Occurrence of Quasi-Periodic Oscillations in Accreting Black Holes

Alexander Mark Joseph Smith
Supervisor: C Done
Degree Programme: Theoretical Physics

April 2012

Abstract

Recently, a model has been proposed which can quantitatively describe the occurrence of quasi-periodic oscillations (QPOs) in the power spectra of accreting black holes. The accretion disc is comprised of a thin outer disc, with a hot, thick, inner accretion flow. The origin of the QPO is from the precession of this inner flow, due to the Lens Thirring effect.

We analyse data from the X-ray source XTE J1550-564, taken during its 1998 outburst by the RXTE, to test this model. The power spectra and rms spectra reveal that the source is transitioning from the low/hard state, into the very high state, then down to an intermediate state. Phase lags are obtained from the cross-spectra between 2-3 keV and 8-13 keV energy bands. Lags between hard and soft photons can be explained by a propagating fluctuation model.

We calculate the frequency resolved spectrum for the source Cyg X-1 with observation ID 10238-01-08-00 in the frequency bands 0.03-0.05 Hz, 4.5-6.8 Hz, and 23-32 Hz. These reveal the Iron Kα line at $\sim$ 6.4 keV, with a smeared absorption edge, due to reflection in the disc. The high frequency bands are harder, which agrees with the propagation model, and lack the reflection features, which agrees with the truncated disc model.

We then calculate the frequency resolved spectra for the observations of XTE J1550-564 around the QPO frequencies. We find that the QPO spectra can be fitted to a Comptonisation model, and this is consistent with the Lens Thirring model. Some produce poor fits, which may be due to reflection. The time averaged spectra are fitted to a model of a blackbody component, a Compton component, and a Gaussian. This fit has a very small blackbody component, and a Comptonisation component which is much softer than the QPO spectrum. Improving the model of the reflection features finds that there is a significant blackbody component, and a Compton component which matches the QPO spectrum.

Contents

1 Introduction

Black holes are one of the most fascinating types of object in the universe. Born during the death of the most massive stars, black holes exhibit gravitational fields so strong that any matter or light which wanders too close cannot escape. Black holes are hidden behind an invisible boundary called the event horizon. Beyond this, the escape velocity exceeds the speed of light, and thus it is impossible to see the singularity at the centre. However, despite the name, it is theorised that black holes can emit radiation from the event horizon, but this Hawking radiation has yet to be observed.

It is thought that all galaxies contain a supermassive black hole at their centre with a mass millions of times the mass of the Sun. The most striking piece of evidence for a supermassive black hole in the centre of our own galaxy is through observations of stellar motions. The orbits of stars in the core of the galaxy imply that they are orbiting an object with millions of solar masses in an area smaller than the solar system. The only possible conclusion is that this is a black hole. The supermassive black hole in the Milky Way is currently quiet, but if we look at more distant galaxies, we can see active galaxies and quasars, where the supermassive black hole is consuming matter through the process of accretion. These are some of the brightest and most energetic objects in the universe. Accretion can also be seen on much smaller scales, with many stellar mass black holes in binary systems where the black hole is stripping gas from a normal star.

The physics of accretion flows is very complicated, with many physical processes taking place, and not all are fully understood. Analysis of the light from many accreting black holes reveal quasi-periodic oscillations (QPOs). These have been known about for many years, but only recently has a quantitative description been proposed which can explain their occurrence. In this project we will analyse data from the X-ray source XTE J1550-564 in order to understand the QPOs and see how the data fits the current models.

1.1 Accretion

Matter falling into a black hole will form into a flattened disc, called the accretion disc. For material in the disc to fall into the black hole, it must lose angular momentum, and therefore there must be a torque, since $\tau = \frac{\mathrm{d}L}{\mathrm{d}t}$. This torque arises from Keplerian motion and disc viscosity. An object in a Keplerian orbit at a distance R from the black hole will orbit with an angular frequency, $\omega \propto R^{-1/3}$. Thus material farther from the black hole will orbit more slowly than the material closer in. Viscosity leads to shear stresses in the differentially rotating disc, which generates a torque (Shakura & Sunyaev, 1973). Friction between different layers in the disc moving at different speeds heats up the disc, and the disc is dense enough to thermalise the energy, it radiates as a blackbody

Viscosity in the disc arises from a magneto-rotational instability (MRI) (Hawley, Gammie & Balbus 1995). However, the physics can be simplified into a dimensionless

parameter, $\alpha \leq 1$ (Shakura & Sunyaev, 1973), and the torque on the disc due to viscosity is,

$$\tau_{visc} = -3\pi\nu_{visc}\Sigma\sqrt{GMr} \tag{1}$$

where Σ is the surface density and r the radius of the disc, M the mass of the black hole, and the viscosity, $\nu_{visc} = \alpha H c_s$, where H is the scale height and c_s the sound speed (Kolb 2010). A negative torque indicates a loss in angular momentum, and therefore material will spiral into the black hole.

For an accreting black hole with a mass accretion rate of $\dot{m}$, the rate of change in potential energy as this mass spirals in from a distance R to $R - dR$ is given by, $\frac{\mathrm{d}E}{\mathrm{d}t} = L_{pot} = \frac{GM\dot{m}}{R^2}\mathrm{d}R$. However, from the Virial theorem, only half of this can be radiated,

$$\mathrm{d}L_{rad} = \frac{GM\dot{m}}{2R^2}\mathrm{d}R \tag{2}$$

If the energy is radiated as a blackbody, then

$$\mathrm{d}L = \mathrm{d}A\sigma_{SB}T^4 \tag{3}$$

where T is temperature, σ_{SB} is the Stephan-Boltzmann constant, and the area is the area of an annulus (with top and bottom) $dA = 2 \times 2\pi R \times dR$. By equating equations 2 and 3, we find that,

$$\sigma_{SB}T^4 = \frac{GM\dot{m}}{8\pi R^3} \tag{4}$$

This derivation does not take into account angular momentum, which produces an extra factor of $3\left(1 - \sqrt{R_{in}/R}\right)$, where R_{in} is the inner radius of the accretion disc (Shakura & Sunyaev, 1973). The closest R_{in} can be to the black hole is at the innermost stable circular orbit (ISCO). For a Schwarzschild (non-rotating) black hole, the ISCO is at a distance of $6GM/c^2$, or $3R_s$, where $R_s = 2GM/c^2$ is the Schwarzschild radius (the radius of the event horizon). If the black hole is more massive, then the ISCO is larger, and thus, from equation 4 the accretion disc radiates at a lower temperature. For stellar mass black holes, this radiation is predominantly emitted in the X-ray region of the spectrum, while for much larger supermassive black holes, it is in the ultra-violet.

1.2 Spectra of Binary Black Hole Systems

The spectrum of a source often follows a power law, $N(E) = N_0 E^{-\Gamma}$, where $N(E)$ is the differential photon number density (the number of photons per second per square cm per energy band), and Γ is the photon index. The energy flux, $F(E) = EN(E) = N_0 E^{-(\Gamma-1)} = N_0 E^{-\alpha}$, where $\alpha \equiv \Gamma - 1$ is the energy index (Done 2010). Typically, a spectrum spans many orders of magnitude in energy, and thus is plotted logarithmically. The total energy per bin is given by $F(E)\mathrm{d}E = EF(E)\mathrm{d}E/E = EF(E)\mathrm{d}\log E$, and

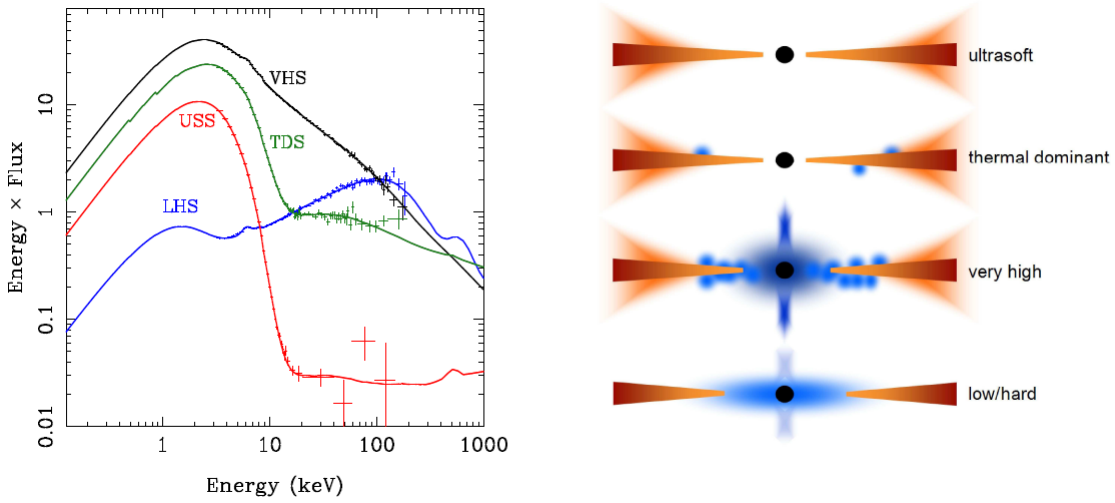

Figure 1: Left: spectra of the source GRO J1655-40, showing the low/hard state (blue) with a weak disc and strong hard power law, the high/soft state (red and green) with a strong disc and weak, soft, power law, and the very high state (black) with both a strong disc and strong soft power law tail. Right: diagram showing how the size and strengths of the different components produce the spectral states. The outer disc is shown in orange, with the inner hot flow in blue. Other processes such as jets and winds can also occur. (Figure from DGK07)

therefore we plot $\log EF(E)$ against $\log E$, rather than simply $\log F(E)$. If $\Gamma < 2$, the spectrum peaks at high energy, and is referred to as hard, while if $\Gamma > 2$, the spectrum peaks at low energies, and is referred to as soft.

Many accreting black holes in binary systems show a spectrum consisting of a blackbody component from the disc, plus a power law tail. Over time, the strengths of these components change to produce several distinct spectral states. Sources with low luminosities are typically seen with a spectrum dominated by a hard ($\Gamma < 2$) power law tail. This is called the low/hard state. When a source brightens, it moves into an intermediate state, where there is an increase in the disc component, and a softer tail ($\Gamma \sim 2$). At even higher luminosities, the source can retain a strong tail ($\Gamma \sim 2.5$) in addition to a strong disc component (very high state), or the disc dominates with a very weak tail ($\Gamma \sim 2.2$) in the high/soft state (Done, Gierliński & Kubota 2007, hereafter DGK07) (figure 1a).

The various spectral states can be described by a truncated disc model. The outer part of the disc is cool, geometrically thin and optically thick, down to a truncation radius. Within this is an advection dominated accretion flow (ADAF) which is hot, geometrically thick and optically thin, down to the last stable orbit. The inner flow produces the power law through Compton up-scattering. Compton scattering is an energy exchange process between an electron and a photon. In Compton (down-)scattering, the electron recoils, gaining energy from the photon, while in Compton up-scattering

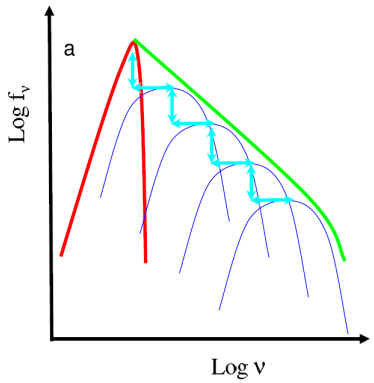

Figure 2: Figure showing how adding together many blackbody spectra can produce a power law spectrum. The spectrum of the seed photons is in red, while the total spectrum, in green, follows a power law, but drops off at high and low energies. (Figure from Done 2010)

(inverse Compton scattering), the photons gain energy from the electrons. Seed photons from the disc enter the hot flow, and are up-scattered by hot electrons. Initially, the seed photons follow a blackbody spectrum. If all the photons scatter once, the scattered photons will retain a blackbody spectrum, but shifted to a higher energy. These scattered photons can then go on to be scattered again, multiple times, and each time gaining more energy (up to the limit of the energy of the electrons), and each time producing a blackbody spectrum shifted to higher energies. Combining these blackbody spectra together produces the power law spectrum (see figure 2). The slope of this power law depends on both the temperature of the electrons and the optical depth (Done 2010).

