numerical Recipes in C*, Cambridge University Press (2002).
- [71] <http://www.gnu.org/software/gsl/>.
- [72] S.V. Dhurandhar and A. Vecchio, *Phys. Rev.* **D63**, 122001 (2001).
- [73] Yousuke Itoh, Gregory Mendell, and Xavier Siemens, *LIGO Technical Report* No. T050052-00-Z (2004).
- [74] Xavier Siemens, *Notes on LALDemod, LSC S2 Investigation page* (2004).
- [75] Yousuke Itoh, *personal Notes on Parameter Estimation, dated July 23, 2004*.
- [76] <http://www.lsc-group.phys.uwm.edu/cgi-bin/enote.pl?nb=puls5knownpulsardemod&action=view&page=11>
- [77] <http://www.ligo-wa.caltech.edu/gmendell/PULG/StackSlideUtilities/review/testNormalization.html>
- [78] <http://www.atnf.csiro.au/research/pulsar/psrcat/>
- [79] Reinhard Prix, *Comparison between CFS\_v2 and CFS\_v1 in pure noise*  
[http://www.aei.mpg.de/repr/EnoteEntries/CFSv2\\_Checks/](http://www.aei.mpg.de/repr/EnoteEntries/CFSv2_Checks/)



- [80] B. Abbott *et al.*. (The LIGO Scientific Collaboration), *All-sky LIGO search for periodic gravitational waves in the S4 data*, arXiv:0708.3818 [gr-qc]
- [81] [http://gallatin.physics.lsa.umich.edu/~keithr/cw/S4\\_injection\\_params.html](http://gallatin.physics.lsa.umich.edu/~keithr/cw/S4_injection_params.html)
- [82] Matthew Pitkin, *PhD Thesis*, University of Glasgow (2006).
- [83] B. Abbott *et al.*. (The LIGO Scientific Collaboration), *Upper limits on gravitational wave emission from 78 radio pulsars*, Phys. Rev. **D76**, 042001 (2007).
- [84] Glen Cowan, *Statistical Data Analysis*, Oxford Science Publications, 1998.
- [85] Réjean Dupuis, *PhD Thesis*, University of Glasgow (2004).
- [86] Réjean Dupuis and Graham Woan, *Bayesian estimation of pulsar parameters from gravitational wave data*, Phys. Rev. **D72**, 102002 (2005).
- [87] Michael Landry, *private communications*.
- [88] Gregory Mendell, *private communications*.
- [89] J.R. Gair, L. Barack, T. Creighton, C. Cutler, S.L. Larson, E.S. Phinney, and M. Vallisneri, *Event rate estimates for LISA extreme mass ratio capture sources*, Proceedings of the 8th GWDAW Meeting (Milwaukee, 2003); Class. Quant. Grav. **21**, S1595 (2004).



# Appendix D

## Curriculum Vitae

### D.1 PERSONAL INFORMATION

|                                   |  |
|-----------------------------------|--|
| <b>Name</b>                       | Iraj Gholami Ghadikolaie   |
| <b>Date/Place of Birth</b>        | July 26, 1973/ Ghaemshahr (Iran).  |
| <b>Gender/ Marital Status</b>     | Male/ Married (one child)  |
| <b>Present Institution</b>        | Max Planck Institut für Gravitationsphysik -<br>Albert Einstein Institute (AEI),<br>Potsdam, Germany.  |
| <b>Address for Correspondence</b> | Max Planck Institut für Gravitationsphysik -<br>Albert Einstein Institute (AEI),<br>Am Mühlenberg 1, D-14476 Potsdam, Germany.<br>Tel: (+49) 331 - 567 7123,<br>Fax: (+49) 331 - 567 7298. |
| <b>Electronic Address</b>         | iraj.gholami@aei.mpg.de, gholamiraj@yahoo.com,<br>igholami@gmail.com   |

