Studying Gravitational Waves with Pulsars: Results from the Parkes Pulsar Timing Array

Daniel Roger Billing Yardley

THE UNIVERSITY OF
SYDNEY

A thesis submitted for the degree of
Doctor of Philosophy
at the University of Sydney

May, 2011

“One man’s noise is another man’s signal.”
– Edward Ng, *The New York Times*, March 20, 1990

Abstract

This thesis addresses the problem of gravitational-wave (GW) detection using radio timing observations of pulsars. We study GW signals in real and simulated pulsar timing observations, and describe the astrophysical implications for cases in which no GW signal is detected. We simulate timing observations from a range of hypothetical pulsar timing array projects. The pulse arrival times are then perturbed by an individual source of GWs. One of the simulated data sets comprises an array of 20 pulsars timed with a root-mean-square residual of 10 ns over 10 years. If there is no detectable GW signal in this data set, then the merger rate of supermassive black-hole binaries (SMBHBs) with a chirp mass of 10^9 solar masses is less than one merger every 10^5 years up to a redshift of $z = 2$. This constraint rules out estimates of the SMBHB merger rate based on hierarchical galaxy formation with standard assumptions on the merger parameters. Applying a similar analysis to recently published observations from the Parkes Pulsar Timing Array (PPTA) yields a constraint on the merger rate of SMBHBs of less than one merger every five years for SMBHBs with chirp mass $\sim 10^{10}$ solar masses up to a redshift of $z = 2$. The results also indicate that it is unlikely that an individual GW source could be detected with existing data sets.

We consider the signal caused by an isotropic stochastic gravitational-wave background (GWB), and show that, with a few more years of observations, either the GWB will be detected or the parameter space of most current galaxy evolution models will be significantly constrained. An analysis of the cross-correlation between the timing residuals of different pulsars in the PPTA shows that there is no detectable GWB signal in the current data. The GWB detection statistic is dominated by only a few pulsars in the current PPTA data. There are good prospects for detection of the GWB using radio timing of pulsars in the next decade. We conclude that the effect of instabilities in realisations of the terrestrial timescale and inaccuracies in the solar system ephemeris must be removed from the timing residuals in order to detect a GW signal in pulsar timing observations in the future. The forthcoming International Pulsar Timing Array project will also significantly increase the probability of detection of GW signals using pulsar timing.

Statement of Originality

This thesis describes work carried out in the Sydney Institute for Astronomy within the School of Physics at The University of Sydney and at CSIRO Astronomy and Space Science (formerly ATNF), between March 2007 and May 2011. I am the author of all text in this thesis. The work presented in this thesis is, to the best of my knowledge and belief, my own original work, except for the following:

- In Section 2.4.1, the FAKE plugin to TEMPO2 was written by George Hobbs.
- In Section 3.3.4, the text is based on Wen et al. (2011) [a paper on which I am a coauthor] and Figures 3.8 and 3.9 are reproduced from Wen et al. (2011).
- Figure 4.5 was published by Yardley et al. (2010), but was created by Zhonglue Wen from a data set that I provided.
- Section 5.1.2 is a summary of work by Hobbs et al. (2009) [a paper on which I am a coauthor], and Figures 5.1 and 5.2 are reproduced from Hobbs et al. (2009).
- Section 5.1.3 is a summary of work by Jenet et al. (2006) and Hobbs et al. (2009).
- Figures 5.3 and 5.4 are reproduced from Wen et al. (2011) and were created by Zhonglue Wen from results that I provided.

I hereby declare that I have not submitted this material, either in full or in part, for a degree at this or any other institution.

....

Daniel Roger Billing Yardley Date

First-author / Co-authored Publications Included

The original research contained in this thesis has formed the basis of several articles in peer-reviewed journals and conference proceedings. All of the work consists of research conducted by myself, in consultation with my supervisor George Hobbs and sometimes other collaborators. Some Sections of this thesis have been reworded or summarised from various publications, as indicated below:

- [Sections 1.6, 2.2.3, 2.2.5, 2.3.2, 2.4.3, 6.1, 6.2, 6.3, 6.4 and 6.5]
  Yardley D. R. B., Coles W. A., Hobbs G. B., et al. 2011, MNRAS, 414, 1777
- [Section 2.5]
  Hobbs G., Jenet F., Lee K. J., et al. 2009, MNRAS, 394, 1945
- [Sections 2.6, 3.2, 3.3 and 5.2]
  Wen Z. L., Jenet F. A., Yardley D., Hobbs G. B., Manchester R. N., 2011, ApJ, 730, 29
- [Sections 3.1, 4.1, 4.2, 4.3 and 4.4]
  Yardley D. R. B., Hobbs G. B., Jenet F. A., et al. 2010, MNRAS, 407, 669
- [Section 6.4]
  Yardley D. R. B., Coles W. A., Hobbs G. B., Manchester R. N., 2011, in Burgay M., D'Amico N., Esposito P., Pellizzoni A., Possenti A., eds, *Radio Pulsars: An Astrophysical Key to Unlock the Secrets of the Universe*, Vol. 1357 of AIP Conference Series. American Institute of Physics, Melville, New York, p. 77

Acknowledgements

Professional acknowledgments: i.e., some people whom I would love to thank for help with the actual wording, editing, researching etc. etc. of this thesis. The

most important person to thank in this regard is my supervisor, Dr. George Hobbs. George has provided extensive feedback on the structure of all Chapters and has been of great assistance in making sure the content was correct and complete. He's also been invaluable in helping me write various papers and debug my computer codes. My other supervisor, Prof. Bryan Gaensler, has also helped in keeping my feet on the ground and in providing feedback from the point of view of a researcher who is generally not working on pulsar timing for the detection of gravitational waves. Prof. Dick Manchester provided helpful feedback on Chapters 1 and 2 as well as providing extensive feedback on many of my publications when they were being developed. Vikram Ravi also gave some useful comments on Chapter 1.

The research presented in this thesis would not have been possible without the time and effort of George Hobbs, Prof. Bill Coles, Prof. Dick Manchester and A/Prof. Fredrick Jenet. I also acknowledge Yuri Levin for commenting on a draft version of the paper that formed Chapter 6, and Xavier Siemens, Larry Price, Rutger van Haasteren, Paul Demorest and Kejia Lee for useful discussions on GWB detection. For the administration side, the assistance of Jo Daniels, Eve Teran, Hyacinth Alfonso and Julia Hockings have been second to none.

Official acknowledgments: This work is undertaken as part of the Parkes Pulsar Timing Array project. The Parkes radio telescope is part of the Australia Telescope, which is funded by the Commonwealth of Australia for operation as a National Facility managed by the Commonwealth Scientific and Industrial Research Organisation (CSIRO). DRBY was funded by an Australian Postgraduate Award and the CSIRO Office of the Chief Executive PhD scholarship program.

"Anything can happen" acknowledgments: This is really just a list of shout-outs to various excellent people. As you can probably infer from above, George has been pretty indispensable throughout the whole 4 years. His boundless enthusiasm and ability to work at incredible speeds while still having plenty of time for his students have certainly made this whole PhD thing much easier. As George's first PhD student, he has nothing short of my highest recommendation for other students.

Thanks to Dick for never-ending enthusiasm for ping pong, the 5pm "all-stars" matches at ATNF/CASS are among my favourite ATNF memories. Thanks to Jono for being an outstanding teammate in the unbeatable ping pong (and international travel) duo, Team Young. Many thanks also to Bryan for many lifts to and from ATNF, an early-morning car-load of astronomers is certainly an experience.

Thanks to the Parkes observatory staff for making the telescope work and generally being a helpful, cheerful and smart bunch. Thanks to Shirley, Anne and all the quarters staff for many delicious meals and warm beds.

Enormous thanks to Christine for many a lazy Kuletos night; Madhura for many trips from our office to Friday drinks, as well as being my tiny friend with her enormous car. Thanks for a laugh to Matt F. for sneezing like a girl, and to Holly for also sneezing like a girl (and being generally fabulous). Thanks to Richard for helping me fill my water bottle and being an all-round good guy. Shout-outs to Prajwal, Chris (good long chats with a cool guy), Juliana (ever helpful and reliable), Magda & Vicente (new officemates!) and Brendon (good for a chat and a magic trick) for fun times in the Outpost/Rm 468.

For the sports that have helped keep me sane, shout-outs to indoor soccer with Butter Chicken/Pub Grub and The Physicists, volleyball with Project Mayhem, indoor cricket with Rhomboids, ultimate frisbee with Fabulous February / Mad March of Frisbee, and table soccer with all my friends, especially my unforgettable and irreplaceable usydgroup friends.

For the music that has provided a much-needed creative outlet, big thanks to Jenny Green and Barefoot Musica Antigua, and various people who've made the Madrigal Society work. Shout-out to James Dixon for being generous with his consistent gig-offerings.

For day-to-day living, thanks to Pat for putting up with a thesis-writing housemate for so long, and shout-outs to Marija and Matt C. For my overseas trips, I'm eternally indebted to all the people who hosted me: Bill & Alma in San Diego, Rick for arranging all my Brownsville accommodation, Alison and Chris Hobbs for a lovely few days in Ottawa, Marija for amazing Berlin experiences, Brendon for fun times in Santa Barbara and Michelle for a very cool Cambridge experience.

Finally, I would like to acknowledge the continuing love and support of my girl-friend, my brothers and my parents. Growing up as the youngest of four smart boys in an academic household was intellectually challenging, to say the least. However, I think the actual acknowledgment was best-expressed in my eldest brother's PhD thesis: my parents (and brothers!) "created an atmosphere in which ideas are valued".

Thanks all.

Contents

List of Abbreviations and Common Units

BH	black hole
BHB	black-hole binary
BIPM	Bureau International des Poids et Mesures
CPSR2	Caltech-Parkes-Swinburne Recorder 2
Dec	declination
DFT	Discrete Fourier Transform
DM	dispersion measure
EM	electromagnetic
EPTA	European Pulsar Timing Array
$f_{1\,\mathrm{yr}}$	$1/(1\,\mathrm{yr})$, which is approximately $3.18 \times 10^{-8}\,\mathrm{Hz}$
FPTM	Caltech fast pulsar timing machine
Galaxy	Milky Way Galaxy
GR	general relativity
GW	gravitational wave
GWB	gravitational-wave background
H_0	the Hubble constant, taken to be $72\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$
ISM	interstellar medium
LAD	least absolute deviation
LIGO	Laser Interferometer Gravitational-Wave Observatory
LISA	Laser Interferometer Space Antenna
$\mathrm{M}_\odot$	solar mass, which is approximately $2 \times 10^{30}\,\mathrm{kg}$
MBHB	massive black-hole binary
MJD	Modified Julian Day/Date
MSP	millisecond pulsar
NANOGrav	North American Nanohertz Observatory for Gravitational Waves (this is the North American Pulsar Timing Array)
N_{psr}	Number of pulsars
N_{pts}	Number of timing residuals
PPTA	Parkes Pulsar Timing Array
PSR	pulsating source of radio

PTA	pulsar timing array
RA	right ascension
rms/RMS	root-mean-square
SMBH	supermassive black hole
SMBHB	supermassive black-hole binary
S/N ratio	signal-to-noise ratio
ToA	time-of-arrival
T_{obs}	The time-span of a set of timing residuals, typically several years
T_{overlap}	The time-span of the overlapping portion of a pair of time series, typically a few years
TT	Terrestrial Time
TT(TAI)	Terrestrial Time as realised by International Atomic Time
UTC	Universal Coordinated Time
WLSQ	weighted least-squares

List of Figures

List of Tables

Chapter 1

Introduction to Pulsar Timing and Gravitational Waves

Chapter Outline: *In this Chapter, we describe:*

- *pulsars and their properties;*
- *millisecond pulsars and techniques for observing them;*
- *gravitational waves including some common sources of gravitational waves and a summary of projects aiming to detect them;*
- *techniques for detecting gravitational waves using radio timing observations of millisecond pulsars;*
- *the current state of the field in detecting and placing limits on gravitational waves with pulsar timing;*
- *an outline of the thesis.*

1.1 Pulsars

A pulsar is a rapidly rotating magnetised neutron star that emits beams of electromagnetic (EM) radiation. The first pulsar was discovered in 1967 (Hewish et al., 1968). Because the EM beams periodically sweep over the Earth (like a lighthouse beam sweeps across an observer), the pulsar detection was made by observing a sequence of pulses in a time series that J. Bell had obtained using a radio telescope (reproduced in Figure 1.1). The pulses were periodic and appeared at the same sidereal time every day. This led Bell and Hewish to confront the possibility that these were artificially-generated signals from extraterrestrial planets (Hewish, 1975). It was soon realised that the pulses could be the radio emission from the super-dense stellar remnant of a supernova (e.g., Gold, 1968), and, in the 44 years that have followed, almost 2000 pulsars have been catalogued (Manchester et al., 2005)[1].

[1] The ATNF Pulsar Catalogue: `http://www.atnf.csiro.au/research/pulsar/psrcat/`.

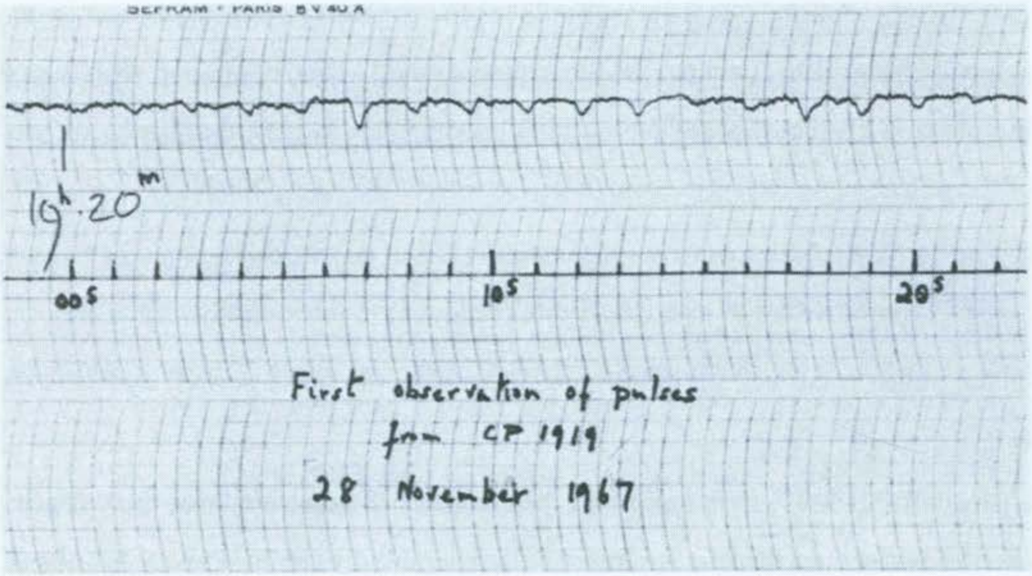

Figure 1.1: The pen chart showing the detection of pulsed radio emission from a pulsar, with time on the horizontal axis and intensity given by vertical deflections (in the top trace). The pulses have a 1.33 s period, and appear at the same sidereal time every day. [Image reproduced from Hewish (1975)]

1.1.1 Properties

A typical pulsar is born after a star with initial mass approximately in the range $8\mathrm{M}_\odot$ to $15\mathrm{M}_\odot$, where $1\mathrm{M}_\odot \approx 2 \times 10^{30}$ kg is one Solar mass, undergoes core collapse to produce a supernova (Lyne & Smith, 2005). The core of the progenitor usually remains after the supernova explosion and forms a neutron star with a typical mass of $1.4\mathrm{M}_\odot$, radius of 10 km and surface magnetic field of $\sim 10^{12}$ gauss (Chandrasekhar, 1935; Pacini, 1967; Gold, 1968). This neutron star can be detected as a pulsar if it emits EM radiation from its magnetic poles, the magnetic and rotation axes are misaligned and its emission beams intersect the line of sight to the Earth. Because pulsars lose energy through a range of processes (e.g., emitting a relativistic particle wind; Lorimer & Kramer, 2005), their angular velocity will decrease as they age. We can model the rate of decrease in the pulsar's angular velocity ω as a power law[2]:

$$\dot{\omega} \propto -\omega^n \, . \tag{1.1}$$

[2] A power-law model is chosen because it gives simple estimates for several pulsar properties, without assuming that the pulsar spin-down is caused by magnetic dipole braking in vacuum.

If we assume that the only mechanism that reduces the rotation rate of the pulsar is magnetic dipole braking (Jackson, 1962), then $n = 3$ in Equation (1.1). This, combined with the assumption that the pulsar's present day period is much greater than its period at birth, allows for an estimate of the pulsar's *characteristic age* (e.g., Lorimer & Kramer, 2005):

$$\tau_c = \frac{P}{2\dot{P}} , \tag{1.2}$$

where P is the pulsar rotational period (assumed equal to the pulse period) and $\dot{P}$ is its time-derivative.

We can extend this analysis to estimate the magnetic field strength at the surface of the pulsar. We assume not only that the magnetic field is a pure dipole (as above), but also that the magnetic axis is perpendicular to the rotation axis. Taking canonical values of the radius and the moment of inertia of the pulsar (10 km and $10^{38}\,\mathrm{kg\,m^2}$ respectively), we obtain the following formula for the surface magnetic field at the pulsar's magnetic equator (e.g., Manchester & Taylor, 1977):

$$B_{surf,eq} = 3.2 \times 10^{19} (P\dot{P})^{1/2} \ \mathrm{G} , \tag{1.3}$$

where P is in units of seconds. However, the assumptions on the radius and moment of inertia of the pulsar and that the magnetic and rotation axes are perpendicular will be inaccurate. Hence, Equation (1.3) gives at best an order of magnitude estimate of $B_{surf,eq}$. The magnetic field strength at the poles is expected to be a factor of two larger (Shapiro & Teukolsky, 1983; Usov & Melrose, 1995).

It is apparent from equations (1.2) and (1.3) that P and $\dot{P}$ are instrumental in determining the present properties of a given pulsar, and also its evolutionary history and future. A popular graphical representation for the pulsar population is the P-$\dot{P}$ diagram, which is plotted in Figure 1.2 for 1702 pulsars. The thin-dotted lines correspond to lines of constant characteristic age, indicating that most pulsars are born in the upper-left corner of the diagram. As pulsars age, it was originally thought that the magnetic field decays exponentially with a $\sim 5 \times 10^6$ yr timescale (Bhattacharya & van den Heuvel, 1991). However, recent work suggests that early pulsar evolution may include an increase in magnetic field strength for some pulsars (Lyne, 2004; Lyne & Smith, 2005). In the standard model, a pulsar's period increases and its period derivative decreases such that it joins the population of "normal" pulsars with periods $\sim$ 1 s, and period derivatives $\sim 10^{-15}$.

The pulsar will eventually evolve to a state in which it can no longer produce its characteristic beams of EM radiation. While the pulsar emission mechanism is not well-understood, it is believed that an electron-positron pair cascade process is required (Melrose, 2004). When there is insufficient energy for this cascade process to continue, pulsars will cease their emission. This occurs when a pulsar's rotation rate slows, and thus provides a "death line" on Figure 1.2. This death line is thought to occur at (adapted from Ruderman & Sutherland, 1975)

$$\dot{P}P^{-3} = 2.8 \times 10^{-17} \ \text{s}^{-3} \, . \tag{1.4}$$

However, this line is by no means a hard limit; Lyne & Smith (2005) point out that some pulsars become faint enough to avoid detection well before crossing the death line, while others are detected with larger P and smaller $\dot{P}$ than this "limit" allows.

After crossing the death line, some pulsars can be reborn because of accretion from a binary companion. These "recycled" pulsars form the topic of the next Section.

1.1.2 Recycled and Millisecond Pulsars

A review of a variety of formation mechanisms for recycled pulsars and so-called "millisecond pulsars" (MSPs) is given by Bhattacharya & van den Heuvel (1991). The most widely accepted mechanism contains the following essential steps:

1. The larger mass member (the primary) of a sufficiently massive binary system evolves to supernova before the smaller mass member (the secondary) leaves the main sequence. The supernova will disrupt the binary orbit in at least 90% of cases (Radhakrishnan & Shukre, 1985; Dewey & Cordes, 1987).

2. After the supernova explosion, the primary forms a neutron star. In the rare case that the binary system is not disrupted, the secondary remains largely unperturbed. The neutron star primary may or may not evolve to cross the death line in Figure 1.2.

3. The secondary evolves and leaves the main sequence, causing it to expand until it overflows its Roche lobe. This causes accretion onto the neutron star primary, which transfers angular momentum to the neutron star, increasing its rotational frequency. If the primary is visible as a pulsar, it will now be termed a "recycled pulsar".

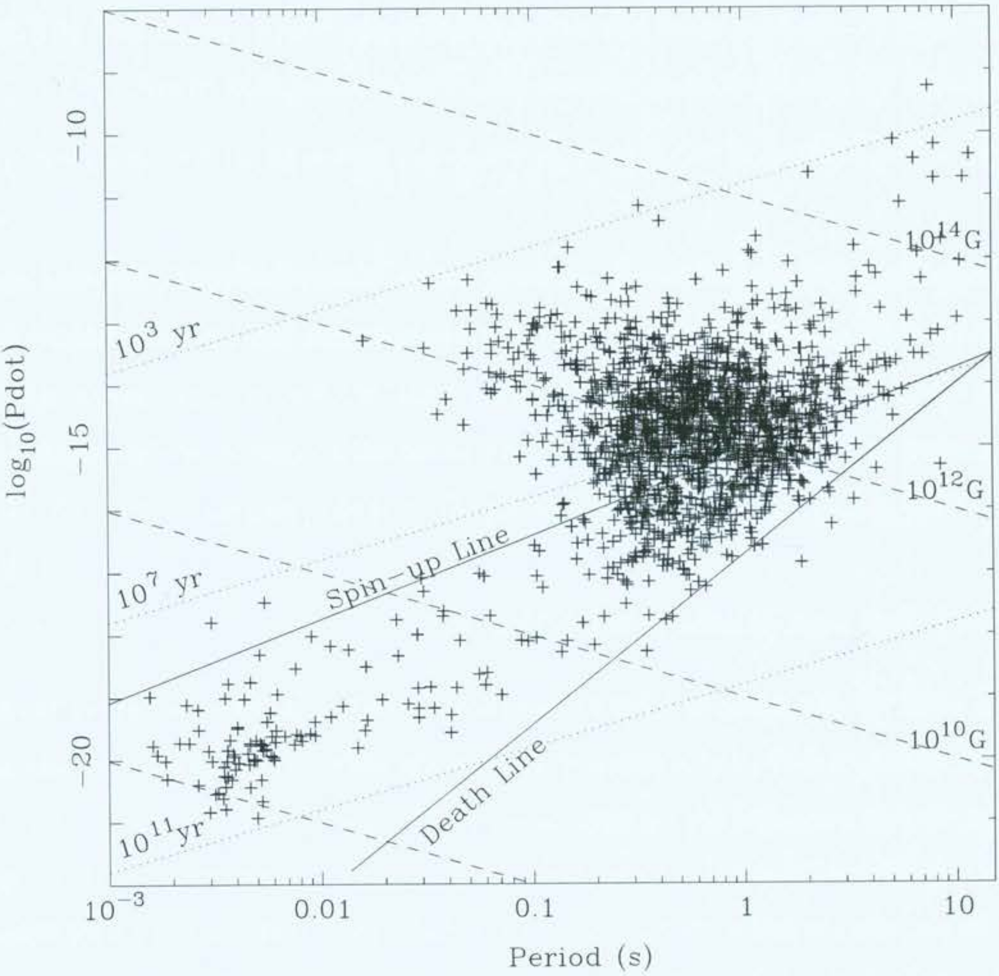

Figure 1.2: The $P - \dot{P}$ diagram for all 1702 pulsars with measurements of P and $\dot{P}$ ('+' symbols), where $\dot{P} > 0$, as at February 19th, 2011. Also shown are the lines of constant characteristic age (dotted lines) and constant magnetic field (dashed lines). The "death line" (solid line with label) indicates a rough lower limit on the product $\dot{P}P^{-3}$ (see Equation 1.4), below which the pulsar emission will cease. Some pulsars in binary systems can then be "recycled" to the lower-left quadrant and reappear as short-period pulsars (see Section 1.1.2). The "Spin-up Line" (solid line with label) indicates an estimate of the minimum spin period of recycled pulsars for a given value of $\dot{P}$.

4. The final rotational period of the recycled pulsar depends, among other things, on the mass of the secondary star. If the secondary star is not massive enough to experience core-collapse, then its red giant phase will provide accreting matter to the primary for $\sim 10^7$ years or more. The primary star will then be spun up to a rotational period of $\sim$ms. The weakening of the primary's magnetic field during the accretion process also ensures that the primary will spin down slowly compared to young pulsars. The primary star is now a MSP.

A lower bound on the period of the MSP can be found by analysing the interaction of the magnetised accreting matter with the magnetosphere of the MSP. The accretion flow exerts torque on the MSP at the outer boundary of its magnetosphere, known as the "Alfvén surface" (Lyne & Smith, 2005). The spin-up process continues until the angular velocity of the magnetosphere equals the angular velocity of the accreting matter at the Alfvén surface. When this condition is satisfied, P and $\dot{P}$ will be related by (Arzoumanian et al., 1999):

$$\dot{P} = \alpha P^{4/3} \,, \tag{1.5}$$

where the value of α is uncertain; we have assumed $\alpha = 8.3 \times 10^{-16}\,\mathrm{s}^{-4/3}$. Equation (1.5) provides a lower bound on P for a given value of $\dot{P}$ that is labelled "Spin-up Line" in Figure 1.2. Thus, the population of MSPs is expected to populate the region between the spin-up line and the death line.

The first MSP was discovered in 1982 (Backer et al., 1982), with a rotational frequency of 642 Hz. About 7% of all known pulsars today can be classed as MSPs, and they form a population in the lower left corner of Figure 1.2 that is distinct from normal pulsars. They are useful laboratories for astrophysics because of their very high matter density, extreme rotational speeds and predictable spin-down behaviour.

1.2 Observing Pulsars

Pulsars are observed using large-aperture radio telescopes fitted with low system-temperature receivers. The signal from the telescope is processed using a "backend" system, which is usually a digital signal processor. When observing their radio pulses, a range of effects must be taken into account during each pulsar observation. The dispersive effects of the interstellar medium

(ISM) must be corrected, as the ISM acts to "smear out" each pulse (see Lorimer & Kramer, 2005). Charged particles in the ISM change the group velocity of radio waves as a function of frequency, which gives the following formula for their group velocity (Shapiro & Teukolsky, 1983):

$$v(f_{\rm EM}) = c\left(1 - \frac{f_e^2}{f_{\rm EM}^2}\right)^{1/2} \quad {\rm m\,s^{-1}}\,, \tag{1.6}$$

where $f_{\rm EM}$ is the radio frequency, v is the group velocity of the waves, c is the speed of light in vacuum and f_e is the plasma frequency for the ISM, which is a function of the density of charged particles n_e. According to Equation (1.6), a wave of infinite frequency will travel at c through plasma. The difference in arrival time at the telescope for two radio waves with frequencies $f_{\rm EM,1}$ and $f_{\rm EM,2}$ after travelling a distance D from their source with velocities v_1 and v_2 is

$$\begin{aligned} \Delta t &= \int_0^D [v_1(l)^{-1} - v_2(l)^{-1}]dl \\ &\approx \frac{e^2}{2\pi m_e c}\left(f_{\rm EM,1}^{-2} - f_{\rm EM,2}^{-2}\right)\int_0^D n_e(l)dl\,, \end{aligned} \tag{1.7}$$

where we have written f_e in terms of the fundamental constants c, e (the charge of an electron) and m_e (the rest mass of an electron). The charged particle density is a function of l because the density of charged particles varies throughout the Galaxy (Taylor & Cordes, 1993). The integral of n_e along the line of sight is called the dispersion measure (DM). We can thus calculate the delay – compared to EM radiation travelling in vacuum – as a function of frequency by assuming that $f_{\rm EM,2} = \infty$ and $f_{\rm EM,1} = f_{\rm EM}$ in Equation (1.7). This leads to the following expression for the delay of a pulse of frequency $f_{\rm EM}$ because of plasma in the ISM:

$$\Delta t \approx 4.15 \times 10^3 \left[\int_0^D n_e(l)dl\right] f_{\rm EM}^{-2} \ \ {\rm s}\,. \tag{1.8}$$

This means that each pulse from the pulsar is smeared out when it is observed at the radio telescope, because each pulse consists of a range of frequencies (see Figure 1.3).

1.2.1 Incoherent De-dispersion

The dispersion effect described in the preceding Section can be corrected using the process of de-dispersion, which is often performed by the observing backend system. One method for de-dispersion is known as "incoherent" de-dispersion, where the range of observed frequencies is divided into many small segments, or "frequency channels". After measuring the DM, a

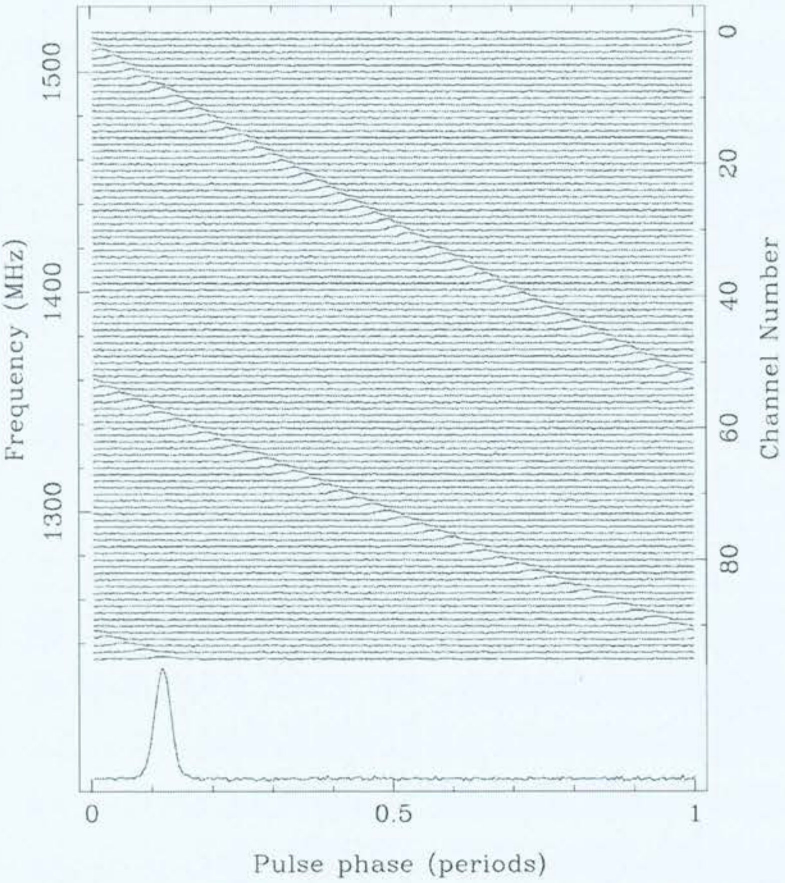

Figure 1.3: Pulse dispersion and "incoherent" de-dispersion. The abscissa gives the pulsar rotational phase. The pulse signal arrives at the telescope at a later time in lower frequency channels because of the dispersion induced by the interstellar plasma, as described by Equation (1.8). In the case shown here, the pulse is so dispersed compared to the pulse period that the difference in time-of-arrival between the highest frequency channel and the lowest is larger than two pulse periods. Thus, the delayed signal has wrapped across multiple cycles of pulse phase. The total bandwidth for this observation was 288 MHz, comprising 96 channels of 3 MHz bandwidth each. An artificial delay is induced in all frequency channels $f_{\mathrm{EM},1}$ with respect to the lowest frequency channel $f_{\mathrm{EM},2}$ using Equation (1.7). The addition of the signal in each delayed frequency channel gives the incoherently de-dispersed pulse shown in the lower panel. The absolute delay experienced by the lowest frequency ($f_{\mathrm{EM}} \approx 1233\,\mathrm{Hz}$) can be calculated using Equation (1.8). [Image obtained from: `http://www.cv.nrao.edu/course/astr534/Pulsars.html`; original image by Lorimer & Kramer (2005)]

time delay (described by Equation 1.7) is then induced in each channel relative to the lowest observed frequency, as shown in Figure 1.3. The delayed signals from each frequency channel can then be summed to produce a time series of de-dispersed pulses where the pulse arrival time is approximately independent of frequency.

1.2.2 Coherent De-dispersion

Incoherent de-dispersion cannot correct the pulse dispersion across the bandwidth of each frequency channel; for example, the dispersion across the 3 MHz bandwidth of each channel in the observation in Figure 1.3. This effect can be overcome using "coherent" de-dispersion (Hankins & Rickett, 1975). This technique is based on the fact that the frequency-dependent delays introduced by propagation through the ISM can be represented as phase rotations of the pulsar signal. These phase rotations depend on the frequency and the distance travelled by the pulse (e.g., Lorimer & Kramer, 2005). The effect of the ISM is to filter the pulsar signal using a filter with transfer function H. If the centre of the telescope's observing frequency range is at f_0, then the value of the transfer function at frequency $f_0 + f$ will be (Lorimer & Kramer, 2005)

$$H(f_0 + f) = \exp\left(\frac{\mathrm{i}2\pi[\mathrm{DM}]\mathcal{D}}{(f_0 + f)f_0^2} f^2\right), \tag{1.9}$$

where $\mathrm{i} = \sqrt{-1}$, $\mathcal{D} = 4.15 \times 10^3\ \mathrm{MHz}^2\ \mathrm{pc}^{-1}\ \mathrm{cm}^3\ \mathrm{s}$ is the dispersion measure constant from Equation (1.8) and $[\mathrm{DM}]$ is the dispersion measure, which can be measured by the telescope.

To perform the coherent de-dispersion, the phase rotations induced by the ISM are first determined by measuring the complex voltage signal observed at the telescope. These phase rotations are then "unwound" by applying the inverse of the transfer function in Equation (1.9) to the observed signal. This process has been implemented in several observing systems around the world (see Bailes, 2003; Demorest et al., 2004).

1.2.3 Folding

After coherent or incoherent de-dispersion, the mean pulse profile is formed using a "timing model" for the pulsar's behaviour. This model can be extremely detailed, and will include the pulsar's basic properties, such as its period, dispersion measure and sky-position, as well as a range of other effects if the pulsar is a member of a binary system. This model is used to "fold" the incoming signal at the apparent pulse period, which increases the signal-to-noise (S/N) ratio of a pulsar observation by summing the individual pulses. This process forms the mean pulse profile, or "folded" profile. After folding, the mean pulse profile of an MSP is largely invariant for that MSP (Lorimer, 2005). We can thus develop a standard template pulse profile, which is either an analytic model of a high S/N ratio observation of the pulsar, or simply a very high S/N ratio observation of the pulsar. By comparing the template to observations of the pulsar over a period of ~years (see Section 1.5.1), predictable spin-down behaviour is observed for many pulsars (Hobbs et al., 2011). This predictable spin-down behaviour will be the focus of most of this thesis, in particular the way this property of MSPs can be used to detect gravitational waves.

1.3 Gravitational Waves

Gravitational waves (GWs)[3] are one of the predictions of general relativity (GR; Einstein, 1916). A GW is a travelling perturbation in space-time, and its effect on a ring of freely-moving test particles with fixed coordinates is shown in Figure 1.4. The GW stretches space in one direction and simultaneously compresses it in the perpendicular direction (e.g., Shawhan & LIGO Scientific Collaboration, 2003). GWs exhibit two orthogonal polarisation modes - the '$+$' polarisation (shown in Figure 1.4) and the '$\times$' polarisation. The '$\times$' polarisation causes space-time deformations that are offset by an angle $\pi/4$ from those caused by a '$+$' polarised GW. The strength of a GW is generally defined by the strain h_c induced in a rod of length l as:

$$h_c = \Delta l / l \, , \tag{1.10}$$

where Δl is the maximum change in l induced by the GW over one period (for a periodic GW). GWs are emitted by any object undergoing acceleration whose motion is neither spherically nor cylindrically symmetric, such as any two objects orbiting their common centre of mass (Peters & Mathews, 1963).

Any system that emits GWs will lose energy via GW radiation (Einstein, 1918; Peters & Mathews, 1963; Phinney, 2001). The energy loss caused by GW emission has been indirectly inferred using observations of the binary pulsar PSR B1913+16 (Hulse & Taylor, 1975; Taylor & Weisberg, 1982). The detection was based on observation of the orbital decay of PSR B1913+16, measured by the cumulative advance of the periastron time for the orbit (see Figure 1.5). That is, the difference between consecutive times at which the pulsar is closest to its companion is decreasing. The decrease in the time taken to complete each orbit is consistent with GR and inconsistent with some other theories of gravitation (Will, 1977; Taylor & Weis-

[3] Note that "gravitational waves" are not to be confused with "gravity waves", which are waves for which the restoring force is gravity, e.g., water waves in the open ocean.

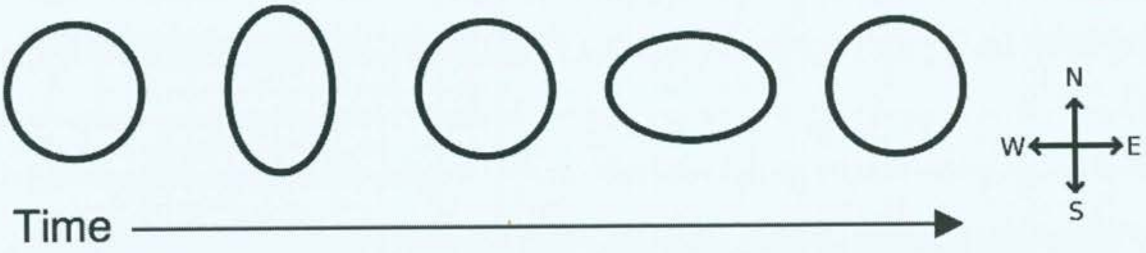

Figure 1.4: The effect of a GW with strength $h_c = 0.5$ on a ring of freely-moving test particles when the GW propagates perpendicularly to the plane of that ring, e.g., caused by a GW propagating into the page. The ring (1st image from left) is compressed East-West and expanded North-South (2nd image from left), then returned to its original state (3rd image from left), then compressed North-South and expanded East-West (4th image from left), then returned to its orignal state (5th image from left). If the GW is periodic with a period of $T_{\rm gw}$, then these images show the form of the ring at times t = 0, $T_{\rm gw}/4$, $T_{\rm gw}/2$, $3T_{\rm gw}/4$ and $T_{\rm gw}$. [Image reproduced from Shawhan & LIGO Scientific Collaboration (2003)]

berg, 1982). In Figure 1.5 we reproduce an updated version of the famous image from Taylor & Weisberg (1982), showing the remarkable level of agreement between the theoretical prediction of GR and the observed orbital shrinkage over 30 years of observations of PSR B1913+16[4]. As a result of such investigations, we are now all but sure of the existence of GWs.

1.4 Detecting Gravitational Waves

The results of Taylor & Weisberg (1982) and Weisberg et al. (2010) do not constitute a direct detection of GWs, as they have not explicitly detected the stretching and compressing of spacetime expected of a GW (see Figure 1.4). Direct detection of GWs will herald a new era in the study of astronomy and astrophysics.

GWs provide a unique way to study the Universe as they provide information about systems in a completely different way to EM waves. In particular, some regions of the Universe that are opaque to EM radiation - including, for example, the first 3×10^5 yr after the Big Bang (Sathyaprakash & Schutz, 2009) - may be observable using GW telescopes. Similarly, much recent work (e.g., Bloom et al., 2009; Sesana & Vecchio, 2010a,b; Corbin & Cornish, 2010) has outlined the benefits for detecting EM counterparts for GW sources and vice versa.

[4] A similar figure was presented by Weisberg et al. (2010).

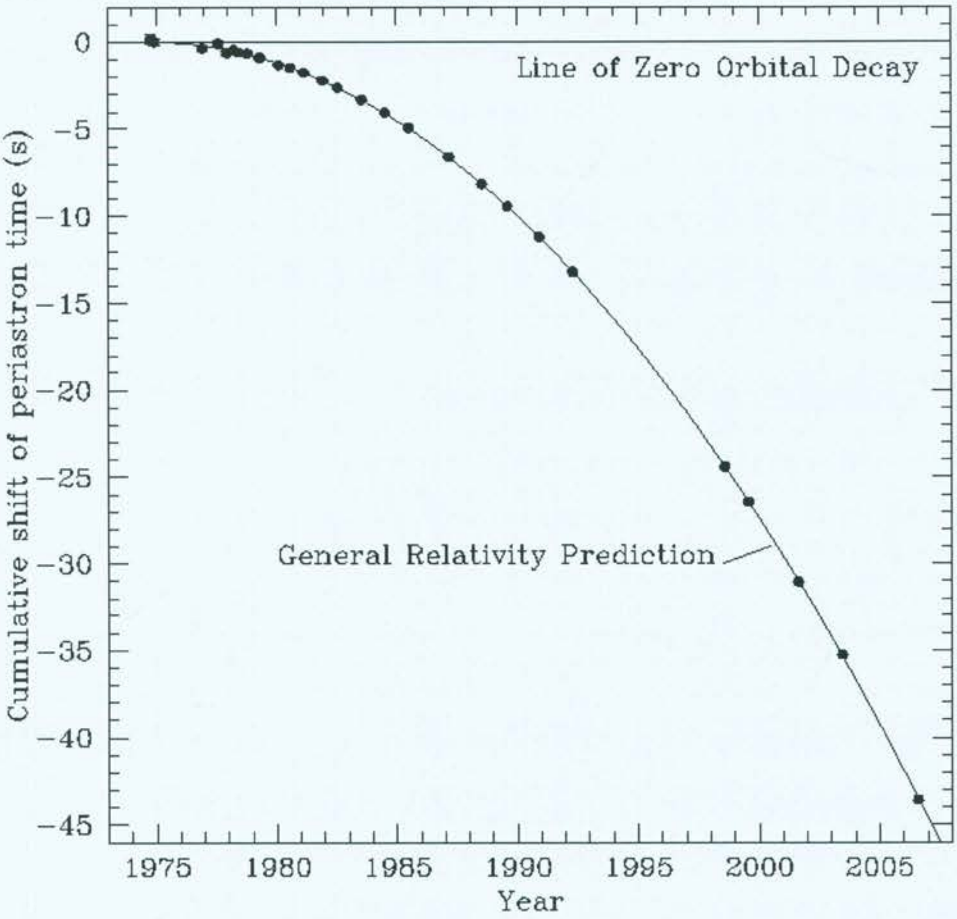

Figure 1.5: The shrinking of the orbit of the binary pulsar PSR B1913+16 (as measured by the decreasing time taken for each complete orbit of the pulsar) as a function of year number. The measurements of the cumulative advance in the periastron time (points) are remarkably consistent with the prediction of GR (parabolic line). If the system were not emitting GWs, its orbit would not be decaying and the points would follow the line of zero orbital decay (horizontal line). [Image credit: `http://www.people.carleton.edu/~jweisber/binarypulsar/B1913+16.gif`]

1.4.1 Current and Future Gravitational-Wave Detection Projects

The huge scientific gains expected from directly observing GWs has led to the establishment of many current global efforts to detect GWs. The Laser Interferometer Gravitational-Wave Observatory (LIGO; e.g., Abbott et al., 2009)[5], VIRGO (Acernese et al., 2006)[6] and The Australian International Gravitational Observatory (Barriga et al., 2010)[7] are some of the more well-known projects that aim to detect and analyse GW signals from astrophysical objects. Each of these projects aims to detect the delay of a locally-generated EM signal along one direction being correlated with the advance of a locally-generated EM signal in a perpendicular direction. For example, each of the widely-separated LIGO stations consists of two evacuated 4 km chambers at 90 degrees to each other. Laser signals are sent and received through these chambers with the objective of detecting a relative delay between the two. Such a detection could correspond to a detection of the compression of space-time in one direction occurring simultaneously with an expansion of space-time along a perpendicular direction, which is the expected action of GWs (see Figure 1.4). Measuring the delay and advance of light due to the action of GWs requires exquisitely precise measurement instruments. For example, LIGO is attempting to measure a path length difference of $\sim 10^{-18}$ m between its two perpendicular arms over 4 km (Shawhan & LIGO Scientific Collaboration, 2003). For comparison, the diameter of an atomic nucleus is $\sim 10^{-15}$ m (e.g., Pohl et al., 2010).

Typical GW sources for LIGO and other ground-based interferometers include the coalescence of compact binaries (containing white dwarfs, neutron stars or low-mass black holes), supernova explosions and nearby non-axisymmetric rotating neutron stars, whose rotation period and sky-position are known if they are detectable as pulsars (Shawhan & LIGO Scientific Collaboration, 2003). These sources all emit GWs with frequencies from ~Hz to ~kHz, corresponding to the frequency range over which LIGO is most sensitive. Advanced LIGO (Smith & LIGO Scientific Collaboration, 2009, and references therein) is an upgrade to LIGO that is currently being implemented and should improve the detector's sensitivity by two orders of magnitude while retaining a similar range of detectable frequencies.

A GW detection experiment in its planning stages is the proposed Laser Interferometer

[5] See `http://www.ligo.caltech.edu/advLIGO/`.

[6] See `http://www.virgo.infn.it/`.

[7] See `http://www.aigo.org.au/`.

Space Antenna (LISA; Larson et al., 2000)[8]. LISA will consist of three instruments in space forming three laser interferometers. For LISA, the evacuated cavities that are required for LIGO to function on Earth are replaced by the near-perfect vacuum of interplanetary space, which means that the length of these "cavities" can be much greater. The three LISA components are expected to be separated by $\sim 5\times 10^6$ km. LISA will be sensitive to the final inspiral and coalescence of binary black holes (with member masses in the range $10^3 M_\odot$ to $10^{10} M_\odot$) and a galactic foreground of neutron star and white dwarf binaries. These processes emit GWs with frequencies of $\sim\mu$Hz to ~mHz. The sensitivity ranges and likely sources for LIGO and LISA are summarised in Figure 1.6.

1.5 Detecting Gravitational Waves with Millisecond Pulsars

GWs can also be detected using the predictable rotation behaviour of MSPs and their beams of EM radiation. This was first suggested by Sazhin (1978) and Detweiler (1979). The likely GW sources whose signals may be detected with MSPs will emit in a different GW frequency range to sources for the other detection experiments mentioned above. MSPs can detect GWs in the frequency range ~nHz to $\sim\mu$Hz. Such GWs will be emitted by a range of sources, including:

- supermassive black-hole binaries (SMBHBs) at the cores of merged galaxies (Jaffe & Backer, 2003; Wyithe & Loeb, 2003; Jenet et al., 2004; Sesana et al., 2008);
- a network of cosmic superstrings (e.g., Damour & Vilenkin, 2005);
- relic GWs from the interaction of the large-scale dynamic cosmological metric with quantum instabilities in metric perturbations that existed in the early Universe (Grishchuk, 2005);
- the quantum chromodynamic (QCD) phase transition in the early Universe, when the Universe's temperature was ~100 MeV (Maggiore, 2000; Caprini et al., 2010, and references therein).

These sources are plotted in Figure 1.6, indicating their typical frequency ranges.

[8] See `http://lisa.nasa.gov/`.

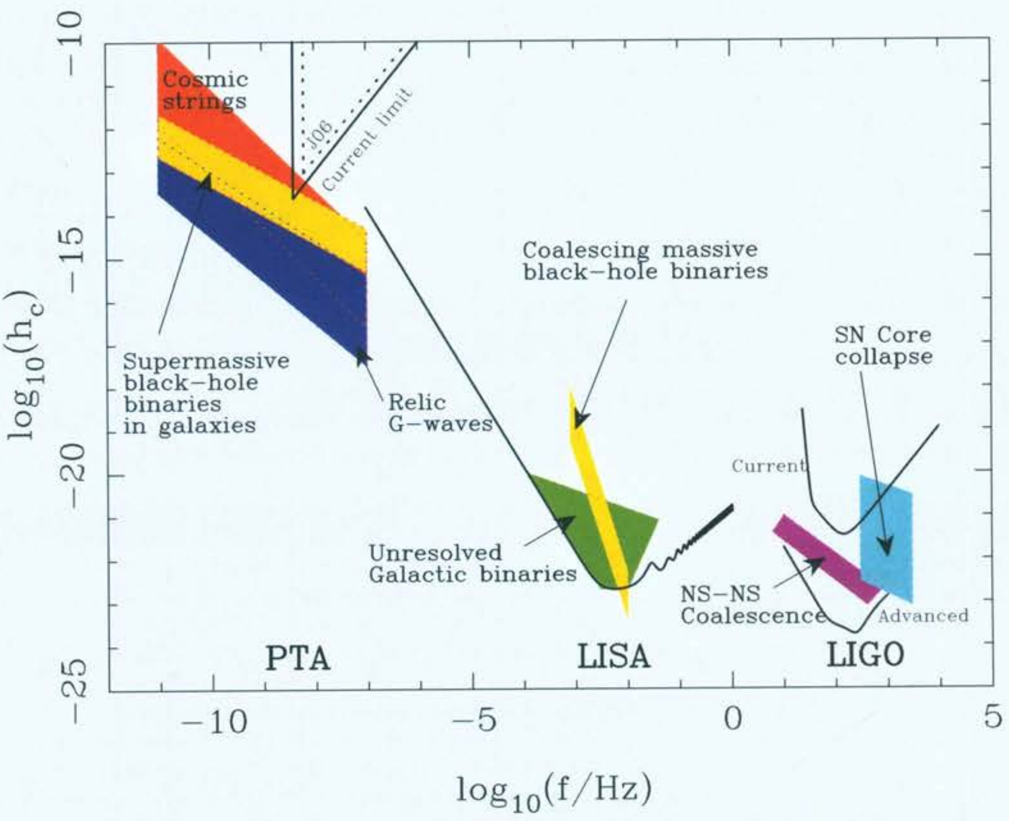

Figure 1.6: GW strain sensitivity as a function of frequency for GW analysis with pulsars ("PTA"), LISA and LIGO. The ordinates show the logarithm of the GW strain, h_c, defined in equations (1.10) and (1.11). In the nHz – μHz frequency range, we have listed sources of stochastic GWBs ("Cosmic strings", "Supermassive black-hole binaries in galaxies" and "Relic G-waves") and the limits that have been placed on their amplitude (see Section 1.6.2 for details; "J06" is the Jenet et al., 2006 limit; "Current limit" is the van Haasteren et al., 2011 limit). The wedge-shaped limits are derived under the approximation that the pulsar sensitivity to GWs is maximum at $f = 1/T_{\rm obs}$, where $T_{\rm obs}$ is the time-span of the observations. The pulsar sensitivity is assumed to be zero for lower frequencies and proportional to frequency for higher frequencies, as described in Equation (53) of Sesana et al. (2008). "Unresolved Galactic binaries" includes white-dwarf and neutron-star binaries. The region labelled "Coalescing massive black-hole binaries" shows the expected range of signals from the final inspiral of massive black-hole binary systems with member masses in the range $10^3 M_\odot$ to $10^{10} M_\odot$. The "Current" LIGO sensitivity shows the capabilities of existing datasets, while "Advanced" LIGO expects to improve GW sensitivity by two orders of magnitude. "SN [supernova] core collapse" and "NS-NS [neutron star] coalescence" are typical signals that LIGO expects to detect. [This image is based on figure 7 presented by Manchester (2010)]

Superstrings, relic GWs and the QCD phase transition all represent GW sources for which the resultant GW signal is believed to be independent of direction and stochastic[9]. Each source forms an isotropic stochastic gravitational-wave background (GWB). Very massive or nearby SMBHBs can be considered as individual sources of GWs (Lommen & Backer, 2001; Jenet et al., 2004; Sesana et al., 2009). Alternatively, the superposition of the GW signals from many SMBHBs throughout the Universe will form a GWB (e.g., Jaffe & Backer, 2003).

Detection of the GWs from an individual SMBHB will require very precise pulsar observations over a period of several years (Sesana et al., 2009). The combination of new observing systems and recently discovered pulsars means we are approaching the required level of precision (Demorest, 2011; Manchester, 2011). However, observations must continue at this level over at least five years to achieve the level of precision calculated by Sesana et al. (2009) as the minimum requirement for detection of a single source of GWs. On the other hand, some of the GWB sources provide a relatively large amplitude signal and may be detected or ruled out in the coming years (Jenet et al., 2006).

GWB sources can be described using many parametrisations. We will use the characteristic strain spectrum $h_c(f)$, which takes the following form for most GWBs:

$$h_c(f) = A(f/f_{1\,\mathrm{yr}})^{\alpha} \,, \tag{1.11}$$

where $A = h_c(f = f_{1\,\mathrm{yr}})$ is a dimensionless constant termed the "amplitude" of the GWB (see Equation 1.10), $f_{1\,\mathrm{yr}} = 1/(1\,\mathrm{yr})$ and α is a constant that satisfies $\alpha < 0$ for all expected backgrounds (Jenet et al., 2005). Another quantity often used to discuss GWB sources is the energy density of the GWB per unit logarithmic frequency interval, $\Omega_{\mathrm{gw}}(f)$ (adapted from Jenet et al., 2006):

$$\Omega_{\mathrm{gw}}(f)H_0^2 = \frac{2\pi^2}{3}h_c(f)^2 f^2 \,, \tag{1.12}$$

where H_0 is the Hubble constant. Typical values for A and α are in the ranges $10^{-17} < A < 10^{-14}$ and $-7/6 \leq \alpha \leq -2/3$ respectively.

However, the GWB due to merging and coalescing SMBHBs exhibits a slightly different form from that shown in Equation (1.11). It is now generally thought that the large elliptical galaxies seen in the present day have formed from the merging of smaller galaxies. Since most

[9] The stochastic signal consists of many GW emitters, each radiating GWs with a different amplitude, frequency and phase.

nearby galaxies appear to have solitary supermassive black holes (SMBHs) at their centre, this implies that the SMBHs at the core of each of the progenitor galaxies must coalesce in some way. It is unknown whether most of the mass transfer that takes place during coalescence is via accretion or via the merging of the two black holes (BHs). Many authors have considered the expected GWB that would result from hierarchical galaxy formation models (Phinney, 2001; Jaffe & Backer, 2003; Wyithe & Loeb, 2003; Enoki et al., 2004; Sesana et al., 2008). GR predicts that the characteristic strain spectrum has a spectral exponent of $\alpha = -2/3$ (Phinney, 2001), and most authors have concluded that the amplitude is in the range $10^{-15} \lesssim A \lesssim 10^{-14}$. However, a recent analysis based on Monte Carlo simulations of the population of SMBHBs was performed by Sesana et al. (2008), which showed that the discrete nature of the GW-emitting sources has a measurable effect on the GWB due to SMBHBs. This led to the derivation of a different form for $h_c(f)$ (Sesana et al., 2008):

$$h_c(f) = h_0 \left(\frac{f}{f_0}\right)^{-2/3} \left(1 + \frac{f}{f_0}\right)^{\gamma_0} \tag{1.13}$$

with the following ranges for the variables γ_0, f_0 and h_0:

- $\gamma_0 \;:\; -1.04 < \gamma_0 < -1.11$
- $f_0 \;:\; 1.4 \times 10^{-8}\,\mathrm{Hz} < f_0 < 5.3 \times 10^{-8}\,\mathrm{Hz}$
- $h_0 \;:\; 0.65 \times 10^{-15} < h_0 < 2.15 \times 10^{-15}$

from the results of the four models of SMBHB assembly considered in their paper. These ranges imply that the predicted range of $A = h_c(f = f_{1\,\mathrm{yr}})$ is

$$10^{-16} < A < 3 \times 10^{-15}\,.$$

In Figure 1.7, we have reproduced a figure from Sesana et al. (2008) that shows their prediction for the characteristic strain spectrum of the GWB and the uncertainty in that prediction.

The obvious next question is "*How can we use MSPs to study such sources?*" We choose to study GWs with MSPs as the detector by using a technique called "pulsar timing".

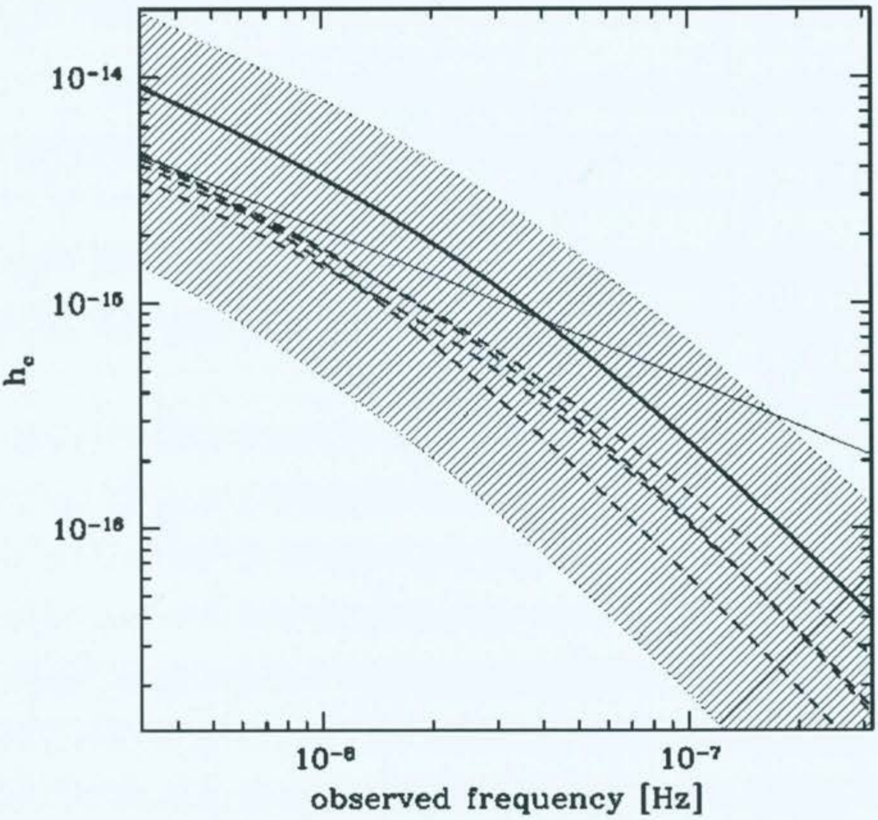

Figure 1.7: The characteristic strain spectrum of the GWB as a function of GW frequency as calculated by Sesana et al. (2008). This spectrum is determined by the parameters of the SMBHB population, which is determined by the evolution of SMBHs. Four recent models of SMBH evolution are compared (dashed lines; see Volonteri et al., 2003; Koushiappas et al., 2004; Begelman et al., 2006; Volonteri et al., 2006), along with the average signal calculated by Sesana et al. (2008) (thick solid line) and its 1-σ uncertainty (hatched region). The signal resulting from the standard GR assumption of a spectral exponent of $\alpha = -2/3$ for the characteristic strain spectrum is also shown (thin solid line; see Equation 1.11), calculated using the SMBHB assembly model discussed by Volonteri et al. (2003). [Image reproduced from Sesana et al. (2008)]

1.5.1 Millisecond Pulsar Timing

The aim of a pulsar timing observation is to measure a precise and accurate time-of-arrival (ToA) for a pulse of EM radiation from a pulsar. Mean pulse profiles are very stable (e.g., Lorimer & Kramer, 2005), which means precise and unbiased ToAs can be obtained over many years.

To perform the most precise pulsar timing, a range of effects must be taken into account during each observation. The dispersive effects of the ISM must be corrected using de-dispersion (Sections 1.2.1 and 1.2.2). The mean pulse profile is then formed by folding the de-dispersed pulses (Section 1.2.3). The observation will typically have a duration of $\sim$1 hr for precision MSP timing, meaning that the data contains $\sim 10^6$ pulses for a typical MSP.

The observed pulse profile is compared with the standard template for the MSP by measuring the time shift that gives maximum cross-correlation between the two, following the method described by Taylor (1992). This measurement results in an estimate of a ToA (that is, the arrival time of the pulse at the radio telescope) and its uncertainty.

The ToA is then corrected using a chain of clock corrections, whereby the observatory timescale is first referenced to Universal Coordinated Time (UTC) and then to Terrestrial Time as realised by International Atomic Time, abbreviated to TT(TAI). This corrected ToA is then transformed to the arrival time at the solar system barycentre using a solar system ephemeris (see Standish, 2004). This ephemeris includes, amongst other things, the relativistic time transformations between the Earth and the solar system barycentre, and the masses and velocities of each planet and many major dwarf planets and asteroids. These transformations provide a barycentric ToA.

For high precision timing, pulsars are usually observed many times per year over at least a few years. The ToAs obtained can then be used to improve the model for the pulsar's timing behaviour that is used to fold the incoming pulses as described earlier in this Section. For example, the estimate of the pulsar's sky-position can be improved after timing observations have been carried out for a year or more. Timing models will be discussed in greater detail in Chapter 2.

By subtracting the arrival time predicted by the pulsar timing model from the observed arrival time, we obtain a "timing residual" for that particular observation. Timing residuals are influenced by noise, but also contain a wealth of information about the telescope hardware and processing systems, or about planetary, solar system, extra-galactic and cosmological physics. In particular, MSP timing residuals may contain the signals induced by GWs. Hence, MSPs may be used to study GW signals from astrophysical and cosmological sources by examining the timing residuals. Many authors have developed strategies for using MSP timing residuals to study GWs (Romani & Taylor, 1983; Kaspi et al., 1994; Jenet et al., 2005, 2006; van Haasteren et al., 2009; Anholm et al., 2009; Burt et al., 2011), and we outline the present state of the field in the next Section.

1.6 Techniques for Studying Gravitational Waves with Pulsar Timing

1.6.1 Detecting Gravitational Waves with Pulsar Timing

After the pioneering work of Sazhin (1978) and Detweiler (1979), the foundation for detection of a GWB with pulsars was laid down by Hellings & Downs (1983). It is now widely accepted that a background of low-frequency GWs that is described by Equation (1.11) causes ToA perturbations with power spectrum, P_g, given by: (Detweiler, 1979; Jenet et al., 2005, 2006)

$$P_g(f) = \frac{A^2}{12\pi^2}\left(\frac{f}{f_{1\,\mathrm{yr}}}\right)^{2\alpha-3}, \tag{1.14}$$

where $f_{1\,\mathrm{yr}} = 1/(1\,\mathrm{yr})$. These ToA perturbations (regardless of the source) are also correlated between pairs of pulsars in a quadrupolar fashion. This correlation, which depends only on the angle between the pair of pulsars as shown in Figure 1.8 (Hellings & Downs, 1983), provides an unambiguous signature of the GWB. The functional form of this signature is given by:

$$\zeta(\theta_{ij}) = \frac{3}{2}x\log x - \frac{x}{4} + \frac{1}{2}, \tag{1.15}$$

where $x = [1 - \cos(\theta_{ij})]/2$ and θ_{ij} is the angle between pulsars i and j subtended at the observer (Hellings & Downs, 1983; Jenet et al., 2005)[10]. The function $\zeta(\theta_{ij})$ is independent of GW frequency, and is derived assuming the GW polarisation modes are as described by GR; other GW modes are analysed by Lee et al. (2008) but are not considered in this thesis.

[10] The right-hand-side of Equation (1.15) is a factor of $3/2$ larger than the original result of Hellings & Downs (1983), but identical to the equation given by Jenet et al. (2005). This is because Hellings & Downs (1983) correlated GWB-induced Doppler shifts in pulse ToAs, whereas Jenet et al. (2005) correlated GWB-induced timing residuals (the integral of the Doppler shift).

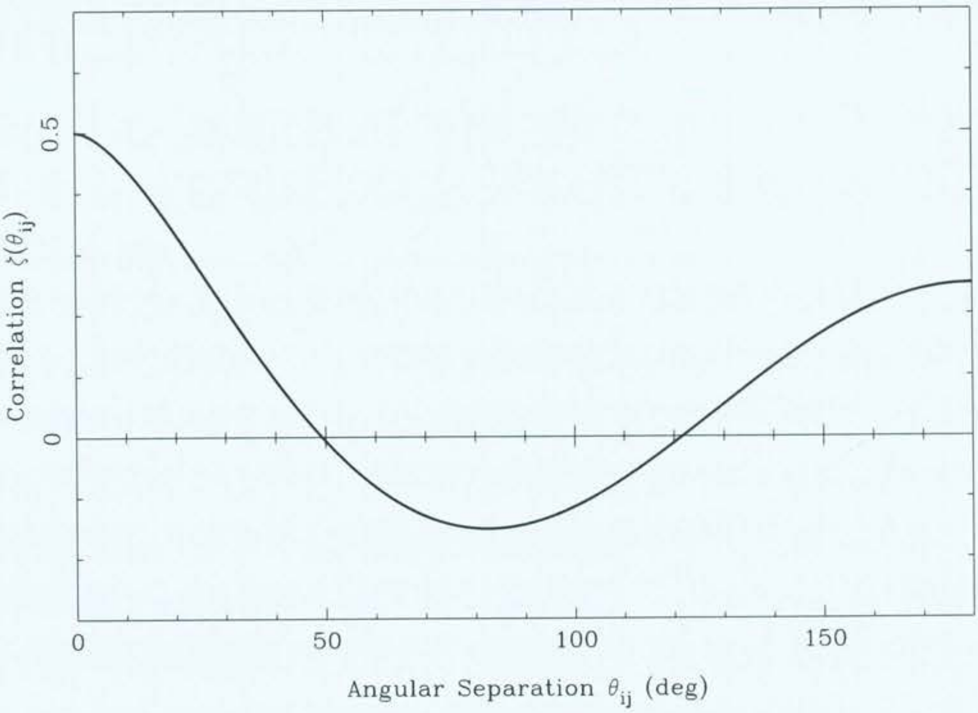

Figure 1.8: The expected correlation in pulsar timing residuals due to an isotropic stochastic GWB. The abscissa gives the angle subtended at the observer by a particular pulsar pair. The ordinate gives the expected correlation (normalised to a value between -1 and 1) between the timing residuals of that pair. This signal is independent of the GW frequency and assumes that GWs behave as predicted by GR.

Detection of such a correlated signal requires timing residuals with overlapping time-spans from many MSPs in order to sample the curve shown in Figure 1.8 with sufficient density to claim a detection of the GWB. This goal can be achieved using a "pulsar timing array" (PTA). PTAs are able not only to detect GWs and the GWB, but also can detect errors in TT(TAI) and errors in the solar system ephemeris (Foster & Backer, 1990).

The first complete account of a method for detecting the correlated signal induced in timing residuals by the GWB, and thus directly detecting the GWB using a PTA, was given by Jenet et al. (2005). Their method involves the calculation of the pairwise correlations between the time series of residuals for each pulsar in the PTA. For an array of $N_{\rm psr}$ pulsars, this process provides $N_{\rm psr}(N_{\rm psr}-1)/2$ measured correlations. A detection of the GWB would then be possible by calculating the correlation between the expected GWB signal shown in Figure 1.8 and the observed pairwise correlations. Jenet et al. (2005) label the value of this correlation as ρ and define the "significance" of the detection as

$$S = \rho/\sigma_\rho \, , \tag{1.16}$$

where $\sigma_\rho^2 = 2/N_{\rm psr}(N_{\rm psr}-1)$ (Jenet et al., 2005). However, this detection scheme is sub-optimal; for instance, pulsars with different amounts of noise in their timing residuals contribute equally to this detection statistic. While Jenet et al. (2005) did consider the effect of analysing residuals with different noise levels, each time series of residuals is given an equal weight in calculating S. There was also no treatment of other issues associated with analysing real pulsar timing residuals, such as the non-simultaneous sampling of the observations, the highly variable ToA error bars, large variation in the time-span of different time series and the issue of non-overlapping observations.

Van Haasteren et al. (2009) present a Bayesian technique for detecting the GWB that improves on the Jenet et al. (2005) technique. The technique assumes that the GWB has the form given in Equation (1.11), but that neither A nor α are known. A joint distribution in these two variables can then be calculated. This technique was applied recently to observations of six pulsars from the EPTA as described in van Haasteren et al. (2011). This paper also includes the detection of a simulated GWB signal that has been artificially added to their data set. However, the GWB detection problem includes many aspects that are difficult to solve exactly and approximate solutions may only be testable with Monte Carlo simulation. Tests based on Monte

Carlo simulation have not yet been performed with the van Haasteren et al. (2009) technique because these tests would require a very large amount of computation time.

A detection technique was also presented by Anholm et al. (2009). Their method is a frequency-domain analysis based on the methods of the laser interferometer community. However, their derivation does not include the effects of the fitting of the pulsar timing model to the observed ToAs. This fit severely attenuates the GWB signal that we aim to detect and complicates the analysis of the timing residuals. The effect of the timing model fit on GWB estimation is discussed at length in Chapter 6 (also published as Yardley et al., 2011a).

The detection of a GWB requires a 100 ns root-mean-square (rms) timing residual on at least 20 pulsars over a period of at least five years (Jenet et al., 2005). Using extensive observations of 20 MSPs over several years (see, e.g., table 1 of Manchester, 2011), it is apparent that some MSPs can be timed more precisely than others because of narrow pulse profiles or greater flux density. This means the reference PTA data set, which consists of observations of 20 pulsars with 100 ns rms residual timed over five years, will be very difficult to obtain with current observation systems and processing algorithms. However, the prospects for future GW detection with pulsars are improved by the fact that the GWB signal from all expected sources has $\alpha \leq -2/3$ in Equation (1.14). This means that the strength of the expected GWB signal increases with observing time-span, $T_{\rm obs}$, at least as fast as $T_{\rm obs}^{13/3}$; for example, doubling the observed data-span increases the expected signal by a factor of 20. Furthermore, recent work by Sesana et al. (2009) suggests that it may be possible to directly detect a single source of GWs with a few very precisely timed pulsars, despite the fact that the GWB signal is expected to be stronger on average than any individual source. This has led to a flurry of recent interest in detecting single sources of GWs with pulsars (e.g., Corbin & Cornish, 2010; Sesana & Vecchio, 2010a; van Haasteren & Levin, 2010; Sesana & Vecchio, 2010b; Pshirkov et al., 2010; Burt et al., 2011).