Transitions between the spectral states can be explained by moving the truncation radius (DGK07). When the truncation radius is large, the inner hot flow is large, and dominates the spectrum. As the truncation radius decreases, the inner flow gets smaller, and there can be an overlap with the outer disc. This overlap means that more seed photons from the outer disc are intercepted by the inner flow, and are up-scattered, resulting in the cooling of the inner flow, and a softer power law (figure 1b).

1.3 Variability

X-ray sources are intrinsically variable on much shorter timescales than their spectral evolution. A satellite observing an X-ray source will sample the source in evenly spaced time bins, creating a plot of count rate (number of photons detected per second) against time, called a light curve. A mathematical function can be decomposed into a series of sine waves of different frequencies. By taking a Fourier transform, the contribution to the function from each frequency can be determined (see section 2). Taking the Fourier transform of a light curve, and squaring it, gives the power spectrum. The power spectra of X-ray sources display the same basic shape. At low frequencies, the power

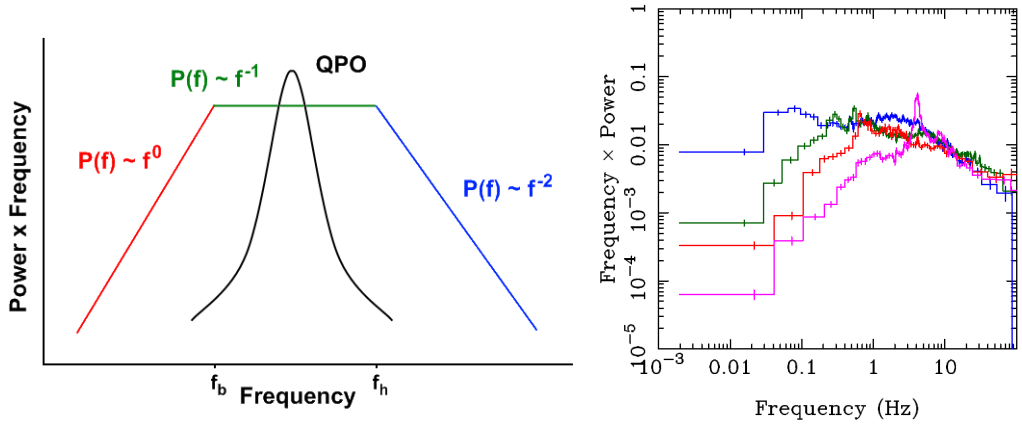

Figure 3: Left: sketch of a typical power spectrum, with power × frequency plotted against frequency, showing the low frequency break, f_b, high frequency break, f_h, and a quasi-periodic oscillation (QPO) which often follows a Lorentzian. Both axes are logarithmic. Right: Power spectra of the source XTE J1550-564 during a transition. As the spectrum softens, the power spectra become narrower, while the power at high frequency stays fairly constant. (Figure from DGK07)

spectrum, $P(\nu)$, is constant (ie proportional to ν^0). As the frequency increases, there is a low frequency break, ν_b, and $P(\nu) \propto \nu^{-1}$ up to a high frequency break, ν_h, beyond which $P(\nu) \propto \nu^{-2}$ (figure 3a). For a typical source in the low/hard state, $\nu_b \sim 0.1$ Hz, and $\nu_h \sim 10$ Hz.

During a transition, ν_h remains roughly constant. However, ν_b can vary. As the source transitions from the low/hard to high/soft state, ν_b increases, causing the power spectrum to narrow (figure 3b). This can also be explained by a truncated disc model. As the truncation radius moves inwards, its characteristic timescales decrease. Fluctuations at this radius propagate inwards through the hot flow, down to the last stable orbit. Fluctuations with a frequency greater than the characteristic frequency of the last stable orbit are damped, and the highest frequency remains constant. (DGK07)

The MRI is the physical origin of the disc viscosity. The MRI is variable, and produces large fluctuations in all quantities (Krolik & Hawley 2002). Fluctuations in viscosity will result in fluctuations in the mass accretion rate and this causes the variability seen in the power spectrum.

1.4 Quasi-Periodic Oscillations

Sources often display a narrow peak in the power spectrum between the low and high frequency breaks. This peak often follows a Lorentzian, and is seen at low frequencies in the low/hard state, increasing in frequency through the intermediate state up to the very high state, where it is strongest, at a frequency 6 - 10 Hz (Ingram, Done & Fragile 2009). This is called a quasi-periodic oscillation (QPO). This is often accompanied by a second smaller peak at a higher frequency (the second harmonic), or at a lower frequency

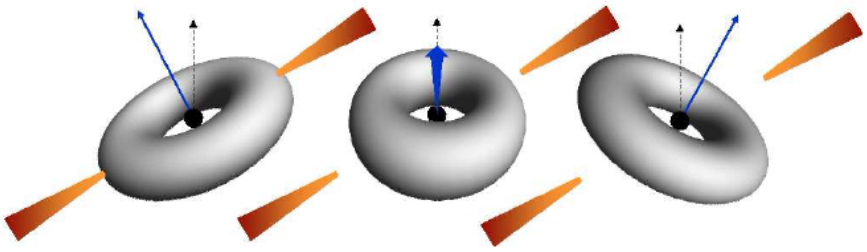

Figure 4: Diagram showing the geometry of the accretion disc. The angular momentum vector of the inner hot flow (blue arrow) is misaligned with the angular momentum of the black hole (black arrow), causing the inner flow (grey) to process, while the outer part of the disc (red) remains in its initial alignment. (Diagram from Ingram, Done & Fragile 2009)

(sub-harmonic), or sometimes both. QPOs are weak or non-existent in the high/soft state.

QPOs have been known about for many decades, but until recently, there were no quantitative models which could explain them. Potential models fell into two categories: misalignment of the black hole spin and accretion flow, and wave modes of the accretion flow. However, this second category has now been ruled out, since fluctuations caused by the MRI would destroy any coherent wave modes in the flow (Henisey et. al. 2009).

The current best explanation for the occurrence of QPOs is Lens-Thirring precession of the inner hot flow (Ingram & Done 2011). The black hole and hot flow are both spinning, but their angular momentum vectors are misaligned (figure 4). This causes the precession of the hot flow due to relativistic frame dragging. Numerical simulations have calculated QPO frequencies which match observed frequencies (Fragile et al 2007).

The truncated disc model also explains the shift in frequency of the QPO as a source moves into a different spectral state. The precession frequency is dependent on the size of the flow (Fragile et al 2007). If the flow is large, the frequency is small. In the low/hard state, the inner flow is large, and thus a QPO is seen at low frequencies. In the very high state, the inner flow gets smaller, so the QPO frequency increases. In the low/hard state, the inner flow is very small or non-existent, and thus a QPO is not seen.

1.5 Reflection & Absorption

Depending on the geometry of the disc, photons emitted by the inner hot flow can illuminate the outer cool disc, leading to absorption features in the X-ray spectrum. The most prominent feature is the fluorescent iron Kα line at 6.4-6.7 keV (depending on the ionisation state), with a smeared absorption edge at $\sim$7.1 keV (Basko, Sunyaev & Titarchuk, 1974). Iron atoms absorb a photon which ejects an electron from the inner K-shell. This is then filled by the transition of an outer electron, producing this characteristic emission line. The Kα line is intrinsically narrow. However, effects such

as Doppler shifts, relativistic beaming and gravitational redshift lead to the broadening of the line.

When the system is in the high/soft state, the inner flow is small, and there can be a large overlap between the two components of the disc. This should lead to stronger reflection features than the low/hard state, where the truncation radius as large, and there is little overlap. As the inner flow precesses, it moves out of, then back into alignment with the outer disc. When the flow is out of alignment, a large area of the disc will be illuminated, leading to strong reflection features. When both components are in alignment, a much smaller number of photons illuminate the disc, and a lot less reflection will occur.

2 Fourier Techniques

2.1 The Fourier Transform

For an infinite, continuous function, the Fourier transform (FT), and its inverse, are defined respectively as,

$$X(\nu) = \int_{-\infty}^{\infty} x(t) e^{2\pi i \nu t} \mathrm{d}t \tag{5}$$

$$x(t) = \int_{-\infty}^{\infty} X(\nu) e^{-2\pi i \nu t} \mathrm{d}\nu \tag{6}$$

However, in reality, a light curve is not a continuous function, but is discretely sampled in evenly spaced time bins. For a light curve of total duration T, split into N bins of time $\delta t = T/N$, the discrete Fourier transform (DFT) is defined as,

$$X(\nu_j) = \sum_{k=0}^{N-1} x_k e^{2\pi i j k/N} \tag{7}$$

where x_k is the number of photons detected in bin k, and $\nu_j = j/T$ ($j = 0, 1, \ldots \frac{N}{2}$). The frequency increases in steps of $\delta\nu = 1/T$ up to a maximum frequency, $\nu_{N/2} = \frac{1}{2}N/T$, called the Nyquist frequency, which is equal to half the sampling frequency ($1/\delta T$).

The DFT of a time series can be computed easily using a fast Fourier transform (FFT) algorithm.

2.2 The Power Spectrum

Parseval's theorem states,

$$\int_{-\infty}^{\infty} |x(t)|^2 \, \mathrm{d}t = \int_{-\infty}^{\infty} |X(\nu)|^2 \, \mathrm{d}\nu \tag{8}$$

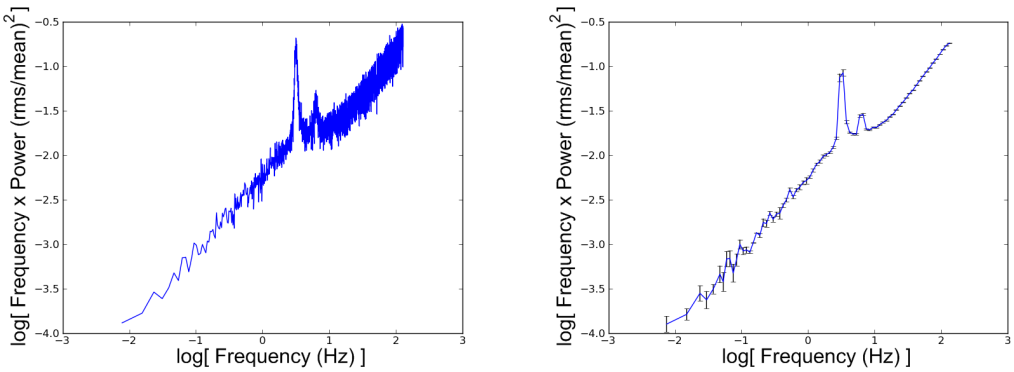

Figure 5: Left: normalised power spectrum of the source XTE J1550-564 before rebinning. Error bars are omitted for clarity. Right: the same power spectrum after rebinning, with 20 bins per decade. In both plots, the white noise has not been subtracted.

that is, the total energy of a signal is equal to the area under the square of the Fourier transform. The power spectral density (PSD) is defined as $|X(\nu)|^2$.

$$P(\nu_j) = A\,|X(\nu_j)|^2 \tag{9}$$

where A is the chosen normalisation (see section 2.2.1)

Generally, the power spectrum spans several orders of magnitude in frequency, thus it is plotted logarithmically. In each frequency bin, the total variability is $P(\nu)d\nu = P(\nu)\nu d\nu/\nu = \nu P(\nu) d\log\nu$. Therefore $\nu P(\nu)$ is plotted, rather than simply $P(\nu)$ (Done 2010). Since the frequency bins are evenly spaced, on a logarithmic power spectrum, points at high frequency become squashed together, and so power spectra plotted on logarithmic axes need to be smoothed. The simplest way to do this is to rebin the power spectrum in evenly space logarithmic bins, averaging together all the values in each bin (figure 5).