### D.2 EDUCATIONS

| Degrees                       | University | Place            | Year Attended           |
|-------------------------------|------------|------------------|-------------------------|
| B.Sc. in Physics              | Yazd Univ. | Yazd, Iran       | 1991-1997               |
| M.Sc. in Astrophysics/Physics | IASBS      | Zanjan, Iran     | 1998- 2000              |
| Ph.D. in Astrophysics/Physics | IIA        | Bangalore, India | 2002-2003<br>(Resigned) |
| Ph.D. in Astrophysics/Physics | MPI-AEI    | Potsdam, Germany | 2003-2007               |

MPI-AEI= [Max Planck Institut für Gravitationsphysik - Albert Einstein Institute](#)

IIA= [Indian Institute of Astrophysics](#)

IASBS= [Institute for Advanced Studies in Basic Sciences](#)

## D.3 PUBLICATIONS

### • Publications related to my PhD thesis

#### Refereed Journal

1. **Improved Stack-Slide Searches for Gravitational-Wave Pulsars,**  
Curt Cutler, Iraj Gholami, and Badri Krishnan, *Phys. Rev. D*, **72**, 042004 (2005) (*preprint: gr-qc/0505082*).

#### Conference proceedings and abstracts

1. **Computing the F-Statistic for continuous gravitational waves using a network of detectors,**  
Iraj Gholami, Reinhard Prix and Curt Cutler, *The 10th Annual Gravitational Wave Data Analysis Workshop (GWDAW-10)*, Brownsville, USA, December 14-17, 2005.
2. **Optimized Search Strategies for Continuous Gravitational Waves,**  
Iraj Gholami, Curt Cutler and Badri Krishnan, *The 9th Annual Gravitational Wave Data Analysis Workshop (GWDAW-9)*, Annecy, France, December 15-18, 2004.
3. **Hierarchical Search Strategies for Unknown Pulsars,**  
Iraj Gholami, Curt Cutler and Badri Krishnan, *The 17th International Conference on General Relativity and Gravitation*, Dublin, Ireland, July 18-23, 2004.
4. **Optimized Hierarchical Continuous-Wave Searches,**  
Badri Krishnan, Curt Cutler and Iraj Gholami, *LSC (LIGO Scientific Collaboration) Meeting (ASIS session)*, LIGO Livingston Observatory, Livingston, Louisiana, USA, March 15-18, 2004.

### • Publications as being a member of LSC during my PhD study at AEI

1. **Searches for periodic gravitational waves from unknown isolated sources and Scorpius X-1: Results from the second LIGO science run,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *Phys. Rev. D* **76**, 082001 (2007).
2. **All-sky search for periodic gravitational waves in LIGO S4 data,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *preprint: arXiv: 0708.3818 [gr-qc]*.
3. **Search for Gravitational Waves Associated with 39 Gamma-Ray Bursts Using Data from the Second, Third, and Fourth LIGO Runs,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *preprint: arXiv: 0709.0766 [gr-qc]*.
4. **A Joint Search for Gravitational Wave Bursts with AURIGA and LIGO,**  
By AURIGA Collaboration, LIGO Scientific Collaboration (L. Baggio et al.), *preprint: arXiv: 0710.0497 [gr-qc]*.
5. **Search for gravitational waves from binary inspirals in S3 and S4 LIGO data,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *preprint: arXiv:0704.3368 [gr-qc]*.
6. **Search for gravitational-wave bursts in LIGO data from the fourth science run,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *Class. Quant. Grav.*, **24**, 5343 (2007) (*preprint: gr-qc/0704.0943*).
7. **Search for gravitational wave radiation associated with the pulsating tail of the SGR 1806-20 hyperflare of 27 December 2004 using LIGO,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *Phys. Rev. D* **76**, 062003 (2007) (*preprint: astro-ph/0703419*).
8. **First Cross-Correlation Analysis of Interferometric and Resonant-Bar Gravitational Wave Data for Stochastic Backgrounds,**  
By LIGO Scientific Collaboration, ALLEGRO Collaboration (B. Abbott et al.), *Phys. Rev. D* **76**, 022001 (2007) (*preprint: gr-qc/0703068*).