These works have considered detection of sinusoidal signals induced by GWs in MSP timing residuals, as well as GW "burst" signals caused by SMBHB coalescence or the periastron approaches of SMBHBs with highly elliptical orbits. In principle, the algorithm of van Haasteren & Levin (2010) is suitable for the study of any GW signal for which the GW waveform has known functional form. Many of the techniques for single source detection proposed in the last five years are yet to be applied to real pulsar timing observations, with the exception of the technique presented in Chapter 4 of this thesis, which is applied to timing residuals from a PTA observed with the Parkes radio telescope (see Section 2.1).

However, the noise levels on many current PTA observations are too high to allow detection of GW signals at the expected levels (e.g., the levels given in Sesana et al., 2008, 2009). It is likely that the GWB will be detected within the next decade (Demorest, 2007; Hobbs et al., 2010a), while a detection of an individual source of GWs depends on the location of the source relative to precisely timed pulsars (Burt et al., 2011). In the meantime, a wealth of astrophysical information can be gleaned by placing limits on the expected amplitude of GW signals. These limits have been calculated using the timing of individual pulsars, as well as the timing of multiple pulsars[11].

1.6.2 Finding Upper Limits on Gravitational Waves with Pulsar Timing

Historically, most authors have focussed on finding upper bounds on parameters of the GWB, whereas interest in constraining the properties of individual sources of GWs has been relatively recent. Romani & Taylor (1983) used the timing residuals of a 1.3 s-period pulsar, PSR B1237+25, to constrain $\Omega_{\rm gw}(f)$ at $f \sim 10^{-8}$ Hz, and thus demonstrate that such GWs do not dominate the energy density of the Universe. While Stinebring et al. (1990) observed two pulsars (PSRs J1857+0943 and J1939+2134), their technique was only used to provide an upper bound on $\Omega_{\rm gw} H_0^2$ using the timing residuals of each pulsar individually. Kaspi et al. (1994) then used a similar technique to provide an upper bound on $\Omega_{\rm gw} H_0^2$ using similar observations with a longer time-span. They made the important step of combining the data from their two pulsars to find the best constraint on the GWB amplitude. However, the statistical method employed by Kaspi et al. (1994) has been criticised by other authors (Thorsett & Dewey, 1996).

The technique of Kaspi et al. (1994) was modified by Jenet et al. (2006) in the wake of such criticism to provide statistically rigorous constraints on $\Omega_{\rm gw} H_0^2$ and also on $h_c(f)$. Jenet et al. considered a range of different GWB sources and found a new limit on the parameters of each source. In particular, their limit of

$$h_c(f = f_{1\,\rm yr}) \leq 1.1 \times 10^{-14} \,, \tag{1.17}$$

[11] A measurement of a binary pulsar's orbital period and the error in the measurement of the rate of change of orbital period can constrain the GWB in the frequency range 10^{-12} Hz$< f < 10^{-9}$ Hz (Bertotti et al., 1983; Kopeikin, 1997). This thesis does not discuss these methods.

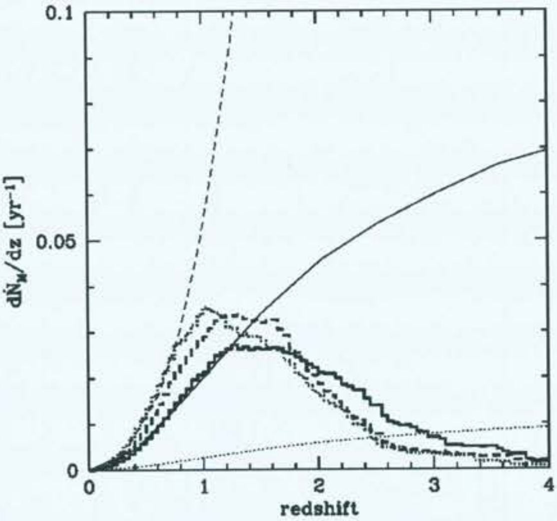

Figure 1.9: MBHB coalescence rates and galaxy merger rates as a function of redshift. The galaxy merger models assumed by Jaffe & Backer (2003) (thin solid line) and Rajagopal & Romani (1995) (thin dashed line) are inconsistent with recent simulations of the coalescence rates for SMBHBs (histograms). Using the model of Jaffe & Backer (2003) in the Monte Carlo simulations of Sesana et al. (2008) yields results that are inconsistent with recent models of the evolution with redshift of the SMBHB coalescence rate (thin dotted line; Sesana et al., 2008). [Image reproduced from Sesana et al. (2008)]

as the 95%-confidence upper bound on a GWB with spectral exponent of $\alpha = -2/3$ constrains the galaxy merger rate evolution with redshift (Jenet et al., 2006). Jaffe & Backer (2003) and Wen et al. (2011) parametrised the merger rate of galaxies $R(z)$ such that $R(z)$ goes as $(1+z)^{\gamma}$, where γ is now thought to be in the range $-1 < \gamma < 3$ (Carlberg et al., 2000; Patton et al., 2002; Lin et al., 2004; Kartaltepe et al., 2007; Lin et al., 2008). The limit given in Equation (1.17) constrains γ to be less than 2.6, but only if we allow the formation of SMBHs at very high redshifts near $z = 100$ (see Wen et al., 2011). Furthermore, figure 12 of Sesana et al. (2008) (reproduced in Figure 1.9) suggests that the SMBH coalescence rate is not a simple power-law with redshift, and that the rate decreases at redshifts greater than $z = 2$ for most current models.

MSP timing is yet to provide constraints, via upper limits on the expected GWB signal, that rule out the most recently proposed models of SMBHB evolution (see Sesana et al., 2008). A very recent upper bound on the GWB amplitude of $h_c(f = f_{1\,\mathrm{yr}}) \leq 6 \times 10^{-15}$ for $\alpha = -2/3$ (van Haasteren et al., 2011) does constrain the parameters of the GWB model of Wyithe & Loeb (2003), but not the currently accepted predictions for the GWB amplitude of Sesana et al. (2008). However, significant progress has been made in constraining the parameter space of

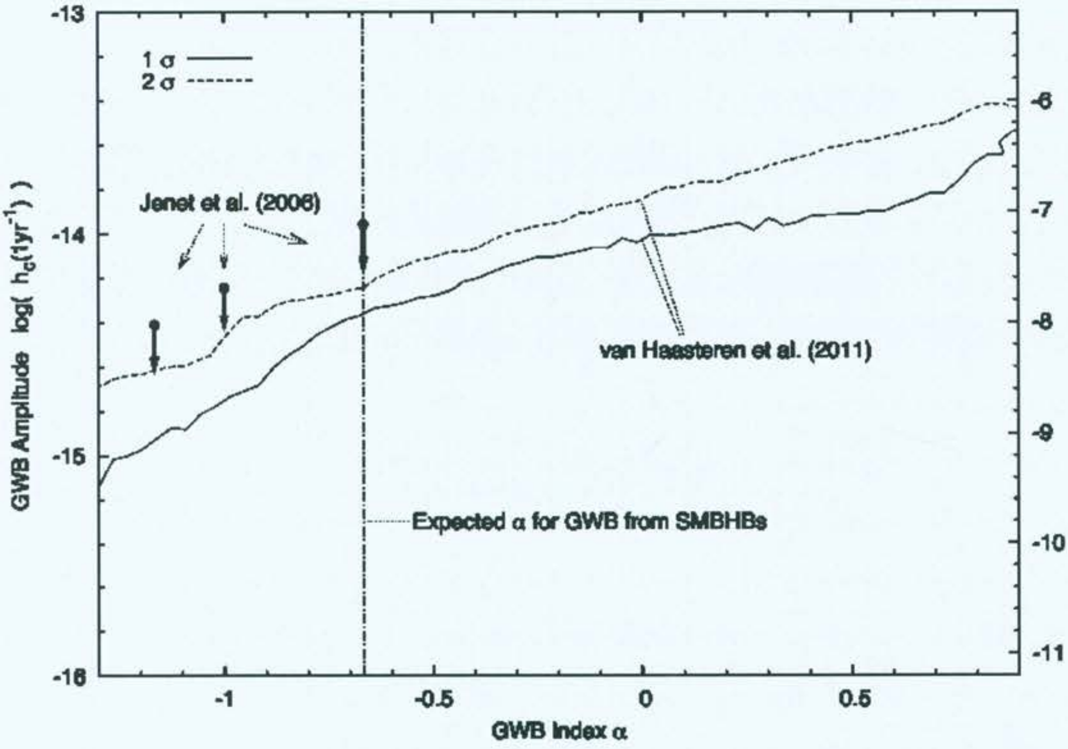

Figure 1.10: An upper bound on the GWB amplitude $h_c(f = f_{1\,\mathrm{yr}})$ as a function of α. The ordinates are measured in $\log(h_c(f = f_{1\,\mathrm{yr}}))$ (the axis on the left-hand side) and in $\log(\Omega_{\mathrm{gw}}(f = f_{1\mathrm{yr}})h_0^2)$, where $h_0 = H_0/100$ (the axis on the right-hand side). The 68%-confidence upper bound (solid line) and the 95%-confidence upper bound (dashed line) from van Haasteren et al. (2011) are more constraining than the upper bounds published by Jenet et al. (2006) (dots). The van Haasteren et al. (2011) upper bound is the first that can be calculated as a continuous function of α. For a GWB caused by SMBHBs, it is expected that $\alpha = -2/3$ at the most sensitive frequencies of current pulsar timing experiments (the value indicated by the vertical dot-dashed line). [Image reproduced from van Haasteren et al. (2011)]

cosmic superstring models (Jenet et al., 2006; Ölmez et al., 2010; van Haasteren et al., 2011). In Figure 1.10, we reproduce a figure from van Haasteren et al. (2011) that shows the GWB upper bounds for different values of α.

With regard to individual GW source limits, MSP timing has been used to provide useful constraints on the parameters of proposed SMBHBs. Lommen & Backer (2001) used ~1000 days of observations on three MSPs (PSRs J1713+0747, J1857+0943, J1939+2134) to constrain the properties of a range of nearby massive dark objects, if any of these objects harboured a black-hole binary. Jenet et al. (2004) ruled out a proposed SMBHB at the core of the radio galaxy 3C66B (Sudou et al., 2003). Using a straightforward periodogram analysis and seven years of publicly-available pulsar timing observations of PSR B1855+09, Jenet et al. (2004) showed that the SMBHB proposed by Sudou et al. (2003) is ruled out with 95% confidence by these observations. They went on to show that they could constrain the mass ratio of the two BHs and / or the orbital eccentricity of any SMBHB in 3C66B with very high confidence.

However, with the exception of the techniques presented by van Haasteren et al. (2009) and in Chapters 4 and 6, many techniques have not been able to adequately account for all the aspects of real pulsar timing observations. These include, but are not limited to, non-white noise sources affecting the timing residuals, the irregularly and non-simultaneously sampled observations of a PTA, and the effects of fitting the timing model to the observed ToAs.

The aim of this thesis is to develop, implement and demonstrate techniques for studying GWs with PTAs that can be applied immediately to almost all real pulsar timing observations. The only data sets to which these techniques may not be applied in their present form are those with time series that exhibit a very steeply-decreasing power spectrum with power-law exponent less than -2, as is seen in many young pulsars and a few MSPs (e.g., PSR J1939+2134; Hobbs et al., 2010b). This is not a significant drawback of the techniques presented because any time series that exhibits such a steeply-sloping spectrum is unlikely to be useful for GW detection. It is also not difficult to augment these techniques using a new method of spectral analysis appropriate for steeply-decreasing power spectra (Coles et al., 2011).

1.7 Thesis Structure

The remainder of this thesis is structured as follows:

Chapter 2:

We show some examples of real pulsar timing data sets from one of the most prominent PTA projects, the Parkes Pulsar Timing Array (PPTA). We describe methods of simulating ToAs and GW signals using "plugins" to the software package TEMPO2. We demonstrate that these simulated observations do resemble real observations from a radio telescope.

Chapter 3:

We develop a technique for detecting GW signals from individual SMBHBs that induce a correlated sinusoidal signal in timing residuals that are otherwise uncorrelated. We test the technique on simulations of a range of possible future PTA observations. We analyse a set of uncorrelated timing residuals observed at the Arecibo and Parkes telescopes to determine their sensitivity to GW sources that induce sinusoidal signals in the residuals. We use these sensitivity calculations to constrain the coalescence rate of SMBHBs as a function of their mass and redshift using a technique developed by Wen et al. (2011). We also describe the limitations of this technique.

Chapter 4:

We improve the technique of Chapter 3 such that we can process the most recently published data from the PPTA, as published by Verbiest et al. (2009). This leads to a measurement of the sensitivity of a PTA to sinusoidal GW sources. We calculate the first realistic GW-sensitivity curve for a PTA that can be compared with LIGO and LISA GW-sensitivity curves. The PTA sensitivity curve includes GW frequencies from 2 nHz to 400 nHz and is calculated for current observations and future predictions. We calculate the constraint on the coalescence rate of SMBHBs for the PPTA data set presented by Verbiest et al. (2008, 2009) using the Wen et al. (2011) technique.

Chapter 5:

We transition from treating individual GW sources to treating the incoherent sum of all GW sources, which forms the isotropic stochastic GWB. We describe the software implemented in TEMPO2 for simulating GWB signals and their effect on timing observations of pulsars. We use TEMPO2 to calculate a limit on the amplitude of the GWB for a variety of data sets using the method of Jenet et al. (2006). Each limit leads to a constraint on the coalescence rate of SMBHBs using the Wen et al. (2011) technique.

Chapter 6:

We introduce a new technique that can detect the expected steep power-law GWB signal in pulsar timing residuals. We demonstrate that no GWB signal has been detected in the PPTA residuals to date, but find previously unpublished effects in the GWB detection process that must be accounted for. In particular, the full effect of the timing model fit that produces the timing residuals must be included in the analysis.

Chapter 7:

We conclude the thesis with some suggestions and predictions for future directions of GWB detection with pulsars. In particular, we briefly discuss the importance of the International

Pulsar Timing Array collaboration.

Chapter 2

Real and Simulated Data Sets

Chapter Outline: *In this Chapter, we:*

- *review current PTA projects across the world and the properties of their data sets.*
- *describe two published data sets from the Parkes Pulsar Timing Array that have been the focus of our GW analysis and will be used in later Chapters.*
- *describe methods for simulation of realistic timing residuals.*
- *describe simulations of GW signals in timing observations.*
- *give examples of simulated data sets for current and future observing programs.*

Chapter 1 listed numerous techniques that have been proposed for estimating the amplitude of GW signals in pulsar timing residuals (Sections 1.5 and 1.6). However, most of these methods cannot be applied directly to recent observations because of, for example, the sampling of the observations or the presence of non-white noise in the residuals. In Chapters 3 – 6, we will introduce new GW-analysis techniques that have been applied to observations from the Parkes Pulsar Timing Array. These methods can be applied to almost any set of pulsar timing observations. To develop algorithms that can be applied to real observations, it is necessary to parameterise the effects that must be accounted for. In this Chapter, we review the current PTA projects and describe the data sets that they are producing. We give a detailed description of data sets that will be analysed in later Chapters. We describe and implement methods for simulating pulsar timing observations and derive and implement the effects of GWs on the pulse ToAs. Finally, we describe the properties of simulated pulsar timing data sets that will be analysed in Chapters 3 and 5.

2.1 Current Pulsar Timing Array Projects

Several pulsar timing research groups around the world are carrying out PTA observing projects. At the time of writing, these are:

- The "Parkes Pulsar Timing Array" (PPTA; Manchester, 2008; Verbiest et al., 2010, and references therein)[12] collaboration, which uses the 64-m diameter Parkes radio telescope. The collaboration aims to time 20 MSPs over a period of at least 5 years. The majority of the pulsars are already yielding a weighted rms residual below 1 μs, with the rms of a few pulsars below 200 ns (Manchester, 2010). The project has been ongoing since late 2004, although a subset of the PPTA pulsars have been timed at the Parkes observatory since 1994, albeit with less regularity and precision (Verbiest et al., 2008, 2009). The PPTA data sets are the focus of the GW analysis presented later in this thesis. Details of these data sets are given below in Sections 2.2.3 – 2.2.5.

- The "European Pulsar Timing Array" (EPTA; e.g., Stappers et al., 2006; Ferdman et al., 2010)[13] collaboration, which currently observes MSPs using four large radio telescopes. These are the 100-m diameter Effelsberg, the 76-m diameter Lovell, the 94-m diameter-equivalent Nançay, and the 96-m diameter-equivalent Westerbork synthesis radio telescopes. The EPTA collaboration also intends to observe pulsars with the 64-m diameter Sardinia radio telescope, which is expected to become operational in 2011 (Tofani et al., 2008). The combination of these telescopes provides observations at a wide range of frequencies from 0.12 GHz to 95.5 GHz with bandwidths ranging from ~100 MHz to 1 GHz. The EPTA collaboration currently times 24 MSPs (Ferdman et al., 2010; Hobbs et al., 2010a).

- The "North American Nanohertz Observatory for Gravitational Waves" (NANOGrav; Jenet et al., 2009)[14] collaboration, which observes MSPs using the 100-m diameter Green Bank Telescope and the 300-m diameter telescope at the Arecibo Observatory. Pulsars are observed at Green Bank at 820 and 1400 MHz, and observed at Arecibo at 327, 430, 1400 and 2300 MHz (Demorest, 2011)[15]. Sources have been observed using coherent dedispersion systems with a 64 MHz bandwidth, though recent upgrades allow a coherently de-dispersed observing bandwidth of up to 800 MHz (Ransom et al., 2009). They are currently timing 20 MSPs (Nice et al., 2011), though new pulsars are being added to

[12] `http://www.atnf.csiro.au/research/pulsar/ppta/`.

[13] `http://www.epta.eu.org/`.

[14] `http://nanograv.org/`.

[15] See also: `http://science.nrao.edu/newscience/9-Wed/17-Demorest/demorest_santa_fe_2011.pdf`.

the source list.

- A PTA at Kalyazin Observatory, Russia, described by Ilyasov et al. (2004b), is observed using the 64-m diameter Kalyazin telescope. This telescope typically times pulsars at frequencies of 600 and 1400 MHz with a bandwidth of 3.2 MHz (Ilyasov, 2006). Kalyazin observatory has performed timing observations on an array of seven MSPs since 1996 (Ilyasov & Oreshko, 2007).

Other collaborations and countries, such as China, may soon commence MSP timing observations that could lead to their own PTA (Nan, 2008; Smits et al., 2009; Nan, 2009). Also, very-low-frequency observations of pulsars using India's Giant Meterwave Radio Telescope may facilitate even more precise timing observations (Jenet et al., 2009).

A new global PTA collaboration is emerging, the International Pulsar Timing Array (IPTA), which is currently a combination of the PPTA, EPTA and NANOGrav (Hobbs et al., 2010a). The current array of pulsars observed as part of the IPTA is shown in Figure 2.1. The IPTA will provide the most sensitive data sets to date for GW detection via pulsar timing. Even though a few experiments have already used shared data (e.g., Champion et al., 2010), more extensive collaboration and data-sharing agreements have not yet been finalised.

2.2 High-Precision Pulsar Timing at Parkes

Throughout this thesis we will focus on the analysis of data from the PPTA to determine its sensitivity to GWs and other signals that are expected to produce a correlated signal in the timing residuals. High-precision timing of pulsars for the PPTA project has been conducted under two long-term observing proposals: P140 and P456. Details of both projects are given below.

2.2.1 P140: "Precision Pulsar Timing"

The first project that carried out repeated observations of an array of MSPs at the Parkes telescope commenced in 1994, entitled "P140: Precision Pulsar Timing". These observations were conducted during the same period of time as the Parkes 70 cm survey for MSPs (Bailes et al., 1994; Lyne et al., 1998) with the original intention of improving the timing models for pulsars that were newly discovered as part of the survey. The project also observed two of the bright

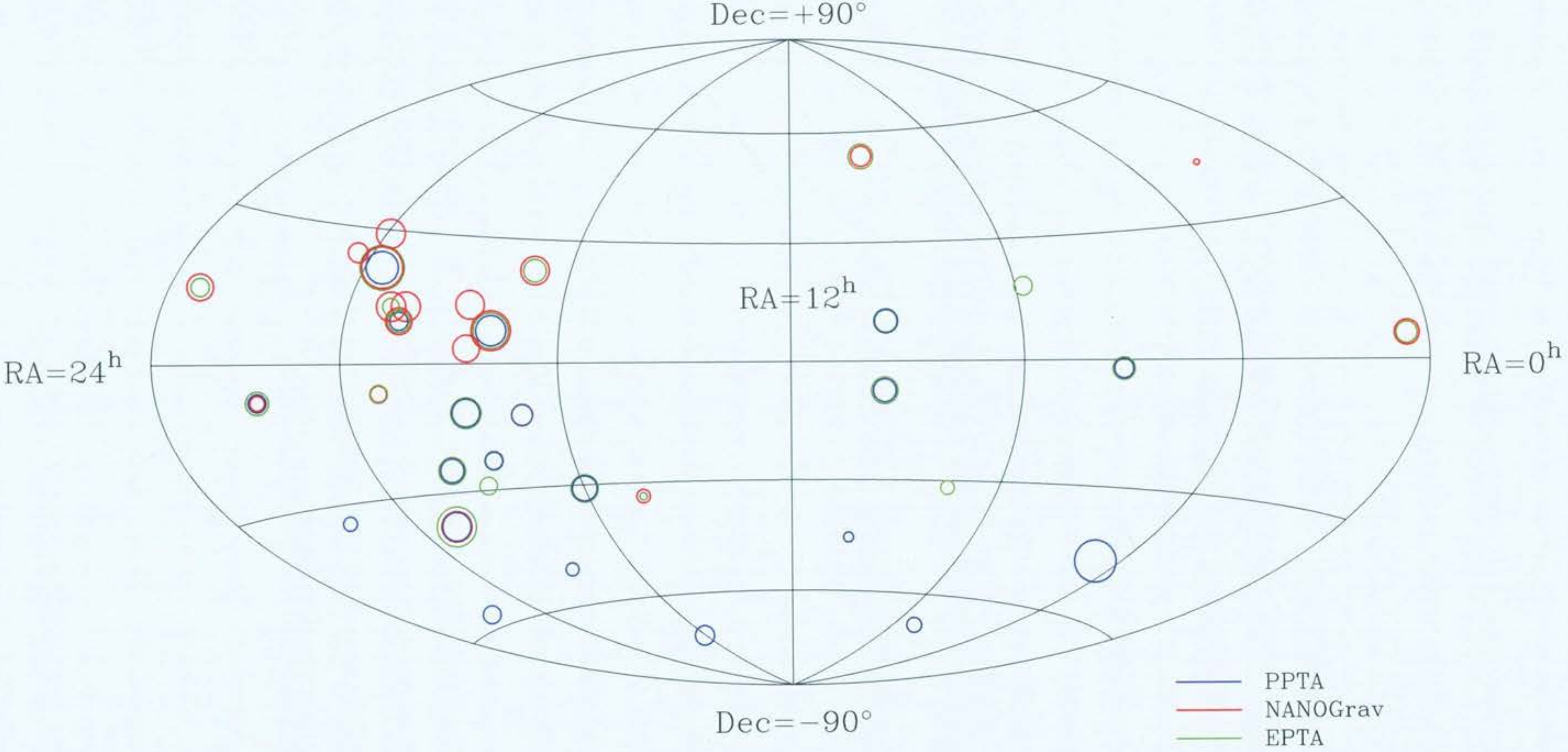

Figure 2.1: The position and typical ToA uncertainty for the IPTA pulsars (as listed in table 1 of Hobbs et al., 2010a). The figure shows pulsars from the PPTA (blue), EPTA (green) and NANOGrav (red) projects described in Section 2.1. The centre of each circle indicates the location of the pulsar in right ascension (RA) and declination (Dec). The radius of the circle, $r_{\rm circ}$, is related to the size of the typical error bar, $\sigma_{\rm typ}$, on a timing observation of that pulsar for that project via $r_{\rm circ} = 0.011(1 - \log(0.25\sigma_{\rm typ}))$. Hence, a larger circle on the above plot indicates that a ToA from the pulsar has a smaller typical error bar. The labelling of RA is non-standard, such that $\rm RA = 12^h$ is the central meridian.

MSPs discovered earlier at Arecibo, PSRs J1713+0747 and J1939+2134. With the discovery of new pulsars in the Swinburne intermediate-latitude survey (Edwards et al., 2001), the list of pulsars that were being observed under the project was extended to 16 MSPs by 2006. The P140 project ended in 2011. Highlights from the P140 project include:

- the measurement of the 3-dimensional binary orbital geometry of PSR J0437−4715, which led to a verification of the space-time distortion near its companion via detection of the Shapiro delay predicted by GR (van Straten et al., 2001)[16];
- the measurement of the mass of the binary companion of PSR J1909−3744 by observing its Shapiro delay (Jacoby et al., 2005). This mass measurement, combined with other measurements and predictions of GR, implied that PSR J1909−3744 has a large mass. This supported the idea that the cause of the high spin-frequency of MSPs is that they undergo a recycling phase via accretion from their companion;
- a comparison between the pulsar timing measurements and the radio interferometry measurements of the position, parallax and proper motion of PSR J0437−4715. This led to independent confirmation of the parallax distance (Deller et al., 2008).

2.2.2 P456: "A millisecond pulsar timing array"

In February 2004, the first observations for the PPTA began under the Parkes observing project "P456: A millisecond pulsar timing array". High-precision observations (see Section 1.5.1) were collected from late 2004. This ongoing project differs from P140 as it specifically aims to detect GWs. The P456 project also aims to detect errors in TT(TAI) and in the solar system ephemeris. These aims require the extension of the list of monitored pulsars from 16 to 20 MSPs, suggested by Jenet et al. (2005) as a minimum requirement for GWB detection. The project includes low- and high-frequency observations of the pulsars to enable correction for variations in the pulse DM (You et al., 2007). Some highlights of the project so far include:

- the use of P456 observations and publicly-available observations (Kaspi et al., 1994) to measure an upper bound on the GWB amplitude that rules out some GWB formation mechanisms (Jenet et al., 2006);

[16] Observations of this pulsar were later used to constrain the parameters of alternative gravity theories (Verbiest et al., 2008; Deller et al., 2008).

- measurement of the instability of Terrestrial Time (TT), as realised by TT(TAI), using a combination of P140 and P456 observations (Hobbs et al., 2011);
- measurement of the mass of all solar system planetary systems from Mercury to Saturn, providing the most precise published value for the Jovian system mass (Champion et al., 2010). This paper used observations of three pulsars from P140 and P456 (published by Verbiest et al., 2009), as well as some observations of a fourth pulsar made at Effelsberg and Arecibo.

2.2.3 Properties of the P140/P456 Observations

For the majority of the analysis discussed in this paper, the P456 and P140 data have been combined. Each pulsar has been observed for ~10 min to 1 h in each observation, depending on the hardware used at the time. Since 2005, the typical length of an observation is ~1 h. Observations of each pulsar are made every few weeks, though PSRs J0437−4715 and J1909−3744 are often observed several times during each observing session. For some pulsars there are gaps of several months during which no observations were taken.

Most observations have been performed at wavelengths centred on 10/50 cm (3100/685 MHz) using a dual-frequency receiver (Granet et al., 2005) or at 20 cm (1400 MHz). Observations in the 20 cm band between 1994 and November 2002 were taken with the Caltech fast pulsar timing machine (FPTM; Navarro, 1994). This backend system timed pulsars with either one or two 128 MHz-wide bands; the observations varied greatly in quality. Observations between November 2002 and June 2010 were taken with the Caltech-Parkes-Swinburne Recorder 2 (CPSR2; Bailes, 2003). These observations were coherently de-dispersed over two 64 MHz-wide observing bands centred at 1341 MHz and 1405 MHz. From 2004, additional simultaneous observations have been taken with a variety of Parkes digital filterbank systems with bandwidths from 256 MHz to 1 GHz (Manchester, 2008). Each MSP monitored at Parkes has been observed for a different time-span, depending in part on when each pulsar was discovered. The observations have been made at irregular intervals and the sampling has been different between pulsars.

ToA uncertainties have varied widely over short and long timescales. Short timescale variation is caused by unequal integration time between observations and by scintillation in the ISM (Lyne & Rickett, 1968; Rickett, 1990; Cordes, 2002). Long timescale variation in the ToA error size is caused by upgrades in the receiver and backend systems at the telescope. The magnitude of the average ToA uncertainty can change discontinuously as a result of these upgrades in the observing hardware at Parkes. In Figure 2.2, we plot the timing residuals obtained from observations of two pulsars to demonstrate the variations in ToA uncertainties, sampling and time-span described above.

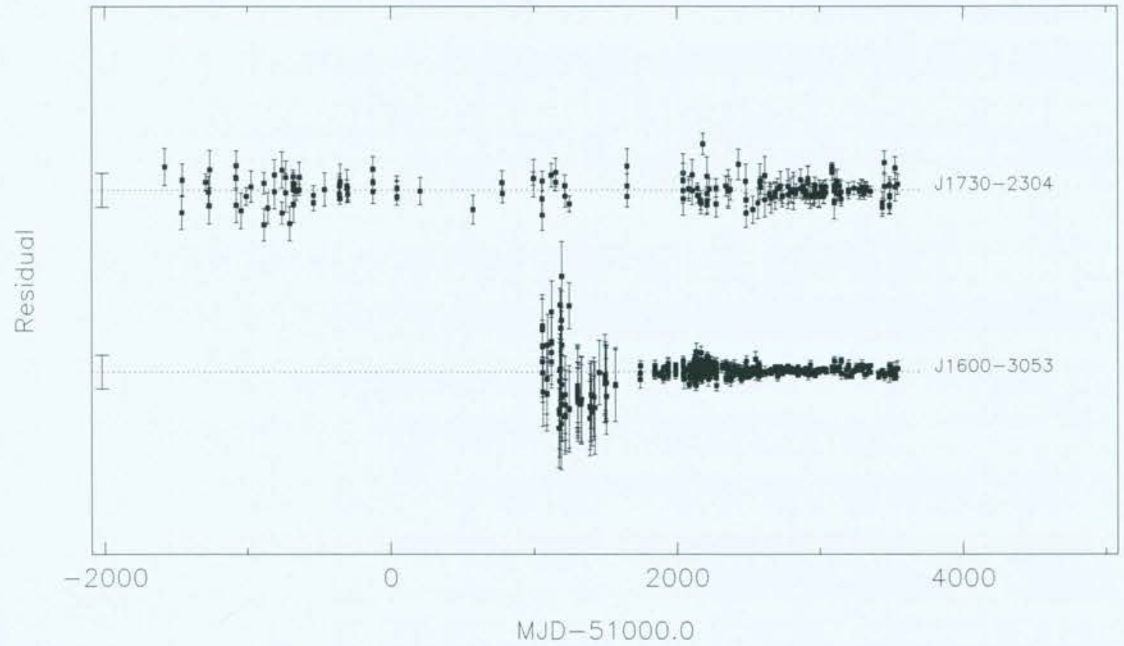

Figure 2.2: The variation in sampling, ToA uncertainties and time-spans of MSP timing observations under P140/P456. The abscissa is time in days as measured by the Modified Julian Date (MJD). The ordinate measures the timing residual for each observation of each pulsar. The dotted lines indicate zero residual for each pulsar, and the length of the vertical bar on the left-hand side in each panel indicates 10μs. The right-hand column gives the pulsar's name in the J2000 coordinate system. The observations of PSRs J1730−2304 and J1600−3053 shown here are a subset of the 20 sets of MSP timing observations published by Verbiest et al. (2009).

2.2.4 Fitting the Timing Model and Estimation of Pulsar Parameters

The ToAs for each pulsar are fit with a model for the pulsar's behaviour to minimise the variance of the timing residuals. A typical model includes the DM, rotational parameters (pulse frequency and its first derivative), astrometric parameters (e.g., position, proper motion) and, if the pulsar is a member of a binary system, orbital parameters (i.e., the Keplerian binary parameters and, if necessary, some post-Keplerian parameters; see Edwards et al., 2006). A significant parallax has only been measured for pulsars that are relatively close to Earth or for those with a small rms residual (Verbiest et al., 2009).

For our data, the fitting of the pulsar timing model is carried out with TEMPO2 (Hobbs et al., 2006; Edwards et al., 2006). TEMPO2 is a software package designed as a plugin architecture in the 'C' programming language. This enables users to write their own programs in C with access to the TEMPO2 core. TEMPO2 supersedes the earlier TEMPO code that was not designed for processing multiple pulsars simultaneously, whereas TEMPO2 can simultaneously process PTA observations (e.g., Hobbs et al., 2006, 2011). Also, the TEMPO code does not account for all effects that cause ToA variations of $\lesssim 100\,\mathrm{ns}$. TEMPO2 accounts for all known timing effects to $\sim$1 ns accuracy (Hobbs et al., 2006). This level of accuracy is an order of magnitude greater than the most precise current timing observations (e.g., Manchester, 2011). In particular, TEMPO2 accounts for

- the pulsar's intrinsic slow-down behaviour;
- its orbital motion;
- its secular motion or that of its binary system;
- dispersion caused by the solar system, Earth's ionosphere, and the ISM;
- the motion of the observatory caused by Earth's rotation, orbital motion, precession, nutation and polar motion;
- pulse delay induced by Earth's troposphere; and
- gravitational time-delays caused by solar system bodies or the pulsar's binary companion.

In this way, TEMPO2 produces very precise parameter estimates for any observed pulsar. Pulsar model parameters for most of the Parkes pulsars timed under P140 and P456 have been published by Verbiest et al. (2008, 2009).

Compared to other measurement techniques, the pulsar parameters are most precisely determined using the timing observations themselves[17]. However, it should not be assumed that every measured parameter has physical meaning. For example, the intrinsic rotational period

[17] A notable exception is the parallax measurement for PSR J0437−4715 obtained with interferometry that is an order of magnitude more precise than the best measurement from pulsar timing (Deller et al., 2008; Verbiest et al., 2008). However, the pulsar timing measurement of other parameters (such as the proper motion) is more precise than the interferometric measurement. Also, while such precise interferometry measurements are possible for the very close and bright pulsar PSR J0437−4715, these measurements will not be practical for the general population of pulsars because of their much greater distance and lower flux density.

of a pulsar cannot be directly measured with current techniques. The pulse period determined by TEMPO2 is affected by the pulsar's radial velocity, intrinsic pulsar rotational instabilities (Hobbs et al., 2010b), instabilities in terrestrial time standards (Hobbs et al., 2011), and even the existence of GW signals (e.g., Pshirkov, 2009), amongst numerous other factors.

2.2.5 Properties of the P140/P456 Timing Residuals

The timing residuals (see Section 1.5.1) produced by TEMPO2 for the P140/P456 observations are shown in Figures 2.4, 2.5 and 2.6. The error bar on each residual, which is equal to the ToA uncertainty, is underestimated on average for almost all of the PPTA pulsars (Verbiest et al., 2009). This means that the uncertainty in the parameter estimates for these pulsars will be underestimated[18]. In an effort to correct this, the standard approach is to multiply each measured ToA uncertainty for a particular pulsar by an "error factor" (EFAC; e.g., Verbiest et al., 2009). Generally, the EFAC (typically a number between one and four and defined as $\sqrt{[\chi_r^2]}$) will be different for each pulsar, though a recent analysis by Verbiest et al. (2009) used EFACs that were also different for each backend system.

Several deterministic signals have been removed from the residuals because of the timing model fit that estimates the parameters of the pulsar model. In Figure 2.3, we show characteristic signatures induced in the timing residuals for an incorrect pulse period (top left), an incorrect pulse period derivative (top right), an incorrect pulsar sky-position (bottom left) and an incorrect pulsar binary orbital period (bottom right). Any physical phenomenon that induces timing residuals resembling the signals in Figure 2.3 – such as a constant Doppler shift of the pulse period, acceleration of the pulsar in the local gravitational potential (e.g., for pulsars in globular clusters; see Freire et al., 2001) or GW signals with a period of 1 yr – will be undetectable after the standard pulsar timing fit has been applied. The implications of this are discussed in Chapter 4 and Chapter 6 and have been discussed by several authors (e.g., Blandford et al., 1984; Hellings, 1989; Kopeikin, 1999).

The weighted rms residual varies over two orders of magnitude between different pulsars (see Table 2.2). This is because of the S/N ratio difference of the pulse profile between different

[18] This can be corrected using the "reduced χ^2" of the fit, defined as $\chi_r^2 = \sum_n \frac{r_n^2}{\sigma_n^2}/N_{\rm dof}$, where r_n is the n-th observed post-fit residual, σ_n is its error and $N_{\rm dof}$ is the number of degrees of freedom in the residuals. By multiplying each measured parameter uncertainty by $\sqrt{\chi_r^2}$, the correct uncertainty can be obtained if the uncertainties have a normal distribution. Ideally, χ_r^2 is close to unity, indicating that the model fits the observations at the accuracy predicted by the noise level of the observations.

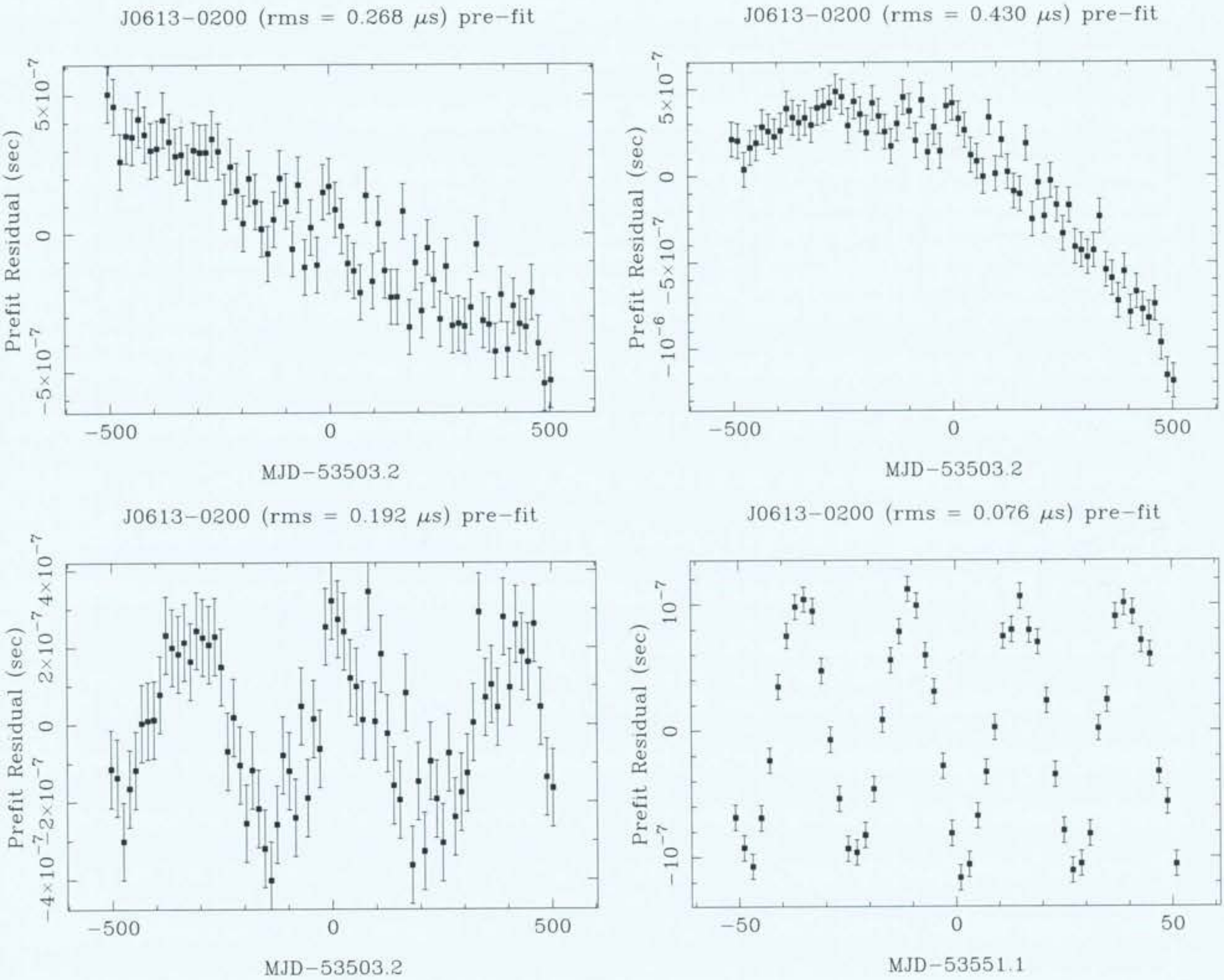

Figure 2.3: The signature in simulated pulsar timing residuals for PSR J0613−0200 as produced by an incorrect value of different pulsar parameter measurements. In each plot, the abscissa is the MJD while the ordinates are the residuals determined using the input pulsar parameter file before fitting is applied within TEMPO2 to improve the pulsar parameter estimates. Here we plot the timing residuals after introducing an error in the pulsar's period (top left), period derivative (top right), sky position (bottom left) and binary orbit period (bottom right). These simulated observations are sampled once every 14 d with an uncertainty of 100 ns on each residual, except for the data set displaying the binary orbit period error. For the binary orbit period error, one observation every two days with a 10 ns uncertainty on each residual was chosen since the orbital period of PSR J0613−0200 is 25 d. Each image was produced using the PLK plugin to TEMPO2 from simulated data created using the FAKE plugin.

pulsars and because of unmodelled signals in the timing observations. For example, most of the published observations from these projects have not been fully corrected for DM variations (You et al., 2007)[19]. This means that the residuals still contain signals due to variations in the DM. Other physical effects that have not been included in the timing model – such as calibration and other instrumental errors (van Straten, 2006), timing noise intrinsic to the pulsar system (Hobbs et al., 2010b; Shannon & Cordes, 2010, and references therein), errors in the solar system ephemeris (Champion et al., 2010), and errors in TT(TAI) (Hobbs et al., 2011) – will induce timing residuals.

Verbiest et al. (2009) showed that such noise sources will not prohibit GWB detection with the PPTA pulsars. Furthermore, these authors provided an in-depth analysis of the noise properties of all the PPTA pulsars. They conclude that instabilities intrinsic to the pulsars or the observing systems do not induce residuals with rms $>$ 100 ns over a five-year timescale for most PPTA pulsars. Hence, GW detection with the PPTA pulsars remains a possibility, using observations carried out under the observing projects P140 and P456. Some recently published observations of the PPTA pulsars are described in the next Section. Both sets of observations will be analysed to study GW signals in Chapters 3, 4, 5 and 6.

2.3 Published Observations from the Parkes Pulsar Timing Array

In this Section we describe the observations that formed the Jenet et al. (2006) data set and those that formed the Verbiest et al. (2009) data set. We show the pulsar timing residuals that result from each timing analysis. In Chapters 3 and 5, we analyse the Jenet et al. (2006) observations to determine their sensitivity to individual sources of GWs and calculate an upper bound on the GWB amplitude. In Chapters 4 and 6, we analyse the Verbiest et al. (2009) observations and search for single sources of GWs and a GWB signal.

2.3.1 The Jenet et al. (2006) Observations

Jenet et al. (2006) presented a statistically-rigorous technique for finding an upper bound on the GWB amplitude. The Jenet et al. (2006) observations have been assembled from:

[19] Such corrections are now part of the standard PPTA data processing and can be applied to most existing residuals for which observations have been made at two or more frequencies.

- publicly-available observations of PSR J1857+0943 taken with the Arecibo radio telescope between 1986 and 1993 (Kaspi et al., 1994); and
- observations of seven pulsars, including PSR J1857+0943, made with the Parkes radio telescope under the P140 and P456 timing projects (Hotan et al., 2006).

The Arecibo observations of PSR J1857+0943 were carried out at ~1400 MHz and span eight years. The Parkes observations used both the 20 cm receiver and the 10/50 cm dual-frequency coaxial receiver. The average sampling interval of these data is ~16 d, with one observation at each of the three frequencies taken during each 16 d period. The observed pulsar signals were recorded with a variety of backend systems, including:

- the Wide-Band Correlator system (You et al., 2007) with 2 bit digital sampling at a bandwidth of up to 1 GHz;
- a digital filterbank with 8 bit digital sampling of a 256 MHz bandwidth (Yan et al., 2011);
- the Caltech Parkes Swinburne Recorder 2 (CPSR2; details in Bailes, 2003; Hotan et al., 2006), a baseband recorder with coherent dedispersion over two observing bands, each of 64 MHz bandwidth. For observations at 20 cm, these bands are centred on 1341 and 1405 MHz; for simultaneous observations at 10/50 cm they are centred on 3100 and 685 MHz.

The residuals are summarised in Table 2.1 and plotted in Figure 2.4. The figure shows the very irregular sampling, the unequal noise levels between pulsars and the significant variation in error bar size between observations of a given pulsar. In Section 3.3.3, we will analyse these observations to determine their sensitivity to individual GW sources that induce sinusoidal timing residuals.

Jenet et al. (2006) did not apply their method to all the Parkes MSP observations that were available in 2006. This is because their technique for finding an upper bound on the GWB amplitude (described in Chapter 5) demanded the use of time series that were consistent with white noise (i.e., their power spectrum is independent of frequency). Many time series from the P140 and P456 projects did not meet their criteria for whiteness because of calibration- and hardware-induced artifacts, as well as other unknown timing noise sources (Jenet et al., 2006). For some other pulsars, only a subset of their observations could be used. This meant that,

Table 2.1: Parameters of the Jenet et al. (2006) data set.

PSRJ	Period (ms)	DM (cm^{-3} pc)	P_b (d)	Weighted RMS Residual (μs)	Span (years)	No. of Observations
J0437−4715	5.757	2.65	5.74	0.12	2.2	233
J1024−0719	5.162	6.49	–	1.10	2.4	92
J1713+0747	4.570	15.99	67.83	0.23	3.2	168
J1744−1134	4.075	3.14	–	0.52	3.3	101
J1857+0943	5.362	13.31	12.33	1.12	20.3	398
J1909−3744	2.947	10.39	1.53	0.29	2.4	2859
J1939+2134	1.558	71.04	–	0.21	2.4	231

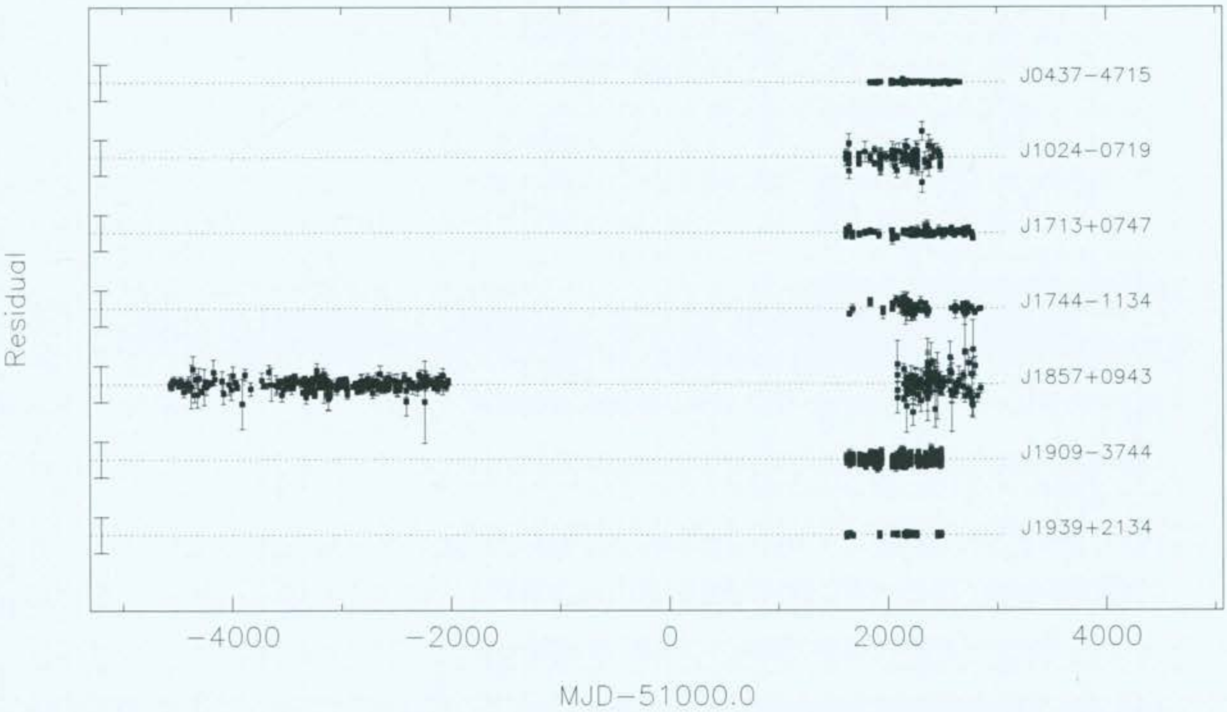

Figure 2.4: The timing residuals from the observations of seven pulsars published by Jenet et al. (2006). The abscissa gives the time of the corresponding observation in MJD. The ordinate measures the timing residual for each observation of each pulsar. The dotted lines indicate zero residual for each pulsar, and the length of the vertical bar on the left-hand side in each panel is 10μs. The right-hand column gives the pulsar's name in the J2000 coordinate system. [Image produced using the PLOTMANY plugin to TEMPO2.]

Table 2.2: Basic information for the Verbiest et al. (2008, 2009) data sets.

PSRJ	Period	DM	P_b	Weighted RMS	Span	No. of
	(ms)	(cm^{-3} pc)	(d)	Residual (μs)	(years)	Observations
J0437−4715	5.757	2.65	5.74	0.20	9.9	2847
J0613−0200	3.062	38.78	1.20	1.56	8.2	190
J0711−6830	5.491	18.41	–	3.23	14.2	227
J1022+1001	16.453	10.25	7.81	1.62	5.1	260
J1024−0719	5.162	6.49	–	4.22	12.1	269
J1045−4509	7.474	58.15	4.08	6.64	14.1	401
J1600−3053	3.598	52.19	14.34	1.14	6.8	477
J1603−7202	14.842	38.05	6.31	1.92	12.4	212
J1643−1224	4.622	62.41	147.02	2.50	14.0	241
J1713+0747	4.570	15.99	67.83	0.20	14.0	392
J1730−2304	8.123	9.61	–	2.51	14.0	180
J1732−5049	5.313	56.84	5.26	3.24	6.8	129
J1744−1134	4.075	3.14	–	0.62	13.2	342
J1824−2452	3.054	119.9	–	1.60[20]	2.8	89
J1857+0943	5.362	13.31	12.33	1.21	22.1[21]	376
J1909−3744	2.947	10.39	1.53	0.17	5.2	893
J1939+2134	1.558	71.04	–	23.9	23.3[22]	588
J2124−3358	4.931	4.62	–	4.03	13.8	416
J2129−5721	3.726	31.85	6.63	2.19	12.5	179
J2145−0750	16.052	9.00	6.84	1.82	13.8	377

because of the nature of their technique, Jenet et al. had to discard a large fraction of their observations.

2.3.2 The Verbiest et al. (2009) Observations

Verbiest et al. (2009) presented long time-span observations of 20 MSPs using the Parkes radio telescope[23]. The pulsars were timed with a weighted rms residual of $\sim 0.2 - 23\,\mu$s for a period of $\sim$12 years. The specifications of each set of timing residuals are given in Table 2.2, where, in column order, we present the pulsar name in the J2000 coordinate system, pulse period, dispersion measure, orbital period, weighted rms residual, data-span and number of recorded ToAs. For full details of ToA estimation and data processing, see Verbiest et al. (2008, 2009). The timing residuals from all observations are shown in figure 1 of Verbiest et al. (2009). Between different pulsars, there is variation in the noise level of the residuals and the sampling

[23] In this thesis, we augment the Verbiest et al. observations by adding eight years of ToAs for PSRs J1857+0943 and J1939+2134 to the beginning of the list of observations for these pulsars. These additional ToAs were obtained from publicly-available data collected using the Arecibo radio telescope and presented by Kaspi et al. (1994).

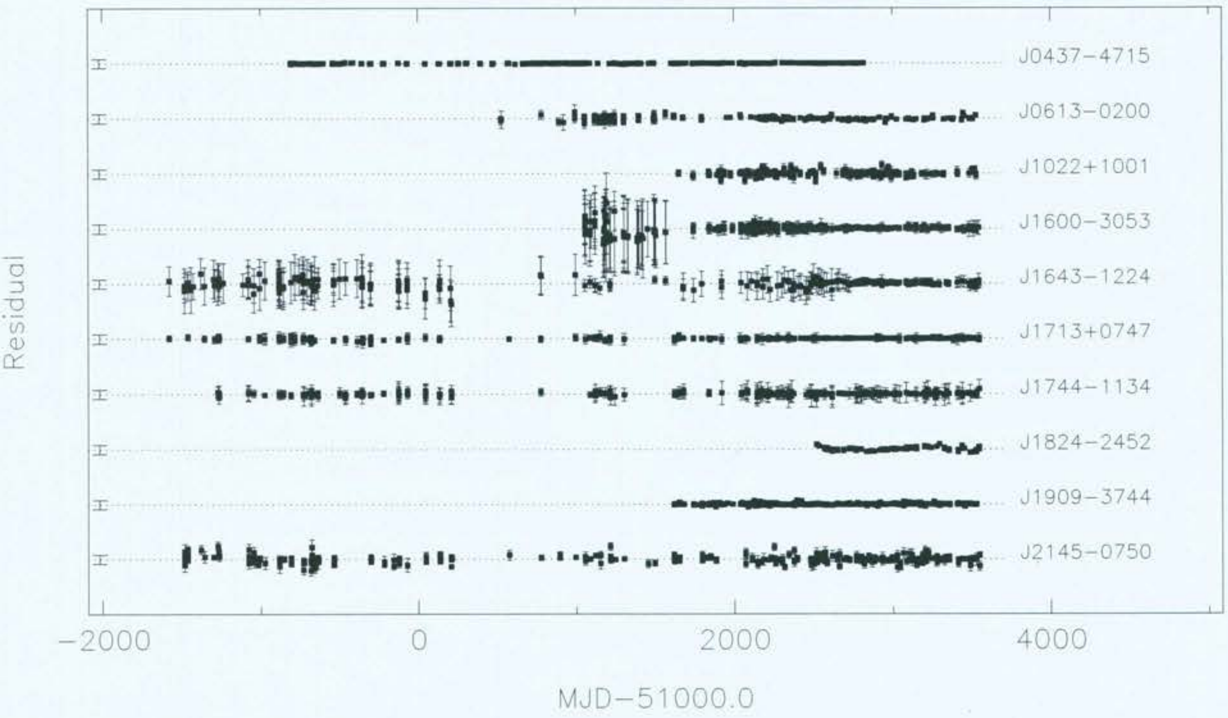

Figure 2.5: The 10 pulsars with the smallest weighted rms residual in the Verbiest et al. (2009) observations. The abscissa gives the date of the observation. The ordinate measures the timing residual for each observation of each pulsar. The dotted lines indicate zero residual for each pulsar, and the length of the vertical bar on the left-hand side in each panel indicates 10μs. The right-hand column gives the pulsar's name in the J2000 coordinate system. Noise levels vary significantly both between pulsars and at different epochs. The time-span also differs for different time series, and in general the observations of each pulsar were begun on different dates.

frequency and start dates for the observations. In Figure 2.5, we plot the timing residuals for the 10 pulsars with the smallest weighted rms residual with the same scaling on the axes. In Figure 2.6, we plot a similar figure for the remaining 10 pulsars.

The observations were made with a number of different observing systems – both the frontend receivers and the backend instrumentation have varied over time. Arbitrary phase offsets have been fitted for and removed between the ToAs from each different observing system for a given pulsar. This reduces the standard deviation of the timing residuals for that pulsar and can remove significant signals from the residuals, especially over long timescales. This effect is shown in Figure 2.7 for an extended set of observations of PSR J0437−4715 that includes more recent data than that published by Verbiest et al. (2009).

The Verbiest et al. (2009) residuals have a number of features that complicate their analysis. While the timing residuals of most of the pulsars are white, nine out of the twenty pulsars exhibit

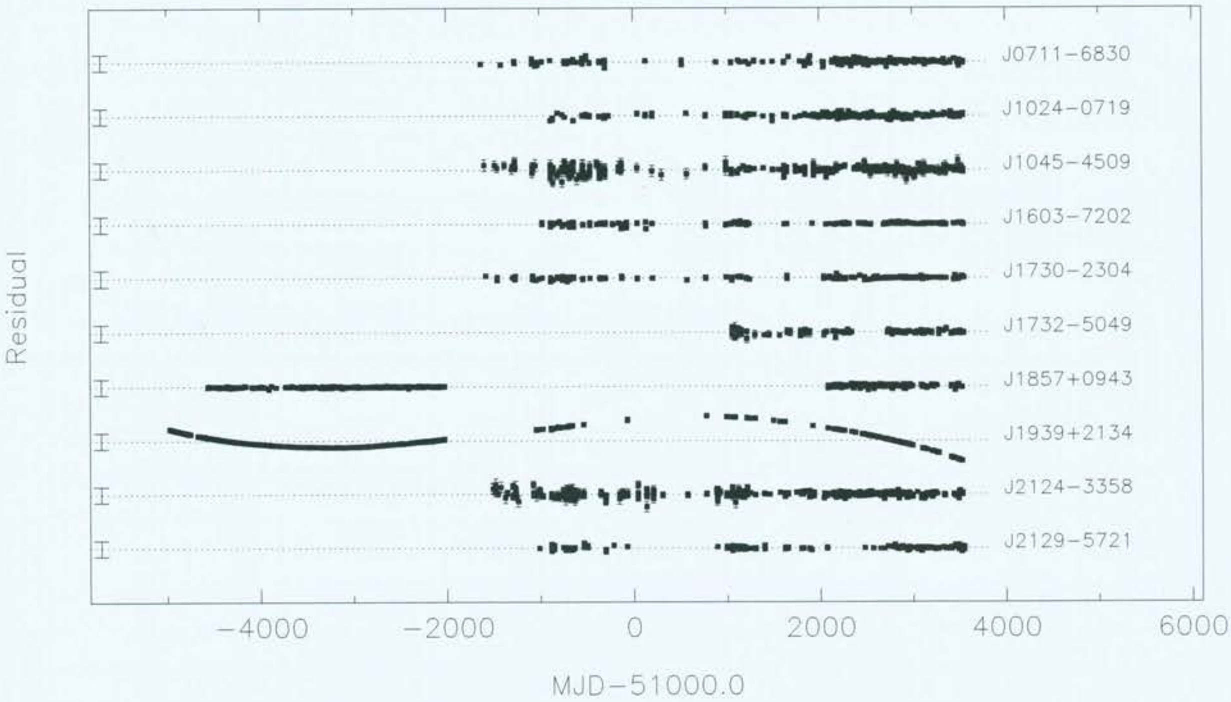

Figure 2.6: The 10 pulsars with the largest weighted rms residual in the Verbiest et al. (2009) observations. The abscissa gives the date of the observation. The ordinate measures the timing residual for each observation of each pulsar. The dotted lines indicate zero residual for each pulsar, and the length of the vertical bar on the left-hand side in each panel indicates 50μs. The right-hand column gives the pulsar's name in the J2000 coordinate system. Noise levels vary significantly both between pulsars and at different epochs. The residuals of PSR J1939+2134 are dominated by a polynomial of unknown origin.

non-white noise[24].

The Verbiest et al. (2009) observations contain a wealth of information on many physical effects. However, the techniques for GW analysis presented to date have difficulty in analysing the residuals. The Jenet et al. (2006) technique cannot be applied in its current form, nor can the Anholm et al. (2009) technique, as outlined in Section 1.6. The technique presented by van Haasteren et al. (2009) can be applied to these observations, but this would require a large amount of computation time and the results would be difficult to confirm via Monte Carlo simulation. This necessitates the development of new techniques for GW analysis that can be applied to pulsar timing observations. Such methods are described in subsequent Chapters of this thesis. To develop and test new GW-analysis techniques, we need to be able to simulate PTA observations. Many methods are possible for creating simulated timing residuals; we choose to use the methods implemented in the FAKE and PSD_SIMULATOR plugins to TEMPO2.

[24] This was determined using a simple two-point correlation analysis to determine the degree of correlation between adjacent residuals using the CHECKWHITE plugin to TEMPO2.

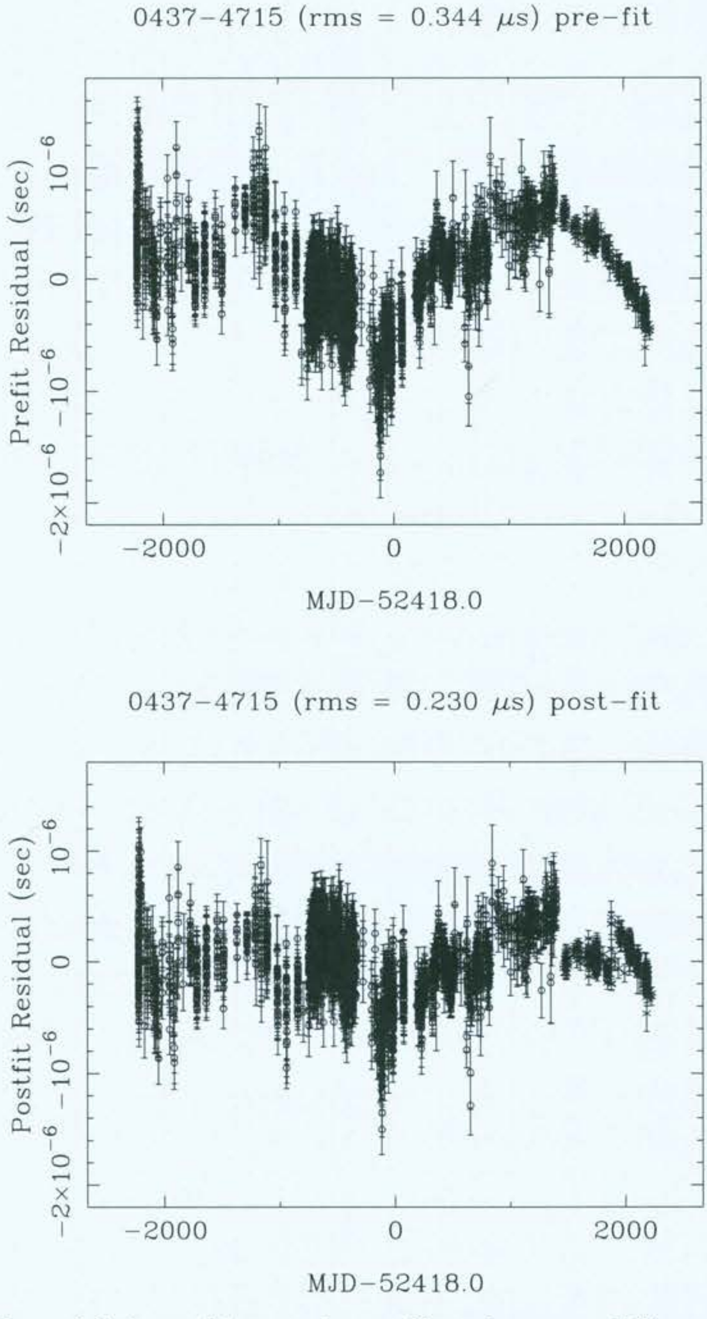

Figure 2.7: The effect of fitting arbitrary phase offsets between different observing systems on the PSR J0437−4715 residuals obtained from observations under P140 and P456. The total time-span of the observations in each figure is 12.2 years; the first 9.9 years of data were presented by Verbiest et al. (2008) and are described above. If the phase offsets are measured using very precise system tests at Parkes (upper figure; Manchester, 2011), then significant low-frequency structure is revealed in the residuals, indicating the presence of an unmodelled signal. If we instead determine these phase offsets using a TEMPO2 fit of the observations, as was done by Verbiest et al. (2008, 2009) for the first 9.9 years of the observations, then most of this signal is removed (lower figure).

2.4 Simulating Pulsar Timing Observations

2.4.1 The FAKE Plugin to TEMPO2

The FAKE plugin produces simulated observations at a user-defined set of times that are affected by user-selected levels of white noise, red noise or other pulsar timing effects such as glitches. The only required input is the timing model for the pulsar. The arrival times predicted by the input timing model are subtracted from the list of observation times, which are assumed to represent pulse ToAs, and the timing residuals are formed. These timing residuals are then subtracted from the initial ToAs, creating a new set of ToAs that will be predicted more accurately by the input timing model than the initial ToAs. The TEMPO2 modelling and fitting process is non-linear in general, meaning that this procedure must be repeated until the timing residuals are negligible (Hobbs et al., 2009). The ToAs will then be exactly predicted by the input timing model; we refer to these as a set of "ideal ToAs". The ideal ToAs have the same sample times as the actual observations. They can then be modified using the specified levels of white noise, red noise, GWs and any other simulated effects. This process creates simulated pulsar timing observations for which we know the form of all effects that influence the ToAs. In the upper panel of Figure 2.9, we show simulated timing residuals for PSR J0613−0200 created with the FAKE plugin. The simulated residuals have the same time-span, average sampling and weighted rms residual as the real observations, but the observation times, error bars and spectral properties of the simulated residuals do not resemble the actual observed residuals shown in Figure 2.8.

2.4.2 Simulating Observations with Variable Error Bars and Irregular Sampling

If we have a list of ToAs and their uncertainties, it is straightforward to simulate ToAs with the same sampling and error bars as the input ToAs using TEMPO2. In particular, this means that the noise level can vary from one ToA to the next, as occurs for real observations of pulsars. The sampling interval between consecutive observations can also vary. In the lower panel of Figure 2.9, we show timing residuals formed from a simulated set of observations for PSR J0613−0200. Each simulated observation has the same error bar and MJD as the corresponding real observation.

These simulated observations give residuals that are more similar to the observed data in Fig-

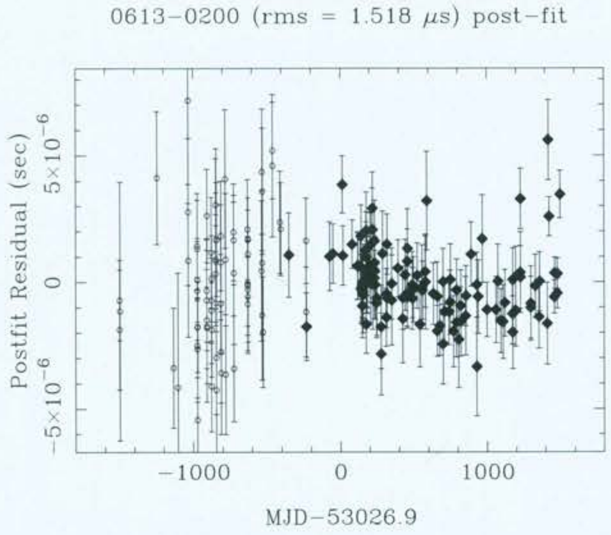

Figure 2.8: The observed timing residuals for PSR J0613−0200 from Verbiest et al. (2009). The residuals collected with CPSR2 (filled diamonds) and other backend systems (open circles) are shown. We investigate three different methods to simulate a time series that resembles this time series as closely as possible. The results of applying each different simulation method are shown in Figures 2.9 and 2.13.

ure 2.8 than the simulated observations created by FAKE. However, the simulations described here still yield residuals that are consistent with white noise of varying standard deviation. The observations in Figure 2.8 exhibit significant low-frequency noise that should be included in simulations of observations of this pulsar[25].

2.4.3 The PSD_SIMULATOR Plugin to TEMPO2

To simulate significant low-frequency noise in pulsar timing residuals, we require a mathematical model of the low-frequency spectrum of the "pre-fit" residuals. We define the pre-fit residuals to be the timing residuals obtained before applying the TEMPO2 parameter fit to determine a new timing model. The predictions of this new timing model can be subtracted from the ToAs to form the "post-fit" residuals. When observing pulsars, multiple fits will already have been applied to the data. For example, in the discovery of a new pulsar, an estimate of the pulsar's pulse period, dispersion measure and sky-position will have been obtained. Subsequent observations of the pulsar will allow these parameters (as well as the period derivative and other parameters) to be measured using the standard pulsar timing procedure, which includes a pulsar parameter fit. As the time-span of the observations increases, it is possible to fit for more and

[25] The low-frequency noise in PSR J0613−0200 was analysed by Verbiest et al. (2009).

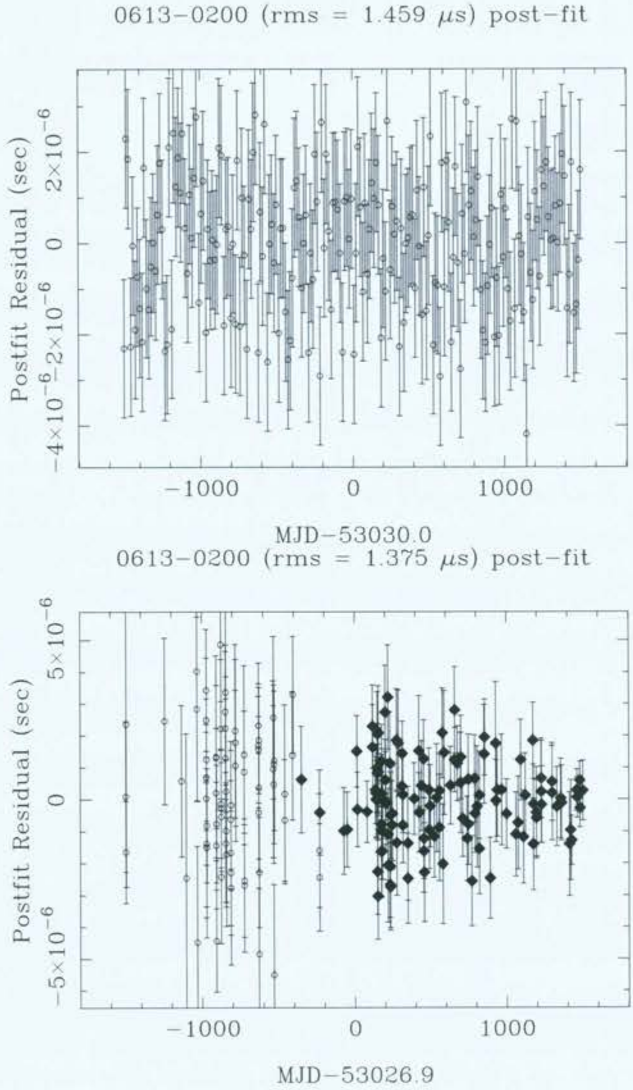

Figure 2.9: Simulated timing residuals for PSR J0613−0200. The upper panel was created using the FAKE plugin to TEMPO2 and assuming regular sampling with equal error bars. The lower panel was created using the same observation epochs and error bars as the observed residuals of PSR J0613−0200. The random gaussian deviation at each residual is given by the error bar size, but does not include any low-frequency noise. The filled diamonds are the residuals resulting from simulated observations taken with the CPSR2 observing backend system; open circles show the residuals resulting from simulated observations using other observing backend systems. Neither of the simulated data sets shown here resemble the actual observed timing residuals in Figure 2.8.