2.2.1 Normalisation

We normalise the power spectrum such that,

$$\int P(\nu)\mathrm{d}\nu = \left(\frac{\sigma}{I_0}\right)^2 \tag{10}$$

where $\sigma^2 = \frac{1}{N}\sum(I - I_0)^2$ is the variance, and I_0 the mean count rate. This normalises the power spectrum such that it is in units of square fractional rms. This choice of normalisation means that integrating $P(\nu)$ over a certain frequency range gives the rms fractional variability (Done 2010).

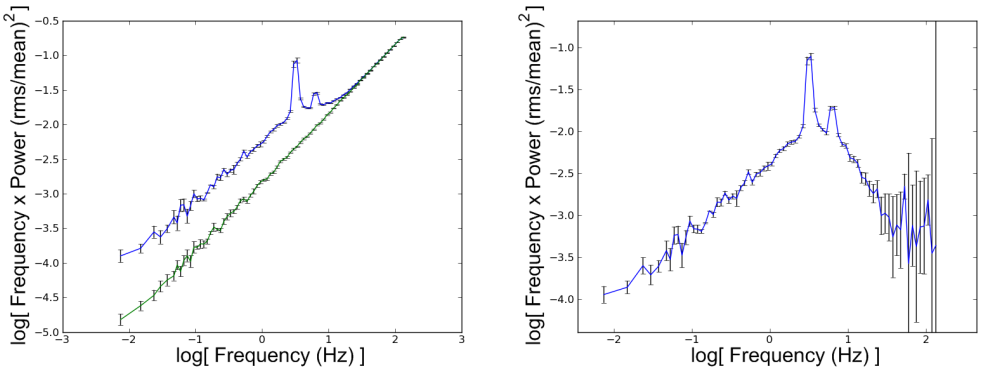

Figure 6: Left: Power spectrum as in Figure 5 (blue), with the white noise level (green). Right: the same power spectrum after white noise subtraction.

2.2.2 Noise

At high frequencies, the power spectrum becomes dominated by white noise. This noise is produced by the detector. Since the detector is counting small numbers of photons in each time bin, this white noise follows Poisson statistics. The power of the white noise is constant and independent of frequency, and thus dominates where the signal is at its lowest. If the noise is not removed, $\nu P(\nu)$ increases at high frequencies.

One way to correct the noise is to simply calculate the average power at high frequencies, where noise dominates, and then subtract this from every point in the power spectrum.

Another way to remove white noise, which we will use, is to generate random numbers. The total light curve, $I(t) = I_{src}(t) - I_{bgd}(t)$, where $I_{src}(t)$ is the count rate from the source, and $I_{bgd}(t)$ is the background count rate. $I_{src}(t)$ and $I_{bgd}(t)$ both follow Poisson statistics, but $I(t)$ does not. The source and background count rates are converted into numbers of photons by dividing by the width of the time bin, and the mean number of photons per bin, $\langle N \rangle_{src}$ and $\langle N \rangle_{bgd}$ are calculated. For the source, a random number, which follows a Poisson distribution, is generated with mean = variance = $\langle N \rangle_{src}$, and a random number for the background is generated with mean = variance = $\langle N \rangle_{bgd}$. These are then added together and converted back into a count rate, creating a new light curve. The power spectrum of this, $P_{noise}(\nu)$, gives the white noise level of the power spectrum, and is subtracted to give the white noise corrected power spectrum, $P_{cor}(\nu) = P(\nu) - P_{noise}(\nu)$ (figure 6).

2.3 The Rms Spectrum & Frequency Resolved Spectroscopy

The lightcurve produced by a satellite can be split into different energy bands. In general, the power spectra obtained from each energy band is different. This variation as a function of energy can be shown in an rms (root-mean-square) spectrum (figure 7). The power spectrum is normalised such that integrating over a frequency range gives

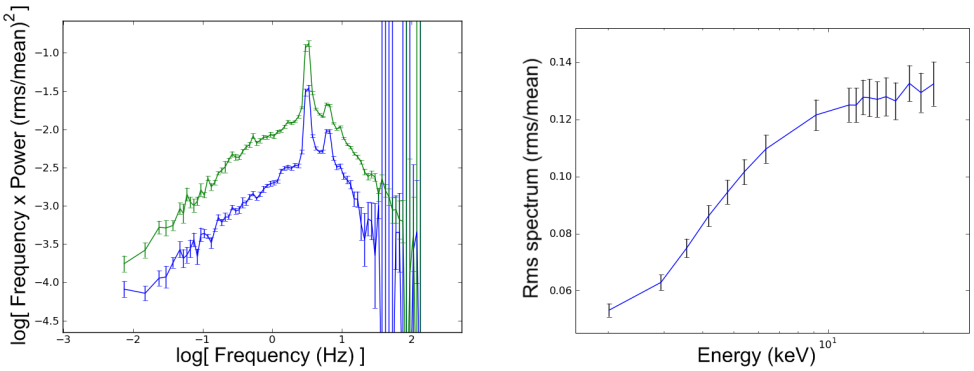

Figure 7: Left: Normalised, white noise subtracted power spectra at energies of 3 keV (blue) and 11 keV (green). The green power spectrum is shifted up, showing that there is more variability at this energy. Right: The rms spectrum, which shows the fractional variability as a function of energy.

the square fractional variability. Rms spectra are constructed from,

$$r(E) = \sqrt{\int_{\nu_1}^{\nu_2} P(\nu, E) \mathrm{d}\nu} \tag{11}$$

We can also construct the frequency resolved spectrum,

$$S(E) = R(E) r(E) \tag{12}$$

where $R(E)$ is the average count rate (Revnivtsev et al, 1999).

2.4 The Cross-Spectrum

In general, the Fourier transform of a light curve, $l(t)$, is a complex function, $L(\nu) = |L(\nu)| \exp(i\Phi(\nu))$, where Φ is a frequency-dependent complex phase. This phase information is lost in the power spectrum. $P(\nu) = L^*(\nu)L(\nu) = |L(\nu)|^2 \exp(-i\Phi(\nu) + i\Phi(\nu)) = |L(\nu)|^2$. This phase information can be retained by computing a cross-spectrum.

The cross-correlation of two time series, $s(t)$ and $h(t)$ is defined as,

$$s \star h = \int_{-\infty}^{\infty} s(\tau) h(t + \tau) \mathrm{d}\tau \tag{13}$$

The Fourier transform of the cross-correlation, $\mathcal{F}(s \star h)$, is equal to the product of the FT of one of the time series, multiplied by the complex conjugate of the other,

$$\mathcal{F}(s \star h) = H^*(\nu) S(\nu) \tag{14}$$

The cross-spectrum is defined as the Fourier transform of the cross-correlation, $C(\nu) \equiv \mathcal{F}(s \star h)$. The Fourier transform of both light curves contains a complex phase,

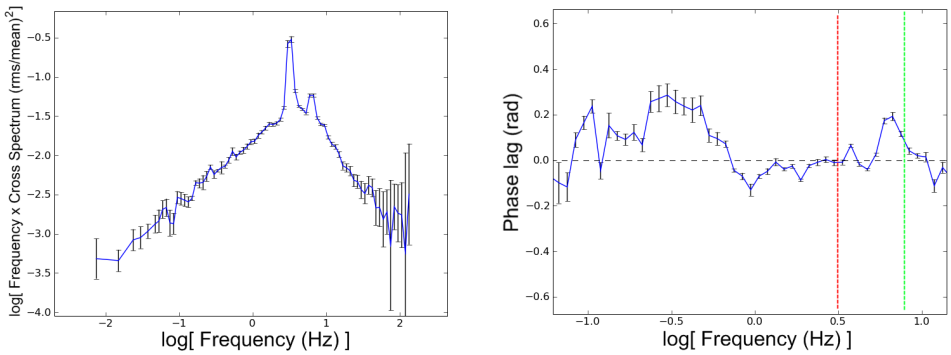

Figure 8: Left: The magnitude of the cross spectrum between the 2-3 keV and 8-13 keV energy bands. Right: phase lag between the 2-13 keV and 13-16 keV energy bands. The frequency of the QPO and harmonic are indicated by the red and green lines respectively.

and therefore $C(\nu) = |H(\nu)| \exp(-i\Phi_H(\nu)) \, |S(\nu)| \exp(i\Phi_S(\nu)) = |H(\nu)| \, |S(\nu)| \exp(i(\Phi_S - \Phi_H)) = |H(\nu)| \, |S(\nu)| \exp(i\Delta)$, where we define the phase difference $\Delta \equiv \Phi_S - \Phi_H$.

By analogy to the power spectrum, the cross-spectrum is normalised such that,

$$\int C(\nu) \mathrm{d}\nu = \left(\frac{\sigma_h}{h_0}\right) \left(\frac{\sigma_s}{s_0}\right) \tag{15}$$

where σ_h and σ_s are the standard deviations of the two time series, and h_0 and s_0 are their means. The phase lag, $\Delta(\nu)$, is the complex argument of the cross-spectrum, which is independent of the normalisation. This can be converted into a time-lag by dividing by $2\pi\nu$.

3 Data Analysis

3.1 Rossi X-ray Timing Explorer (RXTE)

The Rossi X-ray Timing Explorer (RXTE) was a NASA satellite designed to study high energy astronomical X-ray sources, such as black holes and neutron stars, through their X-ray timing properties. The satellite was launched in December 1995, and was in operation for 16 years, before finally being decommissioned in January 2012.

The RXTE consists of three instruments. The All Sky Monitor (ASM) consists of three wide angle cameras, which scan 80% of the sky each orbit, allowing new phenomena to be detected quickly. The Proportional Counter Array (PCA) consists of 5 proportional counters with an energy range of 2 - 60 keV. The High Energy X-ray Timing Experiment (HEXTE) is composed of 2 clusters of phoswich scintillation detectors, and has an energy range of 15 - 250 keV.

3.1.1 Data Extraction

We downloaded publically available RXTE data which was taken using the PCA. Light curves were obtained for each energy bin from the data in binned mode, using the standard FTOOLS software. Background estimates were created, and subtracted from the source light curves. A program was written in Python to compute the PSDs, rms spectra, cross spectra and lags from these light curves.

3.1.2 Dead Time

The PCA detectors consist of layers of Xenon. As an incident X-ray photon enters the detector, it is detected by ionising the Xenon, which will then recombine ready for another detection. However, there is a small period of time before the Xenon recombines, during which an incident photon cannot be detected. This is called the dead time, and results in the PCA producing a smaller count rate than there actually is. The dead time can be corrected for, as described in Revivtsev, Gilfanov & Churazov, 2000. However, dead time is only important for the brightest X-ray sources. Hereafter we assume that dead time effects are negligible and will not correct for them.

3.2 XTE J1550-564

XTE J1550-564 is an X-ray nova and black hole candidate that was discovered using RXTE's ASM. The source underwent a bright outburst between September 1998 and May 1999. During this outburst, detailed observations were made by RXTE. We use data taken from this outburst, in September 1998, with observation IDs 30188-06-01-00 up to 30188-06-11-00 The observation 30188-06-02-00 is omitted because there was no data in the binned data mode. In addition, we use data with observation ID 30191-01-01-00, and every fifth observation from 30188-06-05-00 up to 30188-06-30-00. Hereafter we will number these observations from 1 to 18, as shown in table 1.