9. **Upper limit map of a background of gravitational waves,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *preprint: astro-ph/0703234*.
10. **Upper limits on gravitational wave emission from 78 radio pulsars,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *Phys. Rev. D* **76**, 042001 (2007) (*preprint: gr-qc/0702039*).
11. **Searching for a Stochastic Background of Gravitational Waves with LIGO,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *Astrophysical J.* **659**, 918 (2007) (*preprint: astro-ph/0608606*).
12. **The GEO-HF project,**  
By GEO Scientific Collaboration, (B. Willke et al.), *Class. Quant. Grav.*, **23**, S207-S214 (2006).
13. **Status of the GEO600 detector,**  
By GEO Scientific Collaboration (H. Luck et al.), *Class. Quant. Grav.*, **23**, S71-S78 (2006).
14. **Search for gravitational-wave bursts in LIGO's third science run,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *Class. Quant. Grav.*, **23**, S29-S39 (2006) (*preprint: gr-qc/0511146*).
15. **Upper limits on a stochastic background of gravitational waves,**  
By LIGO Scientific Collaboration (B. Abbott et al.), *Phys. Rev. Lett.*, **95**, 221101 (2005) (*preprint: astro-ph/0507254*).
16. **First report on the S3 analysis by Einstein@Home,**  
[http://einstein.phys.uwm.edu/PartialS3Results/einstein\\_writeup.html](http://einstein.phys.uwm.edu/PartialS3Results/einstein_writeup.html).  
(This report was refereed by many scientists, however the authors decided not to publish it in a journal instead being published for a public usage in the web-page.)

## D.4 SEMINARS & SCHOOLS ATTENDED (during my PhD at AEI)

| Date              | Place  | Title  |
|-------------------|--|--|
| Oct. 20-25, 2007  | Albert Einstein Institute,<br>Hannover, Germany                        | LIGO Scientific Collaboration (LSC)<br>Meeting                                 |
| Dec. 18-21, 2006  | Albert Einstein Institute,<br>Potsdam, Germany                         | The 11th Annual Gravitational Wave<br>Data Analysis Workshop (GWDAW11)         |
| Aug. 11-18, 2006  | Louisiana State University,<br>Baton Rouge, USA                        | LIGO Scientific Collaboration (LSC)<br>Meeting                                 |
| July. 23-29, 2006 | Freie Universität,<br>Berlin, Germany                                  | 11 <sup>th</sup> Marcel Grossmann Meeting                                      |
| May. 22-26, 2006  | European Gravitational Observatory,<br>VIRGO Site, Cascina (PI), Italy | The 1 <sup>st</sup> VESF school on<br>Gravitational Wave                       |
| Dec. 14-17, 2005  | University of Texas at Brwonsville,<br>Brownsville, USA                | The 10th Annual Gravitational Wave<br>Data Analysis Workshop (GWDAW10)         |
| Nov. 17-20, 2005  | MIT,<br>Boston, USA  | LIGO Scientific Collaboration (LSC)<br>Meeting                                 |
| June 20-24, 2005  | Okinawa, Japan   | 6 <sup>th</sup> Edoardo Amaldi Conference<br>on Gravitational Waves (Amaldi 6) |
| Dec. 15-18, 2004  | Lab. d'Annecy-le-vieux de Phys.<br>Annecy, France                      | The 9th Annual Gravitational Wave<br>Data Analysis Workshop (GWDAW-9)          |
| Sept. 15-26, 2003 | Int. Center for Theoretical Phy.,<br>Trieste, Italy                    | Advanced School and Conference on<br>Sources of Gravitational Waves            |
| Aug. 18-21, 2003  | Hannover University,<br>Hannover, Germany                              | LIGO Scientific Collaboration (LSC)<br>Meeting                                 |