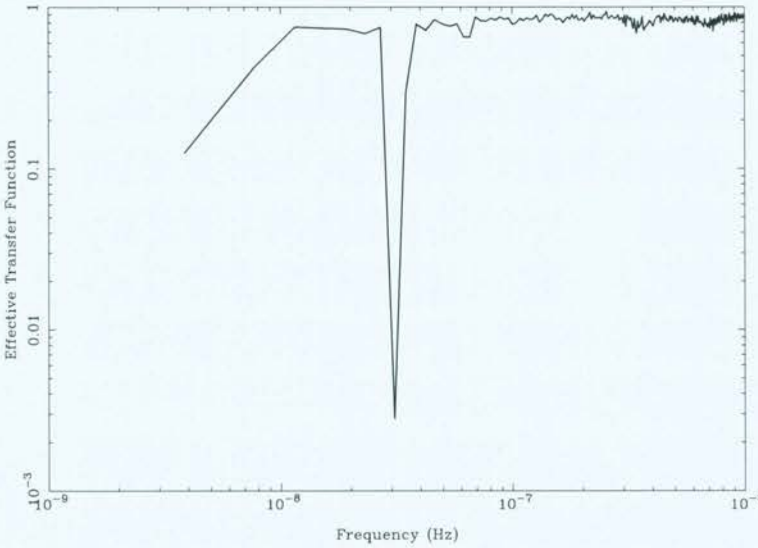

Figure 2.10: The effective transfer function for the Verbiest et al. (2009) observations of PSR J0613−0200. The abscissa gives the frequency, while the ordinate gives the effective transfer function of the TEMPO2 parameter fitting process. See text for more details.

more pulsar parameters. It is therefore not usually possible to determine timing residuals that do not have various signatures subtracted from them by the TEMPO2 parameter fit (see Figure 2.3). However, such residuals can be simulated[26], which enables subsequent investigation of the effect of the TEMPO2 fit on the data. The effect of the fitting is particularly important when applied to observations affected by non-white noise. For instance, significant low-frequency power will be removed when estimating the pulsar period, its derivative (see Figure 2.3) or any arbitrary phase offsets (see Figure 2.7).

Before simulating pre-fit residuals explicitly as a time series, we first simulate their power spectrum. This requires an estimate of the average effect of the TEMPO2 pulsar parameter fit on the particular set of observations being simulated. This estimate can be calculated using the XFER_FUNC plugin to TEMPO2. This plugin estimates the power spectrum – before and after pulsar parameter fitting – of simulated white noise with the same sampling and ToA errors as the input timing residuals (see Section 2.4.2). Dividing the post-fit spectrum by the pre-fit spectrum gives an estimate of the effective "transfer function" of the TEMPO2 fitting procedure (e.g., Blandford et al., 1984; Hellings, 1989). This process is repeated 1000 times to find the average effective transfer function.

[26] We later show how to simulate the timing residuals induced by a GW signal. The simulation creates ToAs from an input timing model by adding user-defined noise levels and GW signals to pulse arrival times that are predicted by the model. Standard TEMPO2 fits can then be applied to determine the post-fit residuals that would actually be observed.

The effective transfer function for the Verbiest et al. (2009) observations of PSR J0613$-$0200 is shown in Figure 2.10. A small value of the effective transfer function indicates a frequency at which the TEMPO2 parameter fit removes most of the power. There are three prominent features. First, at least 5% of the input power is lost at all frequencies during the TEMPO2 fit. This is mainly caused by the fitting of two arbitrary phase offsets to the ToAs. Second, a significant loss of power occurs in the lowest two frequency channels because of the TEMPO2 fits for the period, period derivative and arbitrary phase offsets. Third, near-total loss of power occurs at a frequency of 3.1×10^{-8} Hz because of the TEMPO2 fit for the pulsar's sky-position (see Figure 2.3).

To obtain the pre-fit power spectrum, we divide the measured power spectrum of the actual residuals by the effective transfer function. The pre-fit power spectrum is shown in Figure 2.11, along with the spectrum of the actual residuals and a model of the pre-fit spectrum $P_{\rm model}(f) = af^b$ for the low-frequency portion of the spectrum. We aim to simulate a time series that is consistent with this low-frequency spectral model. For the high-frequency noise in the residuals, we use the ToA error bars in the same way as described in Section 2.4.2. After the TEMPO2 fit is applied to the total time series, the post-fit simulated residuals will resemble the time series of actual residuals in Figure 2.8.

We can simulate a time series with user-defined sampling that has a power spectrum consistent with a given spectral model. We first simulate an equally-spaced time series with the required power spectral density (e.g., the dashed line in Figure 2.11). The power spectral density of a time series r_n of $N_{\rm pts}$ points with time-span $T_{\rm obs}$ at frequency f_k can be defined as[27]

$$P(f_k) = 2T_{\rm obs}\left|\mathcal{F}(f_k)\right|^2 \,, \tag{2.1}$$

where $\mathcal{F}(f_k)$ is the Discrete Fourier Transform (DFT) of the time series. To calculate the DFT, we use the TK_FFT function within the TKSPECTRUM library to TEMPO2. This function uses the following definition of the DFT:

$$\mathcal{F}(f_m) = \frac{1}{N_{\rm pts}} \sum_{n=0}^{N_{\rm pts}-1} r_n {\rm e}^{-2\pi {\rm i} mn/N_{\rm pts}} \,, \tag{2.2}$$

where m is an integer between 0 and $(N_{\rm pts} - 1)$. The power spectral density defined in Equation

[27] See, e.g., equation (6.6) of Albrecht et al. (2003).

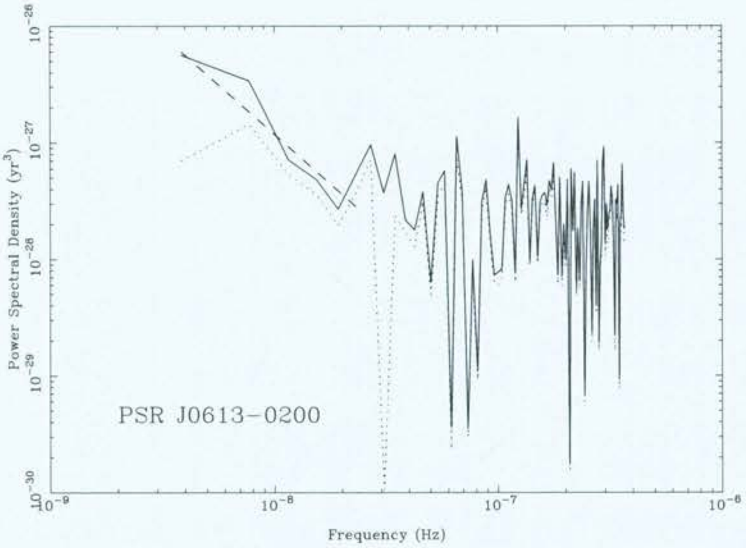

Figure 2.11: Power spectra of the pre-fit residuals (solid trace) and the actual residuals (dotted trace) from the Verbiest et al. (2009) observations of PSR J0613−0200. The abscissa gives the frequency, while the ordinate gives the power level in units of yr^3. The "pre-fit" spectrum is obtained by dividing the power spectrum of the actual residuals by the transfer function plotted in Figure 2.10. We can model the low-frequency portion of the pre-fit spectrum with a power-law (dashed line).

(2.1) is one-sided, meaning that we only allow positive frequency channels between $k = 1$ and $k = (N_{\rm pts} - 1)/2$, rounded down. We define $P(f_0) = 0$ because there is no information contained in the mean of a set of pulsar timing residuals. The DFT defined in Equation (2.2) is two-sided.

Rearranging equation (2.1) yields

$$|\mathcal{F}(f_k)|^2 = \frac{P(f_k)}{2T_{\rm obs}} \ . \tag{2.3}$$

Since the DFT contains real and imaginary parts (R_k and I_k respectively), we have

$$R_k^2 + I_k^2 = \frac{P(f_k)}{2T_{\rm obs}} \ . \tag{2.4}$$

Our method for simulating a time series that has a random power spectral density consistent with the input $P(f_k)$ uses the following procedure:

- We create two arrays of $N_{\rm pts}$ normally distributed random numbers, R_n and I_n. $N_{\rm pts}$ is a parameter that can be defined by the user and must be a power of two in our implementation.

- We multiply each array by $[P_{\rm model}(f = f_k)/4T_{\rm obs}]^{1/2}$, where $P_{\rm model}$ is the model for pre-fit power spectrum and $f_k = k/T_{\rm obs}$. R_n now represents the real part of the DFT while I_n represents the imaginary part. These arrays will now satisfy the identity in Equation (2.4).

- We perform an Inverse DFT on $R_n + {\rm i}I_n$ to form a complex array r_n. We define the Inverse DFT as:
$$r_n = \sum_{k=0}^{N_{\rm pts}-1} \mathcal{F}(f_k){\rm e}^{{\rm i}2\pi kn/N_{\rm pts}}\ . \tag{2.5}$$
The time series r_n will be regularly sampled with sampling interval $t_n = t_0 + nT_{\rm obs}/N_{\rm pts}$, where t_0 is the arbitrary start-time for the series. We shift the time series such that t_0 is the time of the first observation in the actual data.

- We interpolate the residuals onto the arrival times in the observed time series using a constrained cubic spline. This interpolation works well when $N_{\rm pts}$ is sufficiently large, as shown in Figure 2.12.

- The power spectrum of these interpolated residuals will follow the input spectral model.

This procedure can be extended to provide a realistic simulation of a set of observed pulsar timing residuals. We add white noise consistent with the measured error bars on the real data to the values of the interpolated time series described above. This yields a new time series affected by white noise and low-frequency noise that is uncorrelated between pulsars. This new time series has exactly the same sample-times as the actual residuals. This means we can add each value to a set of ideal ToAs – determined using the procedure described in Section 2.4.1 – to form a simulated set of ToAs. Timing residuals formed from these simulated ToAs will have the same power spectrum, ToA uncertainties and sampling as the actual residuals.

The required transformations described above are implemented in the PSD_SIMULATOR plugin, which is given in C-code in the Appendix. The implementation includes the fact that the timing residuals are real, meaning that the DFT will be Hermitian. This means that $R\left[\mathcal{F}(f_k)\right] = R\left[\mathcal{F}(f_{N_{\rm pts}-k})\right]$ and $I\left[\mathcal{F}(f_k)\right] = -I\left[\mathcal{F}(f_{N_{\rm pts}-k})\right]$, which can reduce the number of computations required.

While the FAKE plugin described in Section 2.4.1 can simulate ToAs that yield gaussian red noise in the timing residuals, it is restricted to simulating power-law models. The PSD_SIMULATOR

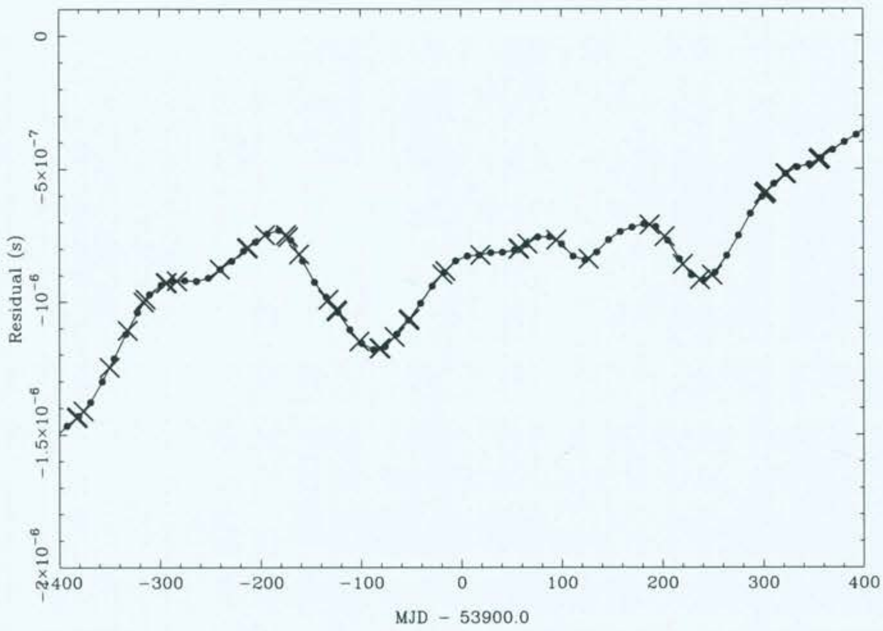

Figure 2.12: Interpolation of the time series calculated in Equation (2.5) onto the actual observation times for PSR J0613−0200. Only a subset of the full time series of PSR J0613−0200 residuals is shown here. The time series (dots connected by solid line) is initially sampled regularly once every 11.7 d. This is then interpolated onto the actual observation times for the ToAs from PSR J0613−0200 in the Verbiest et al. (2009) data set (crosses).

plugin can simulate gaussian noise consistent with most spectral models. In Figure 2.13 we plot an example of the residuals formed from simulated ToAs for PSR J0613−0200 created using the PSD_SIMULATOR plugin, cf. the actual residuals plotted in Figure 2.8.

With the ability to simulate realistic timing residuals, we now seek to add a variety of signals to these simulations. In this thesis we focus on the addition of different GW signals to the ToAs. In the next Section, we describe methods for simulating GW sources and their effect on the arrival times of pulses from a pulsar.

2.5 Simulating GWs with TEMPO2

TEMPO2 simulates the effect of GW signals on ToAs rather than timing residuals, creating data sets that can be processed using exactly the same method as ToAs collected with a telescope. For instance, the same processing tasks (such as parameter fitting, determining arbitrary phase jumps or measuring DM variations) can be applied to the simulated ToAs in exactly the same way as they are applied to the measured ToAs. As described in Section 1.5, ToAs will be affected by GW signals from SMBHBs. The strain amplitude induced by GWs from SMBHBs

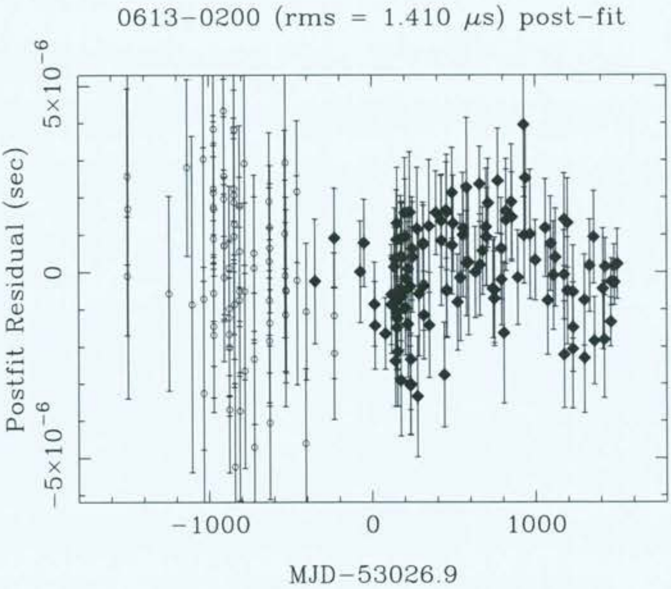

Figure 2.13: Simulated timing residuals for PSR J0613−0200. These residuals were created using the PSD_SIMULATOR plugin, where the simulated data includes the same sampling, error bar sizes and low-frequency noise as the observed residuals shown in Figure 2.8. The filled diamonds are the residuals resulting from simulated observations taken with the CPSR2 observing backend system; open circles show the residuals resulting from simulated observations using other observing backend systems.

will vary as a function of time, but for most pulsar timing experiments the variation will be over such long timescales to cause negligible change in the GW signal (Lommen & Backer, 2001). TEMPO2 treats the simulation of an evolving SMBHB differently from that of a non-evolving SMBHB. We only consider non-evolving GW sources in this thesis. TEMPO2 assumes that non-evolving GW sources have zero eccentricity. This assumption is valid because binary systems tend towards zero eccentricity over a much shorter timescale than the orbital decay timescale (Peters, 1964).

For a non-evolving source of GWs, the GW-induced ToA perturbation at the Earth[28], $R_e(t)$, at time t is given by (Detweiler, 1979; Lee et al., 2011)

$$R_e(t) = \text{Real}\left\{ \int_0^t \frac{P_+ A_+(t) + P_\times A_\times(t)}{2(1-\gamma)} dt \right\} \tag{2.6}$$

[28] This will be a real number, because measured timing residuals are real-valued. However, the calculation of $R_e(t)$ is greatly simplified by including an imaginary part in the integrand in Equation (2.6). This imaginary part corresponds to another set of timing residuals that are out of phase with the real part and are not measured under typical observing conditions.

where $P_{+,\times}$ and γ are real-valued geometrical terms defined by

$$\begin{aligned} P_+ &= \left(\hat{k_p}\cdot\theta\right)^2 - \left(\hat{k_p}\cdot\phi\right)^2 & (2.7)\\ P_\times &= 2\left(\hat{k_p}\cdot\theta\right)\left(\hat{k_p}\cdot\phi\right) & (2.8)\\ \gamma &= \hat{k_p}\cdot\hat{k_g} & (2.9)\end{aligned}$$

where $\hat{k_p}$ is a unit vector directed from Earth to the pulsar and $\hat{k_g}$ is a unit vector directed from Earth to the GW source (so the GW propagates in the direction $-\hat{k_g}$). In equations (2.7) - (2.9), we have the following definitions

$$\begin{aligned} \hat{k_p}\cdot\theta &= \sin\theta_p\cos\theta_g - \cos\theta_p\sin\theta_g\cos(\phi_g-\phi_p) & (2.10)\\ \hat{k_p}\cdot\phi &= \cos\theta_p\sin(\phi_g-\phi_p) & (2.11)\\ \hat{k_p}\cdot\hat{k_g} &= \sin\theta_p\sin\theta_g + \cos\theta_p\cos\theta_g\cos(\phi_g-\phi_p) & (2.12)\end{aligned}$$

In these equations, we define (ϕ_p, θ_p) to be the right ascension (RA) and declination (Dec) of the pulsar, respectively, and (ϕ_g, θ_g) to be the RA and Dec of the GW source respectively.

Prior to this work, TEMPO2 was only capable of simulating GWs with real-valued polarisations; that is, a linear combination of A_+ and $A_\times$ with real coefficients. However, in general we expect SMBHBs to emit elliptically polarised waves (unless the SMBHB is exactly edge-on with respect to our line-of-sight; e.g., Blanchet et al., 1996). Elliptical polarisation requires the introduction of complex-valued coefficients of A_+ and $A_\times$.

The terms $A_{+,\times}$ in Equation (2.6) are given by

$$A_{+,\times}(t) = A_{+,\times}\mathrm{e}^{i(2\pi ft+\Phi_g)}\,, \tag{2.13}$$

where $2\pi f$ is the GW angular frequency at the Earth and Φ_g is a constant phase offset. Assuming that f is constant over the duration of the observations, we can continue from Equation (2.6) as follows:

$$\begin{aligned} R_e(t) &= \mathrm{Real}\left\{\frac{P_+A_+ + P_\times A_\times}{2(1-\gamma)} \times \left(\frac{e^{i(2\pi ft+\Phi_g)}-1}{i2\pi f}\right)\right\}\\ &= \frac{\{P_+\mathrm{Real}\,(A_+) + P_\times\mathrm{Real}\,(A_\times)\}\sin(2\pi ft+\Phi_g)}{4\pi f(1-\gamma)}\\ &+ \frac{\{P_+\mathrm{Imag}\,(A_+) + P_\times\mathrm{Imag}\,(A_\times)\}\{\cos(2\pi ft+\Phi_g)-1\}}{4\pi f(1-\gamma)} \qquad (2.14)\end{aligned}$$

The GW-induced ToA perturbation at the pulsar, $R_p(t)$, is the same except for an additional phase term due to the GW transit time between the pulsar and the Earth. This phase term is (adapted from Hobbs et al., 2009; Lee et al., 2011):

$$\Delta\Phi_g = (1 + \hat{k_p} \cdot \hat{k_g}) \frac{D_p 2\pi f}{c} \,, \tag{2.15}$$

where D_p is the (in general, unknown) distance from Earth to the pulsar and c is the vacuum speed of light, which is also the speed of the GW. Hence we can express $R_p(t)$ as

$$\begin{aligned} R_p(t) &= \frac{\{P_+ \mathrm{Real}\,(A_+) + P_\times \mathrm{Real}\,(A_\times)\} \sin(2\pi f t + \Phi_g - \Delta\Phi_g)}{4\pi f(1-\gamma)} \\ &+ \frac{\{P_+ \mathrm{Imag}\,(A_+) + P_\times \mathrm{Imag}\,(A_\times)\} \{\cos(2\pi f t + \Phi_g - \Delta\Phi_g) - 1\}}{4\pi f(1-\gamma)} \end{aligned} \tag{2.16}$$

Hence, the total ToA perturbation $R(t)$ induced by a GW passing the Earth and the pulsar is

$$\begin{aligned} R(t) &= R_e(t) - R_p(t) \\ &= \frac{\{P_+ \mathrm{Real}\,(A_+) + P_\times \mathrm{Real}\,(A_\times)\} \{\sin(2\pi f t + \Phi_g) - \sin(2\pi f t + \Phi_g - \Delta\Phi_g)\}}{4\pi f(1-\gamma)} \\ &+ \frac{\{P_+ \mathrm{Imag}\,(A_+) + P_\times \mathrm{Imag}\,(A_\times)\} \{\cos(2\pi f t + \Phi_g) - \cos(2\pi f t + \Phi_g - \Delta\Phi_g)\}}{4\pi f(1-\gamma)} \end{aligned} \tag{2.17}$$

The induced ToA perturbation in Equation (2.17) shows two distinct physical effects. The GW-induced ToA perturbation at the Earth, $R_e(t)$, is given in Equation (2.14) and is called the "Earth term". The GW-induced ToA perturbation at the pulsar is given in Equation (2.16) and is called the "pulsar term". Equation (2.17) has been implemented in the TEMPO2 GW simulation engine GWSIM.H.

We now use the techniques described in Section 2.4 to simulate data sets for a range of future timing array projects. In the next Chapter, we will investigate the sensitivity of these simulated data sets to individual GW sources that induce sinusoidal timing residuals as described above in Equation (2.17).

2.6 Simulated Timing Array Observations

For analysis in Chapter 3 and Chapter 5, we have simulated several PTA projects with different characteristics using the FAKE plugin, described in Section 2.4.1. Here, we assume that the residuals are consistent with white noise with equal error bars. These assumptions will be

Scenario Name	Number of Pulsars	Residual rms (ns)	Timespan of Observations (d)
Arecibo	1/5	10	1750 / 3500 / 5250
PPTA	20	100/500	1750 / 3500 / 5250
IPTA	40	20 @ 100 ns, 20 @ 500 ns	1750 / 3500 / 5250
SKA	20/100	10/100	1750 / 3500 / 5250

Table 2.3: Parameters used to simulate different PTA projects.

relaxed in later Chapters. The characteristics of each data set are given in Table 2.3. In every simulated set of timing residuals, one observation is taken every two weeks. In particular, 1750 days of observing produces 127 data points (including end-points), 3500 days produces 252 data points and 5250 days produces either 377 or 378 data points depending on the pulsar parameter file being used.

The characteristics of the "Arecibo" scenarios are intended to emulate the very precise timing but limited sky-coverage attainable with the radio telescope at the Arecibo Observatory. The characteristics of the "PPTA" scenarios illustrate possible data sets that may be obtained by the end of the project, though these data sets would serve equally well as a simulation of the EPTA or NANOGrav data sets. The characteristics of the "IPTA" scenario are chosen to show the large number of pulsars observed as part of the project, with some precisely timed pulsars. The "SKA" scenarios correspond to simulated observations with the proposed Square Kilometre Array telescope (SKA)[29]. While the SKA will improve timing precision and the number of observable pulsars, the exact characteristics of any PTA project using the SKA are hard to predict.

To obtain 100 pulsars for the simulated PTA observed with the SKA, it was necessary to read in pulsars from the pulsar catalogue that have properties consistent with those of the currently known MSPs. In particular, most MSPs are in binary systems so it will be necessary to fit for binary parameters for many of the SKA pulsars. All the pulsars with $P < 60\,\mathrm{ms}$ and $\dot{P} < 10^{-17}$ are plotted on a $P - \dot{P}$ diagram in Figure 2.14. 100 of these pulsars were chosen for the SKA simulation.

After choosing pulsars suitable for timing with the SKA, it was necessary to choose a standard list of pulsar parameters to include in a model for each of these pulsars. In subsequent processing, every parameter measured for each pulsar was included in the timing fit, except for:

[29] See `http://www.skatelescope.org/`.

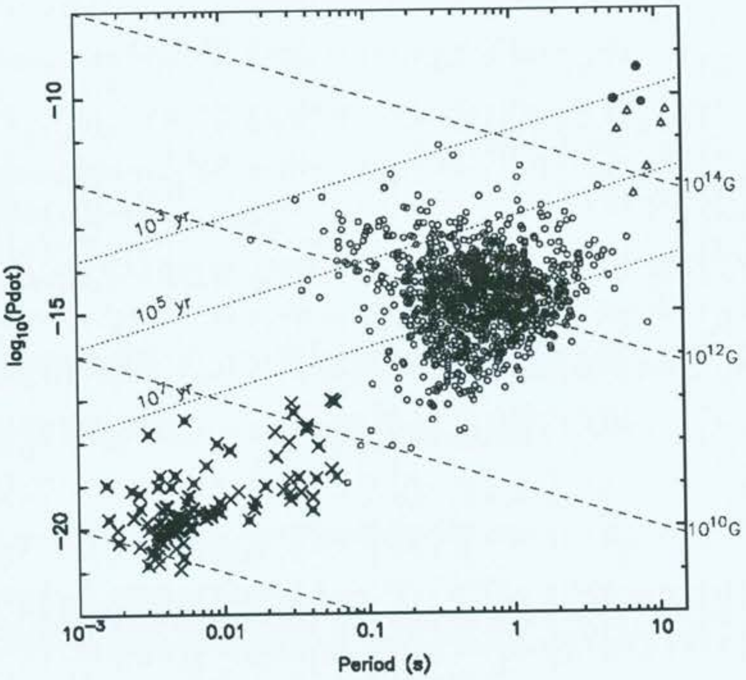

Figure 2.14: A $P - \dot{P}$ diagram showing pulsars used in the SKA simulation (crosses). Other symbols indicate the remaining known pulsar population.

- proper motion;
- dispersion measure (and any derivatives);
- 2nd and higher derivatives of rotational frequency;
- any post-Keplerian orbital terms, including the sine of the inclination angle, mass of the companion and the 1st derivatives of the longitude of periastron, the projected semi-major axis, the orbital period and the eccentricity;

For most pulsars used in the SKA simulation in Chapters 3 and 5, the timing parallax was also excluded from the timing model. However, the timing parallax fit was performed for those pulsars with a significant parallax measurement in the pulsar catalogue[30].

Having simulated a range of PTA scenarios, we will analyse their sensitivity to individual GW sources that induce sinusoidal residuals in Chapter 3. By determining the sensitivity of the different PTAs to such GW sources, it is possible to constrain the merger rate of SMBHBs as a function of redshift and chirp mass. Such constraints can be used to rule out predicted models for the formation and evolution of black-hole binaries.

[30] For the SKA simulation described in Section 4.3.4, the timing parallax was included in the timing model of every pulsar.

Chapter 3

Using Pulsar Timing to Detect Single Sources of Gravitational Waves Embedded in White Noise

Chapter Outline: *In this Chapter, we:*

- *describe a technique for detecting GWs that induce sinusoidally varying perturbations in the ToAs. This technique can only be applied to timing residuals that are consistent with white noise.*
- *apply this technique to simulated and real pulsar timing observations. This provides estimates of the sensitivity of different PTAs to individual sources of GWs that induce sinusoidal residuals.*
- *constrain the coalescence rate of SMBHBs using these GW sensitivity estimates.*

The results of this work (Section 3.3.4 below) were published in:
Wen Z. L., Jenet F. A., Yardley D., Hobbs G. B., Manchester R. N., 2011, ApJ, 730, 29

Some of the introductory work (Section 3.1) was published in §3.1 of:
Yardley D. R. B., Hobbs G. B., Jenet F. A., et al. 2010, MNRAS, 407, 669

In Chapter 2, we described methods for simulating pulsar timing observations (Section 2.4) and showed how GWs affect the timing residuals (Section 2.5). In this Chapter, we introduce a method to measure the sensitivity of pulsar timing observations to GWs from individual non-evolving SMBHBs. Such GWs will induce sinusoidal variations with known amplitude in the ToAs. Our algorithm can detect these sinusoidal variations, but can only be applied to timing residuals that are consistent with white noise. In our algorithm, we perform a Monte Carlo simulation of the ToAs to determine the strength of the sinusoid that is required to give a significant detection at each GW frequency. We then inject sinusoidal GW signals with different strain, h_s, and frequency, f, into the ToAs and measure the detection probability for each value of h_s and f. This process gives the sensitivity of the observations to an individual non-evolving SMBHB that could be in any position on the sky and emits GWs with arbitrary polarisation (see Section 1.3), and a frequency in the nHz to μHz range.

We apply our method to simulated observations from the PTAs that are described in Section 2.6. We also analyse a set of real timing observations presented by Jenet et al. (2006) and described in Section 2.3.1. The resulting estimate of GW-sensitivity as a function of frequency and GW strain places a constraint on the rate of coalescence of SMBHBs (Wen et al., 2011).

3.1 Gravitational Waves from Supermassive Black-Hole Binaries

In Section 2.5, we derived the timing residual induced by a GW. For this analysis, we assume that the total GW-induced ToA perturbation is a sinusoid. This assumption is based on two facts. The first is that, for most SMBHBs, the frequency of the signal will not vary significantly over the time-span of the observations. For an equal-mass binary, the lifetime of a SMBHB scales as (adapted from Lommen & Backer, 2001):

$$\tau = 5.1 \times 10^4 \left(\frac{M_1 + M_2}{10^9\,\mathrm{M}_\odot}\right)^{-5/3} \left(\frac{P_{\mathrm{orb}}}{1000\,\mathrm{d}}\right)^{8/3} \mathrm{yr}\,, \tag{3.1}$$

where M_1 and M_2 are the black-hole masses and P_{orb} is the orbital period[31]. For a SMBHB with $M_1 + M_2 = 10^9\,\mathrm{M}_\odot$ and $P_{\mathrm{orb}} = 1000\,\mathrm{d}$ (which would emit GWs with a 500 d period), the lifetime is four orders of magnitude larger than the typical data-span of pulsar timing observations. This means no significant chirping of the GW signal will occur over the duration of the observations. Therefore, we assume that the GW frequency is constant. In this case, Equation (2.17) can be used to calculate analytically the expected GW-induced residual.

The second fact is that, for most SMBHBs, the light travel-time from the pulsar to Earth is much smaller than the evolutionary timescale of the system. Evolution of the SMBHB over the timescale of the light travel time from the pulsar to Earth was measured in simulations of the proposed SMBHB in 3C66B (Jenet et al., 2004). This resulted in the sinusoid in the residuals caused by the Earth term exhibiting a higher-frequency than the sinusoid caused by the pulsar term (equations 2.14 and 2.16). We now determine whether such evolution will be significant for a typical SMBHB. The observed frequency, $f(t)$, of the GWs emitted by a SMBHB changes

[31] Note that for a binary in a circular orbit, $2P_{\mathrm{GW}} = P_{\mathrm{orb}}$, where P_{GW} is the period of the emitted GWs. This is because the space-time metric for this binary system will be identical at times t and $t + P_{\mathrm{orb}}/2$

at a rate (Peters & Mathews, 1963; Wen et al., 2011):

$$\frac{df}{dt} = \frac{96}{5}\left(\frac{GM_c}{c^3}\right)^{5/3} \pi^{8/3}(1+z)^{5/3} f^{11/3} \,, \tag{3.2}$$

where G is the gravitational constant, $M_c = (M_1 M_2)^{3/5}(M_1 + M_2)^{-1/5}$ is the chirp mass of the SMBHB, c the vacuum speed of light, and z the redshift of the SMBHB. If we assume that the orbital frequency is f_1 at time t_1, and is f_2 at time t_2, we can integrate this to obtain

$$\Delta t = \frac{15}{768}\left[f_2^{-8/3} - f_1^{-8/3}\right]\left(\frac{c^3}{GM_c}\right)^{5/3} \pi^{-8/3}(1+z)^{-5/3} \,, \tag{3.3}$$

since M_c and z are independent of time, and where $\Delta t = t_2 - t_1$. As an example, we consider the time taken for the observed GW frequency to shift by an amount equal to two frequency bins. If this is less than the light travel time from the pulsar to the Earth, then the sinusoidal pulsar term and the sinusoidal Earth term will have different frequencies, meaning our assumption that the GW-induced ToA perturbation is sinusoidal would be invalid. For $T_{\rm obs} = 5\,{\rm yr}$, where $T_{\rm obs}$ is the time-span of the observations of the pulsar, the frequency resolution is $1/5\,{\rm yr} \approx 6.34\,{\rm nHz}$. Therefore we set $f_2 = 50\,{\rm nHz}$ (following Sesana & Vecchio, 2010b) and $f_1 = 50 - 12.7\,{\rm nHz} = 37.3\,{\rm nHz}$. For $M_c = 10^{8.5}{\rm M}_\odot$ (a typical value for a resolvable SMBHB; Sesana et al., 2009; Sesana & Vecchio, 2010b) and $z = 0$, we obtain $\Delta t = 5 \times 10^3\,{\rm yr}$. A typical pulsar distance for a PTA pulsar is 1 kpc, giving a light travel time of approximately $3 \times 10^3\,{\rm yr}$, which is less than Δt. Hence, we ignore this longer timescale evolution, meaning that, in our model, the Earth term and the pulsar term always have the same frequency. However, we have allowed the two periodicities to be offset in phase. This alters the amplitude and phase of the signal in the timing residuals. We hence reduce the problem of detecting ToA perturbations induced by a non-evolving circular binary system to the problem of identifying the presence of a significant sinusoid in the timing residuals. To confirm that such a sinusoid is caused by GWs, one would need to ensure that the expected GW signature is present in the timing residuals of all pulsars (Equation 2.17).

To determine the signal that a particular SMBHB will induce in our timing residuals, we begin with the expected GW strain averaged over all orbital orientations of the binary, h_s, for an individual SMBHB (Thorne, 1987):

$$h_s = 4\sqrt{\frac{2}{5}}\frac{(GM_c)^{5/3}}{c^4 D(z)}\left[\pi f\,(1+z)\right]^{2/3} \ , \tag{3.4}$$

where $D(z)$ is the comoving distance to the SMBHB, given by

$$D(z) = \frac{c}{H_0}\int_0^z \frac{dz'}{E(z')} \ , \tag{3.5}$$

where $E(z) = H(z)/H_0 = \sqrt{\Omega_\Lambda + \Omega_m(1+z)^3}$ under a spatially flat ΛCDM cosmological model (White & Rees, 1978). For our analysis we assume $\Omega_\Lambda = 0.7$ (e.g., Komatsu et al., 2009), giving $\Omega_m = 0.3$.

Using equations (2.17) and (3.4), we can calculate the amplitude, $A_{\rm res}$, of the sinusoidal perturbation induced in the ToAs by a non-evolving SMBHB. The result is (Jenet et al., 2004, and references therein):

$$A_{\rm ToA} = \frac{h_s}{2\pi f}(1+\cos\theta)\sin(2\phi)\sin\left[\frac{\pi f D_p(1-\cos\theta)}{c}\right] \ , \tag{3.6}$$

where $2\pi f = 2\pi/P_{\rm GW}$ is the GW angular frequency, θ is the angle between the direction from which the GWs emanate and a vector from the Earth to the pulsar, ϕ is the GW polarisation angle and D_p is the (usually unknown) distance to the pulsar[32]. Equation (3.6) implies that the signal amplitude in a pulsar GW detector depends on the location of the GW source. For instance, GWs propagating along the line of sight from the Earth to the pulsar will not induce a measurable sinusoid in the timing residuals.

However, the observed timing residuals can be significantly different to the GW-induced ToA perturbations after the fitting process has been carried out. Figure 3.1 shows the effect this can have on GW detection – a GW signal with a period of one year (top left panel) will be almost completely removed after fitting (top right panel) because this signal mimics an error in the pulsar position. A GW signal with a period of two years (bottom left) is only slightly attenuated by fitting (bottom right). In order to simulate realistic post-fit residuals in the presence of a GW signal, we add the GW-induced perturbation directly to a set of ideal ToAs as described in Section 2.4.1. We then perform the standard pulsar timing fitting procedure on these modified ToAs to determine the timing residuals.

[32] For the nearest PPTA pulsars, D_p can be measured using the parallax distance.

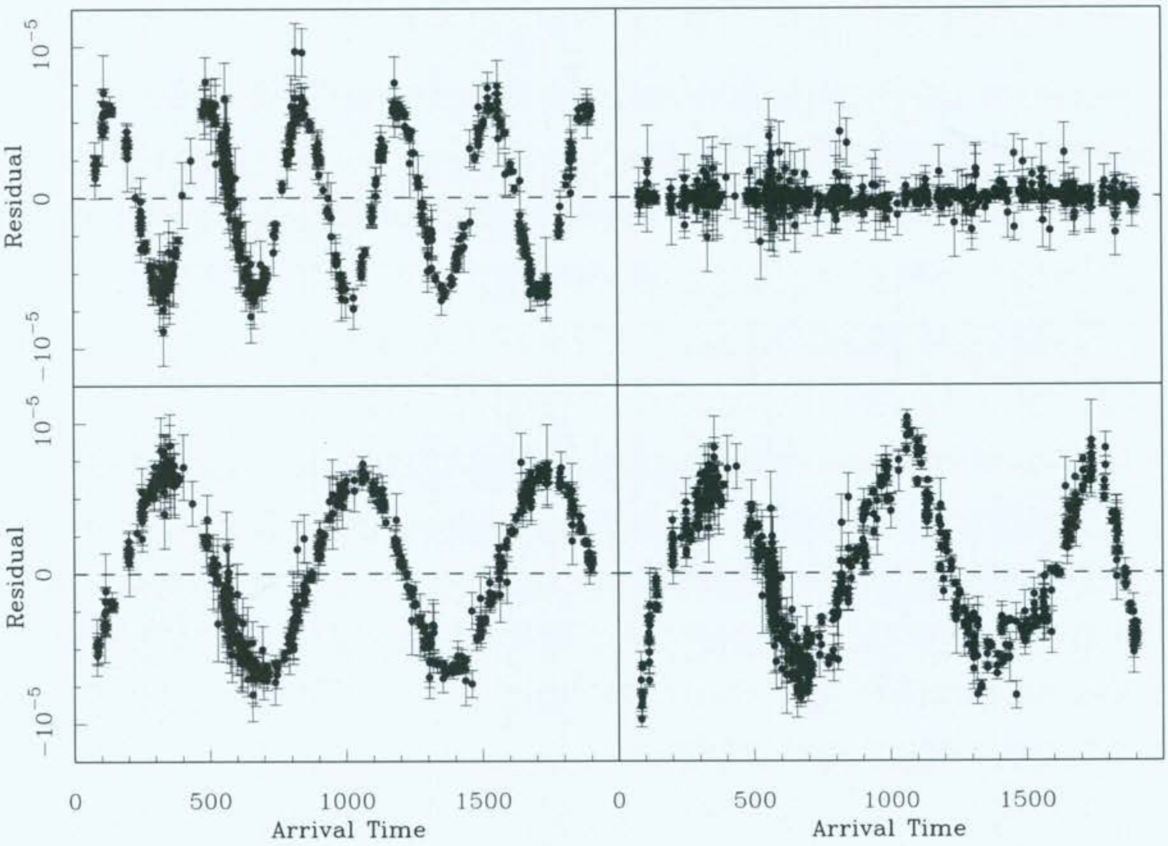

Figure 3.1: Attenuation of GW signals in timing residuals caused by pulsar parameter fitting. In each panel the abscissa is the MJD and the ordinate gives the timing residual in seconds. The dashed lines indicate zero residual. The plotted residuals are formed by adding a simulated GW signal to the timing observations for PSR J1909−3744 that are described in Section 2.3.2 and performing the TEMPO2 timing model fit. The top row shows a GW signal with a period of one year (top left) being completely removed after fitting for the pulsar timing model (top right). The bottom row shows a GW signal with a period of two years (bottom left) being largely unaffected by the fitting procedure (bottom right).

We now describe an algorithm for detecting the presence of this GW-induced perturbation in the timing residuals. In brief, the algorithm determines the sensitivity of any set of white timing residuals from a PTA to the GW signals from individual non-evolving SMBHBs.

3.2 Calculating the Sensitivity of a Pulsar Timing Array to Individual Non-evolving Sources of Gravitational Waves

The detection of a sinusoid in the presence of noise with known statistics is a well-studied problem with a simple optimal solution, the maximum likelihood estimator[33]. A number of algorithms can be used, depending on the characteristics of the data. For our analysis in this Chapter, we assume that each time series of residuals is consistent with white noise with varying error bars and irregular sampling.

To detect the GW-induced sinusoid in the timing residuals, we use one of the most common spectral estimation tools: a normalised Lomb-Scargle periodogram (Lomb, 1976; Scargle, 1982; Press et al., 1992). The periodogram is normalised by the variance of the input timing residuals. When processing multiple pulsars, we add the power measured in each frequency channel to form a "summed periodogram". Note that normalizing each power spectrum by the variance of the residuals is equivalent to weighting each power spectrum by the inverse variance when summing. Our "detection statistic" is the power level, P_i, in some frequency channel i in the summed periodogram. If P_i exceeds the detection threshold T_i, then a detection of a sinusoid has been made. We quantify the significance of this sinusoid using the "false alarm probability", $\mathcal{P}_f$. The false alarm probability gives the probability that a detection is recorded by our algorithm when no signal is present. For our analysis we use $\mathcal{P}_f = 0.001$.

We now describe analytic approximations of the detection thresholds. These analytic approximations are only valid for equally-spaced samples of white noise with constant variance. To obtain detection thresholds for our pulsar timing data sets, we use Monte Carlo simulations described in Section 3.2.2.

[33] E.g., Chapter VII, Section 9 in (Mood et al., 1974)

3.2.1 Analytical Calculation of Detection Thresholds

We analytically determine the threshold power level for a given $\mathcal{P}_f$ for a set of $N_{\rm psr}$ time series of equally-spaced white noise. The power, P_i, in frequency channel i can be written as

$$P_i = \sum_{p=1}^{N_{\rm psr}} \left(R_{i,p}^2 + I_{i,p}^2\right) , \tag{3.7}$$

where $R_{i,p}$ and $I_{i,p}$ are the real and imaginary parts of the DFT of the p-th time series, respectively (see Equation 2.4). $R_{i,p}$ and $I_{i,p}$ are independent, normally-distributed random variables, so P_i is distributed as a χ^2 random variable with $2N_{\rm psr}$ degrees of freedom. If we assume $N_{\rm psr} = 1$, the cumulative distribution function (cdf) of P_i has a simple form:

$$F_{\rm cdf}(P_i) = 1 - {\rm e}^{-P_i/2} . \tag{3.8}$$

Therefore, for an individual time series, the probability that the value of P_i is less than some threshold $T_{1,i}$ is $1 - {\rm e}^{-T_{1,i}/2}$. The probability that P_i exceeds $T_{1,i}$ is therefore $\mathcal{P}_{f,1} = {\rm e}^{-T_{1,i}/2}$. We can express $T_{1,i}$ as a function of $\mathcal{P}_{f,1}$ as

$$T_{1,i} = -2\ln(\mathcal{P}_{f,1}) . \tag{3.9}$$

For $\mathcal{P}_{f,1} = 0.001$, this yields $T_{1,i} = 13.8$ for all i.

Equation (3.9) does not account for the fact that a false detection can occur at any frequency in the power spectrum of the timing residuals. With real data, we calculate the summed power spectrum of $N_{\rm psr}$ time series and, as the frequency of a possible GW signal is usually unknown, search for significant power at any frequency. For the case $N_{\rm psr} = 1$, the probability that P_i is less than some threshold T_i for all i is $\left(1 - {\rm e}^{-T_i/2}\right)^{N_{\rm chan}}$, where $N_{\rm chan}$ is the number of independent frequency channels in the power spectrum. For a time series consisting of $N_{\rm pts}$ measurements of white noise, we have $N_{\rm chan} = N_{\rm pts}/2$ (e.g., Scargle, 1982). Hence, the probability, $\mathcal{P}_f$, that $P_i \geq T_i$ for at least one value of i is

$$\mathcal{P}_f = 1 - \left(1 - {\rm e}^{-T_i/2}\right)^{N_{\rm pts}/2} . \tag{3.10}$$

Rearranging (3.10) yields the detection threshold T_i as a function of $\mathcal{P}_f$:

$$T_i = -2\ln\left\{1 - (1 - \mathcal{P}_f)^{1/(N_{\rm pts}/2)}\right\} , \tag{3.11}$$

where $\mathcal{P}_f$ represents the probability of a false detection at any frequency in the power spectrum. Using $\mathcal{P}_f = 0.001$ means that, in a given data set, any detection is made with greater than 3-σ confidence.

However, when analysing pulsar timing residuals, equations (3.9) and (3.11) cannot be directly applied to determine the detection threshold, for two reasons. First, the effect of parameter fitting on the post-fit timing residuals must be accounted for. Second, the data sets are irregularly sampled, meaning that it is difficult to determine the number of independent frequency channels in the DFT. Hence, equations (3.9) and (3.11) have been used in our analysis only to confirm the accuracy of simulated estimates of T_i when $N_{\rm psr} = 1$. For $N_{\rm psr} > 1$, we confirmed the accuracy of simulated estimates of $T_{1,i}$ using computations of the cdf of a χ^2-distribution with $2N_{\rm psr}$ degrees of freedom.

3.2.2 Calculating Detection Thresholds via Monte Carlo Simulation

To calculate the detection thresholds using simulations, we must be able to create simulated timing residuals for each pulsar that are statistically equivalent to the input timing residuals. For timing residuals that are consistent with white noise, we create statistically equivalent sets of timing residuals by randomly rearranging, or "shuffling", the input residuals for each pulsar. Any shuffled set of timing residuals for a particular pulsar will have the same sample times, error bars, mean and variance as the input residuals. An advantage of this shuffling technique is that it assumes nothing about the distribution of the timing residuals, it simply re-orders them. We can thus calculate detection thresholds for white timing residuals from $N_{\rm psr}$ pulsars using the following procedure.

1. We calculate the ideal ToAs (defined in Section 2.4.1) for each pulsar in the input data set;

2. We create 10^5 sets of observations by shuffling the residuals for each pulsar[34] and adding them to the ideal ToAs;

3. We carry out the TEMPO2 pulsar parameter fit for each realisation to create 10^5 sets of post-fit timing residuals that are statistically equivalent to the input residuals;

[34] This assumes that there are at least nine timing residuals for the pulsar, because $8! < 10^5$. As typical data sets have ~200 observations, the 10^5 data sets are independent.

4. We calculate the normalised Lomb-Scargle periodogram of each shuffled set of post-fit residuals. For our analysis, the periodogram spans frequencies between $1/(30\,\mathrm{yr})$ and $1/(14\,\mathrm{d})$. To sample this frequency range in the periodogram, we must over-sample each pulsar's periodogram by a factor of $30\,\mathrm{yr}/T_{\mathrm{obs},p}$, where $T_{\mathrm{obs},p}$ is the time-span of the observations of pulsar p.

5. We add the periodograms obtained for each pulsar in each data set, giving 10^5 summed periodograms;

6. In each frequency channel in the summed periodogram, we find the 100th-highest power level across the 10^5 power estimates for that channel. This is the detection threshold, $T_{1,i}$, corresponding to $\mathcal{P}_{f,1} = 0.001$ in that frequency channel.

7. We increase $T_{1,i}$ in each frequency channel by a fixed factor $\beta > 1$, such that there are 100 power estimates higher than $\beta T_{1,i}$ across all frequencies in the 10^5 summed periodograms. This higher threshold $T_i = \beta T_{1,i}$ gives a false alarm probability of $\mathcal{P}_f = 0.001$ for false detections at any frequency in the input data set.

In Figure 3.2, we show the detection thresholds $T_{1,i}$ and $\beta T_{1,i}$ for a simulated set of pulsar timing observations of PSR J1713+0747. The simulated observations span 5250 d with one observation every two weeks. Simulating a data set with these parameters using the FAKE plugin to TEMPO2 produces 378 timing observations[35]. The pre-fit timing residuals are samples of white noise. In the detection thresholds, significant power is absorbed at a range of frequencies as described in Table 3.1. The left column gives the frequency range at which power is absorbed. Each absorption is caused by the TEMPO2 fit for the pulsar timing model parameters named in the right column.

Since the sampling interval in this set of timing residuals is $14\,\mathrm{d}$, the sampling frequency is $f_s = 1/\left(14\,\mathrm{d}\right)$. Thus, the Nyquist frequency is $f_{\mathrm{Nyq}} = 1/\left(28\mathrm{d}\right) = 0.0357\,\mathrm{d}^{-1}$. In an equally-spaced time series, every sinusoidal component with frequency f cannot be distinguished from a sinusoid with frequency $f_s - f$. This effect is known as "aliasing", and means that all sinusoids with frequency larger than f_{Nyq} are indistinguishable from a lower frequency equivalent. As we have sampled our power spectra up to twice the Nyquist frequency, our power spectra exhibit

[35] With input parameters of $T_{\mathrm{obs}} = 5250\,\mathrm{d}$ and a sampling interval of 14 d, FAKE produces 378 data points spanning $5263.7\,\mathrm{d}$ with a sampling interval of $13.96\,\mathrm{d}$.

Table 3.1: Causes of significant power absorptions in the thresholds in Figure 3.2.

Frequency (d^{-1})	Cause
$f_i < 0.0004$	period, period derivative
$f_i = 0.003$	sky-position
$f_i = 0.006$	parallax
$f_i = 0.015$	binary orbit period
$f_i = 0.029$	other binary parameters[36]

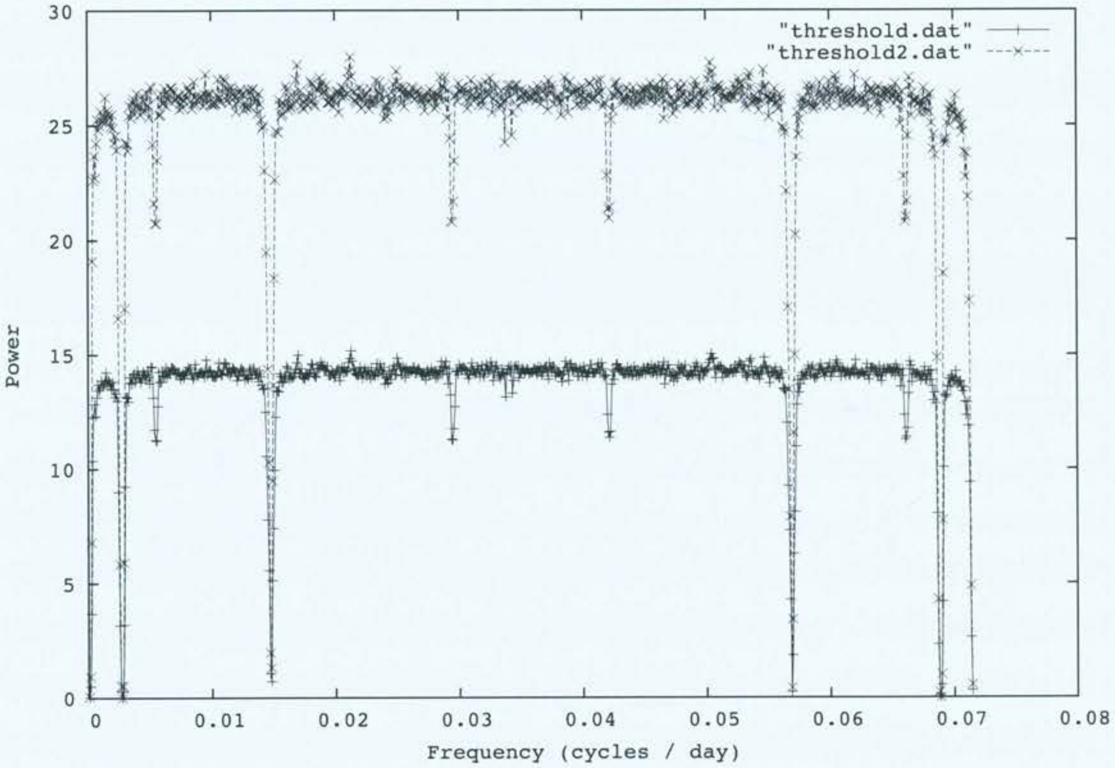

Figure 3.2: Detection thresholds for a single pulsar corresponding to $\mathcal{P}_{f,1} = 0.001$ and $\mathcal{P}_f = 0.001$ for 378 simulated observations sampled once every two weeks. The lower trace (solid line connecting '+' symbols) indicates $T_{1,i}$. The upper trace (dashed line connecting '×' symbols) indicates T_i. Both thresholds shown here have been calculated by simulation (see Section 3.2.2).

aliasing effects. This explains why power absorptions occur in pairs in Figure 3.2, reflected around the frequency $f \approx 0.036\,\mathrm{d}^{-1}$.

$T_{1,i}$ is plotted as the lower detection threshold in Figure 3.2. The value of $T_{1,i}$ agrees with the expected value from Equation (3.9) in frequency channels that are negligibly affected by the TEMPO2 fitting procedure. This agreement has been shown to hold for the simulated detection thresholds for data sets including up to 100 pulsars. The upper detection threshold in Figure 3.2 represents $\beta T_{1,i}$. The level of this threshold agrees with the prediction of Equation (3.11) within 10%. The discrepancy arises because the upper threshold calculated from simulation is calculated using the factor β. This factor essentially treats every frequency channel in the same way, whereas the TEMPO2 fitting affects the power in each channel in different ways.

3.2.3 Detecting Individual Non-evolving Gravitational-Wave Sources

Having obtained a set of detection thresholds for the observations, we inject simulated GW signals into simulations of the input data set and measure the number of such signals that we can detect. We use the following procedure to find the detection rate for a particular GW strain and frequency:

1. We simulate 10^3 sets of timing residuals for each pulsar that are statistically equivalent to the input timing residuals using the shuffling technique.

2. We add the effect of a GW point source with angular frequency $2\pi f_i$, amplitude h_s, random sky-position and random polarisation to the ToAs of every pulsar (see Equation 3.6). This induces sinusoidal ToA perturbations in each pulsar. The distance to each pulsar is assumed to be 0.91 kpc, which is the current best estimate of the distance to PSR J1857+0943 (Kaspi et al., 1994) and is typical for pulsars in the PPTA[37].

3. We perform the standard TEMPO2 pulsar parameter fit.

4. We calculate the periodogram for each pulsar's time series and add the periodograms to form the summed periodogram. If the summed power in channel i is greater than the detection threshold T_i in that channel, then the simulated GW signal has been detected.

5. We repeat the previous three steps for each of the 10^3 realisations of the input data set and find the detection percentage.

This process is repeated for 50 logarithmically-spaced values of h_s in the range $10^{-16} \leq h_s \leq 10^{-10}$ and 51 GW frequencies f_i. The 51 frequencies include 50 logarithmically-spaced values in the range $(30\,\mathrm{yr})^{-1} \leq f_i \leq (14\,\mathrm{d})^{-1}$ and the frequency $f_i = 1/1\,\mathrm{yr}$, enabling analysis of the effect of the pulsar position fit on our sensitivity to GW sources. The result is a "sensitivity

[37] Assuming that the distance to each pulsar remains larger than the GW wavelength, varying the distance to each pulsar would have little effect on the average detection rate in a Monte Carlo simulation because the GW source is non-evolving. In some individual realisations, the last factor in Equation (3.6) may be exactly zero for particular values of the pulsar distance.

matrix", which is a grid of the detection percentages for each GW strain and frequency. In Section 3.3 and Chapter 4, we assume a 95% detection probability, enabling us to plot the GW strain sensitivity of a PTA as a function of GW frequency.

3.3 Results and Discussion

In this Section, we present results from applying the method of Sections 3.2.2 and 3.2.3 to several PTA data sets. Each of the data sets analysed here is consistent with white noise. The simulated PTA observations are discussed in Section 2.6. In brief, the timing residuals obtained from the simulated PTA observations for each pulsar are equally-spaced and have equal error bars. In a given simulated PTA data set, each pulsar has the same rms residual. For the actual observations presented by Jenet et al. (2006) and summarised in Table 2.1, the ToA uncertainty varies for each observation, meaning that the timing residuals do not have equal error bars. The time-span of the observations and the variance of the residuals for each pulsar also vary.

First, we analyse a simulated PTA data set consisting of 20 pulsars timed with a rms of 500 ns over 10 yr. The results exhibit typical features of measurements of the GW sensitivity of pulsar timing measurements, such as sensitivity losses caused by pulsar parameter fitting. Second, we analyse several simulated PTA data sets to determine the dependence of the PTA sensitivity on the number of pulsars, the observing time-span and the rms residual. Third, we analyse the Jenet et al. (2006) observations (described in Section 2.3.1) and compare the results to those obtained for a simulated set of optimistic PPTA observations. Finally, we present the astrophysical implications of these results via the constraint that can be placed on the coalescence rate of SMBHBs (Wen et al., 2011).

3.3.1 Properties of the Sensitivity Curves

In Figure 3.3, we plot the 95% contour of the sensitivity matrix obtained for a simulated PTA data set, consisting of observations of 20 pulsars with a rms residual of 500 ns, with one observation taken every two weeks over 10 yr. This contour is referred to as a "sensitivity curve". There are several frequencies at which the sensitivity is significantly reduced, as described in Table 3.2. The left column gives the frequency range over which the sensitivity is reduced, while the right column describes the cause of the reduction. For comparison, the lowest frequency in the power spectrum of the timing residuals for this PTA simulation is $1/10\,\mathrm{yr} \approx 2.7 \times 10^{-4}\,\mathrm{d}^{-1}$.

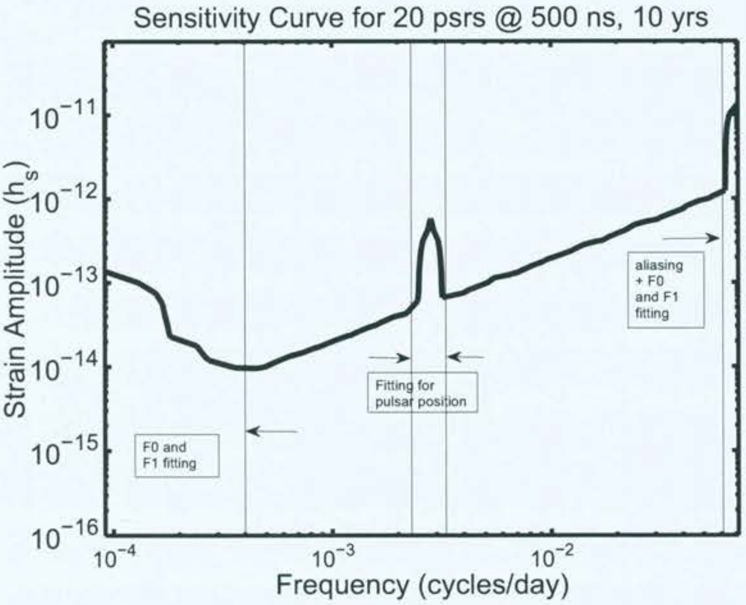

Figure 3.3: The sky-averaged sensitivity of a PTA data set (consisting of 20 pulsars timed with a rms residual of 500 ns over 10 yr) to individual non-evolving sources of GWs. The abscissa gives the observed GW frequency while the ordinate gives the strain amplitude of the GW source. The thick solid line indicates the level at which we detect 95% of the GW sources. The thin solid lines and arrows indicate the regions where the timing parameter fit reduces the sensitivity, as described in Table 3.2.

Table 3.2: Causes of significant sensitivity losses shown in Figure 3.3.

Frequency (d^{-1})	Cause
$f_i < 0.0004$	fitting for period & period derivative of every pulsar
$0.0023 < f_i < 0.0033$	fitting for sky-position of every pulsar
$f_i > 0.061$	aliasing + fitting for period & period derivative

The fitting and aliasing effects for this sensitivity curve, described in Table 3.2, are similar to those described in Table 3.1 for the detection thresholds of a single pulsar.

Fitting for the pulsar's sky-position causes a sensitivity loss over a *range* of frequencies[38]. This is because of the frequency sampling in the sensitivity matrix and the limited frequency resolution of the residuals. Figure 3.3 also shows that, in our technique, a GW-induced sinusoid in these simulated ToAs with $f = 1/1\,\mathrm{yr}$ and $h_s > 10^{-12}$ can be detected in the residuals, even after fitting for the pulsar sky-position.

The process of fitting for an individual pulsar's orbital parameters will reduce the sensitivity of its timing residuals to GWs, but this has only a small effect on the sensitivity of a PTA. This is because the estimation of the orbital parameters (such as the orbital period of a binary pulsar) removes a sinusoid with a different frequency for every pulsar (see Figure 2.3).

At frequencies that are negligibly affected by pulsar parameter fitting, such as the range $4 \times 10^{-3}\,\mathrm{d}^{-1} < f < 5 \times 10^{-2}\,\mathrm{d}^{-1}$, the sensitivity curve in Figure 3.3 has unit slope. This is because, as the GW frequency increases, the magnitude of the GW-induced sinusoid in the ToAs decreases according to Equation 3.6. This means that the amplitude of the sinusoid in the residuals decreases. If the timing residuals from the PTA are not consistent with white noise, or if the PTA data set includes sets of timing residuals with different time-spans, then this slope would not be constant across this frequency range. The maximum sensitivity is at a frequency of $f \approx 1/(0.7\,T_{\mathrm{obs}})$. This is because of the opposing effects of the pulsar parameter fitting (which reduces the amplitude of the sinusoid in the residuals at low frequencies) and the low frequency of the GW source (which increases the amplitude of the induced sinusoid).

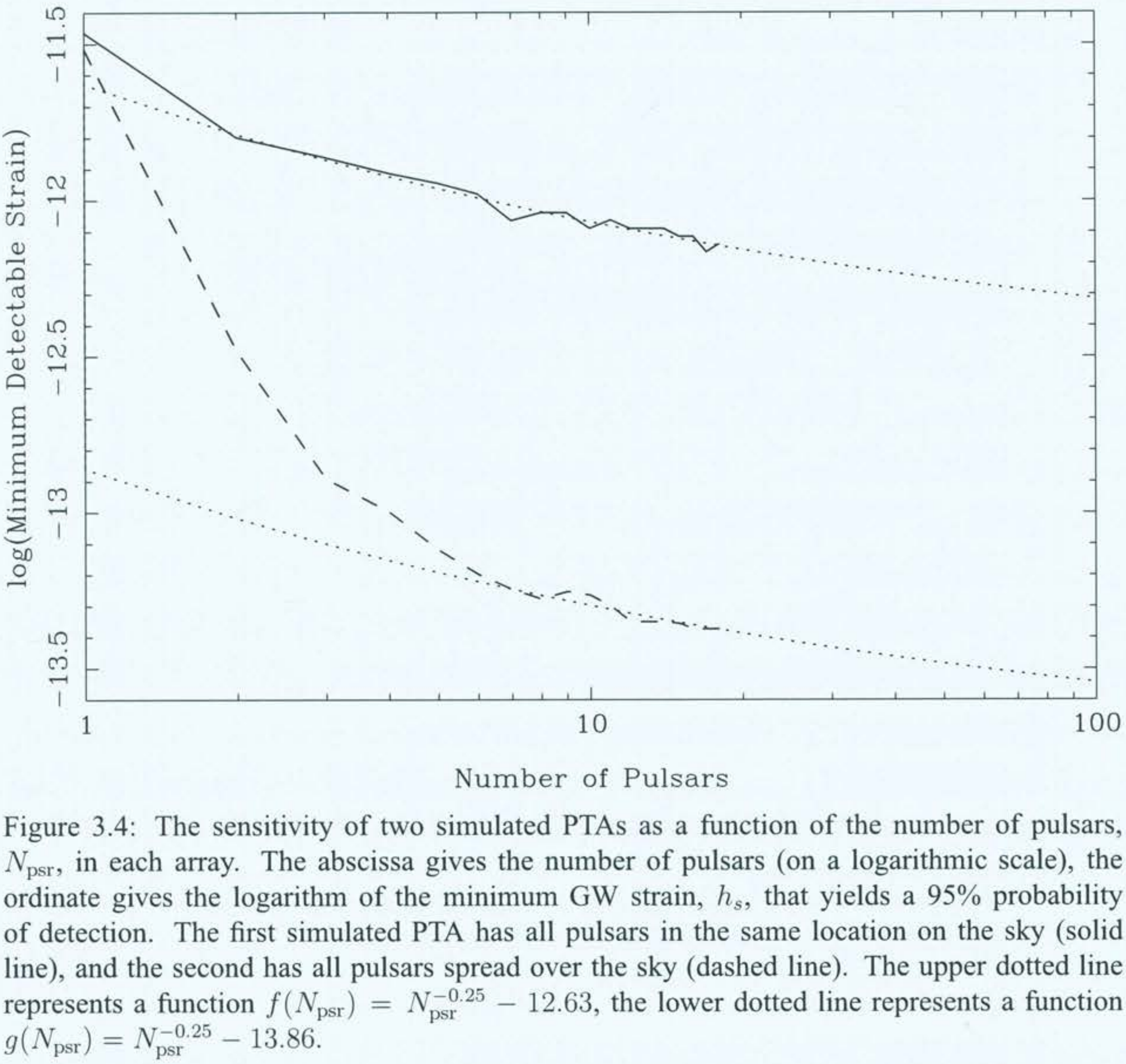

Figure 3.4: The sensitivity of two simulated PTAs as a function of the number of pulsars, N_{psr}, in each array. The abscissa gives the number of pulsars (on a logarithmic scale), the ordinate gives the logarithm of the minimum GW strain, h_s, that yields a 95% probability of detection. The first simulated PTA has all pulsars in the same location on the sky (solid line), and the second has all pulsars spread over the sky (dashed line). The upper dotted line represents a function $f(N_{\text{psr}}) = N_{\text{psr}}^{-0.25} - 12.63$, the lower dotted line represents a function $g(N_{\text{psr}}) = N_{\text{psr}}^{-0.25} - 13.86$.

3.3.2 Scaling of the Sensitivity with Properties of the Observations

In this Section, we investigate the dependence of the PTA sensitivity on $N_{\rm psr}$, $T_{\rm obs}$ and the rms residual. In Figure 3.4, we show the effect on the sensitivity caused by increasing $N_{\rm psr}$ and/or observing pulsars that are distributed evenly on the sky. The solid line shows the improvement in sensitivity obtained by adding more pulsars to the timing array that are all in the same position on the sky. The dashed line shows the improvement when observing pulsars that are spread across the sky. The improvement occurs because the sky-position of the GW sources is unknown (see Equation 3.6). However, the magnitude of the improvement depends on the specific PTA (Burt et al., 2011). For large $N_{\rm psr}$, the sensitivity improves as $N_{\rm psr}^{0.25}$, regardless of the distribution of pulsars on the sky. This is consistent with a recent estimate of the sensitivity of PTAs to individual sources of GWs (Lee et al., 2011)[39].

In Figure 3.5, we plot the sensitivity curves obtained for a simulated PTA observed using the SKA. For this analysis, we have assumed that the SKA will be able to observe 100 pulsars that are suitable for timing. Pulsar ToAs measured using the SKA are expected to be a factor of $\sim$100 more precise than current observations (Kramer & Wex, 2009). However, it is unlikely that timing precision will reach the $\sim$ns level because of pulse shape instabilities, calibration effects and other noise sources (Cordes et al., 2004). Hence, we have analysed two SKA simulations consisting of 100 observed pulsars. In one simulation, all pulsars are timed with a rms residual of 10 ns. In the other simulation, all pulsars are timed with a rms residual of 100 ns. The results of these simulations are analysed in Section 3.3.4 to estimate the constraint on the coalescence rate of SMBHBs that can be obtained with the SKA. They also enable us to investigate the dependence of the PTA sensitivity to individual GW sources on the rms timing residual and on the observing time-span.

The PTA sensitivity to a GW-induced sinusoid in the timing residuals is inversely proportional to the rms residual for residuals that are consistent with white noise (Lee et al., 2011). This is because the S/N ratio of the detection of the sinusoid will increase as the noise reduces. In Figure 3.5, the "minimum detectable amplitude" (defined as the value of h_s indicated by the

[38] This loss in sensitivity could be avoided in cases where pulsar sky-positions are measured using a technique other than pulsar timing, such as interferometry (Deller et al., 2008).

[39] These authors considered a coherent addition of the GW-induced sinusoidal signal from each pulsar, meaning that their measured sensitivity improves as $N_{\rm psr}^{0.5}$. Our analysis simply adds the power spectrum of each pulsar, meaning that the coherence of the GW-induced sinusoidal signals is lost. Hence, our measurement of the sensitivity improves only as $N_{\rm psr}^{0.25}$ for large $N_{\rm psr}$.