To calculate power spectrum of each light curve, the light curve is first split into 128 s sections. Any sections with gaps in the data are rejected. These gaps are where the satellite is not pointed at the source, and there is no way to interpolate the data. Power spectra are then computed for each section, using the Numpy FFT algorithm, and are then averaged together to obtain the total PSD (figure 9). Averaging together many power spectra is done, rather than calculating one single PSD using the whole set of data, as it reduces the errors. It also reduces the time taken to compute the PSD, as the time required to calculate the FFT of a set of data is proportional to $N \log N$, where N is the number of data points (van der Klis, 1988). White noise is then subtracted, and the PSD is rebinned as described in section 2. Sometimes, after subtracting white noise, some of the points in the PSD are negative, which is a problem when plotting logarithmically. This is dealt with by setting these points to an arbitrary small positive value. Errors for such points are very large.

Rms spectra are obtained by firstly computing the power spectra in each energy band for each observation. Each PSD is integrated over the QPO frequency (FWHM)

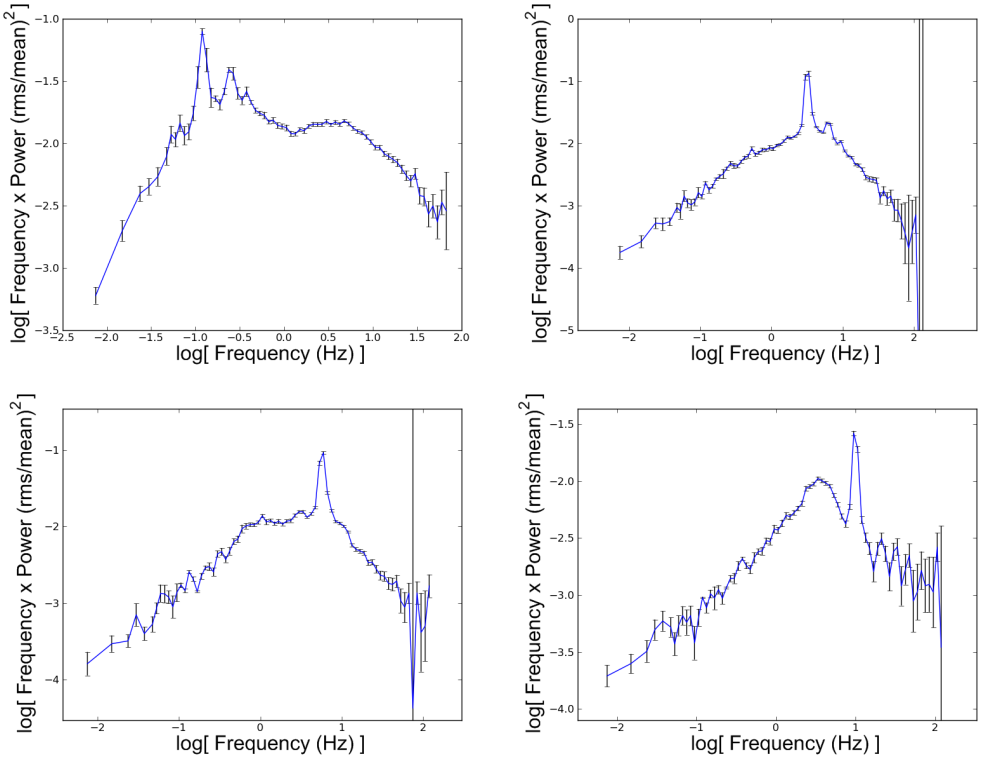

Figure 9: A selection of power spectra of XTE J1550-564. Top row: observation numbers 3 (left) and 7 (right). Bottom row: numbers 12 (left) and 13 (right).

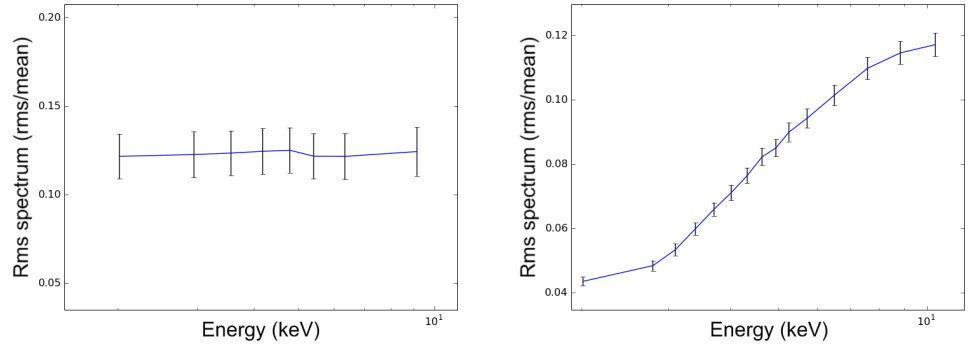

Figure 10: Left: flat rms spectrum of observation 1. Right: rms spectrum of observation 14, which rises as a function of energy.

Number	Observation ID	Data Mode	QPO Freq (Hz)
1	30188-06-01-00	B_4ms_8A_0_35_H	0.22-0.45
3	30188-06-03-00	B_8ms_16A_0_35_H_4P	0.09-0.16
4	30188-06-04-00	B_4ms_8A_0_35_H	1.2-2.0
5	30188-06-05-00	B_4ms_8A_0_35_H	2.0-3.5
6	30188-06-06-00	B_4ms_8A_0_35_H	2.5-5.0
7	30188-06-07-00	B_4ms_8A_0_35_H	2.5-4.5
8	30188-06-08-00	B_4ms_8A_0_35_H	2.5-4.5
9	30188-06-09-00	B_4ms_8A_0_35_H	2.8-5.0
10	30188-06-10-00	B_4ms_8A_0_35_H	2.0-3.5
11	30188-06-11-00	B_4ms_8A_0_35_H	3.0-5.6
12	30191-01-01-00	B_4ms_8A_0_35_H	4.5-8.0
13	30191-01-05-00	B_4ms_8B_0_49_H	8.0-11.0
14	30191-01-10-00	B_8ms_16A_0_49_H	3.3-4.5
15	30191-01-15-00	B_4ms_8A_0_35_H	3.5-5.0
16	30191-01-20-00	B_4ms_8A_0_35_H	2.5-3.6
17	30191-01-25-00	B_4ms_8A_0_35_H	3.8-5.0
18	30191-01-30-00	B_4ms_8A_0_35_H	5.5-8.0

Table 1: The observation IDs, data modes and QPO frequencies of the observations of XTE J1550-564

using the Scipy Simpsons algorithm. Channel numbers were converted to energies using the energy-channel conversion table, which is available online[1] (figure 10).

Cross spectra were computed in the same way as the power spectra, with the light curves split into 128 s sections, discarding any with gaps, and calculating the average normalised cross spectrum. As with the PSDs, we plot frequency $\times$ cross spectrum. There is no need to correct for the white noise when using cross spectral techniques. Both light curves contain noise, but this is uncorrelated between the two energy bands. The cross spectrum, which is the Fourier transform of the cross correlation only shows the correlated signal between the two energy bands. Since the white noise in uncorrelated, it cancels out.

Phase and time lags were then obtained from cross spectra between the 2-3 keV and 8-13 keV energy bands (figure 11). For observation 7, we also used some data which went to higher energies, and calculated the phase lag between the 2-13 keV and 13-16 keV energy bands (figure 8). The phase lag was calculated between this hard band and all the softer energy bands, and were then averaged together to reduce the errors.

[1]http://heasarc.nasa.gov/docs/xte/e-c_table_e03v04.html

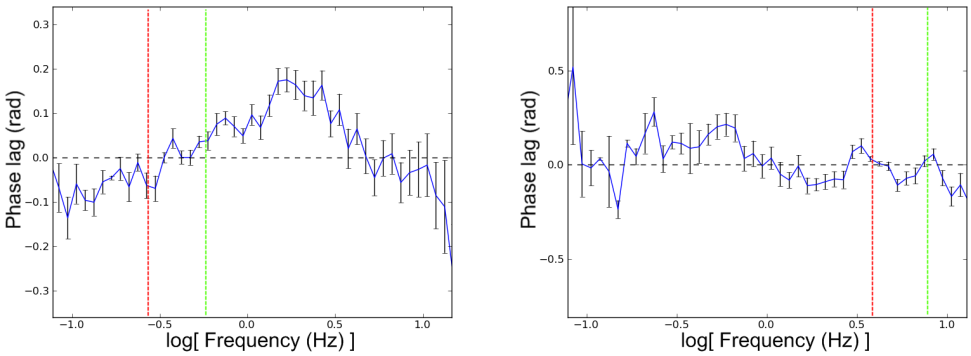

Figure 11: Phase lags between the 2-3 keV and 8-13 keV energy bands for observations 1 (left) and 11 (right). The frequency of the QPO and harmonic are indicated by the red and green lines respectively.

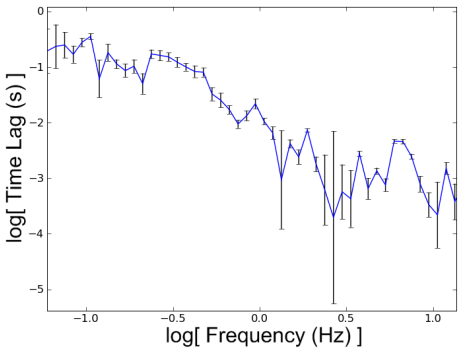

Figure 12: Time lag of observation 7.

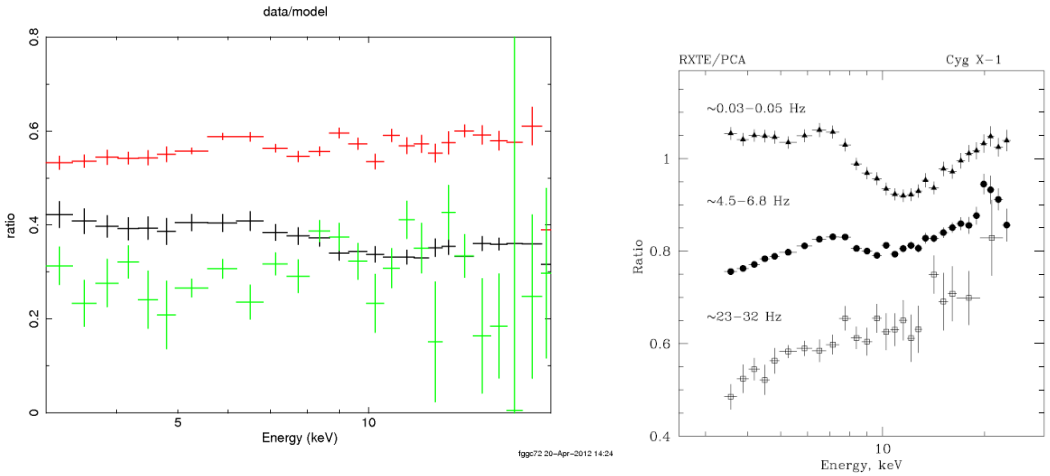

Figure 13: Left: ratio of the frequency resolved spectra of Cyg X-1 in the frequency bands 0.03-0.05 Hz (black), 4.5-6.8 Hz (red) and 23-32 Hz (green) with a power law model of index 1.8. Right: the same plot as shown in Revnivtest, Gilfanov & Churazov 1999.

3.3 Frequency Resolved Spectroscopy

Cyg X-1 is an X-ray binary system, and the brightest X-ray source in the constellation of Cygnus. Using data with observation ID 10238-01-08-00, in data mode B_4ms_8A_0_35_H, we extract the light curves for each energy, compute the PSDs, and hence calculate the frequency resolved spectrum in the frequency bands 0.03-0.05 Hz, 4.5-6.8 Hz and 23-32 Hz, as shown in Revnivtsev, Gilfanov & Churazov, 1999. The frequency resolved spectra are plotted in XSPEC. We compare our results to make sure our program is working correctly (figure 13).

We then calculate the frequency resolved spectrum of XTE J1550-564, using the same observations as previously. For each observation, we integrate the power spectra over the QPO frequency, where the upper and lower limits are determined from the full width at half maximum (FWHM). The QPO frequencies are summarised in table 1. For each observation, the time averaged spectrum is opened in XSPEC, and data below 3 keV and above 20 keV is ignored. This is fit to the model TBABS×(DISKBB + REFLECT×NTHCOMP + GAUSSIAN), where TBABS calculates the absorption from the ISM, DISKBB is the blackbody component from the disc, REFLECT models the reflection from neutral material, NTHCOMP is a thermally Comptonised continuum, and GAUSSIAN is a Gaussian, used to model the relativistically broadened Iron Kα emission line. The QPO spectrum is then imported into XSPEC, and data below 3 keV and above 13 keV is ignored. The fit is then renormalised and fit to the QPO spectrum, with all the shape parameters frozen. The QPO spectrum is then also fit to a simple model of TBABS×NTHCOMP. These fits for observations 1 and 7 are shown in figures 14 and 15.

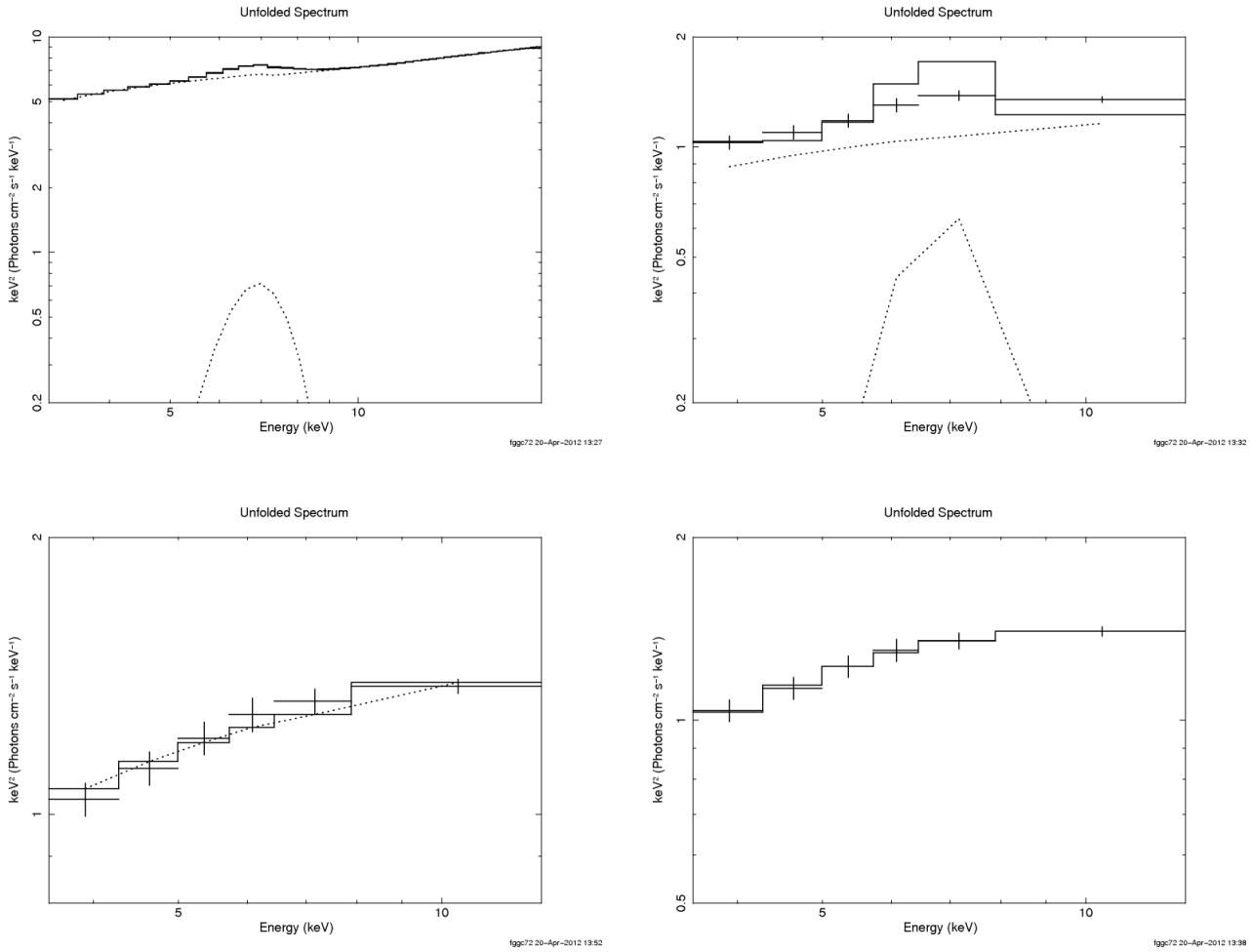

Figure 14: Fits to observation 1. Top left: Fit to the time averaged spectrum. The upper dashed line is the Compton component, and the lower dashed line is the relativistically broadened Iron Kα line, modelled as a Gaussian. Top right: Fit to the QPO spectrum. The shape parameters of the fit to the total spectrum have been frozen, and then be renormalised for the QPO spectrum. Bottom left: Fit to the QPO spectrum after the Gaussian normalisation has been set to zero. Bottom right: Fit to the QPO spectrum consisting of only the Comptonisation component.

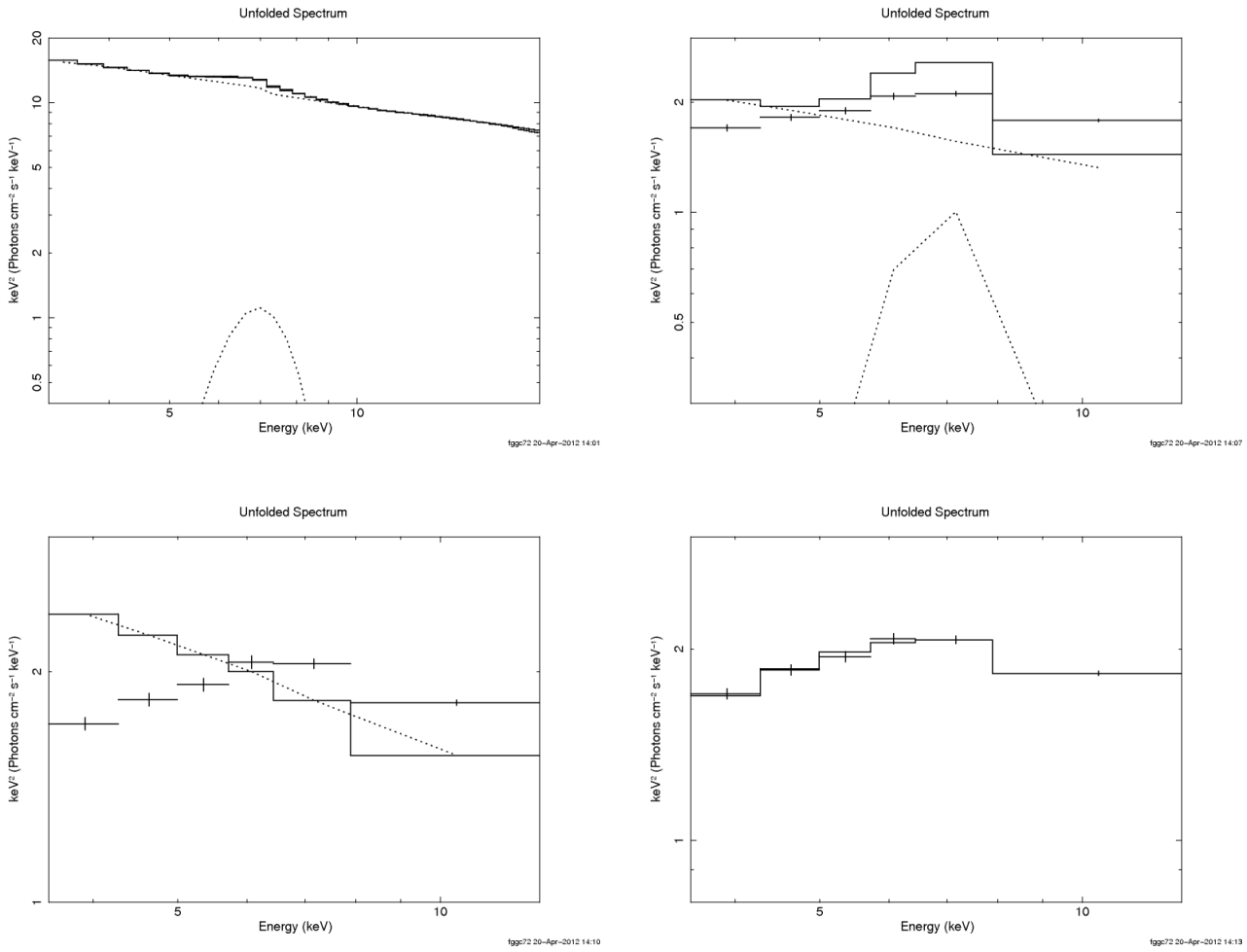

Figure 15: The same as figure 14, but for observation 7.

4 Discussion

All observations of XTE J1550-564 show strong QPOs in the PSDs, and most show the second harmonic (figure 9). In the later observations, the sub harmonic can also be seen. Observations 1 and 3 have the lowest frequency QPOs, of a few tenths of a Hz. This increases to a maximum frequency of ∼10 Hz for observation 13. The rms spectra of observations 4 to 18 are all increasing as a function of energy, whereas the rms spectra of observations 1 and 3 are flat (figure 10). An increase in fractional variability with energy is typically seen in the high/soft and very high states, while decreasing or flat rms spectra are seen in the low/hard state (Gierliński & Zdziarski, 2005). Since QPOs are not observed in the high/soft state, this data is showing that the source is making a transition from the low/hard state, through an intermediate state into the very high state, then back into the intermediate state. The behaviour of the rms spectrum can be described well by Comptonisation in the two-component accretion disc (Gierliński & Zdziarski, 2005).

The shapes of the cross spectra are very similar to the PSDs, displaying a strong peak at the QPO frequency and also its second harmonic. The phase and time lags are plotted between 0.1 and 10 Hz, where the errors are smallest. Positive lags represent a hard lag where the hard (high energy) photons are lagging the soft (low energy photons). Conversely, negative phase lags represent the soft photons lagging hard photons. The phase lags between the 2-3 keV and 8-13 keV show for observation 1 hard lags peaking at ∼2 Hz, with soft lags at low and high frequencies. As the QPO frequency increases, the lags become more complicated. The lags for observation 11 show hard lags at low frequency, with low lags at high frequency, but there are hard lags at the QPO and harmonic (figure 11). The phase lag for observation 7 between the 2-13 keV and 13-16 keV band is a similar shape (figure 8). This shape has been seen in the phase lags of other sources (Reig et al, 2000).