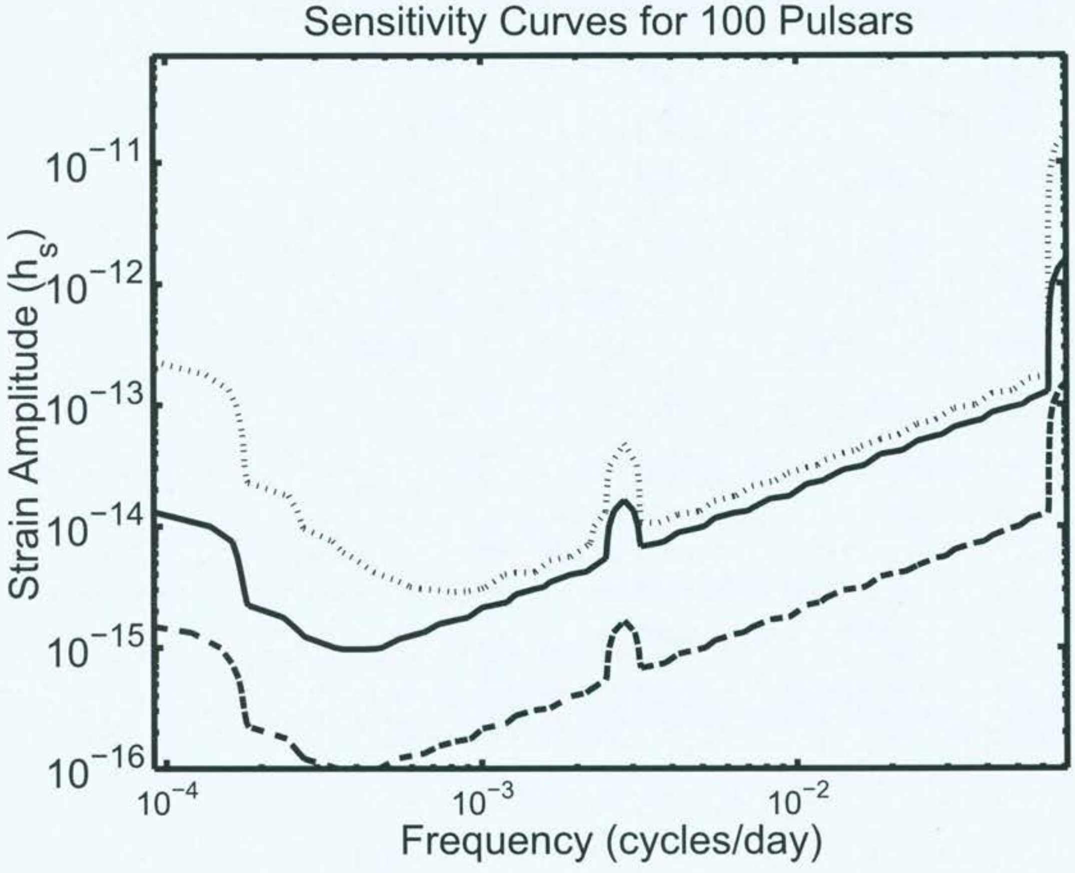

Figure 3.5: The sensitivity curves for three simulations of timing residuals obtained using the SKA. The abscissa gives the observed GW frequency while the ordinate gives the strain amplitude of the GW source. Each line indicates the level at which we detect 95% of the GW sources in that particular simulated data set. The three data sets considered are: 100 pulsars timed with 100 ns rms residual over 10 yr (solid line); 100 pulsars timed with 10 ns rms residual over 10 yr (dashed line); and 100 pulsars timed with 100 ns rms residual over five yr (dotted line).

sensitivity curve at a particular frequency) shown by the dashed line is a factor of 10 lower than that shown by the solid line. This indicates that, as expected, the sensitivity is improved by a factor of 10 when the rms residual is reduced by a factor of 10.

The minimum detectable amplitude is inversely proportional to $N_{\rm pts}^{0.5}$ (Scargle, 1982). In our case, increasing $N_{\rm pts}$ has the same effect on the S/N ratio of a significant sinusoid in the residuals[40] as increasing $T_{\rm obs}$. This is because the sampling interval is fixed at 14 d, meaning that $N_{\rm pts}$ is proportional to $T_{\rm obs}$. Figure 3.5 shows the sensitivity curves for observations of 100 pulsars timed with a rms residual of 100 ns over five yr (dotted line) and over 10 yr (solid line). As expected, increasing the time-span of the observations by a factor of 2 reduces the minimum detectable amplitude by $\approx \sqrt{2}$ for GW frequencies in the range $4 \times 10^{-3}\,{\rm d}^{-1} < f < 6 \times 10^{-2}\,{\rm d}^{-1}$. Other GW frequencies are significantly affected by the pulsar parameter fit. The figure also shows that doubling $T_{\rm obs}$ provides an even larger improvement in sensitivity at low frequencies as it decreases the lowest GW frequency that can be detected using the timing residuals. The PTA is more sensitive to lower frequency GWs, as they induce larger ToA perturbations for a fixed value of h_s (see Equation 3.6).

3.3.3 Sensitivity of the Jenet et al. (2006) Observations and a Prediction for the Full Parkes Pulsar Timing Array

The Jenet et al. (2006) observations differ in three ways from the simulated observations investigated in Sections 3.3.1 and 3.3.2. First, the observations are unequally spaced for each pulsar. In Figure 3.6, we plot the detection thresholds for the Jenet et al. (2006) observations. They do not exhibit the same symmetry about $f = 0.036\,{\rm d}^{-1}$ as the simulated observations, indicating that, as expected (Press et al., 1992), aliasing effects are insignificant in the Jenet et al. (2006) observations.

Second, the Jenet et al. (2006) observations have variable ToA uncertainties. This means that the TEMPO2 parameter fitting in the Monte Carlo simulations can be carried out using two approaches. One approach accounts for the error bar on each ToA by minimising the weighted variance of the residuals. The other approach ignores the ToA error bars by minimising the unweighted variance of the residuals. The two approaches lead to different estimates of the pulsar parameters and to different detection thresholds. In Figure 3.6, we plot the detection thresholds

[40] This is only true at frequencies that are not significantly affected by the parameter fit.

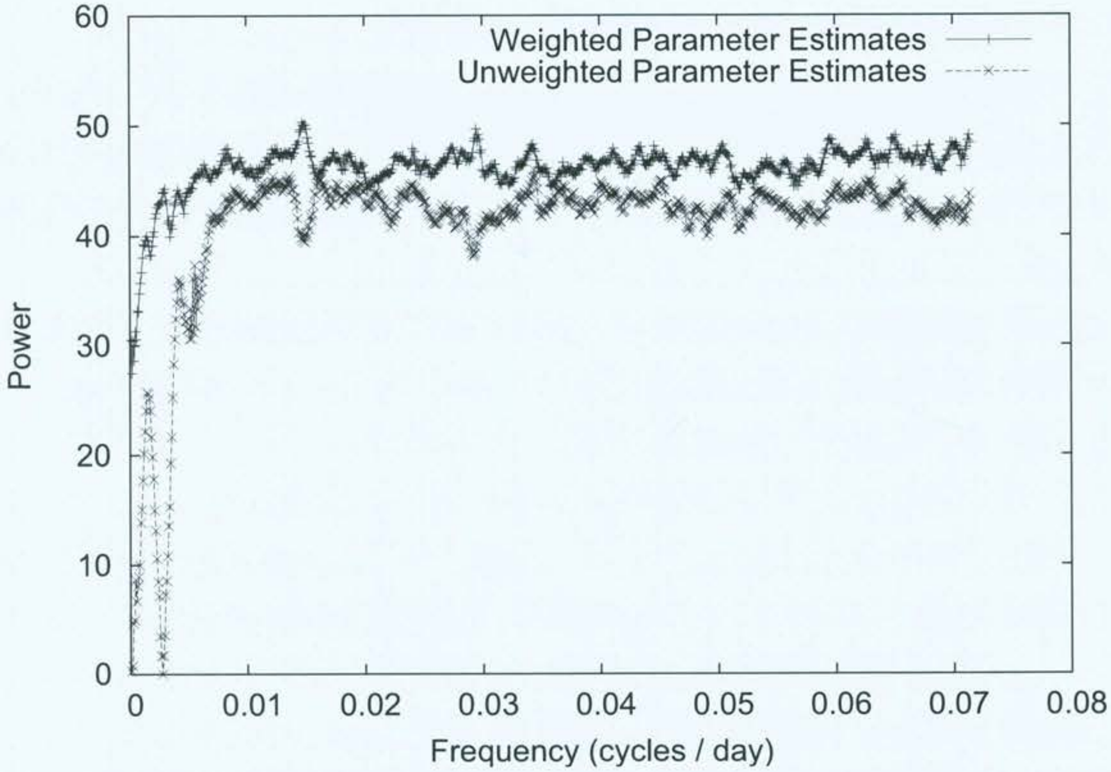

Figure 3.6: Detection thresholds corresponding to $\mathcal{P}_f = 0.001$ for the Jenet et al. (2006) observations. The abscissa gives the frequency while the ordinate gives the power level. The detection thresholds obtained when a weighted parameter fit was carried out (upper trace) do not show the power absorption features seen in the case of an unweighted parameter fit (lower trace).

obtained from each approach. When using a weighted TEMPO2 fit, the power reductions in the detection thresholds described in Table 3.1 are not present. This is not surprising, but adversely affects our sensitivity at these frequencies, as shown in Figure 3.7. When using an unweighted TEMPO2 fit, the detection thresholds are very low near $f = 1/1\,\mathrm{yr}$ and at low frequencies, as expected. This is because the Lomb-Scargle periodogram does not account for the error bar on each ToA when calculating each spectral estimate. We choose to use unweighted pulsar parameter estimates because our spectral estimate is also unweighted. In Figure 3.7 we analyse the sensitivity of both approaches.

Third, the Jenet et al. (2006) observations differ from the simulated observations as each pulsar has been observed over a different time-span. This affects the shape of the sensitivity curve, as can be seen by comparing the solid line in Figure 3.7 to the dashed line. The inclusion of long time-span observations of PSR J1857+0943 (spanning 20.3 yr) in the Jenet et al. (2006) data set improves the sensitivity to GWs in the frequency range $10^{-4}\,\mathrm{d}^{-1} < f < 10^{-3}\,\mathrm{d}^{-1}$ by a

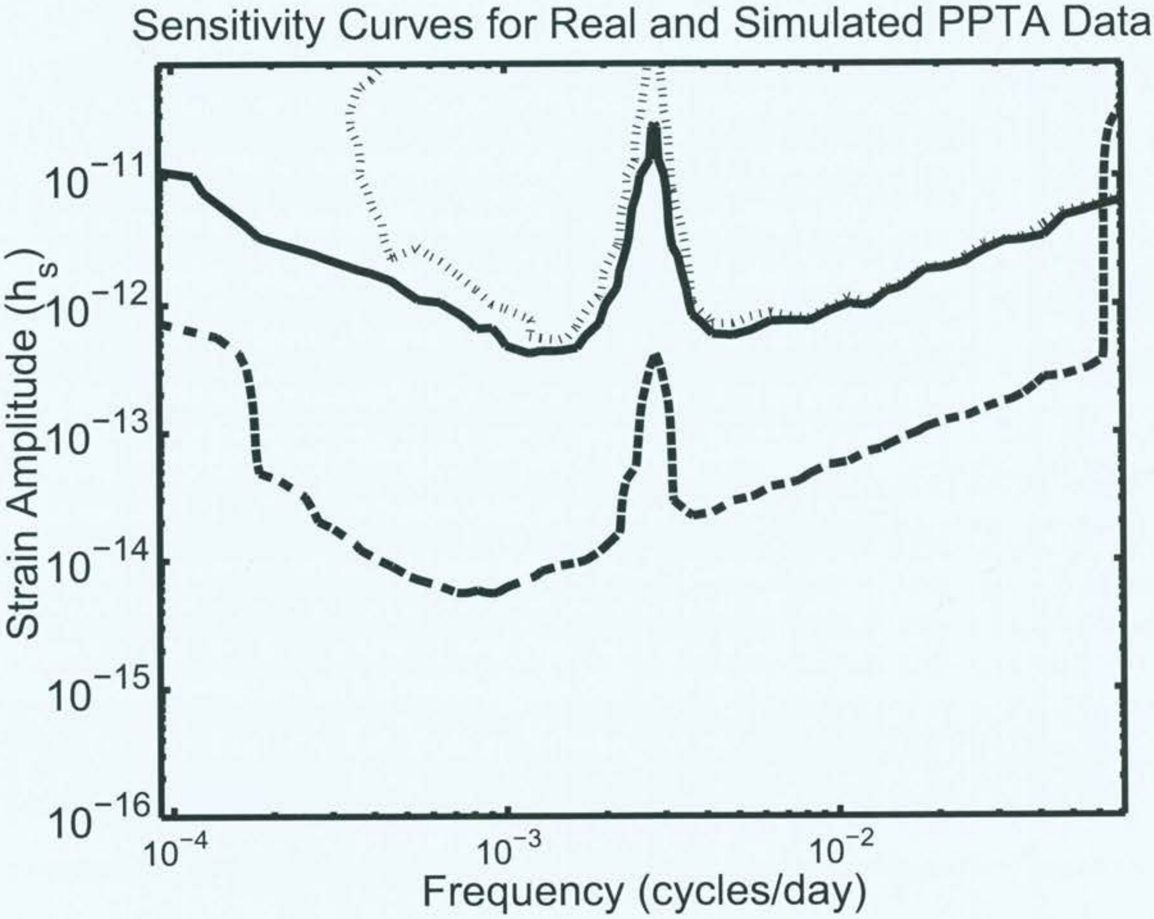

Figure 3.7: Sensitivity curves for real and simulated PPTA data sets. The abscissa gives the GW frequency, while the ordinate gives the GW strain. Each line indicates the level at which 95% of GW sources with any sky-position and polarisation can be detected in that set of observations. The solid line indicates the sensitivity of the Jenet et al. (2006) observations if we weight each ToA equally when estimating the timing model parameters for each pulsar. The dotted line indicates the sensitivity when we account for the varying ToA uncertainties. The dashed line shows the sensitivity for a simulation of a target PPTA data set, consisting of 20 pulsars timed with a rms of 100 ns over five yr.

factor of ~ 3. However, the PSR J1857+0943 observations have little effect on the sensitivity for $f > 10^{-3}\,\mathrm{d}^{-1}$ because the timing residuals for this pulsar have a larger rms variation than the residuals of the other pulsars.

The ToAs for the other six pulsars in the Jenet et al. (2006) data set have been measured over much shorter time-spans (spanning ~2.6 yr) and have significantly lower noise levels. Hence, as the GW frequency increases, the average induced sinusoidal signal becomes weaker (Equation 3.6) at the same time as the signal shifts into the detectable band for more of the pulsars. These two effects alter the sensitivity of the PTA at each frequency in opposite directions. This accounts for the much slower variation in the sensitivity of this data set with frequency, compared with the other data sets we have considered. The sensitivity curve that uses a weighted TEMPO2 parameter fit (the dotted line in Figure 3.7) has much lower sensitivity at low frequencies than the curve obtained from an unweighted fit. This is because our calculated detection thresholds are significantly higher when performing a weighted TEMPO2 parameter fit (Figure 3.6). Figure 3.7 also shows that the sensitivity curve that uses a weighted TEMPO2 parameter fit is multi-valued at frequencies near $f = 4 \times 10^{-4}\mathrm{d}^{-1}$. This is because the TEMPO2 weighted parameter fit is numerically unstable when very large sinusoidal signals are present in the ToAs that are not removed by this fit. As a result, the post-fit residuals can exhibit large noise levels that obscure the large GW signal in more than 5% of cases. This causes the detection percentage to drop below 95% for GW amplitudes larger than some threshold, meaning that the resulting sensitivity curve is multi-valued.

The most optimistic goal for the PPTA is the timing of 20 pulsars with a rms timing residual of 100 ns over five yr. The dashed line in Figure 3.7 shows the sensitivity curve obtained for a simulated data set with these properties. This simulated PPTA data set is a factor of ~15 more sensitive than the Jenet et al. (2006) data set. It is important to note that, when detecting single sources of GWs, a few very precisely timed pulsars are more likely to make a detection[41] than many pulsars with less precise timing (Burt et al., 2011). For example, the addition of a further 20 pulsars timed with a rms residual of 500 ns over five yr to the simulated PPTA data set makes negligible difference to the sensitivity to individual GW sources. However, such a data set would be very sensitive to the isotropic stochastic GWB, as will be described in Chapters 5 and 6.

[41] Recall that, while a single pulsar can be used to detect a significant sinusoidal signal, that signal can only be attributed to a GW if the expected correlated signal is observed in other pulsars.

3.3.4 Constraining the Coalescence Rate of Supermassive Black-Hole Binaries

It is proposed by Wen et al. (2011) that, if no GWs are detected in a given data set, then it is possible to place a constraint on the coalescence rate of SMBHBs. More specifically, we can constrain the quantity $[d^2R/d\log(M_c)d\log(1+z)]$, which gives the rate of coalescence, R, per logarithmic chirp mass interval, $d\log M_c$, per logarithmic redshift interval, where z is the redshift. A constraint on the coalescence rate of SMBHBs constrains the merger rate of galaxies and hence can rule out models of galaxy evolution (e.g., Jaffe & Backer, 2003).

The constraint on $[d^2R/d\log(M_c)d\log(1+z)]$ depends directly on the sensitivity matrix calculated in Section 3.2.3. In Figures 3.8 and 3.9, we show $[d^2R/d\log(M_c)d\log(1+z)]$ as a function of $\log(1+z)$ for chirp masses of $10^9\mathrm{M}_\odot$ and $10^{10}\mathrm{M}_\odot$. The Jenet et al. (2006) observations do not yet constrain the merging frameworks discussed by Jaffe & Backer (2003) or Sesana et al. (2008) at either of the chirp masses we have considered. As shown in Figure 3.8, an extended PPTA project, which times 20 pulsars with a rms residual of 100 ns over 10 yr, can constrain part of the Jaffe & Backer (2003) parameter space. However, only a PTA with a rms timing residual of 10 ns can provide significant constraints on the merger rate predicted by Sesana et al. (2008, 2009). In Chapter 5, we discuss the constraints obtained using upper bounds on the amplitude of the isotropic stochastic GWB. For a given set of white residuals, these upper bounds provide more significant constraints on the SMBHB coalescence rate than the sensitivity matrix for individual GW sources. The constraints obtained using upper bounds on the GWB will significantly constrain galaxy evolution models in the near future, without requiring timing accuracies near 10 ns on each pulsar.

3.4 Conclusion

We have presented a method for determining the sensitivity of a PTA to individual non-evolving GW sources. Such measurements constrain the coalescence rate of SMBHBs as a function of redshift and chirp mass. However, the technique presented in this Chapter has a few significant shortcomings:

1. The technique can only be applied to sets of timing residuals that are consistent with white noise. Many MSPs that are timed with sufficiently high precision over long time-

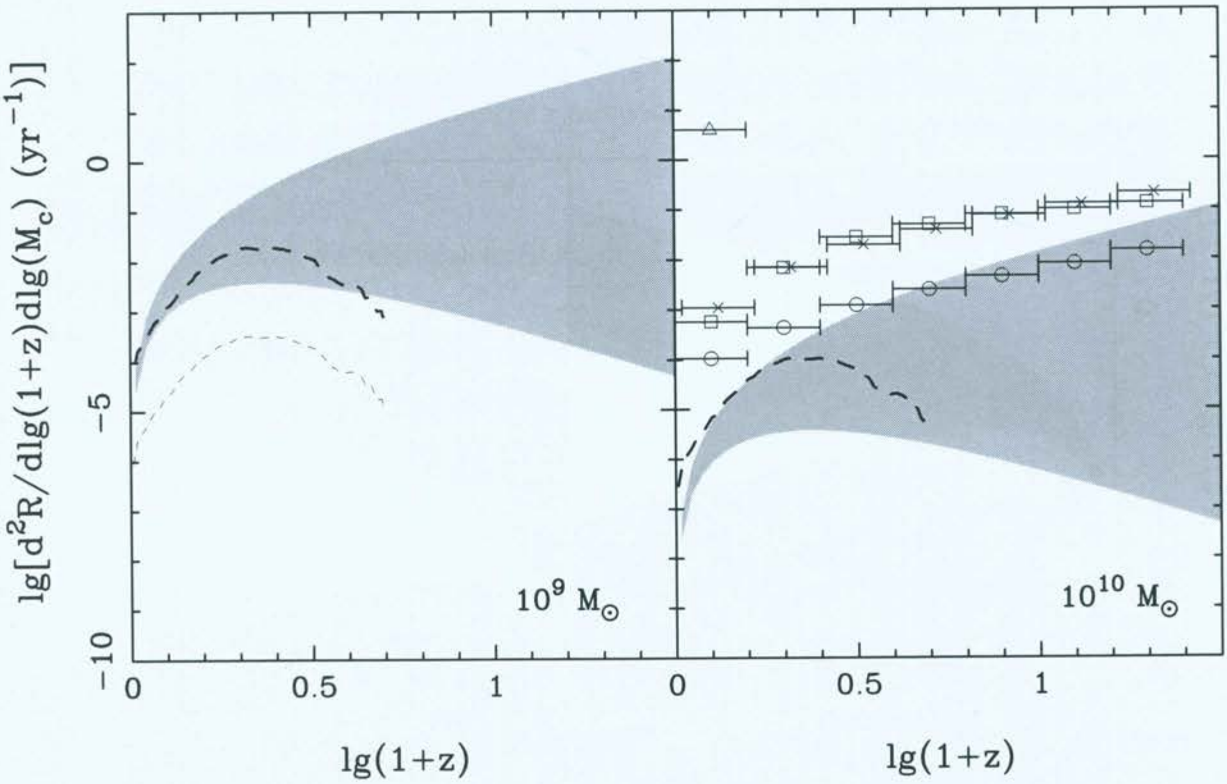

Figure 3.8: Upper bounds on the coalescence rate of SMBHBs using the sensitivity matrices calculated for different sets of PTA observations in Section 3.3. For calculating the abscissa, z is the redshift of the SMBHB. The ordinate gives the logarithm of the differential rate of coalescence per log redshift per log chirp mass. Here, we show the constraints provided by the Jenet et al. (2006) data set (open triangles), 20 pulsars timed with 500 ns rms residual over 10 yr (open squares), the same timed with 100 ns rms residual over five yr (crosses) and the same timed with 100 ns rms residual over 10 yr (open circles). The grey region indicates the expected coalescence rate with evolution index $-1 < \gamma < 3$ (see Section 1.6.2) assuming the framework of Jaffe & Backer (2003) and using observations from the Sloan Digital Sky Survey (Wen et al., 2009). The dashed traces indicate the maximum (thick line) and minimum (thin line) coalescence rates predicted by Sesana et al. (2008, 2009). No bounds can be plotted for chirp masses of $10^9\,\mathrm{M}_\odot$ because of the low sensitivity of these data sets. [Image reproduced from figure 4 of Wen et al. (2011)]

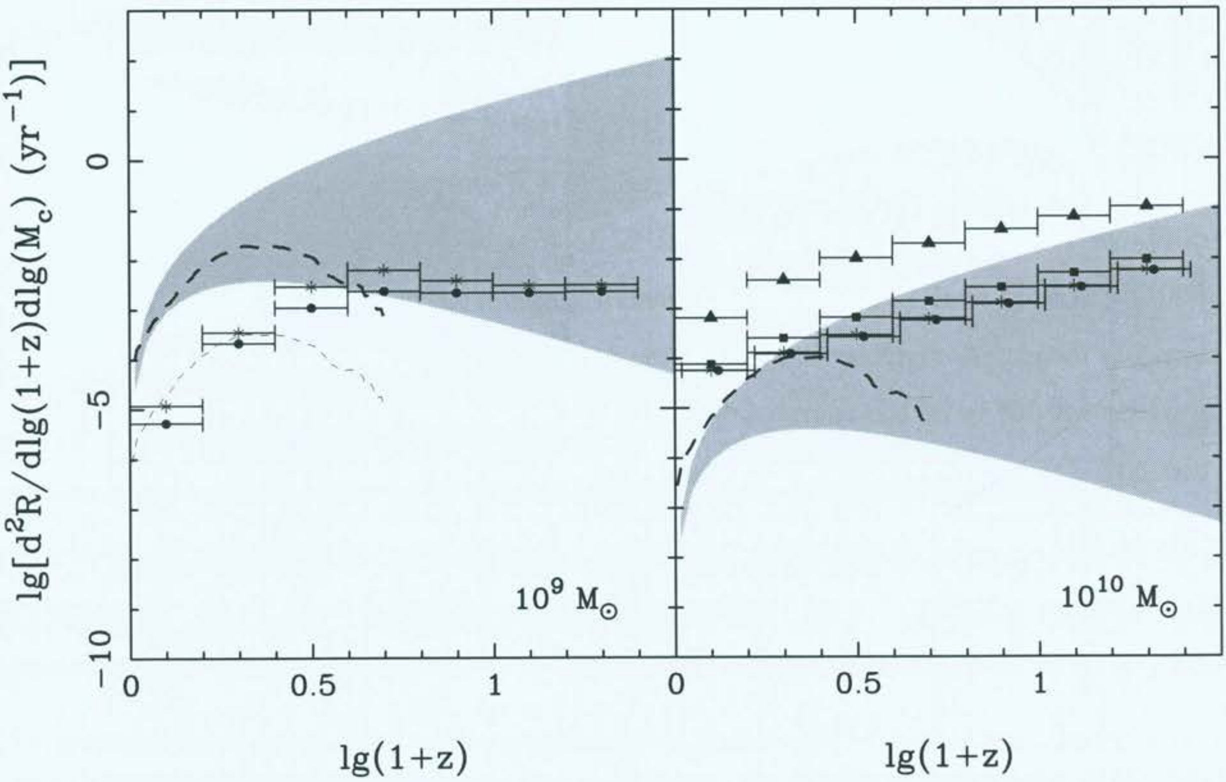

Figure 3.9: All characteristics of this plot are the same as in Figure 3.8, except that we now show the constraints obtained using 20 pulsars timed with a rms timing residual of 10 ns over 10 yr (stars), 100 pulsars timed with 10 ns rms residual over 10 yr (filled circles), the same timed with 100 ns rms residual over 10 yr (filled squares) and the same timed with 100 ns rms timing residual over five yr (filled triangles). [Image reproduced from figure 5 of Wen et al. (2011)]

spans show significant low-frequency noise in their timing residuals (Verbiest et al., 2009; Manchester, 2011). This low-frequency noise must be accounted for.

2. The technique uses the sensitivity matrix to constrain the coalescence rate of SMBHBs. However, this constraint only requires estimates of the largest GW signal that could be present in the timing residuals, as opposed to the smallest GW signal that could be detected using the residuals. Hence, a more stringent constraint could be found with the same observations.

In Chapter 4, we develop a related detection technique that addresses these issues.

Chapter 4

The Sensitivity of the Parkes Pulsar Timing Array to Individual Sources of Gravitational Waves

Chapter Outline: *In this Chapter, we:*

- *describe a technique that can detect single sources of GWs in non-white pulsar timing residuals;*
- *give the sensitivity of current and future GW detection experiments to single GW sources spanning frequencies from nHz to kHz;*
- *place a sky-averaged constraint on the coalescence rate of nearby ($z < 0.6$) SMBHBs.*

Many Sections in this Chapter are heavily based on sections from the refereed journal article:
Yardley D. R. B., Hobbs G. B., Jenet F. A., et al. 2010, MNRAS, 407, 669

In particular, Section 4.1 below is a summary of §2 of Yardley et al. (2010) and Sections 4.2, 4.3 and 4.4 below have been reworded from §3.2, §4 and §5 of Yardley et al. (2010) respectively. Section 4.2 below and Appendix A of this thesis contain material from the appendix of Yardley et al. (2010).

In this Chapter, we develop a new method for detecting individual non-evolving SMBHBs in the residuals obtained from PTA observations. While the technique presented in Chapter 3 can detect these sources, it assumes that the timing residuals being analysed are consistent with white noise. This assumption is only valid for a relatively small number of PTA data sets. Here, we extend the method of Chapter 3, allowing it to be applied to a broader range of MSP timing observations.

The method described in Section 4.2 below can be applied to most sets of timing residuals. Full details of the implementation are described in Appendix A. We apply this method to timing residuals from the PPTA published by Verbiest et al. (2008, 2009) and described in Section

2.3.2. This results in a measurement of the sensitivity of the PPTA to individual non-evolving GW sources as a function of frequency. The frequency range of the resulting sensitivity curve complements the frequency range of the LISA and LIGO GW-sensitivity curves. A constraint on the coalescence rate of nearby ($z \lesssim 2$) SMBHBs with chirp mass $\sim 10^{10}\,\mathrm{M}_{\odot}$ is determined to be less than one coalescence every five years.

4.1 Observations

The observations used in this analysis are a subset of those described in Section 2.3.2, and consist of observations of 18 pulsars[42] using the Parkes and Arecibo radio telescopes. Many of these pulsars exhibit a small amount of low-frequency noise in their timing residuals. These pulsars have been timed with a weighted rms residual, σ_w, in the range $0.17\,\mu\mathrm{s} < \sigma_w < 6.6\,\mu\mathrm{s}$ for a period of ~10 yr.

4.2 Calculating the Sensitivity Curve and Limit Curve

The detection of a sinusoid in the timing residuals is complicated by the fact that the residuals are irregularly sampled and the noise that affects the residuals consists of at least two components. The noise has a white component that varies from sample to sample. This component is well-understood and the square of the error bar gives a variance estimate for the white noise on each residual[43]. The noise also has a non-white component for which the source is unknown. The non-white noise could be due to calibration errors, timing noise intrinsic to the pulsar, a GWB signal or other effects. The spectrum of low-frequency noise in pulsar timing residuals is often modelled using a power-law (e.g., Hobbs et al., 2010b). We make the less-stringent assumption that the non-white noise has a smoothly-varying power spectrum. In all cases, we have estimated the power spectrum from the actual residuals and have shown that the noise can be modelled sufficiently well for our purposes using a smoothly-varying function.

We estimate the power spectrum using a Lomb-Scargle periodogram that, for this analysis, is not normalised by the variance of the residuals. This periodogram technique would not give

[42] We choose to remove the observations of PSRs J1824−2452 and J1939+2134 from our data set because their timing residuals are dominated by low-frequency noise. This low-frequency noise complicates the spectral analysis for little gain in sensitivity.

[43] The timing residuals analysed in Chapter 3 contain only this white component of the noise.

accurate spectral estimates for data sets that exhibit a steeply sloping spectrum. All the data sets used in this Chapter do not exhibit steep power spectra and so this technique is valid. We briefly describe our approach for producing a sensitivity curve here; full details are provided in Appendix A.

To make a detection of a significant sinusoid in our timing residuals, we make a simple model of the noise across all frequencies in the Lomb-Scargle periodogram of the residuals. This model is used to define a set of detection thresholds. These thresholds are set such that the probability of a false detection at any frequency across the entire observed periodogram when no signal is present is $\mathcal{P}_f = 1\%$. In practice, the detection thresholds are given by the noise model multiplied by some fixed factor that is determined from simulation, as described in Appendix A. We then add the effect of sinusoidal GW signals to the ToAs in the same manner as described in Section 3.2.3. We calculate the periodogram of the residuals and make a simple model of the noise. Using the technique described in Appendix A, we ensure that the signal that we aim to detect is not modelled as part of the noise in the periodogram. We adjust the GW strain until we can detect 95% of the GW-induced sinusoids in our timing residuals. This process gives the sky- and polarisation-averaged sensitivity as a function of GW frequency over the range $f \sim (10\,\mathrm{yr})^{-1}$ to $f \sim (10\,\mathrm{d})^{-1}$.

In Figure 4.1, we show the periodogram (thin trace) of the timing residuals for three pulsars where their ToAs are affected by a low-frequency GW source. We also show the noise models for each pulsar (thick line). Details on the calculation of these noise models are given in Appendix A.

There are two aspects to our detection strategy, namely the false alarm probability (1%) and the probability of making a detection (95%). Using $\mathcal{P}_f = 1\%$ means that any detection made will be a 2.6-σ detection. Hence, our sensitivity curves give the GW amplitude at which the probability of making a 2.6-σ detection at a random position on the sky for a randomly polarised GW is 95%. For a single pulsar, when the GW source has favourable sky-location and polarisation, the minimum detectable strain is a factor of $\sim 10 - 15$ smaller than the sky-averaged case (see Section 4.3.1).

We are interested in answering two questions. The first is "What is the largest GW source at a particular frequency that could be present in the timing residuals?" This will give an upper bound on the amplitude of non-evolving individual GW sources in our data set at that frequency.

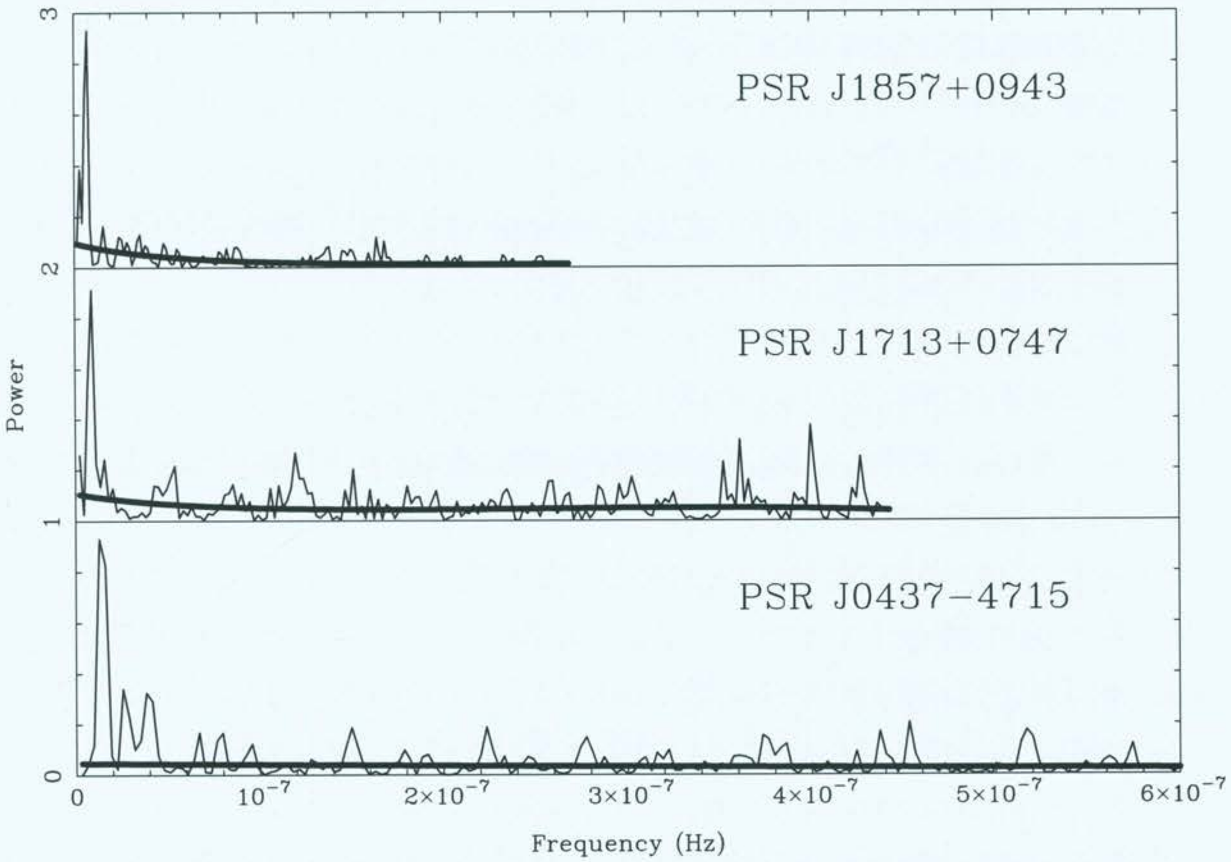

Figure 4.1: The periodogram of each of three sets of timing residuals, where we have added a low-frequency sinusoid to each set of ToAs. The abscissa gives the frequency, the ordinate gives the power in arbitrary units, where we include constant offsets in the periodograms of PSRs J1857+0943 and J1713+0747 to make this plot. The ordinate in each periodogram is scaled by independent values to make this plot. The thin trace is the periodogram, the thick dark line is the adopted model for each periodogram.

This question is answered by comparing simulated GW sources to our observed timing residuals. We simulate a GW source at a given frequency with a random sky-location and polarisation. We adjust the amplitude of this source until the power of the GW sinusoid exceeds the power in the observed timing residuals at that frequency in 95% of simulations. This approach gives the most conservative upper limit, since it allows for the possibility that *all* the power we observe at this frequency results from one GW-induced sinusoid. This process can be repeated to determine the upper limit as a function of frequency, yielding a "limit curve". We will determine the limit curve for our 18-pulsar data set in Section 4.3.

The second question is: "If there were a GW source with a particular frequency somewhere on the sky, what is the minimum strain amplitude that would produce a detectable signal at that frequency in our data set?" This is similar to the question that was addressed by the sensitivity curves in Section 3.3. To answer this question, we add simulated sinusoidal GW signals to our ToAs and perform the standard pulsar timing analysis. We then calculate the minimum amplitude at which we would detect a significant sinusoid at the input GW frequency in our data *if we had collected that data set at a telescope*. Hence, we must account for all the sources of noise in our pulsar detector[44]. The answer to this second question yields our sensitivity for detecting the GW-induced sinusoids, rather than just limiting their amplitude. For large amplitude sinusoids with period $\gtrsim T_{\rm obs}$, a signal will often be detectable at a higher frequency than the input frequency because we can detect the side lobes of the large input signal. In contrast to the approach of Chapter 3, we have not allowed detections at different frequencies to the input GW frequency in this implementation. The sensitivity curves for each of our data sets are calculated in Section 4.3.

The periodogram frequency range is from $\frac{1}{T_{\rm obs}}$ to $\frac{N_{\rm pts}}{2T_{\rm obs}}$ for a single pulsar[45]. Note that $\frac{N_{\rm pts}}{2T_{\rm obs}}$ would be the Nyquist frequency for that pulsar if its timing residuals were equally-spaced. If we are processing multiple pulsars then we can perform a weighted sum of their periodograms to increase our sensitivity. To perform the sum, we calculate the periodogram at a list of frequencies that is identical for all pulsars. The frequencies are equally spaced from $(30\,\rm yr)^{-1}$ to $(28\,\rm d)^{-1}$.

To perform the detection, we first make a simple frequency-dependent model of the noise in

[44] The threshold for detection at any frequency across the observed periodogram will often be ~ 3 times greater than the locally-averaged power level.

[45] The power at a frequency of zero is arbitrary for pulsar timing residuals.

the periodogram for each pulsar (see Figure 4.1) and then weight each pulsar by the inverse of the noise model for that pulsar[46]. This simple weighting scheme gives a factor of ~ 5 improvement in sensitivity over a simple, non-weighted addition of the periodogram of each pulsar.

4.3 Results and Discussion

We now present the sensitivity of the PPTA to GW-induced sinusoidal signals in the ToAs using the data set described in Section 4.1. We account for all the observed features in the sensitivity curves. We also calculate the constraint on the coalescence rate of SMBHBs implied by the non-detection of GWs in the ToAs. Finally, we give a prediction for the sensitivity of a future PTA project using the SKA.

4.3.1 The Sensitivity of Some Individual Pulsars

In Figure 4.2, we plot the sky- and polarisation-averaged sensitivity curves for PSRs J0437$-$4715 (thin solid line), J1713+0747 (dashed line) and J1857+0943 (dot-dashed line) where each pulsar has been analysed individually. The open triangles on the plot indicate that the plotted "detectable" amplitude at that frequency value is a lower bound. The thin dotted line indicates the sensitivity of PSR J0437$-$4715 to a hypothetical SMBHB located at a RA of $4^{\rm h}37^{\rm m}$ and a Dec of $+42^\circ 45^{\rm m}$ and emitting purely 'plus' polarised GWs. This line indicates the much greater sensitivity obtainable with the timing residuals of PSR J0437$-$4715 when the position and polarisation of the simulated GW source are favourable. The ratio of this thin dotted line to the thin solid line gives the factor of $\sim 10 - 15$ improvement in sensitivity for favourable sky-location and polarisation discussed in Section 4.2. Also shown are the expected signals at a range of frequencies from two hypothetical SMBHB systems at the mean distance of the Virgo cluster (taken to be 16.5 Mpc, from Mei et al., 2007), with equal member masses of $10^9 {\rm M}_\odot$ or $10^{10} {\rm M}_\odot$.

The reduction in sensitivity caused by fitting for the pulsar's position is at the same frequency of $(1\,{\rm yr})^{-1}$ for all pulsars. Fits for orbital parameters also reduce sensitivity to GWs, but at different frequencies for each pulsar. For example, the orbital period of PSR J1857+0943 is 12 days (corresponding to a frequency of 9.6×10^{-7} Hz), which is above the average Nyquist

[46] For spectrally-white timing residuals, this is equivalent to weighting by the inverse variance of each set of residuals, as done in Section 3.2.

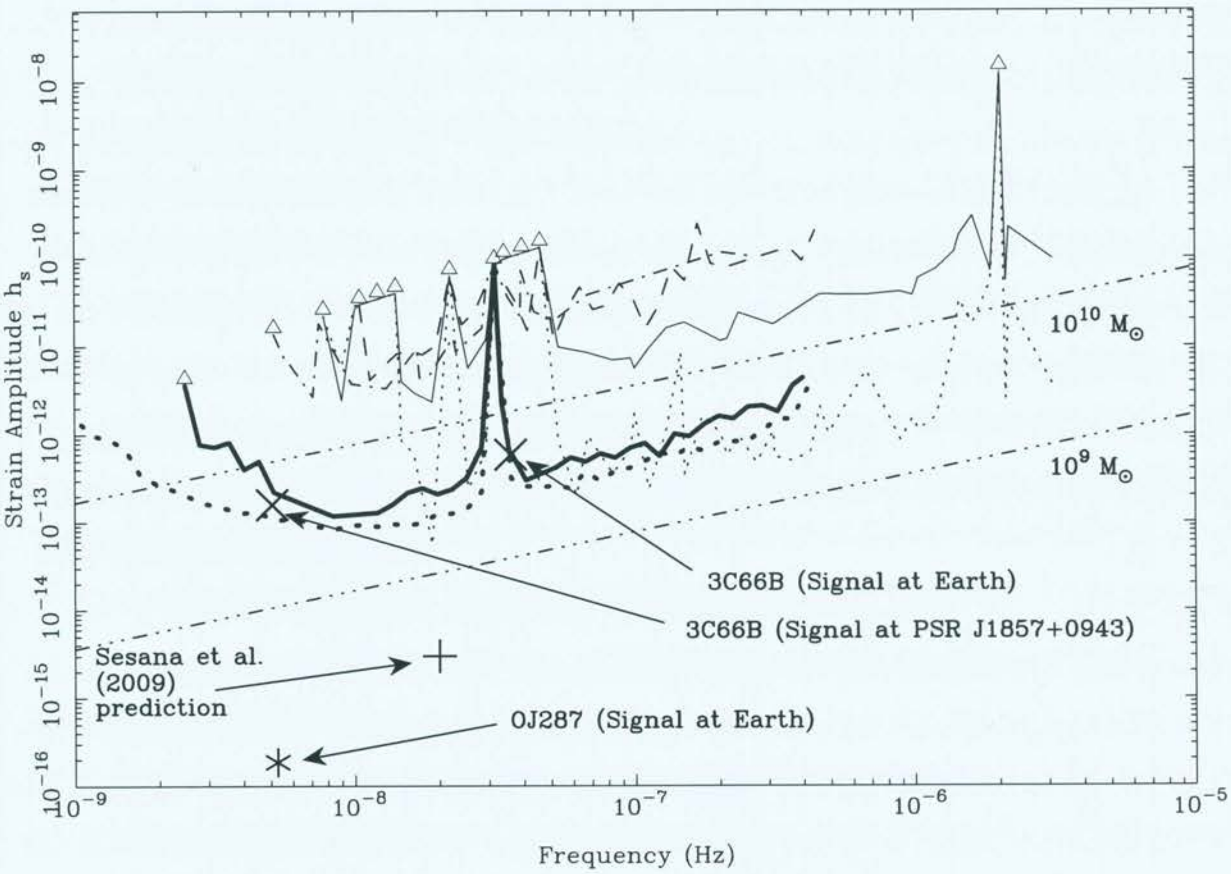

Figure 4.2: Sensitivity curves for PSRs J0437−4715 (thin solid line), J1713+0747 (dashed), J1857+0943 (dot-dashed) and the 18-pulsar timing array using our detection scheme (thick solid line). The abscissa gives the GW frequency, the ordinate gives the minimum detectable strain amplitude of an individual non-evolving GW point source with a random polarisation, phase and sky-position. The thin dotted line is the sensitivity obtained using PSR J0437−4715 and assuming favourable sky-location and polarisation of the GW source. An open triangle indicates that the plotted value is a lower bound on the detectable amplitude at that frequency. The straight triple-dot-dashed lines indicate the expected signal from an individual SMBHB with equal member masses of $10^9 M_\odot$ or $10^{10} M_\odot$ if it were located at the mean distance of the Virgo cluster. The '×' symbols are the expected signals at the Earth in the year 2004 and at PSR J1857+0943 ~ 3000 yr ago caused by the proposed SMBHB at the core of the radio galaxy 3C66B. The '$*$' symbol is the expected signal caused by the proposed SMBHB at the core of OJ287. The '+' symbol is the GW strain and frequency emitted by a typical resolvable SMBHB as plotted in figure 2 of Sesana et al. (2009). Also shown on the plot is the 95%-confidence limit curve for the 18-pulsar timing array (thick dotted line); in this case the ordinate gives the maximum amplitude GW source that could be present in our data.

frequency for this pulsar. We therefore do not see the corresponding loss in sensitivity at this frequency in the PSR J1857+0943 sensitivity curve. All pulsars exhibit a reduction in sensitivity at low frequencies, which is mainly caused by two effects. First, the fit of a quadratic polynomial to the ToAs to model the pulsar spin-down removes some GW signal. Second, the fitting of arbitrary phase offsets to many of the data sets to connect the timing residuals obtained with different backend systems removes some GW signal (see below). Greater sensitivity is obtained at the lowest frequencies if we allow for detection of a sinusoid at *any* frequency in the timing residuals, regardless of the input GW frequency[47]. This is because the pulsar spin-down and phase offset fits do not remove a pure sinusoid from the residuals. This means that not all of the input GW signal is removed by the pulsar parameter fit. However, in this implementation we have only allowed the GW signal to be detected at the input GW frequency.

As the GW frequency increases, the induced signal in the ToAs becomes weaker for a given strain, as described by Equation (3.6). At the highest frequencies, our sensitivity is limited by the sampling of the timing residuals. This is particularly evident in the sensitivity curve for the 18-pulsar timing array where there is a turn-up in the sensitivity curve at the last few frequency values, corresponding to a decrease in sensitivity.

The periodogram of irregularly-sampled residuals will be affected by leakage. There is no clear way to distinguish between spectral leakage from low-frequency GW-induced sinusoids and the red noise seen in many MSPs. Hence, the sensitivity of our detection technique to low-frequency sinusoidal GWs (where the GW period is similar to the data-span) is reduced compared to analysing equally-spaced data. Some pulsars in our sample do not exhibit excess low-frequency noise (e.g., PSR J1857+0943), so the power spectrum with no GWs added may be modelled with a constant. However, our model of the power spectrum must account for the confusion between the spectral leakage from a low-frequency GW signal and red noise. In an equally-spaced time series with weak red noise, spectral leakage is less severe and thus there is no such confusion.

In the sensitivity curve for PSR J0437−4715 there is a loss of sensitivity at a frequency of $(540\,\mathrm{days})^{-1}$, or $\sim$21 nHz. This is caused by the fitting of several arbitrary phase offsets between the ToAs collected using different observing backend systems, as described in Section 2.3.2. If overlapping data exist between the different observing backends, these offsets can be

[47] This approach was taken in Chapter 3.

precisely determined and held fixed in subsequent processing. Even if no overlapping data exist, it is sometimes possible to eliminate these arbitrary offsets without losing phase connection in the timing solution. Our analysis takes into account all of the offsets fitted by Verbiest et al. (2009). There is also a loss in sensitivity just above the $(1\mathrm{yr})^{-1}$ frequency for this pulsar. This is caused by the sampling of the observations – a sinusoid at this frequency induces power in many adjacent frequency channels, depending on the phase of the GW source. This increases the apparent noise level in this region of the periodogram, which increases the noise model and thus also the detection threshold sufficiently to prohibit 95%-confidence detection. In the best-case sensitivity curve for PSR J0437$-$4715 (thin dotted line in Figure 4.2), there is a decrease in sensitivity at a frequency of $\sim 150\,\mathrm{nHz}$. This decrease is caused by significant leakage of the input sinusoid into adjacent frequency channels. This sensitivity decrease is less significant in the sky-averaged case because the variation in the amplitude of the GW-induced sinusoid due to the sky- and polarisation-averaging is a much greater effect.

4.3.2 The Sensitivity of the Parkes Pulsar Timing Array and Probable Single Sources

The thick solid line in Figure 4.2 shows the sensitivity of the 18 pulsars in our data set assuming the GW source position and polarisation are unknown. This sensitivity curve is the first measurement of the sensitivity of a full PTA experiment to individual GW sources. The frequency range analysed $(30\,\mathrm{yr})^{-1}$ – $(28\,\mathrm{d})^{-1}$ is chosen to demonstrate the high- and low-frequency sensitivity limits for our pulsar timing data sets. At the lowest frequencies, our sensitivity is limited by the period derivative and jump fits, as well as the fact that our longest data set is shorter than 30 yr. At the highest frequencies, the sensitivity is limited by the sampling of our timing residuals; that is, $(28\,\mathrm{d})^{-1}$ is the nominal Nyquist frequency for the PPTA.

Figure 4.2 also shows the upper limit attainable using our 18-pulsar data set (thick dotted line). This limit curve was obtained with 95% confidence using the technique described in Sections 4.2 and Appendix A. For some pulsars, a different-order polynomial model to the detection case was chosen in order to accurately model the power spectrum with no GWs added. Lommen & Backer (2001) placed a 99% confidence limit showing that they could rule out signal amplitudes as small as 150 ns in their residuals at a period of 53 days, corresponding to SMBHB orbital periods of 106 days. Using our longer data sets and the same 99% confidence level, we can place a better limit of around 120 ns at this frequency. At signal periods of 1000 days (where some of our sets of timing residuals exhibit excess low-frequency noise), we obtain a 99% confidence limit of 190 ns. This limit is worse than the Lommen & Backer (2001) limit of 170 ns. However, there is no evidence that their analysis takes into account the effects of red noise present in their residuals.

The two '×' symbols in Figure 4.2 indicate the expected strain amplitude and frequency of the proposed SMBHB at the core of the radio galaxy 3C66B (Sudou et al., 2003). In order to determine the expected strain amplitude, we use Equation (3.4) with the redshift and masses given in the original paper ($M_1 = m_1 = 4.91 \times 10^{10}\,\mathrm{M}_\odot, M_2 = m_2 = 4.91 \times 10^{9}\,\mathrm{M}_\odot, z = 0.0215$). The distance to the GW source is assumed to be 90 Mpc, implied by the low-redshift distance approximation $D = cz/H_0$. The frequencies of the signal at the Earth and at PSR J1857+0943 ($f_{\rm Earth} = 1/0.88\,\mathrm{yr}$, $f_{\rm J1857+0943} = 1/6.24\,\mathrm{yr}$) were obtained by Jenet et al. (2004). The signal occurs at two frequencies because of the evolution of the SMBHB in the time interval between the interaction of the GWs with the Earth and the receipt of the GW-affected EM waves from the pulsar (see Equation 3.3). However, according to Equation (3.1), the timescale for evolution of the SMBHB is much longer than the span of the observations, so we assume that the frequency of each signal is constant over the observations. The GW signal at the pulsar will, in general, have a different frequency and amplitude for each pulsar in our array, whereas the Earth term will have the same frequency for observations of all pulsars. This system was ruled out with 95% confidence by Jenet et al. (2004). Our results show that, even with a blind search of the Verbiest et al. (2008, 2009) observations, where we know neither the sky-position nor the frequency of the GWs, we would detect the GW-induced oscillations at the Earth caused by this source. The expected signal is well below the plotted sensitivity curve for PSR J1857+0943 even though Jenet et al. (2004) only used the publicly-available ToAs for PSR J1857+0943. However, their technique is analogous to our limit technique, whereas the sensitivity curve plotted for PSR J1857+0943 in Figure 4.2 assumes we are aiming to *detect* such sources of GWs. Furthermore, our sensitivity curve is sky-averaged whereas they used the known position and frequency of the proposed GW source in their analysis (by chance it had a very favourable sky-location with an angle of 81.5° between the Earth-pulsar vector and the Earth-3C66B vector)[48]. Furthermore, if the frequency of the GW signal is known *a priori*, the false alarm probability

[48] Jenet et al. (2004) also underestimated the distance to the proposed GW source in 3C66B by around 8% by assuming that its redshift was $z = 0.02$.

is considerably decreased, meaning that the detection threshold is around a factor of two lower, increasing the probability of detection. Jenet et al. (2004) also assumed that they were analysing spectrally white timing residuals, an assumption which increases sensitivity, particularly at low frequencies.

The '$*$' symbol in Figure 4.2 indicates the expected GW strain and frequency for the candidate SMBHB in the blazar OJ287. A $\sim$12 yr-periodic signal has been identified in its optical outbursts (Sillanpää et al., 1996), but other parameters of the system are not well-constrained. We parametrise the SMBHB as follows: member masses $1.3 \times 10^8\,\mathrm{M}_\odot$ and $1.8 \times 10^{10}\,\mathrm{M}_\odot$, intrinsic orbital period 9 yr (observed GW period 6 yr because of redshifting), eccentricity zero[49], redshift 0.306, distance 1.3 Gpc. The distance was again obtained using $D = cz/H_0$, which is an acceptable approximation given the imprecision in the other parameter measurements and the fairly low redshift of this system (see footnote 1 of Davis & Lineweaver, 2004). The GW signals emitted by this system induce timing residuals of around 6 ns that are below current limits.

A study was presented by Sesana et al. (2008) of the generation of the stochastic GWB from the cosmic population of SMBHBs. This work showed that the stochastic background of GWs is likely to be detected using a PTA in the near future. In Sesana et al. (2009) the individual resolvable SMBHBs were considered. They predicted that at least one SMBHB will induce ToA perturbations around $5 - 50$ ns, which is below our current sensitivity. We choose (from the upper left panel of their figure 2) a representative resolvable single source from their simulations. This source has an emitted GW frequency of 2×10^{-8} Hz and a characteristic induced timing residual of 25 ns. The signal from this source is indicated by the '$+$' symbol in Figure 4.2. This is a typical resolvable SMBHB, thus it is likely that several sources will emit GWs with a larger amplitude than this. We emphasise that we do not yet have long data-spans with sufficiently low rms residual to detect such sources.

SMBHBs may form in galaxy clusters. The nearest galaxy cluster to Earth is the Virgo cluster. In Figure 4.3 we examine the possibilities for pulsar timing to detect GWs generated by SMBHBs in the Virgo cluster. The mean sky-position of this cluster is at a RA of $12^\mathrm{h}30^\mathrm{m}$ and a Dec of $+12^\circ$ (Mei et al., 2007); to produce the curve in Figure 4.3, all simulated GW signals come from this direction. The plotted sensitivity curve indicates that we have a better than 95%

[49] Valtonen et al. (2009) estimate the eccentricity to be 0.7, but we do not consider eccentric SMBHBs in this thesis.

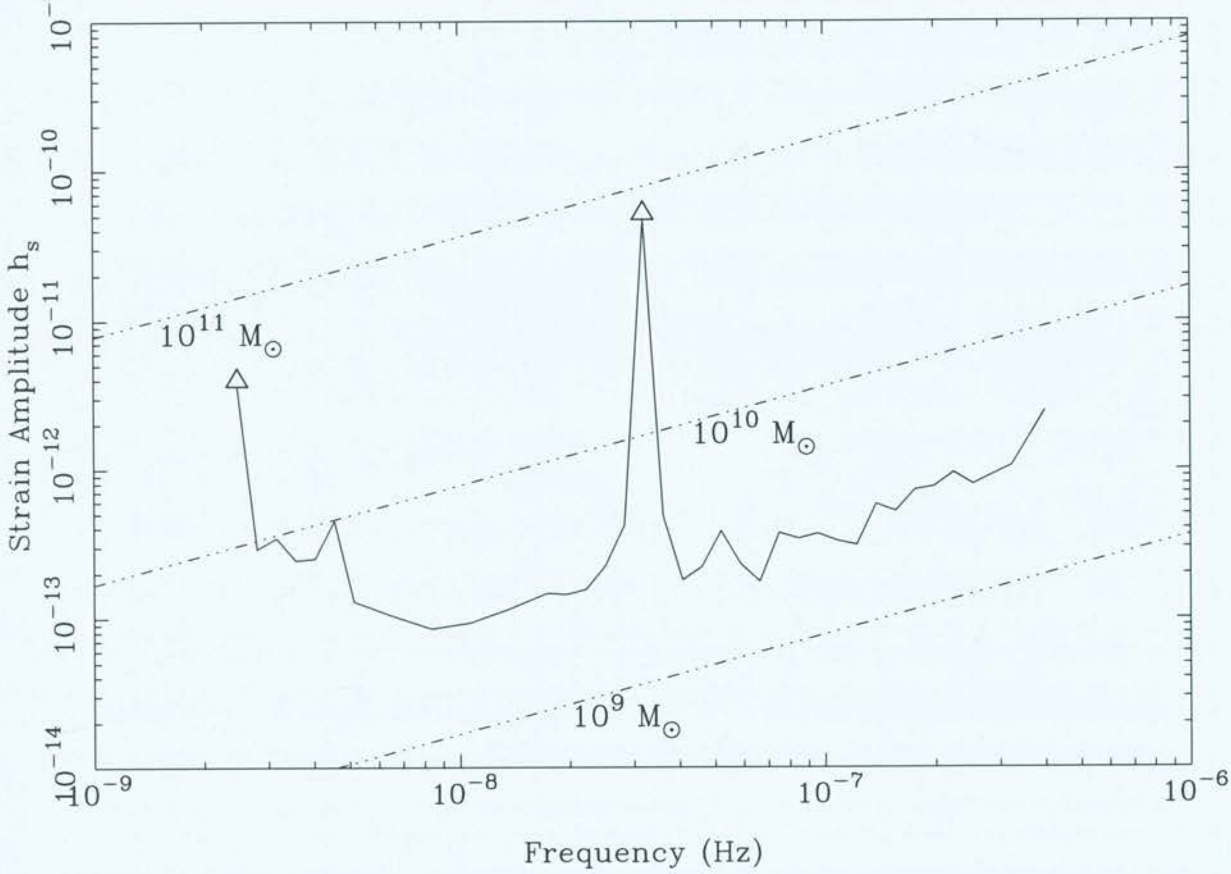

Figure 4.3: Sensitivity of the PPTA using the 18-pulsar Verbiest et al. (2008, 2009) data set for detecting signals from SMBHBs located at the sky-position and mean distance of the Virgo cluster. The abscissa gives the GW frequency. The ordinate gives the minimum detectable strain amplitude of GWs emanating from a non-evolving individual source in the direction of the mean sky-position of the Virgo cluster with a random polarisation and phase. The open triangles indicate that the plotted value is a lower bound on the detectable amplitude at those frequencies. The dot-dashed lines indicate the expected signals from three different types of SMBHB if they were located in the Virgo cluster, with equal member masses $10^9 M_\odot$, $10^{10} M_\odot$ and $10^{11} M_\odot$ as labelled.

probability of detecting sinusoidal signals in our timing residuals caused by SMBHBs with member masses $M_1 = M_2 = 10^{10}\,\mathrm{M}_\odot$ in the Virgo cluster. These SMBHBs could emit GWs with any polarisation, but our detectable frequency range for such sources is $3\times10^{-9}\,\mathrm{Hz} < f < 4\times10^{-7}\,\mathrm{Hz}$. We could marginally detect SMBHBs with $M_1 = M_2 = 10^{9}\,\mathrm{M}_\odot$ if the emitted GWs have favourable polarisation.

The PPTA sensitivity is complementary in GW frequency to the LIGO, VIRGO and LISA sensitivities. In Figure 4.4 we give the detection sensitivity of some current and future GW detection experiments[50]. Also shown on the plot are some likely sources in each of the detectable bands. The combination of the PTA and LISA sensitivity curves almost covers the full GW frequency range from $\sim$nHz through to $\sim$mHz. This GW frequency coverage will enable the study of the evolution of GW-emitting systems.

4.3.3 Constraining the Coalescence Rate of Supermassive Black-Hole Binaries

As described in Section 3.3.4, the non-detection of GWs from SMBHBs in pulsar timing observations enables an upper bound to be placed on the coalescence rate of SMBHBs (Wen et al., 2011). However, the upper bounds in Section 3.3.4 were calculated using a sensitivity matrix that gives the probability of detection of a GW source as a function of f and h_s for a given data set. Here, we calculate upper bounds on the SMBHB coalescence rate using a "limit matrix" that gives the probability that a GW source is ruled out by the observations as a function of f and h_s. For the same data set, the use of the limit matrix provides an upper bound on the SMBHB coalescence rate that is more constraining than the upper bound provided by the sensitivity matrix. We use the limit technique described in Sections 4.2 and Appendix A to calculate the limit matrix element at each f and h_s.

We calculate the limit matrix on a grid of 51 GW frequency values and 50 GW strain values. The 51 frequency values consist of 50 logarithmically-spaced frequencies between $(30\,\mathrm{yr})^{-1}$ and $(28\,\mathrm{d})^{-1}$, and also $f = 1/(1\,\mathrm{yr})$. The 50 strain values were logarithmically-spaced between 10^{-16} and 10^{-10}. 1000 Monte Carlo iterations were used at each value of f and h_s to determine the fraction of such GW sources that are ruled out by the data set. For the Verbiest et al. (2008,

[50] To obtain the LISA sensitivity curve, we have assumed the standard parameters for the LISA design and that it aims to detect sources at a signal-to-noise ratio of three. The LIGO sensitivity curves are obtained from the stated design goals of the project.

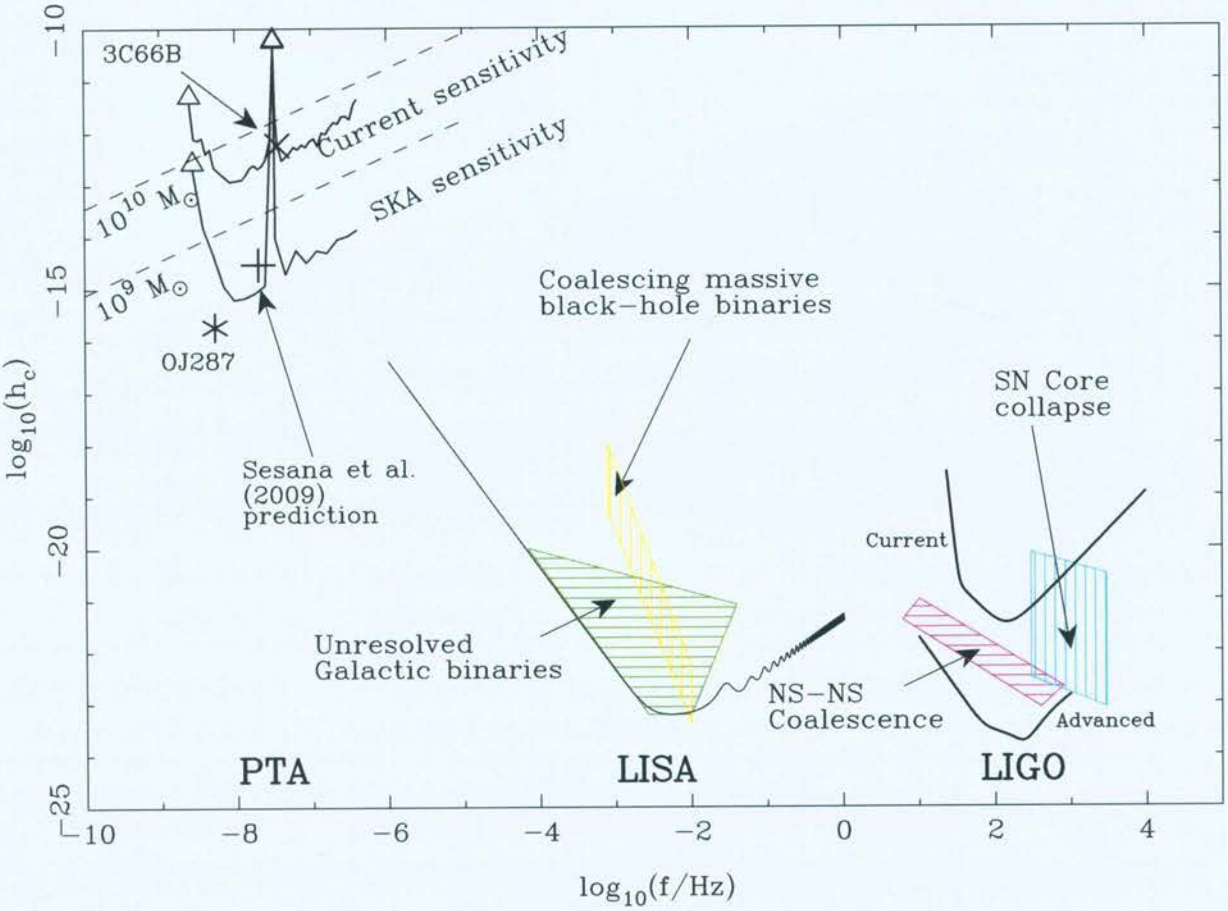

Figure 4.4: Sensitivity of some current and future GW observatories to individual GW sources as a function of frequency. The abscissa gives the GW frequency, the ordinate gives the minimum detectable strain amplitude of a sinusoidal GW point source with a random polarisation, phase and sky-position. The open triangles indicate that the plotted sensitivity at that frequency is a lower bound. The plot also shows potentially detectable sources in the three frequency bands. The straight lines indicate the expected signals from two different types of SMBHB if they were located in the Virgo cluster, with equal member masses 10^9M$_\odot$ and 10^{10}M$_\odot$ as labelled. The '×' symbol is the expected signal at the Earth caused by the proposed SMBHB at the core of the radio galaxy 3C66B. The '$*$' symbol is the expected signal caused by the candidate SMBHB at the core of OJ287. The '+' symbol is the GW strain and frequency emitted by a typical resolvable SMBHB as plotted in figure 2 of Sesana et al. (2009). "Unresolved Galactic binaries" include white-dwarf and neutron-star binaries. "Coalescing massive black-hole binaries" correspond to the final inspiral of black-hole binary systems. The "Current" LIGO sensitivity shows the capabilities of existing data sets, while "Advanced" LIGO expects to improve GW sensitivity by two orders of magnitude. "SN [supernova] core collapse" and "NS-NS [neutron star] coalescence" are typical signals that LIGO expects to detect.

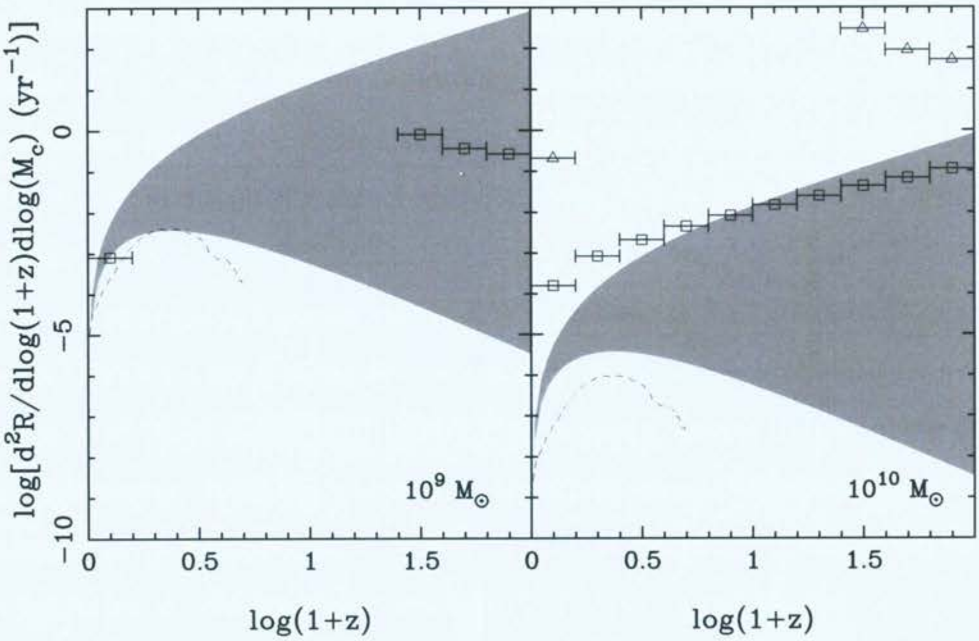

Figure 4.5: Upper bound on the coalescence rate of SMBHBs as a function of redshift. The open triangles give the upper limit on the SMBHB merger rate for the Verbiest et al. (2008, 2009) data set and the open squares give the limit for the simulated SKA data sets. The shaded region indicates the expected coalescence rate obtained from Jaffe & Backer (2003) as well as data from the Sloan Digital Sky Survey (Wen et al., 2009) for SMBHB systems of chirp mass as labelled in each panel. The dashed line indicates the average coalescence rate based on the analysis by Sesana et al. (2008).

2009) observations, the 95% confidence contour of the limit matrix is consistent with the thick dotted line in Figure 4.2.