The occurrence of lags between hard and soft photons is very complicated. One might assume that the hard lags are simply produced from the Comptonisation in the inner flow. As cool seed photons enter the hot flow, they are up-scattered, gaining energy each time. This means that the hard photons have spent more time in the Comptonising region, and therefore lag the softer photons. However, the time lags show that at low frequency, the lags are of the order of 0.1 seconds (figure 12). This would require a large extended corona a few light seconds across (Ciu, 1999). This is too large to produce the observed high frequency variability.

The models which can explain the hard lags in a much smaller Comptonising region involve the propagation of fluctuations in the accretion flow (Kotov, Churazov & Gilfanov, 2001). Low frequency fluctuation are produced far from the black hole. These fluctuation then propagate inwards, to smaller radii, where the higher frequency fluctuations are produced. As the fluctuations propagate inwards, the photons gain energy, so hard photons are emitted close to the black hole, and soft photons farther out.

4.1 Frequency Resolved Spectroscopy

4.1.1 Cyg X-1

The frequency resolved spectrum of Cyg X-1 in the 30-50 Hz frequency band shows a peak at ∼6.4 keV, which is the Iron Kα line. Above this energy is a broad absorption edge (figure 13). The peak from the Iron line can also be seen in the 4.5 - 6.8 Hz range. Between 23 and 32 Hz, no reflection features are seen. The errors in the power spectra are large due to white noise, and thus the frequency resolved spectrum in this range has large errors. In general, the errors for all the spectra are large in comparison to the plot from Revnivtsev, Gilfanov & Churazov, 1999. This is due to the fact that only one observation was used. More data would decrease the errors. At high frequencies, the spectra are harder. This is as expected from the propagation model, where the high frequencies are produced close to the black hole where hard photons are emitted, and low frequencies are produced far out where the emitted photons are soft. At these large distances from the black hole, photons from the inner flow can illuminate the disc, producing the reflection features. Close to the black hole, inside the hot flow, there is no reflection.

Another way to reduce the errors in the frequency resolved spectra, other than using more data, would be to use cross-spectral techniques. This would remove the uncorrelated signal from the white noise, and reduce the errors, especially at high frequencies where the noise is large.

4.1.2 XTE J1550-564

The fits to the time averaged spectra of XTE J1550-564 show that the spectra are dominated by the power law from Comptonisation, with a strong, broad emission line at 6.7 keV (figures 14 and 15). The fits are then renormalised over the QPO spectrum, keeping the shape parameters frozen, and we find that they produce a bad fit, with too much reflection. Setting the normalisation of the blackbody and Gaussian components to zero produces reasonable fits to the QPO spectra in observations 1 and 3, but for the rest of the observations, the QPO spectrum is significantly harder.

Fitting only the Compton component straight to the QPO spectrum provides a good fit. This is showing that the inner part of the accretion flow where the QPO is produced has only a Compton spectrum, with no blackbody component, and no reflection. This supports the idea of Lens Thirring precession producing the QPO. As the inner flow precesses, a different projected area is observed, and so the spectrum of the QPO, which is a Compton spectrum, is modulating at the QPO frequency, and only this is seen in the QPO spectrum.

If we consider a geometry where the inner hot flow and the outer disc are misaligned, with the outer disc aligned with the spin of the black hole, then as the inner flow precesses, the area of the disc illuminated remains constant. Therefore the amount of reflection will be constant. However, if the inner flow and outer disc are initially aligned, then as the flow precesses, they will continuously move into and out of alignment. In

Number	Γ (total)	Γ (QPO)	χ^2 (QPO)
1	1.70 ± 0.04	1.72 ± 0.08	0.059
3	1.766 ± 0.003	1.7 ± 0.3	0.381
4	2.120 ± 0.002	1.65 ± 0.04	0.33
5	2.336 ± 0.003	1.71 ± 0.03	0.907
6	2.565 ± 0.003	1.71 ± 0.03	0.908
7	2.545 ± 0.002	1.71 ± 0.04	0.542
8	2.533 ± 0.002	1.71 ± 0.05	0.28
9	2.644 ± 0.003	1.72 ± 0.04	0.085
10	2.392 ± 0.003	1.69 ± 0.06	0.504
11	2.721 ± 0.003	1.70 ± 0.05	0.14
12	2.968 ± 0.005	1.66 ± 0.02	0.708
13	3.302 ± 0.003	1.53 ± 0.02	9.389
14	2.717 ± 0.003	1.79 ± 0.02	2.906
15	2.792 ± 0.004	1.66 ± 0.03	0.602
16	2.560 ± 0.005	1.66 ± 0.03	0.682
17	2.795 ± 0.006	1.00 ± 0.05	683.813
18	3.018 ± 0.005	1.00 ± 0.09	198.42

Table 2: Comparison of Γ for the fits to the total and QPO spectra, and the reduced χ^2 values for the fits to the QPO spectra.

this situation, the area of the disc being illuminated is varying, and thus the reflection component of the spectrum is modulating at the QPO frequency. In this geometry, the QPO spectrum would contain a reflection component. However, in general the amount that the illuminated area changes is small, and would be difficult to observe.

Most of the fits to the QPO spectra have values of reduced $\chi^2 < 1$ (table 2). This shows that the Compton spectrum fits the data well, but either the uncertainties in the QPO spectra have been overestimated, or the model is over-fitting the data. The fits would be improved by having data with a finer energy resolution, extending to higher energy. However, observations 17 and 18 have $\chi^2 \gg 1$, indicating that the Compton model is a very poor fit. This may be due to reflection, but there are too few data points in the QPO spectrum to constrain the parameters in a more complicated model.

The total spectra have very small statistical errors, making the reduced χ^2 values meaningless. For observation 11, a systematic uncertainty of 1% is applied, which gives a meaningful χ^2 value of 0.407. The model does fit the total spectra, but these spectra are softer than the QPO spectra for most of the observations. Since the total spectrum is the spectrum of the outer disc and inner flow, and the QPO is the spectrum of just the inner flow, we would expect the values of Γ to be the same for each. QPO spectra harder than the total spectra could be explained if the plasma heating rate is modulating at the QPO frequency, while the cooling rate is not (Sobolewska & Życki, 2006). However, this may also be due to the model. Despite fitting the data, the

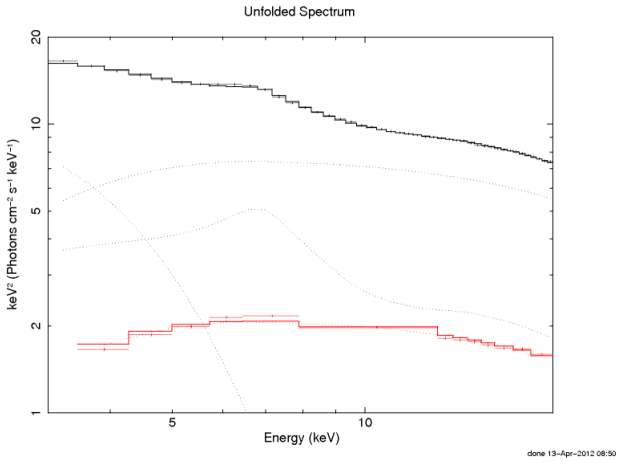

Figure 16: Fit to observation 6 using TBABS×(DISKBB + NTHCOMP + KDBLUR×RFXCONV×NTHCOMP) for the total spectrum (black), and the QPO spectrum with the normalisations of the blackbody and reflection set to zero (red). (Figure by C Done, private email)

blackbody component to the total spectrum is very small, when we would expect it to be much larger. Increasing the contrubution from the disc in the model would be compensated by making the Compton component harder. In our model, we are simply fitting the Iron line to a gaussian. A better model to use would be TBABS×(DISKBB + NTHCOMP + KDBLUR×RFXCONV×NTHCOMP), where RFXCONV models the reflection features, and KDBLUR models the relativistic effects, smoothing the spectrum. This model could not be used in the version of XSPEC we used. This model does produce the expected blackbody component, with a Compton component roughly the same shape as the QPO spectrum (figure 16).

5 Conclusions

For many decades, observations of black hole binary systems have revealed quasi-periodic oscillations (QPOs) in the power spectra. Recently, a model has been proposed which can provide a quantitative explanation for their origin. The accretion disc is comprised of two components: the thin outer disc, and the hot, thick, inner flow. If the inner flow is misaligned with the spin of the black hole, it can precess due to the Lens Thirring effect, with is caused by relativistic frame dragging. The frequency of the QPO depends on the size of this inner flow.

We analyse data taken with the Rossi X-ray Timing Explorer (RXTE) of the X-ray source XTE J1550-564 during its outburst in 1998. We calculate the power spectral density (PSD) for several observations during this outburst. The PSDs show a strong QPO, initially at a low frequency of $\sim$0.1 Hz, which rises up to a high frequency of $\sim$10

Hz, before decreasing in frequency again. Rms spectra, which show the variability as a function of energy, are flat at low QPO frequencies, but rise with energy for higher QPO frequencies. This shows that the source is transitioning from the low/hard state, where the energy spectrum is dominated by a hard power law from the inner flow, through an intermediate state into the very high state, where the spectrum has both a strong blackbody component from the disc and a strong soft power law, then back into the intermediate state.

Cross-spectra are calculated between the 2-3 keV and 8-13 keV energy bands, and are used to determine the phase lags between the hard and soft photons. When the QPO frequency is small, hard lags are seen at ~ 2 Hz, with soft lags at low and high frequencies. For higher QPO frequencies, there are hard lags at low frequencies, and soft lags at high frequencies, with peaks at the QPO and harmonic. The origin of lags between the soft and hard photons can be explained by a propagating fluctuation model.

The frequency resolved spectrum is computed for the X-ray source Cyg X-1, using data from the RXTE with observation ID 10238-01-08-00 in the frequency bands 0.03-0.05 Hz, 4.5-6.8 Hz, and 23-32 Hz. The low frequency band reveals a broad peak at ~ 6.4 keV, with a smeared absorption edge at ~ 7.1 keV. This is from the Iron Kα line, and is produced from photons in the inner flow reflecting off the outer disc. Higher frequency bands are harder, and show less reflection. High frequency fluctuations are produced close to the black hole, where the photons are harder, as expected from the propagating fluctuation model. The errors in our frequency resolved spectra are large, since we only use one set of data. Using cross-spectral techniques would reduce these errors, especially for the high frequency band, since uncorrelated white noise is removed.

Frequency resolved spectra are then calculated around the QPO frequency for the same observations of XTE J1550-564 as before. We then fit a model to the time averaged spectrum in XSPEC, which consists of a blackbody component, a Compton component, and the reflection modelled as a Gaussian. We also fit a model consisting of just the Compton component to the QPO spectra. For most observations, the QPO spectra can be described well by a Compton power law. This supports the Lens Thirring model of the QPOs, as the precessing inner flow produces only this Compton power law, with no blackbody component. However, for some observations, this is not a good fit, and may be due to reflection. We find that the fits to the time averaged spectra lack the blackbody component, which is expected from the disc, and the Compton component is much softer than for the QPO spectrum. This is due to our model. Using a relativistically smeared reflection model, instead of the Gaussian, gives spectra which do contain a blackbody component, with a Compton component the same shape as the QPO spectrum.

This work could be extended by analysing more data, up to a higher energy, and using this improved model. More data would help to constrain the parameters of the model. Cross-spectral techniques could also be used to reduce the uncertainties in the frequency resolved spectra.

Acknowledgements

I would firstly like to thank Chris Done for her help and support throughout the academic year. I would also like to thank Matt Middleton for showing me how to extract the RXTE data, and then for helping when it didn't work! Finally, I would like to thank Adam Ingram for explaining, and helping me to understand the cross-spectral techniques, and also the truncated disc model and Lens Thirring precession.

References

[ ] Ciu W., 1999, ApJ, 524, L59

[ ] Basko M., Sunyaev R., Titarchuk L., 1974, A&A, 31, 249

[ ] Done C., 2010, arXiv, arXiv:1008.2287

[ ] Done C., Gierliński M., Kubota A., 2007, A&ARv, 15, 1

[ ] Fragile P. C., Anninos P., Blaes O. M., Salmonson J. D., 2007, ApJ, 668, 417

[ ] Gierliński M., Zdziarski A. A., 2005, MNRAS, 363, 1349

[ ] Hawley J. F., Gammie C. F., Balbus S. A., 1995, ApJ, 440, 742

[ ] Henisey K. B., Blaes O. M., Fragile P. C. Ferreira B. T., 2009, ApJ, 706, 705

[ ] Ingram A., Done C., Fragile C. P., 2009, MNRAS, 397, L101

[ ] Ingram A., Done C., 2011, MNRAS, 415, 2323

[ ] Kolb U., 2010, 'Extreme Environment Astrophysics', Cambridge University Press, pages 72-75

[ ] Kotov O., Churazov E., Gilfanov M., 2001, MNRAS, 327, 799

[ ] Krolik J. H., Hawley J. F., 2002, ApJ, 573, 754

[ ] van der Klis, M., 'Timing Neutron Stars', NATO ASI Series C, Vol. 262, p. 27

[ ] Reig P., Belloni T., van der Klis M., Méndez M., Kylafis N. D., Ford E. C., 2000, ApJ, 541, 883

[ ] Revnivtsev M., Gilfanov M., Churazov E., 1999, A&A, 347, L23

[ ] Revnivtsev M., Gilfanov M., Churazov E., 2000, A&A, 363, 1013

[ ] Shakura N. I., Sunyaev R. A., 1973, A&A, 24, 337

[ ] Sobolewska M. A., Życki P. T., 2006, MNRAS, 370, 405

[ ] Wilkinson T., 'Using novel timing techniques to explore the X-ray variability of Black-Hole and Neutron star X-ray Binaries', 2011, Thesis (PhD), University of Southampton

Appendix

Python code

power.py

```
"""
Program which takes the Fourier transform, then plots the power spectrum
of a light curve
"""

import numpy as N
import pylab as P
import scipy.integrate as I
import random as R
from scipy.stats import poisson


def power_spectrum(intensity, timestep):
    """
    Function which calculates the power spectrum of a light curve,
    returning arrays of the frequency and square of the Fourier
    transform
    """
    power = abs(N.fft.fft(intensity))**2
    freq = N.fft.fftfreq(len(intensity), timestep)
    freq, power = positive(freq, power)     # remove negative frequencies

    return freq, power


def positive(freq, power):
    """
    Returns an array of positive, non-zero frequencies and and array of
    the corresponding power
    """
    i = abs((len(freq)+1)/2.) # index of first -ve value in freq array
    return freq[1:i], power[1:i]


def normalise(freq, fP, fP_noise, fP_err, intensity):
    """
    Function which normalises the power spectrum
    """
```

```
    power = (fP + fP_noise) / freq

    # integrate using scipy's Simpson's routine
    integral = I.simps(power, x=freq)

    sq_rms_var = squared_rms_variability(intensity)
    norm_constant = sq_rms_var / integral

    fP *= norm_constant
    fP_err *= norm_constant

    return fP, fP_err


def squared_rms_variability(intensity):
    """
    Function which calculates the squared total rms variability of
    the light curve
    """
    I_0 = N.average(intensity)
    sigma = N.std(intensity)

    return (sigma/I_0)**2


def split_power_spectrum(time, intensity, delta_t):
    """
    Function which splits the light curve, finds the individual power
    spectra, then averages them.
    """
    x = round(128./delta_t)
    intensity = bin_128s(time, intensity, delta_t)
    size = len(intensity)    # length of total light curve

    n_split = size / x

    power = N.zeros(((x/2)-1)*n_split) # empty power array
    intensity = intensity.reshape(n_split, x)
    power.shape = (n_split, abs((x/2)-1))

    for n in range(int(n_split)):
        # calculate power spectra
```

```
        freq, power[n] = power_spectrum(intensity[n],delta_t)

    # average seperate power spectra
    fP, fP_error = average_power_spectrum(freq, power, size, n_split)

    return freq, fP, fP_error


def average_power_spectrum(freq, power, size, n_split):
    """
    Function which averages the power spectra and calculates the error
    on the mean
    """
    fP = N.zeros(abs(((size/n_split)-1)/2))         # empty array of fP(f)
    fP_error = N.zeros(abs(((size/n_split)-1)/2))   # empty array of errors

    for i in range(len(fP)):
        #average fP(f)
        fP[i] = N.average(power[:,i]) * freq[i]

        # error on the mean
        fP_error[i] = N.std(power[:,i]*freq[i])/N.sqrt(n_split)

    return fP, fP_error


def log_bin(freq, fP, error, bin_size):
    """
    Function which calculates the average frequency in logarithmic
    frequency bins, returning arrays of frequency, fP(f) and the error
    on the mean
    """
    log_freq = N.log10(freq)
    binned_freq = N.arange(round(log_freq[0],1) - bin_size/2.,
                           round(log_freq[len(log_freq)-1],1) + 1 + bin_size/2.,
                           bin_size) # array of frequency bins

    # create empty arrays. index_list will contain indices of freqs to be deleted
    binned_fP, fP_error, index_list = N.array([]), N.array([]), N.array([])
    temp_index = 0 # index where previous iteration of loop stopped

    for n in range(len(binned_freq)):
        temp_fP = N.array([])           # temporary array of fP in freq bin
```

```
        for index in range(temp_index,len(fP)):
            if log_freq[index] > binned_freq[n] + bin_size/2.:
                temp_index = index      # save index for next iteration
                break
            if log_freq[index] >= binned_freq[n] - bin_size/2.:
                temp_fP = N.append(temp_fP, fP[index])
        if len(temp_fP) == 0:
            index_list = N.append(index_list, n)
        else:
            average_fP = N.average(temp_fP)
            binned_fP = N.append(binned_fP, average_fP)
            sigma = N.std(temp_fP)
            # error on the mean
            fP_error = N.append(fP_error, sigma/N.sqrt(float(len(temp_fP))))

    for e in range(len(fP_error)): # use original errors if 1 point in bin
        if fP_error[e] == 0.:
            fP_error[e] = error[e]

    binned_freq = delete_freq(binned_freq, index_list)

    return 10**binned_freq, binned_fP, fP_error


def delete_freq(binned_freq, index_list):
    """
    Function which removes elements from binned_freq with indices in index_list
    """
    for index in index_list[::-1]:                  # reverse array
        binned_freq = N.delete(binned_freq, index)  # delete freq with no power

    return binned_freq


def noise(c_low, c_high, path):
    """
    Function which generates an array of poisson random numbers with
    sigma=dI, and calculates the power spectrum
    """
    # total light curve
    total_lc = N.loadtxt(path+c_low+'-'+c_high+'src.txt', skiprows=3)
    tot_t, tot_i = total_lc[:,0], total_lc[:,2]
    tot_dt = 2*total_lc[0,1]
```

```
    # background light curve
    bg_lc = N.loadtxt(path+c_low+'-'+c_high+'bgd.txt', skiprows=3)
    bg_t, bg_i = bg_lc[:,0], bg_lc[:,2]

    N_tot_i = tot_i * tot_dt # total number of photons (source + background)
    N_tot_av = N.average(N_tot_i)

    N_bg_i = bg_i * tot_dt # number of background photons in same binning
    N_bg_av = N.average(N_bg_i)

    rand_tot = N.zeros(len(tot_i))
    rand_bg = N.zeros(len(tot_i))
    for i in range(len(tot_i)):
        x = poisson.rvs(N_tot_av) # create random poisson numbers
        y = poisson.rvs(N_bg_av)
        rand_tot[i] = (x+y) / tot_dt

    # calculate power spectrum
    freq, tot_n_pow, tnp_err = split_power_spectrum(tot_t,rand_tot,tot_dt)

    # logarithmic bins
    freq, tot_noise_fP, tnpf_err = log_bin(freq, tot_n_pow, tnp_err, 0.05)

    return freq, tot_noise_fP, tnpf_err


def remove_white_noise(freq, fP, fP_error, noise_fP, nfP_error):
    """
    Function to remove the white noise from the power spectrum, and calculate
    the total error
    """
    new_fP = fP - noise_fP                          # subtract noise
    new_fP_error = N.sqrt(fP_error**2 + nfP_error**2)   # calculate error

    return new_fP, new_fP_error


def fractional_variability(freq, fP, fP_error, f_min, f_max):
    """
    Function to find the QPO fractional variability
    """
    # create array of fP(f) centred around QPO
```

```
    peak_fP, peak_freq, peak_err = N.array([]), N.array([]), N.array([])

    for i in range(len(freq)):
        if freq[i] > f_min and freq[i] < f_max:
            peak_fP = N.append(peak_fP, fP[i])
            peak_err = N.append(peak_err, fP_error[i])
            peak_freq = N.append(peak_freq, N.log10(freq[i]))

    integral = I.simps(peak_fP, x=peak_freq) # integrate
    integral_max = I.simps(peak_fP + peak_err, x=peak_freq)
    integral_min = I.simps(peak_fP - peak_err, x=peak_freq)
    errup = integral_max - integral
    errdown = integral - integral_min
    error = N.average((errup, errdown))

    return integral, error


def split_cross(time_h, h, time_s, s, delta_t):
    """
    Function which splits two light curves, h and s, into n_split pieces,
    then calculates the average cross spectrum
    """
    h = bin_128s(time_h, h, delta_t)
    s = bin_128s(time_s, s, delta_t)

    size = len(h)
    x = round(128./delta_t)
    n_split = size/x

    # create empty arrays
    C_real = N.zeros(((x/2)-1)*n_split)
    C_imag = N.zeros(((x/2)-1)*n_split)
    h = h.reshape(n_split, x)
    s = s.reshape(n_split, x)
    C_real.shape = (n_split, abs((x/2)-1))
    C_imag.shape = (n_split, abs((x/2)-1))

    for n in range(int(n_split)):
        H = N.fft.fft(h[n]) # calculate Fourier transform
        S = N.fft.fft(s[n])
        H_freq = N.fft.fftfreq(len(h[n]), delta_t)
        S_freq = N.fft.fftfreq(len(s[n]), delta_t)
```

```
        H_freq, H = positive(H_freq, H) # remove negative frequencies
        S_freq, S = positive(S_freq, S)
        C_real[n], C_imag[n] = cross(h[n], s[n], H, S, H_freq)

    Cf_real_av, Cf_real_err = average_cross(C_real*H_freq)
    Cf_imag_av, Cf_imag_err = average_cross(C_imag*H_freq)

    return H_freq, Cf_real_av, Cf_real_err, Cf_imag_av, Cf_imag_err


def average_cross(C):
    """
    Function which averages the real or imaginary parts of a cross spectrum.
    C is a 2D array of height n_split
    """
    n_split, x = C.shape
    C_av, C_err = N.array([]), N.array([]) # create empty arrays

    for n in range(x):
        C_av = N.append(C_av, N.average(C[:,n]))

        # error on the mean
        C_err = N.append(C_err, (N.std(C[:,n])/N.sqrt(n_split)))

    return C_av, C_err


def cross(h, s, H, S, freq):
    """
    Function to calculate the cross spectrum of the Fourier transforms of two
    light curves, H and S
    """

    C = H.real*S.real + H.imag*S.imag + (H.real*S.imag - H.imag*S.real)*(0+1j)

    # Normalise
    integral = I.simps(C, x=freq)
    s_rms_h = squared_rms_variability(h)
    s_rms_s = squared_rms_variability(s)
    N_const = N.sqrt(s_rms_h * s_rms_s) / integral

    C = C*N_const
```

```
    return C.real, C.imag


def abs_cross(C_r, C_r_err, C_i, C_i_err):
    """
    Function which calculates |C| and its error from the real and imaginary
    parts
    """
    C = N.sqrt(C_r**2 + C_i**2)
    err = N.sqrt((2*C_r*C_r_err)**2 + (2*C_i*C_i_err)**2) / (2*C)

    return C, err


def phase(C_r, C_r_err, C_i, C_i_err):
    """
    Function which calculates the phase shift, and its error, from the
    real and imaginary parts of a cross spectrum
    """
    x = C_i / C_r
    delta = N.arctan(x)
    error = x * N.sqrt((C_i_err/C_i)**2 + (C_r_err/C_r)**2) /(1 + x**2)

    return delta, error


def bin_128s(time, light_curve, delta_t):
    """
    Function which creates 128 second bins, skipping over any gaps in the
    data
    """
    x = round(128/delta_t)
    binned_light_curve = N.array([])
    i_temp = 0      # last index of previous iteration
    while i_temp <= len(light_curve) - x:
        append = True
        for i in range(int(i_temp), int(i_temp+x-1)):
            if time[i+1] - time[i] > 1.5*delta_t:      # check for gap
                i_temp = i+1
                append = False
                break

        if append == True:
```

```
            binned_light_curve = N.append(binned_light_curve,
                                          light_curve[i_temp:i_temp+x])
            i_temp += x

    return binned_light_curve


def plot(freq, fP, fP_error):
    """
    Function which plots a logarithmic power spectrum
    """
    log_fP_error = fP_error / (N.log(10)*fP) # calculate log errors

    P.figure()
    P.errorbar(N.log10(freq), N.log10(fP), yerr=log_fP_error, ecolor='black')
    P.xlabel('log[Frequency (Hz)]')
    P.ylabel('log[Frequency x Power (rms/mean)^2]')
    P.show()


def freq_dep_spec(intensity, fP, fP_err, freq, f_low, f_high):
    """
    Function which calculates the frequency resolved spectrum
    """
    Pow = fP / freq
    Pow_err = fP_err / freq

    powsnip = 0.0
    error = 0.0
    for i in range(len(freq)):
        if freq[i] > f_low and freq[i] < f_high:
            powsnip += Pow[i] * (freq[i]-freq[i-1])
            x = Pow_err[i] * (freq[i]-freq[i-1])
            error += x**2
    error = N.sqrt(error)

    sqrt_err = N.sqrt(powsnip) * 0.5 * error / powsnip

    f_var, err = fractional_variability(freq, fP, fP_err,
                                        f_low, f_high)

    return f_var, err
```

```
def test_negative(fP):
    for i in range(len(fP)):
        if fP[i] < 0:
            # set fP to small, non-zero value
            fP[i] = 1E-6
    return fP

############################ main program ############################

if __name__ == "__main__":

    path = 'data/data_30191-01-30-00/'
    filename = 'rms5_30191-01-30-00_ch022-035_b256_lc.txt'

    light_curve = N.loadtxt(path+filename, skiprows=3) # import data

    time, intensity, i_error = light_curve[:,0], light_curve[:,2], light_curve[:,3]

    delta_t = 2*light_curve[0,1]

    bin_width = 0.05    # width of log freq bins

    # calculate power spectrun
    freq, fP, fP_error = split_power_spectrum(time, intensity, delta_t)

    # create log freq bins
    freq, fP, fP_error = log_bin(freq, fP, fP_error, bin_width)

    # calculate white noise
    freq, noise_fP, noise_fP_err = noise(filename[22:25], filename[26:29],
                                         'data/noise_30188-06-01-00/')

    # remove white noise
    fP, tot_err = remove_white_noise(freq, fP, fP_error, noise_fP,
                                     noise_fP_err)

    # normalise power spectrum
    fP, fP_err = normalise(freq, fP, noise_fP, tot_err, intensity)

    fP = test_negative(fP)

    # plot
```

```
plot(freq, fP, fP_err)
```