The resulting constraint on the SMBHB coalescence rate is shown in Figure 4.5 as a function of $\log(1+z)$. The plot shows that the coalescence rate of SMBHBs with $z \lesssim 2$ with chirp mass $\sim 10^{10}\,\mathrm{M}_\odot$ is less than one merger every five yr. Our observations do not yet constrain the merging frameworks discussed by Jaffe & Backer (2003) or Sesana et al. (2008) at the range of chirp masses we have considered. However, some of the high-mass and high-redshift predictions may soon be ruled out or confirmed using pulsar timing. Furthermore, the limit on the amplitude of the isotropic stochastic GWB obtainable from the Verbiest et al. (2008, 2009) observations may provide a more constraining upper bound on the SMBHB coalescence rate (see Chapter 5).

4.3.4 A Predicted Sensitivity Curve for the Square Kilometre Array

Figure 4.4 also gives a predicted sensitivity curve for the SKA. To produce this figure we used simulated observations for the 100 pulsars described in Section 2.6. We have assumed we can time each pulsar with an accuracy of 20 ns over five yr, obtaining one observation per pulsar every 14 d. We have also assumed that their power spectra will be statistically white. It is unlikely that pulsar timing residuals will exhibit a white power spectrum at this timing precision and, hence, the plotted sensitivity is a lower bound on what is achievable with the SKA. In particular, the sensitivity at low frequencies is expected to be worse than that shown here, because we expect higher noise levels caused by the GWB, intrinsic pulsar timing noise and other unmodelled effects.

The simulated SKA data are equally-spaced, which causes the level of spectral leakage to be much lower than that observed in irregularly-sampled data sets. Hence, the confusion between red noise and low-frequency signal is no longer an issue in these simulations because a sinusoidal GW signal will induce a very narrow peak in each pulsar's periodogram, even at low frequencies. We have therefore modelled each pulsar power spectrum with a constant.

There are three prominent losses in sensitivity - at frequencies smaller than $(T_{\rm obs})^{-1}$ and at periods of one year and six months. The partial loss in sensitivity at a period of six months ($\sim 6 \times 10^{-8}$ Hz) is caused by fitting for the pulsar parallax. The total loss in sensitivity at GW periods of one yr could be mitigated using independent measurements of the position of the pulsar, for example using very-high-precision interferometry; such precision may be available in the SKA era (Smits et al., 2011). The SKA sensitivity curve differs from that shown in Figure 3.5 because the noise level in the residuals is different, and we have only allowed detection of the GW-induced sinusoid at the input GW frequency.

The SKA sensitivity curve shown in Figure 4.4 is calculated assuming we do not know the location or frequency of a potential GW source. Using these two additional pieces of information it may be possible to confirm or deny the binarity of the massive dark object at the core of OJ287. It may also be possible to detect many of the resolvable SMBHBs predicted by Sesana et al. (2009). Using the SKA and LISA, it may be possible to observe the full evolution of SMBHBs from emitting GWs in the pulsar timing band (during the early phases of coalescence) to emitting GWs in the LISA band (during coalescence) (e.g., Pitkin et al., 2008).

4.4 Conclusion

We have presented the strain sensitivity of the PPTA to non-evolving point sources of GWs as a function of frequency. The sources most likely to produce a detectable sinusoid in the pulsar timing frequency range are SMBHB systems in the early phases of coalescence at the cores of merged galaxies. The sensitivity curve is analogous to the LIGO, VIRGO and LISA sensitivity curves and shows the unique GW frequency range accessible with pulsar timing. These results can be used to place an upper bound on the number of coalescing binary systems of a given chirp mass as a function of redshift. Current observations do not yet rule out any recently proposed models of galaxy evolution.

However, the isotropic stochastic GWB is expected to provide a larger amplitude signal in the ToAs than most individual sources (Sesana et al., 2008). If the amplitude of the GWB signal remains large after the pulsar parameter fit, then it may be detectable in the timing residuals from a PTA. Furthermore, non-detection of the expected GWB signal provides a constraint on the coalescence rate of SMBHBs (Wen et al., 2011). Therefore, in the next two Chapters, we examine the GWB signal using simulated and real observations of a PTA.

Chapter 5

Limiting the Amplitude of the Gravitational-Wave Background

Chapter Outline: *In this Chapter, we:*

- *describe the method for simulating a GWB as implemented in* TEMPO2.
- *briefly describe the technique for placing an upper limit on the GWB amplitude developed by Jenet et al. (2006).*
- *apply this technique to the same white data sets used in Chapter 3.*
- *calculate the constraints on the SMBHB coalescence rate using the technique published by Wen et al. (2011).*
- *briefly describe which models of galaxy evolution are ruled out by the limits obtained with different simulated data sets.*

As described in the text, a more detailed version of Section 5.1 was published as
Hobbs G., Jenet F., Lee K. J., et al. 2009, MNRAS, 394, 1945
In Section 5.2, I created the simulated data sets and measured the limit on the amplitude of the GWB for each data set. The constraints on the coalescence rate of SMBHBs (Section 5.2.1) were calculated by Zhonglue Wen and published as
Wen Z. L., Jenet F. A., Yardley D., Hobbs G. B., Manchester R. N., 2011, ApJ, 730, 29

In Chapter 4, we showed that it is likely that the GWs emitted by an individual non-evolving SMBHB will induce ToA perturbations that are below current sensitivity levels. A stronger signal may be induced in pulsar ToAs by an isotropic stochastic background of GWs. Such a background is formed from the incoherent sum of many individual SMBHBs with different frequencies, amplitudes and phases (see Section 1.5).

In this Chapter, we calculate upper limits on the GWB amplitude using the technique presented by Jenet et al. (2006). This technique requires simulation of the effect of a GWB on ToAs. We apply this technique to real and simulated PTA observations that are consistent with white noise[51]. The results indicate that it is likely that a variety of models of galaxy evolution may be ruled out using PTA observations in the near future.

A new technique for detection of the GWB due to SMBHBs will be described in Chapter 6. While the direct detection of GWs will have significant consequences for astrophysics and cosmology, the sensitivity of current PTA data sets is insufficient for making such a detection[52]. However, upper limits on the GWB amplitude can be obtained with any set of PTA observations. These upper limits constrain the coalescence rate of SMBHBs as a function of redshift (Wen et al., 2011) and models of galaxy evolution (Jaffe & Backer, 2003; Wyithe & Loeb, 2003; Sesana et al., 2008).

5.1 Method

5.1.1 The Expected Signal Induced by a Gravitational-Wave Background in Timing Residuals

The signal induced by a GWB in pulsar ToAs is described in Section 1.6.1. For this work, we assume that a GWB due to SMBHBs has characteristic strain spectrum of the form

$$h_c(f) = A(f/f_{1\,\mathrm{yr}})^{-2/3} \,. \tag{5.1}$$

This is consistent with most models in Section 1.5. The Sesana et al. (2008) model predicts a more complicated form for the strain spectrum of the GWB (given in Equation 1.13 of this thesis). This model shows significant deviation from a simple power law for $f > 10^{-8}$ Hz. However, our analysis probes GWB frequencies $f \lesssim 10^{-8}$ Hz (see Figure 1.6), where the difference between Equation (1.13) and Equation (5.1) is insignificant.

A GWB of the form given in Equation (5.1) will induce perturbations in the ToAs of each pulsar with the following power spectrum (Hobbs et al., 2009):

$$P_g(f) = \frac{A^2}{12\pi^2}\left(\frac{f}{f_{1\,\mathrm{yr}}}\right)^{-13/3} \,. \tag{5.2}$$

In order to use the technique of Jenet et al. (2006) described in Section 5.1.3, we must be able to simulate a GWB that reproduces this expected signal. This is done using TEMPO2. We briefly

[51] These are the same observations that were analysed in Chapter 3 to detect individual non-evolving SMBHBs.
[52] As described in Section 1.6, a detection of a GWB signal is expected within the next decade.

describe the GWB simulations in the next section; full details were published by Hobbs et al. (2009).

5.1.2 Simulating a Background of Gravitational Waves With TEMPO2

TEMPO2 simulates a GWB using many individual monochromatic GWs. For each GW source, the phase, Φ_g, the right ascension, ϕ_g, and the sine of the declination are each chosen from uniform probability distributions:

$$\begin{aligned} \mathrm{Prob}(\Phi_g) &= 1/2\pi\ , && (5.3) \\ \mathrm{Prob}(\phi_g) &= 1/2\pi\ , && (5.4) \\ \mathrm{Prob}(\sin\theta_g) &= 1/2 && (5.5) \end{aligned}$$

respectively, where θ_g is the declination of the GW source, as in Section 2.5. The GW frequency f is chosen from a uniform distribution in $\log(f)$[53]:

$$\mathrm{Prob}(f) = \begin{cases} \frac{1}{f\log(\frac{f_h}{f_l})} & f_l \leq f \leq f_h \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

where f_l and f_h are the lowest and highest frequencies for the simulated GWs, respectively. In this thesis, we used $f_l = 0.05/T_{\rm obs}$ and $f_h = 1\mathrm{d}^{-1}$.

The imaginary parts of A_+ and $A_\times$ are set to zero for every GW because the GWB will be unpolarised. The real parts are each normally distributed with zero mean and standard deviation given by:

$$\sigma_A(f) = \sqrt{\frac{\log(f_h/f_l)}{N}}h_c(f)\ , \tag{5.7}$$

where $h_c(f)$ is given in Equation (5.1) and N is the number of GW sources simulated. In general, $N \sim 1000$.

The resulting set of N GWs forms an isotropic, unpolarised GWB with a gaussian amplitude distribution with mean characteristic strain $h_c(f)$, as given in Equation (5.1). It is essential for the work both in this Chapter, and Chapter 6, that we can simulate a realistic GWB signal. In order to provide confidence in the accuracy of the GWB simulations, we reproduce work that

[53] This choice is motivated by the large spread of simulated GW frequencies over many orders of magnitude. We suspect that the exact form of the distribution has little effect on our sensitivity to the GWB signal.

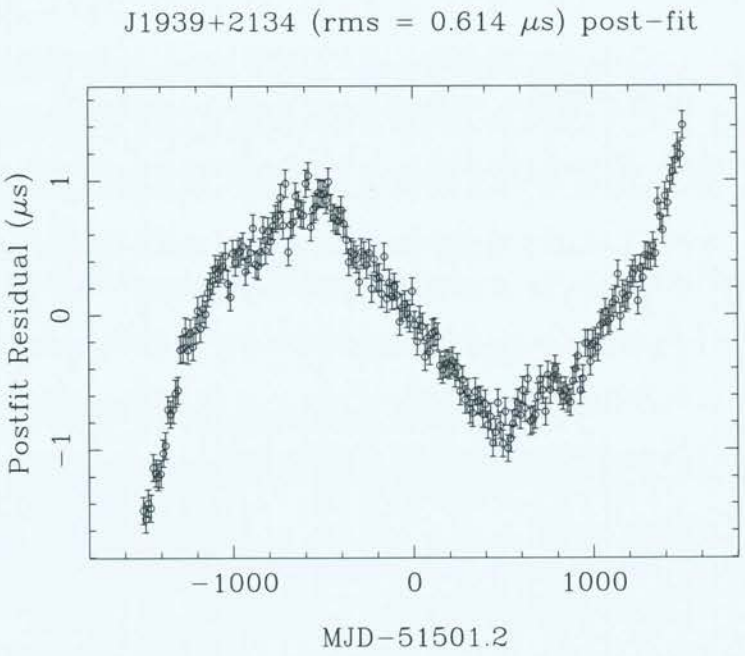

Figure 5.1: Timing residuals for PSR J1939+2134 obtained from simulated ToAs that are affected by 100 ns of white noise and a GWB signal with $A = 10^{-14}$. The abscissa gives the MJD, the ordinate gives the residual after estimating the pulsar's period and period derivative. The uncertainty in each simulated ToA (error bars) is 100 ns. [Image reproduced from Hobbs et al. (2009)]

was first shown by Hobbs et al. (2009).

First, Hobbs et al. (2009) simulated ToAs for PSR J1939+2134 that have been sampled once every two weeks over 3000 d. The ToAs consist of ideal ToAs (that is, the ToAs predicted by the timing model; see Section 2.4.1) that are then perturbed by 100 ns of white gaussian noise and a GWB signal. After carrying out a TEMPO2 fit to estimate the pulsar's period and period derivative, the resulting post-fit timing residuals are shown in Figure 5.1.

Second, Hobbs et al. (2009) calculated the average power spectrum of 1000 realisations of a set of 512 GWB-induced ToA perturbations. The 512 simulated ToAs occur at weekly intervals and consist of ideal ToAs that are then perturbed by a simulated GWB with $A = 10^{-15}$. The predictions of the timing model are then subtracted from the simulated ToAs, without carrying out a TEMPO2 fit. In this case, the pre-fit residuals are identical to the post-fit residuals. The average power spectrum of the residuals is plotted in Figure 5.2. Also shown is the theoretical level of the GWB power spectrum assuming $A = 10^{-15}$ in Equation (5.2). At high frequencies in the plot, the average spectrum of the simulated residuals is not consistent with the theoretical spectrum. This is because of rounding errors in the TEMPO2 processing that, in this case, induce noise with a standard deviation of 0.2 ns. Given that TEMPO2 was designed to maintain

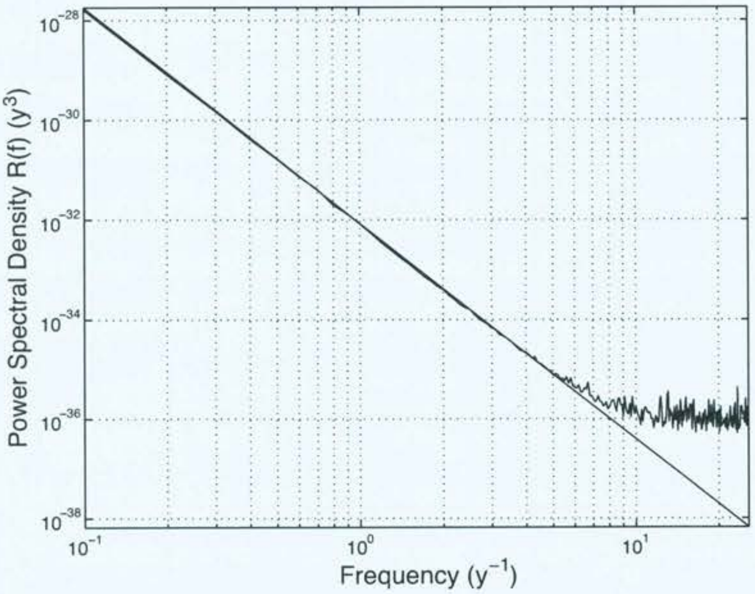

Figure 5.2: The power spectrum of GWB-induced ToA perturbations. The abscissa gives the frequency, the ordinate gives the power spectral density. The spectrum of 512 weekly-spaced ToA perturbations averaged over 1000 GWB simulations (solid trace) reproduces the theoretical spectrum (solid diagonal line) at all but the highest frequencies. [Image reproduced from Hobbs et al. (2009)]

1 ns precision, and the smallest observed residuals are currently greater than 20 ns, this noise can be neglected. Thus we conclude that the GWB simulation engine in TEMPO2 induces ToA perturbations caused by a GWB that reproduce the theoretical power spectrum.

5.1.3 Calculating an Upper Bound on the Amplitude of the Gravitational-Wave Background

We now use the TEMPO2 GWB simulation engine to calculate an upper bound on the GWB amplitude using timing residuals that are consistent with white noise. We use the technique published by Jenet et al. (2006). These authors aim to calculate an upper bound on the GWB by finding an upper bound on the level of red noise present in the timing residuals. Red noise is one of the characteristic signatures of GWB-induced residuals (see Figure 5.1). Jenet et al. define a statistic, Υ, that can detect red noise in timing residuals. We briefly describe the calculation of Υ here; full details are given by Jenet et al. (2006) and Hobbs et al. (2009).

Υ is the sum of the power in the first seven channels of the weighted average "polynomial spectrum" for the data set. For each pulsar, the l-th channel of the polynomial spectrum is defined by the result of a fit of an orthonormal polynomial of order l to the residuals. The average is calculated over all pulsars in the data set and is weighted by the square of the weighted rms residual. For residuals that are dominated by a GWB signal of the form of Equation (5.2), the sum of the first seven channels of the polynomial spectrum contains 95% of the power in the GWB signal (Jenet et al., 2006). In practice, the remaining 5% of the GWB power in higher channels will be negligible compared to other noise sources in the residuals. If these higher channels were to be included in the sum, then the detection statistic would be less sensitive to a GWB.

The limit on the GWB amplitude for a given white data set is obtained using two Monte Carlo simulations. The process is very similar to that used in Chapter 3 for individual GW sources. The first simulation calculates a particular value of the statistic, Υ_0, such that any measured Υ that exceeds Υ_0 indicates that significant red noise is present in the data set. We use the following procedure:

- We determine a set of ideal ToAs predicted by the timing model of each pulsar in the data set.
- We randomly rearrange the input residuals and then add them to the ideal ToAs.
- We perform the TEMPO2 parameter fit to obtain a new set of post-fit timing residuals that are statistically equivalent to the input residuals.
- We calculate Υ for this new set of residuals.
- We repeat the previous five steps 10^4 times and find the 10th highest value of Υ, which we set as Υ_0. The probability that a statistically-equivalent data set would yield a value of Υ larger than Υ_0 when no red noise is present is 0.1%.

The second Monte Carlo procedure obtains the limit on the GWB amplitude as follows:

- We add GWB-induced perturbations and a shuffled set of the original residuals to the ideal ToAs. The GWB has amplitude $A = A_{\rm test}$ in Equation (5.2).
- We perform the TEMPO2 parameter fit, which absorbs some of the GWB signal in the post-fit residuals.
- We calculate Υ for this new set of residuals.

Table 5.1: The upper bound on A obtained from each set of observations.

Data set (N_{psr}-rms residual-T_{obs})	A_{up}
20 PSRs-500 ns-10 yr	1.1×10^{-15}
20 PSRs-100 ns-5 yr	9.9×10^{-16}
20 PSRs-100 ns-10 yr	2.2×10^{-16}
20 PSRs-10 ns-10 yr	2.0×10^{-17}
100 PSRs-100 ns-5 yr	5.7×10^{-16}
100 PSRs-100 ns-10 yr	1.3×10^{-16}
100 PSRs-10 ns-10 yr	8.8×10^{-18}
Jenet et al. (2006) observations	1.1×10^{-14}

- We repeat the previous three steps 10^4 times and measure the percentage of Υ values that exceed Υ_0.

- If the percentage is more than 95%, then we repeat the previous four steps with a smaller value of A_{test}. If less than 95%, repeat the previous four steps with a larger A_{test}.

- When a value of A_{test} is found such that 95% of the measured values of Υ exceed Υ_0, then $A_{\text{up}} = A_{\text{test}}$ is the 95%-confidence upper limit on the GWB amplitude.

This procedure calculates statistically-rigorous upper bounds on the amplitude of the GWB (Jenet et al., 2006). However, if it is applied to data sets that contain some red noise, then the resulting GWB limit is lower than the value that the data set actually implies (Hobbs et al., 2009). We now apply this method to each of the white-noise data sets analysed in Chapter 3 to determine the upper bound on A from each set of observations.

5.2 Results and Discussion

In the first column of Table 5.1, we describe the data sets used. The second column gives the corresponding upper bound, A_{up}. The upper bound we obtain for the Jenet et al. (2006) observations is equal to the upper bound published by Jenet et al. (2006). Jenet et al. also analysed a simulated white-noise data set consisting of five years of observations of 20 pulsars with a rms residual of 100 ns, and obtained $A_{\text{up}} = 6.5 \times 10^{-16}$. This is significantly less than the upper bound of $A_{\text{up}} = 9.9 \times 10^{-16}$ shown in Table 5.1 for a fortnightly-sampled data set. It turns out that the simulated observations presented by Jenet et al. (2006) were sampled weekly

(Hobbs & Jenet, private communication). We obtained $A_{\rm up} = 6.6 \times 10^{-16}$ for a simulated data set consisting of weekly-sampled observations of 20 pulsars with a rms residual of 100 ns over five years. Hence, we conclude that our results are consistent with the work of Jenet et al. (2006).

5.2.1 Constraining the Coalescence Rate of Supermassive Black-Hole Binaries

Wen et al. (2011) derive a constraint on the coalescence rate of SMBHBs for a given upper bound on the GWB amplitude. This constraint is calculated by assuming that there are many SMBHBs emitting GWs in each frequency channel of the residuals. However, if the residuals have been timed accurately or over a long period of time, then the constraint on the coalescence rate is sufficiently low that this assumption is violated at some redshifts. This means that the constraint for a given chirp mass is only valid over certain ranges of redshift for which the induced timing residual is low for each SMBHB (Wen et al., 2011).

In Figures 5.3 and 5.4, we plot the upper bound on the SMBHB coalescence rate as a function of redshift for our data sets. Valid constraints can be obtained at all redshifts for these data sets using the methods presented in Chapter 3 or Chapter 4, because these methods allow for an arbitrarily small number of GW sources in each frequency bin. However, these methods also give a higher upper bound on the coalescence rate compared to the method presented in this Chapter, so the method of this Chapter should be used whenever it provides a valid constraint.

In Figure 5.4, we show that no valid constraints can be obtained with this method for $M_c = 10^9 {\rm M}_\odot$ with either of the simulated data sets that have a rms timing residual of 10 ns. However, these data sets provide significant constraints on the merger rate of SMBHBs with smaller chirp masses. We do not present such constraints here because the precision of current data sets is not at the required 10 ns level for more than a few pulsars.

5.2.2 Predictions of Galaxy Evolution Models

Wyithe & Loeb (2003) predict that the amplitude of the GWB could be as high as $A = 10^{-14}$. This prediction has recently been ruled out by van Haasteren et al. (2011) using a Bayesian analysis technique. Our results indicate that this prediction could also be ruled out by applying the Jenet et al. (2006) technique to the "20PSRs-500 ns-10 yr" data set.

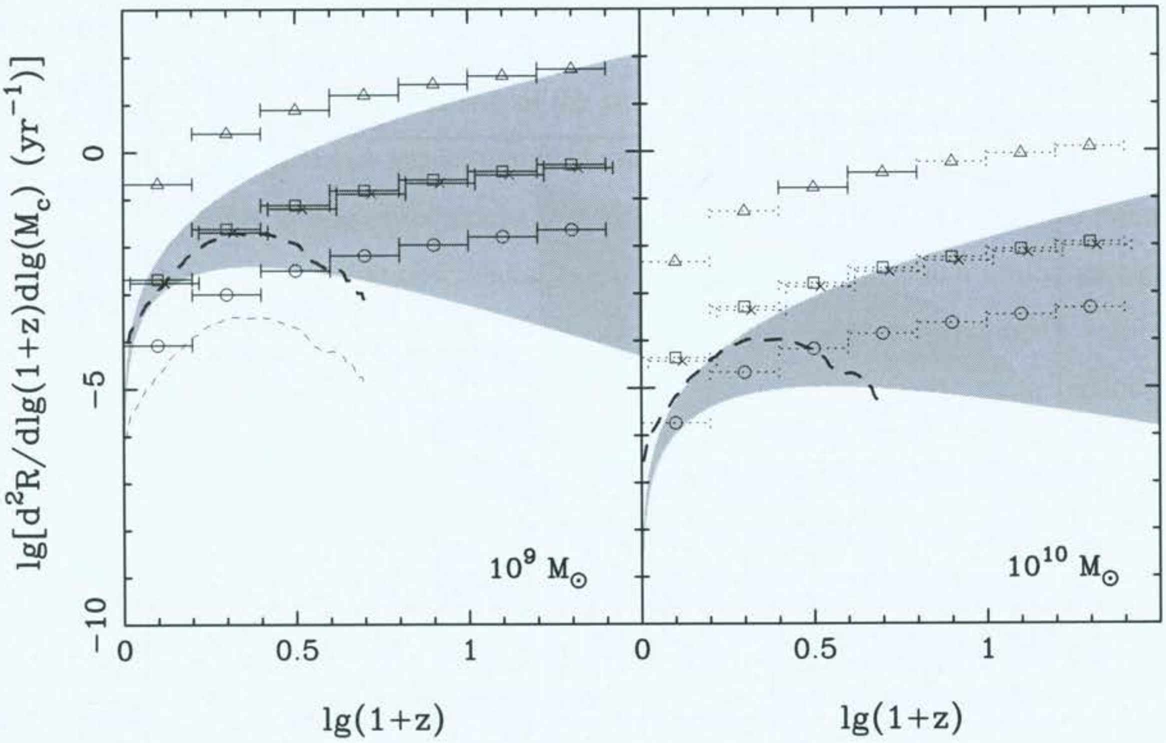

Figure 5.3: Upper bounds on the coalescence rate of SMBHBs using the values of A_{up} calculated for different sets of PTA observations given in Table 5.1. For calculating the abscissa, z is the redshift of the SMBHB. The ordinate gives the logarithm of the differential rate of coalescence per log redshift per log chirp mass. The solid horizontal bars indicate that the constraint is valid in that redshift interval, while the dotted horizontal bars indicate that the constraint is invalid. The plot includes the constraints provided by the Jenet et al. (2006) observations (open triangles), 20 pulsars timed with 500 ns rms residual over 10 yr (open squares), the same timed with 100 ns rms residual over five yr (crosses) and the same timed with 100 ns rms residual over 10 yr (open circles). The grey region indicates the expected coalescence rate with evolution index $-1 < \gamma < 3$ (see Section 1.6.2) assuming the framework of Jaffe & Backer (2003) and using observations from the Sloan Digital Sky Survey (Wen et al., 2009). The dashed traces indicate the maximum (thick line) and minimum (thin line) coalescence rates predicted by Sesana et al. (2008, 2009). [Image reproduced from Figure 2 of Wen et al. (2011).]

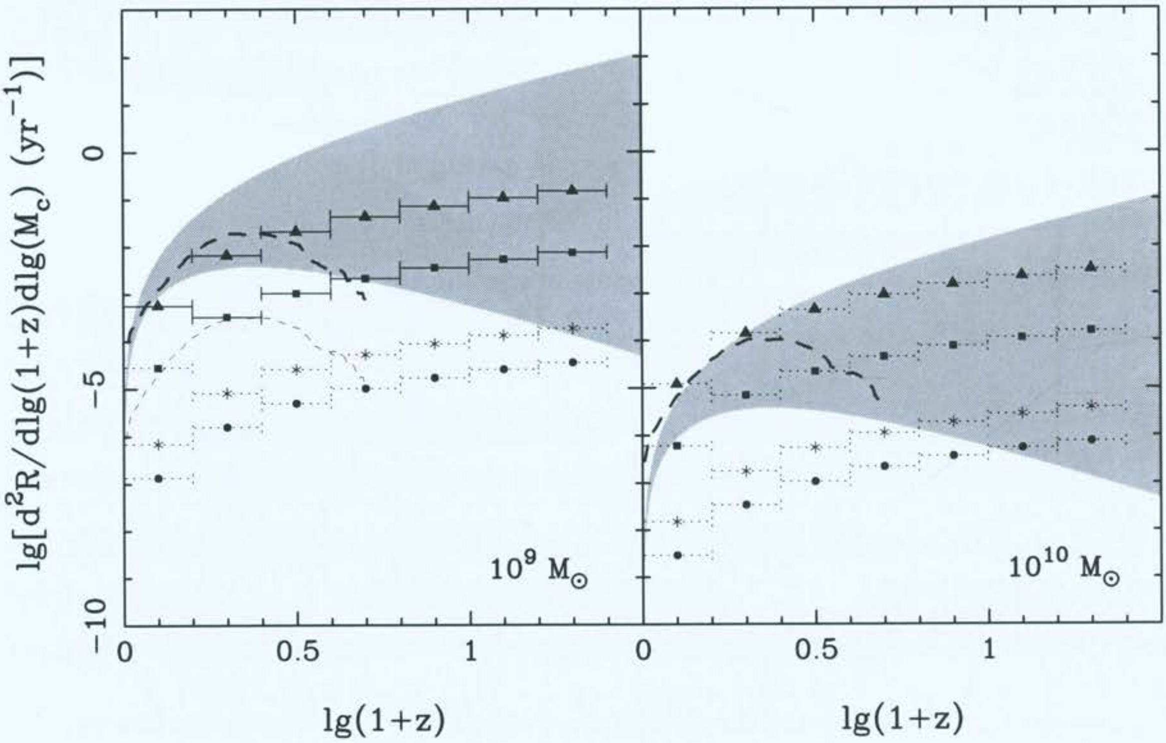

Figure 5.4: Lines and regions on this plot are the same as in Figure 3.8, except that we now show the constraints obtained using 20 pulsars timed with a rms timing residual of 10 ns over 10 yr (stars), 100 pulsars timed with a rms residual of 10 ns over 10 yr (filled circles), the same timed with a rms residual of 100 ns over 10 yr (filled squares) and the same timed with a rms residual of 100 ns over five yr (filled triangles). [Image reproduced from Figure 3 of Wen et al. (2011).]

Jaffe & Backer (2003) and Enoki et al. (2004) predict that the GWB amplitude is around $A = 10^{-15}$. If the timing residuals are white, then this prediction could be ruled out using the most optimistic prediction for the full PPTA observations, which is the "20PSRs-100 ns-5 yr" data set. An improved technique for calculating an upper bound on the GWB amplitude could rule out this prediction with the "20PSRs-500 ns-10 yr" data set.

Sesana et al. (2008) predict that the characteristic strain of the GWB is in the range $10^{-16} < h_c < 3 \times 10^{-15}$ at $f = f_{\rm 1yr}$. While any of the simulated data sets shown in Table 5.1 could constrain part of this parameter space, the whole range of $h_c(f = f_{\rm 1yr})$ can only be ruled out by the simulated SKA timing observations that yield a rms residual of 10 ns on ≥ 20 pulsars. However, a timing program that observed a PTA with at least 20 pulsars at a timing precision of $\sim$100 ns for more than 10 years could significantly constrain the predicted range.

If a significant upper bound on the GWB amplitude is obtained, several characteristics of SMBHB formation and evolution may be constrained (see Sesana et al., 2008, and references therein). Aside from the constraints on the coalescence rate described in Section 5.2.1, a smaller GWB amplitude could mean that the proportion of SMBHBs that proceed to coalescence may be less than 100%. The BH mass function could also be over-estimated currently, which would lead to inflated predictions of the GWB amplitude. Alternatively, the gravitational recoil experienced by each SMBH during merger could be larger than currently predicted, meaning that fewer SMBHBs would form. These parameters are difficult to measure using direct observations of SMBHBs.

5.3 Conclusion

We have applied the technique of Jenet et al. (2006) to both real and simulated observations to determine upper bounds on the GWB amplitude. We have also briefly described the astrophysical consequences of such limits. However, the technique can only be applied to sets of timing residuals that are consistent with white noise, which is only the case for a relatively small number of data sets. Most current sets of pulsar timing residuals show evidence of red noise. Also, the Jenet et al. (2006) technique cannot easily be extended to provide a direct detection of a GW signal. In Chapter 6, we develop a technique that can detect a GWB signal and can be applied to almost any set of pulsar timing residuals.

Chapter 6

On Detecting the Gravitational-Wave Background Using a Pulsar Timing Array

Chapter Outline: *In this Chapter, we:*

- *correct minor errors in the Verbiest et al. (2008, 2009) data set and also adjust the ToA uncertainties.*
- *develop a frequency-domain correlation technique to search for a GWB signature in the PPTA residuals.*
- *apply this technique to the corrected Verbiest et al. data set and find no detectable GWB signal.*
- *discuss characteristics of the PPTA residuals and the GWB signal that affect the GWB analysis.*
- *examine the effect of instabilities in a realisation of Terrestrial Time and errors in the solar system ephemeris on the GWB detectability.*

Sections 6.1, 6.2, 6.3, 6.4 and 6.5 are adapted from sections in the following journal article:
Yardley D. R. B., Coles W. A., Hobbs G. B., et al. 2011, MNRAS, 414, 1777

Figure 6.5 and its associated text are from the following conference proceedings:
Yardley D. R. B., Coles W. A., Hobbs G. B., Manchester R. N., 2011, in Burgay M., D'Amico N., Esposito P., Pellizzoni A., Possenti A., eds, *Radio Pulsars: An Astrophysical Key to Unlock the Secrets of the Universe*, Vol. 1357 of AIP Conference Series. American Institute of Physics, Melville, New York, p. 77

In this Chapter, we attempt to detect a GWB signal caused by SMBHBs in an updated version of the PPTA observations presented by Verbiest et al. (2008, 2009) (introduced in Section 2.3.2). While an upper bound on the GWB amplitude can rule out models of galaxy evolution (see Section 5.2.2) and cosmic strings (see Section 1.6.2), a detection of the GWB would lead to increased understanding of physics and cosmology. In order to confirm that any signal observed in a data set is caused by a GWB, it is essential to detect (unambiguously) the expected correlation in the timing residuals of pairs of pulsars, as shown in Figure 1.8

Several techniques have already been proposed in the literature to detect the GWB (see Section 1.6.1). However, most methods have not taken into account all the details of analysing pulsar timing data, or are restricted to particular observations.

The GWB detection technique we present in this Chapter is based on the method of Jenet et al. (2005). It improves on their technique in a number of ways:

- we study the pairwise correlation described by Hellings & Downs (1983) in the form of pairwise cross-power spectra;
- we obtain independent estimates of the GWB from each frequency component in each cross-power spectrum;
- we use an optimally-weighted linear combination of the cross-power estimates as the detection statistic;
- we account for the effect of different overlapping time-spans between the pulsar pairs;
- we calibrate the cross-power spectra and their estimated errors using simulations that completely account for the fitting of the pulsar timing model.

In this Chapter, we discuss a number of issues that are common to both the Jenet et al. (2006) limit technique and any limit technique based on measuring the GWB-induced correlation between pulsars. Such issues include the estimation of power spectra when the sampling is irregular and the ToA uncertainties are variable, and the effects of fitting the timing model.

In Section 6.1 we describe the observations and the analysis that led to the timing residuals we use in this Chapter. Section 6.2 describes the theoretical background and our method for making a detection of the isotropic stochastic GWB. Section 6.3 describes the results obtained, Section 6.4 describes their implications and the outstanding issues for GWB detection via pulsar timing, and Section 6.5 summarises our conclusions.

6.1 Observations

High-precision timing observations of 20 MSPs over ~10 yr were presented by Verbiest et al. (2008, 2009). The timing residuals for all pulsars obtained from these observations are shown in Figure 2.5 and Figure 2.6 and described in Section 2.3.2. However, to form the data set we use in

Table 6.1: Basic information for our corrected version of the Verbiest et al. data set.

PSRJ	Period (ms)	DM ($\text{cm}^{-3}\,\text{pc}$)	P_b (d)	Weighted RMS Residual (μs)	Span (years)	No. of Observations
J0437−4715	5.757	2.65	5.74	0.20	9.9	2847
J0613−0200	3.062	38.8	1.20	1.52*	8.2	190
J0711−6830	5.491	18.4	–	3.24*	14.2	227
J1022+1001	16.45	10.3	7.81	1.63*	5.1	260
J1024−0719	5.162	6.49	–	4.17*	12.1	269
J1045−4509	7.474	58.2	4.08	6.80*	14.1	375*
J1600−3053	3.598	52.2	14.3	1.11*	6.8	474*
J1603−7202	14.84	38.1	6.31	1.98*	12.4	212
J1643−1224	4.622	62.4	147	1.94*	14.0	241
J1713+0747	4.570	16.0	67.8	0.20	14.0	392
J1730−2304	8.123	9.61	–	2.52*	14.0	180
J1732−5049	5.313	56.8	5.26	3.23*	6.8	129
J1744−1134	4.075	3.14	–	0.62	13.2	342
J1824−2452	3.054	120	–	1.63*	2.8	89
J1857+0943	5.362	13.3	12.3	1.14*	22.2[a]	376
J1909−3744	2.947	10.4	1.53	0.17	5.2	893
J1939+2134	1.558	71.0	–	15.0[b]	23.3[c]	588
J2124−3358	4.931	4.62	–	4.01*	13.8	415*
J2129−5721	3.726	31.9	6.63	2.19	12.5	177*
J2145−0750	16.05	9.00	6.84	1.88*	13.8	376*

[a] There is a gap of ~11 years between the end of the observations presented by Kaspi et al. (1994) and the beginning of observations with the Parkes telescope. In our analysis we use the Arecibo observations of PSR J1857+0943 only to assist in the estimation of the pulsar parameters and then discard the Arecibo residuals in further processing.

[b] We have altered the value of the phase offsets between different observing systems for these timing residuals compared with the analysis of Verbiest et al. (2009), which lowers our measured rms.

[c] This time series features several large gaps and includes the Kaspi et al. (1994) data.

*These values differ slightly from those presented by Verbiest et al. (2009) because we have removed duplicated observations in five pulsars, and corrected a minor processing error involving the uncertainties on observations made with different observing systems.

this Chapter, we have made minor corrections to the original Verbiest et al. observations. While we have not repeated the ToA estimation process already described and performed by Verbiest et al. (2008, 2009), we have removed erroneous duplicated ToAs from some pulsars in the data set. Also, in forming the timing residuals, we have treated the observations of every pulsar in the same manner when measuring the arbitrary phase offsets between different backend systems. This caused the PSR J1939+2134 timing residuals to exhibit a lower rms than in the original Verbiest et al. data set because the value of these offsets changed. A summary of the data set used in this Chapter is given in Table 6.1.

In our data set, the data-spans vary widely, ranging from 2.8 years for PSR J1824−2452 to 23.3 years for PSR J1939+2134. The weighted rms residual also varies over two orders of magnitude, from 170 ns for PSR J1909−3744 to 15 μs for PSR J1939+2134. The residuals are unequally spaced and the sample times are different between pulsars. As described in Section 2.2.3, for some pulsars the average magnitude of the ToA error bar changes discontinuously at a particular point in the time series because of upgrades in the observing hardware. For most pulsars, the upgrade that caused the most significant change in the ToA uncertainty was the transition from the FPTM backend system to CPSR2 in the year 2002. For pulsars whose ToA uncertainties significantly improved after this upgrade, a weighted fit to the residuals would be mostly influenced by the most recent observations, thus reducing the GWB sensitivity of such data sets. To ameliorate this effect, we attempt to reduce the variation in the magnitude of the ToA uncertainties so that, in subsequent weighted estimates using the timing residuals, the weights are spread more evenly across the data set.

In Table 6.2, we provide a list of the pulsars whose ToA uncertainties exhibit a "step-change" in average magnitude. For these pulsars, we have calculated the unweighted variance of the residuals both before and after the upgrade that caused this step-change. These variances are added in quadrature with the original error bars in each portion of the time series before commencing any further processing. For all other pulsars, there was no significant change in data quality at the epoch of the hardware change. We thus calculate the unweighted variance of the whole time series and add it in quadrature with the original error bars before any further processing. This simple process increases the uncertainty on all ToAs[54]. However, the uncertainty

[54] This process of augmenting the error bar on each residual is ad hoc because we do not have a good error model for the PPTA timing residuals. Improving the white noise model is an important goal of PPTA research at present. The basic problem is that we know that there are other white noise processes that affect the timing residuals in addition to radiometer noise, such as pulse jitter (Cordes & Shannon, 2010; Oslowski et al., 2011). However,

Table 6.2: Pulsars with non-stationary timing residuals. For these pulsars, we estimate the unweighted rms of the residuals before and after an important hardware change at the telescope.

PSRJ	Type of change	Epoch (MJD)	RMS before change (μs)	RMS after change (μs)
J1600−3053	backend	52654.0	9.61	1.31
J1713+0747	backend	52462.5	1.24	0.48
J1732−5049	backend	52967.5	7.57	4.03
J1744−1134	backend	52462.6	1.54	1.29
J2124−3358	backend	52984.5	9.74	4.64
J2129−5721	receiver	51410.0	5.47	3.48
J2145−0750	backend	52975.5	4.14	3.17

on the most precise ToAs will increase by more than the corresponding increase for less precise ToAs, meaning that there will be less variation in the weight of each residual across the time series.

6.2 Method

The GWB-induced residuals are correlated between different pulsar pairs as shown in Figure 1.8. Although limits on the amplitude of the GWB can be obtained from the residuals of a single pulsar (e.g., Kaspi et al., 1994), the GWB can only be detected with confidence by observing this pairwise correlation. We now describe our technique for detecting a GWB signal in pulsar timing residuals.

6.2.1 Detecting the Gravitational-Wave Background Signal

The expected GWB signal in pulsar timing residuals was described in Section 5.1.1. For this analysis, we assume that the GWB due to SMBHBs is described by equations (5.1) and (5.2). In this case, the cross-power spectrum between the induced ToA perturbations in pulsars i and j is

$$X_{ij}(f) = P_g(f)\zeta\left(\theta_{ij}\right) \ , \tag{6.1}$$

where $X_{ij}(f)$ is the value of the cross-power spectrum at frequency f, $P_g(f)$ is the power spectrum of the GWB-induced ToA perturbations given in Equation (5.2) and $\zeta\left(\theta_{ij}\right)$ is given in Equation (1.15).

we don't yet know what processes are involved or how much they contribute to the timing residuals. An interim solution is now available in the form of the FIXDATA plugin to TEMPO2.

In order to detect the GWB signal, we estimate $X_{ij}(f)$ for each pair of pulsars. As the spectrum of the GWB is very steep, only the lowest frequencies are of interest. Also, because we want to detect the correlated GWB signal between pulsars, we will focus on the overlapping portion of each pair of pulsars. The observations of each pair of pulsars overlap over some time-span, $T_{\rm overlap}$. For $N_{\rm psr} = 20$ there are $N_{\rm pairs} = 190$ pairs. For each pair we estimate the cross-power spectrum at harmonics of $f = 1/T_{\rm overlap}$. If the sampling were uniform and the ToA uncertainties were equal, these estimates would be uncorrelated. In practice we find that they are not uncorrelated and this reduces the sensitivity of our detection algorithm. It is probable that the independence can be restored using the Cholesky spectral estimation procedure recently discussed by Coles et al. (2011). However, this is beyond the scope of this thesis.

For some pairs, $T_{\rm overlap}$ can be much smaller than the length of one or both time series. For our time series, $T_{\rm overlap}$ ranges from just 0.8 yr for PSRs J0437$-$4715 and J1824$-$2452, to 14.1 yr for PSRs J0711$-$6830 and J1939+2134. The use of only the overlapping residuals causes a bias in the cross-power spectral estimates, the cause of which is currently not known. We correct this bias by removing a quadratic function from the overlapping section of the two time series using a weighted least-squares (WLSQ) fit, as shown in Figure 6.6. This fit is in addition to the standard timing model fit that estimates the pulsar parameters. We estimate the cross-power spectrum using:

$$X_{ij}(f) = \mathcal{F}_i(f)\mathcal{F}_j^*(f)/T_{\rm overlap}\ , \tag{6.2}$$

where $\mathcal{F}_i$ denotes the DFT of the timing residuals of pulsar i and * denotes complex conjugation. We use the following standard definition of the one-sided DFT (a factor of two larger than the two-sided DFT given in Equation 2.2):

$$\mathcal{F}(f_k) = \frac{2}{N_{\rm pts}} \sum_{n=0}^{N_{\rm pts}-1} r_n \mathrm{e}^{-2\pi \mathrm{i} kn/N_{\rm pts}}, \tag{6.3}$$

where $\mathrm{i} = \sqrt{-1}$ in this particular equation, r_n is the n-th residual and k is an integer between 1 and $(N_{\rm pts} - 1)/2$, rounded down. Calculating the DFT is not trivial because of the uneven sampling and variable error bars. We calculated $\mathcal{F}_i(f_k)$ for every pulsar using a WLSQ fit of a sine term plus a cosine term at each $f_k = k/T_{\rm overlap}$. This gives identical results to a weighted Lomb-Scargle estimate of the spectrum (Scargle, 1982; Zechmeister & Kürster, 2009). The variance of each cross-power spectral estimate is

$$\sigma^2_{X_{ij}}(f) = \langle P_i(f)\rangle \langle P_j(f)\rangle / 2 \,, \tag{6.4}$$

where $\langle ... \rangle$ indicates an expectation value and $P_i(f)$ is the spectral estimate of the residuals of pulsar i at frequency f. In practice, we calculate these expectation values using a power-law fit to the lowest frequencies in the periodogram of each pulsar. This power-law fit gives a spectral model for low frequencies in this pulsar.

We account for the effects of fitting the timing model to the observations using two Monte Carlo simulations. The first simulation estimates the effective transfer function for each pulsar (see Section 2.4.3). The transfer function for each pulsar is similar to that shown in Figure 2.10 for PSR J0613−0200. We then correct the measured cross-power spectrum for each pulsar pair at each frequency by dividing by the geometric mean of the transfer functions of the two pulsars at that frequency. This correction is common between our analysis and that of Verbiest et al. (2009), but this is the only pulsar parameter fitting correction performed by Verbiest et al. (2009).

However, this process of correcting the cross-spectrum using TEMPO2 transfer functions can only correct the effects of the timing model fit as it acts on white noise in the residuals. This is because, although fitting the timing model is a linear operation, it is not a filter. In particular, this means the effect of the timing model fit will be different when the residuals are contaminated by red noise, compared to the case where the residuals are consistent with white noise. When a set of residuals with time-span $T_{\rm obs}$ is affected by red noise, fitting the full timing model to the residuals reduces $P(f = 1/T_{\rm obs})$ by significantly more than the white noise transfer function. This is easily confirmed by simulation. A second correction is therefore necessary to measure the effect of the full timing model fit on the non-white GWB contribution to the residuals. We simulate $\sim$10000 realisations of the residuals and add a simulated GWB signal with $A = 3 \times 10^{-15}$ and $\alpha = -2/3$ to all pulsars using the methods described in Sections 2.4.3 and 5.1.2. This value of A was chosen because it gives the largest GWB signal that is still small compared with the noise, hence reducing the number of required simulations. We further reduce the number of simulations by fixing every pulsar to be at the same position and distance, giving the maximum correlated GWB signal between pulsars. We perform the full pulsar parameter fit using TEMPO2, estimate the cross-power spectrum in each realisation and apply the transfer function correction described above. We divide the average corrected cross-power spectrum of each pulsar pair by the theoretical level of the cross-power spectrum given in Equation (6.1). This process defines a set of "calibration factors", $\gamma_{ij}(f_k)$. When forming subsequent estimates of the cross-power spectrum using Equation (6.2), we calibrate each estimate at the lowest three frequencies of the cross-power spectrum by dividing the cross-power-spectral estimate for pulsars i and j at frequency f_k by $\gamma_{ij}(f_k)$.

After performing both of these corrections, we estimate A^2. For each frequency channel, f_k, of the cross-power spectrum (measured in yr^{-1}), we have (see equations 1.14 and 6.1)

$$\left[A_{ij}^2\zeta\left(\theta_{ij}\right)\right]_k = 12\pi^2 f_k^{3-2\alpha}\text{Real}\left[X_{ij}(f_k)\right] \tag{6.5}$$

where A_{ij}^2 indicates the measurement of A^2 obtained from pulsars i and j and Real $\left[X_{ij}(f_k)\right]$ is the real part[55] of the cross-power spectrum. The variance of $A_{ij}^2\zeta\left(\theta_{ij}\right)$ is then proportional to the variance of X_{ij}.

To compare directly with the technique of Jenet et al. (2005), we perform a weighted sum of the $A_{ij}^2\zeta\left(\theta_{ij}\right)$ estimates over cross-spectral frequency to obtain a single estimate of $A_{ij}^2\zeta\left(\theta_{ij}\right)$ for each pulsar pair. This gives

$$A_{ij}^2\zeta\left(\theta_{ij}\right) = \frac{12\pi^2\sum_k X_{ij}(f_k)k^{2\alpha-3}/\sigma_{X_{ij}}^2(f_k)}{\left(T_{\text{overlap}}\right)^{3-2\alpha}\sum_k k^{4\alpha-6}/\sigma_{X_{ij}}^2(f_k)} \tag{6.6}$$

where both summations range from $k = 1$ to $N_{\text{spec},ij}$, and where $N_{\text{spec},ij}$ is the number of cross-spectral frequencies for pulsars i and j. This final estimate of $A_{ij}^2\zeta\left(\theta_{ij}\right)$ is similar to the unnormalised covariance between the residuals of pulsars i and j. We also use the observed scatter in estimates of $A_{ij}^2\zeta\left(\theta_{ij}\right)$ obtained from simulated observations to estimate the uncertainty $\delta A_{ij}^2\zeta\left(\theta_{ij}\right)$ for each pulsar pair.

Having fully calibrated our technique using simulations, we estimate the squared amplitude of the GWB, $\hat{A}^2$, by forming an average of the $A_{ij}^2\zeta\left(\theta_{ij}\right)$ estimates weighted by the inverse variance of each estimate. In practice this average is done by performing a WLSQ fit to find the amplitude $\hat{A}^2$ (and its corresponding uncertainty) for which the quantity $\hat{A}^2\zeta$ best fits the observed values of $A_{ij}^2\zeta\left(\theta_{ij}\right)$. For ease of notation, we index over all possible pulsar pairs using m, where m is an index running from 1 to N_{pairs} and we set $\zeta_m \equiv \zeta\left(\theta_{ij}\right)$. In this case, the

[55] The imaginary part of the cross-power spectrum contains uncorrelated terms and is discussed in Section 6.4.

expression for $\hat{A}^2$ is

$$\hat{A}^2 = \frac{\sum_m \left[A_m^2 \zeta_m\right] \zeta_m / \sigma^2_{A_m^2 \zeta_m}}{\sum_m \zeta_m^2 / \sigma^2_{A_m^2 \zeta_m}} = \frac{\sum_m A_m^2 / \sigma^2_{A_m^2}}{\sum_m 1 / \sigma^2_{A_m^2}} \tag{6.7}$$

and its unweighted variance is

$$\sigma^2_{\hat{A}^2} = \frac{1}{\sum_m \zeta_m^2 / \sigma^2_{A_m^2 \zeta_m}} = \frac{1}{\sum_m 1 / \sigma^2_{A_m^2}} \,. \tag{6.8}$$

This initial estimate of the error assumes that each of the $\delta A_{ij}^2 \zeta\left(\theta_{ij}\right)$ is well-estimated. If this is not true, then we need to augment the error on $\hat{A}^2$ by an extra term that describes the amount of scatter in the residuals. This corresponds to accounting for a non-unity reduced χ^2 of the WLSQ fit that determines $\hat{A}^2$. Thus our final estimate for the variance of $\hat{A}^2$ is

$$\begin{aligned} \sigma^2_{\hat{A}^2} &= \frac{1}{(N_{\text{pairs}} - 1)} \frac{\sum_m \left(\left[A_m^2 \zeta_m\right] - \hat{A}^2 \zeta_m\right)^2 / \sigma^2_{A_m^2 \zeta_m}}{\sum_m \zeta_m^2 / \sigma^2_{A_m^2 \zeta_m}} \\ &= \frac{1}{(N_{\text{pairs}} - 1)} \frac{\sum \left(A_m^2 - \hat{A}^2\right)^2 / \sigma^2_{A_m}}{\sum 1 / \sigma^2_{A_m}} \,, \end{aligned} \tag{6.9}$$

which is just the weighted estimate of the variance of $\hat{A}^2$. If $\hat{A}^2$ is significantly larger than $\sigma_{\hat{A}^2}$, then a detection of the GWB has been made. This algorithm has been implemented as a TEMPO2 plugin, which is included in Appendix B.

6.3 Results

From our data set we estimate the squared GWB amplitude to be $\hat{A}^2 = -4.5 \times 10^{-30}$, with an uncertainty $\sigma_{\hat{A}^2} = 9.1 \times 10^{-30}$. Our result is consistent with the null hypothesis, that there is no GWB present. Although the estimate is negative and therefore would lead to an unphysical GWB, it is not improbable because the standard deviation is a factor of 2 larger than the magnitude of the mean[56]. We simulated many realisations of the observations using the method of Section 2.4.3, including the uncertainty given by the ToA error bars and a random process consistent with the low-frequency spectrum of the residuals but no GWB signal. These simulations showed that our estimate is consistent with the null hypothesis with 76% confidence. This result

[56] This would not be an issue for a Bayesian approach to GWB detection wherein the prior can restrict the value of $\hat{A}^2$ to being positive.

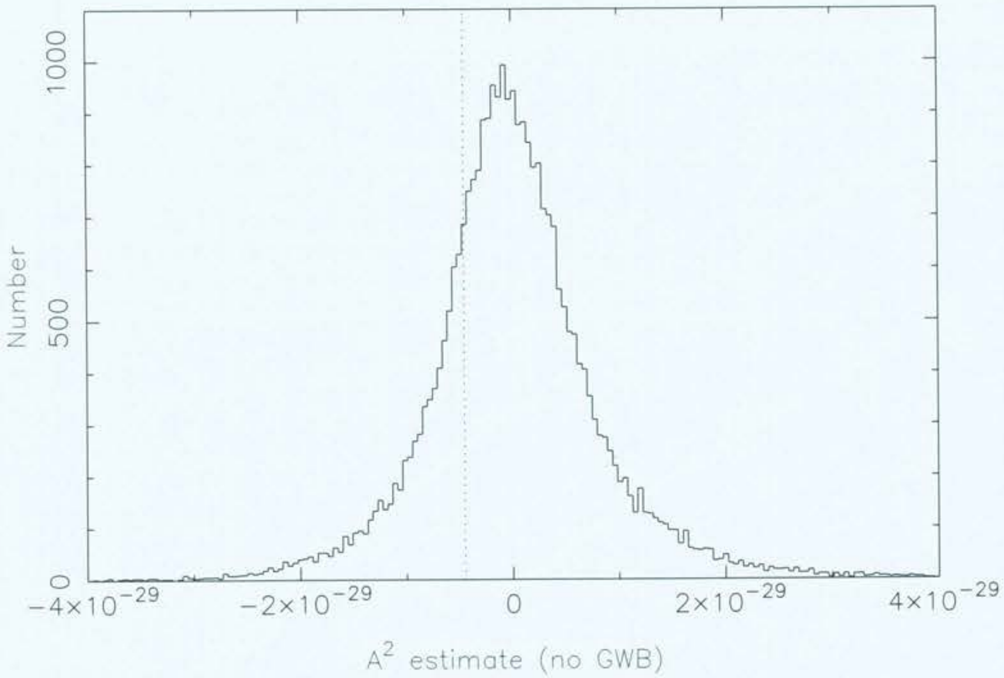

Figure 6.1: The histogram shows the distribution of $\hat{A}^2$ for simulations of our residuals with no GWB present. The thin dotted line shows the value of $\hat{A}^2$ obtained from the observations. The estimates to the right of the dotted line include 76% of the simulation results. All physical GWBs have $A^2 > 0$.

is shown as a histogram in Figure 6.1. At first, it appears that this histogram could be used to provide a 95% confidence upper bound on the GWB amplitude. However, as discussed further below, any limit thus obtained would not take account of "self-noise" (Jenet et al., 2005) due to the GWB-induced perturbations at the pulsar.

In Figure 6.2, we plot the 15 estimates of $A^2_{ij}\zeta\left(\theta_{ij}\right)$ with the smallest uncertainties. It is clear from this Figure that the current noise levels are larger than 4.5×10^{-30} and that our result is consistent with the null hypothesis. The dot-dashed curve for $A^2 = 10^{-28}$ seems to imply that such a large GWB signal is ruled out by the observations. These observations probably do rule out such a GWB signal (though this has not been investigated), but if A^2 were actually 10^{-28} the noise levels on each $A^2_{ij}\zeta\left(\theta_{ij}\right)$, which provide the upper bound, would be much higher. As the noise levels come from the power spectrum of the residuals of each pulsar, obtaining an upper bound using the noise levels is equivalent to obtaining an upper bound directly from the individual power spectra and ignoring the cross correlations. Such a bounding technique is not pursued in this Chapter.

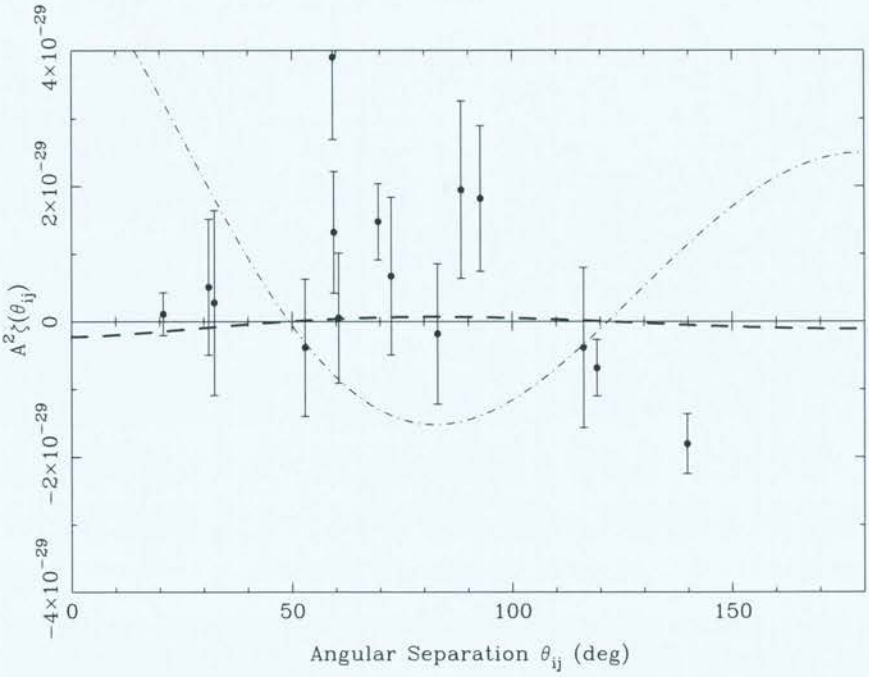

Figure 6.2: The 15 most precise estimates of $A_{ij}^2\zeta\left(\theta_{ij}\right)$ obtained from our data set (points with error bars), the best-fit value of $\hat{A}^2\zeta = -4.5\times10^{-30}\times\zeta$ (dashed curve) and the signal expected from a strong GWB with $A^2 = 10^{-28}$ (dot-dashed curve).

6.4 Discussion

The results of applying this algorithm to the corrected version of the Verbiest et al. (2008, 2009) data are disappointing in the sense that the sensitivity is considerably poorer than that calculated in the Appendix provided by Verbiest et al. (2009). We believe the estimated errors to be correct because they are calibrated by simulation, so we ask the question: *Why are the cross-power spectra of the GWB lower than expected?* To investigate this we have run a series of simulations[57] with GWB signals of differing amplitudes injected into the observations. The results are shown in Figure 6.3. The mean values of the derived $\hat{A}^2$ are plotted as solid lines connecting error bars (that indicate the uncertainty in the mean) for two cases: (1) the algorithm including correction with the γ_{ij} calibration factors (thick solid line); and (2) the algorithm with $\gamma_{ij}\equiv 1$ (thin solid line). These results show that our method returns a GWB amplitude estimate $\hat{A}^2_{\rm out}$ such that, on average, $\hat{A}^2_{\rm out} = A^2_{\rm in}$. Figure 6.4 shows that this GWB signal is at the correct

[57] These simulations use a spread of pulsar distances and synthesise residuals with the same sampling as our observations, using the methods of Section 2.4.3 and 5.1.2. The simulated residuals include white noise consistent with the observed error bars, red noise consistent with the spectral model mentioned in Equation (6.4) and a signal from a GWB with $\alpha = -2/3$ and with a range of amplitudes between $A^2 = 6.4\times10^{-33}$ and $A^2 = 4\times10^{-28}$. We did not perform post-Keplerian pulsar parameter fits.

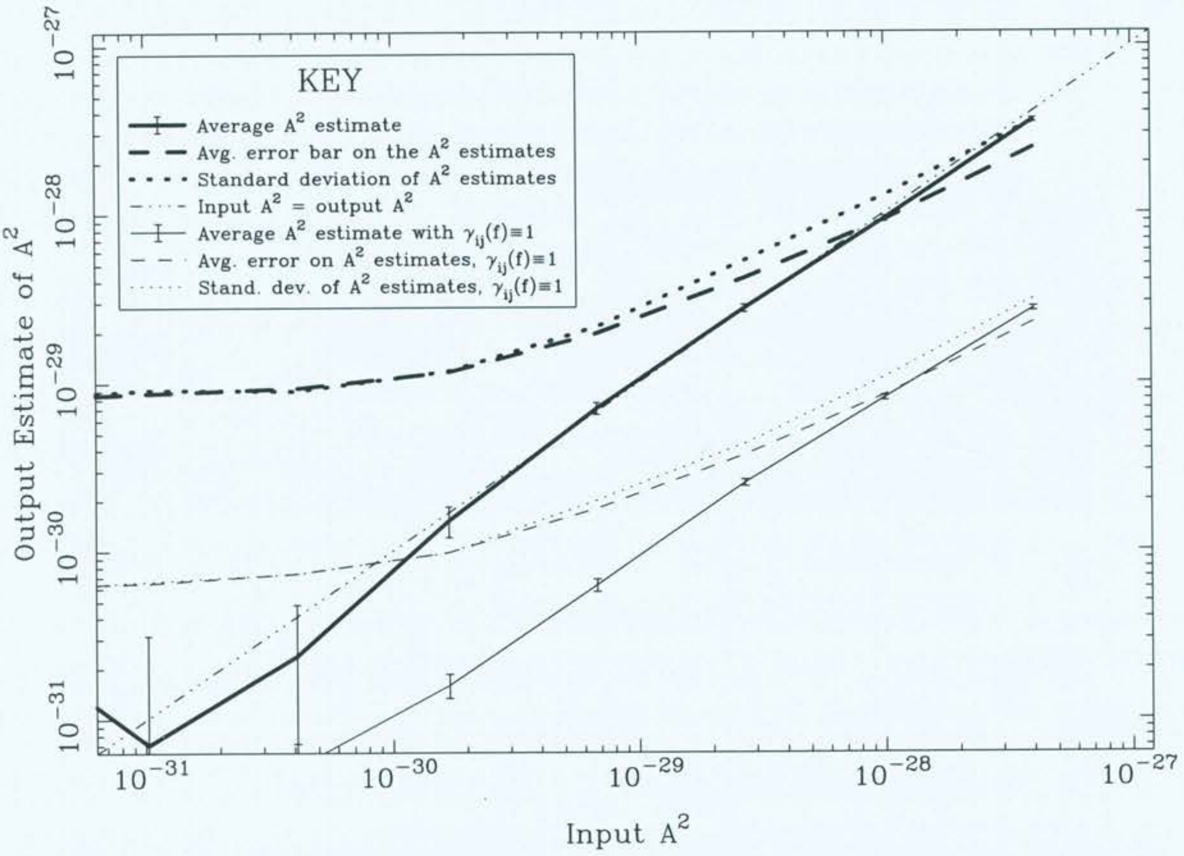

Figure 6.3: Average $\hat{A}^2$ as a function of input GWB A^2 for our residuals. The ordinate gives the average output $\hat{A}^2$ from our detection algorithm. The triple-dot-dashed line indicates points where the input A^2 is equal to the output $\hat{A}^2$. We have considered 2 cases: performing the full detection procedure (thick lines) and the uncalibrated detection procedure that uses $\gamma_{ij}(f) \equiv 1$ (thin lines). In both cases we have averaged over 1400 realisations for each input A^2, and estimated the average output $\hat{A}^2$ (solid lines), where the error bars give the error in the mean of $\hat{A}^2$. The dashed lines give the square root of the average of $\sigma^2_{\hat{A}^2}$ in each case, and are in good agreement with the sample standard deviations over the amplitude range of interest (dotted lines).

level on average in every pulsar pair. The difference between the thick solid line and the thin solid line in Figure 6.3 indicates that the GWB power is reduced by a factor of ~12 because of the pulsar parameter fitting, even after adjusting the cross-power spectra using the effective transfer function.

We can estimate the amount of GWB signal lost in estimation of different timing parameters by calculating the weighted average calibration factor in the lowest frequency channel of each pulsar pair. While this will be at a different frequency for each pair, it nevertheless provides a straightforward figure of merit for comparing the effect of fitting dif-

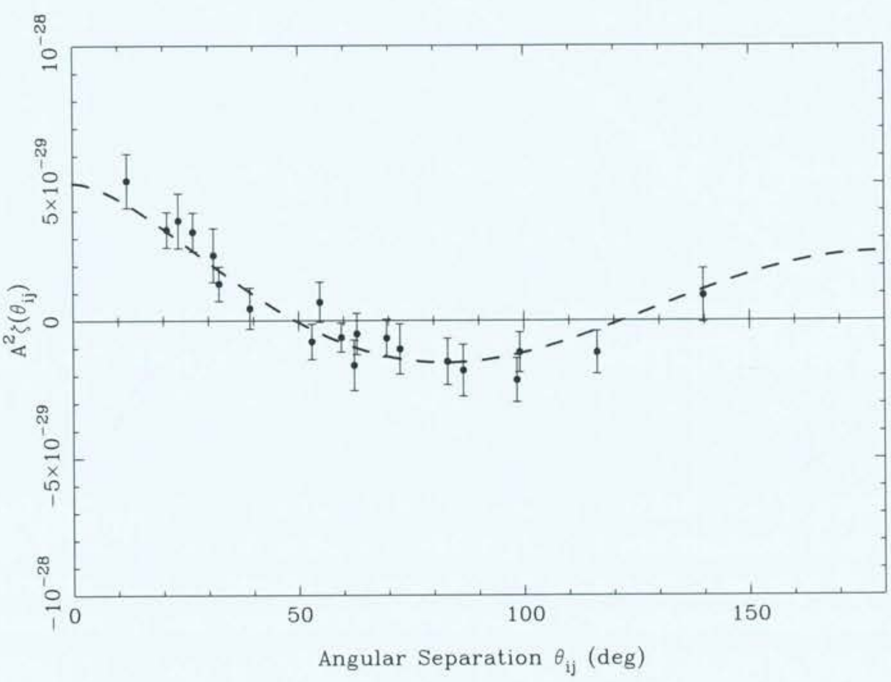

Figure 6.4: The expected covariance in simulated residuals that include a GWB component with squared amplitude $A^2 = 10^{-28}$. The smooth dashed curve corresponds to the theoretical covariance for an input $A^2 = 10^{-28}$. The points correspond to the mean of the estimates of $A_{ij}^2\zeta\left(\theta_{ij}\right)$ (Equation 6.6) from 200 simulated sets of timing residuals for the 20 PPTA pulsars. The error bars give the uncertainties in these mean estimates. For clarity we only plot the 20 pairs with the smallest rms scatter in their estimates of $A_{ij}^2\zeta\left(\theta_{ij}\right)$ over the 200 simulations.

Table 6.3: The effect of fitting different combinations of timing model parameters on the GWB signal in the lowest frequency channel. Values in the 4th column are the inverse of values in the 2nd column. The symbols are: ν (pulse frequency); $\dot{\nu}$ (pulse frequency derivative); "JUMP" (arbitrary phase offsets between different observing systems were removed from all pulsars); "ALL" (all timing model parameters were fit).

Timing Model Parameters	Weighted mean of $\gamma_{ij}(f = 1/T_{\rm overlap})$	Uncertainty in Weighted Mean	Sensitivity Loss Factor
ν, $\dot{\nu}$	0.1716	0.0003	5.83
ν, $\dot{\nu}$, JUMP	0.0796	0.0002	12.6
ALL	0.0790	0.0002	12.7

ferent timing model parameters. For the full TEMPO2 fit acting on our residuals, we find $\overline{\gamma_{ij}}(f = 1/T_{\rm overlap}) = 0.0790 \pm 0.0002$, which represents an average loss of $0.0790^{-1} = 12.7$ in GWB signal at $f = 1/T_{\rm overlap}$. This explains the large decrease in sensitivity of our method compared to that presented in the Appendix of Verbiest et al. (2009), which did not fully account for the effect of pulsar parameter estimation on the GWB signal. In Table 6.3, we give the weighted average calibration factor at $f = 1/T_{\rm overlap}$ when fitting for different parameters in the timing model. The estimation of the pulsar position and parallax have little effect on $\overline{\gamma_{ij}}(f = 1/T_{\rm overlap})$ since $T_{\rm overlap}$ is a few times greater than 1 yr for most of our pulsar pairs, and so are not shown in Table 6.3. This table indicates that one can almost determine the complete effect of fitting on the GWB sensitivity by only including fits for the spin frequency, its derivative and the arbitrary phase offsets between different observing systems. Additionally, while the spin frequency derivative fit only significantly affects the power in the lowest frequency channel, the arbitrary phase offsets affect the power in the lowest few channels and hence can significantly affect our estimate of A^2.

The dashed lines in Figure 6.3 show that for GWB amplitudes around $A^2 = 5 \times 10^{-30}$, the average uncertainty on $\hat{A}^2$ is double the average uncertainty when there is no input GWB. This extra contribution to the uncertainty comes from the effect of the GWs passing near the pulsar, which we refer to as the self-noise of the GWB. For larger values of A^2, the uncertainty on $\hat{A}^2$ is dominated by the GWB self-noise as discussed by Jenet et al. (2005).

For comparison with previous limits, we attempted to place a 95% confidence upper bound as follows. Using the same simulations that produced Figure 6.3, we attempted to find the amplitude of a simulated GWB that gave a measurement of A^2 larger than -4.5×10^{-30} – the value obtained from the actual observations – with probability 0.95. The results, shown in

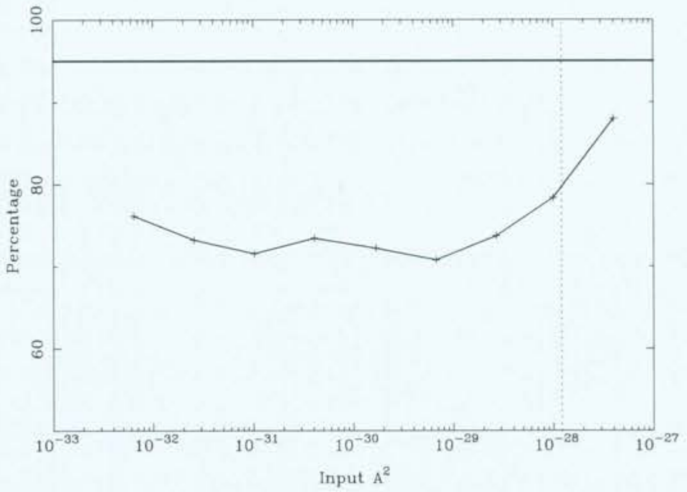

Figure 6.5: Percentage of A^2 estimates from simulated observations above the observed value of $A^2 = -4.5 \times 10^{-30}$, as a function of input simulated A^2. The dotted vertical line shows the limit of $A^2 < 1.2 \times 10^{-28}$ (Jenet et al., 2006). The percentage of A^2 estimates above the observed value of $A^2 = -4.5 \times 10^{-30}$ (thin solid line) does not reach 95% (thick horizontal line) within the plotted range of simulated A^2 values.

Figure 6.5, show that the percentage of estimates of A^2 above -4.5×10^{-30} does not reach 95% for any simulated $A^2 \leq 4 \times 10^{-28}$. We traced this to the issue that the GWB sensitivity of the different time series varies widely between the different pulsars analysed by Verbiest et al. (see Section 6.4.4). The maximum simulated GWB amplitude shown in Figure 6.5 is much larger than the Jenet et al. (2006) upper bound ($A^2 \leq 1.2 \times 10^{-28}$). Thus we cannot obtain a sufficiently low upper bound with 95% confidence to warrant further investigation with our current time series and weighting scheme.

Furthermore, any limit obtained in this way would be considerably worse than one obtained through other methods, such as direct power estimation, because of the huge variation in noise levels amongst our pulsars[58]. A power spectral analysis of the Verbiest et al. (2008, 2009) similar to that presented by Jenet et al. (2006) is expected to provide a much lower bound on the GWB amplitude than cross-correlation analysis.

We confirm the accuracy of the measured uncertainty on each estimate of $A^2_{ij}\zeta\left(\theta_{ij}\right)$ using

[58] We cannot apply the Jenet et al. (2006) limit method to these observations because it requires that the timing residuals of each pulsar be white. The method presented by van Haasteren et al. (2009) could be applied to these observations, but this would require a large amount of computation time and any limit obtained would be difficult to confirm via Monte Carlo simulation.

the reduced χ^2 of the WLSQ fit that determines $\hat{A}^2$. The reduced χ^2 of this fit is

$$\chi_{\mathrm{r}}^2 = \frac{1}{(N_{\mathrm{pairs}} - 1)} \sum_k \frac{\left([A_k^2\zeta_k] - \hat{A}^2\zeta_k\right)^2}{\sigma^2_{A_k^2\zeta_k}} , \tag{6.10}$$

which has a value of 1.3 for our residuals, indicating that the uncertainty estimates $\sigma_{A_k^2}$ are consistent with the rms variation of the estimates A_k^2. We obtain an independent estimate of the accuracy of the measured errors by making use of the information contained in the imaginary part of the cross-power spectrum, which we denote $\mathrm{Imag}\,[X_{ij}(f)]$. We calculate $\mathrm{Imag}\left[A_{ij}^2\zeta\left(\theta_{ij}\right)\right]$ by evaluating Equation (6.5) with $\mathrm{Imag}\,[X_{ij}(f)]$ in place of $\mathrm{Real}\,[X_{ij}(f)]$. We then process $\mathrm{Imag}\left[A_{ij}^2\zeta\left(\theta_{ij}\right)\right]$ in exactly the same way as the real part is processed. Since correlation coefficients are real, we expect that $\mathrm{Imag}\left[A_{ij}^2\zeta\left(\theta_{ij}\right)\right]$ will contain no correlated signal. This means that we can calculate the analogue of the reduced χ^2 using $\mathrm{Imag}\left[A_{ij}^2\zeta\left(\theta_{ij}\right)\right]$:

$$\chi_{\mathrm{r,im}}^2 = \frac{1}{(N_{\mathrm{pairs}} - 1)} \sum_k \frac{\left(\mathrm{Imag}\,[A_k^2\zeta_k]\right)^2}{\sigma^2_{A_k^2\zeta_k}} . \tag{6.11}$$

Similar to the reduced χ^2, if the errors on $A_{ij}^2\zeta\left(\theta_{ij}\right)$ are well-estimated then this quantity should be near unity. For our residuals, we find $\chi_{\mathrm{r,im}}^2 = 0.87$, indicating that the errors are well-estimated.

Although both χ_{r}^2 and $\chi_{\mathrm{r,im}}^2$ show that the uncertainties $\sigma_{A_k^2}$ are reliable on average, these uncertainties come from power spectral estimates so they are random variables. We estimated the sensitivity of $\hat{A}^2$ to variations in $\sigma_{A_k^2}$ by multiplying each $\sigma_{A_k^2}$ by a random factor, distributed as the square root of the product of two χ^2 random variables with two degrees of freedom. This is the expected distribution for each $\sigma_{A_k^2}$. We found that $\sigma_{\hat{A}^2}$ increased by a factor of 1.6, indicating that the use of incorrect $\delta A_{ij}^2\zeta\left(\theta_{ij}\right)$ estimates degrades the sensitivity of the $\hat{A}^2$ measurement by only a factor of 1.6.

However, the $A_{ij}^2\zeta\left(\theta_{ij}\right)$ are not Gaussian; rather they come from the sum of two pairwise products of independent Gaussian variables and thus have a two-sided exponential distribution that is reflected in Figure 6.1. This means that the maximum likelihood estimator for A^2 is not a WLSQ estimator but a weighted least absolute deviation (LAD) fit (e.g., Cox, 2006). We tested both weighted and unweighted LAD fits and found that the results for WLSQ and unweighted LAD fits were very similar, while the weighted LAD fit introduced a small bias in the mean. These results are shown in Table 6.4. We suspect that the bias occurs because any LAD fit

Table 6.4: The results from estimating A^2 with different estimators averaged over 10^5 simulations of realistic residuals including a GWB with $A^2 = 10^{-30}$.

Estimator	Mean $\hat{A}^2$	Error in Mean	rms of $\hat{A}^2$
	$(\times 10^{-30})$	$\hat{A}^2$ $(\times 10^{-30})$	$(\times 10^{-30})$
WLSQ [our method]	0.99	0.038	12
Unweighted LAD	1.0	0.038	12
Weighted LAD	0.84	0.041	13

Table 6.5: The results from our observations using different methods of spectral analysis of the timing residuals.

Processing	$\hat{A}^2$	$\sigma_{\hat{A}^2}$
Performed	$(\times 10^{-30})$	$(\times 10^{-30})$
Smoothing & Interpolation	3.0	10
Smoothing only	-7.8	10
No smoothing [our method]	-4.5	9.1

includes a 'dead-zone' feature, where a range of parameter estimates give the same minimum absolute deviation. This dead zone is negligible when the number of estimates is large, but can be significant otherwise. Since our A^2 estimates are dominated by a small number of A_k^2 measurements and the results of the different estimators are similar, we chose the more standard WLSQ fit in calculating $\hat{A}^2$. Although the WLSQ estimator is not maximum likelihood, it is apparently more robust in our particular case.

Estimation of A^2 is also largely independent of changes to the method of spectral analysis. We experimented with reducing the white noise in the residuals by smoothing each time series over a 60-d period before commencing the spectral analysis. We also tested interpolation using a constrained cubic spline of each smoothed time series onto a 14-d grid common to all pulsars before the spectral analysis. The results of these different approaches are given in Table 6.5. Since there was no statistically significant difference between the different approaches, for simplicity we elected not to smooth or interpolate the residuals.

6.4.1 Treatment of a Large Amplitude Gravitational-Wave Background

For their detection statistic, Jenet et al. (2005) calculated the normalised cross correlation between the timing residuals of each pulsar pair. They optimised the S/N ratio using a filter designed to whiten the residuals before correlation. For a simulation of the 20 PPTA pulsars, this approach increased the maximum achievable detection significance for a GWB from 3σ to 13σ. However, their filter cannot be applied to real pulsar timing observations without modification. We investigated the effect of such a filter by performing simulations of our residuals where each simulation included a signal from a GWB with $A \gtrsim 3 \times 10^{-15}$. In the frequency domain, the filter takes the form of a weighting factor, so we optimised this weighting factor to match the large input GWB amplitude. We found that this method did not improve the S/N ratio, and we traced this under-performance to the problem of spectral leakage from the lowest frequencies to the higher frequencies. We found that the first few cross-spectral estimates, which make the largest contribution to our detection statistic, were all more than 90% correlated with the lowest spectral estimate (i.e., at frequency $f = 1/T_{\rm overlap}$), meaning that re-weighting cannot change the overall S/N ratio. The spectral leakage is particularly significant because of the irregular sampling and variable ToA uncertainties in these observations. We expect that an improved spectral analysis technique (e.g., Coles et al., 2011) will eliminate the spectral leakage and enable us to take advantage of more degrees of freedom[59] when the GWB signal is larger than the noise.

6.4.2 Fitting Timing Models over Different Data-Spans

The time series we consider in this Chapter have widely varying time-spans. The effect of such variation has not been considered in most PTA analyses to date. As part of the pulsar parameter estimation, we fit for the pulse period and its derivative over the full duration of each time series. Originally, we then computed the cross-power spectra from the overlapping portion of residuals of each pulsar pair with no further processing. However, upon simulating this procedure, we found that the lowest frequencies in the cross-power spectra were biased whenever $T_{\rm obs} > T_{\rm overlap}$. This bias took the form of a significantly non-zero imaginary part in the cross-power spectrum. Also, we found that much of the correlated signal at low frequencies was removed, as shown in Figure 6.6. We were unable to eliminate these effects unless we performed a WLSQ fit of a quadratic function for each time series over the overlapping time range. This restores the correlation in the GWB signal between different pulsars (right panels of Figure 6.6). This additional WLSQ fit will introduce a new bias because of removing some of

[59] In contrast to Verbiest et al. (2009), who state that quadratic fitting removes one degree of freedom from the power spectrum of each pulsar's residuals, we have shown that quadratic fitting does not affect the number of degrees of freedom in the lowest few frequency channels of each power spectrum.

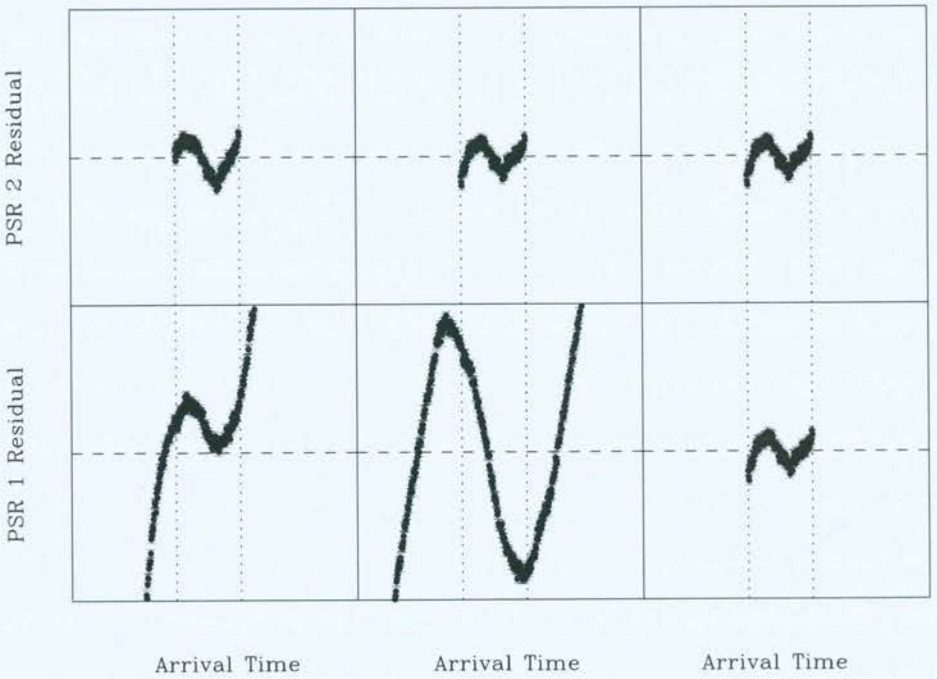

Figure 6.6: The effect of fitting a timing model over different data-spans. The simulated time series in the upper three panels are 5 years long, those in the lower three panels are 15 years long (the longer time series in the first 2 panels have been truncated because deviation in the y-direction has the same magnitude in each panel). The bottom right panel only includes the overlapping simulated observations. The vertical dotted lines indicate the overlapping timing residuals for these time series. We added the same large signal to both time series and the time series are identical in the overlapping region (left panels). After fitting the timing model (middle panels), this signal is no longer correlated between the two time series. The correlation is restored by performing a WLSQ fit of a quadratic function in the overlapping region of the two time series (right panels).

Table 6.6: The results from using updated realisations of TT and the solar system ephemeris. The last column gives the change in the value of $\hat{A}^2$ with respect to processing the observations with TT(TAI) and DE405, the realisations used for our data set.

Realisation of Terrestrial Time	Solar System Ephemeris	$\hat{A}^2$ $(\times 10^{-30})$	$\sigma_{\hat{A}^2}$ $(\times 10^{-30})$	Change in $\hat{A}^2$ $(\times 10^{-30})$
TT(TAI)	DE405	-4.5	9.1	0.0
TT(TAI)	DE421	-2.3	9.4	2.2
TT(BIPM2010)	DE405	-3.7	8.7	0.8

the GWB signal at $f = 1/T_{\text{overlap}}$, but this new bias is easily corrected with the calibration factors $\gamma_{ij}(f_k)$. However, there is an additional loss of 10% of the GWB signal in our observations because of this extra WLSQ fit.

6.4.3 Correlated Signals in the Timing Residuals

The GWB analysis is complicated by the unknown effects of other correlated signals in the timing residuals. Instabilities in TT(TAI) and errors in the solar system ephemeris both produce signals that are correlated between different pulsars. An instability in TT(TAI) will affect all pulsars in the same way, inducing a correlated signal that is independent of the angular separation of the pulsars on the sky. This would lead to a positive offset in the correlation curve in Figure 1.8. An inaccuracy in the solar system ephemeris will typically induce residuals that are positively correlated for small pairwise angular separations. Such a signal could be correlated with the GWB signal shown in Figure 1.8. We estimated the effect of these uncertainties by using an updated realisation of TT and the most recent solar system ephemeris.

Instabilities in realisations of TT produce a positive cross correlation independent of angular separation. Any estimate of the clock error will thus be correlated with the estimate of the GWB amplitude. Had we made a significant detection of the GWB, this would have to be accounted for. To estimate the importance of possible clock instabilities, we processed our observations using the version of TT released by Bureau International des Poids et Mesures (BIPM) in 2010 (e.g., Petit, 2003). This post-corrected timescale has revealed statistically significant inaccuracies in TT(TAI). The results are shown in Table 6.6. While the change of clock reference only changes our estimated GWB level by nine per cent of the uncertainty, the absolute change (0.8×10^{-30}) is at a significant level for some predictions of the GWB (see Section 5.2.2). This implies that such instabilities in realisations of TT must be accounted for when analysing future data sets.

The results from using the newest solar system ephemeris DE421 (Folkner et al., 2009) are given in Table 6.6. While there have been some improvements in this ephemeris version compared to DE405, most of the changes are absorbed by the pulsar parameter fit. The estimated GWB level has changed by 24% of the uncertainty. If we assume DE421 is correct, then the use of DE405 is similar to introducing a spurious GWB signal with $A = 1.5 \times 10^{-15}$, a signal that is undetectable in most of our time series. However, future observations will need to account for the effects of inaccuracies in the solar system ephemeris.

6.4.4 Contribution of Different Pulsars to the Estimate of A^2

It is difficult to determine the exact contributions to the weighting of each pulsar pair when using error bars derived from Monte Carlo simulations. The dominant effect is the size of T_{overlap}. For a GWB caused by SMBHBs, the weighting factor increases approximately as $T_{\text{overlap}}^{4.3}$. A higher noise level in the residuals of each pulsar in the pair will decrease the weight of that pair approximately linearly. The angle subtended at the observer by the pair of pulsars θ_{ij} can be important if θ_{ij} is near the zeroes of the function plotted in Figure 1.8.

To determine which pulsars contribute the most to our estimate of the GWB, we perform the WLSQ fit described by equations (6.7) and (6.8) to only 189 of the possible 190 $A_{ij}^2 \zeta\left(\theta_{ij}\right)$ estimates. By varying which estimate of $A_{ij}^2 \zeta\left(\theta_{ij}\right)$ is removed, we can find the pulsar pairs that have the greatest influence over the measurement of $\hat{A}^2$ in these residuals. This is performed by finding $\Delta \hat{A}^2$ for each pair of pulsars, which is the measured $\hat{A}^2$ from all pulsar pairs minus the value of $\hat{A}^2$ when *not* including the given pulsar pair. Those pairs with the largest contribution to this measure are given in Table 6.7, and a histogram of the absolute value $\left|\Delta \hat{A}^2\right|$ for all pulsar pairs is provided in Figure 6.7.

This analysis shows that the measurement of $\hat{A}^2$ is determined by only a few pulsar pairs. This severely reduces the number of degrees of freedom when detecting the GWB, and thus decreases the maximum attainable detection confidence (see Jenet et al., 2005) because it reduces our ability to average out the self-noise in the residuals caused by the GWB signal at each pulsar. Observing more strong pulsars is essential to increasing the number of degrees of freedom in order to detect the GWB with reasonable confidence. This is further endorsement of

Table 6.7: The nine pulsar pairs whose absence from the array changes the measurement of $\hat{A}^2$ from our residuals by more than 10^{-30}. The first column contains the names of the pulsars in the pair, the second column lists values of $\Delta\hat{A}^2$, and the third column gives the change as a percentage of the value of $\hat{A}^2$ derived when using all our data.

Removed Pulsar Pair	$\Delta\hat{A}^2$ ($\times 10^{-30}$)	Percentage change
J1713+0747, J1744-1134	18.0	-400%
J2124-3358, J2145-0750	2.32	-52%
J1730-2304, J1744-1134	2.10	-47%
J0711-6830, J2145-0750	1.26	-28%
J0437-4715, J1909-3744	-1.07	24%
J0437-4715, J2129-5721	-1.36	30%
J0437-4715, J2145-0750	-1.41	31%
J1713+0747, J2145-0750	-3.97	88%
J0437-4715, J1713+0747	-7.15	159%

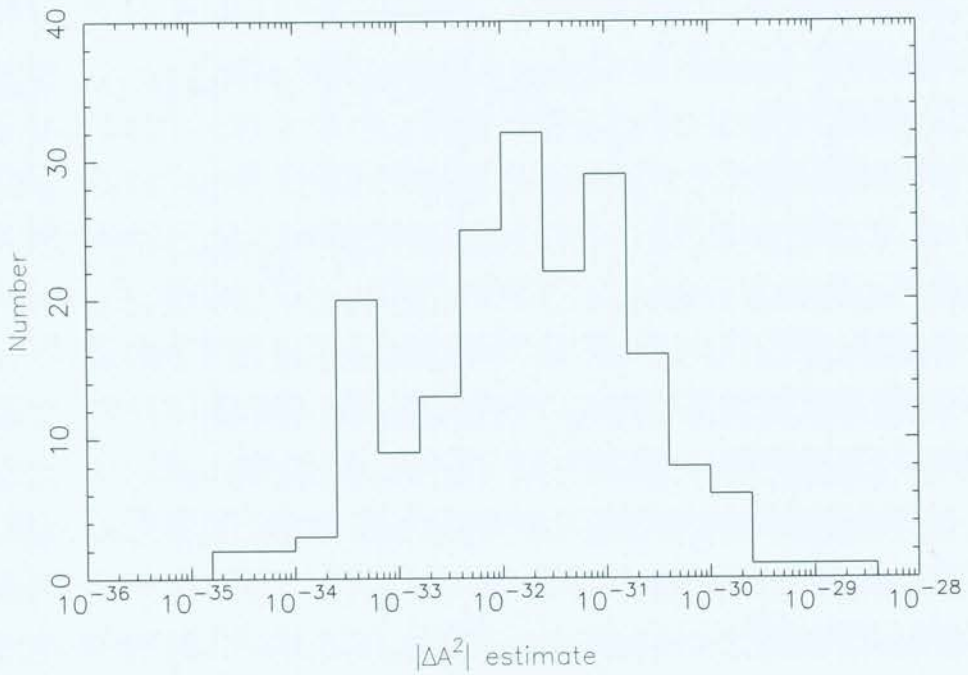

Figure 6.7: The effect on $\hat{A}^2$ of the removal of different pulsar pairs, as measured by $|\Delta\hat{A}^2|$. Almost all pulsar pairs have no significant effect on the value of $\hat{A}^2$ obtained from our residuals.

the International Pulsar Timing Array concept (Section 2.1) and agrees with the conclusions of Jenet et al. (2005), but is contrary to a suggested strategy for detection of individual GW sources (Burt et al., 2011). This is a fundamental difference between the single GW source detection problem and the GWB detection problem.

6.5 Conclusion

In implementing a GWB detection algorithm along the lines originally proposed by Jenet et al. (2005), we have confronted a number of issues that must be addressed when using real observations. We find that in practice the S/N ratio can be reduced by a factor of $\sim$12 compared with the ideal situation discussed by Verbiest et al. (2009) because of the fitting of a timing model to form the residuals. In particular, almost all of the signal loss is caused by the fitting of a quadratic term and arbitrary phase offsets between different observing systems. We also find that it will be important to estimate and correct both clock errors and ephemeris errors when attempting to detect the GWB at a level less than $A = 2 \times 10^{-15}$. As pointed out by Jenet et al. (2005), prewhitening will be required to obtain detection significance larger than 3σ. We find that this cannot be done without solving the problem of spectral leakage due to irregular sampling and variable ToA uncertainties.

Fortunately, there are encouraging indications that many of these problems can be solved. Recent work (Hobbs et al., 2011; Champion et al., 2010) shows that clock errors and ephemeris errors can be estimated and removed. These errors are at a level that would disrupt the GWB signal in pulsar timing observations in the near future, and could even impact the analysis of a modified version of the Verbiest et al. (2008, 2009) observations that did not include arbitrary phase offsets between observing systems. The clock and ephemeris communities will continue to improve their data sets as systems with more sensitivity become available. It appears possible to improve the process of fitting a timing model and also to improve the spectral leakage using the algorithm discussed by Coles et al. (2011). It has proved possible to calibrate most of the phase discontinuities between different observing systems in the PPTA observations and this alone can improve the S/N ratio of the GWB signal by a factor of two.

We have not discussed DM variations, but it is likely that some of the low frequency noise in our residuals is due to such interstellar propagation effects. Certainly as the various PTA data sets improve it will be essential to estimate and remove any frequency-dependent effects.

Our analysis shows that, although the Verbiest et al. (2008, 2009) data set contains observations of 20 pulsars spanning many years, only a few of the pulsars in this data set contribute significantly to detecting the GWB, thereby reducing our detection confidence. It is uncertain whether this will be the case for the most recent observations from the PPTA. Observations of a larger sample of pulsars with precise ToA measurements will help to overcome this problem.

Chapter 7

Conclusion and Future Prospects

Chapter Outline: *In this Chapter, we:*

- *describe the field of GW research using pulsar timing as it was in 2007 when this thesis commenced.*
- *summarise the results of Chapters 3 – 6.*
- *outline future work and a path to a possible detection of GWs with pulsar timing.*

This thesis has addressed the problem of how to study GWs using observations of MSPs. We have focussed on current results using observations from the PPTA project. A detection has not yet been made, but we have constrained the amplitude of single GW sources. In Section 7.1, we highlight the status of GW detection experiments with pulsars before and during this thesis work. In Section 7.2, we discuss the possible future of this exciting project.

7.1 The Past

At the start of 2007, it was already clear that pulsar observations could be used to make a direct detection of GWs (e.g., Sazhin, 1978; Detweiler, 1979). First attempts to create a PTA had been described by Foster & Backer (1990), and Hellings & Downs (1983) had shown that an unambiguous detection of a GWB could be made by measuring correlated timing residuals with a specific angular dependence. Jenet et al. (2005) had shown that a PTA project would need to observe $\geq$20 pulsars over $\gtrsim$5 years to be sensitive to expected GW sources. However, Jenet et al. (2004) had already used pulsar observations to rule out a proposed SMBHB system in the radio galaxy 3C66B with a high degree of confidence.

In 2005, the main data collection for the PPTA project started. A small subset of the data were studied to provide a limit on the amplitude of the GWB (Jenet et al., 2006). This work led to the most constraining limit on the GWB amplitude until early 2011 and was used to rule out some models of cosmic superstrings. However, this method assumed that the timing residuals were statistically white and therefore limited the application of the algorithm to only a few data sets. Jenet et al. (2005) had proposed a method to detect a GWB, but this method could only be applied to timing residuals that had the same regular sampling for all pulsars. The treatment of the effects of pulsar parameter fitting on the algorithms for GWB detection had scarcely been considered (particularly for data sets in which each pulsar was observed for a different time-span).

Over the course of this thesis, we have described techniques that can be applied to almost any set of observations from a PTA. Through the work of Chapters 3 and 4, it is now possible to measure the sensitivity of almost any set of PTA observations to an individual GW-induced sinusoid. This means that a realistic sensitivity curve, which is analogous to the LIGO sensitivity curve, can now be calculated for GW analysis with a PTA. From Chapter 5, we reported that the non-detection of a GWB signal in the near future would provide significant constraints on currently-accepted models of SMBH formation and evolution. However, if a GWB signal can be detected, it is possible to use the technique of Chapter 6 to detect a GWB signal in almost any PTA data set in a straightforward and unambiguous way. Any detection using this technique takes account of the effects of pulsar parameter estimation on irregularly-sampled observations over different time-spans with unequal error bars. This work also shows that the GWB detection statistic currently relies on only a handful of pulsars in the PPTA, whereas a successful detection of the GWB requires the contribution of many pulsars. This is contrary to the optimal observing strategy for detecting single sources of GWs that has been outlined in other recent work (Burt et al., 2011).

7.2 The Future

We currently have not detected any GW signal using the PPTA observations. It is clear that the detection technique should be improved, the current data sets need to become more sensitive and that observations of even more pulsars are required. The following steps are being carried out to achieve this:

First, the IPTA project (Section 2.1) will allow data from all the major PTA projects to be shared. This project is essential for a high-confidence detection of a GW signal in PTA observations. Under the IPTA, observations of many more pulsars will be available. This will significantly increase the probability of a detection of a single GW source and especially the probability of a GWB detection. Aside from the statistical benefits of a larger number of pulsars, it could also prove vital for convincing a wider physics audience that any GW signal that is detected is in fact caused by GWs.

Second, Chapter 6 showed that instabilities in TT(TAI) may obscure the GWB signal. While such an instability can be distinguished from the GWB signal using their different correlated signatures, the noise level in the residuals revealed by the post-corrected realisation TT(BIPM2010) is at the level of most predictions for the GWB signal due to SMBHBs. Algorithms have been implemented to detect and remove this signal (Hobbs et al., 2011), and these algorithms should now be combined with a GWB detection algorithm.

Third, Chapter 6 also showed that errors in the solar system ephemeris may induce a stronger signal in the timing residuals than the GWB signal. While a spectral analysis technique has been applied to measure the mass of known planets in the solar system (Champion et al., 2010), the effect of such errors on the likelihood of GWB detection with pulsars has not been considered in detail. In particular, it is possible that the correlated signal induced in timing residuals by an error in the solar system ephemeris will be related to the correlated signal that is caused by a GWB.

Fourth, while we have not discussed the importance of DM variations in obscuring a GWB signal, it is generally accepted that the DM variations will induce significant low-frequency noise in the timing residuals of many pulsars. It is possible to correct for such variations by comparing observations obtained at widely-separated observing frequencies.

Fifth, if no GW signal is detected in the IPTA data, it is very likely that almost all current predictions for the amplitude of the GWB signal caused by SMBHBs will be ruled out in the next few years. This would have important consequences for current models of galaxy formation and evolution. However, which of the parameters that are used in modelling the GWB signal (i.e., the black-hole merger rate, the merging efficiency, the black-hole mass function; see Section 5.2.2) are ruled out (or constrained) using a given upper bound on the GWB is not yet clear.

Finally, while it is uncertain whether pulsar timing or very precise interferometry will make the first direct detection of GWs, any GW signal that is detected will herald the era of GW astronomy. This opens up an entirely new method of observation, providing simultaneous EM and GW information for some sources while also illuminating previously unobservable regions of the Universe.

References and Bibliography

Abbott B. P., Abbott R., Adhikari R., et al. 2009, Phys. Rev. D, 80, 102001

Acernese F., Amico P., Al-Shourbagy M., et al. 2006, Class. and Quant. Grav., 23, 63

Albrecht H.-E., Damaschke N., Borys M., Tropea C., 2003, *Laser doppler and phase doppler measurement techniques*. Springer-Verlag, Heidelberg

Anholm M., Ballmer S., Creighton J. D. E., Price L. R., Siemens X., 2009, Phys. Rev. D, 79, 084030

Arzoumanian Z., Cordes J. M., Wasserman I., 1999, ApJ, 520, 696

Backer D. C., Kulkarni S. R., Heiles C., Davis M. M., Goss W. M., 1982, Nature, 300, 615

Bailes M., 2003, in Bailes M., Nice D. J., Thorsett S., eds, *Radio Pulsars*, Astronomical Society of the Pacific, San Francisco, p. 57

Bailes M., Harrison P. A., Lorimer D. R., Johnston S., Lyne A. G., Manchester R. N., D'Amico N., Nicastro L., et al. 1994, ApJ, 425, L41

Barriga P., Blair D. G., Coward D., et al. 2010, Class. and Quant. Grav., 27, 084005

Begelman M. C., Volonteri M., Rees M. J., 2006, MNRAS, 370, 289

Bertotti B., Carr B. J., Rees M. J., 1983, MNRAS, 203, 945

Bhattacharya D., van den Heuvel E. P. J., 1991, Phys. Rep., 203, 1

Blanchet L., Iyer B. R., Will C. M., Wiseman A. G., 1996, Class. and Quant. Grav., 13, 575

Blandford R. D., Narayan R., Romani R. W., 1984, J. Astrophys. Astr., 5, 369

Bloom J. S., Holz D. E., Hughes S. A., et al. 2009, submitted to the US Astro2010 Decadal Survey, eprint arxiv:astro-ph/0902.1527

Burt B. J., Lommen A. N., Finn L. S., 2011, ApJ, 730, 17

Caprini C., Durrer R., Siemens X., 2010, Phys. Rev. D, 82, 063511

Carlberg R. G., Cohen J. G., Patton D. R., et al. 2000, ApJ, 532, L1

Champion D. J., Hobbs G. B., Manchester R. N., et al. 2010, ApJL, 720, L201

Chandrasekhar S., 1935, MNRAS, 95, 676

Coles W., Hobbs G., Champion D. J., Manchester R. N., Verbiest J. P. W., 2011, MNRAS, 418, 561

Corbin V., Cornish N. J., 2010, eprint arXiv:astro-ph/1008.1782

Cordes J. M., 2002, in Stanimirovic S., Altschuler D. R., Goldsmith P. F., Salter C. J., eds, *ASP Conf. Ser. 278: Single-Dish Radio Astronomy: Techniques and Applications* Astronomical Society of the Pacific, San Francisco, p. 227

Cordes J. M., Kramer M., Lazio T. J. W., Stappers B. W., Backer D. C., Johnston S., 2004, New Astr., 48, 1413

Cordes J. M., Shannon R. M., 2010, eprint arXiv:astro-ph/1010.3785

Cox D. R., 2006, *Principles of Statistical Inference*. Cambridge University Press, Cambridge

Damour T., Vilenkin A., 2005, Phys. Rev. D, 71, 063510

Davis T. M., Lineweaver C. H., 2004, PASA, 21, 97

Deller A. T., Verbiest J. P. W., Tingay S. J., Bailes M., 2008, ApJ, 685, L67

Demorest P., 2011, BAAS, 43, 124.02

Demorest P., Ramachandran R., Backer D., Ferdman R., Stairs I., Nice D., 2004, BAAS, 36, 149.01

Demorest P. B., 2007, PhD thesis, University of California, Berkeley

Detweiler S., 1979, ApJ, 234, 1100

Dewey R. J., Cordes J. M., 1987, ApJ, 321, 780

Edwards R. T., Bailes M., van Straten W., Britton M. C., 2001, MNRAS, 326, 358

Edwards R. T., Hobbs G. B., Manchester R. N., 2006, MNRAS, 372, 1549

Einstein A., 1916, Annalen der Physik, 354, 769

Einstein A., 1918, Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin), p. 154

Enoki M., Inoue K. T., Nagashima M., Sugiyama N., 2004, ApJ, 615, 19

Ferdman R. D., van Haasteren R., Bassa C. G., et al. 2010, Class. and Quant. Grav., 27, 084014

Folkner W. M., Williams J. G., Boggs D. H., 2009, *The Planetary and Lunar Ephemeris DE421*, IPN Progress Report 42-178 (Pasadena, CA: NASA Jet Propulsion Laboratory)

Foster R. S., Backer D. C., 1990, ApJ, 361, 300

Freire P. C., Camilo F., Lorimer D. R., Lyne A. G., Manchester R. N., D'Amico N., 2001, MNRAS, 326, 901

Gold T., 1968, Nature, 218, 731

Granet C., Zhang H. Z., Forsyth A. R., et al. 2005, IEEE Antennas and Propagation Magazine, 47, 13

Grishchuk L. P., 2005, Physics Uspekhi, 48, 1235

Hankins T. H., Rickett B. J., 1975, in *Methods in Computational Physics Volume 14 — Radio Astronomy* Academic Press, New York, p. 55

Hellings R. W., 1989, in R. W. Hellings ed., *NASA Conference Publication 3046*, p. 93

Hellings R. W., Downs G. S., 1983, ApJL, 265, L39

Hewish A., 1975, Science, 188, 1079

Hewish A., Bell S. J., Pilkington J. D. H., Scott P. F., Collins R. A., 1968, Nature, 217, 709

Hobbs G., 2002, PhD thesis, University of Manchester

Hobbs G., Archibald A., Arzoumanian Z., et al. 2010a, Class. and Quant. Grav., 27, 084013

Hobbs G., Coles W., Manchester R., Chen D., 2011, in Capitaine N., ed., *New Challenges for Reference Systems and Numerical Standards in Astronomy*, Présence Graphique, France, p. 237

Hobbs G., Jenet F., Lee K. J., et al. 2009, MNRAS, 394, 1945

Hobbs G., Lyne A. G., Kramer M., 2010b, MNRAS, 402, 1027

Hobbs G. B., Edwards R. T., Manchester R. N., 2006, MNRAS, 369, 655

Hotan A. W., Bailes M., Ord S. M., 2006, MNRAS, 369, 1502

Hulse R. A., Taylor J. H., 1975, ApJ, 195, L51

Ilyasov Y. P., 2006, Chin. J. Astron. Astrophys. Suppl., 6, 148

Ilyasov Y. P., Imae M., Hanado Y., Oreshko V. V., Potapov V. A., Rodin A. E., Sekido M., 2004a, in Camilo F., Gaensler B., eds, *IAU Symposium 218: Young Neutron Stars and Their Environments*, Astronomical Society of the Pacific, San Francisco, p. 435

Ilyasov Y. P., Oreshko V. V., 2007, in van der Hucht K. A., ed., *Highlights of Astronomy #14, Vol. 2*, p. 489

Ilyasov Y. P., Oreshko V. V., Potapov V. A., Rodin A. E., 2004b, in Camilo F., Gaensler B., eds, *IAU Symposium 218: Young Neutron Stars and Their Environments*, Astronomical Society of the Pacific, San Francisco, p. 433

Jackson J. D., 1962, *Classical Electrodynamics*. John Wiley & Sons, New York

Jacoby B. A., Hotan A., Bailes M., Ord S., Kulkarni S. R., 2005, ApJ, 629, L113

Jaffe A. H., Backer D. C., 2003, ApJ, 583, 616

Jenet F., Finn L. S., Lazio J., et al. 2009, submitted to the US Astro2010 Decadal Survey, eprint arXiv:astro-ph/0909.1058

Jenet F. A., Hobbs G. B., Lee K. J., Manchester R. N., 2005, ApJ, 625, L123

Jenet F. A., Hobbs G. B., van Straten W., et al. 2006, ApJ, 653, 1571

Jenet F. A., Lommen A., Larson S. L., Wen L., 2004, ApJ, 606, 799

Kartaltepe J. S., Sanders D. B., Scoville N. Z., et al. 2007, ApJS, 172, 320

Kaspi V. M., Taylor J. H., Ryba M., 1994, ApJ, 428, 713

Komatsu E., Dunkley J., Nolta M. R., et al. 2009, ApJS, 180, 330

Kopeikin S. M., 1997, Phys. Rev. D, 56, 4455

Kopeikin S. M., 1999, MNRAS, 305, 563

Koushiappas S. M., Bullock J. S., Dekel A., 2004, MNRAS, 354, 292

Kramer M., Wex N., 2009, Class. and Quant. Grav., 26, 073001

Larson S. L., Hiscock W. A., Hellings R. W., 2000, Phys. Rev. D, 62, 062001

Lee K. J., Jenet F. A., Price R. H., 2008, ApJ, 685, 1304

Lee K. J., Wex N., Kramer M., Stappers B. W., Bassa C. G., Janssen G. H., Karuppusamy R., Smits R., 2011, MNRAS, 414, 3251

Lin L., Koo D. C., Willmer C. N. A., et al. 2004, ApJL, 617, L9

Lin L., Patton D. R., Koo D. C., et al. 2008, ApJ, 681, 232

Lomb N. R., 1976, Astrophys. Space Sci., 39, 447

Lommen A. N., Backer D. C., 2001, ApJ, 562, 297

Lorimer D. R., 2005, Living Reviews in Relativity, 8, 7

Lorimer D. R., Kramer M., 2005, *Handbook of Pulsar Astronomy*. Cambridge University Press, Cambridge

Lyne A. G., 2004, in Camilo F., Gaensler B., eds, *IAU Symposium 218: Young Neutron Stars and Their Environments*, Astronomical Society of the Pacific, San Francisco, p. 257

Lyne A. G., Manchester R. N., Lorimer D. R., et al. 1998, MNRAS, 295, 743

Lyne A. G., Rickett B. J., 1968, Nature, 218, 326

Lyne A. G., Smith F. G., 2005, *Pulsar Astronomy*, 3rd edn. Cambridge University Press, Cambridge

Maggiore M., 2000, Phys. Rep., 331, 283

Manchester R. N., 2008, in C. G. Bassa Z. Wang A. C., Kaspi V. M., eds, *40 Years of Pulsars: Millisecond Pulsars, Magnetars and More*, Vol. 983. American Institute of Physics, Melville, New York, p. 584

Manchester R. N., 2010, in Damour T., Jantzen R. T., Ruffini R., eds, *Proceedings of the Twelfth Marcel Grossman Meeting on General Relativity* World Scientific, Singapore, in press

Manchester R. N., 2011, in Burgay M., D'Amico N., Esposito P., Pellizzoni A., Possenti A., eds, *Radio Pulsars: An Astrophysical Key to Unlock the Secrets of the Universe* Vol. 1357 of AIP Conference Series. American Institute of Physics, Melville, New York, p. 65

Manchester R. N., Hobbs G. B., Teoh A., Hobbs M., 2005, AJ, 129, 1993

Manchester R. N., Taylor J. H., 1977, *Pulsars*. Freeman, San Francisco

Mei S., Blakeslee J. P., Côté P., et al. 2007, ApJ, 655, 144

Melrose D., 2004, in Camilo F., Gaensler B., eds, *IAU Symposium 218: Young Neutron Stars and Their Environments*, Astronomical Society of the Pacific, San Francisco, p. 349

Mood A. M., Graybill F. A., Boes D. C., 1974, *Introduction to the Theory of Statistics*, 3rd edn. McGraw-Hill College, New York

Nan R., 2008, in Stepp L., Gilmozzi R., eds, *Ground-based and Airborne space Telescopes II* Vol. 7012 of SPIE Conference Series. p. 70121E

Nan R., 2009, BAAS, 41, 226.03

Navarro J., 1994, PhD thesis, California Inst. of Tech.

Nice D. J., Demorest P. B., Gonzalez M. E., Ferdman R. D., Ransom S. M., Stairs I. H., NANOGrav 2011, BAAS, 43, 139.06

Ölmez S., Mandic V., Siemens X., 2010, Phys. Rev. D, 81, 104028

Oslowski S., van Straten W., Hobbs G., Bailes M., Demorest P., 2011, MNRAS, in press, eprint arXiv:astro-ph/1108.0812

Pacini F., 1967, Nature, 216, 567

Patton D. R., Pritchet C. J., Carlberg R. G., et al. 2002, ApJ, 565, 208

Peters P. C., 1964, Phys. Rev., 136, 1224

Peters P. C., Mathews J., 1963, Phys. Rev., 131, 435

Petit G., 2003, in Time Service Department, ed., *35th Annual Precise Time and Time Interval Systems and Applications Meeting 2003*, Curran Associates, Inc. (published in 2008), p. 307

Phinney E. S., 2001, eprint arXiv:astro-ph/0108028

Pitkin M., Clark J., Hendry M. A., Heng I. S., Messenger C., Toher J., Woan G., 2008, J. Phys.: Conf. Ser., 122, 012004

Pohl R., Antognini A., Nez F., et al. 2010, Nature, 466, 213

Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 1992, *Numerical recipes in C: The art of scientific computing*, 2nd edn. Cambridge University Press, Cambridge

Pshirkov M. S., 2009, MNRAS, 398, 1932

Pshirkov M. S., Baskaran D., Postnov K. A., 2010, MNRAS, 402, 417

Radhakrishnan V., Shukre C. S., 1985, in Srinivasan G., Radhakrishnan V., eds, *Supernovae, Their Progenitors and Remnants*, Indian Academy of Sciences, Bangalore, p. 155

Rajagopal M., Romani R. W., 1995, ApJ, 446, 543

Ransom S. M., Demorest P., Ford J., McCullough R., Ray J., DuPlain R., Brandt P., 2009, BAAS, 214, 605.08

Rickett B. J., 1990, Ann. Rev. Astr. Ap., 28, 561

Romani R. W., Taylor J. H., 1983, ApJ, 265, L35

Ruderman M. A., Sutherland P. G., 1975, ApJ, 196, 51

Sathyaprakash B. S., Schutz B. F., 2009, Living Reviews in Relativity, 12, 2

Sazhin M. V., 1978, Sov. Astron., 22, 36

Scargle J. D., 1982, ApJ, 263, 835

Sesana A., Vecchio A., 2010a, Class. and Quant. Grav., 27, 084016

Sesana A., Vecchio A., 2010b, Phys. Rev. D, 81, 104008

Sesana A., Vecchio A., Colacino C. N., 2008, MNRAS, 390, 192

Sesana A., Vecchio A., Volonteri M., 2009, MNRAS, 394, 2255

Shannon R. M., Cordes J. M., 2010, ApJ, 725, 1607

Shapiro S. L., Teukolsky S. A., 1983, *Black Holes, White Dwarfs and Neutron Stars. The Physics of Compact Objects*. Wiley–Interscience, New York

Shawhan P. S., LIGO Scientific Collaboration 2003, Nucl. Instr. and Methods in Phys. Res. A, 502, 396

Sillanpää A., Takalo L. O., Pursimo T., et al. 1996, A&A, 305, L17

Smith J. R., LIGO Scientific Collaboration 2009, Class. and Quant. Grav., 26, 114013

Smits R., Lorimer D. R., Kramer M., Manchester R., Stappers B., Jin C. J., Nan R. D., Li D., 2009, A&A, 505, 919

Smits R., Tingay S. J., Wex N., Kramer M., Stappers B., 2011, A&A, 528, 108

Standish E. M., 2004, A&A, 417, 1165

Stappers B. W., Kramer M., Lyne A. G., D'Amico N., Jessner A., 2006, Chin. J. Astron. Astrophys. Suppl. 2, 6, 298

Stinebring D. R., Ryba M. F., Taylor J. H., Romani R. W., 1990, Phys. Rev. Lett., 65, 285

Sudou H., Iguchi S., Murata Y., Taniguchi Y., 2003, Science, 300, 1263

Taylor J. H., 1992, Phil. Trans. Roy. Soc. A, 341, 117

Taylor J. H., Cordes J. M., 1993, ApJ, 411, 674

Taylor J. H., Weisberg J. M., 1982, ApJ, 253, 908

Thorne K. S., 1987, in Hawking S., Israel W., eds, *300 Years of Gravitation*, Cambridge University Press, Cambridge, p. 330

Thorsett S. E., Dewey R. J., 1996, Phys. Rev. D, 53, 3468

Tofani G., Alvito G., Ambrosini R., et al. 2008, in Stepp L., Gilmozzi R., eds, *Ground-based and Airborne space Telescopes II* Vol. 7012 of SPIE Conference Series. p. 70120F

Usov V. V., Melrose D. B., 1995, Aust. J. Phys., 48, 571

Valtonen M. J., Nilsson K., Villforth C., et al. 2009, ApJ, 698, 781

van Haasteren R., Levin Y., 2010, MNRAS, 401, 2372

van Haasteren R., Levin Y., Janssen G. H., et al. 2011, MNRAS, 414, 3117

van Haasteren R., Levin Y., McDonald P., Lu T., 2009, MNRAS, 395, 1005

van Straten W., 2003, PhD thesis, Swinburne University of Technology

van Straten W., 2006, ApJ, 642, 1004

van Straten W., Bailes M., Britton M., Kulkarni S. R., Anderson S. B., Manchester R. N., Sarkissian J., 2001, Nature, 412, 158

Verbiest J. P. W., Bailes M., Bhat N. D. R., et al. 2010, Class. and Quant. Grav., 27, 084015

Verbiest J. P. W., Bailes M., Coles W. A., et al. 2009, MNRAS, 400, 951

Verbiest J. P. W., Bailes M., van Straten W., et al. 2008, ApJ, 679, 675

Volonteri M., Haardt F., Madau P., 2003, ApJ, 582, 559

Volonteri M., Salvaterra R., Haardt F., 2006, MNRAS, 373, 121

Weisberg J. M., Nice D. J., Taylor J. H., 2010, ApJ, 722, 1030

Wen Z. L., Jenet F. A., Yardley D., Hobbs G. B., Manchester R. N., 2011, ApJ, 730, 29

Wen Z. L., Liu F. S., Han J. L., 2009, ApJ, 692, 511

White S. D. M., Rees M. J., 1978, MNRAS, 183, 341

Will C. M., 1977, ApJ, 214, 826

Wyithe J. S. B., Loeb A., 2003, ApJ, 590, 691

Yan W. M., Manchester R. N., van Straten W., et al. 2011, MNRAS, 414, 2087

Yardley D. R. B., Coles W. A., Hobbs G. B., et al. 2011a, MNRAS, 414, 1777

Yardley D. R. B., Coles W. A., Hobbs G. B., Manchester R. N., 2011b, in Burgay M., D'Amico N., Esposito P., Pellizzoni A., Possenti A., eds, *Radio Pulsars: An Astrophysical Key to Unlock the Secrets of the Universe*, Vol. 1357 of AIP Conference Series. American Institute of Physics, Melville, New York, p. 77

Yardley D. R. B., Hobbs G. B., Jenet F. A., et al. 2010, MNRAS, 407, 669

You X. P., Hobbs G., Coles W. A., et al. 2007, MNRAS, 378, 493

Zechmeister M., Kürster M., 2009, A&A, 496, 577

Appendix A

Our Technique for Detection of a Gravitational-Wave-Induced Sinusoid in Actual Pulsar Timing Observations

This Appendix contains supplementary material relevant to Chapter 4.

In the following sections, we give full details of the implementation of the algorithms described in Section 4.2. In particular, we describe some of the problems that arose during the analysis. Solutions to these problems are described below, while their implementation as a TEMPO2 plugin is given in Appendix B.

A.1 Our Technique for Producing a Sensitivity Curve

Our method for creating curves showing the sensitivity of our timing residuals to GW-induced sinusoidal signals from individual SMBHBs takes into account non-white noise. To produce a sensitivity curve for a given set of pulsars and their timing residuals, we use a three-step process as follows:

1. We choose logarithmically-spaced GW frequencies between $\frac{1}{T_{\rm obs}}$ and $\frac{N_{\rm pts}}{2T_{\rm obs}}$ (single pulsar) or between $(30\,{\rm yr})^{-1}$ and $(28\,{\rm d})^{-1}$ (multiple pulsars). The frequency sampling we used for multiple pulsars requires oversampling each periodogram by a factor $30\,{\rm yr}/T_{\rm obs}$ for that pulsar.

2. At each frequency, we:

(a) add the effect of a sinusoidal GW point source with angular frequency $2\pi f_i$, amplitude h_s and random sky-position and polarisation to the ToAs, as described by Equation (3.6).

(b) process the data using TEMPO2 to obtain post-fit timing residuals.

(c) run a detection algorithm (described below) on the post-fit residuals that reports either a detection or a non-detection.

(d) repeat steps 2a – 2c a large number of times (we use 10^3 iterations) and record the detection percentage.

(e) If we have detected (95 ± 1) % of the signals then we have satisfied our detection criterion and we record f_i and h_s, which places a point on the pulsar timing sensitivity curve. If the detection criterion is not satisfied, adjust h_s higher if too few detections have been made and lower if too many, then return to step 2a.

3. Select the next frequency in the grid and repeat.

Our detector functions as follows:

1. For each pulsar in the input data set, we calculate a non-normalised Lomb-Scargle periodogram of the residuals with the frequency range described above.

2. We smooth the periodogram by taking the logarithm of the power values and using a boxcar median filter. By default, the number of points in the filter is 11 times the oversampling factor for that pulsar. This accounts for the correlated spectral estimates induced by the oversampling of the periodogram and by the irregular spacing of the timing residuals.

3. We use a least-squares fit to the median-smoothed log-periodogram to obtain a low-order polynomial (i.e., of order less than six) that provides a simple model of the power spectrum (see Section A.4). The median-smoothing and model-fitting are performed only on those points in the periodogram with frequency $\geq (T_{\rm obs})^{-1}$. This three-step spectral modelling process ensures that the simulated GW source is not included in the model as part of the noise in the spectrum. This is particularly important at the low- and high-frequency edges of the periodogram. When analysing the data collected from multiple pulsars we combine their periodograms using a weighted sum. The weight used for each pulsar is the inverse of the simple frequency-dependent model of the power spectrum for that pulsar.

4. We multiply the noise model obtained above by a factor of $\sim 2-3$ (determined from simulations; see Section A.3) to define a set of detection thresholds for any given false alarm probability (we use $P_f = 1\%$). These detection thresholds are set such that there is a 1% probability of any observed power across the whole periodogram being greater than the threshold when there is no signal present.

5. If the measured power in the channel containing the input GW frequency is greater than the detection threshold in that channel, then we have made a detection of a significant sinusoid.

In place of step 1 of the GW detection algorithm described above, Lommen & Backer (2001) used a floating-mean periodogram. Such a periodogram allows for the sinusoid fitted by the Lomb-Scargle algorithm to have non-zero mean. This can be important when the observations are sparsely sampled. We have not used a floating-mean periodogram because we expect the improvement to be relatively small for our well-sampled observations. Furthermore, the detection algorithm we present in this Appendix is a simple implementation that we acknowledge is not optimal.

Some of the simulated sinusoidal GW point sources produce large signals in the timing residuals, depending on their amplitude, polarisation and location on the sky. If a set of timing residuals showed evidence of a strong signal, a typical analysis would use a model of the pulsar with the fewest possible parameters (i.e. a period, period-derivative and any arbitrary phase offsets) to obtain residuals. This allows the observations to be examined more closely to determine the source of signal. To simulate this process, in step 2b above we calculate the full parameter fit as normal and measure the reduced χ^2 for the fit. If the reduced χ^2 is larger than 20, then we instead only fit for the pulsar period and spin-down, and for arbitrary phase jumps between different backend systems.

The weighted fit for the pulsar parameters sometimes increases the power calculated at certain frequencies by the periodogram. This is because our periodogram technique does not account for the uncertainty in each ToA estimate. For example, extra power may be induced at a period of six months because the weighted pulsar parameter fit gives an updated value of the parallax that increases the unweighted power at this frequency. This is not surprising, but it can lead to false detections. This was accounted for in the modelling of the power spectrum - conservative models were used in general. An optimal treatment would require a more-complicated weighted power spectral estimate. A weighted spectral estimate would also increase the leakage in the periodogram because of the highly variable ToA uncertainties.

A.2 Our Technique for Producing an Upper Limit or a Limit Matrix

As described in Section 4.2, we have developed a technique for ruling out GWs with a particular strain amplitude as a function of frequency. The important assumption in producing an upper bound is that, at any frequency in our periodogram, the power caused by GWs cannot be more than the observed power. If it were, we would have observed a higher power level at that frequency. This means that we assume that all the power at a given frequency is caused by an individual non-evolving source of GWs. We then calculate the GW strain that gives a power greater than this level in 95% of simulations. This value of the GW strain becomes the upper bound.

To produce this upper bound, we first calculate the periodogram of the observed timing residuals of each pulsar. We make a simple polynomial model of the noise in this periodogram and use the inverse of this noise model as the weight in calculating a weighted and summed periodogram. This weighted and summed periodogram is the "limit threshold" in this case. The limit threshold roughly represents the weighted average noise level in the residuals.

We then simulate the ToAs induced by a non-evolving SMBHB using Equation (3.6). These induced ToAs are consistent with a noiseless sinusoid. We apply the TEMPO2 parameter fit directly to these ToAs to calculate the residuals induced by this SMBHB in each pulsar. We calculate the weighted and summed periodogram of the induced residuals using the same noise model for each pulsar that is used for the actual observations. We compare this weighted sum of noiseless sinusoids to the limit threshold. We then scale the strain amplitude so that the induced signal produces more power than the limit threshold in 95% of simulations. We can thus rule out the existence of any stronger GW sources at this frequency (with random sky position and polarisation) with 95% confidence.

Unlike in the detection case, we cannot use the reduced χ^2 to inform us of the quality of the pulsar parameter fit. This is because we are fitting pulsar parameters to a noiseless sinusoid, so the reduced χ^2 is meaningless for these parameter fits. However, the amplitude of each simulated GW signal when producing a limit is considerably smaller than that required to make a detection because the limit threshold is lower than the detection threshold. An upper bound on the amplitude of a sinusoid present in timing residuals will always be lower than the lowest detectable amplitude, because detection must account for the noise in the detector.

A.3 The False Alarm Probability

We have used simulations to calculate the detection threshold for a false alarm probability of 1% across the whole weighted and summed periodogram of a given data set. The statistics of each channel in the periodogram approximately follow a χ^2-distribution with $2N_{\rm psr}$ degrees of freedom, but many other effects change the statistics of each channel, as described below.

After adding a large GW signal to our ToAs that induces a sinusoid in frequency channel i, the statistics of channel i follow a non-central χ^2-distribution, i.e., a Ricean distribution. This does not affect the false-alarm probablility determination but would affect analytic determinations of pulsar timing sensitivity. Other effects include:

- the irregular sampling of the time series (which can cause correlated estimates of the power in different channels);
- the oversampling of the periodogram when analysing multiple pulsars (which means that the peaks in the periodogram will be more fully resolved and thus the peak value is higher); and
- the median filtering (which lowers the height of each peak in the periodogram as well as raising the troughs).

Our method for calculating the detection threshold is similar to the method described in Section 3.2.2. We calculate $T_{1,i}$, which is the detection threshold in an individual frequency channel that gives a 1% false alarm probability, by assuming that the power follows a χ^2-distribution with $2N_{\rm psr}$ degrees of freedom. For this analysis, $T_{1,i}$ is a factor of 2 lower than the level implied by a χ^2-distribution with $2N_{\rm psr}$ degrees of freedom because the mean of such a distribution is $2N_{\rm psr}$, whereas the mean of the weighted and summed periodogram is $N_{\rm psr}$. We then choose $\beta' > 1$ (generally in the range $1.3 < \beta' < 2.5$) and calculate $T_i' = \beta' T_{1,i}$. T_i' forms a first estimate of the detection threshold corresponding to a particular false alarm probability across all frequencies in the periodogram.

We then simulate $\sim 10^3$ realisations of white noise with a rms residual of 100 ns and the same sampling as the original time series. We do not perform a TEMPO2 parameter fit, nor add any SMBHB signals to the data. We calculate a weighted and summed periodogram for each realisation and compare it to T_i'. We find the number of simulated data sets yielding a detection at any frequency. If this number is more than 1% of the total number of simulated data sets, then we increase β' and repeat the process. The factor β' is adjusted until we find the correct factor $\beta' = \beta$ such that the detection rate above the correct detection threshold $T_i = \beta T_{1,i}$ equals the false alarm probability. Note that the process of calculating β described here is equivalent to manually calculating β in Section 3.2.2.

A.4 Modelling the Power Spectrum

Some typical spectral models used in our analysis are shown in Figure 4.1. The three pulsars shown in this figure are the same three whose individual sensitivity is displayed in Figure 4.2. In general, the models chosen are conservative in the presence of red noise to minimise the number of spurious detections at low frequencies.

The spectral models in Figure 4.1 exhibit some typical features from our analysis. In particular, the models account for the confusion between red noise in the timing residuals and signal leakage caused by irregular sampling. Many of the PPTA pulsars (including PSRs J0437−4715 and J1713+0747) exhibit high power levels at the lowest frequencies. This requires the inclusion of more terms in the chosen polynomial model; for example, PSRs J1713+0747 and J0437−4715 are both modelled with quartic polynomials. On the other hand, the timing residuals of PSR J1857+0943 exhibit a flat periodogram at all frequencies before the addition of simulated GW signals. However, if the actual residuals had been affected by a low-frequency GW source, we would be unable to distinguish between leakage from the GW signal and low-frequency noise. As shown in Figure 4.1, it is conceivable that these timing residuals are affected by low-frequency noise in the channels adjacent to the signal. Hence, we model its periodogram with a cubic polynomial to take account of the fact that we cannot distinguish between a low-frequency GW source affecting the ToAs and red noise affecting the ToAs.

When limiting the amplitude of the individual non-evolving GW sources that could be affecting our observations, we do not add sinusoids to the measured timing residuals. Hence, a different model for the power spectrum may be used from those shown here, because the features in the periodogram are different.

Appendix B

Computer Programs Implemented

This Appendix contains the source code for three of the TEMPO2 plugins that I developed during my PhD. This code is included in my thesis in case the original source code is lost or deleted, and as a reference for some details described in the thesis. While the code has been thoroughly tested for functionality, it has been only slightly edited to aid readability.

All three codes are freely available online at:
`http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Main.Plugins`