cross.py

```
"""
Program to calculate the cross spectrum, phase lag, or time lag between
two light curves
"""

import numpy as N
import power
import pylab as P

def cross_spectrum(s_time, s, h_time, h, delta_t):
    """
    Function which calculates and plots the cross spectrum between a hard, h
    and soft, s, lightcurve
    """
    # calculate cross spectrum
    freq, C_real, C_real_err, C_imag, C_imag_err = power.split_cross(h_time, h,
                                                                     s_time, s,
                                                                     delta_t)
    # create log bins
    f_bin, C_r, C_r_err = power.log_bin(freq, C_real, C_real_err, 0.05)
    f_bin, C_i, C_i_err = power.log_bin(freq, C_imag, C_imag_err, 0.05)

    Cf, err = power.abs_cross(C_r, C_r_err, C_i, C_i_err)

    log_err = err / (Cf * N.log(10))
    P.figure()
    P.errorbar(N.log10(f_bin), N.log10(Cf), yerr=log_err, ecolor='black')
    P.xlabel('log[Frequency (Hz)]')
    P.ylabel('log[Frequency x Cross Spectrum]')
    P.show()

    return f_bin, Cf, err


def lag(s_time, s, h_time, h, delta_t, time_lag=False):
    """
    Function which calculates and plots the phase lag between a hard, h, and
    soft, s, light curve. If time_lag = True, it converts this into a time
    lag
    """
    # calculate cross spectrum
```

```
    freq, C_real, C_real_err, C_imag, C_imag_err = power.split_cross(h_time, h,
                                                                     s_time, s,
                                                                     delta_t)
    # create log bins
    f_bin, C_r, C_r_err = power.log_bin(freq, C_real, C_real_err, 0.05)
    f_bin, C_i, C_i_err = power.log_bin(freq, C_imag, C_imag_err, 0.05)

    x = C_i / C_r
    delta = N.arctan(x)
    delta_err = x * N.sqrt((C_i_err/C_i)**2 + (C_r_err/C_r)**2) /(1 + x**2)

    if time_lag == False:  # plot phase lag
        x = N.array([-2,2])
        y = N.array([0,0])

        P.figure()
        P.errorbar(N.log10(f_bin), delta, yerr=delta_err, ecolor='black')
        P.plot(x,y, color='black', linestyle='dashed')
        P.xlabel('log[Frequency (Hz)]')
        P.ylabel('Phase Lag (rad)')
        P.show()

    if time_lag == True: # plot time lag
        log_f_bin = N.log10(f_bin)
        log_f_err = log_f_bin[len(log_f_bin)-1] - log_f_bin[len(log_f_bin)-2]
        f_bin_err = log_f_err * f_bin * N.log(10)

        lag = delta / (2*N.pi*f_bin)
        lag_err = lag*N.sqrt((delta_err/delta)**2 + (f_bin_err / f_bin)**2)

        log_err = lag_err / (lag * N.log(10))

        P.figure()
        P.errorbar(N.log10(f_bin), N.log10(abs(lag)), yerr=log_err, ecolor='black')
        P.xlabel('log[Frequency (Hz)]')
        P.ylabel('log[Time Lag (s)]')
        P.show()


### main program ###

if __name__ == "__main__":
    path = 'data/data_30188-06-11-00/'
```

```
filename1 = 'rms5_30188-06-11-00_ch000-007_b256_lc.txt'
filename2 = 'rms5_30188-06-11-00_ch022-035_b256_lc.txt'

light_curve1 = N.loadtxt(path+filename1, skiprows=3)
light_curve2 = N.loadtxt(path+filename2, skiprows=3)

s_time, s = light_curve1[:,0], light_curve1[:,2]
h_time, h = light_curve2[:,0], light_curve2[:,2]

delta_t = 2*light_curve1[0,1]

cross_spectrum(s_time, s, h_time, h, delta_t)
```

frs.py

```
"""
Program which calculates the rms spectrum
"""

import power as p
import numpy as N
import os

converter = N.loadtxt('converter.txt') # energy-channel conversion table
abs_chan, std2_chan, energy = converter[:,0], converter[:,1], converter[:,3]

S_E = N.array([]) # array of values of rms
S_E_err = N.array([]) # array of errors
single_bin = N.array([]) # array of energy bins

path = raw_input("path of data: ")
f_low = float(raw_input("low frequency: "))
f_high = float(raw_input("high frequency: "))
savename = 'rms_spectrum_'+path[10:24]+'.txt'

listing = os.listdir(path)
for filename in listing:
    print "current file is: " + filename
    index_l = N.where(abs_chan == float(filename[22:25]))
    index_h = N.where(abs_chan == float(filename[26:29]))
    bin_low = std2_chan[index_l][0]  # convert abs channel numbers to std2
    bin_high = std2_chan[index_h][0]
    bin_low = float(filename[22:25])
    bin_high = float(filename[26:29])

    n = bin_high - bin_low + 1
    for i in range(int(n)):
# split into individual channels
        single_bin = N.append(single_bin, bin_low + i)

    light_curve = N.loadtxt(path+filename, skiprows=3)
    time, intensity, i_error = light_curve[:,0], light_curve[:,2], light_curve[:,3]

    delta_t = 2*light_curve[0,1]    # time bin width
    bin_width = 0.05                # width of log freq bins
```

```
    # calculate power spectrun
    freq, fP, fP_error = p.split_power_spectrum(time, intensity, delta_t)

    # create log freq bins
    freq, fP, fP_error = p.log_bin(freq, fP, fP_error, bin_width)


    # calculate white noise
    freq, noise_fP, noise_fP_err = p.noise(filename[22:25], filename[26:29],
                                        'noise_'+path[5:])

    # remove white noise
    fP, tot_err = p.remove_white_noise(freq, fP, fP_error, noise_fP,
                                       noise_fP_err)

    #normalise
    fP, fP_error =  p.normalise(freq, fP, noise_fP, tot_err, intensity)

    fP = p.test_negative(fP)

    # calculate frequency resolved spectrum
    S, S_err = p.freq_dep_spec(intensity, fP, fP_error, freq, f_low, f_high)

    for i in range(int(n)):
# if multiple energy channels, repeat same value for each channel
        S_E = N.append(S_E, S)
        S_E_err = N.append(S_E_err, S_err)

# save as a .txt file
data = N.zeros(3*len(S_E))
data.shape = (len(S_E),3)
data[:,0], data[:,1], data[:,2] = single_bin, S_E, S_E_err
N.savetxt(savename, data)
print 'saved as', savename
```

spectrum.py

```
"""
Program which multiplies the rms spectrum by the count rate to
calculate the frequency resolved spectrum
"""

import numpy as N

filename = raw_input("input filename: ")

# source count rate
source = N.loadtxt('data/s_'+filename[:14]+'.txt')
chan, scounts, serr = source[:,1], source[:,2], source[:,3]

# background count rate
background = N.loadtxt('data/b_'+filename[:14]+'.txt')
bcounts, berr = background[:,2], background[:,3]

frs = N.loadtxt(filename)
chan2, rms,  rmserr = frs[:,0], frs[:,1], frs[:,2]

totcounts = scounts - bcounts # total count rate
toterr = N.sqrt(serr**2 + berr**2)

S = rms * totcounts[0:len(rms)]
S_err = S * N.sqrt((rmserr/rms)**2 + (toterr[0:len(rms)]/totcounts[0:len(rms)])**2)

data = N.zeros(3*len(chan2), dtype='int32')
data.shape = (len(chan2),3)
data[:,0], data[:,1], data[:,2] = chan2, S, S_err
N.savetxt('rms_'+filename, data)
```