B.1 The XFER_FUNC4_PLUG.C plugin

This plugin is described in Section 2.4.3 of this thesis. It has been slightly edited for its appearance from the original source code.

```
/*****************************************************************************/
/* a plugin to determine the transfer function of tempo2 as it acts on a particular data set.
   You need to rerun this code everytime you change which parameters are being fit for OR
   have a new data set if you want to correct for the effects of tempo2. This code also uses
   smoothed and interpolated white noise as the "prefit" spectrum and performs the same
   smoothing and interpolation on the post-fit residuals to calculate the post-fit spectrum.
   This does not affect the transfer function at low frequencies (which is what we care about
   , given the weighting function used in the GWB detection statistic) and appears to have
   negligible effect on the transfer function at high frequencies.
*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKspectrum.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include "GWsim.h"

using namespace std;
```

```
void makeIdealSats(pulsar *psr,int npsr,char parFile[MAX_PSR_VAL][MAX_FILELEN],char timFile[
    MAX_PSR_VAL][MAX_FILELEN]);
void TK_weightLS2(double *x,double *y,double *sig,int n,double *outX,double *outY,int *outN,
    double *outY_re, double *outY_im, int useWeight);
void interpolateSplineSmooth(double *inX, double *inY, int inN, double separation, double *
    interpX, double *interpY, int *nInterp);
void unique(double *in, int nIn, double *out, int *nOut);
double TKfindWeightedRMS_d(double *x,double *wt,int n);
double TKWeightedmean_d(double *x,double *wt,int n);
double TKfindWeightedVariance_d(double *x,double *wt,int n);
void interpolateSplineSmoothFixedPhase(double *inX, double *inY, int inN, double separation,
    double *interpX, double *interpY, int *nInterp, double fixedStart);

/* The main function called from the TEMPO2 package is 'graphicalInterface' */
/* Therefore this function is required in all plugins                        */
extern "C" int graphicalInterface(int argc,char *argv[],pulsar *psr,int *npsr)
{
  printf("\n\n!!! NB your tim file must be sorted because this program uses TKspectrum!!!
      Which sorts your data!!!\n\n\n\n");
  char parFile[MAX_PSR_VAL][MAX_FILELEN];
  char timFile[MAX_PSR_VAL][MAX_FILELEN];
  int i,k,j,p,it,nit;
  nit = 1000;   //default value
  int specType = 2; //default is Lomb Periodogram
  double hifac = 3.0; //default is to go to 3 times higher frequency than the average - this
      should hopefully cover all possible overlapping data spans.
  int smooth = 0; //default is no smoothing by a 60-day width exponential smoother.
  int interp = 1; //default is to do the interpolation.
  int useWeight = 0; //default is to just do a LSQ fit for the spectral estimates with
      specType = 4.
  double phase = 13.0;   //the phase offset in the "fixedPhase variable" - defines where in
      the fortnight we take each sample after interpolation.
  double globalParameter;
  *npsr = 0;  /* For a graphical interface that only shows results for one pulsar */
  printf("Graphical Interface: xfer_func\n");
  printf("Author:              DY\n");
  printf("Version:             1\n");
  /* Obtain all parameters from the command line */
  for (i=2;i<argc;i++)
    {
      if (strcmp(argv[i],"-f")==0)
        {
          strcpy(parFile[*npsr],argv[i+1]);
          strcpy(timFile[*npsr],argv[i+2]);
          (*npsr)++;
          printf("*npsr = %d ",*npsr);
        }
      else if (strcmp(argv[i],"-nit")==0)
        sscanf(argv[++i], "%d", &nit);
      else if (strcmp(argv[i],"-phase")==0)
        sscanf(argv[++i], "%lf", &phase);
      else if (strcmp(argv[i],"-specType")==0)   //2 => Lomb Scarlge, 4 => Weighted Lomb
          Scargle, 1==> DFT
        sscanf(argv[++i], "%d", &specType);
      else if (strcmp(argv[i],"-smooth")==0)   //Do a 60-day smooth and re-interpolate the
          data onto a daily grid.
        smooth = 1;
      else if (strcmp(argv[i],"-noInterp")==0)   //Don't do the interpolation onto the regular
          grid
      interp = 0;
    else if (strcmp(argv[i],"-useWeight")==0)    //=0 to do an unweighted LSQ fit of sin and
        cos to determine spectrum, = 1 to do a weighted LSQ fit.
      useWeight = 1;
    else if (strcmp(argv[i],"-h")==0||strcmp(argv[i],"—help")==0)
      {
        printf("\n TEMPO2 xfer_func plugin: determines the transfer function of tempo2 as it
             acts on a particular data set. That is, what effect does tempo2 have on the
            power spectrum of any input data set automatically, e.g. the dip in the power
            spectrum at 1/1year, the dip at the lowest few frequencies caused by quadratic
            fitting, etc., and what are the error bars on the spectral estimates?\n");
        printf("==================\n");
        printf("\nUSAGE: \n\t tempo2 -gr xfer_func -f par1.par tim1.tim -f par2.par tim2.tim
             ...(as many as desired) [options]\n");
        printf("\n Command line options:\n");
        printf("-h or —help:\t display help and exit\n");
        printf("-nit:\t number of iterations to do in determining the transfer function\n");
        printf("-specType:\t determines which kind of periodogram to do —> 1 gives DFT, 2
            gives Lomb Scargle, 4 gives Weighted Lomb Scargle\n");
        printf("-smooth:\t turns on smoothing and interpolating the input data set.\n");
        printf("-useWeight:\t uses weights to calculate the LSQ fit of sins and cosines to
            determine the power spectrum\n");
        printf("-noInterp:\t turns off the interpolation step when used with '-smooth' \n")
            ;
        exit(0);
      }
  }

printf("\n");

if (specType == 1)
  {
    printf("DFT selected, you MUST have regular sampling to use it!!!!\n");
  }
else if (specType != 1 && specType != 2 && specType != 4)
  {
    printf("Unknown spectral analysis type\n");
    exit(1);
  }
readParfile(psr,parFile,timFile,*npsr); /* Load the parameters        */
readTimfile(psr,timFile,*npsr);          /* Load the arrival times     */
preProcess(psr,*npsr,argc,argv);
for (i=0;i<2;i++)                     /* Do two iterations for pre- and post-fit residuals */
  {
    formBatsAll(psr,*npsr);           /* Form the barycentric arrival times */
    formResiduals(psr,*npsr,1);      /* Form the residuals                  */
    if (i==0) doFit(psr,*npsr,0);    /* Do the fitting      */
  }
//long seed = TKsetSeed();
long seed = -1540;  printf("\n\n\n\nHARD CODING SEED \t\t\t\tWARNING!!!\t\t\n\n\n\n");
long double **sat0, toffset;
double **checkResY;
double **resY, **resX, **resE;

FILE *fout, *foutPre, *foutPost;
int badFitFlagWeighted = 0;   //a counter which measures how often the weighted postfit rms
    is greater than the weighted prefit rms.
int badFitFlagNotWeighted = 0;  //a counter which measures how often the unweighted postfit
    rms is greater than the unweighted prefit rms.
```

```
int badFitFlag = 0; //a flag to tell us if there was a bad fit.
double prefitRMS, postfitRMS, postfitVAR, postfitWeightedVAR; //the unweighted pre- and post-fit rmses and the unweighted postfit variance.

char fname[100];
double tspan[*npsr], fnyq[*npsr]; //fnyq is the "average" Nyquist frequency, as used in the TKperiod program, which is the Nyquist frequency one obtains if the points are evenly spaced over time.
double maxTspan;
double minTspan;
long double minsat[*npsr], maxsat[*npsr], avgTspan=0.0;
double ***white; //a 3D array that contains all the white noise realisations I'll use.

//Allocte Memory
sat0 = (long double **)malloc(MAX_PSR*sizeof(long double *));
checkResY = (double **)malloc(MAX_PSR*sizeof(double *));
resY = (double **)malloc(MAX_PSR*sizeof(double *));
resX = (double **)malloc(MAX_PSR*sizeof(double *));
resE = (double **)malloc(MAX_PSR*sizeof(double *));
white = (double ***)malloc(MAX_PSR*sizeof(double **));
for (i=0;i<MAX_PSR;i++)
  {
    checkResY[i] = (double *)malloc(MAX_OBSN*sizeof(double));
    resY[i] = (double *)malloc(MAX_OBSN*sizeof(double));
    resX[i] = (double *)malloc(MAX_OBSN*sizeof(double));
    resE[i] = (double *)malloc(MAX_OBSN*sizeof(double));
    sat0[i] = (long double *)malloc(MAX_OBSN*sizeof(long double));
    white[i] = (double **)malloc(nit*sizeof(double *));
    for (j=0;j<nit;j++)
      white[i][j] = (double *)malloc(MAX_OBSN*sizeof(double));
  }

//START PLUGIN
//following is essentially for splitting up the 1857 data set, though it could be generally applicable.
double maxallowablegap = 2000.0; //in units of days. THIS IS AN ARBITRARY CHOICE, simply so that we know 1857 gets split up.
long double maxgap[*npsr];
int locmaxgap[*npsr]; //the location of the biggest gap in the time series.
//Calculate timespans
toffset = psr[0].param[param_pepoch].val[0];
for (p=0;p<*npsr;p++)
  {
    minsat[p]=maxsat[p]=psr[p].obsn[0].sat;
    maxgap[p] = 0.0L;
    for (j=0;j<psr[p].nobs;j++)
      {
        if (psr[p].obsn[j].sat < minsat[p]) minsat[p] = psr[p].obsn[j].sat;
        if (psr[p].obsn[j].sat > maxsat[p]) maxsat[p] = psr[p].obsn[j].sat;
        if (psr[p].obsn[j+1].sat - psr[p].obsn[j].sat > maxgap[p] && j < psr[p].nobs-1)
          {
            maxgap[p] = psr[p].obsn[j+1].sat - psr[p].obsn[j].sat;
            locmaxgap[p] = j; //so the biggest gap appears between the j-th and j+1-th observations.
          }
      }
    tspan[p] = (double) (maxsat[p] - minsat[p]);
    fnyq[p] = psr[p].nobs / 2.0 / (tspan[p] * 86400.0); //and now fnyq is the average Nyquist frequency for this pulsar in sec^-1
    avgTspan += (long double) tspan[p];
    printf("maxgap[p] = %Lg\n",maxgap[p]);
  }

avgTspan = avgTspan / (long double) *npsr;
maxTspan = TKfindMax_d(tspan,*npsr);
minTspan = TKfindMin_d(tspan,*npsr);
printf("max time span present in data is %lg\n",maxTspan);
printf("min time span present in data is %lg\n",minTspan);
printf("average time span present in data is %Lg\n\n",avgTspan);

// Store residuals
int nres[*npsr];
for (p=0;p<*npsr;p++)
  {
    nres[p] = psr[p].nobs;
    for (i=0;i<psr[p].nobs;i++)
      {
        if (psr[p].obsn[i].deleted!=0)
          {
            printf("Must remove deleted points from the .tim file for psr %s\n",psr[p].name)
                ;
            exit(1);
          }
        resY[p][i] = (double)psr[p].obsn[i].residual;     //for 1857, this is MORE obsns than
             we need, but will have to do truncating later on in the code.
        resX[p][i] = (double)(psr[p].obsn[i].sat - toffset + 1000.0L);
        resE[p][i] = (double)(psr[p].obsn[i].toaErr*1.0e-6);
      }
  }
printf("\n");

makeIdealSats(psr,*npsr,parFile,timFile);  // Determine the idealised site arrival times

//Store ideal sats in sat0[][], TESTED that ideal sats really are ideal (rms of resid's is
    0)
for (p=0;p<*npsr;p++)
  {
    for (i=0;i<psr[p].nobs;i++)
      sat0[p][i] = psr[p].obsn[i].sat;      //note these are the ideal site arrival times
  }

//power spectral estimation parameters and allocate memory
int MAX_SPEC = 2000 * (int)hifac;
double *specXPre,*specXPost,*specY, *avgPreSpecY, *avgPostSpecY,*specY_re, *specY_im;
double **allPrefitSpectra; //to store each iteration, thus enabling calculation of a
    statistical error bar
double **allPostfitSpectra; //same as above.
specXPre = (double *)malloc(MAX_SPEC*sizeof(double));
specXPost = (double *)malloc(MAX_SPEC*sizeof(double));
specY = (double *)malloc(MAX_SPEC*sizeof(double));
specY_re = (double *)malloc(MAX_SPEC*sizeof(double));
specY_im = (double *)malloc(MAX_SPEC*sizeof(double));
avgPreSpecY = (double *)malloc(MAX_SPEC*sizeof(double));
avgPostSpecY = (double *)malloc(MAX_SPEC*sizeof(double));
allPrefitSpectra = (double **)malloc(MAX_SPEC*sizeof(double*));
allPostfitSpectra = (double **)malloc(MAX_SPEC*sizeof(double*));
for (i=0;i<MAX_SPEC;i++)
  {
    allPrefitSpectra[i] = (double *)malloc(nit*sizeof(double));
    allPostfitSpectra[i] = (double *)malloc(nit*sizeof(double));
  }
int nSpecPre,nSpecPost;   //if we are doing smoothing, then the pre- and post-fit, smoothing
    , interpolation number of spectral estimates will be different.
double checkVar;
double errPrefitSpectra[MAX_SPEC];
double errPostfitSpectra[MAX_SPEC];
double errPrefitSpectraLower[MAX_SPEC];
double errPostfitSpectraLower[MAX_SPEC];
FILE *allPre, *allPost, *xfer_funcs;

//SMOOTHING parameters and allocate memory.
double avgTau;
int nres_uniq[*npsr],nres_interp[*npsr];
double sum1,currentday,weight,count1,mean1,separation;
double **resY_interp,**resE_interp,**resX_interp;
double **resY2,**resE2,**resX2;
double **resX_uniq,**resE2_uniq,**resY2_uniq;

if (smooth == 1)
  {
    resX_interp = (double **)malloc(MAX_PSR*sizeof(double *));
    resX2  = (double **)malloc(MAX_PSR*sizeof(double *));
    resX_uniq = (double **)malloc(MAX_PSR*sizeof(double *));
    resY_interp = (double **)malloc(MAX_PSR*sizeof(double *));
    resE_interp = (double **)malloc(MAX_PSR*sizeof(double *));
    resY2 = (double **)malloc(MAX_PSR*sizeof(double *));
    resY2_uniq = (double **)malloc(MAX_PSR*sizeof(double *));
    resE2  = (double **)malloc(MAX_PSR*sizeof(double *));
    resE2_uniq  = (double **)malloc(MAX_PSR*sizeof(double *));
    for (p=0;p<*npsr;p++)
      {
        resX_interp[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resY_interp[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resE_interp[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resX2[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resX_uniq[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resY2[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resY2_uniq[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resE2[p] = (double *)malloc(MAX_OBSN*sizeof(double));
        resE2_uniq[p] = (double *)malloc(MAX_OBSN*sizeof(double));
      }
  }


double maxVariance[MAX_PSR], maxWeightedVariance[MAX_PSR]; //the maximum variance of the
    white noise simulated for pulsar p;
double var;  //for computational speed
double wtvar; //weighted variance
double wts[MAX_OBSN];  //the weights used in calculating the wtvar

double fixedPhase = -15000.0 - (double)toffset + phase; //fixes the phase of the grid of
    points for interpolation. The 0.0 is a variable that can change the results by shifting
    which points are in the cross spectrum  and which ones aren't
printf("fixedPhase = %g, resX[0][0] = %g, xres[1][0] = %g\n",fixedPhase, resX[0][0],resX
    [1][0]);

//BEGIN iteration to find average pre- and post-fit spectrum.
```

```
for (p=0;p<*npsr;p++)
  {
    badFitFlagWeighted = 0;
    badFitFlagNotWeighted = 0;
    sprintf(fname,"AllPrefitSpectra.dat_psr%s",psr[p].name);
    sprintf(fname,"AllPostfitSpectra.dat_psr%s",psr[p].name);
    if (useWeight == 1)
      {
        for (i=0;i<psr[p].nobs;i++)
          wts[i] = 1.0 / resE[p][i] / resE[p][i];
      }

    printf("\n=======================================================\n");
    printf("Reading data for psr %s\n",psr[p].name);
    for (it=0;it<nit;it++)
      {
        if ((it+1)%(nit/10) == 0)
          {
            printf("it: %d/%d     Simulating white noise in array white[p][it][i]\r",it+1,
                nit);
            fflush(stdout);
          }

        //Fill up "white" array with white noise realisations.
        for (i=0;i<psr[p].nobs;i++)
          {
            white[p][it][i] = TKgaussDev(&seed) * (double)(psr[p].obsn[i].toaErr*1.0e-6);
                // creates white noise consistent with error bars on each point. Note that I
                 CAN'T USE resE here because, if I'm doing an unweighted fit, resE gets
                reset to 1.0 (!!!!!) May not be true with my new TK_weightLS2 code.
            // = TKgaussDev(&seed) *1.0e-7; // creates 100ns of white noise.
          }
        if (useWeight == 0)      var = TKvariance_d(white[p][it],psr[p].nobs);
        else if (useWeight == 1) wtvar = TKfindWeightedVariance_d(white[p][it],wts,psr[p].
            nobs);

        if (it == 0)
          {
            if (useWeight == 0) maxVariance[p] = var;
            else if (useWeight == 1) maxWeightedVariance[p] = wtvar;
          }
        else
          {
            if (useWeight == 0 && var > maxVariance[p])    maxVariance[p] = var;
            else if (useWeight == 1 && wtvar > maxWeightedVariance[p])  maxWeightedVariance[
                p] = wtvar;
          }
      }
     if (useWeight == 0)       printf("\nuseWeight = %d and maxVariance[p] variance = %g\n",
         useWeight,maxVariance[p]);
     else if (useWeight == 1)       printf("\nuseWeight = %d and maxWeightedVariance[p]
         variance = %g\n",useWeight,maxWeightedVariance[p]);

     //begin transfer function calculation.
     for (it=0;it<nit;it++)
       {
         if ((it+1)%(nit/10) == 0)
           {
```

```
      printf("it: %d/%d    setting residuals equal to white noise and processing\r",it
          +1,nit);
      fflush(stdout);
    }

  if (badFitFlag == 1)  //if we had a bad fit last time, then
    {
      // Create some BRAND NEW white data
      for (i=0;i<psr[p].nobs;i++)
        {
          resY[p][i] = TKgaussDev(&seed) * (double)(psr[p].obsn[i].toaErr*1.0e-6);
          psr[p].obsn[i].sat = sat0[p][i]+((long double)(resY[p][i]))/86400.0L;
        }
      badFitFlag = 0;
    }
  else if (badFitFlag == 0)   //if the last fit was fine, use the next iteration of
      stored white data
    {
      for (i=0;i<psr[p].nobs;i++)
        {
          resY[p][i] = white[p][it][i];
          psr[p].obsn[i].sat = sat0[p][i]+((long double)(resY[p][i]))/86400.0L;
        }
    }

  psr[p].nJumps = 0;
  for(i=0;i<MAX_PARAMS;i++)            //to avoid memory errors due to array size
      overflow (e.g. Kin and SinI are linked parameters, so every iteration we will
      have a new link)
    {
      psr[p].param[i].nLinkTo = 0;      //to avoid memory errors due to array size
          overflow
      psr[p].param[i].nLinkFrom = 0;    //to avoid memory errors due to array size
          overflow
    }

  readParfile(psr+p,parFile+p,timFile+p,1);    /* Load the parameters for pulsar p
      only      Note that we are NOT re-reading the tim-file */

  vectorPulsar(psr+p,1);              /* 1. Form a vector pointing at the pulsar */
  calculate_bclt(psr+p,1);
  formBats(psr+p,1);                  /* Form Barycentric arrival times */
  formResiduals(psr+p,1,0);      //these are PREFIT residuals

  //TRUNCATE prefit data set if it has a gap in it larger than MAXALLOWABLEGAP!!! if
      maxgap for this pulsar is bigger than maxallowable gap, then choose the latest
      portion of this pulsar and discard the first few observations.
  if (maxgap[p] <= maxallowablegap)
    {
      locmaxgap[p] = -1;
    }
  if (it == 0) printf("locmaxgap[p] = %d\n",locmaxgap[p]);
  if (it == 0) printf("nres[p] = %d, psr[p].nobs = %d\n",nres[p],psr[p].nobs);
  for (i=0;i<psr[p].nobs - locmaxgap[p] - 1;i++)      //from the other side of the
      biggest gap onwards, start recording observations.
    {
      resX[p][i] = (double)(psr[p].obsn[i + locmaxgap[p] + 1].sat - toffset + 1000.0L)
          ; //Recall toffset = psr[0].param[param_pepoch].val[0].
        checkResY[p][i] = (double)psr[p].obsn[i + locmaxgap[p] + 1].residual;  // These
            are the PREFIT residuals
        resE[p][i] = (double)psr[p].obsn[i + locmaxgap[p] + 1].toaErr*1e-6; //so err is
            in seconds now
      }

    if (maxgap[p] <= maxallowablegap)
      nres[p] = psr[p].nobs;
    else
      {
        nres[p] = psr[p].nobs - locmaxgap[p] - 1;
        tspan[p] = resX[p][psr[p].nobs - 1] - resX[p][locmaxgap[p] + 1];
      }
    if (it == 0) printf("nres[p] = %d, psr[p].nobs = %d, tspan[p] = %g\n",nres[p],psr[p
        ].nobs, tspan[p]);

    if (it == 0)
      {
        if (useWeight == 0)              printf("\nvariance is %g and smooth = %d\n",
            TKvariance_d(checkResY[p],nres[p]),smooth);
        else if (useWeight == 1)         printf("\nWGTED variance is %g and smooth = %d\n"
            ,TKfindWeightedVariance_d(checkResY[p],wts,nres[p]),smooth);
      }

  //Now smooth and interpolate the white not-tempo2-fit data, calculate the power
      spectrum after smoothing and interpolation and set this as the "PREFIT" power
      spectrum.
    if (smooth == 1)  //do smoothing and then the interpolation
      {
        avgTau = 60.0; //this is the smoothing width
        if (strcmp(psr[p].name,"1939+2134")==0)          avgTau = 30.0;       //to
            remove the bump near the end of the time series.
        if (it == 0) printf("FIXING!!! avgTau = %lg\n",avgTau);

        for (i=0; i<nres[p]; i++)  //i is observation number in post-interpolated time
            series.
          {
            sum1=0.0;
            resY_interp[p][i]=0; resE_interp[p][i]=0;
            currentday = resX[p][i];      //smooth onto the same time points as the input
                time series.
            resX_interp[p][i] = currentday;

            //DO the smoothing. filterid controls whether to use gaussian or exponential
                smoother - 1 = Gaussian, 2 = expnl.
            for (k=0;k<nres[p];k++)
              {
                weight = exp(-fabs(resX[p][k] - currentday)/avgTau) / pow(resE[p][k],2);

                sum1+=weight;
                resY_interp[p][i]+=weight*checkResY[p][k];
                resE_interp[p][i]+=pow(weight*resE[p][k],2);
              }
            resY2[p][i] = resY_interp[p][i] / sum1;
            resE2[p][i] = sqrt(resE_interp[p][i] / pow(sum1,2) );
            resX2[p][i] = currentday;
          }

        count1 = 0;                 // Remove means   UNWEIGHTED!!!!!!!!
```

```
mean1 = 0.0;
for (i=0;i<nres[p];i++)
  {
    if (resY2[p][i] != 0)
      {
        mean1+=resY2[p][i];
        count1++;
      }
  }
mean1/=(double)(count1);
for (i=0;i<nres[p];i++)
  if (resY2[p][i] != 0)              resY2[p][i]-=mean1;

if (TKmean_d(resY2[p],(int)count1) > 1.0e-10)    {printf("ERROR!! mean of resY2
    [%d] = %g != 0, count1 = %d\n",p,TKmean_d(resY2[p],(int)count1), count1);
    exit(1);}


if (interp == 1)
  {
    //now interpolate smoothed data onto a regular grid using a constrained
        cubic spline - day separations given by "separation".
    separation = 14.0;  //2 weekly time series
    if (it == 0) printf("separation = %lg\n",separation);

    //NOW run unique() code on resX2 and resY2 to get a list of unique SATs and
        (SAT-sorted) residuals;
    unique(resX2[p],nres[p],resX_uniq[p],&nres_uniq[p]);
    unique(resY2[p],nres[p],resY2_uniq[p],&nres_uniq[p]);
    unique(resE2[p],nres[p],resE2_uniq[p],&nres_uniq[p]);  //MAKE SURE ERRORS
        AREN'T ALL EQUAL AT THIS POINT!!


    //Now run the spline interpolation.
    interpolateSplineSmoothFixedPhase(resX_uniq[p], resY2_uniq[p], nres_uniq[p],
         separation, resX_interp[p], resY_interp[p], &nres_interp[p],fixedPhase)
        ;

    //Now run the spline interpolation ON THE ERROR BARS using their variance.
        need calculate variance of the err2_uniq[p] array
    for (i=0;i<nres_uniq[p];i++)
      resE2_uniq[p][i] = resE2_uniq[p][i] * resE2_uniq[p][i];

    interpolateSplineSmoothFixedPhase(resX_uniq[p], resE2_uniq[p], nres_uniq[p],
         separation, resX_interp[p], resE_interp[p], &nres_interp[p],fixedPhase)
        ;

    //reset value of uniq error.
    for (i=0;i<nres_uniq[p];i++)
      resE2_uniq[p][i] = sqrt(resE2_uniq[p][i]);

    //make the interpolated resE_interp the standard deviation (error), not the
        variance.
    for (i=0;i<nres_interp[p];i++)
      resE_interp[p][i] = sqrt(resE_interp[p][i]);
  }
 else //not performing interpolation
  {
    for (i=0;i<nres[p];i++)
```

```
          {
            resX_interp[p][i] = resX2[p][i];
            resY_interp[p][i] = resY2[p][i];
            resE_interp[p][i] = resE2[p][i];
          }
        nres_interp[p] = nres[p];
      }
  }

if (smooth == 1)
  {
    if (specType!=4)
      TKspectrum(resX_interp[p],resY_interp[p],resE_interp[p],nres_interp[p
          ],0,0,0,0,0,specType,1,1,1,specXPre,specY,&nSpecPre,0,0, specY_re,
          specY_im, useWeight);
    else
         TKspectrum(resX_interp[p],resY_interp[p],resE_interp[p],nres_interp[p
             ],0,0,0,0,0,6,1,hifac,1,specXPre,specY,&nSpecPre,0,0, specY_re,
             specY_im, useWeight);
  }
else if (smooth == 0) //no smoothing done
  {
    if (specType!=4)
      TKspectrum(resX[p],checkResY[p],resE[p],nres[p],0,0,0,0,0,specType,1,1,1,
          specXPre,specY,&nSpecPre,0,0, specY_re, specY_im,useWeight);
    else
      {
        TKspectrum(resX[p],checkResY[p],resE[p],nres[p],0,0,0,0,0,6,1,hifac,1,
             specXPre,specY,&nSpecPre,0,0, specY_re, specY_im, useWeight);
      }
  }
for (i=0;i<nSpecPre;i++)
  {
    if (it == 0) avgPreSpecY[i] = 0;
    avgPreSpecY[i] += specY[i];
    allPrefitSpectra[i][it] = specY[i];
  }
//END OF SMOOTHING AND INTERPOLATING PREFIT DATA  NOW Do fitting

doFit(psr+p,1,0);              /* Do the fitting     */
vectorPulsar(psr+p,1);              /* 1. Form a vector pointing at the pulsar */
calculate_bclt(psr+p,1);            /* 3. Calculate bclt */
formBats(psr+p,1);                    /* Form Barycentric arrival times */
formResiduals(psr+p,1,0);     /* Form the residuals                */

if (maxgap[p] > maxallowablegap)    // so if there is a large gap in the data and we
     will have to truncate the data set
  {
    for (i=0;i<psr[p].nobs;i++)
      checkResY[p][i] = (double)psr[p].obsn[i].residual;
    if (useWeight == 0)         postfitVAR = TKvariance_d(checkResY[p],psr[p].nobs);
           //this is the unweighted post-fit variance!!!!!!!
    else if (useWeight == 1)  postfitWeightedVAR = TKfindWeightedVariance_d(
         checkResY[p],wts,psr[p].nobs);   //this weighted post-fit variance!
  }


//TRUNCATE postfit data set if it has a gap in it larger than MAXALLOWABLEGAP!!!
     choose the latest portion of this pulsar and discard the first few observations.
  for (i=0;i<psr[p].nobs - locmaxgap[p] - 1;i++)      //from the other side of the
      biggest gap onwards, start recording observations.
    {
      resX[p][i] = (double)(psr[p].obsn[i + locmaxgap[p] + 1].sat - toffset + 1000.0L)
          ; //Recall toffset = psr[0].param[param_pepoch].val[0],
      checkResY[p][i] = (double)psr[p].obsn[i + locmaxgap[p] + 1].residual;  // These
          are the PREFIT residuals
      resE[p][i] = (double)psr[p].obsn[i + locmaxgap[p] + 1].toaErr*1e-6; //so err is
          in seconds now
    }

  if (maxgap[p] <= maxallowablegap)
    nres[p] = psr[p].nobs;
  else
    {
      nres[p] = psr[p].nobs - locmaxgap[p] - 1;
      tspan[p] = resX[p][psr[p].nobs - 1] - resX[p][locmaxgap[p] + 1];
    }
if (it == 0)    printf("nres[p] = %d, psr[p].nobs = %d, tspan[p] = %g\n",nres[p],psr[p
    ].nobs, tspan[p]);

if (maxgap[p] <= maxallowablegap)    //because we haven't calculated the postfit
    variance yet if it's a normal data set.
  {
    if (useWeight == 0)          postfitVAR = TKvariance_d(checkResY[p],nres[p]);    //
        this is the unweighted post-fit variance!!!!!!!
    else if (useWeight == 1)  postfitWeightedVAR = TKfindWeightedVariance_d(checkResY[
        p],wts,nres[p]);    //this weighted post-fit variance!
  }

  if (useWeight == 0 && maxVariance[p] < postfitVAR)    //i.e. if the fit has pushed
      the variance to be greater than the largest variance we have input, then
    {
      ++badFitFlagNotWeighted; //record a bad fit;
      printf("\nBAD UNWEIGHTED VARIANCE FIT RECORDED, maxVariance[p] = %g, postfitVAR
          = %g, it = %d\n",maxVariance[p],postfitVAR, it);
      --it; //reset the iteration number;
      badFitFlag = 1; //this was a bad fit.
      continue;  //return to start of iteration loop with iteration value reset.
    }
  else if (useWeight == 1 && maxWeightedVariance[p] < postfitWeightedVAR)    //i.e. if
      the fit has pushed the variance to be greater than the largest weighted variance
       we have input, then
    {
      ++badFitFlagWeighted; //record a bad fit;
      printf("\nBAD WEIGHTED VARIANCE FIT RECORDED, maxWeightedVariance[p] = %g,
          postfitWeightedVAR = %g, it = %d\n",maxWeightedVariance[p],
          postfitWeightedVAR, it);
      --it; //reset the iteration number;
      badFitFlag = 1; //this was a bad fit.
      continue;  //return to start of iteration loop with iteration value reset.
    }

  //BEGIN SMOOTHING AND INTERP on postfit data
  if (smooth == 1)  //do smoothing and then the interpolation
    {
      avgTau = 60.0; //this is the smoothing width
      if (strcmp(psr[p].name,"1939+2134")==0)            avgTau = 30.0;        //to
          remove the bump near the end of the time series.
```

```
if (it == 0) printf("FIXING!!! avgTau = %lg\n",avgTau);
for (i=0; i<nres[p]; i++)  //i is observation number in post-interpolated time
    series.
  {
    sum1=0.0;
    resY_interp[p][i]=0; resE_interp[p][i]=0;
    currentday = resX[p][i];    //smooth onto the same time points as the input
        time series.
    resX_interp[p][i] = currentday;

    //DO the smoothing. filterid controls whether to use gaussian or exponential
         smoother - 1 = Gaussian, 2 = expnl.
    for (k=0;k<nres[p];k++)
      {
        weight = exp(-fabs(resX[p][k] - currentday)/avgTau) / pow(resE[p][k],2);

        sum1+=weight;
        resY_interp[p][i]+=weight*checkResY[p][k];
        resE_interp[p][i]+=pow(weight*resE[p][k],2);
      }
    resY2[p][i] = resY_interp[p][i] / sum1;
    resE2[p][i] = sqrt(resE_interp[p][i] / pow(sum1,2) );
    resX2[p][i] = currentday;
  }
// Remove means
count1 = 0;
mean1 = 0.0;
for (i=0;i<nres[p];i++)
  {
    if (resY2[p][i] != 0)
      {
        mean1+=resY2[p][i];
        count1++;
      }
  }
mean1/=(double)(count1);
for (i=0;i<nres[p];i++)
  {
    if (resY2[p][i] != 0)          resY2[p][i]-=mean1;
  }
if (TKmean_d(resY2[p],(int)count1) > 1.0e-10)     {printf("ERROR!! mean of resY2
    [%d] = %g != 0, count1 = %d\n",p,TKmean_d(resY2[p],(int)count1), count1);
    exit(1);}

if (interp == 1)
  {
    //now interpolate smoothed data onto a regular grid using a constrained
        cubic spline - day separations given by "separation".
    separation = 7.0;  //2 weekly time series
    if (it == 0) printf("separation = %lg\n",separation);

    //NOW run unique() code on resX2 and resY2 to get a list of unique SATs and
        (SAT-sorted) residuals;
    unique(resX2[p],nres[p],resX_uniq[p],&nres_uniq[p]);
    unique(resY2[p],nres[p],resY2_uniq[p],&nres_uniq[p]);
    unique(resE2[p],nres[p],resE2_uniq[p],&nres_uniq[p]);  //MAKE SURE ERRORS
        AREN'T ALL EQUAL AT THIS POINT!!

    //Now run the spline interpolation.
```

```
        interpolateSplineSmoothFixedPhase(resX_uniq[p], resY2_uniq[p], nres_uniq[p],
             separation, resX_interp[p], resY_interp[p], &nres_interp[p],fixedPhase)
            ;

        //Now run the spline interpolation ON THE ERROR BARS using their variance.
        //need calculate variance of the err2_uniq[p] array
        for (i=0;i<nres_uniq[p];i++)
          resE2_uniq[p][i] = resE2_uniq[p][i] * resE2_uniq[p][i];

        interpolateSplineSmoothFixedPhase(resX_uniq[p], resE2_uniq[p], nres_uniq[p],
             separation, resX_interp[p], resE_interp[p], &nres_interp[p],fixedPhase)
            ;
        //reset value of uniq error.
        for (i=0;i<nres_uniq[p];i++)
          resE2_uniq[p][i] = sqrt(resE2_uniq[p][i]);

        //make the interpolated resE_interp the standard deviation (error), not the
            variance.
        for (i=0;i<nres_interp[p];i++)
          resE_interp[p][i] = sqrt(resE_interp[p][i]);

      }
    else //not performing interpolation
      {
        for (i=0;i<nres[p];i++)
          {
            resX_interp[p][i] = resX2[p][i];
            resY_interp[p][i] = resY2[p][i];
            resE_interp[p][i] = resE2[p][i];
          }
        nres_interp[p] = nres[p];
      }
  }

//if no smoothing, then proceed straight to calculating postfit spectrum below.
if (smooth == 1)
  {
    if (specType!=4)
      TKspectrum(resX_interp[p],resY_interp[p],resE_interp[p],nres_interp[p
          ],0,0,0,0,0,specType,1,1,1,specXPost,specY,&nSpecPost,0,0, specY_re,
          specY_im, useWeight);
    else
          TKspectrum(resX_interp[p],resY_interp[p],resE_interp[p],nres_interp[p
              ],0,0,0,0,0,6,1,hifac,1,specXPost,specY,&nSpecPost,0,0, specY_re,
              specY_im, useWeight);
  }
else if (smooth == 0) //no smoothing done
  {
    if (specType!=4)
      TKspectrum(resX[p],checkResY[p],resE[p],nres[p],0,0,0,0,0,specType,1,1,1,
          specXPost,specY,&nSpecPost,0,0, specY_re, specY_im, useWeight);
    else
          TKspectrum(resX[p],checkResY[p],resE[p],nres[p],0,0,0,0,0,6,1,hifac,1,
              specXPost,specY,&nSpecPost,0,0, specY_re, specY_im, useWeight);
  }

for (i=0;i<nSpecPost;i++)
  {
    if (it == 0) avgPostSpecY[i] = 0;
```

```
      avgPostSpecY[i] += specY[i];
      allPostfitSpectra[i][it] = specY[i];
    }

  if (it == nit - 1)
    {
      //OUTPUT AVERAGE PREFIT SPECTRA
      printf("For pulsar %s, badFitFlagNotWeighted = %d, badFitFlagWeighted = %d, nit
          = %d\n",psr[p].name,badFitFlagNotWeighted,badFitFlagWeighted,nit);
      if (specType == 1)
        sprintf(fname,"%s_Avg_Prefit_DFT_%d",psr[p].name,nit);
      else if (specType == 2)
        sprintf(fname,"%s_Avg_Prefit_Lomb_%d",psr[p].name,nit);
      else if (specType == 4)
        {
          if (resE[p][0] == 1.0e-7 && resE[p][1] == 1.0e-7)
            sprintf(fname,"%s_Avg_Prefit_UnWLS_%d",psr[p].name,nit);
          else if (resE[p][0] == 1.0 && resE[p][1] == 1.0)
            sprintf(fname,"%s_Avg_Prefit_UnWLS_%d",psr[p].name,nit);
          else
            sprintf(fname,"%s_Avg_Prefit_WLS_%d",psr[p].name,nit);
        }
      printf("\nwriting to file %s: FREQ-PREFIT MEAN-PREFIT UPPER-PREFIT LOWER-
          PREFIT\n",fname);

      for (i=0;i<nSpecPre;i++)
        {
          printf("Sorting Prefit channel: %d/%d\r",i+1,nSpecPre);
          fflush(stdout);
          avgPreSpecY[i] /= nit;
          TKsortit(allPrefitSpectra[i],allPrefitSpectra[i],allPrefitSpectra[i],nit);

          errPrefitSpectra[i] = allPrefitSpectra[i][(int)round(0.975*nit)]; //upper
              error bar
          errPrefitSpectraLower[i] = allPrefitSpectra[i][(int)round(0.025*nit)]; //
              lower error bar

        }
      //OUTPUT ALL AVERAGE POSTFIT SPECTRA

      if (specType == 1)
        sprintf(fname,"%s_Avg_Postfit_DFT_%d",psr[p].name,nit);
      else if (specType == 2)
        sprintf(fname,"%s_Avg_Postfit_Lomb_%d",psr[p].name,nit);
      else if (specType == 4)
        {
          if (resE[p][0] == 1.0e-7 && resE[p][1] == 1.0e-7)
            sprintf(fname,"%s_Avg_Postfit_UnWLS_%d",psr[p].name,nit);
          else if (resE[p][0] == 1.0 && resE[p][1] == 1.0)
            sprintf(fname,"%s_Avg_Postfit_UnWLS_%d",psr[p].name,nit);
          else
            sprintf(fname,"%s_Avg_Postfit_WLS_%d",psr[p].name,nit);
        }

      printf("\nwriting to file %s: FREQ-POSTFIT MEAN-POSTFIT UPPER-POSTFIT LOWER-
          POSTFIT\n",fname);

      for (i=0;i<nSpecPost;i++)
        {
                printf("Sorting Postfit channel: %d/%d\r",i+1,nSpecPost);
                fflush(stdout);
                avgPostSpecY[i] /= nit;
                TKsortit(allPostfitSpectra[i],allPostfitSpectra[i],allPostfitSpectra[i],nit)
                    ;
                errPostfitSpectra[i] =  allPostfitSpectra[i][(int)round(0.975*nit)];  //
                    upper error bar
                errPostfitSpectraLower[i] =  allPostfitSpectra[i][(int)round(0.025*nit)]; //
                    lower
              }

            //OUTPUT TRANSFER FUNCTION - need to only calculate for whatever is the minimum
                of nSpecPre and nSpecPost
            sprintf(fname,"Transfer_function_SmoothInPlace.dat_psr%s",psr[p].name);
            printf("\nwriting to file %s: FREQ   AVGPRE       AVGPOST      POST / PRE\n",fname
                );
            xfer_funcs = fopen(fname,"w");
            for (i=0;i<nSpecPost;i++)
              {
                if (i >= nSpecPre)    break;   //ensures we get the minimum of nSpecPre or
                    nSpecPost
                fprintf(xfer_funcs,"%.45g %.15g %.15g %.15g\n",specXPost[i] / 86400.0,
                    avgPreSpecY[i], avgPostSpecY[i], avgPostSpecY[i] / avgPreSpecY[i]);
              }
            fclose(xfer_funcs);
            printf("DONE WRITING TO FILE\n");  printf("\nCOMPLETE\n");
  return 0;
}

//This code does a (can be weighted) least squares fit of "Asin(wt) + Bcos(wt)" to a data set.
    This least squares fit is used to determine the amplitude of the real and imaginary parts
    of the DFT.

/*******The definition of the Discrete Fourier Transform used in this code is:***********
 * X_k = 2 * (1/numPts) * X_k(wikipedia)                                                 *
 *   where we have the following definitions and justification:                         *
 * "2" ==> the spectrum below is one-sided because all the -ve frequencies are           *
 *          folded into the positive ones (so the DC term must be mult. by 2)            *
 * "1/nPts" ==> makes the numbers match up - just one of the arbitrary definitions of DFT *
 *"X_k(wikipedia)" ==> The definition of X_k given at the top of the wikipedia           *
 *                     article at "http://en.wikipedia.org/wiki/Discrete_Fourier_transform"*
 * The definition given in this articles is:                                             *
 *    X_k(wikipedia) = sum_{n=0}^{numPts} x_n * exp{-2*pi*i*k*n/numPts}                  *
 *****************************************************************************************/

//DANIEL'S ALGORITHM!! Calculate weighted least squares fit of sinusoids to the data,
    equivalent to a weighted Lomb-Scargle periodogram. Note that the returned amplitudes of
    sines and cosines correspond to the imaginary and real parts of the dft, NOT the amplitude
    of the bestfit sine or cosine that fits in the data.
void TK_weightLS2(double *x,double *y,double *sig2,int n,double *outX,double *outY,int *outN,
    double *outY_re, double *outY_im, int useWeight)
{
  int i,j,nfreq = (int)floor((double)n / 2.0) - 1;
  long double s1,s2,s3,s4,s5;
  long double omega=0.0L;
  long double si,ci;
  long double omega0;
  double sig[n];
  double mean;
double wt[n];  //an array of weights for calculating the weighted mean.
*outN = nfreq;

if (useWeight == 0)  //No weighting
  {
    for (i=0;i<n;i++)
      sig[i] = 1.0;
  }
else if (useWeight == 1)  //then use weighting
  {
    for (i=0;i<n;i++)
      {
        sig[i] = sig2[i];
        wt[i] = 1.0 / (sig[i] * sig[i]); //preparing for taking weighted mean soon;
      }
  }
else
  {
    printf("Unknown value of 'useWeight' in TK_weightLS2\n");
    exit(1);
  }

//zero mean input data - if it is a weighted fit, then the weighted mean must be zero.
if (useWeight == 0)
  mean = TKmean_d(y,n);
else if (useWeight == 1)
  mean = TKWeightedmean_d(y,wt,n);
for (i=0;i<n;i++)        y[i] -= mean;

//Assuming the input x values are in days, then tspan is in days:
double tspan = TKrange_d(x,n);
omega0 = 2.0L*M_PI/tspan;  //this matches the freq sampling of TK_dft.

for (j=0;j<nfreq;j++)
  {
    omega = omega0*(long double)(j+1);

    s1 = s2 = s3 = s4 = s5 = 0.0L;
    for (i=0;i<n;i++)
      {
        si = sinl(omega*x[i]);
        ci = cosl(omega*x[i]);

        //19th Nov 2009 - DY has checked that these are the correct expressions for a
            weighted least-squares
        s1 += y[i]*si/sig[i]/sig[i];
        s2 += si*si/sig[i]/sig[i];
        s3 += si*ci/sig[i]/sig[i];
        s4 += y[i]*ci/sig[i]/sig[i];
        s5 += ci*ci/sig[i]/sig[i];
      }
    outY_re[j] = (s4*s2 - s1*s3) / (s5*s2 - s3*s3);    //the amplitude of the best fitting
        cos wave.
    outY_im[j] = (s4*s3 - s1*s5) / (s3*s3 - s2*s5); //amp of best fitting sine wave
    outY_re[j] = outY_re[j] * (double)n / 2.0;    //the real Fourier component
    outY_im[j] = outY_im[j] * (double)n / 2.0; //imag Fourier component
    outX[j] = omega/2.0/M_PI;
    outY[j] = outY_re[j]*outY_re[j]+outY_im[j]*outY_im[j];
```

```
      //The following line assumes we want PSD output!
      outY[j] = (outY[j]/pow(365.25*86400.0,2))*2*(tspan/365.25)/(double)n/(double)n;
    }
}

//unique: a function that returns the list of unique values in an array. NB!!! It assumes the
    array has been sorted. the output array will always be smaller than or equal to the input
    array.
void unique(double *in, int nIn, double *out, int *nOut)
{
  int i;
  *nOut=0;
  for (i=0;i<nIn-1;i++)
    {
      if (in[i] == in[i+1])
        {
          continue;
        }
      else
        {
          out[*nOut] = in[i];
          ++(*nOut);
        }
    }
  out[*nOut] = in[nIn-1];
  ++(*nOut);
  return;
}

//Adapted from Stefan / George's plugin. interpolation (spline): this function interpolates a
    data set using constrained spline
void interpolateSplineSmooth(double *inX, double *inY, int inN, double separation, double *
    interpX, double *interpY, int *nInterp)
{
    //array needed by TKcmonot
    double yd[MAX_OBSN][4];

    //auxilary 'i'
    int i;
    TKcmonot(inN, inX, inY, yd);
    *nInterp = 0;
    do
      {
        interpX[*nInterp] = inX[0] + separation * (*nInterp);
        (*nInterp)++;
      } while (interpX[*nInterp - 1] < inX[inN - 1]);

    (*nInterp)--;
    TKspline_interpolate(inN, inX, inY, yd, interpX, interpY, *nInterp);
} //interpolateSplineSmooth

//interpolation (spline): this function interpolates a data set using constrained spline onto
    a fixed phase grid of points within the obseration baseline
void interpolateSplineSmoothFixedPhase(double *inX, double *inY, int inN, double separation,
    double *interpX, double *interpY, int *nInterp, double fixedStart)
{
  //array needed by TKcmonot
  double yd[MAX_OBSN][4];
  //auxilary 'i'
  int i;
  TKcmonot(inN, inX, inY, yd);
  *nInterp = 0;
  i=0;
  do
    {
      if (fixedStart + separation * i > inX[0])
        {
          interpX[*nInterp] = fixedStart + separation * i;  //only put down a point if we're
              within the observation baseline for this pulsar (we don't want to EXTRAPOLATE,
              just INTERPOLATE).
          (*nInterp)++;
          i++;
        }
      else
        i++;
    } while (interpX[*nInterp - 1] < inX[inN - 1]);

  (*nInterp)--;
  TKspline_interpolate(inN, inX, inY, yd, interpX, interpY, *nInterp);
} //interpolateSplineSmoothFixedPhase

//TKfindWeightedVariance is a function to find the weighted Variance of an input series. x is
    the array of values, wt is the array of weights, n is length of series.
double TKfindWeightedVariance_d(double *x,double *wt,int n)
{
  int i;
  double mean,var=0.0,sumwt=0.0;
  mean = TKWeightedmean_d(x,wt,n);
  for (i=0;i<n;i++)
    {
      var += pow(x[i]-mean,2)*wt[i];
      sumwt += wt[i];
    }
  var/=sumwt;
  var*=(double)n / (double)(n-1);
  return var;
}

//TKfindWeightedRMS is a function to find the weighted RMS of an input series. x is the array
    of values, wt is the array of weights, n is length of series.
double TKfindWeightedRMS_d(double *x,double *wt,int n)
{
  int i;
  double mean,sdev=0.0,sumwt=0.0;
  mean = TKWeightedmean_d(x,wt,n);
  for (i=0;i<n;i++)
    {
      sdev += pow(x[i]-mean,2)*wt[i];
      sumwt += wt[i];
    }
  sdev/=sumwt;
  sdev*=(double)n / (double)(n-1);
  sdev = sqrt(sdev);
  return sdev;
}
```

```
//makeIdealSats: a function which replaces the sats in psr[p].obsn[*].sat with ideal site
    arrival times (i.e., the sats that give 0 residuals).
void makeIdealSats(pulsar *psr,int npsr,char parFile[MAX_PSR_VAL][MAX_FILELEN],char timFile[
    MAX_PSR_VAL][MAX_FILELEN])
{
  int j,p,i;

  for (j=0;j<5;j++)
    {
      for (p=0;p<npsr;p++)
        {
          psr[p].nJumps = 0;
          for(i=0;i<MAX_PARAMS;i++)
            {
              psr[p].param[i].nLinkTo = 0;
              psr[p].param[i].nLinkFrom = 0;
            }
        }
      readParfile(psr,parFile,timFile,npsr); /* Load the parameters         */
      formBatsAll(psr,npsr);                 /* Form the barycentric arrival times */
      formResiduals(psr,npsr,0);             /* Form the residuals          */
      for (p=0;p<npsr;p++)
        {
          for (i=0;i<psr[p].nobs;i++)
            psr[p].obsn[i].sat -= (long double)psr[p].obsn[i].residual/86400.0L;
        }
    }
}
```

B.2 The PSD_SIMULATOR_PLUG.C plugin

This plugin is described in Section 2.4.3 of this thesis. It has been slightly edited for its appearance from the original source code.

```
/****************************************************************/
/*
 * This plugin takes in a powerlaw model (A*F^B for each freq F) for the low freq part of the
     PSD, then extrapolates that model assuming you've used a smoother of the form exp(-|t/tau
     |) which has a known (and hard coded) transfer function, and then simulates regularly
     spaced time series which are consistent with that PSD. There are options to use just the
     power law model an not the smoother as well, or to use the George spectral model when he
     was testing something. avgTau = 0.0 if no smoothing applied.
 * am now editing it so it can take in George's spectral density model.
 */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "TKfit.h"

using namespace std;

void makeIdealSats(pulsar *psr,int npsr,char parFile[MAX_PSR_VAL][MAX_FILELEN],char timFile[
    MAX_PSR_VAL][MAX_FILELEN]);
double TKfindWeightedVariance_d(double *x,double *wt,int n);
double TKWeightedmean_d(double *x,double *wt,int n);
#define MAX_FLAG 10
#define MAX_FREQ 10000

/* The main function called from the TEMPO2 package is 'graphicalInterface' */
/* Therefore this function is required in all plugins                      */
extern "C" int graphicalInterface(int argc,char *argv[],pulsar *psr,int *npsr)
{
  short int dir;
  int i,p,n,j,k,pp;
  double globalParameter;
  //long seed = -125;
  long seed = TKsetSeed();
  char fname[100];
  int noRed = 0;   // =1 ==> don't simulate red noise; =0 ==> do simulate red noise.
  int yesClock = 0;   // =0 ==> don't simulate clock red noise (common red noise to all
      pulsars); =1 ==> do simulate red noise component which is the same for all pulsars.

  //GWB parameters
  int   ngw=1000;
  double dist[MAX_PSR], alpha = -0.6666666666, gwamp = 1.0e-15;
  int distNum = 0;
  char parFile[MAX_PSR][MAX_FILELEN], timFile[MAX_PSR][MAX_FILELEN];
  double modelspec_y[MAX_PSR][2];  //stores the analytic power-law model spectrum for each
      pulsar in the form (mean, exponent).
  double clockspec_y[2];   //the model of the clock spectrum.
  int    nspec[MAX_PSR];
  double tspan[MAX_PSR];
  double maxTspan;
```

```
char line[1000];
FILE *fin, *fout;
int nread;
char dummy[100];   //for ensuring the scanning in of spectral models goes in the right order
     (the models match the pulsars they are supposed to match)
int nSpec;
double mean,mean2;
char specModelFile[MAX_FILELEN];
sprintf(specModelFile,"SpectralModels_Final2_psr");    //this is the prefix that goes before
     all the Spectral model files used.
*npsr = 0;
/* Obtain all parameters from the command line */
for (i=2;i<argc;i++)
  {
    if (strcmp(argv[i],"-f")==0)
      {
        strcpy(parFile[*npsr],argv[i+1]);
        strcpy(timFile[*npsr],argv[i+2]);
        (*npsr)++;
      }
    else if (strcmp(argv[i],"-specModelFile")==0) //changes prefix of input file containing
        the mean and the spectral exponent for each pulsar.
      sscanf(argv[i+1],"%s",&specModelFile);
    else if (strcmp(argv[i],"-ngw")==0)
        sscanf(argv[++i], "%d", &ngw);
    else if (strcmp(argv[i],"-gwamp")==0)
      sscanf(argv[++i], "%lf", &gwamp);
    else if (strcmp(argv[i],"-alpha")==0)
      sscanf(argv[++i], "%lf", &alpha);
    else if (strcmp(argv[i],"-seed")==0)
      sscanf(argv[++i], "%d", &seed);
    else if  (strcmp(argv[i],"-noRed")==0)
      noRed = 1;
    else if  (strcmp(argv[i],"-yesClock")==0)
      yesClock = 1;
    else if (strcmp(argv[i],"-dist")==0)
      {
        sscanf(argv[++i], "%lf", &dist[distNum]);
        dist[distNum]*=3.086e19;
        distNum++;
      }
  }

double *specX, *specY, *outY_re, *outY_im;
double **freq_in, **psd_in, **x, **y, **t, **avgSpecX, **avgSpecY;  //t is the array of x
    values obtained from the inverse DFT for each p; the PSD of the input data I'm
    simulating; AND x is real part of DFT, y is imag part of DFT
long double **sat0;
double **sat0_d;

//ALLOCATE MEMORY!!!!!!!!
specX = (double *)malloc(MAX_FREQ*sizeof(double));
specY = (double *)malloc(MAX_FREQ*sizeof(double));
outY_re = (double *)malloc(MAX_FREQ*sizeof(double));
outY_im = (double *)malloc(MAX_FREQ*sizeof(double));
sat0 = (long double **)malloc(MAX_PSR*sizeof(long double *));
freq_in = (double **)malloc((*npsr)*sizeof(double *));
psd_in = (double **)malloc((*npsr)*sizeof(double *));
x = (double **)malloc((*npsr)*sizeof(double *));
```

```
y = (double **)malloc((*npsr)*sizeof(double *));
t = (double **)malloc((*npsr)*sizeof(double *));
sat0_d = (double **)malloc(MAX_PSR*sizeof(double *));
avgSpecX = (double **)malloc((*npsr)*sizeof(double *));
avgSpecY = (double **)malloc((*npsr)*sizeof(double *));
for (p=0;p<*npsr;p++)
  {
    freq_in[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    psd_in[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    x[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    y[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    t[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    avgSpecX[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    avgSpecY[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    sat0[p] = (long double *)malloc(MAX_OBSN*sizeof(long double));
    sat0_d[p] = (double *)malloc(MAX_OBSN*sizeof(double));
  }

//Now read par and tims and form residuals. We need this for timespan of data and for
    position of pulsars in GWB
readParfile(psr,parFile,timFile,*npsr); /* Load the parameters       */
readTimfile(psr,timFile,*npsr);         /* Load the arrival times    */
preProcess(psr,*npsr,argc,argv);
//printf("Number of pulsars = %d and psr[0].nobs = %d and noClock = %d and noPlot = %d\n",*
    npsr, psr[0].nobs, noClock, noPlot);
formBatsAll(psr,*npsr);                /* Form the barycentric arrival times */
formResiduals(psr,*npsr,0);             /* Form the residuals     these are PREFIT residuals
              */
long double meanVal;
long double kp[MAX_PSR][3];
long double gwRes[MAX_PSR][MAX_OBSN];
gwSrc *gw;
if ((gw = (gwSrc *)malloc(sizeof(gwSrc)*ngw))==NULL)
  {
    printf("Unable to allocate memory for %d GW sources\n",ngw);
    exit(1);
  }
for (p=0;p<*npsr;p++)
  {
    setupPulsar_GWsim(psr[p].param[param_raj].val[0],psr[p].param[param_decj].val[0],kp[p]);
    if (distNum == 0) dist[p] = (0.91+p/10.0)*3.08568e19;
  }

//NOW read in Clock model if it was selected.
if (yesClock == 1)
  {
    sprintf(fname,"SpectralModel_Clock");
    printf("Scanning %s\n",fname);
    if ( (fin = fopen(fname,"r")) == NULL)
      {
        printf("Unable to open/find file %s\n",fname);
        exit(1);
      }

    while (!feof(fin))
      {
        if (fgets(line,1000,fin)!=NULL)
          {
            nread = sscanf(line,"%s %lg %lg",dummy,&clockspec_y[1],&clockspec_y[0]);
            if (strcmp("clock",dummy)!=0)
              {
                printf("ERROR IN MODEL SCANNING! 'clock' does not equal %s\n",dummy);
                fprintf(stderr,"ERROR scanning %s\n",fname);
                exit(1);
              }
          }
      }
    fclose(fin);
  }

//NOW READ IN MODELS for each pulsar power spectrum to be able to simulate the PSD
for (p=0;p<*npsr;p++)
  {
    sprintf(fname,specModelFile);
    strcat(fname,psr[p].name);
    printf("Scanning %s\n",fname);
    if ( (fin = fopen(fname,"r")) == NULL)
      {
        printf("Unable to open/find file %s\n",fname);
        exit(1);
      }

    while (!feof(fin))
      {
        if (fgets(line,1000,fin)!=NULL)
          {
            nread = sscanf(line,"%s %lg %lg",dummy,&modelspec_y[p][1],&modelspec_y[p][0]);
            if (strcmp(psr[p].name,dummy)!=0)
              {
                printf("ERROR IN MODEL SCANNING! %s does not equal %s\n",psr[p].name,dummy);
                fprintf(stderr,"ERROR scanning %s\n",fname);
                exit(1);
              }
          }
      }
    fclose(fin);
  }
//now we have all the spectral models (maybe including the clock spectrum). Note these will
    be multiplied by the transfer function of the expnl smoother to give the spectrum across
     all freqs.

//Calculate timespan of each data set, needed for frequency value calculation
// the following section of code finds the first and last observations for each pulsar.
long double minsat[*npsr], maxsat[*npsr], avgTspan=0.0;
double avDeltaT = 0.0; //the average sampling time

for (p=0;p<*npsr;p++)
  {
    minsat[p]=maxsat[p]=psr[p].obsn[0].sat;
    for (j=0;j<psr[p].nobs;j++)
      {
        if (psr[p].obsn[j].sat < minsat[p]) minsat[p] = psr[p].obsn[j].sat;
        if (psr[p].obsn[j].sat > maxsat[p]) maxsat[p] = psr[p].obsn[j].sat;
      }
    tspan[p] = maxsat[p] - minsat[p];
    printf("tspan of pulsar %s = %g days\n",psr[p].name,tspan[p]);
    avgTspan += (long double) tspan[p];
    avDeltaT += (tspan[p] / (psr[p].nobs - 1));
  }
maxTspan = TKfindMax_d(tspan,*npsr);
avgTspan = avgTspan / (long double) *npsr;
avDeltaT = avDeltaT / (double) *npsr;

long double toffset = psr[0].param[param_pepoch].val[0];
long double toffset2 = 7000.0L;  //so that the smoothing algorithm doesn't mess up with
    negative measured SATs.

double delta_t[*npsr];  //this is the timestep between days of simulated observation.
double numPts[*npsr];   //this is the number of points in the interpolated time series.

//for setting flo and fhi, we use t_obs and also the average Nyquist frequency:
double flo;
double fhi;
flo = 1.0L/(20.0L * maxTspan * 86400.0L); //lowest GW simulate freq is 20 times the data
    span
fhi = 1.0L/(1.0L * 86400.0L); //highest GW simulated freq is one day
gwamp*= (pow(365.2425*86400.0,alpha));

//Now calculate time between consecutive observations
for (p=0;p<*npsr;p++)
  {
    numPts[p] = 256.0;   //NB this needs to be a power of 2.
    if (numPts[p] > MAX_FREQ) fprintf(stderr,"Too many points in simulated time series -->
        increase value of MAX_FREQ\n");
    delta_t[p] = tspan[p] / numPts[p];
    printf("delta t = %g and numPts = %g\n",delta_t[p],numPts[p]);
  }
double avgTau;
long nfreq_in[*npsr];
double weights[MAX_OBSN];
avgTspan = (long double) TKmean_d(tspan,*npsr);

// Determine the idealised site arrival times
makeIdealSats(psr,*npsr,parFile,timFile);

//Store ideal sats in sat0[][] TESTED that ideal sats really are ideal (rms of resid's is 0)
for (p=0;p<*npsr;p++)
  {
    for (i=0;i<psr[p].nobs;i++)
      {
        sat0[p][i] = psr[p].obsn[i].sat;      //note these are the ideal site arrival times
        sat0_d[p][i] = (double)psr[p].obsn[i].sat;      //note these are the ideal site
            arrival times

      }
  }

//FROM HERE ON we need to choose days vs. seconds. Choosing days.
//now calculate number of channels in psd_in and compute value of psd_in which depends on
    the input spectral model and the smoothing filter used
for (p=0;p<*npsr;p++)
  {
    avgTau = 60.0;   //60 day smoothing filter for most pulsars if smoothing used.
    if (strcmp(psr[p].name,"1939+2134")==0)    avgTau = 30.0;     //to remove the bump near
        the end of the time series.
    nfreq_in[p] = (int)round((1.0 / delta_t[p]) / (1.0 / tspan[p]));
    for(i=0;i<nfreq_in[p];i++)
      {
        freq_in[p][i] = i/tspan[p];    //this is in 1/days; it is important that the
            numerator is "i", so that we include a DC term.
        if (i == 0)
          {
            psd_in[p][i] = 0.0;   //no DC term
          }
        else if (i <= (nfreq_in[p] / 2.0) )
          {
            psd_in[p][i] = modelspec_y[p][0] * pow((freq_in[p][i]/86400.0),modelspec_y[p
                ][1]);  //psd assuming freq is in days, psd measured in years
          }
        else
          {
            psd_in[p][i] = psd_in[p][ nfreq_in[p] - i ];
          }
      }
  }
int it,nit = 1;
for (it=0;it<nit;it++)
  {
    //NOW set up the GWB for this iteration!
    GWbackground(gw,ngw,&seed,flo,fhi,gwamp,alpha,1);
    for (k=0;k<ngw;k++)
      setupGW(&gw[k]);
    for (p=0;p<*npsr;p++)
      {
        for (i=0;i<nfreq_in[p];i++)
          {
            //FILL IN NEGATIVE FREQUENCIES
            if (i <= (nfreq_in[p] / 2.0) )
              {
                x[p][i] = TKgaussDev(&seed) * sqrt(psd_in[p][i] / 4.0 / tspan[p] * 365.2425)
                    ;    //so x[p] in yr
                y[p][i] = TKgaussDev(&seed) * sqrt(psd_in[p][i] / 4.0 / tspan[p] * 365.2425)
                    ;    //so y[p] in yr
              }
            else
              {
                x[p][i] = x[p][ nfreq_in[p] - i ];
                y[p][i] = -1.0 * y[p][ nfreq_in[p] - i ];
              }
          }
        //run the inverse FFT on complex array (x[p], y[p]) with x and y in units of years ^
            -1
        dir = -1;                   //-1 = inverse FFT, 1 = normal FFT
        /* DESCRIPTION OF TK_fft This computes an in-place complex-to-complex FFT x and y
            are the real and imaginary arrays of nres[p] = 2^m points. It assumes the first
            point is the DC term
        */
        TK_fft(dir,nfreq_in[p],x[p],y[p]);

        //now x[p] is a complex time series measured in years and y[p] is an independent
            complex time series in years. both are realisations of the PSD given by psd_in.
            The number of points in x[p] is nfreq_in[p].

        //convert units of x[p] y[p] to seconds (same units as residuals) and find sample
            times of x[p] and y[p]
          for (i=0;i<nfreq_in[p];i++)
            {
              x[p][i] *= 86400.0*365.2425;   //now x[p] measured in seconds.
              t[p][i] = psr[p].obsn[0].sat + i * delta_t[p];  //t is measured in days
            }

          //NOW interpolate the derived time series function onto the grid of observations for
               this pulsar.
          TKinterpolateSplineSmoothFixedXPts(t[p],x[p],nfreq_in[p],sat0_d[p],y[p],psr[p].nobs)
              ;

          //so y[p] is now the same function as x[p] but sampled at the REAL observation times
               of this pulsar.

          //NOW calculate the effect of a GWB on the pulsars
          for (i=0;i<psr[p].nobs;i++)
            {
              gwRes[p][i] = 0.0L;
              for (j=0;j<ngw;j++)
                gwRes[p][i] += calculateResidualGW(kp[p],&gw[j],(psr[p].obsn[i].sat-toffset+
                    toffset2)*86400.0L, dist[p]);
            }
          //zero mean the GWB residuals for this pulsar
          meanVal=0.0L;
          for (i=0;i<psr[p].nobs;i++)
            meanVal+=gwRes[p][i];
          meanVal/=(double)psr[p].nobs;

          //NOW add GWB + white noise + timing noise to the ideal site arrival times we made
              earlier sat0[][].

          for (i=0;i<psr[p].nobs;i++)
            {
              if (noRed == 0)
                {
                  psr[p].obsn[i].sat = sat0[p][i] + (long double)(y[p][i]/SECDAY) + ((gwRes[p
                      ][i]-meanVal)/SECDAY) + (TKgaussDev(&seed) * psr[p].obsn[i].toaErr * 1.0
                      e-6 / SECDAY);
                }
              else
                {   //don't simulate red noise from the model, just simulate white noise
                    consist. with error bars and a GWB.
                  psr[p].obsn[i].sat = sat0[p][i] + ((gwRes[p][i]-meanVal)/SECDAY) + (
                      TKgaussDev(&seed) * psr[p].obsn[i].toaErr * 1.0e-6 / SECDAY);
                }
            }

          sprintf(fname,”%s.sim.sort.tim”,psr[p].name);
          writeTim(fname,psr+p,”tempo2”);
        }
      if (it%1==0) {printf(”COMPLETE, it+1/nit = %d/%d \r”,it+1,nit); fflush(stdout);}
    }
  printf(”\nCOMPLETE\n”);
}
```

B.3 The PTA_CORRELATION_PLUG.C plugin

The algorithm that is implemented by this plugin is described in Section 6.2.1 of this thesis. It has been slightly edited for its appearance from the original source code.

```
/*************************************************************/
/* This plugin estimates the significance of a GWB signal in a set of data. The correlation is
     performed in the frequency domain. It uses the frequency domain cross-covariance
     recommended by Bill with WEIGHTING. It also uses the transfer function which can be
     calculated in xfer_func or xfer_func[2,3,4] to improve the spectral analysis. It also uses
     models for the power spectra which need to be in the specModelFiles described below.
 */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "TKfit.h"

using namespace std;
#define MAX_FLAG 10
#define MAX_FREQ 5000

double calcHD(double angle);
double calcSignificance(double *corr,double *angle,int ncorr,int npsr);
double psrangle(double centre_long,double centre_lat,double psr_long,double psr_lat);
void averagePts(float *x,float *y,int n,int width,float *meanX,float *meanY,int *nMean);
void fitHDcurve(double *x,double *y,double *e,int n,int wErr, int nharm,double *A2,int *outN,
    double *eA2, double *reduced_chisq);
void HDfunc(double x,double afunc[],int ma);
void HDfuncClk(double x1,double afunc[],int ma);   //used to fit the HD function AND a
    constant
double TKfindWeightedRMS_d(double *x,double *wt,int n);
double TKfindWeightedVariance_d(double *x,double *wt,int n);
void interpolateSplineSmooth(double *inX, double *inY, int inN, double separation, double *
    interpX, double *interpY, int *nInterp);
void unique(double *in, int nIn, double *out, int *nOut);
void TK_weightLS2(double *x,double *y,double *sig,int n,double *outX,double *outY,int *outN,
    double *outY_re, double *outY_im, int useWeight);
void TK_weightLSorig(double *x,double *y,double *sig,int n,double *outX,double *outY,int *outN
    , double *outY_re, double *outY_im);
void interpolateSplineSmoothFixedPhase(double *inX, double *inY, int inN, double separation,
    double *interpX, double *interpY, int *nInterp, double fixedStart);
void interpolateSplineSmoothFixedXPts(double *inX, double *inY, int inN, double *interpX,
    double *interpY, int nInterp);
float TKfindWeightedRMS_f(float *x,double *wt,int n);

/* The main function called from the TEMPO2 package is 'graphicalInterface' */
/* Therefore this function is required in all plugins                       */
extern "C" int graphicalInterface(int argc,char *argv[],pulsar *psr,int *npsr)
{
  char outFile[MAX_FILELEN];
  outFile[0] = 'A';
  outFile[1] = '\0';
  char outFilePairs[MAX_FILELEN];
  outFilePairs[0] = 'A';
```

```
outFilePairs[1] = '2';
outFilePairs[2] = '\0';
long idum = TKsetSeed();
int fast=0;
char parFile[MAX_PSR][MAX_FILELEN];
char timFile[MAX_PSR][MAX_FILELEN];
int i,p,n,j,k;
double globalParameter;
double **xres,**xres_uniq, **xres2, **xres3; //xres is the times of the input residuals,
    xres3 is overlapping residuals
double firstday, lastday, minDiff;
double **xres_interp, **xres_interp_1, **xres_interp_2; //for testing the power spectra of
    the 2-portion pulsars.
double **yres_interp,**yres_interp_1, **yres_interp_2; //the interpolated y values, and _1
    and _2 denote different sections of the time series with different white noise error
    bars.
double **err_interp, **err_interp_1, **err_interp_2, **var_interp;
int nres_interp[MAX_PSR]; // number of points in interpolated series.

double *clock_x, *clock_y, *clock_err; //the clock error time series.
int nclock;
double Pclock_up; //The upper bound on the power in the clock error.
double **XFER, **XFER_interp, **XFER_x, **meanPre; //meanPre is the average spectrum of
    the white noise, used for plotting purposes
int nXFER[MAX_PSR];
double **yres,**yres2, **yres2_uniq, **yres3; //yres is the initial set of residuals, yres3
    is the overlapping set.
double **err, **err2, **err2_uniq, **err3, **weights; //same definitions as yres above
double **xspec;
double **yspec_re, **yspec_im; //for the real and imaginary parts of the Fourier transform.
double **yspec,mean,mean1,mean2,**errspec; //errspec is the error on the fit of the power
    spectrum
double clockspec_x[MAX_FREQ],clockspec_y[MAX_FREQ],clockspec_err[MAX_FREQ], clockspec_y_re[
    MAX_FREQ], clockspec_y_im[MAX_FREQ];
int nclockspec;
double **crossspec_x, **crossspec_y_re, **crossspec_y_im, **crossspec_err; //the cross
    spectrum of each pair of pulsars.
int numCrossspec[MAX_PSR*MAX_PSR];
double **P_g; //this describes the gravitational wave power at that cross-spectral
    frequency for use with the prewhitening

double modelspec_y[MAX_PSR][2]; //stores the analytic power-law model spectrum for each
    pulsar in the form (mean, exponent).
int    autoFlag=0;
int    nspec[MAX_PSR];
double tspan[MAX_PSR];
double maxTspan;
long double toffset;
int    nres[MAX_PSR], nres_uniq[MAX_PSR];
int    nres2[MAX_PSR],nres3[MAX_PSR],maxNres;
float *fx1, *fy1, *yerr1,*yerr2,*fx2,*fy2,*fx3,*fy3;
float firstCommonX ,lastCommonX;
double minx,maxx,miny,maxy,minx2,maxx2,miny2,maxy2;
double rad2deg = 180.0/M_PI;
int readPar=0, readTim=0, noClock=1, noPlot=0, yesXSpec=0, yesPlotXSpec=0, noXFER=1, noEQUAD
    =1, noSpecModel = 1, noquad=1; // noClock = 0 —> DO calculate clock errors; noPlot = 0
    —> DO make a pgplot postscript file. noXFER = 1 ==> noXFER is false, so DO CORRECT by
    the transfer function. If noquad = 1, then DO fit a quadratic.
```

```
int regular = 1;  // regular = 1 ==> do smoothing IN PLACE, then interpolate onto a regular
    grid using a constrained cubic spline interpolator.
int specType = 4;  //= 1 —> DFT (requires regular sampling!!), = 2 —> Lomb periodogram, =
    4 —> (can be weighted) least squares fit of Asinwt + Bcoswt
int useWeight = 1; //0 to not use a weighted spectral estimate, = 1 to use the weights in
    the spectral estimate (only can use weights for specType = 4)
int nharm = 1;  //=1 —> default is to fit ONLY for the HD curve; if nharm = 2 then fit for
     clock error as well
int smooth = 1; //1==> default is to do smoothing, = 0 means no smoothing or interpolation.
int interp = 1; //1==> default is to do interpolation, = 0 means no interpolation.
int yesCalFac = 0;  //1 = true, so we do do the calibration, 0 = false so we don't do it.
int numCal = 10;  //the number of channels to be calibrated.
char str[1000];
double width = 100.0;
int  ngw=0;
double dist[MAX_PSR], alpha = -0.6666666666, gwamp = 1.0e-20;
double preWhAmp = 0.0;  //the amplitude by which we will pre-whiten
double factor = 1.0;
int distNum = 0;
double separation = 14.0;
//What value of alpha (the gwb spectral exponent) are we hunting for when we convert the
    estimate of A^2 into a limit etc.?
double alphaGWB = -2.0 / 3.0 ;
for (p=0;p<MAX_PSR;p++)
  dist[p] = 0.91;  //default distance is 0.91 kiloparsecs for all pulsars
int jmax;
double tk_var;
int nFreq;
float lfreqVal_f[MAX_FREQ],lpy_f[MAX_FREQ]; //log of frequency values and power values (for
    pgplot)
char line[1000];
FILE *fin;
int nread;
float meanPost[MAX_FREQ];
double sum1,sum2,sum3,sum4,sum5,sum6,weight;
int p1,p2,plotCol;
float fx[MAX_OBSN],fy[MAX_OBSN];
int ncorr=0,totalcorr;
double corr[MAX_PSR*MAX_PSR]; //correlation between pulsar pairs
double a2zeta[MAX_PSR*MAX_PSR]; //covariance between pulsar pairs
double a2zeta_im[MAX_PSR*MAX_PSR]; //imaginary part of A^2 zeta estimate for each pulsar
    pair
double a2zeta_err[MAX_PSR*MAX_PSR]; //error in each covariance between pulsar pairs
double angle[MAX_PSR*MAX_PSR]; //angle on sky betw pulsar pairs
double Toverlap[MAX_PSR*MAX_PSR]; //overlapping time interval betw pulsar pairs
double avToverlap; //the weighted average of the overlap times.
float corr_f[MAX_PSR*MAX_PSR]; //correlation between pulsar pairs
float a2zeta_f[MAX_PSR*MAX_PSR]; //covariance between pulsar pairs
float a2zeta_err_f[MAX_PSR*MAX_PSR]; //covariance between pulsar pairs
float angle_f[MAX_PSR*MAX_PSR];
float ymin,ymax; //axis limits for plotting
FILE *fout;
char fname[100];
char dummy[100];  //for ensuring the scanning in of spectral models goes in the right order
     (the models match the pulsars they are supposed to match)
char specModelFile[MAX_FILELEN];
sprintf(specModelFile,"SpectralModels_Final2_psr");
double phase = 13.0;  //the phase offset in the "fixedPhase variable" - defines where in
    the fortnight we take each sample after interpolation.
```

```
*npsr = 0;

printf("Graphical Interface: PTA_Correlation\n");
printf("Author:              D. Yardley\n");
printf("Version:             v2.0 \n");
printf(" --- type 'tempo2 -gr PTA_Correlation -h' for help information\n");

/* Obtain all parameters from the command line */
for (i=2;i<argc;i++)
  {
    if (strcmp(argv[i],"-f")==0)
      {
        strcpy(parFile[*npsr],argv[i+1]);
        strcpy(timFile[*npsr],argv[i+2]);
        (*npsr)++;
        readPar=0; readTim=0;
      }
    else if (strcmp(argv[i],"-par")==0)
      {
        readPar=1;
        readTim=0;
        *npsr=0;
      }
    else if (strcmp(argv[i],"-tim")==0)
      {
        readPar=0;
        readTim=1;
        *npsr=0;
      }
    else if (strcmp(argv[i],"-dist")==0)
      {
        sscanf(argv[++i], "%lf", &dist[distNum]);
        dist[distNum]*=3.086e19;
        distNum++;
      }
    else if (strcmp(argv[i],"-ngw")==0)
      {
        //printf("ngw = %d  2\n",ngw);
        sscanf(argv[++i], "%d", &ngw);
        //printf("ngw = %d  3\n",ngw);
      }
    else if (strcmp(argv[i],"-seed")==0)
      sscanf(argv[++i], "%d", &idum);
    else if (strcmp(argv[i],"-specModelFile")==0) //input file containing the mean and the
        spectral exponent for this range of pulsars.
      sscanf(argv[i+1],"%s",&specModelFile);
    else if (strcmp(argv[i],"-alpha")==0)
      sscanf(argv[++i], "%lf", &alpha);
    else if (strcmp(argv[i],"-alphaGWB")==0)
      sscanf(argv[++i], "%lf", &alphaGWB);
    else if (strcmp(argv[i],"-factor")==0) // factor to multiply rmses by.
      sscanf(argv[++i], "%lf", &factor);
    else if (strcmp(argv[i],"-gwamp")==0)
      sscanf(argv[++i], "%lf", &gwamp);
    else if (strcmp(argv[i],"-separation")==0)
      sscanf(argv[++i], "%lf", &separation);
    else if (strcmp(argv[i],"-preWhAmp")==0)
      sscanf(argv[++i], "%lf", &preWhAmp);
    else if (strcmp(argv[i],"-phase")==0)
  sscanf(argv[++i], "%lf", &phase);
else if  (strcmp(argv[i],"-yesClock")==0)
  noClock = 0;                        // carry out clock error estimation
else if  (strcmp(argv[i],"-noPlot")==0)
  noPlot = 1;     // do not produce the pgplot postscript file
else if  (strcmp(argv[i],"-noXFER")==0)
  noXFER = 0;     // do not correct by the transfer function
else if  (strcmp(argv[i],"-noquad")==0)
  noquad = 0;     // do not fit out a weighted quadratic to each overlapping data span
else if  (strcmp(argv[i],"-noSpecModel")==0)
  noSpecModel = 0;    // do not include spectral models
else if  (strcmp(argv[i],"-noEQUAD")==0)
  noEQUAD = 0;     // do not correct by any EQUAD terms
else if  (strcmp(argv[i],"-noSmooth")==0)
  smooth = 0;     // do not do smoothing or interpolation
else if  (strcmp(argv[i],"-noInterp")==0)
  interp = 0;     // do not do interpolation
else if  (strcmp(argv[i],"-yesXSpec")==0)
  yesXSpec = 1;     // produce an output file with the first few harmonics of the cross
      power spectrum (real and imag parts) for each pair.
else if  (strcmp(argv[i],"-yesPlotXSpec")==0)
  yesPlotXSpec = 1;     // produce plot of the cross power spectrum for each pair of
      pulsars.
else if  (strcmp(argv[i],"-yesCalFac")==0)
  yesCalFac = 1;     // Calibrate each cross spectrum using CalibrationFactors%s%s files
else if (strcmp(argv[i],"-numCal")==0)
  sscanf(argv[++i], "%d", &numCal);
else if  (strcmp(argv[i],"-noregular")==0)
  regular = 0;
else if (strcmp(argv[i],"-specType")==0)    //2 => Lomb Scarlge, 4 => Weighted Lomb
    Scargle, 1==> DFT
  sscanf(argv[++i], "%d", &specType);
else if (strcmp(argv[i],"-nharm")==0)    //=1 --> only fit for HD curve. =2 --> fit for
    HD curve and a constant simultaneously.
  sscanf(argv[++i], "%d", &nharm);
else if (strcmp(argv[i],"-nouseWeight")==0)    //=0 to do an unweighted LSQ fit of sin
    and cos to determine spectrum, = 1 to do a weighted LSQ fit.
  useWeight = 0;
else if (strcmp(argv[i],"-outFile")==0) //output file for estimate of A, A^2,
    significances and the reduced chi-squared.
  sscanf(argv[i+1],"%s",&outFile);
else if (strcmp(argv[i],"-outFilePairs")==0) //output file for each cross spectrum of
    each pair. Order is Re(1), Imag(1), Re(2), Imag(2), Re(3), Imag(3)...  One row is
    one iteration of the code.
  sscanf(argv[i+1],"%s",&outFilePairs);
else if (strcmp(argv[i],"-auto")==0)
  {
    readPar=0;
    readTim=0;
    autoFlag=1;
  }
else if (readPar==1)
  {
    strcpy(parFile[*npsr],argv[i]);
    (*npsr)++;
  }
else if (readTim==1)
  {
    strcpy(timFile[*npsr],argv[i]);
```

```
        (*npsr)++;
      }
    else if (strcmp(argv[i],"-h")==0||strcmp(argv[i],"—help")==0)
      {
        printf("\n TEMPO2 PTA_Correlation plugin\n");
        printf("==================\n");
        printf("\nUSAGE: \n\t tempo2 -gr PTA_Correlation -par *.par -tim *.tim ...(as many
             as desired) [options]\n");
        printf("\n Command line options:\n");

        printf("-yesClock: calculates clock errors (default is no clock corrections)\n");
        printf("-noPlot: does not produce the pgplot output\n");
        printf("-outFile:\t choose output file for statistics (default is 'A')\n");
        printf("-seed:\t change the random number seed (default = -123) \n");
        printf("-dist:\t input the distance to the pulsar in kpc (default is 0.91 kpc)\n");
        printf("-specType:\t determines which kind of periodogram to do —> 1 gives DFT, 2
             gives Lomb Scargle, 4 gives Weighted Lomb Scargle (default is 4)\n");
        printf("-noregular:\t  regular = 0 ==> do smoothing and interpolating together, and
             don't interpolate across gaps that are larger than 2*tau (default is smooth in
             place and then interpolate onto a regular grid)\n");
        printf("-nouseWeight:\t does not use weights to calculate the LSQ fit of sins and
             cosines to determine the power spectrum (default is to use weights)\n");
        printf("-specModelFile: \t  prefix of input file containing the mean and the
             spectral exponent for this range of pulsars (default is 'SpectralModels_Final2
             '.\n");
        printf("-outFilePairs: \t  prefix of output file for each cross spectrum of each
             pair. Order is Re(1), Imag(1), Re(2), Imag(2), Re(3), Imag(3).\n");
        printf("-noSmooth:\t don't do smoothing or interpolation\n");
        printf("-noInterp:\t don't do interpolation\n");
        printf("-noEQUAD:\t don't use the EQUAD-like correction step\n");
        printf("========GWB input stuff=====\n");
        printf("-ngw:\t Number of gws to put in simulated background.\n");
        printf("-alpha:\t spectral exponent of background\n");
        printf("-gwamp:\t dimensionless amplitude of background, but it assumes that 'f' is
             measured in 1/1s, not 1/1yr\n");
        printf("-alphaGWB:\t spectral exponent of background when determining limits etc.
             really this should be the same as alpha!!\n");
        exit(0);
      }
  }
/* Form pulsar timing residuals */
readParfile(psr,parFile,timFile,*npsr); /* Load the parameters        */
readTimfile(psr,timFile,*npsr);          /* Load the arrival times     */
preProcess(psr,*npsr,argc,argv);
formBatsAll(psr,*npsr);                /* Form the barycentric arrival times */
formResiduals(psr,*npsr,0);              /* Form the residuals      these are PREFIT residuals
     */

doFit(psr,*npsr,0);
formBatsAll(psr,*npsr);                /* Form the barycentric arrival times */
formResiduals(psr,*npsr,0);              /* Form the residuals      these are POSTFIT residuals
     */

// Allocate memory
crossspec_x    = (double **)malloc(MAX_PSR*MAX_PSR*sizeof(double *));
crossspec_y_re = (double **)malloc(MAX_PSR*MAX_PSR*sizeof(double *));
crossspec_y_im = (double **)malloc(MAX_PSR*MAX_PSR*sizeof(double *));
crossspec_err  = (double **)malloc(MAX_PSR*MAX_PSR*sizeof(double *));
P_g  = (double **)malloc(MAX_PSR*MAX_PSR*sizeof(double *));
```

```
xres = (double **)malloc(MAX_PSR*sizeof(double *));
xres2 = (double **)malloc(MAX_PSR*sizeof(double *));
xres3 = (double **)malloc(MAX_PSR*sizeof(double *));
xres_interp = (double **)malloc(MAX_PSR*sizeof(double *));
xres_interp_1 = (double **)malloc(MAX_PSR*sizeof(double *));
xres_interp_2 = (double **)malloc(MAX_PSR*sizeof(double *));
xres_uniq = (double **)malloc(MAX_PSR*sizeof(double *));
yres_interp = (double **)malloc(MAX_PSR*sizeof(double *));
yres_interp_1 = (double **)malloc(MAX_PSR*sizeof(double *));
yres_interp_2 = (double **)malloc(MAX_PSR*sizeof(double *));
err_interp = (double **)malloc(MAX_PSR*sizeof(double *));
err_interp_1 = (double **)malloc(MAX_PSR*sizeof(double *));
err_interp_2 = (double **)malloc(MAX_PSR*sizeof(double *));
var_interp = (double **)malloc(MAX_PSR*sizeof(double *));
yres = (double **)malloc(MAX_PSR*sizeof(double *));
yres2 = (double **)malloc(MAX_PSR*sizeof(double *));
yres2_uniq = (double **)malloc(MAX_PSR*sizeof(double *));
yres3 = (double **)malloc(MAX_PSR*sizeof(double *));
err  = (double **)malloc(MAX_PSR*sizeof(double *));
err2  = (double **)malloc(MAX_PSR*sizeof(double *));
err2_uniq  = (double **)malloc(MAX_PSR*sizeof(double *));
err3  = (double **)malloc(MAX_PSR*sizeof(double *));
weights  = (double **)malloc(MAX_PSR*sizeof(double *));
xspec = (double **)malloc(MAX_PSR*sizeof(double *));
yspec = (double **)malloc(MAX_PSR*sizeof(double *));
yspec_re = (double **)malloc(MAX_PSR*sizeof(double *));
yspec_im = (double **)malloc(MAX_PSR*sizeof(double *));
meanPre = (double **)malloc(MAX_PSR*sizeof(double *));
XFER = (double **)malloc(MAX_PSR*sizeof(double *));
XFER_x = (double **)malloc(MAX_PSR*sizeof(double *));
XFER_interp = (double **)malloc(MAX_PSR*sizeof(double *));
errspec = (double **)malloc(MAX_PSR*sizeof(double *)); //errspec is the error on the fit of
    the power spectrum
clock_x = (double *)malloc(MAX_OBSN*sizeof(double));
clock_y = (double *)malloc(MAX_OBSN*sizeof(double));
clock_err = (double *)malloc(MAX_OBSN*sizeof(double));

for (p=0;p<MAX_PSR;p++)
  {
    xres[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    xres2[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    xres3[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    xres_uniq[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    xres_interp[p] = (double *)malloc(10000*sizeof(double));
    xres_interp_1[p] = (double *)malloc(10000*sizeof(double));
    xres_interp_2[p] = (double *)malloc(10000*sizeof(double));
    yres_interp[p] = (double *)malloc(10000*sizeof(double));
    yres_interp_1[p] = (double *)malloc(10000*sizeof(double));
    yres_interp_2[p] = (double *)malloc(10000*sizeof(double));
    err_interp[p] = (double *)malloc(10000*sizeof(double));
    err_interp_1[p] = (double *)malloc(10000*sizeof(double));
    err_interp_2[p] = (double *)malloc(10000*sizeof(double));
    var_interp[p] = (double *)malloc(10000*sizeof(double));
    yres[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    yres2[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    yres2_uniq[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    yres3[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    err[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    err2[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    err2_uniq[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    err3[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    weights[p] = (double *)malloc(MAX_OBSN*sizeof(double));
    xspec[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    yspec[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    yspec_re[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    yspec_im[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    errspec[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    meanPre[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    XFER[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    XFER_x[p] = (double *)malloc(MAX_FREQ*sizeof(double));
    XFER_interp[p] = (double *)malloc(MAX_FREQ*sizeof(double));
  }
for (k=0;k<MAX_PSR*MAX_PSR;k++)
  {
    crossspec_x[k] = (double *)malloc(MAX_FREQ*sizeof(double));
    crossspec_y_re[k] = (double *)malloc(MAX_FREQ*sizeof(double));
    crossspec_y_im[k] = (double *)malloc(MAX_FREQ*sizeof(double));
    crossspec_err[k] = (double *)malloc(MAX_FREQ*sizeof(double));
    P_g[k] = (double *)malloc(MAX_FREQ*sizeof(double));
  }

long double minsat[*npsr], maxsat[*npsr], avgTspan=0.0;
//following is essentially for splitting up the 1857 data set, though it could be generally
    applicable.
double maxallowablegap = 2000.0; //in units of days. THIS IS AN ARBITRARY CHOICE, simply so
     that we know 1857 gets split up.
long double maxgap[*npsr];
int locmaxgap[*npsr]; //the location of the biggest gap in the time series.
for (p=0;p<*npsr;p++)
  {
    minsat[p]=maxsat[p]=psr[p].obsn[0].sat;
    maxgap[p] = 0.0L;
    for (j=0;j<psr[p].nobs;j++)
      {
        if (psr[p].obsn[j].sat < minsat[p]) minsat[p] = psr[p].obsn[j].sat;
        if (psr[p].obsn[j].sat > maxsat[p]) maxsat[p] = psr[p].obsn[j].sat;
        if (psr[p].obsn[j+1].sat - psr[p].obsn[j].sat > maxgap[p] && j < psr[p].nobs-1)
          {
            maxgap[p] = psr[p].obsn[j+1].sat - psr[p].obsn[j].sat;
            locmaxgap[p] = j;   //so the biggest gap appears between the j-th and j+1-th
                observations.
          }
      }
    tspan[p] = maxsat[p] - minsat[p];
    printf("tspan of pulsar %s = %g days\n",psr[p].name,tspan[p]);
    avgTspan += (long double) tspan[p];
    printf("maxgap[p] = %Lg\n",maxgap[p]);
  }
toffset = psr[0].param[param_pepoch].val[0];
long double toffset2 = 3100.0L; //so that the smoothing algorithm doesn't mess up with
    negative measured SATs. However, the GWB code freaks out if the day number is too big.
avgTspan = avgTspan / (long double) *npsr;
maxTspan = TKfindMax_d(tspan,*npsr);

//READ PULSAR DATA INTO MEMORY and caluclate weighted variance of residuals
sprintf(fname,"Weighted_VAR%dpsrs_GWB",*npsr);
if ( (fout = fopen(fname,"a")) == NULL)
  {
    printf("Unable to open/find file %s\n",fname);
    exit(1);
  }
for (p=0;p<*npsr;p++)
  {
    printf("Reading data for psr number %d\n",p+1);
    //now if maxgap for this pulsar is bigger than maxallowable gap, then choose the latest
        portion of this pulsar and discard the first few observations.
    if (maxgap[p] <= maxallowablegap)
      {
        locmaxgap[p] = -1;
      }
    printf("locmaxgap[p] = %d\n",locmaxgap[p]);
    printf("nres[p] = %d, psr[p].nobs = %d\n",nres[p],psr[p].nobs);
    for (i=0;i<psr[p].nobs - locmaxgap[p] - 1;i++)      //from the other side of the biggest
        gap onwards, start recording observations.
      {
        xres[p][i] = (double)(psr[p].obsn[i + locmaxgap[p] + 1].sat - toffset + toffset2);
        yres[p][i] = (double)psr[p].obsn[i + locmaxgap[p] + 1].residual;
        err[p][i] = (double)psr[p].obsn[i + locmaxgap[p] + 1].toaErr*1e-6; //so err is in
            seconds now
        weights[p][i] = 1.0 / err[p][i] / err[p][i];
      }
    if (maxgap[p] <= maxallowablegap)
      {
        nres[p] = psr[p].nobs;
      }
    else
      {
        nres[p] = psr[p].nobs - locmaxgap[p] - 1;
        psr[p].nobs = nres[p];
        tspan[p] = xres[p][psr[p].nobs - 1] - xres[p][0];
      }
    fprintf(fout,"%g ",TKfindWeightedVariance_d(yres[p],weights[p],psr[p].nobs));
  }
fprintf(fout,"\n");
fclose(fout);

// Remove unweighted mean
for (p=0;p<*npsr;p++)
  {
    mean=0.0;
    for (i=0;i<psr[p].nobs;i++)
      mean+=yres[p][i];
    mean/=(double)(psr[p].nobs);
    for (i=0;i<psr[p].nobs;i++)
      yres[p][i]-=mean;
  }

      int filterid = 2;          //filterid is an integer describing which kind of filter to
          use: 1 = Gaussian filter, 2 = Exponential filter
int filterPlot = 0;        //0 to not plot the filters, 1 to plot them.
double filter[(int)width]; //only take mean if there is more than 1 day between obsns.
double tau[*npsr];
double avgTau = 1.0;       // this is needed for the calculation of separation (how far the
    points are separated).
double invsdev = 2.5, bw;
double currentday;         //describes which day in the interpolated data set we are looking
    at.
```

```
int count1, count2;
for (p=0;p<*npsr;p++)
  {
    tau[p] = tspan[p] / 2 / M_PI / 3;      // time constant for filter, note different for
        each pulsar. The "3" is because we want the filter to fall to -6dB at the 3rd point
        in the spectrum, which occurs at 3 / tspan[p];
    avgTau *= tau[p]; //we want the geometric mean, not arithmetic...
    for (i=0;i<(int)width;i++)
      {
        if (filterid==1) //Choose Gaussian filter
          {
            invsdev = 700.0 / tau[p]; //changes (inverse of) standard deviation of filter (
                default MATLAB value is 2.5)
            filter[i]=exp(-0.5*pow(invsdev*((double)((i+1)-0.5*(width+1))/(width+1)/2.0),2))
                ;
          }
        else if (filterid==2) //form and plot exponential filter. Note we're using a
            different expnl. filter for each pulsar
          filter[i]=exp(-fabs((i-width/2)/tau[p]));
        if (noPlot == 0 && filterPlot == 1)
          {
            fx[i] = (float)i;
            fy[i] = (float)filter[i];
          }
      }
  }
avgTau = pow(avgTau, (double) (1.0 / (double)*npsr));

double fixedPhase = -15000.0 - (double)toffset + phase; //fixes the phase of the grid of
    points for interpolation.

//NOW PREPARE TIME SERIES using smoothing and interpolation
double varp_1 = 0.0, varp_2 = 0.0;  //the variance of two different sections of time series
    - e.g. if sudden change in white noise variance.
int count_1 = 0, count_2 = 0;  //the number of points before and after the cutoff point for
    the non-stationarity of the time series.

if (smooth == 1)   //then do the smoothing and possibly also the interpolation.
  {
    for (p=0;p<*npsr;p++)      //to analyse and plot all the timing resids etc.
      {
        avgTau = 60.0;
        if (strcmp(psr[p].name,"1939+2134")==0)          avgTau = 30.0;       //to remove the
            bump near the end of the time series.
        if (p == 0)  printf("fixing smoothing width to be avgTau = %lg\n",avgTau);
        if (regular == 1)            //SMOOTH pulsar p in place if we want "regular"
            resampling
          {
            for (i=0; i<nres[p]; i++)
              {
                sum1 = 0.0;  yres_interp[p][i] = 0.0; var_interp[p][i] = 0.0;
                currentday = xres[p][i];   //smooth onto the same points as the input time
                    series
                for (k=0;k<psr[p].nobs;k++)
                  {
                    if (filterid == 1) weight = exp(-0.5 * pow(xres[p][k] - currentday, 2) /
                        avgTau);
                    else if (filterid == 2) weight = exp(-fabs(currentday - xres[p][k]) /
                        avgTau) / err[p][k] / err[p][k];
        sum1+=weight;
        yres_interp[p][i]+=weight*yres[p][k];
        var_interp[p][i]+=(weight*weight*err[p][k]*err[p][k]);
      }
    yres2[p][i] = yres_interp[p][i] / sum1;
    err2[p][i] = sqrt(var_interp[p][i]) / sum1;
    xres2[p][i] = currentday;
    if (interp == 1)
      {
        //Now if any observations are too close together in time, we don't
            really need them any more (since we've already smoothed the data, we
            've taken advantage of the multiple observations). The interpolation
             step does not need lots and lots of observations to be more
            accurate, so we can set the observations equal to each other, then
            the unique() function which I run below will remove them.
        if (xres2[p][i] - xres2[p][i-1] < 1.0e-3 && i > 0)
          {
            yres2[p][i] = yres2[p][i-1];
            err2[p][i] = err2[p][i-1];
            xres2[p][i] = xres2[p][i-1];
          }
      }
  }
// Remove means
count1 = 0;
mean1 = 0.0;
for (i=0;i<nres[p];i++)          mean1+=yres2[p][i];
mean1/=(double)(nres[p]);
for (i=0;i<nres[p];i++)          yres2[p][i]-=mean1;
if (TKmean_d(yres2[p],nres[p]) > 1.0e-10)    {printf("ERROR!! mean of yres2[%d]
    = %g != 0, nres[p] = %d\n",p,TKmean_d(yres2[p],nres[p]), nres[p]); exit(1);}
if (interp == 1)
  {
    //now interpolate smoothed data onto a regular grid using a constrained
        cubic spline - day separations given by "separation".
    unique(xres2[p],nres[p],xres_uniq[p],&nres_uniq[p]);
    unique(yres2[p],nres[p],yres2_uniq[p],&nres_uniq[p]);
    unique(err2[p],nres[p],err2_uniq[p],&nres_uniq[p]);  //MAKE SURE ERRORS AREN
        'T ALL EQUAL AT THIS POINT!!

    //Now run the spline interpolation to get the residuals.
    interpolateSplineSmoothFixedPhase(xres_uniq[p], yres2_uniq[p], nres_uniq[p],
         separation, xres_interp[p], yres_interp[p], &nres_interp[p],fixedPhase)
        ;

    //Now run the spline interpolation ON THE ERROR BARS using their variance.
        First  calculate variance of the err2_uniq[p] array
    for (i=0;i<nres_uniq[p];i++)
      err2_uniq[p][i] = err2_uniq[p][i] * err2_uniq[p][i];
    interpolateSplineSmoothFixedPhase(xres_uniq[p], err2_uniq[p], nres_uniq[p],
        separation, xres_interp[p], var_interp[p], &nres_interp[p],fixedPhase);
    //reset value of uniq error to be the standard deviation FOR COMPLETENESS'
        SAKE
    for (i=0;i<nres_uniq[p];i++)
      err2_uniq[p][i] = sqrt(err2_uniq[p][i]);
    //So the almost final data set is contained in xres_interp[p],yres_interp[p
       ],var_interp[p], nres_interp[p], where var_interp[p][i] is the SQUARE of
        the error on the ith observation. This gets corrected below when we do
```

```
            the "error boosting"
      }
    else if (interp == 0)  //don't do the interpolation
      {
        for (i=0;i<nres[p];i++)
          {
            xres_interp[p][i] = xres2[p][i];
            yres_interp[p][i] = yres2[p][i];
            var_interp[p][i]  = err2[p][i] * err2[p][i];
          }
        nres_interp[p] = nres[p];
      }
  }
else if (regular == 0)     //don't smooth in place, rather smooth onto the
    interpolated grid that we want (so smoothing and interp are done together)
  {
    avgTau = 60.0; //this is the smoothing width
    if (strcmp(psr[p].name,"1939+2134")==0)
      {
        avgTau = 30.0;      //to remove the bump near the end of the time series.
        printf("FIXING!!! avgTau = %lg for psr 1939\n",avgTau);
      }
    firstday = ceil(TKfindMin_d(xres[p],psr[p].nobs));
    lastday = floor(TKfindMax_d(xres[p],psr[p].nobs));
    separation = 2.0 * avgTau;   //resampling rate is just twice the smoothing width
        .

    count1 = 0;  //a counter to tell us how many observations are actually in the
        post-interpolated time series.
    nres_interp[p] = 1 + (int)floor((lastday - firstday) / separation);   //this is
        the first guess at the number of points in the post-interpolated series.
    for (i=0; i<nres_interp[p]; i++)  //i is observation number in post-interpolated
         time series.
      {
        currentday = firstday + (double)i*separation;    //x-values are a time
            series with samples separated by "separation".
        minDiff = fabs(xres[p][0] - currentday);
        //DO the smoothing and interpolation only if the new point (xres2[p][i])
            will be within tau/1.0 of a point in the original time series. filterid
            controls whether to use gaussian or exponential smoother - 1 = Gaussian,
             2 = expnl.
        for (k=0;k<psr[p].nobs;k++)
          {
            if (fabs(xres[p][k] - currentday) < minDiff)  minDiff = fabs(xres[p][k]
                - currentday);  //could use while loop here - faster.
          }
        printf("minDiff = %g\n",minDiff);

        //if the minimum difference is still more than tau/1.0, then DON'T put down
            an interpolated point.
        if (minDiff <= (avgTau / 1.0) )   //then do the interpolation
          {
            sum1=0.0;
            yres_interp[p][count1]=0; var_interp[p][count1]=0;
            for (k=0;k<psr[p].nobs;k++)
              {
                if (filterid == 1)      weight = exp(-0.5 * pow(xres[p][k] -
                    currentday, 2) / avgTau);
                else if (filterid == 2) weight = exp(-fabs(xres[p][k] - currentday)/
                    avgTau) / pow(err[p][k],2);
                sum1+=weight;
                yres_interp[p][count1]+=weight*yres[p][k];
                var_interp[p][count1]+=pow(weight*err[p][k],2);
              }
            yres_interp[p][count1] = yres_interp[p][count1] / sum1;
            var_interp[p][count1] = var_interp[p][count1] / pow(sum1,2);
            xres_interp[p][count1] = currentday;
            ++count1;  //one more observation in the post-interpolated series.
          }
        else  //this point in the post-interpolated series is too far from the
            nearest point in the raw data series.
          continue; //return to start of loop over observation number in
              interpolated series.
      }
    nres_interp[p] = count1; //count1 now measures the correct number of points.
  }
//now adjust error bars using an EQUAD term, which may be different for the first
    few years of data compared to the last few years for the "2-portion" pulsars.
varp_1 = 0.0, varp_2 = 0.0;  //the variance of two different sections of time series
    - e.g. if sudden change in white noise variance.
count_1 = 0, count_2 = 0;  //the number of points before and after the cutoff point
    for the non-stationarity of the time series.
if (noEQUAD == 1)
  {
    if (strcmp(psr[p].name,"1600-3053")==0)
      {
        for (i=0;i<nres_interp[p];i++)
          {
            if (xres_interp[p][i] + toffset - toffset2 < 52654.0) { yres_interp_1[p
                ][count_1] = yres_interp[p][i]; ++count_1; }
            else                                                   { yres_interp_2[p
                ][count_2] = yres_interp[p][i]; ++count_2; }
          }
        varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
            yres_interp_2[p],count_2);

        //Correct error bar using unweighted variance of interpolated time series,
            in 2 pieces:
        for (i=0;i<nres_interp[p];i++)
          {
            if (xres_interp[p][i] + toffset - toffset2 < 52654.0) { err_interp[p][i]
                = sqrt(var_interp[p][i]+varp_1); }
            else                                                   { err_interp[p][i]
                = sqrt(var_interp[p][i]+varp_2); }
          }
      }
    else if (strcmp(psr[p].name,"1713+0747")==0)
      {
        for (i=0;i<nres_interp[p];i++)
          {
            if (xres_interp[p][i] + toffset - toffset2 < 52462.5) { yres_interp_1[p
                ][count_1] = yres_interp[p][i]; ++count_1;}
            else                                                   { yres_interp_2[p
                ][count_2] = yres_interp[p][i]; ++count_2;}
          }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1); varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52462.5) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"1744-1134")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52462.61) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                   { yres_interp_2[p
            ][count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1); varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);

    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52462.61) { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_1); }
        else                                                   { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"J1732-5049")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52967.5) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                 { yres_interp_2[p][
            count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1); varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52967.5) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"2124-3358")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52984.5) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                  { yres_interp_2[p][
            count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52984.5) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"2129-5721")==0)  //change in front end!!! not back
    end!!! rest are due to change in back end...
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 51410.0) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                  { yres_interp_2[p][
            count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 51410.0) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"2145-0750")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52975.5) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                  { yres_interp_2[p
            ][count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52975.5) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
```

```
              }
            else  //there are no significant sudden white noise changes in the data set, and we prefer uniformity where possible.
              {
                varp_1 = TKvariance_d(yres_interp[p],nres_interp[p]);
                varp_2 = 0.0;
                //Correct error bar using unweighted variance of interpolated time series:
                for (i=0;i<nres_interp[p];i++)
                  err_interp[p][i] = sqrt(var_interp[p][i]+varp_1);
              }
          }
        else  //don't do the correction by an EQUAD term.
          {
            varp_1 = 0.0;
            varp_2 = 0.0;
            //Correct error bar using unweighted variance of interpolated time series:
            for (i=0;i<nres_interp[p];i++)
              err_interp[p][i] = sqrt(var_interp[p][i]);
          }
      }
  }
else  //don't do any smoothing or interpolation, but DO do the error bar augmentation step !!!! This makes the least squares fitter work better.
  {
    for (p=0;p<*npsr;p++)
      {
        for (i=0; i<nres[p]; i++)
          {
            xres_interp[p][i] = xres[p][i];
            yres_interp[p][i] = yres[p][i];
            err_interp[p][i] = err[p][i];
            var_interp[p][i] = err[p][i] * err[p][i];
          }
        nres_interp[p] = nres[p];

        varp_1 = 0.0, varp_2 = 0.0;  //the variance of two different sections of time series - e.g. if sudden change in white noise variance.
        count_1 = 0, count_2 = 0;  //the number of points before and after the cutoff point for the non-stationarity of the time series.

        if (noEQUAD == 1)
          {
            if (strcmp(psr[p].name,"1600-3053")==0)
              {
                for (i=0;i<nres_interp[p];i++)
                  {
                    if (xres_interp[p][i] + toffset - toffset2 < 52654.0) { yres_interp_1[p][count_1] = yres_interp[p][i]; ++count_1; }
                    else                                                  { yres_interp_2[p][count_2] = yres_interp[p][i]; ++count_2; }
                  }
                varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(yres_interp_2[p],count_2);
                //Correct error bar using unweighted variance of interpolated time series, in 2 pieces:
                for (i=0;i<nres_interp[p];i++)
                  {
                    if (xres_interp[p][i] + toffset - toffset2 < 52654.0) { err_interp[p][i] = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"1713+0747")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52462.5) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                  { yres_interp_2[p
            ][count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52462.5) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"1744-1134")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52462.61) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                   { yres_interp_2[p
            ][count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52462.61) { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_1); }
        else                                                   { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"J1732-5049")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52967.5) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                { yres_interp_2[p][
            count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
```

```
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52967.5) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"2124-3358")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52984.5) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                  { yres_interp_2[p][
            count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52984.5) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"2129-5721")==0)  //change in front end!!! not back
    end!!!rest are due to change in back end...
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 51410.0) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                  { yres_interp_2[p][
            count_2] = yres_interp[p][i]; ++count_2; }
      }
    varp_1 = TKvariance_d(yres_interp_1[p],count_1);  varp_2 = TKvariance_d(
        yres_interp_2[p],count_2);
    //Correct error bar using unweighted variance of interpolated time series,
        in 2 pieces:
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 51410.0) { err_interp[p][i]
            = sqrt(var_interp[p][i]+varp_1); }
        else                                                  { err_interp[p][i
            ] = sqrt(var_interp[p][i]+varp_2); }
      }
  }
else if (strcmp(psr[p].name,"2145-0750")==0)
  {
    for (i=0;i<nres_interp[p];i++)
      {
        if (xres_interp[p][i] + toffset - toffset2 < 52975.5) { yres_interp_1[p
            ][count_1] = yres_interp[p][i]; ++count_1; }
        else                                                  { yres_interp_2[p
            ][count_2] = yres_interp[p][i]; ++count_2; }
```

```
                  }
                varp_1 =  TKvariance_d(yres_interp_1[p],count_1);  varp_2 =  TKvariance_d(
                    yres_interp_2[p],count_2);
                //Correct error bar using unweighted variance of interpolated time series,
                    in 2 pieces:
                for (i=0;i<nres_interp[p];i++)
                  {
                    if (xres_interp[p][i] + toffset - toffset2 < 52975.5) { err_interp[p][i]
                          = sqrt(var_interp[p][i]+varp_1); }
                    else                                                   { err_interp[p][i
                        ] = sqrt(var_interp[p][i]+varp_2); }
                  }
              }
            else  //there are no significant sudden white noise changes in the data set, and
                we prefer uniformity where possible.
              {
                varp_1 = TKvariance_d(yres_interp[p],nres_interp[p]);
                varp_2 = 0.0;
                //Correct error bar using unweighted variance of interpolated time series:
                for (i=0;i<nres_interp[p];i++)
                  err_interp[p][i] = sqrt(var_interp[p][i]+varp_1);
              }
        printf("psr = %s, nres_interp[p] = %d, count_1 = %d, varp_1 = %g, so unweighted rms
            = %g, coun_2 = %d, varp_2 = %g, unw rms = %g\n",psr[p].name, nres_interp[p],
            count_1, varp_1, sqrt(varp_1), count_2, varp_2, sqrt(varp_2));
          }
      }
  }
//NOW READ in transfer functions in preparation for the a2zetaariance calculation. We care
    about the prefit spectrum too, since this is a measure of the white noise described by
    the error bars. The transfer functions go to much higher frequencies than the "average"
    Nyquist frequency, because when we take the overlapping portions between different data
    sets, that overlapping portion may correspond to a region with a much higher density of
    points than the overall data set, meaning that the average Nyquist frequency is much
    higher for the overlapping portion than for the overall dataset.
for (p=0;p<*npsr;p++)
  {
    nXFER[p]=0;
    sprintf(fname,"Transfer_function_SmoothInPlace.dat_psr%s",psr[p].name);
    if ( (fin = fopen(fname,"r")) == NULL)
      {
        printf("Unable to open/find file %s\n",fname);
        exit(1);
      }
    while (!feof(fin))
      {
        if (fgets(line,1000,fin)!=NULL)
          {
            nread = sscanf(line,"%lg %lg %f %lg",&XFER_x[p][nXFER[p]],&meanPre[p][nXFER[p
                ]],&meanPost[nXFER[p]],&XFER[p][nXFER[p]]);
            XFER_x[p][nXFER[p]] *= 86400.0L;    //to convert back to cycles per day
            nXFER[p]++;
          }
      }
    fclose(fin);
  }
//NOW READ IN MODELS for each pulsar power spectrum in preparation for calculating errors on
     the cross power spectrum.
for (p=0;p<*npsr;p++)
```

```
  {
    sprintf(fname,specModelFile);
    strcat(fname,psr[p].name);
    if ( (fin = fopen(fname,"r")) == NULL)
      {
        printf("Unable to open/find file %s\n",fname);
        exit(1);
      }
    while (!feof(fin))
      {
        if (fgets(line,1000,fin)!=NULL)
          {
            nread = sscanf(line,"%s %lg %lg",dummy,&modelspec_y[p][1],&modelspec_y[p][0]);
            if (strcmp(psr[p].name,dummy)!=0)
              {
                printf("ERROR IN MODEL SCANNING! %s does not equal %s\n",psr[p].name,dummy);
                fprintf(stderr,"ERROR scanning %s\n",fname);
                exit(1);
              }
          }
      }
    fclose(fin);
  }
//NOW read in all the calibration factors from a file.
double calFac[*npsr * *npsr][numCal];    //this is the calibration factor
double calFreq[*npsr * *npsr][numCal];  //this is the frequency of the first 10 factors
    since we only care about the first 10 calibration factors.
double calFacErr[*npsr * *npsr][numCal];  //probably don't need this, but it's good to read
    it in since the CalFac files will now be 10 columns wide.
char dummy2[100], dummy3[100];
ncorr = 0;
for (p2=1;p2<*npsr;p2++)
  {
    for (p1=0;p1<p2;p1++)      //so p1 is always less than p2, which makes more sense given
        their names.
      {
        if (yesCalFac == 1)
          {
            sprintf(fname,"CalibrationFactors_%s%s%s",outFilePairs,psr[p1].name,psr[p2].name
                );
            if ( (fin = fopen(fname,"r")) == NULL)
              {
                printf("Unable to open/find file %s\n",fname);
                exit(1);
              }
            i = 0;
            while (i < numCal)
              {
                if (fgets(line,1000,fin)!=NULL)
                  {
                    nread = sscanf(line,"%s %s %lg %lg %lg",&dummy2, &dummy3, &calFreq[ncorr
                        ][i],&calFac[ncorr][i],&calFacErr[ncorr][i]);
                    i++;
                  }
              }
            fclose(fin);
          }
        else
          {
            for (i=0;i<numCal;i++)
              {
                calFac[ncorr][i] = 1.0;
              }
          }
        ncorr++;
      }
  }
//NOW read in the errors obtained from simulation for each estimate of a2zeta_err. This step
     saves having to run my "fitHDfast.csh" script every time. It means that this code now
    in a single pass produces the correct estimate of A^2 and the correct error bar on that
    estimate.
double junk1, junk2;
int foundErrs = 0;
ncorr = 0;
sprintf(fname,"REAL_PAIR_RESULTS");
if ( (fin = fopen(fname,"r")) == NULL)
  {
    printf("Unable to open/find file %s\n",fname);
  }
else foundErrs = 1;   //yes, we found a file with errors in it.
if (foundErrs == 1)    //i.e., if we've found a file containing the correct simulated errors
    , then read it.
  {
    for (p2=1;p2<*npsr;p2++)
      {
        for (p1=0;p1<p2;p1++)      //so p1 is always less than p2, which makes more sense
            given their names.
          {
            if (fgets(line,1000,fin)!=NULL)
              {
                nread = sscanf(line,"%s %s %lg %lg %lg %lg %lg",&dummy2, &dummy3, &angle[
                    ncorr],&a2zeta[ncorr],&a2zeta_err[ncorr], &junk1, &junk2);
              }
            ncorr++;
          }
      }
    fclose(fin);
  }
//NOW CALCULATE CORRELATION AND A2ZETAARIANCE BETWEEN DATA SETS IN THE FREQUENCY DOMAIN
int extraObsFlag = 0;
ncorr = 0;
double chisqsum = 0.0;  //CONSISTENCY check!!!! this is the sum of the chisq of the
    imaginary part of each a2zeta estimate.
double chisq_im = 0.0;  //the chisquared of the imaginary part (is chisqsum / totalcorr).
int start1, start2, end1, end2;   //the starting and ending observations of each pulsar in
    this pair. So, start1 is the first observation from pulsar 1 which IS included in the
    overlapping region (so "start1 - 1" will be the first 1 NOT to be included.)
FILE *fout2;
sprintf(fname,"Pulsar_Pairs_Results");
fout = fopen(fname,"a");
for (p2=1;p2<*npsr;p2++)
  {
    for (p1=0;p1<p2;p1++)      //so p1 is always less than p2, which makes more sense given
        their names.
      {
        printf("\n——— p1 = %s, p2 = %s ———\n",psr[p1].name, psr[p2].name);
        //for each pulsar pair, there will be a different number of overlapping points:
```

```
count1 = 0;
count2 = 0;
//find earliest and latest common data points for these two pulsars
firstCommonX = xres_interp[p1][0];
lastCommonX = xres_interp[p1][nres_interp[p1]-1];
if (firstCommonX < xres_interp[p2][0])
  {
    firstCommonX = xres_interp[p2][0];
  }
if (lastCommonX > xres_interp[p2][nres_interp[p2]-1])        lastCommonX = xres_interp
    [p2][nres_interp[p2] - 1];
//find overlapping portion of pulsar p1, and put into xres3, yres3, err3
for (i=0;i<nres_interp[p1];i++)
  {
    if (xres_interp[p1][i] - firstCommonX >= -1.0e-3 && xres_interp[p1][i] -
        lastCommonX <= 1.0e-3)
      {
        //if the previous obs does NOT fall in the overlapping region, then set the
            start1 number.
        if ((xres_interp[p1][i-1] - firstCommonX < -1.0e-3 && xres_interp[p1][i-1] -
             lastCommonX <= 1.0e-3) || i == 0)
          {
            start1 = i;
            printf("start1 = %d\n",start1);
          }
        //if the next obs does NOT fall in the overlapping region, then set the end1
             number.
        if ((xres_interp[p1][i+1] - firstCommonX >= -1.0e-3 && xres_interp[p1][i+1]
            - lastCommonX > 1.0e-3) || i == nres_interp[p1]-1)
          {
            end1 = i;
            printf("end1 = %d\n",end1);
          }
        xres3[p1][count1] = xres_interp[p1][i];
        yres3[p1][count1] = yres_interp[p1][i];
        err3[p1][count1] = err_interp[p1][i];
        ++count1;
      }
  }
//do same for pulsar p2.
for (i=0;i<nres_interp[p2];i++)
  {
    if (xres_interp[p2][i] - firstCommonX >= -1.0e-3 && xres_interp[p2][i] -
        lastCommonX <= 1.0e-3)
      {
        //if the previous obs does NOT fall in the overlapping region, then set the
            start1 number.
        if ((xres_interp[p2][i-1] - firstCommonX < -1.0e-3 && xres_interp[p2][i-1] -
             lastCommonX <= 1.0e-3) || i == 0)
          {
            start2 = i;
            printf("start2 = %d\n",start2);
          }
        //if the next obs does NOT fall in the overlapping region, then set the end1
             number.
        if ((xres_interp[p2][i+1] - firstCommonX >= -1.0e-3 && xres_interp[p2][i+1]
            - lastCommonX > 1.0e-3) || i == nres_interp[p2]-1)
          {
            end2 = i;
            printf("end2 = %d\n",end2);
          }
        xres3[p2][count2] = xres_interp[p2][i];
        yres3[p2][count2] = yres_interp[p2][i];
        err3[p2][count2] = err_interp[p2][i];
        ++count2;
      }
  }
nres3[p1] = count1; nres3[p2] = count2;
//NOW fit out a quadratic from both data sets in the overlapping portion if
    requested.
if (noquad == 1)
  {
    TKremoveWtdPoly_d(xres3[p1],yres3[p1],err3[p1],nres3[p1],3);
    TKremoveWtdPoly_d(xres3[p2],yres3[p2],err3[p2],nres3[p2],3);
  }
//so now xres3,yres3,err3,nres3 describe the two smoothed, interpolated, overlapping
    data sets we have for this pair, possibly including removal of a weighted
    quadratic from the overlap region.
printf("\n——— p1 = %s, p2 = %s ———\n numpts1 = %d, numpts2 = %d, \nfirstdayp1
    = %g, firstdayp2 = %g, lastdayp1 = %g, lastdayp2 = %g\nfirstCommonX = %g,
    lastCommonX = %g, \nnum overlapping: count1 = %d, count2 = %d [may be different
    due to differing gap size without the '-regular' option], \nspan of overlap = %g
    years\n",psr[p1].name,psr[p2].name, nres_interp[p1], nres_interp[p2],
    xres_interp[p1][0], xres_interp[p2][0], xres_interp[p1][nres_interp[p1] - 1],
    xres_interp[p2][nres_interp[p2]-1], firstCommonX, lastCommonX, nres3[p1], nres3[
    p2], (lastCommonX - firstCommonX) / 365.25);
//NOW calculate the ONE-SIDED power spectrum of each of p1 and p2 in the overlapping
    portion.
double ofacp1=1.0, ofacp2=1.0, freq0;
//now set the values of ofac such that the lowest frequency is at the SHORTER of the
    two data spans
if ((xres3[p1][nres3[p1] - 1] - xres3[p1][0]) - (xres3[p2][nres3[p2] - 1] - xres3[p2
    ][0]) > 1.0e-8)
  {
    //then p1 has a longer data span than p2, so take the frequency that goes with
        p2:
    freq0 = 1.0 / ((xres3[p2][nres3[p2] - 1] - xres3[p2][0]) * 86400.0);
    ofacp1 = 1.0 / freq0 / ((xres3[p1][nres3[p1] - 1] - xres3[p1][0]) * 86400.0);
    ofacp2 = 1.0;
  }
else if ((xres3[p1][nres3[p1] - 1] - xres3[p1][0]) - (xres3[p2][nres3[p2] - 1] -
    xres3[p2][0]) < - 1.0e-8)
  {
    //this means p2 has a longer data span than p1, so take the freq that goes with
        p1.
    freq0 = 1.0 / ((xres3[p1][nres3[p1] - 1] - xres3[p1][0]) * 86400.0);
    ofacp1 = 1.0;
    ofacp2 = 1.0 / freq0 / ((xres3[p2][nres3[p2] - 1] - xres3[p2][0]) * 86400.0);
  }
else
  {
    //this means they are the same length, so use ofacp1 = ofacp2 = 1.0;
    ofacp1 = 1.0;
    ofacp2 = 1.0;
  }
TKspectrum(xres3[p1],yres3[p1],err3[p1],nres3[p1],0,0,0,0,0,6,ofacp1,1.0,1,xspec[p1
    ],yspec[p1],&nspec[p1],0,0,yspec_re[p1],yspec_im[p1],useWeight);
```

```
TKspectrum(xres3[p2],yres3[p2],err3[p2],nres3[p2],0,0,0,0,0,6,ofacp2,1.0,1,xspec[p2
    ],yspec[p2],&nspec[p2],0,0,yspec_re[p2],yspec_im[p2],useWeight);
//Check overlap has worked because they should have the same number of channels.
if (interp != 0)
  {
    if (fabs(nspec[p1] - nspec[p2]) >= 1 ) {printf("ERROR in overlapping data: nspec
        p1 = %d, nspec p2 = %d\n",nspec[p1],nspec[p2]); fprintf(stderr,"ERROR in
        overlap\n"); exit(1);}
  }
//choose the maximum loop index to be the shorter of the two spectra
int maxloop = nspec[p1];
if (nspec[p1] > nspec[p2]) maxloop = nspec[p2];
//set number of cross spectral channels
numCrossspec[ncorr] = maxloop;
//NOW interp Xfer func of each dataset onto x-values given by their power spectra
    above if the data span has changed by taking the overlapping portion
if (TKrange_d(xres_interp[p1],nres_interp[p1]) - TKrange_d(xres3[p1],nres3[p1]) >
    1.0e-20)
  {
    //first check that XFER function is long enough for interpolation
    if (TKrange_d(xres3[p1],nres3[p1]) - 1.0/XFER_x[p1][0] > 1.0e-8)
      {
        fprintf(stderr,"Huge problem with p1 because LOWFREQ transfer function is
            too short - code will crash: psr = %s, psr # = %d, datalength = %.20g,
            lowest XFER_func frequency = %.20g\n", psr[p1].name, p1, TKrange_d(
            xres_interp[p1],nres_interp[p1]), 1.0/XFER_x[p1][0]);
        exit(1);
      }
    printf("interpolating XFER %s, since range has changed by %g days due to overlap
        \n", psr[p1].name, TKrange_d(xres_interp[p1],nres_interp[p1]) - TKrange_d(
        xres3[p1],nres3[p1]));
    interpolateSplineSmoothFixedXPts(XFER_x[p1], XFER[p1], nXFER[p1], xspec[p1],
        XFER_interp[p1], nspec[p1]);
  }
else  //the range hasn't changed, so the frequency sampling is the same and the
    value of the interpolated function is the same.
  {
    printf("All pulsar %s data contained in overlapping portion\n",psr[p1].name);
    for (i=0;i<nspec[p1];i++)          XFER_interp[p1][i] = XFER[p1][i];
  }
//NOW do the same for pulsar p2.
if (TKrange_d(xres_interp[p2],nres_interp[p2]) - TKrange_d(xres3[p2],nres3[p2]) >
    1.0e-8)
  {
    //first check that XFER function is long enough at the low frequency end
    if (TKrange_d(xres3[p2],nres3[p2]) - 1.0/XFER_x[p2][0] > 1.0e-8)
      {
        fprintf(stderr,"Huge problem with p2 because LOWFREQ transfer function is
            too short - code will crash: psr = %s, psr # = %d, datalength = %.20g,
            lowest XFER_func frequency = %.20g\n", psr[p2].name, p2, TKrange_d(
            xres_interp[p2],nres_interp[p2]), 1.0/XFER_x[p2][0]);
        exit(1);
      }
    printf("interpolating XFER %s, since range has changed by %g days due to overlap
        \n", psr[p2].name, TKrange_d(xres_interp[p2],nres_interp[p2]) - TKrange_d(
        xres3[p2],nres3[p2]));
    interpolateSplineSmoothFixedXPts(XFER_x[p2], XFER[p2], nXFER[p2], xspec[p2],
        XFER_interp[p2], nspec[p2]);
  }
```

```
else  //the range hasn't changed, so the frequency sampling is the same and the
    value of the interpolated function is the same.
  {
    printf("All pulsar %s data contained in overlapping portion\n",psr[p2].name);
    for (i=0;i<nspec[p2];i++)           XFER_interp[p2][i] = XFER[p2][i];
  }
//the following variable is a bit of a fudge factor - sometimes there is a numerical
      glitch with the last few channels of the cross spectra, or the high freqeuncies
      in the transfer function are not sufficient because the irregular sampling
    means that if, as part of an overlapping region, we encounter the higher than
    average point-density, then the transfer function will suddenly be too short at
    the high frequency end because the highest frequency in the transfer function is
      calculated as the AVERAGE separation of points.
int extraSpecChans = 4;
if (xspec[p1][maxloop - extraSpecChans - 1] - XFER_x[p1][nXFER[p1] - 1] > 1.0e-8)
  {
    fprintf(stderr,"Huge problem with p1 because HIGHFREQ transfer function is too
        short - code will crash: psr = %s, psr # = %d, pair # = %d, max xspec = %.20
        g, max XFER_func freq = %.20g\n", psr[p1].name, p1, ncorr, xspec[p1][nspec[
        p1] - extraSpecChans - 1], XFER_x[p1][nXFER[p1] - 1]);
    exit(1);
  }
if (xspec[p2][maxloop - extraSpecChans - 1] - XFER_x[p2][nXFER[p2] - 1] > 1.0e-8)
  {
    fprintf(stderr,"Huge problem with p2 because HIGHFREQ transfer function is too
        short - code will crash: psr = %s, psr # = %d, pair # = %d, max xspec = %.20
        g, max XFER_func freq = %.20g\n", psr[p2].name, p2, ncorr, xspec[p2][nspec[
        p2] - extraSpecChans - 1], XFER_x[p2][nXFER[p2] - 1]);
    exit(1);
  }
//NOW form cross spectrum of pulsars p1 and p2, only need real part since when we
    sum it to find the a2zetaariance, the imaginary parts will cancel out for
    negative and positive frequencies: therefore Re{FT[p1]xFT[p2]*} = FT_re[p1] x
    FT_re[p2] + FT_im[p1] x FT_im[p2]. However the imaginary part maybe useful for
    giving us a good estimate of the noise on each pulsar (since it will not be
    affected by GWs or clock errors)
for (i=0;i<maxloop;i++)
  {
    if (interp !=0)
      {
        if (fabs(xspec[p1][i] - xspec[p2][i]) > 1e-3) {printf("ERROR in overlapping
            data: i = %d, freq %s = %g, freq %s = %g\n",i,psr[p1].name,xspec[p1][i],
            psr[p2].name,xspec[p2][i]);  fprintf(stderr,"ERROR in overlap\n"); exit
            (1);}
      }
    if (fabs(XFER_x[p1][i] - xspec[p1][i]) < 0 || fabs(XFER_x[p2][i] - xspec[p2][i])
        < 0) {printf("Transfer function has incorrect sampling - leads to
        extrapolation not interpolation!! %g %g %g %g %g %g %g %g\n",XFER_x[p1][i],
        XFER[p1][i],xspec[p1][i], XFER_interp[p1][i],XFER_x[p2][i], XFER[p2][i],
        xspec[p2][i], XFER_interp[p2][i]);  fprintf(stderr,"ERROR in overlap\n");
        exit(1);}
    crossspec_x[ncorr][i] = xspec[p1][i];   //p1 and p2 have same x-values
    crossspec_y_re[ncorr][i] = yspec_re[p1][i] * yspec_re[p2][i] + yspec_im[p1][i] *
         yspec_im[p2][i];   //real part of 2-sided cross spectrum in (DFT units)^2
    crossspec_y_im[ncorr][i] = yspec_im[p1][i] * yspec_re[p2][i] - yspec_re[p1][i] *
         yspec_im[p2][i];   //imag part of 2-sided cross spect in (DFT units)^2
    //NOW convert the units of the real and imaginary parts of the 2-sided cross
        spectrum into units of 1-sided PSD using the same conversion factor as we
        use for the power spectra. This conversion for the power spectra is   outY[j
        ] = (outY[j]/pow(365.25*86400.0,2))*2*(tspan/365.25)/(double)n/(double)n;
        note the factor of 2 that converts 2-sided to 1-sided spectrum.
    crossspec_y_re[ncorr][i] = (crossspec_y_re[ncorr][i]/pow(365.25*86400.0,2))
        *2*(1/sqrt(xspec[p1][0]*xspec[p2][0])/365.25)/(double)nres3[p1]/(double)
        nres3[p2];
    crossspec_y_im[ncorr][i] = (crossspec_y_im[ncorr][i]/pow(365.25*86400.0,2))
        *2*(1/sqrt(xspec[p1][0]*xspec[p2][0])/365.25)/(double)nres3[p1]/(double)
        nres3[p2];
    //CONSISTENCY CHECK: real part of the cross spectrum can't exceed the sqrt of
        the product of the 2 input power spectra. This is a consequence of the
        identity (A-B)^2 >= 0, with A = Rp1*Ip2; B = Rp2*Ip1.
    if (crossspec_y_re[ncorr][i] / sqrt(yspec[p1][i] * yspec[p2][i]) > 1.01 )
      {
        fprintf(stderr,"ERROR! Cross spectrum exceeds sqrt of product of input power
             spectra: freqnum = %d, cross spec = %g, sqrt(yspec[p1] * yspec[p2] = %g
           \n", i, crossspec_y_re[ncorr][i], sqrt(yspec[p1][i] * yspec[p2][i]));
        exit(1);
      }
  }
//NOW divide the 1-sided cross spectrum by the square root of the product of the
    XFER functions. Also divide each pulsar spectrum by its own transfer function.
double sqrtprod;   //for computational speed
for (i=0;i<maxloop;i++)
  {
    sqrtprod = sqrt(XFER_interp[p1][i] * XFER_interp[p2][i]);
    if (noXFER == 0)    sqrtprod = 1.0;
    yspec[p1][i] /= XFER_interp[p1][i];
    yspec[p2][i] /= XFER_interp[p2][i];
    crossspec_y_re[ncorr][i] /= sqrtprod;
    crossspec_y_im[ncorr][i] /= sqrtprod;
    P_g[ncorr][i] = preWhAmp * preWhAmp * pow( (double)(xspec[p1][i] * 365.2425)
        ,(2.0*alphaGWB - 3.0) )   / 12.0 / M_PI / M_PI;
    crossspec_err[ncorr][i]=sqrt( (modelspec_y[p1][0] * pow(xspec[p1][i]/86400.0,
        modelspec_y[p1][1]) + P_g[ncorr][i]) * (modelspec_y[p2][0] * pow(xspec[p1][i
        ]/86400.0,modelspec_y[p2][1]) + P_g[ncorr][i]) / 2.0 );
    //NOW INCLUDING THE CALIBRATION FACTOR!!!!!! remember to calibrate the error as
        well.
    if (i < numCal && yesCalFac == 1)
      {
        crossspec_y_re[ncorr][i] = 1.0 / calFac[ncorr][i] * (crossspec_y_re[ncorr][i
            ]);
        crossspec_y_im[ncorr][i] = 1.0 / calFac[ncorr][i] * (crossspec_y_im[ncorr][i
            ]);
        crossspec_err[ncorr][i]  = 1.0 / calFac[ncorr][i] * (crossspec_err[ncorr][i
            ]);
      }
  }
//Now calculate the estimate of A^2 times zeta (using the big summation formulae)
    where the errors in the cross spectrum are NOT independent of frequency (because
    the spectra are not white in general). Note there are 2 available versions here
    - the pre-whitening version where we allow for the idea that the gravitational
    wave signal might be large enough that P_G ~ P_N, and the non-pre-whitening
    version where we assume P_G << P_N.
sum1=0.0;
sum2=0.0;
sum4=0.0;
for(k=0;k<maxloop - extraSpecChans;k++)
    sum1+= ( crossspec_y_re[ncorr][k]*pow((double)(k+1),-1.0*(3.0 - 2.0*alphaGWB)) /
         crossspec_err[ncorr][k] / crossspec_err[ncorr][k] );
    sum2+= ( pow((double)(k+1),-2.0*(3.0 - 2.0*alphaGWB)) / crossspec_err[ncorr][k]
        / crossspec_err[ncorr][k] );
    //CONSISTENCY CHECK - we don't expect the imaginary part to be correlated, so
        the chi squared of the imaginary part should be 1.
    sum4 += ( crossspec_y_im[ncorr][k]*pow((double)(k+1),-1.0*(3.0 - 2.0*alphaGWB))
        / crossspec_err[ncorr][k] / crossspec_err[ncorr][k] );
a2zeta[ncorr] = 12.0*M_PI*M_PI*pow((xspec[p1][0] * 365.2425),(3.0 - 2.0*alphaGWB)) *
    sum1 / sum2;
//CONSISTENCY check with imaginary part of the cross power spectrum.
a2zeta_im[ncorr] = 12.0*M_PI*M_PI*pow((xspec[p1][0] * 365.2425),(3.0 - 2.0*alphaGWB)
    ) * sum4 / sum2;
//NOW calculate the error on the estimate of a2zeta given that the error on the
    cross spectrum DOES vary with frequency.
sum3=0.0;
for (k=0;k<maxloop - extraSpecChans;k++)
  {
    //using the calibrated error estimate on the cross spectrum
    sum3+= (1.0 / crossspec_err[ncorr][k] / crossspec_err[ncorr][k] / pow((double)(k
        +1),2.0*(3.0 - 2.0*alphaGWB)));
  }

//If we haven't found a file with the errors from our process contained in it, then
    calculate the error analytically.
if (foundErrs == 0)
  {
    a2zeta_err[ncorr] = 12.0 * M_PI * M_PI * pow(xspec[p1][0] * 365.2425,(3.0 - 2.0*
        alphaGWB)) / sqrt(sum3);
    printf("calculating errors from the spectral models!!!! DUE to no file called
        REAL_PAIR_RESULTS !!!!!!!!!!!!!!!!!!\n");
  }
//CONSISTENCY CHECK with chisq of imaginary part
chisqsum += ( a2zeta_im[ncorr] / a2zeta_err[ncorr]) * (a2zeta_im[ncorr] /
    a2zeta_err[ncorr]);
angle[ncorr] = psrangle(psr[p2].param[param_raj].val[0],psr[p2].param[param_decj].
    val[0],
                        psr[p1].param[param_raj].val[0],psr[p1].param[param_decj].
                            val[0]);
printf("A2zetaEstimates %s %s %d %g %g %g %g\n",psr[p1].name,psr[p2].name,ncorr,
    angle[ncorr], a2zeta[ncorr], a2zeta_err[ncorr], a2zeta[ncorr] / a2zeta_err[ncorr
    ]);
/* Quick correlation */
sum1=0.0;
sum2=0.0;
sum3=0.0;
for (i=0;i<nres3[p1];i++)
  {
    sum1 += yres3[p1][i]*yres3[p2][i];
    sum2 += yres3[p1][i]*yres3[p1][i];
    sum3 += yres3[p2][i]*yres3[p2][i];
  }
corr[ncorr] = (float)(sum1/sqrt(sum2*sum3));
Toverlap[ncorr] = 1.0 / xspec[p1][0];  //the overlapping time interval of the two
    pulsars, in units of days.
fprintf(fout, "%s %s %.8g %.8g %.8g %.8g %.8g %.8g\n",psr[p1].name,psr[p2].name,
    angle[ncorr], a2zeta[ncorr], a2zeta_err[ncorr], Toverlap[ncorr] / 365.2425, corr
    [ncorr], a2zeta_im[ncorr]);
if (yesXSpec == 1)
  {
```

```
            //NB!! If you change the name of the Calibration factor files, you need to
                change the name of these ones too!!
            sprintf(fname,outFilePairs);
            strcat(fname,psr[p1].name);
            strcat(fname,psr[p2].name);
            fout2 = fopen(fname,"a");
            fprintf(fout2, "%.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g
                 %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g %.2g
                %.2g %.2g  ", crossspec_y_re[ncorr][0], crossspec_y_im[ncorr][0],
                crossspec_err[ncorr][0],crossspec_y_re[ncorr][1], crossspec_y_im[ncorr][1],
                crossspec_err[ncorr][1],crossspec_y_re[ncorr][2], crossspec_y_im[ncorr][2],
                crossspec_err[ncorr][2],crossspec_y_re[ncorr][3], crossspec_y_im[ncorr][3],
                crossspec_err[ncorr][3],crossspec_y_re[ncorr][4], crossspec_y_im[ncorr][4],
                crossspec_err[ncorr][4],crossspec_y_re[ncorr][5], crossspec_y_im[ncorr][5],
                crossspec_err[ncorr][5],crossspec_y_re[ncorr][6], crossspec_y_im[ncorr][6],
                crossspec_err[ncorr][6],crossspec_y_re[ncorr][7], crossspec_y_im[ncorr][7],
                crossspec_err[ncorr][7],crossspec_y_re[ncorr][8], crossspec_y_im[ncorr][8],
                crossspec_err[ncorr][8],crossspec_y_re[ncorr][9], crossspec_y_im[ncorr][9],
                crossspec_err[ncorr][9]);
            fprintf(fout2, "\n");
            fclose(fout2);
          }
        ncorr++;
        printf("************************END OF THIS PAIR*******************\n");
      }
  }
fclose(fout);
totalcorr = ncorr; //this is the total number of estimates of the correlations, equal to
    the nmber of pulsar pairs.
//CONSISTENCY CHECK
chisq_im = chisqsum / (double)totalcorr;
printf("chi squared of the imaginary part of the %d a2zeta estimates is %g; this number
    should be close to 1\n",totalcorr,chisq_im);
//Determine the weighted average T_overlap (in units of days) in the data set, weighted by
    error on each A^2 zeta estimate.
double wts[totalcorr];
for (i=0;i<totalcorr;i++)
  wts[i] = 1.0 / a2zeta_err[i] / a2zeta_err[i];
avToverlap = TKWeightedmean_d(Toverlap,wts,totalcorr);
printf("average overlap time is %g years\n", avToverlap / 365.2425);
// Calculate significance of detection using the Jenet et al. significance parameter "S"
double R_sig = calcSignificance(corr,angle,totalcorr,*npsr);
//PERFORM Least Squares fit to A2zetaariances...
//2 fit parameters if we are searching for the HD curve AND a constant. Just 1 if we are
    only searching for HD curve.
printf("nharm = %d\n",nharm);
double A2[nharm], eA2[nharm], afunc[10], e[totalcorr];
int outN;
eA2[0] = 0.0;
if (nharm == 2) eA2[1] = 0.0;
// Do LSQ Fit of the HD function to the data.
for (i=0;i<totalcorr;i++)
  {
    e[i] = a2zeta_err[i];
  }
int wErr = 1;   //with Error
if (wErr == 0)   printf("\n\n\n\nwErr = 0!!! This means not using errors\n\n\n");
double reduced_chisq[1];
//DO THE FIT
```

```
//FIT TO THE REAL PART (which contains the correlated signal).
fitHDcurve(angle,a2zeta,e,totalcorr,wErr,nharm,A2,&outN,eA2, reduced_chisq);
fout = fopen(outFile,"a");
char msg[100];
double B_sig = A2[0] / eA2[0];  //significance of detection
double gwAmp,UpperBoundAmp;

if (A2[0] < 0)   //if the estimate of A^2 is negative, then:
  gwAmp = 0;
else
  gwAmp = sqrt(A2[0]);

if (ngw==0) gwamp = 0.0;

printf("derived gwAmp = %g, input gwAmp = %g \n", gwAmp, gwamp/pow(365.2425*86400.0,alpha));
      //the factor is to get the input gwamp normalised to 1 year (instead of 1 second)
if (B_sig < -1.7)
  UpperBoundAmp = 0.0;
else
  UpperBoundAmp = sqrt(A2[0] + 1.7 * eA2[0]); //the number 1.7 comes from the erf function;
      it gives us a 90% confidence interval; since we only want the upper side, this
      corresponds to a 95% confidence upper bound.
//Now determine the equivalent values of omega for the estimate and the limit, using eq. 36
    from Anholm et al. paper
double h = 0.72;                          //assumed value of H_0 = 72 km / s / Mpc
double alphaOmega_GWB = 2 * alphaGWB + 2.0;       //this is the spectral exponent of the omega
      background
double H_0 = h * 100.0 * 1000.0 / 3.08568025e22 * 365.2425 * 86400.0;     //hubble constant
    in 1 / years
double f_1yr = 1.0;     // 1 / 1 year in years
double omega_GWB = A2[0] * 2.0 * M_PI * M_PI / 3.0 / H_0 / H_0 / pow(f_1yr,2.0*alphaGWB) *
    pow((1.0 / avToverlap), 2.0*alphaGWB + 2.0 - alphaOmega_GWB);
double errOmega_GWB = eA2[0] * 2.0 * M_PI * M_PI / 3.0 / H_0 / H_0 / pow(f_1yr,2.0*alphaGWB)
      * pow((1.0 / avToverlap), 2.0*alphaGWB + 2.0 - alphaOmega_GWB);
double UpperOmega_GWB = omega_GWB + 1.64485 * errOmega_GWB;
if (gwamp > 0) //if we have added an input GWB with positive amplitude:
  {
    if (nharm == 2)   //i.e. if we have fitted a clock error as well, then print out the
        parameters of the estimated clock error
      fprintf(fout,"%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%d\t%.4g\t%.4g\t%.4g\n",
          B_sig, gwAmp,gwamp*1e5,UpperBoundAmp, A2[0], eA2[0], reduced_chisq[0],chisq_im,
          totalcorr,A2[1], eA2[1],A2[1]/eA2[1]);
    else if (nharm == 1)
      fprintf(fout,"%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%d\n", B_sig, gwAmp,
          gwamp*1e5,UpperBoundAmp, A2[0], eA2[0],reduced_chisq[0],chisq_im,totalcorr);
  }
else //no simulated GWB added
  {
    if (nharm == 2)   //i.e. if we have fitted a clock error as well, then print out the
        parameters of the estimated clock error
      fprintf(fout,"%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%d\t%.4g\t%.4g\t%.4g\n", B_sig
          , gwAmp,UpperBoundAmp, A2[0], eA2[0], reduced_chisq[0],chisq_im,totalcorr,A2[1],
          eA2[1],A2[1]/eA2[1]);
    else if (nharm == 1)
      fprintf(fout,"%.4g %.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%.4g\t%d\n", R_sig, B_sig
          , gwAmp,UpperBoundAmp, A2[0], eA2[0],reduced_chisq[0],chisq_im,corr[0],totalcorr);
  }
fclose(fout);
return 0;
```

```
}

//TKfindWeightedRMS is a function to find the weighted RMS of an input series. x is the array
    of values, wt is the array of weights, n is length of series. This should agree with plk
    but is DIFFERENT to finding the rms of the weighted mean!!
double TKfindWeightedRMS_d(double *x,double *wt,int n)
{
  int i;
  double mean,sdev=0.0,sumwt=0.0;
  mean = TKWeightedmean_d(x,wt,n);
  for (i=0;i<n;i++)
    {
      sdev += pow(x[i]-mean,2)*wt[i];
      sumwt += wt[i];
    }
  sdev/=sumwt;
  sdev*=(double)n / (double)(n-1);
  sdev = sqrt(sdev);
  return sdev;
}
//TKfindWeightedRMS is a function to find the weighted RMS of an input series. x is the array
    of values, wt is the array of weights, n is length of series. NOT rms of weighted mean.
float TKfindWeightedRMS_f(float *x,double *wt,int n)
{
  int i;
  float mean;
  double sdev=0.0;
  double sumwt=0.0;
  mean = TKWeightedmean_f(x,wt,n);
  for (i=0;i<n;i++)
    {
      sdev += pow(x[i]-mean,2)*wt[i];
      sumwt += wt[i];
    }
  sdev/=sumwt;
  sdev*=(double)n / (double)(n-1);
  sdev = (float)sqrt(sdev);
  return sdev;
}

//psrangle: calculates angle on the sky between psr 1 and psr 2
double psrangle(double centre_long,double centre_lat,double psr_long,double psr_lat)
{
  double dlon,dlat,a,c;
  double deg2rad = M_PI/180.0;

  /* Apply the Haversine formula */
  dlon = (psr_long - centre_long);
  dlat = (psr_lat  - centre_lat);
  a = pow(sin(dlat/2.0),2) + cos(centre_lat) *
    cos(psr_lat)*pow(sin(dlon/2.0),2);
  if (a==1)
    c = M_PI/deg2rad;
  else
    c = 2.0 * atan2(sqrt(a),sqrt(1.0-a))/deg2rad;

  return c;
}
```

```
double calcSignificance(double *corr,double *angle,int ncorr,int npsr)
{
  double meanR,meanR2,meanEta,meanEta2,sigmaRho,sigmaR,sigmaEta;
  double hd,rho,sig,sigma_g2, rEta, clockerr;
  int i;
  char str[100];
  sigmaRho = sqrt(2)/sqrt(npsr*(npsr-1));
  meanR = 0.0;
  meanR2 = 0.0;
  meanEta = 0.0;
  meanEta2 = 0.0;
  rEta = 0.0;
  for (i=0;i<ncorr;i++)
    {
      meanR += corr[i]; //meanR is average correlation
      meanR2 += corr[i]*corr[i];
      hd = calcHD(angle[i]); //the Hellings Downs coefficient
      meanEta += hd;
      meanEta2 += hd*hd;
      rEta += corr[i]*hd; //related to a2zetaariance between HD curve and our measured
           correlations.
    }
  meanR /= ncorr;
  meanEta /= ncorr;
  rEta /= ncorr;
  sigmaR   = meanR2 - meanR*meanR;
  sigmaEta = meanEta2 - meanEta*meanEta;
  rho = 0.0;
  /* Note: subtracting the means implies that any clock error gets removed */
  for (i=0;i<ncorr;i++)
    rho += (corr[i]-meanR)*(calcHD(angle[i])-meanEta)/sqrt(sigmaR*sigmaEta);
  sig = rho/sigmaRho;
  sigma_g2 = 0.0;
  for (i=0;i<ncorr;i++)
    sigma_g2 += corr[i]*calcHD(angle[i]);
  sigma_g2 /= meanEta2;
  sprintf(str,"Significance of GW background signal = %.2g",sig);
  /* Calculate clock error */
  {
    clockerr = (meanEta2*meanR-meanEta*rEta)/sigmaEta;
    printf("Variance of clock error = %g (s^2)\n",clockerr);
  }
    /* Strength of gravity wave background */
  printf("GW background = %g (s^2) sqrt() = %g (s)\n",rho*sqrt(sigmaR/sigmaEta),sqrt(rho*sqrt(
      sigmaR/sigmaEta)));
  return sig;
}

double calcHD(double angle)
{
  if (angle == 0)     return 0.5;
  double x,ctheta;
  ctheta = cos(angle*M_PI/180.0);
  x = (1.0-ctheta)/2.0;
  return (x*log(x)-x/6.0+1.0/3.0)*3.0/2.0;
}

// This version does a fit of the HD curve only
```

```
void fitHDcurve(double *x,double *y,double *e,int n,int wErr, int nharm,double *A2,int *outN,
    double *eA2, double *reduced_chisq)
{
  FILE *fout;
  int i,j;
  long idum2 = TKsetSeed();
  double **cvm;
  double chisq=0.0;
  cvm = (double**)malloc(nharm*sizeof(double *));
  for (i=0;i<nharm;i++)
    cvm[i] = (double*)malloc(nharm*sizeof(double));
  if (nharm == 1)        TKleastSquares_svd(x,y,e,n,A2,eA2,nharm,cvm,&chisq,HDfunc,wErr);
  else if (nharm == 2)   TKleastSquares_svd(x,y,e,n,A2,eA2,nharm,cvm,&chisq,HDfuncClk,wErr);
  for (i=0;i<nharm;i++)
    for (j=0;j<nharm;j++)
      printf("cvm[%d][%d] = %g\n",i,j,cvm[i][j]);
  for (i=0;i<nharm;i++)
    for (j=0;j<nharm;j++)
      printf("correlation matrix[%d][%d] = %g\n",i,j,cvm[i][j] / sqrt(cvm[i][i]*cvm[j][j]));
  *outN = nharm;
  reduced_chisq[0] = chisq / (n - nharm); // the chi squared divided by the number of degrees
      of freedom = npts - numfits
  printf("chisq = %g & reduced_chisq = %g and wErr = %d\n\n",chisq,reduced_chisq[0],wErr);
  //NOW Correct the error bars on the "nharm" fitted parameters for the chi-squared value:
  if (wErr == 1)
    {
      printf("Correcting errors on fit using the square root of the reduced chisquared\n");
      for (i=0;i<nharm;i++)
        eA2[i] *= (sqrt(reduced_chisq[0]));
    }
  if (nharm == 2)  printf("clock error = %g +/- %g, significance of clock error = %g\n", A2
      [1], eA2[1],A2[1] / eA2[1]);
}

//This function is used when fitting for a clock error as well as the GWB
void HDfuncClk(double x1,double afunc[],int ma)
{
  int i;
  double x;
  double ctheta;
  ctheta = cos(x1*M_PI/180.0);
  x = (1.0-ctheta)/2.0;
  afunc[0] = (x*log(x)-x/6.0+1.0/3.0)*3.0/2.0;
  afunc[1] = 1.0;       //fits a constant simultaneously
}

//This function fits only for the GWB
void HDfunc(double x1,double afunc[],int ma)
{
  int i;
  double x;
  double ctheta;
  ctheta = cos(x1*M_PI/180.0);
  x = (1.0-ctheta)/2.0;
  afunc[0] = (x*log(x)-x/6.0+1.0/3.0)*3.0/2.0;
}

//Adapted from Stefan / George's plugin
```

```
//interpolation (spline): this function interpolates a data set using constrained spline onto
    an input set of interpX and nInterp values
void interpolateSplineSmoothFixedXPts(double *inX, double *inY, int inN, double *interpX,
    double *interpY, int nInterp)
{
    //array needed by TKcmonot
    double yd[MAX_OBSN][4];
    //auxilary 'i'
    int i;
    double tempX[MAX_OBSN];
    int nTemp = nInterp;
    for (i=0;i<nTemp;i++)
      {
        tempX[i] = interpX[i];
      }
    TKcmonot(inN, inX, inY, yd);
    TKspline_interpolate(inN, inX, inY, yd, tempX, interpY, nTemp);
} //interpolateSplineSmoothFixedXPts
```